

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA SEGURA PARA EL CONTROL DE ACCESOS VEHICULARES Y GESTIÓN DE ESTACIONAMIENTOS EN EDIFICIOS RESIDENCIALES

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL TELEMÁTICO

Diego Alonso Maldonado Martínez

PROFESORA GUÍA: Patricia Franco
PROFESOR CORREFERENTE: Luis Lizama

Santiago - Chile
2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo:
DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA SEGURA PARA EL CONTROL DE ACCESOS VEHICULARES Y GESTIÓN DE ESTACIONAMIENTOS EN EDIFICIOS RESIDENCIALES

Nombre del candidato(a): Diego Alonso Maldonado Martínez

Carrera / Grado: Ingeniería Civil Telemática

Campus: San Joaquín **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Patricia Franco Troya, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 23/10/2026 **Firma:** PATRICIA FRANCO TROYA Firmado digitalmente por PATRICIA FRANCO TROYA Fecha: 2026.10.23 14:20:21 -03'00'

Estudiante o Candidato(a):

Fecha: 23/10/2026 **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero agradecer, en primer lugar, a Kathalina Antonia Pino Gonzalez, por su amor, su compañía y su apoyo incondicional durante estos últimos años. Su confianza en mí ha sido una de las fuerzas más importantes detrás de todo lo que he logrado, no solo en la universidad o en el trabajo, sino también en la vida. En los momentos de mayor cansancio o inseguridad, siempre ha sabido estar cerca, darme ánimo y recordarme de lo que soy capaz. Su forma de hacerme sentir orgulloso de mí mismo, incluso cuando a mí me cuesta verlo, ha sido fundamental en este proceso. Agradezco también profundamente a su familia, por la cercanía, el cariño y la generosidad con que siempre me han recibido y acompañado.

A mi mamá, a mi papá y a mi hermana, gracias por estar siempre presentes y por hacerme sentir su orgullo de manera tan sincera a lo largo de esta etapa. Saber que han valorado mi esfuerzo, mi constancia y mi desempeño ha sido una motivación muy importante para seguir avanzando, incluso en los momentos más exigentes.

A mis amigos y compañeros de carrera, gracias por la amistad, la buena convivencia y el apoyo mutuo que marcaron estos años. Gran parte de este camino fue posible gracias al trabajo en conjunto, a las horas compartidas de estudio, a los proyectos desarrollados entre todos y a esa disposición constante de ayudarnos cuando alguno lo necesitaba. Esa red de apoyo hizo mucho más llevadero el proceso universitario y dejó aprendizajes que van mucho más allá de lo académico.

Finalmente, agradezco a todas y todos los profesores de la carrera por la formación entregada a lo largo de estos años. De manera muy especial, agradezco al profesor Luis Lizama, quien me acompañó como profesor guía durante el año 2025, etapa en la que se desarrolló gran parte de esta memoria, y cuyo apoyo fue fundamental para orientar el trabajo. Asimismo, agradezco sinceramente a la profesora Patricia Franco por su disposición a acogerme como estudiante guía en la etapa final del proceso, por sus observaciones, comentarios y apoyo constante para poder consolidar y finalizar esta memoria. A ambos, muchas gracias por su tiempo, dedicación y confianza.

Dedicatoria

Dedico esta memoria a mi familia, por su amor y apoyo incondicional, y a Kathalina, por acompañarme, creer en mí y recordarme siempre, incluso en los momentos difíciles, que el esfuerzo vale la pena.

Resumen

La seguridad en edificios residenciales y la gestión eficiente de espacios comunes constituyen desafíos relevantes en ciudades con comunidades de alta rotación de visitas, estacionamientos compartidos y dependencia de conserjería. Los sistemas tradicionales de control de acceso vehicular, basados en procesos manuales, presentan limitaciones de eficiencia operativa, trazabilidad y seguridad. Estas condiciones incrementan el riesgo de accesos indebidos, retrasos en la operación y pérdida de evidencia sobre eventos críticos.

Esta memoria diseña e implementa una plataforma de control de acceso vehicular y gestión de estacionamientos para edificios residenciales. El enfoque integra criterios de ciberseguridad desde la arquitectura de un sistema ciberfísico compuesto por visión por computador, tecnologías de Internet de las Cosas (IoT), backend web e interfaces de operación. Cada componente amplía la superficie de ataque, por lo que se consideran controles coordinados en las capas física, lógica y operacional.

