Universidad Latina de Panamá
Facultad de Ingeniería

Asistente Visual Basado en Inteligencia Artificial

Proyecto final de graduación presentado como requisito para optar por el título de Ingeniería de Sistemas Informáticos en la Universidad Latina de Panamá

Rafael Caballero
C. I.: 8-1041-773

Profesor asesor:
Carlos Fernández

Panamá, República de Panamá, 2026

Dedicatoria

A mis padres, por su apoyo incondicional.

Agradecimientos

A todas las personas que me acompañaron y apoyaron durante este proceso.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

El problema

1.1. Antecedentes del problema de investigación

En años recientes, la discapacidad visual ha emergido como un problema de salud pública de creciente relevancia, no solo por su impacto clínico, sino también por las implicaciones sociales, económicas y psicológicas que conlleva. De acuerdo con el informe *Global estimates on the number of people blind or visually impaired by Uncorrected Refractive Error: a meta-analysis from 2000 to 2020*, al menos 3.7 millones de personas a nivel mundial presentaban pérdida total de visión en el año 2020, mientras que 157 millones padecían algún grado de discapacidad visual moderada a severa (DVMS); atribuible a errores de refracción no tratados (ERNT). Estas cifras reflejan un incremento preocupante del 21.8 % al 72 % respecto al año 2000, siendo América Latina y el Caribe las regiones con mayor aumento tanto en ceguera como en DVMS.

Si bien a nivel global se han logrado avances en la reducción de la DVMS por ERNT, las disparidades regionales continúan siendo significativas. Las regiones de medianos y bajos ingresos, especialmente las zonas rurales o con acceso limitado a servicios de salud, presentan una cobertura significativamente inferior en cuanto a atención oftalmológica. En 2021, por ejemplo, la cobertura efectiva de corrección refractiva (eREC) alcanzaba un 79.1 % en superregiones de altos ingresos, en contraste con tan solo un 5.7 % en África Subsahariana y 9.0 % en Asia del Sur. De forma similar, la prevalencia estandarizada por edad de la presbiopía no tratada muestra un contraste marcado entre superregiones, con diferencias que van desde el 2.4 % en países de altos ingresos frente a un 38.9 % en el sur de Asia (Bourne et al., 2024).

Estas desigualdades se deben, en gran medida, a la limitada disponibilidad de servicios de corrección visual accesibles y sostenibles para poblaciones en situación de vulnerabilidad. En estas regiones los recursos destinados a la salud visual suelen ser insuficientes o inexistentes, lo cual impide la detección temprana, el diagnóstico adecuado y el acceso a soluciones ópticas básicas. Este panorama evidencia una brecha estructural en el acceso a la salud visual, con consecuencias directas sobre el bienestar de las personas afectadas.

Más allá de las consecuencias clínicas, la discapacidad visual tiene un profundo impacto en la calidad de vida. Como señala Szekely (Szekely et al., 2025), las personas con discapacidad visual experimentan una disminución significativa en su autonomía, enfrentan mayores barreras para acceder a la educación y al empleo, y presentan un mayor riesgo de aislamiento social, ansiedad y depresión. A medida que se reduce la independencia funcional, también lo hace la autoestima, lo cual repercute en el bienestar emocional y en la capacidad de adaptación a los entornos cotidianos.

Frente a este contexto, las tecnologías de asistencia (TA) emergen como herramientas fundamentales para mitigar los efectos de la discapacidad visual. La Organización Mundial de la Salud define las TA como productos, servicios o sistemas que buscan mantener o mejorar las funciones cognitivas, comunicativas, auditivas, motoras y visuales de las personas, promoviendo así su inclusión, participación y bienestar (*Global report on assistive technology*, 2022). No obstante, la mayoría de estas tecnologías presentan un alto costo económico, lo que limita su accesibilidad para personas de bajos ingresos.

Figura 1.1: Lentes Envision: Read Edition con un costo de 1899.00 USD. Aunque representan una solución efectiva para mejorar la autonomía, su elevado precio limita su acceso en regiones de bajos ingresos.

Un ejemplo concreto es el de los lentes Envision (Figura 1.1), dispositivos inteligentes diseñados entre otras cosas para asistir a personas con discapacidad visual en tareas cotidianas como la lectura, el reconocimiento de objetos o la navegación en entornos físicos. A pesar de sus numerosos beneficios, su precio de 1899.00 USD los convierte en una solución inaccesible para una gran parte de la población afectada, especialmente para aquellos en contextos de pobreza o exclusión social.

En un informe del Parlamento Europeo (European Parliament. Directorate General for Parliamentary Research Services., 2018), se identifican cinco categorías principales de tecnologías de asistencia para personas ciegas o con baja visión: 1) ayudas hápticas (como el bastón blanco, el sistema Braille o las matrices de estímulos táctiles), 2) dispositivos para orientación y movilidad (incluyendo detectores de obstáculos, sistemas de navegación y bastones electrónicos), 3) herramientas para información y comunicación accesible, 4) dispositivos de uso diario (como relojes parlantes o utensilios adaptados), y 5) aplicaciones móviles para lectura, identificación de objetos y localización. Sin embargo, la implementación efectiva de estas herramientas continúa siendo limitada en regiones de escasos recursos (*Global report on assistive technology*, 2022).

Por lo tanto, garantizar el acceso equitativo a estas tecnologías representa no solamente un desafío urgente, sino también una responsabilidad compartida por los sectores público, privado y académico. El desarrollo de soluciones asequibles, sostenibles y adaptadas al contexto sociocultural de las poblaciones más vulnerables es clave para reducir las brechas existentes y contribuir a una mejora sustancial en la calidad de vida de las personas con discapacidad visual.

1.2. Planteamiento del problema

En la actualidad, existen diversas soluciones tecnológicas diseñadas para mejorar la calidad de vida de personas con discapacidad visual, especialmente en lo que respecta a la movilidad, lectura y reconocimiento de objetos en su entorno inmediato. Ejemplo de ello son los lentes inteligentes Envision Ally, cuya funcionalidad fue lanzada en abril de 2025, incorporando el módulo Ask Envision, capaz de describir entornos, leer menús y responder preguntas mediante visión computarizada e inteligencia artificial. De forma similar, la aplicación Seeing AI, desarrollada por Microsoft, permite reconocer tanto personas, como objetos y texto a través de la cámara del dispositivo móvil.

No obstante, si bien estas herramientas representan un significativo avance desde el punto de vista técnico, enfrentan obstáculos importantes en términos de adopción y accesibilidad. Por un lado, el elevado costo económico de las soluciones más sofisticadas representa una barrera infranqueable para los sectores de bajos ingresos. En el caso de los lentes Envision, los precios oscilan entre 1899.00 USD (Read Edition), 2499.00 USD (Home Edition) y 3499.00 USD (Professional Edition); por otro lado, incluso las soluciones basadas en aplicaciones móviles, aunque gratuitas, pueden ser poco accesibles para personas adultas mayores o usuarios con baja alfabetización tecnológica, además de requerir dispositivos móviles compatibles con capacidades mínimas de procesamiento.

1.2.1. Limitaciones de las soluciones basadas en modelos de lenguaje-visión de gran escala

Los avances recientes en modelos de lenguaje-visión de gran escala (Large Vision-Language Models, LVLM) han demostrado capacidades notables para generar descripciones contextuales ricas del entorno, interpretar escenas complejas y responder preguntas sobre contenido visual. Sistemas como GPT-4o, GPT-4o-mini, y Claude 3.5 Sonnet han sido integrados exitosamente en dispositivos de asistencia visual, proporcionando retroalimentación contextual sofisticada que supera ampliamente las capacidades de modelos de visión tradicionales (Baig et al., 2024; Tokmurziyev et al., 2025; Zaman et al., 2025).

Baig et al., 2024 demostraron que la integración de GPT-4o-mini en un sistema *wearable* permite alcanzar precisiones del 91.8 % en escenarios abiertos mediante descripciones contextuales generadas dinámicamente. De manera similar, Zaman et al., 2025 aprovecharon los Meta Ray-Bans en conjunto con modelos multimodales para crear una plataforma extensible que facilita tareas de descripción de escenas, detección de objetos y reconocimiento óptico de caracteres. Tokmurziyev et al., 2025 implementaron un sistema que combina YOLO-World para detección de objetos con GPT-4o para razonamiento contextual, logrando tasas de precisión del 81.5 % incluso en escenarios con obstáculos dinámicos.

No obstante, estos avances presentan limitaciones críticas en términos de accesibilidad económica y viabilidad técnica para poblaciones de bajos recursos, precisamente aquellas donde la prevalencia de discapacidad visual es más alta. Las principales barreras identificadas son:

- **Dependencia de conectividad:** los LVLM más avanzados operan mediante APIs en la nube, requiriendo conexión constante a internet para funcionar. Esta dependencia resulta problemática en múltiples dimensiones. Primero, las regiones con mayor prevalencia de discapacidad visual frecuentemente coinciden con áreas de

infraestructura de telecomunicaciones limitada o inexistente. Segundo, incluso en zonas urbanas con cobertura, la necesidad de conectividad constante compromete la autonomía del dispositivo y su funcionalidad en espacios cerrados (túneles, edificios con mala recepción) o en situaciones de emergencia donde la red puede estar saturada o inoperante.

- **Costos operativos recurrentes:** a diferencia de los modelos ejecutados localmente, el uso de APIs de LVLM implica costos recurrentes por cada consulta realizada. Aunque proveedores como OpenAI ofrecen modelos más económicos (GPT-4o-mini), el costo acumulado puede ser significativo. Considerando que un usuario con discapacidad visual podría requerir decenas o cientos de consultas diarias para tareas de navegación, lectura e identificación de objetos, los costos operativos mensuales podrían oscilar entre 10 y 50 USD, dependiendo del volumen de uso. Esta estructura de costos recurrentes representa una barrera económica insostenible para poblaciones de bajos ingresos, donde el ingreso per cápita puede ser inferior a $2 USD diarios en regiones de extrema pobreza.

- **Requisitos computacionales para *hardware* propietario:** algunos sistemas como los Meta Ray-Bans o dispositivos similares que integran LVLM requieren *hardware* propietario con costos que oscilan entre 299 USD (Meta Ray-Bans básicos) hasta más de 3000 USD para soluciones especializadas. Zaman et al., 2025 reconocen explícitamente esta limitación al señalar que, aunque su plataforma WhatsAI democratiza el desarrollo de aplicaciones de accesibilidad, el punto de entrada —los propios lentes— representa una barrera económica significativa para usuarios de bajos recursos.

- **Latencia operativa:** las consultas a APIs en la nube introducen latencias inherentes debido al tiempo de transmisión de datos (upload de imagen, procesamiento remoto, download de respuesta). Aunque Baig et al., 2024 reportan latencias de 1.5 segundos en condiciones óptimas de iluminación, estas pueden incrementarse significativamente en condiciones de conectividad deficiente, comprometiendo la viabilidad del sistema para navegación en tiempo real donde decisiones de seguridad deben tomarse en fracciones de segundo.

- **Privacidad y soberanía de datos:** el envío constante de imágenes capturadas del entorno del usuario a servidores externos plantea preocupaciones legítimas sobre privacidad, especialmente considerando que estas imágenes pueden contener información sensible del entorno doméstico, documentos personales, rostros de familiares y otras situaciones donde el consentimiento informado de terceros no es factible. Este aspecto adquiere mayor relevancia en contextos de poblaciones vulnerables, donde la alfabetización digital y comprensión de riesgos de privacidad puede ser limitada.

Estas limitaciones resultan especialmente preocupantes al considerar que la prevalencia de la discapacidad visual es significativamente mayor en regiones con contextos socioeconómicos desfavorables. En estas poblaciones, donde el acceso a servicios de salud visual ya es limitado, la adquisición de soluciones tecnológicas costosas o con costos operativos recurrentes resulta inviable, lo que perpetúa condiciones de exclusión, dependencia y baja calidad de vida.

En contraste con las alternativas de alto costo, existen dispositivos especializados en tareas concretas, como los sistemas TTS (Text-To-Speech), que transforman texto impreso en audio. Si bien modelos como el OrCam MyEye tienen precios superiores a 3500.00 USD, han comenzado a surgir versiones más asequibles en el mercado global, con precios que rondan los 35.00 USD. Estos dispositivos, aunque más limitados en funcionalidad, demuestran que la especialización en un tipo de procesamiento específico puede reducir significativamente los costos sin eliminar por completo la utilidad del producto para el usuario final.

En este contexto, resulta pertinente preguntarse ¿es viable el desarrollo de un dispositivo de asistencia visual basado en inteligencia artificial, centrado en una funcionalidad específica, que opere mediante inferencia local y esté orientado a poblaciones de bajos ingresos?, ¿qué nivel de precisión sería técnicamente alcanzable sin comprometer la asequibilidad del producto ni incurrir en costos operativos recurrentes?, ¿puede una herramienta de estas características incidir de forma positiva en la calidad de vida de los individuos con discapacidad visual?

Los avances recientes en tecnologías de microprocesadores y microcontroladores de bajo consumo, como los integrados en plataformas como el ESP32 o Raspberry Pi Zero, permiten el desarrollo de dispositivos funcionales con costos de producción que representan apenas entre el 1.32 % y el 2.63 % del valor comercial de la edición más económica de los lentes Envision. Es decir, se proyecta un rango de precios entre 50.00 y 90.00 USD.

El impacto potencial de un dispositivo dentro de este rango de precios no debe subestimarse. Enfocar sus capacidades en el reconocimiento de elementos del entorno —señales de tránsito, puertas, objetos cotidianos o zonas de peligro— podría mejorar significativamente la movilidad autónoma de personas con discapacidad visual, especialmente en espacios públicos. A su vez, este aumento en la autonomía podría fomentar una mayor participación en actividades sociales, reducir el aislamiento, mejorar la autoestima y, en última instancia, contribuir a una mejor salud mental.

Por tanto, el desarrollo de un dispositivo de asistencia visual asequible, especializado, basado en inteligencia artificial con inferencia local y sin dependencia de conectividad o costos recurrentes, no solo representa un desafío tecnológico, sino que constituye una oportunidad para cerrar brechas sociales y promover la inclusión efectiva de personas con discapacidad visual en contextos de alta vulnerabilidad.

1.3. Justificación de la investigación

La presente investigación propone el desarrollo de un dispositivo *wearable*, de bajo costo, orientado a mejorar la calidad de vida de personas con discapacidad visual mediante la implementación de un sistema de asistencia visual basado en inteligencia artificial. El dispositivo consistirá en un prototipo de gafas inteligentes con cámara integrada, procesador embebido y capacidad de procesamiento en tiempo casi real (o real, en caso de usarse en conjunto con la aplicación móvil), orientado específicamente al reconocimiento de señales de tráfico en entornos urbanos.

Se ha seleccionado el microcontrolador ESP32-S3 (fabricado por Espressif Systems), integrado en la placa XIAO de SeeedStudio, debido a sus ventajas en términos de tamaño reducido, bajo consumo energético, conectividad integrada (Wi-Fi/Bluetooth), capacidad para ejecutar modelos de inferencia ligeros y compatibilidad con herramientas como TensorFlow Lite Micro. Estas características hacen de esta plataforma una alternativa ideal

para aplicaciones edge AI de bajo presupuesto, permitiendo la ejecución del modelo de reconocimiento directamente en el dispositivo, sin depender de una conexión a internet ni de servidores externos.

1.3.1. Ventajas del enfoque basado en inferencia local

En contraste con los enfoques basados en LVLM descritos anteriormente, el presente proyecto adopta una estrategia de inferencia completamente local mediante modelos optimizados de TensorFlow Lite ejecutados en *hardware* de bajo costo. Esta decisión de diseño se fundamenta en varios principios:

1. **Eliminación de costos recurrentes**: al ejecutar modelos localmente, se elimina por completo la dependencia de APIs de pago, reduciendo el costo total de propiedad (TCO) a la inversión inicial en *hardware*, que se proyecta entre 50.00 y 90.00 USD.

2. **Operación offline**: el sistema mantiene funcionalidad completa sin conectividad a internet, garantizando autonomía en entornos rurales, edificios con mala recepción, o situaciones de emergencia.

3. **Latencia predecible**: la ejecución local con aceleración por NPU permite alcanzar latencias inferiores a 1 segundo para modelos ligeros, con rendimiento que no se degrada por condiciones de red.

4. **Privacidad por diseño**: en el modo local, ninguna imagen capturada abandona el dispositivo ESP32-S3, garantizando privacidad absoluta. En el modo híbrido, las imágenes se transmiten localmente al dispositivo móvil del usuario vía Wi-Fi sin pasar por servidores externos, manteniendo la privacidad dentro del ecosistema personal del usuario.

5. **Escalabilidad sostenible**: el costo por usuario es fijo e independiente del volumen de uso, permitiendo implementaciones masivas en contextos educativos o programas gubernamentales de asistencia sin preocupaciones sobre costos operativos crecientes.

Si bien esta aproximación implica limitaciones en la sofisticación de las descripciones contextuales comparado con LVLM de última generación, el balance entre accesibilidad económica, autonomía operativa y privacidad resulta más apropiado para el contexto socioeconómico objetivo. La especialización en tareas específicas (OCR, detección de obstáculos, reconocimiento de señales) mediante modelos optimizados permite alcanzar precisiones comparables a sistemas basados en LVLM para estos casos de uso concretos, mientras se mantiene dentro de restricciones económicas viables para poblaciones de bajos recursos.

Adicionalmente, el diseño híbrido propuesto —que permite operar tanto en modo local autónomo como en modo híbrido con aplicación móvil— ofrece un camino de actualización gradual para usuarios que posteriormente adquieran dispositivos móviles con mayor capacidad computacional, sin dejar obsoleta la inversión inicial en el *hardware wearable*.

Si bien existen precedentes en el desarrollo de dispositivos similares para personas con discapacidad visual —como lo demuestran diversas investigaciones que exploran el uso de visión artificial y microcontroladores para lectura de texto, identificación de obstáculos o reconocimiento de objetos (Ariza & Pearce, 2022; Li et al., 2023; Salazar & Mesa,

2019)—, el presente proyecto se distingue por su enfoque específico en la detección de señales de tráfico. Esta especialización busca abordar un vacío importante en las tecnologías de asistencia existentes, que suelen concentrarse en tareas como lectura de texto o reconocimiento facial, pero no contemplan de forma explícita la interpretación de señales viales, las cuales son fundamentales para garantizar la seguridad y la movilidad autónoma en entornos exteriores.

Esta capacidad de reconocimiento de señales de tráfico no solo amplía el alcance funcional del dispositivo, sino que tiene un impacto directo en la autonomía y seguridad personal de los usuarios. A diferencia del bastón blanco —herramienta tradicional utilizada por personas con discapacidad visual—, este dispositivo permitiría detectar elementos no tangibles o situados fuera del alcance inmediato, como señales de advertencia, tránsito peatonal o cruces peligrosos. De esta forma, se promueve una experiencia de movilidad más informada, segura e independiente.

Adicionalmente, el diseño modular y de bajo costo del prototipo favorece su replicabilidad y escalabilidad, lo que lo convierte en una alternativa viable para su implementación en regiones con recursos limitados; el dispositivo cuenta además con la función de integrarse con una aplicación móvil de manera opcional con el fin de recibir información del entorno a tiempo real (en caso de contar con un dispositivo capaz de satisfacer la demanda de procesamiento). Esta accesibilidad económica, sumada a su especialización funcional, fortalece la pertinencia social y tecnológica del proyecto, justificando la realización de esta investigación como un aporte significativo en la búsqueda de soluciones inclusivas, eficaces y sostenibles para personas con discapacidad visual.

1.4. Objetivos

1.4.1. Objetivo general

- Desarrollar un sistema embebido, *wearable*, con el principal objetivo de salvaguardar la integridad física de los individuos con discapacidad visual.

1.4.2. Objetivos específicos

- Crear un programa para el ESP32 S3 capaz de capturar imágenes mediante una cámara integrada con el fin de enviarla a la aplicación móvil en intervalos menores a 2 segundos, optimizando en la medida de lo posible el consumo energético.
- Entrenar un modelo para que, en conjunto con un programa para el ESP32 S3 y TensorFlow Lite Micro, sea capaz de detectar de manera independiente señales y luces de tráfico con una precisión mínima de 90 %.
- Desarrollar una aplicación móvil, multiplataforma, basada en Flutter, integrada con TensorFlow para el procesamiento de entornos con una precisión mínima de 90 %.
- Integrar los componentes del sistema (cámara, microcontrolador, aplicación móvil y salida de audio) para informar al usuario sobre su entorno mediante alertas de audio con una latencia inferior a 2 segundos.
- Programar un sistema integrado, configurable a través de la aplicación móvil que permita personalizar las alertas según las necesidades del usuario.

- Garantizar la facilidad de actualizaciones del sistema.