La plataforma incorpora un backend desarrollado en FastAPI, una aplicación web para la gestión de accesos y reservas, una red de sensores IoT para monitorear ocupación y un pipeline de visión por computador. Los detectores finales se entrenaron con los conjuntos *huaung77* y *License Plate Recognition* publicados en Roboflow. Sobre esta base se implementaron autenticación basada en JSON Web Tokens (JWT), validación contextual por etapas y registros de eventos para abordar riesgos de suplantación, abuso de API e intercepción de comunicaciones.

Los resultados muestran la factibilidad técnica del enfoque bajo las condiciones evaluadas. Las pruebas del backend móvil verificaron el rechazo de credenciales, tokens y entradas inválidas, mientras que el piloto permitió observar el flujo integrado de detección, autorización, actuación y trazabilidad. Las medidas de red, autenticación robusta de dispositivos y resistencia completa frente a ataques de presentación se mantienen como requisitos pendientes para un despliegue productivo.

Palabras clave: Ciberseguridad, Control de acceso, Internet de las Cosas, Sistemas ciberfísicos, Visión por computador.

Abstract

Security in residential buildings and the efficient management of shared spaces are relevant challenges in cities with communities that have frequent visitor turnover, shared parking spaces, and dependence on concierge staff. Traditional vehicle access control systems, based on manual procedures, present limitations in operational efficiency, traceability, and security. These conditions increase the risk of unauthorized access, operational delays, and loss of evidence regarding critical events.

This thesis designs and implements a vehicle access control and parking management platform for residential buildings. The approach integrates cybersecurity criteria into the architecture of a cyber-physical system composed of computer vision, Internet of Things (IoT) technologies, a web backend, and operational interfaces. Each component expands the attack surface; therefore, coordinated controls are considered across the physical, logical, and operational layers.

The platform incorporates a backend developed with FastAPI, a web application for access and reservation management, an IoT sensor network for occupancy monitoring, and a computer vision pipeline. The final detectors were trained with the *huaung77* and *License Plate Recognition* datasets published on Roboflow. Based on this platform, JSON Web Token (JWT) authentication, staged contextual validation, and event logging were implemented to address risks of spoofing, API abuse, and communication interception.

The results show the technical feasibility of the approach under the evaluated conditions. Tests on the mobile backend verified the rejection of invalid credentials, tokens, and inputs, while the pilot allowed observation of the integrated detection, authorization, actuation, and traceability flow. Network controls, robust device authentication, and complete resistance to presentation attacks remain pending requirements for a production deployment.

Keywords: Computer vision, Cyber-physical systems, Cybersecurity, Internet of Things, Vehicle access control.

Índice general

1. Introducción **1**
1.1. Contexto . . . 1
1.2. Problemática . . . 1
1.3. Objetivos del Proyecto . . . 2
1.3.1. Objetivo General . . . 2
1.3.2. Objetivos Específicos . . . 2
1.4. Estructura de la Memoria . . . 2

2. Estado del Arte y Marco Teórico **3**
2.1. Estado del Arte . . . 3
2.1.1. Control de acceso vehicular: contexto general y motivación tecnológica . . . 3
2.1.2. Del control vehicular convencional al enfoque ciberfísico residencial . . . 3
2.1.3. Revisión de plataformas residenciales y brechas de integración seguridad-operación . . . 5
2.1.4. Control de acceso y seguridad en edificios inteligentes . . . 8
2.1.5. Sistemas ciberfísicos (CPS) y arquitectura por capas . . . 9
2.1.6. Reconocimiento Automático de Patentes y Detección Vehicular . . . 10
2.1.7. Amenazas de suplantación en ALPR y estrategias de mitigación . . . 11
2.1.8. Arquitectura de detección desacoplada: justificación técnica y de seguridad 12
2.1.9. Gestión de Estacionamientos: Visión por Computador versus Sensores IoT 13
2.1.10. Análisis comparativo de trabajos relacionados (2021–2026) . . . 14
2.1.11. Síntesis de brechas y posicionamiento del aporte . . . 15
2.2. Marco Teórico . . . 16
2.2.1. Sistemas Ciberfísicos, IoT y Comunicación . . . 16
2.2.2. Visión por Computador y Aprendizaje Profundo . . . 17
2.2.3. Arquitectura YOLO (You Only Look Once) . . . 20
2.2.4. Detección vehicular y clasificación supervisada . . . 20
2.2.5. Métricas de Evaluación de Modelos . . . 21
2.2.6. Reconocimiento Óptico de Caracteres (OCR) . . . 22
2.2.7. Internet de las Cosas (IoT) y Comunicación . . . 23
2.2.8. Arquitectura de Software y Seguridad . . . 24