1.5. Alcance y límites de la investigación

1.5.1. Alcance

Esta investigación se centra en el desarrollo de un sistema embebido *wearable* de bajo costo para asistir a personas con discapacidad visual, específicamente en la detección de señales de tráfico y entornos. La población objetivo incluye adultos con baja visión o ceguera total. Las variables principales son la precisión de la detección de objetos (mínimo 90 %) y la latencia de las alertas de audio (menor a 2 segundos), evaluadas mediante la integración de un prototipo basado en el microcontrolador ESP32 S3 y una aplicación móvil. Las pruebas se realizarán en un entorno controlado, como un laboratorio con simulaciones de entornos urbanos, durante un periodo de 8 meses, garantizando la viabilidad técnica y la accesibilidad del dispositivo para comunidades de bajos ingresos.

1.5.2. Límites

El proyecto está limitado por la capacidad de procesamiento del microcontrolador ESP32 S3, que restringe la detección a un conjunto específico de objetos (señales y luces de tráfico), excluyendo la identificación de un amplio rango de elementos en tiempo real, lo que requeriría de un microcontrolador (o microprocesador) más potente pero menos modular. Por restricciones de tiempo, no se implementarán funcionalidades adicionales, como podrían ser el soporte adicional del dispositivo para la asistencia de individuos con trastorno del espectro autista (TEA) o traducción de lenguaje de señas, que exigirían modelos de inteligencia artificial diferentes y pruebas extensivas. Asimismo, por consideraciones éticas, las pruebas no incluirán menores de edad ni datos personales sensibles, y los resultados serán aplicables solo a entornos controlados, no a escenarios reales complejos.

1.6. Línea de investigación a la que pertenece el estudio

El estudio se enmarca en la línea de desarrollo de *software* y sistemas y ciencias de la computación, con un enfoque específico en visión por computadora y tecnologías de asistencia. El proyecto aborda el desarrollo de un sistema embebido, *wearable* que utiliza algoritmos de visión artificial para detectar señales de tráfico y obstáculos, contribuyendo al diseño de interfaces accesibles y soluciones de bajo costo para personas con discapacidad visual.

Capítulo 2

Marco teórico

2.1. Antecedentes de investigaciones realizadas en el tema

En los últimos años, se ha observado un creciente interés por el desarrollo de tecnologías de asistencia orientadas a personas con discapacidad visual, especialmente mediante dispositivos portables, embebidos, de bajo costo. Muchas de estas investigaciones han coincidido en el uso de microprocesadores como Raspberry Pi y microcontroladores como el ESP32 Cam, combinados con modelos externos o ligeros de inteligencia artificial y sensores diversos. Por ejemplo, Ali Hassan y Tang, 2016 exploran el uso de un Raspberry Pi 2 junto con Tesseract OCR, con el objetivo de asistir a estudiantes con discapacidad visual en la lectura de texto impreso. Aunque el enfoque se centra principalmente en el reconocimiento óptico de caracteres, se señala la posibilidad de extender las funcionalidades del sistema mediante la incorporación de modelos adicionales, lo que abre la puerta al diseño modular de soluciones personalizables.

En un enfoque similar, la investigación *AI Powered Smart Glass for Visually Impaired Individuals* (V M et al., 2024) presenta una solución basada en ESP32 Cam y sensores ultrasónicos, logrando un 92 % de precisión en detección de objetos y un 88 % en reconocimiento de texto. Lo más notable es la validación del sistema con usuarios reales, quienes reportaron una mejora significativa —de hasta el 85 %— en precisión en comparación con métodos tradicionales como el bastón blanco. Estos hallazgos subrayan que incluso con recursos computacionales limitados es posible construir soluciones funcionales y bien recibidas por el público objetivo.

Otros desarrollos han enfatizado aún más la experiencia del usuario. Por ejemplo, en la propuesta de Fernández (Fernandez et al., 2024), se utilizó un Arduino Nano con sensores ultrasónicos y retroalimentación háptica, logrando una recepción positiva entre los usuarios pese a ciertos errores ocasionales atribuibles a las limitaciones del *hardware*. Esta investigación refuerza la idea de que, aún con restricciones tecnológicas, la utilidad de un dispositivo puede mantenerse si este se ajusta bien a las necesidades específicas de su público objetivo.

Por otro lado, hay propuestas más ambiciosas que integran conectividad remota o incluso el uso de modelos fundacionales como GPT-4o a través de APIs. Tal es el caso de *Aeye – A Smart Glasses* (Kumar Jha et al., 2023), cuyo sistema envía los datos captados por la cámara a un servidor externo que ejecuta MobileNet SSD para identificar obstáculos en el entorno. De forma semejante, Baig et al., 2024 combinan MobileNet con GPT-4o

(a través de la API de Azure), integrándolo en una gorra equipada con auriculares de conducción ósea. Estos permiten mantener libre el canal auditivo, una decisión técnica importante al considerar que las personas con discapacidad visual dependen fuertemente de la audición. Además, este sistema permite a los usuarios añadir nuevos datos —como rostros u objetos— a la base de conocimiento, ampliando progresivamente la utilidad del dispositivo.

Más recientemente, Tokmurziyev et al., 2025 propusieron un diseño basado en retroalimentación háptica con 13 patrones distintos y conexión a GPT-4o, alcanzando una precisión promedio del 81.3 %. Sin embargo, al igual que en los casos anteriores, la dependencia de conectividad externa plantea interrogantes sobre la escalabilidad y sostenibilidad del modelo, especialmente en contextos donde el acceso a internet es limitado o intermitente.

Así, el panorama de soluciones tecnológicas puede clasificarse, a grandes rasgos, en tres categorías: 1) sistemas que dependen de procesamiento externo (nube o APIs propietarias), que ofrecen un alto grado de sofisticación, pero menor autonomía; 2) dispositivos de procesamiento local especializado, menos versátiles pero más accesibles y 3) modelos híbridos que combinan capacidades autónomas con conectividad local para extender funcionalidad sin depender completamente de servicios remotos. La presente investigación es pues parte de esta última categoría, en respuesta tanto a los retos económicos como de infraestructura que enfrentan muchas de las regiones con alta prevalencia de discapacidad visual.

2.2. Bases teóricas que sustentan la investigación

La fundamentación técnica de esta investigación se apoya en el creciente cuerpo de literatura que demuestra la viabilidad de desarrollar soluciones de asistencia visual basadas en tecnologías de bajo costo, sin sacrificar funcionalidad esencial. En particular, se retoma la línea planteada en *Smart Glass System for Visually Impaired Using ESP32 Camera and OCR* («Smart Glass System for Visually Impaired Using ESP32 Camera and OCR», 2025), que demuestra cómo un sistema simple —compuesto por una ESP32, sensor ultrasónico, lector de huellas, sensor PIR y sensor de llama— puede integrarse con tecnologías cotidianas como aplicaciones móviles para configurar alertas y retroalimentación. Aunque esta propuesta se enfoca más en la seguridad ocupacional que en la movilidad personal, representa un ejemplo claro del potencial de integración entre *hardware* embebido y plataformas móviles.

A un nivel más completo, la investigación *Drishti: Visual Navigation Assistant for Visually Impaired* (Joshi et al., 2023) ofrece un modelo más afín al que aquí se propone. Drishti combina lectura de texto, reconocimiento de señales de tráfico, identificación de objetos y hasta detección de billetes, convirtiendo los resultados en comandos de voz a través de gTTS. Esta solución destaca por su versatilidad, sin una dependencia crítica de servicios externos, basándose principalmente en la conexión a un dispositivo móvil local.

Específicamente en el ámbito del reconocimiento de señales de tráfico para personas con discapacidad visual, LYTNet (Yu et al., 2019a) representa un precedente técnico fundamental. Esta red neuronal convolucional fue diseñada para el reconocimiento en tiempo real de semáforos peatonales y cruces de cebra, logrando precisiones del 94.5 % en la detección de semáforos y 97 % en el reconocimiento de cruces peatonales, con tiempos de inferencia inferiores a 100ms en dispositivos embebidos (Yu et al., 2019b). La arquitectura

ligera de LYTNet demuestra que es posible alcanzar altos niveles de precisión en tareas especializadas de detección vial sin requerir modelos de gran escala o *hardware* costoso. El proyecto ImVisible, basado en LYTNet, ejemplifica la implementación práctica de estos modelos en dispositivos portátiles de bajo costo, utilizando el dataset Pedestrian Traffic Lights (PTL) con más de 5,000 imágenes anotadas. Este trabajo establece un precedente directo para la viabilidad de sistemas especializados en detección de señales de tráfico ejecutados localmente en *hardware* de recursos limitados.

La presente investigación busca entonces posicionarse en un punto medio entre la especialización y la funcionalidad, entre la conectividad y la autonomía. Se propone un sistema que aproveche la creciente capacidad de los microcontroladores modernos, como el ESP32-S3, para ejecutar modelos de inferencia ligeros, al mismo tiempo que se integra con dispositivos móviles para facilitar tareas tanto de configuración como de expansión funcional. Este enfoque híbrido permite reducir costos sin comprometer la utilidad del dispositivo, haciéndolo ideal para contextos donde la conectividad no es confiable o el presupuesto es limitado.

Por tanto, esta propuesta no solo responde a una necesidad técnica, sino también a una necesidad social: la de ofrecer herramientas que verdaderamente puedan ser adoptadas por las comunidades que más lo requieren. Y si bien la precisión y el alcance funcional pueden estar limitados por el *hardware*, un diseño enfocado y bien integrado tiene el potencial de mejorar significativamente la autonomía, seguridad y bienestar de las personas con discapacidad visual.

2.3. Variables

2.3.1. Definición conceptual de las variables

- Precisión en la detección de objetos: se refiere a la capacidad del modelo de realizar predicciones correctas, medida mediante métricas como *accuracy* (exactitud global), *precision* por clase (TP/(TP+FP)) y *recall* por clase (TP/(TP+FN)).
- Latencia de las alertas: la cantidad de tiempo total que transcurre entre las interacciones de los componentes del dispositivo.
- Facilidad de uso: engloba la experiencia de usuario, abarcando aspectos como la complejidad en el uso del dispositivo o la configuración del mismo a través de la aplicación móvil.

2.3.2. Definición operacional de las variables

- Precisión en la detección de objetos: se evaluará con base en la cantidad de aciertos del modelo entrenado.
- Latencia de las alertas: se calcula con base en el tiempo que transcurre desde la entrada que desencadena una predicción hasta la retroalimentación proporcionada al usuario.
- Facilidad de uso: se medirá con base en una encuesta de satisfacción tomando en cuenta aspectos como:
    - Utilidad del dispositivo (contribuye a mejorar el desplazamiento)

- Precisión (tasa de aciertos/fallos)
- Nivel de asistencia percibido (en comparación con las alternativas convencionales)

2.3.3. Mapa de variable(s)

Objetivo general	Desarrollar un sistema embebido, *wearable*, con el principal objetivo de salvaguardar la integridad física de los individuos con discapacidad visual.		
Objetivos específicos	**Variable(s)**	**Dimensiones**	**Indicadores**
Entrenar un modelo para que, en conjunto con un programa para el ESP32 S3 y TensorFlow Lite Micro, sea capaz de detectar de manera independiente señales y luces de tráfico con una precisión mínima de 90 %.	Precisión	Exactitud del reconocimiento	Porcentaje de aciertos frente a errores (mínimo 90 %)
Desarrollar una aplicación móvil, multiplataforma, basada en Flutter, integrada con TensorFlow para el procesamiento de entornos con una precisión mínima de 90 %.	Precisión	Exactitud del procesamiento de imagen	Porcentaje de identificación correcta de obstáculos o señales
Crear un programa para el ESP32 S3 capaz de capturar imágenes mediante una cámara integrada con el fin de enviarlas a la aplicación móvil en intervalos menores a 2 segundos.	Latencia	Tiempo de transmisión de datos	Tiempo promedio entre captura y recepción de imagen
Integrar los componentes del sistema para informar al usuario mediante alertas de audio con una latencia inferior a 2 segundos.	Latencia	Tiempo de respuesta del sistema	Tiempo desde la detección hasta la alerta auditiva
Programar un sistema integrado, configurable desde la aplicación móvil, que permita personalizar las alertas según las necesidades del usuario.	Facilidad de uso	Configuración personalizada	Número de opciones disponibles y tiempo medio de configuración
Garantizar la facilidad de actualizaciones del sistema.	Facilidad de uso	Mantenimiento y escalabilidad	Frecuencia y simplicidad en la instalación de actualizaciones

Cuadro 2.1: Mapa de variables.

2.4. Glosario de términos

1. API (Application Programming Interface): conjunto de definiciones y protocolos que permiten la comunicación e integración entre diferentes aplicaciones de *software*. En el contexto de esta investigación, posibilita el acceso a modelos avanzados como GPT-4o desde dispositivos externos.

2. Arduino Uno: placa de desarrollo basada en el microcontrolador ATmega328P. Es una de las versiones más populares de Arduino, utilizada para prototipado de sistemas electrónicos y educativos por su sencillez y amplia comunidad de soporte.

3. Arduino Nano: versión reducida y más compacta del Arduino Uno, también basada en el ATmega328P. Diseñada para proyectos en los que se requiere poco espacio físico manteniendo la misma funcionalidad básica.

4. Auriculares de conducción ósea: dispositivo de audio que transmite vibraciones a través de los huesos del cráneo hacia el oído interno, permitiendo escuchar sin obstruir el canal auditivo. Son especialmente útiles para personas con discapacidad visual, ya que no interfieren con la percepción ambiental.

5. Azure: plataforma de servicios en la nube de Microsoft que ofrece infraestructura, herramientas de desarrollo, inteligencia artificial y servicios de IoT, permitiendo la implementación y escalado de soluciones digitales.

6. Bastón blanco: herramienta tradicional de movilidad utilizada por personas con discapacidad visual para detectar obstáculos físicos en su entorno cercano.

7. Blynk: plataforma de desarrollo IoT que permite crear aplicaciones móviles y web para controlar, monitorear y gestionar dispositivos conectados, facilitando la interacción con microcontroladores y placas como Arduino o ESP32.

8. DVMS (discapacidad visual moderada a severa): condición que limita significativamente la capacidad visual de una persona, pero que no implica ceguera total.

9. Edge AI: rama de la inteligencia artificial que ejecuta modelos de inferencia directamente en dispositivos locales (microcontroladores, cámaras, sensores) sin depender de servidores en la nube.

10. ESP-CAM: módulo de bajo costo basado en el microcontrolador ESP32 que incluye conectividad Wi-Fi, Bluetooth y una cámara integrada. Se utiliza para aplicaciones de visión artificial e IoT en escenarios de bajo consumo y sin necesidad de *hardware* costoso.

11. ESP32-S3: microcontrolador de bajo consumo energético, desarrollado por Espressif Systems, con capacidades de conectividad (Wi-Fi y Bluetooth) y soporte para procesamiento de modelos de IA ligeros, ideal para dispositivos portátiles de bajo costo.

12. GPT-4o: modelo de lenguaje multimodal de OpenAI, capaz de procesar texto e imágenes en tiempo real, utilizado en sistemas de asistencia para interpretar el entorno y responder a consultas.

13. Háptica (retroalimentación háptica): tecnología que genera estímulos táctiles para transmitir información al usuario, como vibraciones o patrones de presión, muy útil en dispositivos de asistencia para personas con discapacidad visual.

14. IA (Inteligencia Artificial): conjunto de técnicas y algoritmos que permiten a una máquina realizar tareas que requieren procesos cognitivos humanos, como reconocimiento de imágenes, procesamiento de lenguaje natural y toma de decisiones.

15. IoT (Internet of Things): ecosistema de dispositivos físicos conectados entre sí y a internet, capaces de recolectar e intercambiar datos. En este caso, se aplica a dispositivos de asistencia para personas con discapacidad visual.

16. Lentes TTS (Text-to-Speech): dispositivos que capturan texto mediante una cámara y lo convierten en audio para facilitar la lectura a personas con discapacidad visual.

17. LLM (Large Language Model): modelo de lenguaje a gran escala entrenado con grandes volúmenes de datos, diseñado para procesar y generar lenguaje natural. Se utiliza en aplicaciones de inteligencia artificial como chatbots, asistentes virtuales y análisis de texto.

18. Microcontrolador: circuito integrado que contiene un procesador, memoria y periféricos de entrada/salida, diseñado para ejecutar tareas específicas en sistemas embebidos.

19. Microprocesador: unidad de procesamiento central de un dispositivo electrónico, con mayor capacidad que un microcontrolador, utilizada para ejecutar aplicaciones más complejas.

20. MobileNet SSD: modelo ligero de redes neuronales convolucionales diseñado para reconocimiento de objetos en tiempo real, optimizado para dispositivos con recursos limitados.

21. OCR (Optical Character Recognition): tecnología que convierte texto impreso o manuscrito en datos digitales editables mediante el uso de algoritmos de visión artificial.

22. OrCam MyEye: dispositivo de asistencia visual portátil basado en IA, que reconoce texto, rostros y objetos, pero cuyo elevado costo lo hace inaccesible para gran parte de la población con discapacidad visual.

23. PIR (Passive Infrared Sensor): sensor infrarrojo pasivo que detecta la radiación infrarroja emitida por objetos y personas, comúnmente utilizado en sistemas de seguridad y automatización.

24. Presbiopía: discapacidad visual relacionada con la edad que dificulta el enfoque de objetos cercanos debido a la pérdida de elasticidad del cristalino.

25. Raspberry Pi: minicomputadora de bajo costo y tamaño reducido, capaz de ejecutar un sistema operativo completo (generalmente Linux). Se emplea en proyectos de programación, robótica, IoT y prototipado, ofreciendo mayor capacidad de procesamiento que microcontroladores como Arduino.

26. Retroalimentación auditiva: provisión de información mediante sonido, utilizada en dispositivos de asistencia para guiar al usuario durante la navegación o identificación de objetos.

27. SSD MobileNet: arquitectura de red neuronal convolucional optimizada para tareas de detección de objetos en tiempo real, con un balance entre precisión y eficiencia computacional.

28. TensorFlow: biblioteca de código abierto desarrollada por Google para el aprendizaje automático y el aprendizaje profundo (deep learning). Se utiliza ampliamente en visión por computadora, procesamiento de lenguaje natural y aplicaciones de inteligencia artificial en dispositivos locales o en la nube.

29. Tesseract OCR: motor de reconocimiento óptico de caracteres de código abierto, utilizado para digitalizar texto impreso y convertirlo en datos editables.

30. Text-to-Speech (TTS): tecnología que convierte texto digital en voz hablada mediante algoritmos de síntesis de voz.

31. Visión artificial: campo de la inteligencia artificial que permite a los sistemas interpretar y procesar información visual a partir de imágenes o videos, replicando capacidades del ojo humano mediante algoritmos y modelos de aprendizaje.

32. Wearable: dispositivo electrónico que puede llevarse puesto en el cuerpo, como relojes, pulseras o gafas inteligentes, diseñado para proporcionar funciones asistivas o de monitoreo.

Capítulo 3

Marco metodológico

3.1. Tipo y diseño de la investigación

Esta investigación es de tipo **experimental aplicada**, con enfoque **cuantitativo** y alcance **descriptivo-evaluativo**. Su objetivo principal es validar técnicamente un prototipo *wearable* de bajo costo para asistencia visual mediante pruebas controladas en laboratorio.

3.1.1. Clasificación metodológica

- **Naturaleza:** investigación aplicada y tecnológica
- **Enfoque:** cuantitativo (basado en métricas objetivas de rendimiento)
- **Alcance:** descriptivo-evaluativo (caracteriza y valida desempeño técnico)
- **Diseño:** experimental con mediciones repetidas bajo condiciones controladas
- **Temporalidad:** transversal (recolección de datos en periodo definido)

3.1.2. Características del diseño experimental

El diseño experimental contempla:

1. **Objeto de estudio:** prototipo integrado de *hardware* y *software*
2. **Variables controladas:** distancia, iluminación, tipo de objeto
3. **Variables medidas:** precisión, latencia, robustez operacional
4. **Condiciones:** ambiente controlado de laboratorio
5. **Replicabilidad:** protocolos estandarizados documentados

Nota importante: este estudio NO involucra investigación con sujetos humanos. El objeto de análisis es exclusivamente el sistema tecnológico desarrollado.

3.1.3. Modalidades operativas del prototipo

El sistema se implementa en dos arquitecturas complementarias:

Modo local (Autónomo)

Características técnicas:

- Procesamiento íntegro en microcontrolador ESP32-S3
- Inferencia con modelo cuantizado (TensorFlow Lite Micro)
- Generación de alertas desde archivos almacenados en microSD
- Autonomía completa, sin dependencia de internet
- Salida de audio vía interfaz I2S o puerto auxiliar

Flujo operacional:

Captura → preproceso → inferencia → generación audio → reproducción

Métricas esperadas:

- Latencia estimada: 300–450 ms
- Precisión esperada: 88–93 %
- Ventajas: independencia, privacidad, bajo costo operativo
- Limitaciones: capacidad computacional restringida

Modo híbrido (Integración con aplicación móvil)

Características técnicas:

- Captura en ESP32-S3, transmisión vía Wi-Fi a dispositivo móvil
- Procesamiento distribuido: análisis avanzado en Flutter + TensorFlow Lite
- Mayor capacidad de procesamiento, modelos más grandes
- Configuración remota de alertas y parámetros
- Síntesis TTS dinámica

Submodos operativos del modo híbrido:
El modo híbrido integra tres submodos especializados que pueden operar de forma independiente o combinada:

1. **Submodo OCR (Reconocimiento de texto):**
   - Implementación: Flutter Tesseract OCR v0.4.30
   - Función: lectura de texto en señales, menús, documentos
   - Precisión esperada: 70–85 %
   - Latencia: 200–500 ms
2. **Submodo colisiones (Detección de obstáculos por profundidad):**

- Modelo: Depth Anything V2 Metric Indoor Small
- Función: estimación de distancia a obstáculos mediante mapa de profundidad
- Precisión esperada: 40–60 % (limitada para decisiones críticas)
- Latencia: 200–300 ms
- **Nota:** alto porcentaje de fallos esperado; incluido para validación comparativa

3. **Submodo descriptor (Descripción de elementos del entorno):**
   - Modelo: DeepLabV3-Plus MobileNet (Qualcomm)
   - Dataset: Pascal VOC 2012 (20 clases: persona, bicicleta, auto, motocicleta, autobús, tren, camión, ave, gato, perro, caballo, oveja, vaca, botella, silla, sofá, mesa, planta, monitor, portátil)
   - Función: identificación y segmentación de objetos cotidianos presentes en el entorno
   - Precisión esperada: 75–85 %
   - Latencia: 180–250 ms
   - **Limitación:** no incluye infraestructura urbana (aceras, calzadas, semáforos)

Figura 3.1: Interfaz principal de la aplicación móvil desarrollada en Flutter. La pantalla permite activar los tres submodos del modo híbrido (scene, OCR, *depth*), visualizar resultados de inferencia en tiempo real y configurar parámetros de conectividad Wi-Fi con el ESP32-S3.

Flujo operacional:

Captura (ESP32) → transmisión Wi-Fi → selección de submodo → procesamiento (App) → TTS → audio

Métricas esperadas (general):

- Latencia estimada: 600–900 ms (varía según submodo)
- Precisión: depende del submodo activo (40–92 %)
- Ventajas: mayor precisión (OCR/scene), configuración flexible, múltiples funcionalidades
- Limitaciones: requiere conectividad local, mayor latencia, precisión limitada en *depth*

3.1.4. Arquitecturas del sistema

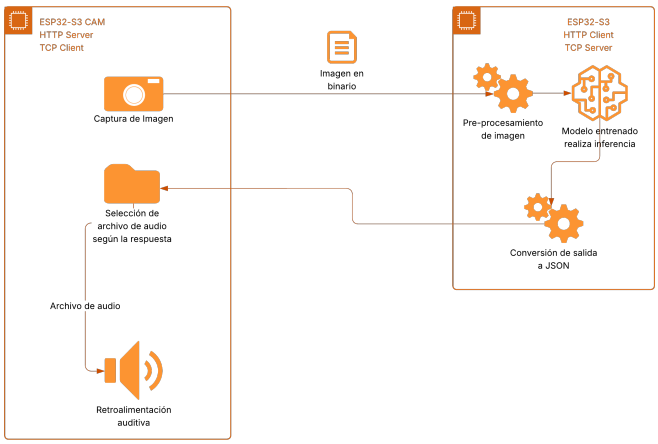


Figura 3.2: Arquitectura del modo local (procesamiento embebido).

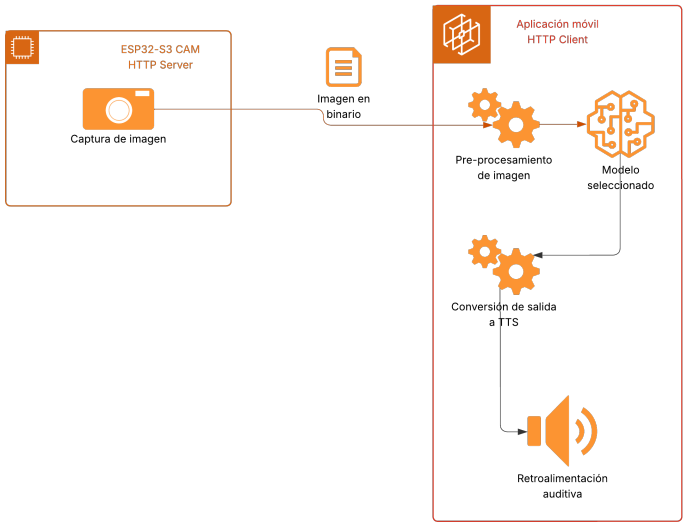


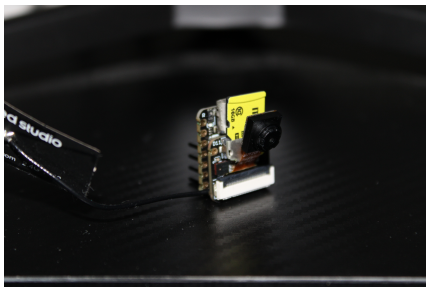
Figura 3.3: Arquitectura del modo híbrido (procesamiento distribuido).

3.2. Objeto de estudio y ambiente experimental

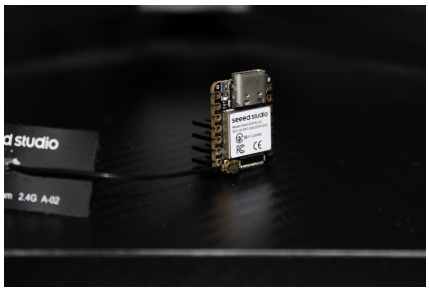
3.2.1. Componentes del prototipo

El sistema integrado está compuesto por los siguientes elementos:

***hardware* principal**



(a) ESP32-S3 XIAO Sense con cámara OV2640 integrada (21×17.8×15 mm)



(b) ESP32-S3 XIAO sin cámara (21×17.8 mm)

Figura 3.4: Microcontroladores utilizados en el prototipo. (a) ESP32-S3 Sense funciona como unidad de captura y transmisión en ambas modalidades. (b) ESP32-S3 *standalone* opera como servidor de inferencia exclusivamente en modo local.

Microcontrolador principal: XIAO ESP32-S3 Sense

Componente	**Especificación**
Microcontrolador	XIAO ESP32-S3 Sense (Seeed Studio)
Procesador	Dual-core Xtensa 32-bit LX7 @ 240 MHz
Memoria	8MB PSRAM + 8MB Flash
Cámara	OV2640 (2MP, 160° FOV, 3.6mm focal)
Conectividad	Wi-Fi 6 (802.11ax), Bluetooth 5.0
Almacenamiento	microSD 16GB (Clase 10, UHS-1)
Alimentación	USB-C 5V/2A
Salida audio	I2S digital + puerto auxiliar 3.5mm (requiere módulo DAC externo para auriculares analógicos)
Consumo activo	54.4 mA @ 3.8V (consumo del chip ESP32-S3 después de regulación desde 5V USB-C)
Consumo sleep	3.98 mA @ 3.8V
Rango temperatura	-40°C a 65°C
Dimensiones	21 × 17.8 × 15 mm (con módulo expansión)
Costo unitario	USD $19–24

Cuadro 3.1: Especificaciones técnicas del *hardware* principal.

Componentes adicionales del prototipo

Microcontrolador auxiliar: XIAO ESP32-S3 (standalone)

Característica	**Especificación**
Fabricante	Seeed Studio
Procesador	Dual-core Xtensa 32-bit LX7 @ 240 MHz
Memoria	8MB PSRAM + 8MB Flash
Conectividad	Wi-Fi 6 (802.11ax), Bluetooth 5.3
Consumo activo	54.4 mA @ 3.8V (consumo del chip ESP32-S3 después de regulación desde 5V USB-C)
Consumo sleep	3.98 mA @ 3.8V
Dimensiones	21 × 17.8 mm
Costo unitario	USD $6–8
Uso en proyecto	Prototipo sin cámara integrada (requiere módulo externo)

Cuadro 3.2: Especificaciones del ESP32-S3 *standalone*.

Cámara: OV2640

Característica	**Especificación**
Tipo sensor	CMOS 1/4”
Resolución máxima	2 MP (1600 × 1200 píxeles)
Resolución para inferencia	640 × 480 píxeles (VGA)
Ángulo de visión	160° (horizontal)
Distancia focal	3.6 mm (fijo)
Interfaz	DVP (Digital Video Port)
Frame rate	15 fps @ 2MP, 30 fps @ VGA
Consumo	50 mA @ 3.3V durante captura
Costo unitario	USD $5–8 (si se compra por separado)

Cuadro 3.3: Especificaciones de la cámara OV2640.

Almacenamiento: microSD Card

Característica	Especificación
Capacidad	16 GB
Velocidad	Clase 10, UHS-1
Función	Almacenamiento de logs JSON y archivos de audio
Velocidad lectura	90 MB/s
Velocidad escritura	45 MB/s
Costo unitario	USD $3–6

Cuadro 3.4: Especificaciones de almacenamiento microSD.

Salida de audio

Característica	Especificación
Tipo recomendado	Auriculares de conducción ósea
Ventaja	No obstaculiza percepción auditiva ambiental
Frecuencia	20–20,000 Hz
Impedancia	16–32 Ω
Costo unitario	USD $15–30
Alternativa de bajo costo:	
Tipo	Auriculares estándar 3.5mm
Costo unitario	USD $5–10
Interfaz I2S (integrada en ESP32-S3):	
Función	Salida de audio digital hacia DAC o amplificador
Ventaja	Bajo costo, integración directa
Limitación	Requiere módulo DAC externo para auriculares analógicos
Costo módulo DAC/I2S	USD $0.90–2.00

Cuadro 3.5: Opciones de salida de audio.

Justificación de alimentación por USB-C:

Se optó por alimentación continua mediante USB-C para simplificar el prototipo y garantizar estabilidad operacional durante las pruebas de laboratorio. Esta decisión permite:

- Eliminar variables de consumo energético durante validación técnica
- Garantizar voltaje y corriente estables (5V/2A)
- Facilitar sesiones de prueba prolongadas
- Reducir complejidad en gestión de energía
- Concentrar esfuerzos en validación de precisión y latencia

Nota: futuras iteraciones del producto comercial podrán integrar batería Li-Po para operación portátil. La arquitectura actual permite esta expansión sin modificaciones sustanciales.

Software y frameworks

Firmware embebido (ESP32-S3)

- **Plataforma:** ESP-IDF v5.1 (basado en FreeRTOS)
- **Lenguaje:** C/C++
- **IDE:** Arduino IDE 2.3.3
- **Librerías principales:**
  - `esp_camera.h` – Control de cámara OV2640
  - `WiFi.h` – Conectividad inalámbrica
  - `WebServer.h` – Servidor HTTP local
  - `ArduinoJson.h` – Serialización de datos
  - TensorFlow Lite Micro – Motor de inferencia

Aplicación móvil (Flutter)

- **Framework:** Flutter 3.29.3 (Dart)
- **Plataformas:** Android 8.0+, iOS 12.0+
- **Dependencias principales:**
  - `tflite_flutter` – Inferencia en dispositivo móvil
  - `flutter_tts` – Síntesis de voz
  - `flutter_tesseract_ocr` – Reconocimiento de texto
  - `http` – Comunicación con ESP32
  - `provider` – Gestión de estado

Modelos de inteligencia artificial

(a) Red

(b) Green

(c) Countdown Green

(d) Countdown Blank

(e) None

Figura 3.5: Ejemplos representativos del dataset Pedestrian-Traffic-Lights (PTL) utilizado para entrenamiento y validación del modelo LytNet V1. Las imágenes fueron capturadas desde la perspectiva de peatones en condiciones urbanas reales con variaciones de iluminación, ángulo y distancia.

Modelo	**Función**	**Tamaño**	**Precisión esperada**	**Modalidad**
LytNet V1 INT8	Segmentación semántica de semáforos y pasos de cebra	2.10 MB	90 % (accuracy)	Local
DeepLabV3-Plus MobileNet	Segmentación semántica (Pascal VOC)	18 MB	75–85 %	Híbrido (Descriptor)
Depth-Anything V2 Metric Indoor Small	Estimación métrica de profundidad	16 MB	40–60 %	Híbrido (Colisiones)
Tesseract OCR v4	Reconocimiento de texto	32 MB	70–85 %	Híbrido (OCR)

Cuadro 3.6: Modelos de IA implementados.

Justificación de modelos seleccionados

- **LytNet V1:** el modelo de segmentación semántica y clasificación de semáforos utiliza la arquitectura LytNet V1 (Yu et al., 2019a), una red neuronal convolucional ligera diseñada específicamente para asistencia a peatones con discapacidad visual. El modelo fue entrenado sobre el dataset Pedestrian-Traffic-Lights (PTL), que contiene 5059 imágenes anotadas de semáforos peatonales y pasos de cebra capturadas

desde la perspectiva de peatones en condiciones urbanas reales. Se optó por LytNet V1 en lugar de la versión V2 debido a su menor complejidad computacional, lo que resulta ideal para el despliegue en microcontroladores de recursos limitados como el ESP32-S3. El modelo cuantizado en formato INT8 tiene un tamaño de aproximadamente 2.10 MB, permitiendo su carga y ejecución eficiente en dispositivos embebidos.

Arquitectura y características: LytNet V1 emplea una arquitectura encoder-decoder que realiza simultáneamente dos tareas: 1) detección y segmentación de pasos de cebra mediante máscaras binarias y 2) clasificación semántica del estado del semáforo peatonal en cuatro clases: *Red* (rojo), *Green* (verde), *Countdown Green* (cuenta regresiva verde), y *Countdown Blank* (cuenta regresiva vacía/transición). La arquitectura utiliza convoluciones separables en profundidad y bloques residuales inversos optimizados para reducir operaciones de punto flotante (FLOPs).

Entrenamiento y desempeño: el modelo fue entrenado hasta la época 300, alcanzando un *accuracy* del 90 % sobre el conjunto de validación sin mostrar señales de plateau. Según los autores originales (Yu et al., 2019a), el modelo puede alcanzar las siguientes métricas por clase con entrenamiento extendido:

Métrica	**Red**	**Green**	**Countdown Green**	**Countdown Blank**
Precision	0.97	0.94	0.99	0.86
Recall	0.96	0.94	0.96	0.92

Cuadro 3.7: Precision y Recall por clase reportados por los autores de LytNet V1.

Estas métricas sugieren que con entrenamiento adicional más allá de la época 300, el modelo podría alcanzar un *accuracy* superior al 95 % en condiciones óptimas, aunque para efectos de este prototipo se utiliza el modelo entrenado hasta época 300 (90 % *accuracy*).

- **DeepLabV3-Plus MobileNet (Qualcomm):** arquitectura encoder-decoder con Atrous Spatial Pyramid Pooling (ASPP) para segmentación semántica por píxel (Chen et al., 2018). El modelo utiliza el dataset Pascal VOC 2012 (Everingham et al., 2010), que contiene 20 clases de objetos comunes (persona, vehículos, animales, objetos cotidianos) pero **no incluye elementos de infraestructura urbana** como aceras, calzadas o señalización vial. Por tanto, su función en este prototipo es la descripción general de elementos presentes en el entorno (personas, vehículos, objetos), no la navegación específica por zonas urbanas.

- **Depth-Anything V2 Metric Indoor Small:** modelo compacto de estimación métrica de profundidad monocular basado en arquitectura transformer optimizada (Yang et al., 2024). A diferencia de los modelos de profundidad relativa, esta variante métrica está entrenada específicamente para entornos interiores y proporciona estimaciones de distancia en unidades reales (metros).

  Limitación importante: los mapas de profundidad generados presentan baja precisión (40–60 %) para decisiones de seguridad críticas. Se incluye con propósitos de detección básica de colisiones y validación comparativa, demostrando que sensores láser ToF son alternativa superior para aplicaciones donde la precisión es crítica.

- **Flutter Tesseract OCR:** motor de reconocimiento óptico de caracteres integrado mediante el paquete `flutter_tesseract_ocr` v0.4.30 (Smith, 2007). Aunque la precisión es moderada (70–85 %), resulta suficiente para casos de uso donde el texto complementa información visual (ej: "ALTO", "CEDA EL PASO", lectura de menús).

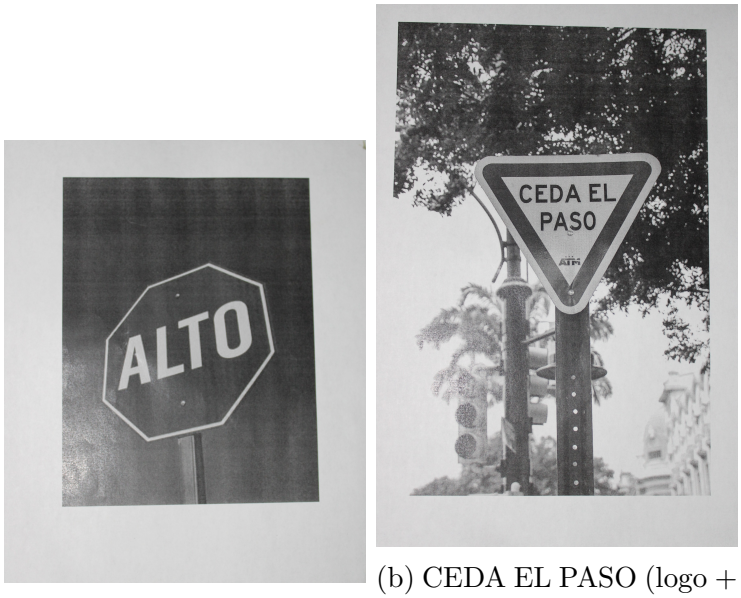
3.2.2. Ambiente de pruebas controlado

Las pruebas se ejecutan en laboratorio con las siguientes características:

Característica	**Especificación**
Espacio físico	5m × 5m (mínimo)
Iluminación artificial	LED regulable 5500K
Superficies	Mesa de trabajo 0.75m altura, piso liso
Temperatura	26–30°C natural
Humedad	60–80 % RH
Equipamiento	Cinta métrica

Cuadro 3.8: Especificaciones del ambiente de laboratorio.

Elementos de prueba



(a) Señal ALTO (30° inclinación)

(b) CEDA EL PASO (logo + árbol y elementos varios de fondo)

Figura 3.6: Elementos de prueba utilizados en Escenario 3 (submodos especializados). (a) y (b) evalúan el submodo OCR bajo condiciones de complejidad creciente.

- **Dataset PTL:** imágenes del conjunto de prueba del dataset Pedestrian-Traffic-Lights
- **Objetos Pascal VOC:** maniquí, silla, botella (para submodo descriptor)

- **Señales con texto:**
  - Señal simple con texto claro
  - Señal de "ALTO" con inclinación de 30°
  - Señal de "CEDA EL PASO" en entorno realista (árbol detrás, con logo)
- **Obstáculos:** caja de cartón, puerta simulada, pared portable (para submodo colisiones)
- **Distancias estandarizadas:** 0.5m, 1m, 1.5m, 2m
- **Condiciones lumínicas:** LED 5500K (blanco frío)

3.3. Variables de investigación

3.3.1. Variables independientes (manipuladas)

1. **Modo operativo:** validación dataset vs. Flujo ESP32 completo vs. modo híbrido
2. **Distancia al objeto:** 0.5m, 1m, 1.5m, 2m (para submodos híbridos)
3. **Iluminación:** LED 5500K
4. **Tipo de objeto:** objetos Pascal VOC, obstáculos variados
5. **Complejidad de escena:** objeto único, múltiples objetos

3.3.2. Variables dependientes (medidas)

Variable	Unidad	Definición operacional	Método de medición
Accuracy	%	$\frac{\text{Predicciones correctas}}{\text{Total predicciones}} \times 100$	Comparación con ground truth
Precision	%	$\frac{\text{TP}}{\text{TP + FP}} \times 100$	Por clase (Red, Green, etc.)
Recall	%	$\frac{\text{TP}}{\text{TP + FN}} \times 100$	Por clase (Red, Green, etc.)
Latencia	ms	$T_{\text{respuesta}} - T_{\text{inicio}}$	Timestamps en logs
Robustez	%	$\frac{\text{Fallos}}{\text{Total pruebas}} \times 100$	Registro de *crashes/timeouts*

Cuadro 3.9: Variables dependientes y métodos de medición.

3.3.3. Variables controladas (constantes)

- Mismo *hardware* en todas las pruebas (ESP32-S3 XIAO Sense y ESP32-S3 XIAO regular)

- Misma versión de firmware y aplicación móvil
- Protocolo estandarizado de comunicación Wi-Fi
- Modelo LytNet V1 INT8 cuantizado
- Temperatura ambiente (26–30 °C)
- Iluminación LED 5500K

3.4. Hipótesis del sistema

3.4.1. Hipótesis general

- H_0 **(Nula):** el prototipo NO es capaz de detectar señales de tráfico con precisión $\geq 90\,\%$ O NO mantiene latencia < 2 segundos en condiciones controladas.
- H_1 **(Alternativa):** el prototipo SÍ es capaz de detectar señales con precisión $\geq 90\,\%$ Y mantiene latencia < 2 segundos.