3. Metodología y Diseño de la Plataforma Segura ResiPark **27**
3.1. Metodología de Desarrollo . . . 27
3.2. Arquitectura General de la Plataforma Segura . . . 28
3.3. Modelo de Amenazas y Vulnerabilidades Consideradas . . . 30
3.3.1. Amenazas Priorizadas . . . 30

3.3.2. Criterios de Mitigación Adoptados . . . 31
3.4. Ciberseguridad Transversal por Capas . . . 32
3.5. Desarrollo del Backend y Seguridad . . . 32
3.5.1. Diseño Modular y Gestión de Rutas . . . 33
3.5.2. Implementación de Autenticación con JWT . . . 34
3.6. Ingeniería de Modelos de Inteligencia Artificial . . . 35
3.6.1. Conjuntos de datos y detectores integrados . . . 35
3.6.2. Entrenamiento y Selección de Arquitectura . . . 35
3.7. Prototipado y Validación (Prueba de Concepto) . . . 36
3.7.1. Optimización Computacional y Evolución del OCR . . . 37
3.7.2. Evolución del Módulo OCR . . . 37
3.8. Integración Frontend . . . 37
3.8.1. Actualización en Tiempo Real . . . 37
3.9. Implementación de los Componentes por Capas . . . 38
3.10. Capa de Obtención de Datos (Edge Computing) . . . 39
3.10.1. Módulo de Visión y Acceso (Raspberry Pi) . . . 39
3.10.2. Red de Sensores de Ocupación . . . 39
3.11. Capa de Red y Comunicaciones . . . 40
3.11.1. Topología LoRa para Subterráneos . . . 40
3.11.2. Medidas Recomendadas de Seguridad en la Transmisión . . . 41
3.12. Capa de Soporte y Lógica de Negocio (Backend) . . . 42
3.12.1. Persistencia, Seguridad y Tiempo Real . . . 42
3.12.2. API RESTful . . . 42
3.13. Capa de Aplicación y Experiencia de Usuario . . . 43
3.13.1. Aplicación Móvil (Residentes) . . . 43
3.13.2. Plataforma Web (Administración) . . . 45

4. Resultados, Validación Experimental y Análisis de Seguridad 48
4.1. Alcance y Criterios de Evaluación . . . 48
4.2. Resultado del Objetivo 1: Identificación de Amenazas y Superficie de Ataque . . . 48
4.3. Resultado del Objetivo 2: Backend y Capa de Aplicación . . . 49
4.3.1. Verificación Funcional de la Plataforma Web . . . 49
4.4. Resultado del Objetivo 3: Visión por Computador y Validación de Acceso . . . 50
4.4.1. Métricas de Entrenamiento . . . 50
4.4.2. Comparativa de Motores OCR . . . 52
4.5. Validación Transversal de Medidas de Ciberseguridad . . . 53
4.5.1. Protocolo de pruebas de autenticación, autorización y entradas . . . 53
4.5.2. Pruebas funcionales preliminares frente a escenarios de suplantación visual 53
4.5.3. Mitigación de Abuso de API y Sesión . . . 58
4.5.4. Mitigación en Red e Infraestructura IoT . . . 59
4.6. Resultado del Objetivo 4: Factibilidad Técnica y Operación en Terreno . . . 59
4.6.1. Fiabilidad de la Detección Física . . . 59
4.6.2. Desempeño de la Conectividad LoRa . . . 59
4.6.3. Pruebas de Integración y Campo (End-to-End) . . . 60
4.6.4. Retroalimentación Cualitativa de Usuarios . . . 61
4.7. Discusión de Resultados y Limitaciones . . . 61
4.7.1. Análisis Crítico de Seguridad . . . 62
4.7.2. Limitaciones del Prototipo . . . 63