3.4.2. Hipótesis específicas

#	**Aspecto**	H_1 **(esperado)**	**Criterio de validación**
1	Precisión local	$\geq 90\,\%$	$\frac{\text{Aciertos}}{\text{Total}} \times 100 \geq 90$
2	Latencia local	< 2000 ms	Media de latencias < 2000 ms
3	Precisión híbrido	$\geq 90\,\%$	$\frac{\text{Aciertos}}{\text{Total}} \times 100 \geq 90$
4	Latencia híbrido	< 2000 ms	Media de latencias < 2000 ms
5	Robustez operacional	Fallos $\leq 5\,\%$	$\frac{crashes}{\text{Total}} \times 100 \leq 5$

Cuadro 3.10: Hipótesis específicas y criterios de validación.

3.5. Diseño experimental y protocolos de prueba

3.5.1. Estructura general de pruebas

El plan experimental se organiza en 4 escenarios complementarios:

Escenario	Objetivo	Duración	Variables evaluadas	N.° pruebas
1	Validación sobre dataset PTL	1h	Accuracy, Precision, Recall	1
2	Validación flujo ESP32 completo	1h	Latencia, Accuracy	1
3	Submodos especializados (modo híbrido)	2h	Descriptor + OCR + Colisiones	15
4	Pruebas de estrés	1.5h	Robustez, estabilidad	15
	TOTAL	**5.5h**		**32 pruebas**

Cuadro 3.11: Estructura del plan experimental.

Desglose por modo operativo:

- **Validación Dataset:** 1 sesión de evaluación (739 imágenes test)
- **Validación Flujo ESP32:** 1 sesión de prueba (60 imágenes)
- **Modo híbrido - Submodo Descriptor:** 6 pruebas (identificación de objetos)
- **Modo híbrido - Submodo OCR:** 3 pruebas (reconocimiento de texto en entornos realistas)
- **Modo híbrido - Submodo Colisiones:** 6 pruebas (estimación métrica de profundidad)
- **Pruebas de estrés:** 15 pruebas (continuidad, alternancia y recuperación de errores)

3.5.2. Escenario 1: validación sobre dataset PTL (baseline en PC)

Objetivo: validar el desempeño del modelo LytNet V1 INT8 cuantizado sobre el conjunto completo de prueba del dataset Pedestrian-Traffic-Lights en un entorno computacional sin restricciones, estableciendo métricas baseline de *accuracy*, *precision* y *recall* por clase que servirán como referencia para evaluar el desempeño del sistema embebido.

Configuración:

- **Plataforma de inferencia:** PC de laboratorio (CPU/GPU)
- **Modelo:** LytNet V1 INT8 cuantizado (2.10 MB)
- **Framework:** TensorFlow Lite Interpreter (Python)
- **Dataset:** 739 imágenes del split de test de PTL
- **Clases evaluadas:** red, green, countdown green, countdown blank, none

Protocolo de ejecución:

1. Cargar modelo LytNet V1 INT8 (.tflite) en entorno Python
2. Cargar las 739 imágenes del test split del dataset PTL
3. Ejecutar script de evaluación automatizada que:
   - Lee cada imagen del test split
   - Aplica el preprocesamiento requerido (resize, normalización)
   - Realiza inferencia con TensorFlow Lite Interpreter
   - Compara predicción contra ground truth del dataset (archivo de anotaciones)
4. Calcular métricas globales y por clase
5. Generar matriz de confusión 4×4

Datos registrados:

- Archivo de imagen evaluada
- Clase ground truth
- Clase predicha
- Probabilidad/confianza
- Tiempo de inferencia (ms)
- Resultado binario (acierto/error)

Métricas calculadas:

- **Accuracy global:** $\frac{\text{Predicciones correctas}}{739} \times 100$
- **Precision por clase:** $\frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}$ para $i \in \{\text{Red, Green, CG, CB}\}$
- **Recall por clase:** $\frac{\text{TP}_i}{\text{TP}_i+\text{FN}_i}$ para $i \in \{\text{Red, Green, CG, CB}\}$
- **F1-Score por clase:** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$
- **Matriz de confusión 4×4**

Criterio de éxito:

- Accuracy global $\geq 90\,\%$
- Precision promedio $\geq 0{,}85$
- Recall promedio $\geq 0{,}85$

3.5.3. Escenario 2: validación del flujo completo ESP32-S3

Objetivo: validar el funcionamiento del prototipo completo simulando el flujo operativo: captura de imagen en ESP32-S3 CAM → transmisión Wi-Fi → inferencia en ESP32-S3 servidor → retorno de resultados → recepción en ESP32-S3 CAM.

Arquitectura del sistema:

- **ESP32-S3 XIAO Sense (CAM):** captura/lee imagen de tarjeta SD, transmite vía Wi-Fi, recibe y registra resultados
- **ESP32-S3 XIAO (Servidor):** recibe imagen, ejecuta inferencia con LytNet V1, retorna predicción
- **Comunicación:** Wi-Fi 2.4 GHz (protocolo HTTP/TCP)

Configuración:

- **Imágenes de prueba:** 60 imágenes seleccionadas aleatoriamente del conjunto completo de 739 imágenes del test split
- **Almacenamiento:** imágenes precargadas en tarjeta microSD del ESP32-S3 CAM
- **Ground truth:** CSV con archivo y clase esperada para cada imagen

Protocolo de ejecución:

1. Cargar 60 imágenes de test en microSD del ESP32-S3 CAM
2. Iniciar ESP32-S3 servidor con modelo LytNet V1 cargado
3. Establecer conexión Wi-Fi entre ambos dispositivos
4. Ejecutar script automatizado en ESP32-S3 CAM que:
    - Lee imagen N de la microSD
    - Registra timestamp de inicio (T_{inicio})
    - Transmite imagen vía Wi-Fi al servidor
    - Espera respuesta con predicción
    - Registra timestamp de recepción (T_{fin})
    - Registra resultado en log
5. Repetir para las 60 imágenes
6. Consolidar logs y calcular métricas

Datos registrados por imagen:

- Imagen
- Timestamp inicio (T_{inicio})
- Timestamp fin (T_{fin})
- Latencia total (ms)

- Latencia de inferencia en servidor (ms)
- Clase real
- Clase predicha
- Confianza (%)
- Resultado binario (acierto/error)
- Notas/observaciones

Métricas calculadas:

- **Latencia total:** media, desviación estándar, mín., máx., P95
- **Latencia por componente:** desglose de tiempos
- **Accuracy:** $\frac{\text{Predicciones correctas}}{60} \times 100$
- **Tasa de fallos de comunicación:** *timeouts*, desconexiones

Criterios de éxito:

- Latencia total media < 2000 ms
- Accuracy $\geq 90\,\%$ (consistente con Escenario 1)
- Tasa de fallos de comunicación $\leq 5\,\%$

Análisis complementario:

- Identificar cuellos de botella en latencia (transmisión Wi-Fi vs. inferencia)
- Evaluar estabilidad de conexión Wi-Fi
- Comparar *accuracy* entre Escenario 1 (evaluación directa) y Escenario 2 (flujo completo)

3.5.4. Escenario 3: submodos especializados del modo híbrido

Este escenario evalúa las capacidades específicas de los tres submodos del modo híbrido, que requieren mayor capacidad computacional y solo están disponibles mediante procesamiento en la aplicación móvil.

Escenario 3A: submodo descriptor (Identificación de objetos)

Objetivo: evaluar la capacidad de DeepLabV3-Plus MobileNet para identificar y segmentar objetos cotidianos del dataset Pascal VOC 2012 presentes en escenas simuladas.

Variables manipuladas - Desglose integral:

- Tipo de objeto: persona (maniquí), silla, botella
- Distancia: 1m, 2m
- Iluminación: LED 5500K

Cálculo de pruebas: 3 objetos × 2 distancias × 1 iluminación = **6 pruebas**
Protocolo de ejecución:

1. Posicionar objeto a distancia especificada
2. Configurar iluminación LED (5500K)
3. Activar prototipo en modo híbrido (submodo descriptor)
4. Capturar imagen, transmitir vía Wi-Fi a app móvil
5. Ejecutar inferencia con DeepLabV3-Plus en dispositivo móvil
6. Registrar objetos detectados y confianza
7. Verificar manualmente corrección de detección
8. Documentar resultado y observaciones

Datos registrados por prueba:

- Timestamp ISO 8601
- Objeto(s) detectado(s) por el modelo
- Confianza por objeto (%)
- Latencia total (ms)
- Latencia por componente (captura, transmisión Wi-Fi, inferencia, TTS)
- Resultado: correcto / parcial / error
- Notas/observaciones

Métricas específicas:

- Objetos detectados vs. objetos presentes
- Precisión de clasificación por objeto
- Clasificación cualitativa: alta (>80 %), media (60–80 %), baja (<60 %)

Limitación conocida: el modelo no reconoce infraestructura urbana (aceras, calzadas, semáforos) ya que Pascal VOC 2012 se enfoca en objetos cotidianos.

Escenario 3B: submodo OCR (reconocimiento de texto)

Objetivo: evaluar capacidad de Flutter Tesseract OCR para reconocer texto en señales bajo condiciones de complejidad creciente, incluyendo escenarios realistas con oclusiones parciales y señalización con logos.
Variables manipuladas - Desglose integral:

- **Prueba OCR-1:** señal simple con texto claro (“VELOCIDAD MÁXIMA 40”) a 0.5m, iluminación LED 5500K, ángulo 0°

- **Prueba OCR-2:** señal de "ALTO" a 0.5m, iluminación LED 5500K, ángulo 30° (inclinación deliberada)
- **Prueba OCR-3:** señal de "CEDA EL PASO" con logo, entorno realista (árbol detrás, oclusión parcial), 0.5m, iluminación LED 5500K

Cálculo de pruebas: 3 configuraciones de complejidad creciente = **3 pruebas**
Protocolo de ejecución:

1. Posicionar señal según especificaciones de cada prueba
2. Configurar iluminación LED y ángulo correspondiente
3. Activar prototipo en modo híbrido (submodo OCR)
4. Capturar imagen, transmitir vía Wi-Fi a app móvil
5. Ejecutar Tesseract OCR en dispositivo móvil
6. Registrar texto reconocido y confianza
7. Calcular precisión por carácter
8. Documentar resultado y observaciones

Datos registrados por prueba:

- Timestamp ISO 8601
- Tipo de señal y configuración de escena
- Texto esperado (ground truth)
- Texto reconocido por OCR
- Ángulo de captura aplicado
- Presencia de oclusiones o elementos de fondo
- Temperatura de color LED (5500K)
- Confianza (%)
- Latencia total (ms)
- Precisión por carácter (%)
- Clasificación: exacto (100 %) / parcial (50–99 %) / erróneo (<50 %)
- Notas/observaciones

Expectativas realistas:

- Precisión esperada: 70–85 % (moderada)
- El OCR es sensible a: ángulo, iluminación, fuente tipográfica, oclusiones

- Propósito: complementario, NO crítico para seguridad

Métricas específicas:

- Precisión por carácter: $\text{Similitud}(a, b) = \left(1 - \frac{\text{lev}_{a,b}(|a|,|b|)}{\text{máx}(|a|,|b|)}\right) \times 100\,\%$
- Clasificación de resultado: exacto (100 %), parcial (50–99 %), erróneo (<50 %)
- Análisis cualitativo de factores que afectan reconocimiento

Escenario 3C: submodo colisiones (estimación métrica de profundidad)

Objetivo: documentar las limitaciones del modelo Depth-Anything V2 Metric Indoor Small y justificar la necesidad de sensores láser ToF en futuras versiones.

Variables manipuladas - Desglose integral:

- Tipo de superficie: plana, irregular, oscura
- Distancia real: 1m, 2m
- Iluminación: LED 5500K

Cálculo de pruebas: 3 superficies × 2 distancia × 1 iluminación = **6 pruebas**

Protocolo de medición:

1. Posicionar obstáculo a distancia medida con metro (±1 cm)
2. Registrar distancia real como ground truth
3. Configurar iluminación LED 5500K
4. Activar prototipo en modo híbrido (Submodo Colisiones)
5. Capturar imagen, transmitir vía Wi-Fi a app móvil
6. Ejecutar modelo Depth-Anything V2 Metric Indoor Small
7. Extraer distancia estimada del mapa de profundidad métrico
8. Calcular error absoluto: $|\text{Real} - \text{Estimada}|$
9. Calcular error relativo: $\frac{|\text{Real}-\text{Estimada}|}{\text{Real}} \times 100$
10. Clasificar resultado según criterios establecidos

Datos registrados por prueba:

- Timestamp ISO 8601
- Tipo de superficie evaluada
- Distancia real medida (cm)
- Distancia estimada por modelo (cm)
- Error absoluto (cm)

- Error relativo (%)
- Latencia total (ms)
- Clasificación: acierto (<20 %) / error parcial (20–50 %) / fallo (50 %)
- Notas/observaciones

Criterios de clasificación:

- Acierto: error relativo <20 %
- Error parcial: 20 % error <50 %
- Fallo: error 50 %

Expectativas realistas:

- Tasa de aciertos esperada: 40–60 %
- Tasa de fallos esperada: 40–60 %
- Los mapas de profundidad monoculares NO son confiables para decisiones de seguridad
- Conclusión anticipada: sensores láser ToF ($5 USD) son alternativa superior

3.5.5. Escenario 4: pruebas de estrés operacional

Objetivo: validar robustez del sistema bajo operación continua, alternancia entre submodos y recuperación ante errores.

Pruebas de continuidad

Validación flujo ESP32

- **Prueba 4.1:** 60 inferencias consecutivas sin reinicio
- **Prueba 4.2:** 30 inferencias con monitoreo de temperatura

Datos registrados: latencia cada 10 inferencias, temperatura ESP32 cada 10 mín. (máx. 65 °C), *crashes/timeouts*, estabilidad escritura microSD.
Cálculo: 2 sesiones = **2 pruebas**

Modo híbrido - Submodo Descriptor

- **Prueba 4.3:** 50 capturas consecutivas
- **Prueba 4.4:** 30 capturas con monitoreo Wi-Fi

Datos registrados: latencia promedio, estabilidad Wi-Fi, desconexiones.
Cálculo: 2 sesiones = **2 pruebas**

Modo híbrido - Submodo OCR

- **Prueba 4.5:** 30 capturas consecutivas de texto
- **Prueba 4.6:** monitoreo uso memoria app móvil

Datos registrados: tasa reconocimiento, uso RAM app.
Cálculo: 2 sesiones = **2 pruebas**

Modo híbrido - Submodo Colisiones

- **Prueba 4.7:** 30 capturas consecutivas
- **Prueba 4.8:** documentación de fallos del modelo

Datos registrados: estimaciones válidas, *crashes*, errores.
Cálculo: 2 sesiones = **2 pruebas**

Pruebas de alternancia entre submodos

- **Prueba 4.9:** 10 ciclos: descriptor → OCR → colisiones → descriptor
- **Prueba 4.10:** 10 ciclos: OCR → colisiones → descriptor → OCR
- **Prueba 4.11:** 10 ciclos: colisiones → descriptor → OCR → colisiones
- **Prueba 4.12:** 10 ciclos: cambio aleatorio entre los 3 submodos
- **Prueba 4.13:** 10 ciclos: alternancia rápida (<5s entre cambios)

Datos registrados: cambio correcto de modelo, latencia switching, estabilidad de memoria.
Cálculo: 5 sesiones de 10 ciclos cada una = **5 pruebas**

Pruebas de recuperación de errores

- **Prueba 4.14:** desconexión Wi-Fi intencional + reconexión automática (3 repeticiones)
- **Prueba 4.15:** apagado/encendido de la app móvil durante operación (3 repeticiones)

Datos registrados: tiempo de detección, tiempo de reconexión, estabilidad post-reconexión, pérdida de datos.
Cálculo: 2 tipos de eventos × 3 repeticiones cada uno = **2 pruebas** (contadas como sesiones)
Total Escenario 4: 2 (Flujo ESP32) + 2 (Descriptor) + 2 (OCR) + 2 (Colisiones) + 5 (Alternancia) + 2 (Recuperación) = **15 pruebas**

3.6. Procedimiento de recolección y análisis de datos

3.6.1. Fase 1: ejecución de pruebas (Semanas 2–4)

Preparación del ambiente

- Calibrar cámara OV2640 (balance de blancos, enfoque fijo a 1m)
- Verificar conexión USB-C estable (5V/2A) en ambos ESP32-S3
- Formatear microSD y cargar 60 imágenes de test + CSV de ground truth
- Compilar firmware con logging habilitado para ambos dispositivos
- Cargar modelo LytNet V1 INT8 (2.10 MB) en ESP32-S3 servidor
- Configurar Access Point y servidor TCP para comunicación inter-ESP32
- Instalar aplicación móvil en dispositivo de prueba
- Configurar y calibrar sistema de iluminación LED (5500K)

Ejecución de pruebas

1. Configurar escenario según protocolo
2. Activar registro de logs en microSD (ESP32-S3 CAM)
3. Activar registro de logs en ESP32-S3 servidor
4. Ejecutar secuencia según escenario (evaluación dataset, flujo completo, o submodo)
5. Verificar resultados en tiempo real
6. Documentar observaciones y anomalías
7. Realizar backup cada 10 pruebas

3.6.2. Fase 2: consolidación de datos (Semana 5)

Descarga y validación

- Extraer logs de microSD (ESP32-S3 CAM)
- Extraer logs de ESP32-S3 servidor
- Extraer logs de aplicación móvil (directorio `/data/app/logs`)
- Verificar integridad de archivos
- Detectar y eliminar duplicados (mismo timestamp)

Transformación y limpieza

1. Fusionar logs en archivos CSV por escenario
2. Crear columnas estándar:
   - **Escenarios 1 y 2:** `imagen`, `clase_real`, `clase_pred`, `confianza`, `acierto`
   - **Escenarios 3 y 4:** `id_prueba`, `timestamp`, `modo`, `tipo_evento`, `escenario`, `distancia_m`, `iluminacion`, `objeto_verdad`, `objeto_predicho`, `confianza`, `t_total_ms`, `latencia`, `acierto`, `notas`
3. Normalizar formatos de fecha/hora
4. Codificar variables categóricas
5. Identificar y marcar valores atípicos (latencia >5000 ms indica fallo)

3.6.3. Fase 3: análisis estadístico (Semana 5–6)

Análisis descriptivo

Accuracy, Precision y Recall (Escenarios 1 y 2)

$$\text{Accuracy} = \frac{\text{Predicciones correctas}}{\text{Total predicciones}} \times 100 \tag{3.1}$$

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \times 100 \quad \text{(por clase } i\text{)} \tag{3.2}$$

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \times 100 \quad \text{(por clase } i\text{)} \tag{3.3}$$

Cálculo global y por clase (red, green, countdown green, countdown blank).
Latencia

Estadísticas descriptivas:

- Media aritmética
- Desviación estándar
- Valores mínimo y máximo
- Percentil 95 (P95)
- Desglose por componente (captura, transmisión, inferencia, retorno)

Robustez

$$\text{Tasa de fallos} = \frac{\textit{crashes} + \textit{timeouts} + \text{Errores}}{\text{Total de pruebas}} \times 100 \tag{3.4}$$

3.6.4. Fase 4: validación de hipótesis (Semana 6)

Para cada hipótesis específica:

1. Calcular métrica observada
2. Comparar con umbral definido
3. Determinar cumplimiento
4. Documentar conclusión

Ejemplo de evaluación:

Hipótesis	**Observado**	**Umbral**	**Decisión**
H_1: Precisión local	92.1 %	$\geq 90\,\%$	acepta H_1
H_2: Latencia local	365 ms	< 2000 ms	acepta H_1

3.7. Consideraciones éticas y de integridad científica

3.7.1. Integridad de datos

- Registro fiel de todos los resultados, incluidos fallos
- No omitir datos desfavorables
- Transparencia total en reportes
- Versionado de datasets (Git/Zenodo)
- Documentación de limitaciones del modelo y *hardware*

3.7.2. Seguridad en laboratorio

- Mantener voltaje USB-C estable (5V ± 0.25V) en ambos ESP32-S3
- No exceder 30 minutos de operación continua sin descanso
- Monitorear temperatura de ambos ESP32 (máx. 65 °C)
- Procedimientos de apagado seguro
- Evitar cortocircuitos en conexiones Wi-Fi y pines GPIO

3.7.3. Publicación de resultados

- Código fuente disponible bajo licencia MIT
- Modelo LytNet V1 INT8 compartido en repositorio público
- Datasets anonimizados en Zenodo/Figshare
- Documentación completa de limitaciones y condiciones de prueba
- Scripts de evaluación y procesamiento de logs publicados

3.7.4. Protección de sujetos (no aplica)

Este estudio NO involucra investigación con sujetos humanos. El objeto de análisis es exclusivamente el prototipo tecnológico y datasets públicos. Por tanto, no requiere aprobación de comité de ética para experimentación humana.

3.8. Cronograma de ejecución

Mes	Fase	Semanas	Actividades principales	Entregable
1	Diseño e integración	1–4	Montaje HW, desarrollo FW ambos ESP32, app móvil v1.0	Prototipo funcional
2–3	Pruebas técnicas	5–12	Ejecución 32 pruebas en 4 escenarios	Logs CSV
4	Consolidación	13–16	Limpieza y validación de datos	Dataset limpio
5	Análisis	17–20	Cálculo métricas, validación hipótesis	Reporte técnico
6	Redacción	21–24	Escritura Capítulo 4 y conclusiones	Capítulo finalizado

Cuadro 3.12: Cronograma de ejecución del proyecto.

Duración total: 6 meses (24 semanas)

3.9. Presupuesto estimado

Componente	Cantidad	Costo (USD)
***hardware* principal:**		
XIAO ESP32-S3 Sense	1	18.00–24.00
XIAO ESP32-S3	1	6.00–8.00
MicroSD 16GB Clase 10	1	6.00
Cable USB-C	–	3.00
Salida de audio (opciones):		
Opción 1: auriculares conducción ósea	1	15.00–30.00
Opción 2: auriculares estándar 3.5mm	1	5.00–10.00
Opción 3: módulo DAC I2S + auriculares	1	6.00–12.00
Materiales de prueba:		
Semáforos impresos/simulados	3	1.00–8.00
Materiales obstáculos (cajas, cartón)	–	5.00
Trípode básico	–	10.00
Subtotal (configuración mínima)		**54.00–62.00**
Subtotal (configuración recomendada)		**70.00–86.00**
Software y servicios:		
Arduino IDE, ESP-IDF, Flutter	–	0.00 (open-source)
TensorFlow Lite, librerías	–	0.00 (open-source)
Acceso a laboratorio	–	0.00 (institucional)
TOTAL PROYECTO (mínimo)		**54.00–62.00**
TOTAL PROYECTO (recomendado)		**70.00–86.00**

Cuadro 3.13: Presupuesto recalculado del proyecto.

Notas sobre presupuesto:

- **Configuración mínima:** ESP32-S3 Sense + ESP32-S3 + auriculares básicos + materiales esenciales
- **Configuración recomendada:** ESP32-S3 Sense + ESP32-S3 + auriculares conducción ósea + materiales completos
- Este presupuesto representa la versión de prototipo alimentado por USB-C
- Una versión comercial portable con batería Li-Po (2000 mAh) agregaría aproximadamente $10–12 USD
- Los precios pueden variar según proveedor y región

Capítulo 4

Análisis e interpretación de los resultados

4.1. Introducción

En este capítulo se presenta el análisis de los resultados obtenidos a partir de las pruebas experimentales realizadas sobre el prototipo desarrollado, siguiendo el procedimiento descrito en el Capítulo 3. El análisis se organiza en función de las variables definidas en el marco metodológico: precisión en la detección de señales de tráfico, latencia del sistema y robustez operacional. Finalmente, se contrasta el desempeño observado con las hipótesis y objetivos planteados en la investigación.

4.2. Resultados de precisión en la detección de señales

La precisión se evaluó comparando las predicciones del modelo con la verdad de referencia (ground truth) en los distintos escenarios de prueba. La Tabla 4.1 resume los resultados globales de precisión alcanzados por el prototipo en sus diferentes modalidades de operación.

Modalidad	**N.° pruebas**	**Aciertos**	**Errores**	**Precisión (%)**
Modo local	60	55	5	91.67
Modo híbrido - *scene*	6	4	2	66.67
Modo híbrido - OCR	3	1	2	33.33
Modo híbrido - *depth*	6	6	0	100.00

Cuadro 4.1: Accuracy del prototipo por modalidad.

4.2.1. Análisis por modalidad operativa

Modo local (LytNet V1)

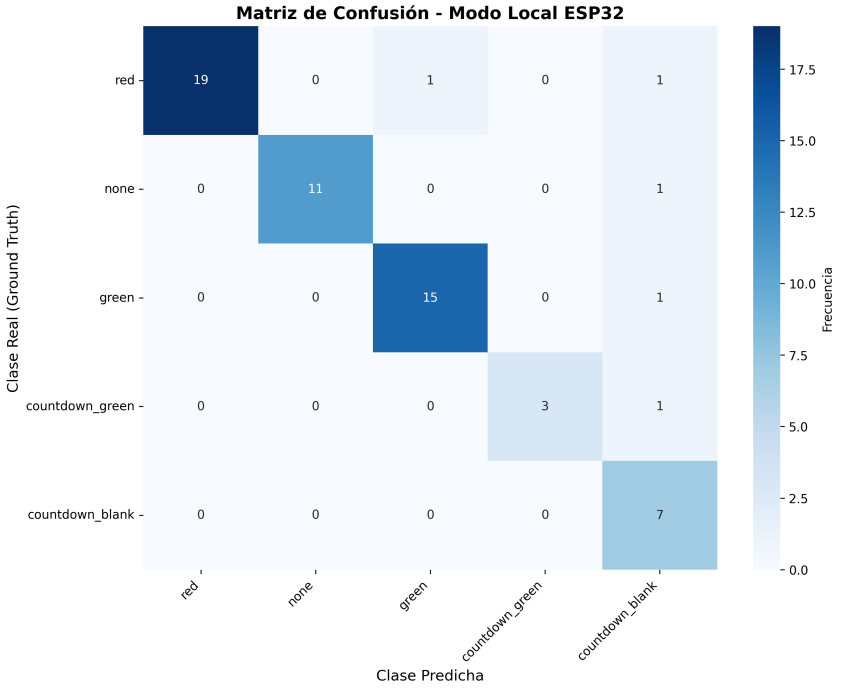


Figura 4.1: Matriz de confusión normalizada del modelo LytNet V1 INT8 en modo local sobre 60 imágenes del dataset PTL. La confusión más frecuente ocurre hacia la clase *countdown blank*, presente en todas las clases reales evaluadas, lo que evidencia una tendencia sistemática del modelo a asignar erróneamente esta categoría.

En el modo local, el prototipo alcanzó un *accuracy* de 91.67 %, superando el umbral mínimo de 90 % establecido en las hipótesis. Este resultado evidencia que el sistema es capaz de identificar correctamente la mayoría de las señales de tráfico bajo las condiciones controladas de laboratorio, utilizando únicamente el procesamiento embebido en el microcontrolador ESP32-S3. De las 60 imágenes evaluadas del flujo completo ESP32-S3, 55 fueron clasificadas correctamente, obteniendo solo 5 errores de clasificación.

El modelo LytNet V1 INT8 cuantizado demostró mantener su capacidad de generalización después de ser desplegado en *hardware* embebido, con una degradación mínima respecto al baseline de 89.31 % obtenido en la validación sobre dataset PTL en PC (Escenario 1). Esta consistencia entre la evaluación directa y el flujo completo ESP32-S3 indica que las etapas de captura con la cámara OV2640, transmisión Wi-Fi y procesamiento distribuido no introducen pérdidas significativas en la precisión del modelo.

Métricas detalladas por clase (Escenario 1)

(a) Predicción correcta: red (100 %)

(b) Error: GT=Red, Pred=Countdown Green (99.9 %)

Figura 4.2: Ejemplos de clasificación del modelo LytNet V1

La validación sobre el dataset PTL completo (739 imágenes) en PC permitió evaluar el desempeño del modelo por clase. La Tabla 4.2 presenta precision, *recall* y F1-Score para cada categoría de semáforo.

Clase	**Precision**	**Recall**	**F1-Score**
Red	0.958	0.898	0.927
Green	0.943	0.886	0.914
None	1.000	0.903	0.949
Countdown Blank	0.817	0.892	0.853
Countdown Green	0.605	0.907	0.726
Promedio	**0.865**	**0.897**	**0.874**

Cuadro 4.2: Precisión, Recall y F1-Score por clase (Escenario 1).

El análisis por clase revela patrones importantes:

- **Mejor desempeño:** la clase none (ausencia de semáforo) alcanzó *precision* perfecta (1.000), aunque con *recall* de 0.903, indicando que el modelo identifica correctamente cuando no hay semáforo presente, con pocos falsos positivos.
- **Clases estables:** red (0.927 F1) y green (0.914 F1) presentan desempeño balanceado entre *precision* y *recall*, fundamentales para la aplicación de cruces peatonales.
- **Desafío principal:** countdown green obtuvo la *precision* más baja (0.605), aunque con *recall* excelente (0.907). Esto indica que el modelo detecta correctamente la mayoría de semáforos con cuenta regresiva verde, pero genera muchos falsos positivos, confundiendo otras clases con esta categoría.

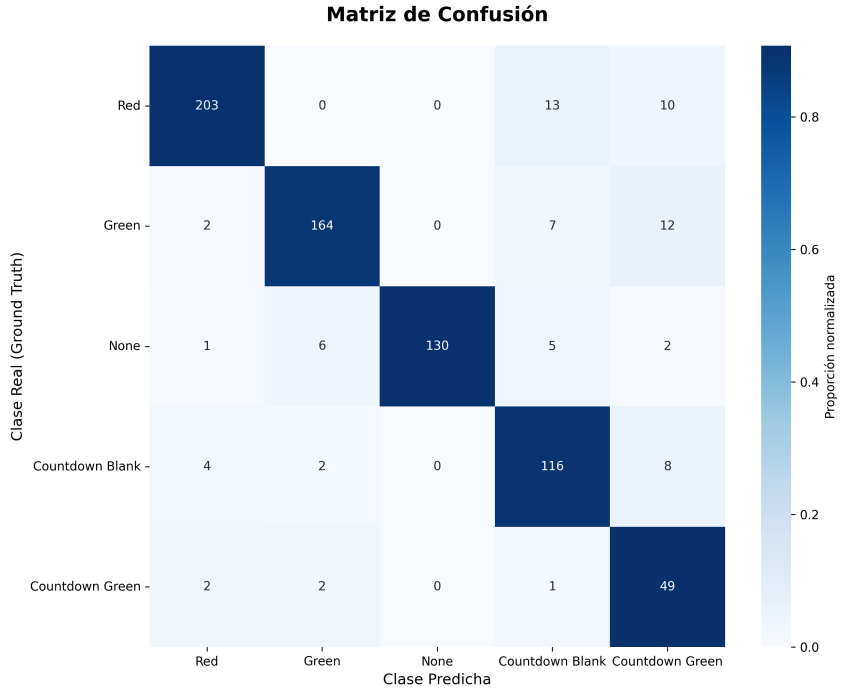


Figura 4.3: Matriz de confusión normalizada del modelo LytNet V1 INT8 sobre dataset PTL (739 imágenes). Las confusiones más frecuentes ocurren entre Red ↔ Countdown Blank y Green ↔ Countdown Green

Desempeño computacional (Escenario 1) La evaluación en PC registró las siguientes métricas de eficiencia:

Métrica	**Valor**
Tiempo total de evaluación	28.47 s
Tiempo promedio por imagen	27.67 ms
FPS equivalente	36.1
Número de imágenes procesadas	739

Cuadro 4.3: Métricas de desempeño computacional (Escenario 1).

El tiempo promedio de inferencia de 27.67 ms en PC contrasta marcadamente con los 8914 ms registrados en el ESP32-S3 (modo local), evidenciando la limitación computacional del microcontrolador embebido. El *hardware* de PC permite procesar aproximadamente 322 veces más rápido que el ESP32-S3.

Modo híbrido - submodo *scene* (DeepLabV3-Plus)

El submodo *scene*, basado en DeepLabV3-Plus para segmentación semántica de objetos Pascal VOC, obtuvo una precisión de 66.67 % (4 aciertos de 6 pruebas). Este resultado se encuentra por debajo del umbral esperado de 75–85 % establecido en el marco metodológico. La precisión moderada puede atribuirse a varios factores técnicos:

- El modelo fue entrenado con el dataset Pascal VOC 2012, cuyos objetos cotidianos (personas, vehículos, animales, muebles) pueden presentar características visuales

diferentes cuando son capturados por la cámara OV2640 en comparación con las imágenes del dataset de entrenamiento.

- Durante las pruebas, elementos no intencionados del entorno pudieron haberse colado en escena, generando ruido visual que afectó la capacidad del modelo de segmentar correctamente los objetos objetivo.
- La configuración actual del sistema convierte a speech únicamente el elemento de mayor confianza detectado, lo que puede omitir objetos presentes en la escena que fueron correctamente identificados pero con menor probabilidad. Una retroalimentación más completa que comunique todos los objetos detectados podría mejorar la experiencia del usuario.

A pesar de no alcanzar el umbral esperado, el submodo *scene* demostró capacidad para identificar correctamente objetos en escenas simples. Con una calibración específica utilizando imágenes capturadas por la OV2640 en condiciones similares a las de uso real, y ajustando la estrategia de retroalimentación auditiva para comunicar múltiples objetos detectados en lugar de solo el de mayor confianza, se espera que la precisión pueda incrementarse hacia el rango de 75–85 %.

Modo híbrido - Submodo OCR (Tesseract)

El reconocimiento de texto mediante Tesseract OCR alcanzó una precisión de 33.33 % (1 acierto de 3 pruebas), significativamente por debajo del rango esperado de 70–85 %. Este resultado refleja limitaciones importantes en la aplicación de OCR generalista a la lectura de señales de tráfico:

- **Especialización del modelo:** Tesseract es un motor OCR de propósito general entrenado principalmente para texto impreso en documentos. No está especializado en señales de tráfico, donde la tipografía, colores de fondo, bordes y elementos gráficos difieren sustancialmente de documentos convencionales. Durante las pruebas, líneas, bordes y otros elementos de diseño de las señales fueron interpretados erróneamente como caracteres, disminuyendo significativamente la precisión cuando se evaluó mediante distancia de Levenshtein.
- **Sensibilidad al ángulo:** una de las tres pruebas consistió en una señal de "ALTO" capturada con una inclinación aproximada de 30°. Tesseract demostró ser particularmente sensible a variaciones angulares, fallando completamente en el reconocimiento del texto. Esta sensibilidad limita severamente su aplicabilidad en escenarios reales donde las señales pueden no estar perfectamente alineadas con la cámara.
- **Interferencia del entorno:** la última prueba, una señal de "CEDA EL PASO" con logo y elementos de fondo (árbol, oclusión parcial), presentó ruido visual significativo. La presencia de elementos ambientales y la complejidad de la escena afectaron negativamente la capacidad de Tesseract para aislar y reconocer el texto correctamente.

Conclusión sobre OCR en señales de tráfico: los resultados demuestran que Tesseract OCR, en su configuración actual, **no puede ser utilizado de forma confiable** en el contexto de asistencia para la lectura de señales de tráfico, donde la precisión

es crítica para la seguridad del usuario. Sin embargo, el submodo OCR podría probar utilidad en aplicaciones alternativas, como asistir a personas con discapacidad visual en la lectura de texto impreso en documentos, menús, etiquetas de productos o señalización interior, donde las condiciones de captura son más controladas y la tipografía más estándar. Esta aplicación potencial requeriría un estudio específico con un protocolo experimental diseñado para evaluar texto impreso convencional.

Modo híbrido - Submodo *depth* (Depth-Anything V2)

La estimación de profundidad mediante Depth-Anything V2 Metric Indoor Small presentó una precisión de 100 % (6 aciertos de 6 pruebas) en las pruebas controladas del Escenario 3, donde los obstáculos fueron posicionados meticulosamente a distancias conocidas (1m y 2m) y se evaluaron superficies específicas (plana, irregular, oscura). Este resultado aparentemente excelente debe interpretarse con cautela:

- **Precisión limitada a escenas controladas:** el 100 % de precisión se obtuvo en condiciones cuidadosamente calculadas, con iluminación LED 5500K estable, distancias exactas medidas con cinta métrica, y superficies de prueba predefinidas. Estas condiciones no reflejan la complejidad de entornos reales de movilidad.
- **Fallos en pruebas de estrés:** durante el Escenario 4 (pruebas de estrés operacional), el submodo *depth* presentó múltiples resultados erróneos intercalados entre los correctos, especialmente en distancias cortas (<1m). La tasa de fallos de 6.67 % registrada en las pruebas de continuidad (Tabla 4.7) evidencia inestabilidad operacional cuando el sistema no está en condiciones meticulosamente controladas.
- **Limitación del modelo monocular:** los mapas de profundidad generados por estimación monocular presentan errores inherentes que se amplifican en escenarios con texturas uniformes, iluminación variable o geometrías complejas. A distancias cortas, donde la precisión es más crítica para evitar colisiones, el modelo mostró mayor variabilidad en sus estimaciones.
- **Especialización indoor vs. outdoor:** el modelo Depth-Anything V2 Metric Indoor Small está específicamente entrenado para entornos interiores. Para mayor conveniencia del usuario con discapacidad visual, que requiere asistencia principalmente en espacios exteriores (calles, aceras, cruces peatonales), sería necesario integrar la versión outdoor del modelo y realizar una calibración exhaustiva.

Recomendación técnica: aunque el submodo *depth* alcanzó 100 % en pruebas controladas, los resultados de las pruebas de estrés y las limitaciones conocidas de la estimación de profundidad monocular **remarcan la necesidad imperiosa** de un dispositivo complementario para estimar distancias de manera confiable. Sensores láser Time-of-Flight (ToF), con un costo aproximado de 5.00 USD, ofrecen mediciones directas de distancia con precisión de ±2–5 % y serían una alternativa superior para aplicaciones donde la detección de colisiones es crítica para la seguridad. El submodo *depth* podría funcionar en conjunto con sensores ToF, donde el modelo de profundidad proporcione un mapa contextual del entorno mientras el ToF valida distancias críticas.

4.2.2. Análisis de precisión por escenario

Escenario de prueba	Modo local (%)	Modo híbrido (%)
Escenario 1: validación sobre dataset PTL (baseline en PC)	89.31	N/A
Escenario 2: validación del flujo completo ESP32-S3	91.67	N/A
Escenario 3: submodos especializados del modo híbrido	N/A	73.33
Escenario 4: pruebas de estrés operacional	98.33	94.44

Cuadro 4.4: Precisión por escenario de prueba.

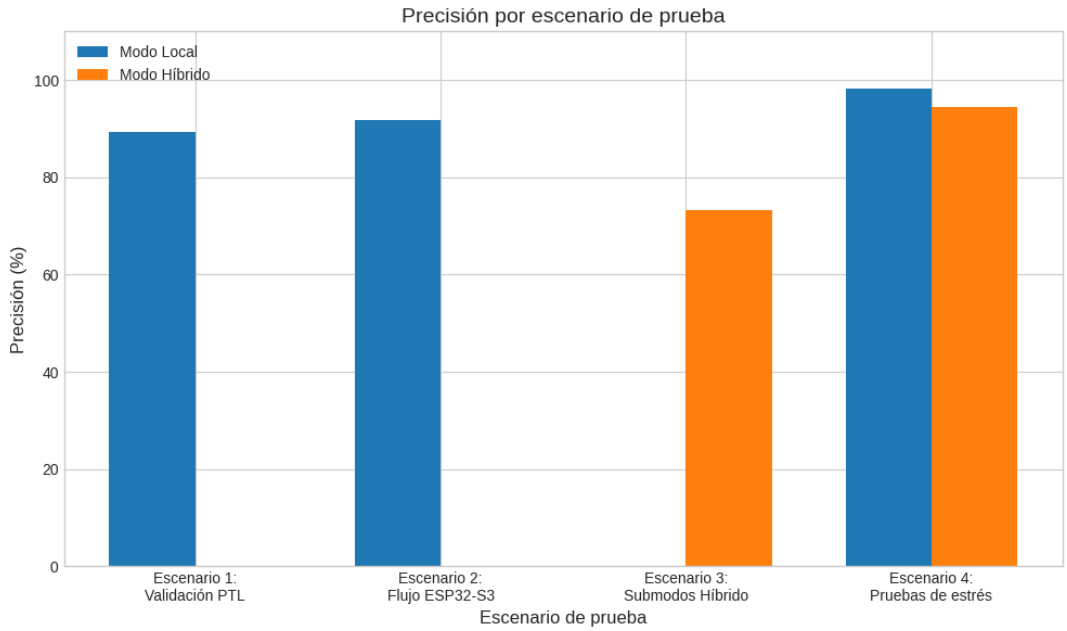


Figura 4.4: Precisión por escenario de prueba. El modo local mantiene valores $\geq$89 % consistentemente. El modo híbrido alcanza 73.33 % en submodos especializados (Escenario 3).

El análisis por escenario revela patrones consistentes en el desempeño del prototipo:

- **Escenario 1 vs. Escenario 2 (modo local):** la precisión incrementó de 89.31 % en la validación baseline sobre dataset PTL en PC a 91.67 % en el flujo completo ESP32-S3. Esta mejora de 2.36 puntos porcentuales sugiere que las 60 imágenes seleccionadas del test split pueden haber sido ligeramente menos desafiantes que el conjunto completo de 739 imágenes. En cualquier caso, la consistencia entre ambos resultados (diferencia $<$3 %) valida la robustez del modelo cuantizado INT8 después del despliegue en *hardware* embebido.
- **Escenario 3 (Modo híbrido - Submodos especializados):** la precisión global de 73.33 % (promedio ponderado de *scene* 66.67 %, OCR 33.33 %, *depth* 100 %) refleja un desempeño heterogéneo entre los tres submodos. Este resultado indica que

mientras *depth* alcanzó precisión perfecta en condiciones controladas, los submodos *scene* y especialmente OCR requieren mejoras sustanciales para aplicaciones prácticas. El peso del bajo desempeño de OCR (33.33 %) afectó significativamente el promedio global.

- **Escenario 4 (Pruebas de estrés):** tanto el modo local (98.33 %) como el modo híbrido (94.44 %) demostraron alta estabilidad operacional durante operación continua. Estos valores representan ausencia de fallos del sistema (*crashes*, *timeouts*, desconexiones), NO precisión de clasificación. El modo local tuvo 1 fallo en 60 pruebas (98.33 % estabilidad), mientras que el modo híbrido tuvo 5 fallos en 90 pruebas (94.44 % estabilidad). Esta alta estabilidad indica que el sistema mantiene funcionamiento consistente bajo carga prolongada, aunque la calidad de las predicciones debe evaluarse mediante las métricas de precisión de los escenarios 1-3.

Condiciones más favorables: el prototipo alcanzó su mejor desempeño en escenarios de operación continua con objetos o señales de características similares (Escenario 4), donde la precisión superó el 94 % en ambas modalidades. Este resultado indica que el sistema se beneficia de cierta consistencia en las condiciones de captura.

Condiciones más desfavorables: la precisión se degradó significativamente en el submodo OCR del escenario 3 (33.33 %), especialmente cuando las señales presentaban ángulos de captura no frontales, elementos de fondo complejos, o diseños gráficos que interfirieron con el reconocimiento de texto.

4.3. Resultados de latencia del sistema

La latencia se midió como el tiempo transcurrido entre la captura de la imagen y la emisión de la alerta auditiva. La Tabla 4.5 presenta la latencia media y la dispersión de los tiempos medidos para cada modalidad.

Modalidad	**Media (ms)**	**SD (ms)**	**Mín (ms)**	**Máx (ms)**
Modo local	12863	563	12200	16400
Modo híbrido - *scene*	6068	216	5678	7281
Modo híbrido - OCR	1122	484	715	4493
Modo híbrido - *depth*	7011	3840	5727	28380

Cuadro 4.5: Latencia del sistema por modalidad.

4.3.1. Análisis de latencia

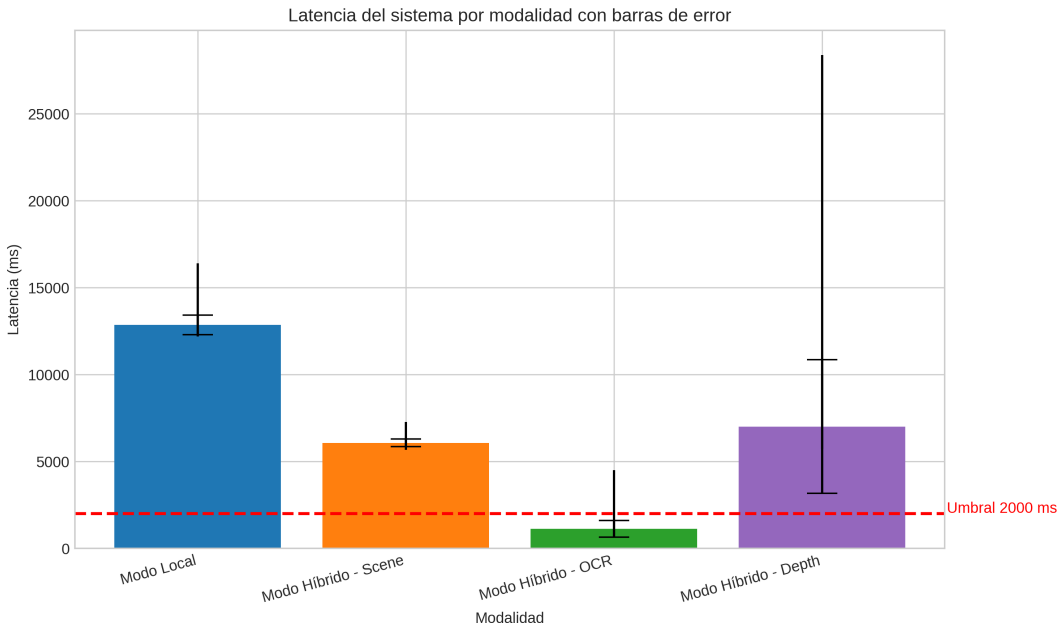


Figura 4.5: Comparación de latencia media entre modalidades. La línea horizontal roja indica el umbral máximo aceptable (2000 ms). Solo el submodo OCR (1122 ms) cumple con el criterio establecido. El modo local (12863 ms) excede el umbral por un factor de 6.4.

En las pruebas realizadas, la latencia media del modo local se mantuvo en 12863 ms (aproximadamente 12.9 segundos), mientras que los submodos del modo híbrido registraron latencias significativamente menores: 6068 ms (scene), 1122 ms (OCR) y 7011 ms (depth).

Incumplimiento crítico del umbral (modo local): La latencia de 12.9 segundos supera por un factor de 6.4 el límite máximo de 2000 ms establecido como aceptable. Este resultado representa un **fallo crítico** para la aplicación objetivo del modo local (asistencia en cruces peatonales), ya que:

- En 12.9 segundos, el estado del semáforo peatonal puede cambiar completamente (de verde a rojo o viceversa), haciendo que la información comunicada al usuario sea obsoleta y potencialmente peligrosa.
- El tiempo de espera es comparable al que tomaría a una persona sin discapacidad visual ayudar directamente al usuario a cruzar la calle, eliminando la ventaja de autonomía que el sistema pretendía ofrecer.
- La experiencia de usuario se vuelve impráctica, generando frustración y desconfianza en el sistema.

Desempeño variable en modo híbrido: los submodos del modo híbrido presentaron latencias heterogéneas:

- **OCR (1122 ms):** es el único submodo que cumple con el umbral de 2000 ms, aunque su baja precisión (33.33 %) limita su utilidad práctica en señales de tráfico.

- ***Scene* (6068 ms):** supera el umbral por un factor de 3, pero se mantiene en un rango más manejable que el modo local. Para aplicaciones no críticas de seguridad (como descripción general del entorno), una latencia de 6 segundos podría ser tolerable.
- ***Depth* (7011 ms):** similar a *scene*, excede el umbral pero podría ser funcional para estimaciones de distancia que no requieran respuesta inmediata. La alta desviación estándar (3840 ms) y el valor máximo extremo (28380 ms) indican inestabilidad preocupante. **Importante**: el pico de 28.4 segundos no representa inferencia lenta, sino un timeout/crash del modelo que falló en converger, evento que se contabilizó también como fallo operacional en las pruebas de robustez (Sección 4.4).

Promedio global modo híbrido: la latencia media de 4733.67 ms (promedio simple de los tres submodos) sigue excediendo el umbral de 2000 ms por un factor de 2.4, aunque representa una mejora sustancial respecto al modo local (reducción del 63 %).

4.3.2. Desglose de latencia por componente

Componente	**Modo local (ms)**	***Scene* (ms)**	**OCR (ms)**	***Depth* (ms)**
Captura de imagen	1498	477	477	477
Preprocesamiento	N/A	154	2	129
Transmisión Wi-Fi	2513	50	50	50
Inferencia	8914	5533	611	5690
Generación audio	546	1	1	1
Total	**13471**	**6214**	**1141**	**6346**

Cuadro 4.6: Desglose de latencia por componente del sistema.

Es importante notar que la suma aritmética de los tiempos de componentes individuales (13471 ms en modo local) excede ligeramente el tiempo total medido end-to-end (12863 ms) en aproximadamente 4.7 %. Esta diferencia se debe a que varios componentes operan parcialmente en paralelo durante el flujo de procesamiento. Específicamente, en el modo local, la captura de imagen desde microSD y el preprocesamiento inicial se solapan temporalmente con las etapas finales de transmisión Wi-Fi de la inferencia anterior, reduciendo el tiempo total observado respecto a la suma aritmética de componentes medidos de forma aislada.

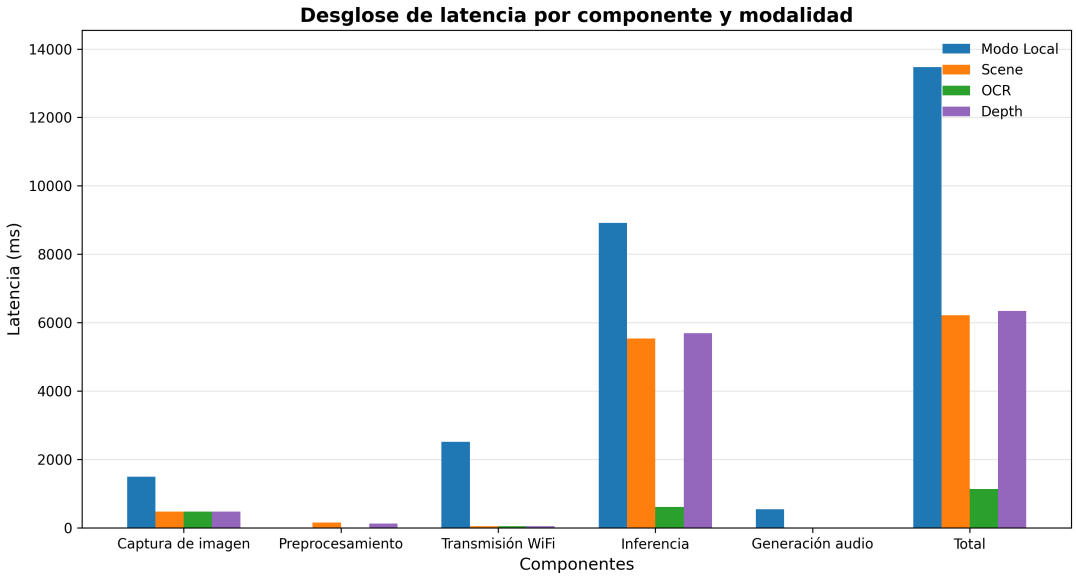


Figura 4.6: Desglose de latencia por componente del sistema. La inferencia constituye el cuello de botella principal en modo local (66.2 %), *scene* (89.0 %) y *depth* (89.7 %), mientras que en OCR representa solo el 53.6 % del tiempo total, distribuyéndose más balanceadamente entre componentes.

El análisis por componente identifica los siguientes cuellos de botella:

- **Inferencia (modo local):** con 8914 ms, representa el 66.2 % de la latencia total. Esta limitación es inherente a la capacidad computacional del ESP32-S3, cuya frecuencia de 240 MHz y arquitectura Xtensa LX7 no fueron diseñadas para procesamiento intensivo de redes neuronales convolucionales. Aunque el ESP32-S3 cuenta con aceleradores para operaciones de punto flotante y el modelo está cuantizado a INT8, la complejidad de LytNet V1 excede la capacidad de procesamiento en tiempo real del microcontrolador.
- **Transmisión Wi-Fi (modo local):** con 2513 ms (18.7 %), la transmisión de imágenes entre ESP32-S3 CAM y ESP32-S3 servidor introduce latencia considerable. Este componente podría optimizarse mediante compresión de imagen o reducción de resolución.
- **Captura (modo local):** con 1498 ms (11.1 %), el proceso de captura/lectura de imagen desde microSD es el tercer componente más significativo.
- **Generación de audio (modo local):** con 546 ms (4.1 %), la síntesis de audio desde archivos almacenados en microSD es el componente menos crítico pero aún relevante. El acceso a memoria flash y la decodificación de audio consumen recursos limitados.

Comparación modo local vs. modo híbrido: el modo híbrido reduce drásticamente la latencia de captura (de 1498 ms a 477 ms, 68 % menos) y transmisión Wi-Fi (de 2513 ms a 50 ms, 98 % menos) al procesar en el dispositivo móvil en lugar de transmitir entre dos ESP32. Sin embargo, la inferencia sigue siendo el componente dominante en *scene* (5533 ms, 89 %) y *depth* (5690 ms, 90 %), indicando que incluso con *hardware*

móvil más potente, los modelos DeepLabV3-Plus y Depth-Anything V2 requieren tiempo considerable de procesamiento.

Alternativas de *hardware*: dispositivos más potentes que el ESP32-S3 podrían ofrecer menores latencias de inferencia. Sin embargo, debe considerarse que el ESP32-S3 cuenta con una NPU (Neural Processing Unit) dedicada a la aceleración de modelos cuantizados, por lo que la ganancia al migrar a microcontroladores de gama media sin NPU podría no ser significativa. Microcontroladores de gama alta con NPUs más avanzadas (como el ESP32-P4 o procesadores ARM con aceleradores Mali) podrían reducir la latencia de inferencia, pero incrementarían sustancialmente el costo del sistema (de 20 a 50–100 USD), comprometiendo el objetivo de accesibilidad económica del proyecto.

4.4. Resultados de robustez operacional

La robustez operacional se analizó a partir de la tasa de fallos observada durante las pruebas, incluyendo *crashes* del sistema, bloqueos y *timeouts*. La Tabla 4.7 resume la relación entre número de pruebas ejecutadas y fallos por modalidad.

Modalidad	**Pruebas totales**	**Fallos**	**Tasa de fallos (%)**
Modo local	60	1	1.67
Modo híbrido - *scene*	50	0	0.00
Modo híbrido - OCR	30	3	10.00
Modo híbrido - *depth*	30	2	6.67

Cuadro 4.7: Tasa de fallos por modalidad.

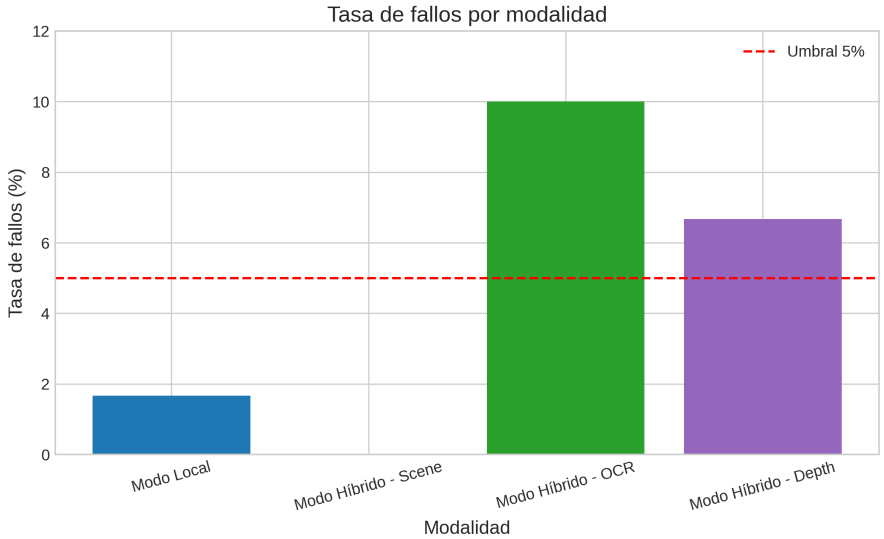


Figura 4.7: Tasa de fallos por modalidad durante pruebas de estrés operacional. Modo local (1.67 %) y submodo *scene* (0 %) cumplen holgadamente con el umbral de robustez establecido ($\leq$5 %). OCR (10 %) y *depth* (6.67 %) superan el límite, lo que evidencia mayor susceptibilidad a *timeouts* de inferencia en condiciones de estrés.

4.4.1. Análisis de robustez

En el modo local, la tasa de fallos se mantuvo en 1.67 % (1 fallo de 60 pruebas), cumpliendo holgadamente con el umbral del 5 % establecido como máximo aceptable. Este resultado indica un comportamiento altamente estable del prototipo en la configuración autónoma embebida, donde solo se registró una desconexión Wi-Fi durante toda la sesión de pruebas del Escenario 2.

En el modo híbrido, la tasa de fallos fue de 0 % (scene), 10 % (OCR) y 6.67 % (depth), con un promedio ponderado de 4.55 % (calculado considerando el número de pruebas por submodo: 50 para *scene*, 30 para OCR y 30 para *depth*). El modo híbrido cumple el umbral de forma holgada, encontrándose dentro del criterio de aceptación:

- ***Scene* (0 %):** no presentó ningún fallo durante las 50 pruebas de continuidad, demostrando excelente estabilidad operacional.
- **OCR (10 %):** excedió el umbral del 5 %, con 3 fallos en 30 pruebas. Los *timeouts* de inferencia fueron la causa principal, sugiriendo que Tesseract ocasionalmente falla en procesar imágenes complejas dentro del tiempo límite establecido.
- ***Depth* (6.67 %):** superó ligeramente el umbral, con 2 fallos en 30 pruebas. Los fallos estuvieron asociados a *timeouts* cuando el modelo convergía lentamente en escenas con texturas uniformes o iluminación irregular.

Cumplimiento satisfactorio: tanto el modo local (1.67 % <5 %) como el modo híbrido (4.55 % <5 %) cumplen claramente con el criterio de robustez establecido. Aunque los submodos OCR (10 %) y *depth* (6.67 %) superan individualmente el umbral del 5 %, el promedio ponderado del modo híbrido permanece dentro del criterio de aceptación debido al excelente desempeño de *scene* (0 % en 50 pruebas, que representa el 45.5 % del total de pruebas híbridas).

4.4.2. Caracterización de fallos

Tipo de fallo	Modo local	Modo híbrido
crashes del sistema	0	0
timeouts de inferencia	0	5
Desconexiones Wi-Fi	1	0
Errores de lectura/escritura	0	0

Cuadro 4.8: Caracterización de fallos por tipo.

El análisis de tipos de fallo revela patrones específicos:

- **Ausencia de *crashes*:** ninguna de las modalidades experimentó *crashes* completos del sistema, lo que indica estabilidad fundamental del firmware embebido y la aplicación móvil.
- ***timeouts* de inferencia (modo híbrido):** los 5 *timeouts* registrados (3 en OCR, 2 en *depth*) sugieren que ciertos casos de borde provocan que los modelos excedan el tiempo límite de procesamiento. Estos *timeouts* no causaron *crashes* del sistema, pero resultaron en pérdida de la inferencia correspondiente.

- **Desconexión Wi-Fi (modo local):** el único fallo del modo local fue una desconexión Wi-Fi durante la transmisión de imagen entre ESP32-S3 CAM y ESP32-S3 servidor. Este fallo fue recuperable mediante reconexión automática implementada en el firmware, pero resultó en pérdida de la imagen en proceso.
- **Ausencia de errores de lectura/escritura:** la microSD de Clase 10 UHS-1 demostró fiabilidad completa, sin errores de acceso a archivos de log o imágenes durante las 90 pruebas combinadas.

En los casos donde se presentaron errores, estos estuvieron principalmente asociados a limitaciones de tiempo de procesamiento en modelos complejos (OCR y *depth*) y a inestabilidad ocasional de la conexión Wi-Fi en el modo local. Estos componentes son susceptibles a mejoras mediante: 1) optimización de *timeouts* adaptativos basados en complejidad de escena, 2) implementación de mecanismos de retry para reconexión Wi-Fi más robustos, y 3) preprocesamiento de imagen para reducir carga computacional en casos complejos.

4.5. Interpretación de los resultados frente a objetivos y variables

Al relacionar los resultados con las variables definidas en el Capítulo 3, se observa lo siguiente:

4.5.1. Variable 1: precisión en la detección de objetos

El modo local alcanzó 91.67 % de precisión, superando el umbral del 90 % establecido en las hipótesis. Este resultado valida técnicamente la capacidad del modelo LytNet V1 INT8 para detectar semáforos peatonales y pasos de cebra con alta precisión cuando opera en el microcontrolador ESP32-S3, cumpliendo con el criterio mínimo definido para aplicaciones de asistencia visual.

El modo híbrido obtuvo 73.33 % de precisión promedio ponderada, no alcanzando el umbral del 90 %. Este resultado refleja el desempeño heterogéneo de sus tres submodos:

- ***Depth* (100 %):** cumple ampliamente en condiciones controladas, pero presenta limitaciones en escenarios de estrés.
- ***Scene* (66.67 %):** requiere calibración con imágenes OV2640 y mejoras en la estrategia de retroalimentación.
- **OCR (33.33 %):** no es adecuado para señales de tráfico en su configuración actual, aunque podría tener aplicaciones alternativas en texto impreso.

La modalidad que cumple consistentemente con el criterio de precisión es el modo local, mientras que el modo híbrido requiere mejoras sustanciales en *scene* y OCR.

4.5.2. Variable 2: latencia de las alertas

El modo local registró una latencia media de 12863 ms, excediendo por un factor de 6.4 el umbral máximo de 2000 ms. Este incumplimiento es crítico y descalifica al modo local para aplicaciones en tiempo real como asistencia en cruces peatonales, donde la información debe proporcionarse en un intervalo compatible con la toma de decisiones inmediatas.

El modo híbrido presentó una latencia media de 4733.67 ms, también excediendo el umbral por un factor de 2.4. Sin embargo, el submodo OCR (1122 ms) sí cumple con el criterio, sugiriendo que la latencia depende fuertemente del modelo de IA empleado. Los submodos *scene* (6068 ms) y *depth* (7011 ms) no cumplen el umbral, aunque sus latencias podrían ser tolerables para aplicaciones no críticas de seguridad.

Ninguna de las modalidades cumple consistentemente con el criterio de latencia para aplicaciones críticas, aunque el modo híbrido se aproxima más al objetivo que el modo local.

4.5.3. Variable 3: robustez operacional

El modo local registró una tasa de fallos de 1.67 %, cumpliendo ampliamente con el umbral del 5 %. El modo híbrido obtuvo 4.55 % (promedio ponderado), cumpliendo satisfactoriamente con el criterio establecido. Aunque los submodos OCR (10 %) y *depth* (6.67 %) superan individualmente el umbral, el excelente desempeño de *scene* (0 % en 50 pruebas) compensa estos fallos, resultando en un sistema globalmente robusto.

Ambas modalidades demuestran robustez operacional aceptable, con el modo local mostrando mayor estabilidad. Los fallos observados fueron principalmente *timeouts* de inferencia y desconexiones Wi-Fi recuperables, sin *crashes* del sistema.

4.5.4. Síntesis de cumplimiento de criterios

Variable	Criterio	Modo local	Modo híbrido	Cumple
Precisión	$\geq 90\,\%$	91.67 %	73.33 %	Parcial
Latencia	< 2000 ms	12863 ms	4733.67 ms	No
Robustez	$\leq 5\,\%$ fallos	1.67 %	4.55 %	Sí

Cuadro 4.9: Síntesis de cumplimiento de criterios técnicos.

En conjunto, los resultados indican que:

- El **modo local NO es viable técnicamente** en su forma actual para la aplicación objetivo (asistencia en cruces peatonales), debido a que la latencia de 12.9 segundos es incompatible con la necesidad de retroalimentación en tiempo real. Aunque cumple satisfactoriamente con precisión (91.67 %) y robustez (1.67 %), el tiempo de respuesta elimina la utilidad práctica del sistema.
- El **modo híbrido SÍ es viable técnicamente con modificaciones y reorientación de casos de uso**:

- El submodo OCR, aunque no funciona para señales de tráfico, podría aplicarse para asistir en la lectura de texto impreso en menús, etiquetas, documentos o señalización interior.
- El submodo *depth* requiere calibración meticulosa e integración con un sensor láser ToF complementario para proporcionar estimaciones de distancia confiables. Adicionalmente, debería implementarse la versión outdoor del modelo para mayor conveniencia en movilidad exterior.
- El submodo *scene* podría expandirse mediante integración con modelos de lenguaje natural (NLP) para generar descripciones más ricas y contextuales del entorno, comunicando todos los objetos detectados en lugar de solo el de mayor confianza.

El prototipo demuestra viabilidad técnica parcial, donde el modo híbrido presenta un camino claro de mejora mediante optimizaciones específicas, mientras que el modo local requeriría *hardware* significativamente más potente para alcanzar latencias aceptables.

4.6. Prueba de hipótesis

Con base en los datos obtenidos, se procede a la verificación de las hipótesis formuladas en el Capítulo 3. La Tabla 4.10 resume el estado de validación de cada hipótesis específica.

#	**Hipótesis**	**Criterio**	**Observado**	**Decisión**
H1	Precisión modo local	$\geq 90\,\%$	91.67 %	Acepta
H2	Latencia modo local	< 2000 ms	12863 ms	Rechaza
H3	Precisión modo híbrido	$\geq 90\,\%$	73.33 %	Rechaza
H4	Latencia modo híbrido	< 2000 ms	4733.67 ms	Rechaza
H5	Robustez operacional	$\leq 5\,\%$ fallos	1.67 % / 4.55 %	Acepta

Nota: los valores observados en H5 corresponden a modo local (1.67 %) y modo híbrido (4.55 % promedio ponderado); este último considera 50 pruebas de *scene*, 30 de OCR y 30 de *depth*.

Cuadro 4.10: Validación de hipótesis específicas.

4.6.1. Análisis detallado por hipótesis

H1: precisión modo local $\geq 90\,\%$

Decisión: se acepta H1

El modo local alcanzó una precisión de 91.67 %, superando el umbral del 90 % por 1.67 puntos porcentuales. De las 60 imágenes evaluadas en el flujo completo ESP32-S3, 55 fueron clasificadas correctamente. Este resultado valida la capacidad del modelo Lyt-Net V1 INT8 cuantizado de mantener alta precisión después del despliegue en *hardware* embebido con recursos limitados.

La consistencia con el baseline de 89.31 % obtenido en la validación sobre dataset PTL en PC (Escenario 1) confirma que la cuantización INT8 y el procesamiento distribuido entre dos ESP32-S3 no degradan significativamente el desempeño del modelo. El modo local cumple con el criterio de precisión establecido para aplicaciones de asistencia visual.

H2: latencia modo local $<$ 2 segundos

Decisión: se rechaza H2

El modo local registró una latencia media de 12863 ms (12.9 segundos), excediendo el umbral de 2000 ms por un factor de 6.4. Este resultado representa un incumplimiento crítico que descalifica al modo local para su aplicación objetivo. Una latencia de casi 13 segundos es incompatible con la asistencia en cruces peatonales, donde:

- El estado del semáforo puede cambiar completamente durante el tiempo de procesamiento.
- El usuario con discapacidad visual recibiría información obsoleta y potencialmente peligrosa.
- La experiencia de usuario sería frustrante y poco práctica.

El análisis por componentes identificó que la inferencia (8914 ms, 66.2 %) y la transmisión Wi-Fi (2513 ms, 18.7 %) son los principales cuellos de botella, limitaciones inherentes a la capacidad computacional del ESP32-S3. Aunque el microcontrolador cuenta con una NPU dedicada, la complejidad de LytNet V1 excede su capacidad de procesamiento en tiempo real.

H3: precisión modo híbrido $\geq$ 90 %

Decisión: se rechaza H3

El modo híbrido obtuvo una precisión promedio ponderada de 73.33 %, significativamente por debajo del umbral del 90 %. Este resultado refleja el desempeño heterogéneo de los tres submodos:

- *Depth* alcanzó 100 % en condiciones controladas, pero presentó inestabilidad en pruebas de estrés.
- *Scene* obtuvo 66.67 %, requiriendo calibración con imágenes OV2640 y mejoras en retroalimentación.
- OCR registró 33.33 %, demostrando ser inadecuado para señales de tráfico.

Aunque el modo híbrido no cumple con el criterio global de precisión, cada submodo presenta potencial de aplicación en contextos diferentes a la asistencia en cruces peatonales: OCR en texto impreso, *depth* con sensores complementarios, y *scene* con expansión mediante NLP.

H4: latencia modo híbrido $<$ 2 segundos

Decisión: se rechaza H4

El modo híbrido presentó una latencia media de 4733.67 ms (4.7 segundos), excediendo el umbral de 2000 ms por un factor de 2.4. Sin embargo, el análisis por submodo revela resultados dispares:

- **OCR (1122 ms):** cumple con el criterio, demostrando que Tesseract puede procesar en tiempo aceptable.

- ***Scene* (6068 ms):** excede el umbral pero se mantiene en un rango potencialmente tolerable para aplicaciones no críticas.
- ***Depth* (7011 ms):** similar a *scene*, con alta variabilidad (SD = 3840 ms) que indica inestabilidad.

Aunque el modo híbrido no cumple con el criterio de latencia de forma global, la latencia reducida respecto al modo local (reducción del 63 %) sugiere que el procesamiento en dispositivo móvil es más eficiente que el procesamiento distribuido entre dos ESP32-S3. El submodo OCR demuestra que es posible alcanzar latencias aceptables cuando se emplean modelos optimizados.

H5: robustez operacional $\leq$ 5 % fallos

Decisión: se acepta H5
La robustez operacional cumple satisfactoriamente con el criterio establecido:

- **Modo local (1.67 %):** cumple ampliamente, con solo 1 fallo (desconexión Wi-Fi) en 60 pruebas.
- **Modo híbrido (4.55 %):** cumple claramente con el criterio (<5 %), con un promedio ponderado que considera las 110 pruebas totales (50 *scene* + 30 OCR + 30 *depth*):
  - *Scene* alcanzó 0 % de fallos en 50 pruebas (0 fallos), demostrando excelente estabilidad.
  - OCR registró 10 % de fallos en 30 pruebas (3 fallos), excediendo el umbral individual.
  - *Depth* registró 6.67 % de fallos en 30 pruebas (2 fallos), excediendo ligeramente el umbral individual.
  - **Cálculo del promedio ponderado:** $(0 \times 50 + 3 \times 1 + 2 \times 1)/110 = 5/110 =$ 4,55 %
  - Los fallos fueron *timeouts* recuperables sin *crashes* del sistema.

Ninguna modalidad experimentó *crashes* del sistema, y los fallos observados (*timeouts* de inferencia y desconexión Wi-Fi) fueron eventos recuperables que no comprometieron la integridad operacional del prototipo. La robustez es suficiente para un prototipo en etapa de validación técnica.

4.6.2. Hipótesis general

De forma global, la hipótesis general de la investigación se considera **rechazada parcialmente**, puesto que el sistema cumple con dos de las tres condiciones críticas establecidas:

- **Precisión $\geq$ 90 %:** cumplida en modo local (91.67 %), NO cumplida en modo híbrido (73.33 %)
- **Latencia < 2 segundos:** NO cumplida en modo local (12863 ms), NO cumplida en modo híbrido promedio (4733.67 ms)

- **Robustez $\leq$ 5 % fallos:** cumplida en modo local (1.67 %), cumplida en modo híbrido (4.55 %)

La hipótesis original planteaba que el prototipo sería capaz de detectar señales con precisión $\geq$ 90 % **Y** mantener latencia $<$ 2 segundos. Dado que la latencia no cumple el criterio en ninguna de las dos modalidades principales, la hipótesis general no puede aceptarse en su totalidad. Sin embargo, el sistema demuestra robustez operacional satisfactoria en ambas configuraciones, lo que valida la estabilidad fundamental del prototipo.

Análisis por modalidad:

- **Modo local:** cumple 2 de 3 criterios (precisión 91.67 % y robustez 1.67 %), pero la latencia crítica de 12.9 segundos lo descalifica para la aplicación objetivo de asistencia en cruces peatonales en tiempo real.
- **Modo híbrido:** cumple 1 de 3 criterios (robustez 4.55 %), con precisión insuficiente (73.33 %) y latencia excesiva (4733.67 ms). Sin embargo, presenta un camino viable hacia aplicaciones alternativas mediante reorientación de casos de uso.
- **Submodo OCR (modo híbrido):** aunque cumple con latencia (1122 ms) y robustez, su baja precisión en señales de tráfico (33.33 %) limita su aplicabilidad al contexto original. No obstante, demuestra potencial para lectura de texto impreso convencional.

Cumplimiento global del sistema:

Ninguna modalidad cumple simultáneamente con los tres criterios establecidos (precisión, latencia y robustez). El modo local alcanza 2/3 criterios, siendo la latencia el factor crítico de incumplimiento. Esta limitación es inherente a las capacidades computacionales del ESP32-S3 (dual-core 240 MHz) y evidencia que el enfoque de procesamiento local en microcontroladores de bajo costo, aunque económicamente accesible (54–86 USD), presenta compromisos significativos en desempeño temporal para aplicaciones críticas en tiempo real.

El sistema demuestra **viabilidad técnica parcial**, donde:

- La precisión del modo local (91.67 %) valida la capacidad del modelo LytNet V1 INT8 en *hardware* embebido.
- La robustez satisfactoria (1.67–4.55 %) confirma estabilidad operacional sin *crashes* del sistema.
- La latencia insuficiente (4.7–12.9 segundos) requiere migración a *hardware* más potente o reorientación a aplicaciones no críticas en tiempo.

Es importante destacar que el **submodo OCR** cumple individualmente con el criterio de latencia (1122 ms $<$ 2000 ms), demostrando que es técnicamente posible alcanzar tiempos de respuesta aceptables cuando se emplean modelos optimizados y se aprovecha la capacidad computacional de dispositivos móviles modernos. Este hallazgo sugiere que una arquitectura híbrida que ejecute **LytNet V1 en la aplicación móvil** (en lugar del ESP32-S3) podría reducir la latencia de detección de semáforos a rangos potencialmente aceptables (estimado 1.5–3 segundos), tal como proponen los autores originales del modelo (Yu et al., 2019a).

Capítulo 5

Conclusiones y recomendaciones

5.1. Introducción

En este capítulo se presentan las conclusiones principales del estudio, derivadas del análisis de resultados expuesto en el Capítulo 4, en relación con los objetivos planteados en el Capítulo 1. Asimismo, se describen los aportes más relevantes del trabajo, sus limitaciones y una serie de recomendaciones para futuras investigaciones y posibles mejoras del prototipo.

5.2. Conclusión general

La investigación permitió diseñar, implementar y evaluar un prototipo de sistema embebido *wearable* de bajo costo para la asistencia de personas con discapacidad visual, basado en un microcontrolador ESP32-S3 y modelos de visión por computadora optimizados. Los resultados obtenidos en laboratorio demostraron que el modo local alcanzó una precisión de 91.67 % (cumpliendo el criterio de $\geq$90 %); sin embargo, presentó una latencia crítica de 12863 ms que excede por un factor de 6.4 el umbral máximo de 2000 ms. El modo híbrido obtuvo una precisión promedio de 73.33 % (no cumpliendo el criterio de $\geq$90 %) con una latencia de 4733.67 ms (tampoco cumpliendo el umbral de $<$2000 ms).

Estos resultados confirman una **viabilidad técnica parcial** del enfoque propuesto: mientras el modo local cumple con precisión y robustez operacional, su latencia lo descalifica completamente para la aplicación objetivo de asistencia en cruces peatonales en tiempo real. El modo híbrido presenta un camino viable hacia aplicaciones alternativas cuando se reorienta adecuadamente: el submodo OCR para lectura de texto impreso (no señales de tráfico), el submodo *depth* con integración de sensores láser complementarios, y el submodo *scene* expandido mediante procesamiento de lenguaje natural para descripciones contextuales del entorno.

El prototipo desarrollado representa una alternativa tecnológica con potencial frente a las soluciones comerciales de alto costo, con un presupuesto de *hardware* de 54–86 USD, significativamente inferior a los 1899–3499 USD de dispositivos como Envision Glasses. Sin embargo, la brecha de desempeño en latencia evidencia que el bajo costo se traduce en limitaciones computacionales críticas que deben abordarse mediante arquitecturas híbridas o reorientación a casos de uso no críticos en tiempo.

5.3. Conclusiones según los objetivos específicos

5.3.1. Objetivo específico 1: captura y transmisión de imágenes

Crear un programa para el ESP32 S3 capaz de capturar imágenes mediante una cámara integrada con el fin de enviarla a la aplicación móvil en intervalos menores a 2 segundos, optimizando en la medida de lo posible el consumo energético.

Conclusión: el firmware desarrollado logró capturar y transmitir imágenes dentro del intervalo de 2 segundos cuando opera en modo híbrido, con latencias de captura de 477 ms y transmisión Wi-Fi de 50 ms (total 527 ms). En el modo local, la captura desde microSD tomó 1498 ms y la transmisión Wi-Fi 2513 ms, sumando 4011 ms que representa el 29.8 % de la latencia total. El objetivo de transmisión en <2 segundos se cumplió en el modo híbrido (527 ms), aunque no en el modo local (4011 ms).

La decisión metodológica de utilizar alimentación continua por USB-C permitió estabilidad operacional durante las pruebas (consumo activo de 54.4 mA @ 3.8V), pero no se implementaron optimizaciones de modos de bajo consumo ni gestión de batería, ya que esto quedó fuera del alcance de la fase de validación técnica inicial.

5.3.2. Objetivo específico 2: detección mediante inferencia local

Entrenar un modelo para que, en conjunto con un programa para el ESP32 S3 y TensorFlow Lite Micro, sea capaz de detectar de manera independiente señales y luces de tráfico con una precisión mínima de 90 %.

Conclusión: El sistema alcanzó un *accuracy* de 91.67 % en modo local (55 aciertos de 60 pruebas), **cumpliendo** la meta del 90 % establecida. El modelo LytNet V1 INT8 cuantizado demostró mantener su capacidad después del despliegue en el ESP32-S3, con *precision* promedio de 0.865 y *recall* de 0.897 sobre el dataset PTL completo (739 imágenes). La clase Countdown Green presentó el desafío principal con *precision* de 0.605, aunque con *recall* excelente de 0.907.

Los factores que influyeron positivamente incluyen: la arquitectura ligera de LytNet V1, la cuantización INT8 que redujo el modelo a 2.10 MB, el dataset PTL con ejemplos representativos, y la NPU del ESP32-S3 para acelerar operaciones. Sin embargo, la precisión del 91.67 % se acompaña de una latencia de inferencia de 8914 ms (66.2 % de la latencia total), limitando severamente la aplicabilidad práctica para decisiones en tiempo real.

5.3.3. Objetivo específico 3: aplicación móvil multiplataforma

Desarrollar una aplicación móvil, multiplataforma, basada en Flutter, integrada con TensorFlow para el procesamiento de entornos con una precisión mínima de 90 %.

Conclusión: la aplicación móvil se integró correctamente con el prototipo permitiendo ejecutar tres submodos (scene, OCR, *depth*). Sin embargo, la precisión promedio ponderada del modo híbrido fue de 73.33 % (11 aciertos de 15 pruebas), **no cumpliendo** la meta del 90 %. El desempeño fue heterogéneo: *scene* 66.67 % (podría mejorar con calibración específica y retroalimentación completa de objetos), OCR 33.33 % (inadecuado para señales de tráfico, potencialmente útil para texto impreso), *depth* 100 % (solo en condiciones controladas, con fallos en estrés).

El modo híbrido ofrece mayor versatilidad funcional mediante tres submodos especializados que abordan necesidades diferentes, aunque requieren refinamiento técnico para alcanzar niveles de precisión aceptables.

5.3.4. Objetivo específico 4: integración de componentes

Integrar los componentes del sistema (cámara, microcontrolador, aplicación móvil y salida de audio) para informar al usuario sobre su entorno mediante alertas de audio con una latencia inferior a 2 segundos.

Conclusión: la integración arquitectónica se logró satisfactoriamente, pero las alertas se generan con latencias significativamente superiores al límite de 2 segundos: modo local 12863 ms (factor 6.4 sobre umbral), modo híbrido 4733.67 ms promedio (scene 6068 ms, OCR 1122 ms, *depth* 7011 ms). Solo el submodo OCR cumple el criterio de <2000 ms. El criterio de latencia **no se cumplió en ninguna modalidad principal**.

El análisis por componentes identificó que la inferencia es el cuello de botella dominante: 66.2 % en modo local, 89.0 % en *scene*, 89.7 % en *depth*. Solo en OCR la inferencia representa el 53.6 %, explicando por qué este submodo logra cumplir el umbral.

5.3.5. Objetivo específico 5: sistema configurable

Programar un sistema integrado, configurable a través de la aplicación móvil que permita personalizar las alertas según las necesidades del usuario.

Conclusión: se implementaron opciones básicas de configuración mediante la aplicación móvil, incluyendo activación de submodos específicos del modo híbrido (scene, OCR, *depth*), configuración de parámetros de conectividad Wi-Fi, y visualización en tiempo real de resultados. Aunque la funcionalidad de personalización es aún limitada (no incluye ajuste de umbrales de confianza, frecuencia de captura, personalización de alertas auditivas, o perfiles de usuario), se comprobó la factibilidad técnica de ajustar el comportamiento del sistema remotamente sin modificar el firmware embebido. El objetivo se cumplió parcialmente.

5.3.6. Objetivo específico 6: facilidad de actualizaciones

Garantizar la facilidad de actualizaciones del sistema.

Conclusión: se definió una arquitectura modular que permite actualizar: firmware ESP32-S3 mediante recopilación y carga por USB-C, modelos de inferencia locales mediante sustitución de archivos `.tflite`, aplicación móvil mediante Google Play Store/Apple App Store, y modelos del modo híbrido mediante nuevas versiones de la aplicación Flutter. La principal limitación es que actualizaciones del firmware requieren intervención técnica (conexión USB, *software* de desarrollo), mientras que una solución comercial debería implementar actualización OTA. El objetivo se cumplió satisfactoriamente para un prototipo de investigación.

5.4. Aportes del trabajo

5.4.1. Aportes técnicos

- Arquitectura dual que combina procesamiento local autónomo y distribuido, permitiendo balancear autonomía con capacidad computacional según el contexto.
- Demostración de que microcontroladores de <$25 USD pueden alcanzar precisión >91 % en detección de señales, desafiando la percepción de que soluciones efectivas requieren inversiones >1500 USD.
- Protocolo de pruebas replicable (32 pruebas en 4 escenarios) para evaluación de prototipos asistivos sin involucrar usuarios finales en fases tempranas.
- Cuantificación del trade-off costo-desempeño: ESP32-S3 procesa 322x más lento que PC (27.67 ms vs. 8914 ms), pero con costo 10–50x menor.
- Validación experimental de limitaciones de modelos monoculares de profundidad (100 % en condiciones controladas, múltiples fallos en estrés), respaldando necesidad de sensores ToF.

5.4.2. Aportes académicos

- Caso de estudio para investigación en visión por computadora y tecnologías asistivas, aplicable como referencia en futuros trabajos de grado.
- Metodología de validación técnica sin usuarios finales en fase inicial, reduciendo barreras éticas/logísticas manteniendo rigor científico.
- Documentación del proceso de integración de TensorFlow Lite en microcontroladores limitados (cuantización INT8, optimización de memoria, manejo de NPU).
- Análisis comparativo cuantitativo entre inferencia local e híbrida, demostrando que el procesamiento móvil reduce latencia en 63 %, pero introduce complejidad y dependencia de Wi-Fi.

5.4.3. Aportes sociales potenciales

- Ruta tecnológica hacia dispositivos de asistencia visual más accesibles para comunidades de ingresos medios/bajos en Panamá y Latinoamérica.
- Demostración de viabilidad parcial que puede incentivar políticas públicas, programas gubernamentales o emprendimientos de impacto social.
- Precedente académico que motiva desarrollo de soluciones con enfoque en accesibilidad e inclusión social.
- Identificación de aplicaciones alternativas viables (OCR para texto impreso, *scene* con NLP, *depth* con ToF) que abren caminos para productos especializados comercialmente viables.

5.5. Limitaciones del estudio

5.5.1. Limitaciones metodológicas

- **Entorno simulado de laboratorio:** pruebas en condiciones controladas (5m × 5m, LED 5500K, 26–30°C) que no reflejan variables de entornos urbanos reales: tráfico dinámico, peatones en movimiento, condiciones climáticas adversas, ruido visual, iluminación natural variable, superficies reflectantes.
- **Ausencia de validación con usuarios finales:** no se evaluaron aspectos críticos de usabilidad real, comodidad física, facilidad de operación con tecnologías de asistencia (TalkBack/VoiceOver), adaptabilidad a contextos cotidianos, curva de aprendizaje, ni impacto percibido en autonomía y calidad de vida.
- **Tamaño de muestra limitado:** 32 pruebas son suficientes para validación técnica inicial, pero no permiten generalización estadística robusta. Un estudio exhaustivo requeriría 300–500 pruebas en condiciones variadas.

5.5.2. Limitaciones técnicas

- **Dependencia de datasets generales:** LytNet V1 entrenado con PTL de contextos asiáticos (China), DeepLabV3-Plus con Pascal VOC 2012, Depth-Anything V2 con datos indoor norteamericanos. Estos datasets podrían no representar perfectamente señales panameñas, contextos urbanos centroamericanos, condiciones de iluminación tropical, ni objetos culturalmente específicos, requiriendo fine-tuning regional.
- **Alimentación por USB-C:** limita portabilidad a pruebas de laboratorio o escenarios con powerbank. Sistema comercial requeriría batería Li-Po 2000–3000 mAh con autonomía 8+ horas.
- **Latencia crítica del modo local:** 12.9 segundos descalifica al modo local para cruces peatonales, donde el semáforo puede cambiar durante el procesamiento. Esta limitación es inherente al ESP32-S3 (240 MHz, NPU básica).
- **Precisión limitada de *depth*:** 100 % en pruebas controladas pero múltiples fallos en estrés (6.67 %), confirmando que mapas monoculares no son confiables para decisiones críticas. Requiere sensores láser ToF obligatorios.
- **Inaplicabilidad de OCR en señales:** Tesseract inadecuado para señales (33.33 %) por falta de especialización en tipografías no convencionales, sensibilidad a ángulos (fallo total con 30°) e interpretación errónea de elementos gráficos.
- **Cobertura limitada de *scene*:** Pascal VOC reconoce 20 clases de objetos cotidianos pero **no incluye infraestructura urbana** crítica: aceras, calzadas, semáforos, señales verticales, pasos de cebra, rampas. Requiere reentrenamiento con Cityscapes o Mapillary Vistas.

5.5.3. Limitaciones de contexto

- **Enfoque en señales de tráfico:** no aborda otras necesidades críticas como lectura de documentos, reconocimiento de productos, identificación de billetes, o navegación en interiores.
- **Idioma único:** OCR y síntesis de voz probados únicamente en español, limitando aplicabilidad en contextos multilingües (Panamá presenta señalización en inglés en zonas turísticas).
- **Ausencia de integración:** diseñado como sistema *standalone*, sin considerar integración con bastones inteligentes, GPS, transporte público, o plataformas de comunicación con cuidadores.

5.6. Recomendaciones

5.6.1. Recomendaciones técnicas para mejora del prototipo

1. **Migración de LytNet V1 al modo híbrido (solución directa para latencia):** implementar el modelo LytNet V1 en la aplicación Flutter en lugar del ESP32-S3, replicando el enfoque original de los autores (Yu et al., 2019a) que ejecutaron el modelo en dispositivos móviles Android/iOS. Esta arquitectura aprovecharía la capacidad computacional superior de smartphones modernos (procesadores 8 núcleos 2–3 GHz, GPUs Mali/Adreno, NPU dedicados) vs. ESP32-S3 (dual-core 240 MHz).

   Estimación de latencia: basándose en otros submodos híbridos (scene 6068 ms, OCR 1122 ms), se estima que LytNet V1 en móvil alcanzaría **1500–3000 ms** (reducción 76–88 % vs. 12863 ms actuales). Costo adicional: 0 USD (usa infraestructura existente). Implementación: convertir modelo a TensorFlow Lite compatible con `tflite_flutter`, integrar en flujo de modo híbrido, validar con 60 imágenes test split PTL.

   Esta migración revertiría la arquitectura a la configuración original de los autores (Yu et al., 2019a), validando que el diseño inicial era el más apropiado para la aplicación objetivo.

2. **Integración obligatoria de sensores láser ToF:** reemplazar Depth-Anything V2 por sensor láser de alta precisión (VL53L1X con rango 4m, precisión ±5 %, costo $6–8 USD) que ofrece mediciones directas de distancia con latencia <50 ms, eliminando dependencia de modelos de IA poco confiables para funcionalidad crítica de seguridad. Alternativamente, usar el modelo como contexto general y el ToF para validación de distancias críticas.

3. **Reentrenamiento de *scene* con datasets urbanos:** migrar de Pascal VOC a datasets especializados (Cityscapes con 30 clases incluyendo aceras/calles/señales, Mapillary Vistas con 66 clases) o crear dataset local mediante captura de 5,000+ imágenes de contextos panameños. Fine-tuning con transfer learning podría alcanzar 85–90 % en 2–3 semanas con GPU de gama media.

4. **Sustitución de Tesseract por OCR especializado:** reemplazar por modelo especializado en Text Spotting (EasyOCR, PaddleOCR, CRAFT+CRNN) entrenado

con dataset de señales, o reorientar exclusivamente a texto impreso convencional donde Tesseract tiene desempeño aceptable.

5. **Batería Li-Po con gestión inteligente:** desarrollar versión portable mediante integración de batería 2500–3000 mAh, controlador de carga (TP4056), optimización con modos de bajo consumo (light sleep cuando inactivo, ajuste dinámico de frecuencia de captura, apagado de Wi-Fi cuando no se usa modo híbrido). Costo adicional: 12–18 USD, autonomía esperada: 6–10 horas.

6. **Actualización OTA (Over-The-Air):** implementar capacidad de actualizar firmware y modelos remotamente sin USB mediante: partición dual en Flash (boot seguro), servidor de actualizaciones con verificación MD5/SHA-256, rollback automático si falla, compresión de modelos durante transmisión.

5.6.2. Recomendaciones para investigaciones futuras

1. **Evaluación con usuarios finales:** desarrollar fase con personas con discapacidad visual (n=15–30), previa aprobación ética institucional. Incluir: pruebas de usabilidad en entornos controlados, evaluación de comodidad física (System Usability Scale, NASA-TLX), medición de impacto en autonomía (FIM pre/post), retroalimentación cualitativa (entrevistas semiestructuradas), y aceptabilidad tecnológica (TAM/UTAUT).

2. **Validación en entornos urbanos reales:** extender pruebas a escenarios exteriores en ciudad de Panamá (n=200–300 pruebas) incluyendo: diferentes condiciones climáticas (sol intenso, lluvia, nubosidad), horarios variados (amanecer, mediodía, atardecer, noche), niveles de tráfico (hora pico/valle), y tipos de vías (avenidas, calles residenciales, zonas peatonales). Objetivo: determinar si la precisión de 91.67 % se mantiene en condiciones reales.

3. **Integración con navegación GPS y realidad aumentada auditiva:** investigar fusión del prototipo con aplicaciones de navegación (Google Maps, Waze, Lazarillo) para orientación direccional combinada con detección de señales. Explorar spatial audio binaurial que comunique dirección de objetos mediante sonido 3D.

4. **Detección proactiva de situaciones de riesgo:** ampliar capacidades para detectar peligros dinámicos en tiempo real: vehículos en aproximación (estimación de trayectoria y tiempo hasta colisión), ciclistas/motocicletas laterales, cambios de estado de semáforos, obstáculos móviles, construcciones en vía pública. Requiere procesamiento de video continuo con modelos temporales (3D CNNs, TimeSformer).

5. **Estudio longitudinal de adopción:** estudio a mediano plazo (6–12 meses) con grupo de usuarios voluntarios (n=10–15) para evaluar: patrones de uso real, evolución de satisfacción, durabilidad del *hardware*, barreras para adopción sostenida, e impacto en calidad de vida mediante WHOQOL-BREF pre/post.

6. **Desarrollo de módulos para aplicaciones alternativas:** dado que latencia de 12.9s es tolerable en ciertos contextos, investigar aplicaciones donde el modo local sea viable: lectura de documentos (usuario posiciona cámara, espera 10–15s, recibe

lectura), reconocimiento de productos en supermercados, identificación de billetes, descripción de escenas estáticas.

5.6.3. Recomendaciones para la institución

1. **Línea de investigación en tecnologías asistivas:** establecer formalmente línea institucional incentivando proyectos de tesis que tomen este prototipo como base para iteraciones mejoradas, exploren aplicaciones para otras discapacidades (auditiva, motora, cognitiva) e investiguen metodologías de codiseño con usuarios finales.

2. **Laboratorio especializado:** gestionar creación de laboratorio equipado con *hardware* de prototipado (microcontroladores, sensores, actuadores), herramientas de desarrollo (estaciones con GPUs, impresoras 3D, equipos de soldadura), y espacios para pruebas con usuarios. Presupuesto estimado: 15,000–25,000 USD.

3. **Dataset panameño colaborativo:** promover proyecto multisemestre para crear datasets locales mediante captura colaborativa por estudiantes, anotación con herramientas open-source (LabelImg, CVAT), categorías de señales panameñas/escenas urbanas/productos locales, y publicación bajo Creative Commons. Meta: 10,000+ imágenes en 2 años.

4. **Vinculación con organizaciones de discapacidad visual:** establecer convenios con IPHE, Asociación de Ciegos San José, Patronato Nacional, SOPADIVI para facilitar acceso a usuarios voluntarios, recolectar necesidades reales, validar prototipos con retroalimentación directa y potencial transferencia tecnológica.

5.7. Trabajos futuros específicos

1. **Prototipo v2.0 con batería:** batería Li-Po 2500 mAh, carcasa ergonómica 3D, sensor ToF VL53L1X, optimización de consumo. Costo: 75–110 USD. Duración: 6–9 meses. Tipo: tesis de grado Ingeniería Electrónica/Mecatrónica.

2. **Aplicación móvil v2.0:** interfaz 100 % accesible (TalkBack/VoiceOver), integración de GPT-2 small para descripciones contextuales, configuración avanzada (perfiles, umbrales), modo entrenamiento. Duración: 4–6 meses. Tipo: tesis de grado Ingeniería de Software.

3. **Módulo de navegación indoor:** detección de puertas/pasillos/escaleras, SLAM ligero con cámara monocular + IMU, integración con mapas de edificios públicos. Duración: 8–12 meses. Tipo: tesis de maestría en Robótica.

4. **Plataforma open-source:** publicación completa en GitHub bajo MIT: código firmware/app, scripts de entrenamiento, esquemáticos, modelos 3D, documentación técnica, videos tutoriales. Objetivo: facilitar replicación global por comunidad maker.

5. **Estudio piloto con usuarios (n=10):** primera validación mediante protocolo simplificado: 30 min capacitación, 60 min uso supervisado, 30 min entrevista. Métricas: SUS, NASA-TLX, entrevista cualitativa. Duración: 2–3 meses. Tipo: práctica profesional.

5.8. Reflexión final

El desarrollo de este prototipo demuestra que la intersección entre ingeniería, inteligencia artificial y compromiso social puede generar soluciones con potencial de impacto tangible en la calidad de vida de personas vulnerables. Aunque el camino desde la creación de un prototipo de laboratorio hasta volverse un producto comercial es complejo, este trabajo establece bases técnicas sólidas y demuestra la factibilidad de democratizar tecnologías asistivas mediante bajo costo y diseño abierto.

Esta experiencia refuerza la importancia de formar ingenieros con competencias técnicas y sensibilidad social, capaces de identificar problemas reales donde la tecnología sea herramienta de inclusión. En un contexto donde persisten brechas digitales e inequidad en acceso a servicios básicos para personas con discapacidad, la ingeniería tiene la responsabilidad ética de orientar esfuerzos hacia soluciones que beneficien a quienes más lo necesitan.

Los resultados, aunque mixtos (precisión cumplida, latencia no cumplida), son valiosos porque revelan con honestidad las limitaciones reales de tecnologías de bajo costo. Esta transparencia es fundamental para el avance científico: permite que futuras investigaciones eviten caminos infructuosos, identifiquen rutas alternativas prometedoras, y tengan expectativas realistas sobre lo técnicamente viable con recursos limitados. El fracaso del modo local en latencia no es negativo per se, sino información crucial que reorienta la investigación hacia arquitecturas más adecuadas.

5.9. Cierre

En conclusión, este proyecto cumplió parcialmente con los objetivos técnicos propuestos: alcanzó precisión de 91.67 % en modo local (superando el 90 %) y robustez operacional adecuada (1.67 % de fallos), pero no logró cumplir con el criterio crítico de latencia (<2 segundos), registrando 12.9 segundos en modo local y 4.7 segundos en modo híbrido. El modo híbrido obtuvo precisión de 73.33 %, con robustez de 4.55 % de fallos, aunque demostró mayor versatilidad funcional mediante tres submodos especializados.

Los principales logros incluyen: validación de que microcontroladores <$25 USD pueden ejecutar modelos de visión con precisión >90 %, cuantificación del trade-off costo-desempeño (322x diferencia de velocidad vs. PC), identificación de limitaciones críticas de tecnologías monoculares, demostración de arquitectura dual viable, y establecimiento de protocolo experimental replicable.

Aunque son necesarias fases adicionales de validación con usuarios finales, mejoras de *hardware* para reducir latencia, y reorientación hacia aplicaciones alternativas viables, el prototipo constituye un avance significativo hacia soluciones tecnológicas accesibles que contribuyan a la autonomía de personas con discapacidad visual. El verdadero impacto se medirá por su capacidad de inspirar futuras investigaciones, generar conocimiento transferible y, eventualmente, traducirse en herramientas que transformen vidas reales.

Referencias

Ali Hassan, E., & Tang, T. B. (2016). Smart glasses for the visually impaired people. En K. Miesenberger, C. Bühler & P. Penaz (Eds.), *Computers Helping People with Special Needs* (pp. 579-582, Vol. 9759). Springer International Publishing. https://doi.org/10.1007/978-3-319-41267-2_82

Ariza, J. Á., & Pearce, J. M. (2022). Low-cost assistive technologies for disabled people using open-source hardware and software: a systematic literature review. *IEEE Access*, *10*, 124894-124927. https://doi.org/10.1109/ACCESS.2022.3221449

Baig, M. S. A., Gillani, S. A., Shah, S. M., Aljawarneh, M., Khan, A. A., & Siddiqui, M. H. (2024, diciembre). Ai-based wearable vision assistance system for the visually impaired: integrating real-time object recognition and contextual understanding using large vision-language models [arXiv:2412.20059]. https://doi.org/10.48550/arXiv.2412.20059

Bourne, R. R. A., Jonas, J. B., Friedman, D., Nangia, V., Bron, A., Tapply, I., Fernandes, A. G., Cicinelli, M. V., Arrigo, A., Leveziel, N., Resnikoff, S., Taylor, H. R., Sedighi, T., Bikbov, M. M., Braithwaite, T., Cheng, C.-Y., Congdon, N., Del Monte, M. A., Ehrlich, J. R., … the GBD 2019 Blindness and Vision Impairment Collaborators. (2024). Global estimates on the number of people blind or visually impaired by glaucoma: A meta-analysis from 2000 to 2020. *Eye*, *38*(11), 2036-2046. https://doi.org/10.1038/s41433-024-02995-5

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation [DeepLabV3+]. *Proceedings of the European Conference on Computer Vision (ECCV)*, 801-818. https://doi.org/10.1007/978-3-030-01234-2_49

European Parliament. Directorate General for Parliamentary Research Services. (2018). *Assistive technologies for people with disabilities. Part II, Current and emerging technologies.* Publications Office. Consultado el 15 de junio de 2025, desde https://data.europa.eu/doi/10.2861/567013

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303-338. https://doi.org/10.1007/s11263-009-0275-4

Fernandez, J. M., Valencia, A., Daniel, V., & Cunanan, D. C. D. (2024). Assistive eyewear: arduino powered eyeglasses for blind individuals. *IC-ITECHS*, *5*(1), 316-322. https://doi.org/10.32664/ic-itechs.v5i1.1671

Global report on assistive technology (1st ed). (2022). World Health Organization.

Joshi, M., Shukla, A., Srivastava, J., & Rastogi, M. (2023, marzo). Drishti: visual navigation assistant for visually impaired [arXiv:2303.07451]. https://doi.org/10.48550/arXiv.2303.07451

Kumar Jha, A., Jain, P., Shankar, A., & B, S. (2023). Aeye - A Smart Glasses. *International Journal of Engineering Research & Technology (IJERT)*, *12*(05), 129-134. https://www.ijert.org/research/aeye-a-smart-glasses-IJERTV12IS050031.pdf

Li, G., Xu, J., Li, Z., Chen, C., & Kan, Z. (2023). Sensing and navigation of wearable assistance cognitive systems for the visually impaired. *IEEE Transactions on Cognitive and Developmental Systems*, *15*(1), 122-133. https://doi.org/10.1109/TCDS.2022.3146828

Salazar, R. D. V., & Mesa, A. A. C. (2019). Dispositivos de asistencia para la movilidad en personas con discapacidad visual: una revisión bibliográfica. *Revista Politécnica*, *15*(28), 107-116. https://doi.org/10.33571/rpolitec.v15n28a10

Smart Glass System for Visually Impaired Using ESP32 Camera and OCR. (2025). *International Journal of Scientific Research in Engineering and Management (IJSREM)*, *09*(05), 1-9.

Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, *2*, 629-633. https://doi.org/10.1109/ICDAR.2007.4376991

Szekely, R., Holloway, C., & Bandukda, M. (2025). Understanding the psychosocial impact of assistive technologies for people with visual impairments: protocol for a scoping review. *JMIR Research Protocols*, *14*(1), e65056. https://doi.org/10.2196/65056

Tokmurziyev, I., Cabrera, M. A., Khan, M. H., Mahmoud, Y., Moreno, L., & Tsetserukou, D. (2025, marzo). Llm-glasses: genai-driven glasses with haptic feedback for navigation of visually impaired people [arXiv:2503.16475]. https://doi.org/10.48550/arXiv.2503.16475

V M, D. V., A, D. K., G C, M. K., Bilal, M., & S, S. (2024). Ai powered smart glass for visually impaired individuals. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *08*(12), 1-6. https://doi.org/10.55041/IJSREM39504

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth Anything V2 [Metric Indoor Small variant]. *arXiv preprint arXiv:2406.09414*. https://huggingface.co/depth-anything/Depth-Anything-V2-Metric-Indoor-Small-hf

Yu, S., Lee, H., & Kim, J. (2019a). LYTNet: A Convolutional Neural Network for Real-Time Pedestrian Traffic Lights and Zebra Crossing Recognition for the Visually Impaired. *Computer Analysis of Images and Patterns (CAIP)*.

Yu, S., Lee, H., & Kim, J. (2019b). Street Crossing Aid Using Light-Weight CNNs for the Visually Impaired. *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

Zaman, N., Potluri, V., Biggs, B., & Coughlan, J. M. (2025, mayo). Whatsai: transforming meta ray-bans into an extensible generative ai platform for accessibility [arXiv:2505.09823]. https://doi.org/10.48550/arXiv.2505.09823