4.7.3. Proyección y Transferencia Tecnológica 63

5. Conclusiones y Trabajo Futuro **64**
5.1. Trabajo Futuro 65

Referencias Bibliográficas **67**

Índice de tablas

2.1. Comparación funcional entre control vehicular convencional y enfoque ciberfísico residencial 5
2.2. Vulnerabilidades frecuentes en plataformas residenciales y subsistemas de acceso no integrados 7
2.3. Amenazas relevantes y mitigaciones propuestas/analizadas para la arquitectura. . 8
2.4. Resumen de seguridad transversal por capas en la arquitectura analizada. 9
2.5. Comparación de arquitecturas por capas en sistemas IoT/CPS [1, 2]. 10
2.6. Comparación de enfoques para reconocimiento de patentes y detección vehicular. 11
2.7. Comparación de enfoques para detección de ocupación de estacionamientos. . . . 14
2.8. Comparativa de trabajos relacionados en control de acceso inteligente y ALPR (2021–2026). 15
2.9. Tipología de fuentes y rol metodológico en el estado del arte. 16
2.10. Evolución de enfoques de visión por computador relevantes para detección vehicular 19
2.11. Comparación conceptual de enfoques OCR aplicables a lectura de patentes . . . 23
2.12. Comparación de tecnologías inalámbricas para monitoreo de estacionamientos. . 25

3.1. Activos críticos y puntos de entrada asociados al modelo de amenazas 30
3.2. Relación entre capa, activo a proteger, amenaza identificada y mecanismo de mitigación 32
3.3. Criterios de selección de FastAPI para el alcance del prototipo 33
3.4. Configuración documentada de los entrenamientos finales integrados al prototipo 35
3.5. Endpoints principales de la API de ResiPark. 43

4.1. Métricas globales de validación de los detectores finales. 51
4.2. Observaciones de integración que motivaron el cambio de motor OCR. 52
4.3. Resultados de pruebas controladas sobre autenticación, autorización y validación de entradas del backend móvil. 54
4.4. Matriz de validación de seguridad alineada con las amenazas priorizadas. 55
4.5. Resultados observados en pruebas funcionales preliminares frente a escenarios de suplantación visual 58
4.6. Resultados de la campaña de simulación del enlace LoRa P2P 60

Índice de figuras

2.1. Comparación conceptual entre un sistema de control vehicular convencional y un sistema ciberfísico residencial. 4
2.2. Ejemplo de dashboard de una plataforma de administración residencial. La interfaz prioriza indicadores de gestión comunitaria y operación administrativa, más que funciones nativas de control de acceso vehicular seguro en tiempo real. 6
2.3. Arquitectura propuesta para el sistema de control de acceso vehicular: componentes de borde, comunicaciones y backend en la nube. 12
2.4. Representación conceptual de amenazas relevantes en sistemas de acceso vehicular ciberfísicos. 13
2.5. Esquema genérico de un sistema ciberfísico orientado a monitoreo, procesamiento, decisión y actuación. 17
2.6. Composición interna general de una red neuronal convolucional, desde la extracción jerárquica de rasgos hasta la salida de clasificación/detección. Elaboración propia basada en [3, 4]. 18
2.7. Esquema conceptual de una arquitectura tipo YOLO para detección de objetos en tiempo real. 21
2.8. Flujo general de detección, recorte de región, reconocimiento de caracteres y validación posterior en un pipeline OCR. Elaboración propia. 22

3.1. Esquema conceptual de la plataforma segura propuesta para *ResiPark*, mostrando las cuatro capas del sistema y la ciberseguridad como propiedad transversal. . . . 28
3.2. Flujo de interacción entre los componentes físicos, de comunicación y de aplicación. 29
3.3. Superficie de ataque del sistema, mostrando activos críticos y puntos plausibles de entrada. 31
3.4. Interfaz del prototipo de escritorio (PyQt5) desarrollado para la validación de modelos y lógica de acceso. 36
3.5. Topología técnica de implementación de ResiPark, organizada por componentes de hardware, red y software. 38
3.6. Esquema de conexión del nodo de borde para el control de acceso. 40
3.7. Topología de red híbrida: LoRa para sensores subterráneos y Wi-Fi para el control de acceso en superficie. 41
3.8. Pantalla de inicio de sesión de la aplicación móvil. 44
3.9. Interfaces de gestión de visitas en la aplicación móvil. 44
3.10. Dashboard principal: Visualización en tiempo real de la ocupación de estacionamientos. 45
3.11. Módulo de historial de accesos con evidencia fotográfica de las patentes detectadas. 46
3.12. Interfaz de administración de residentes y vehículos autorizados. 47

4.1. Módulo de historial de accesos desarrollado en React, visualizando datos desde Supabase. 49
4.2. Interfaz de gestión de vehículos y residentes. 50
4.3. Curvas del entrenamiento final del detector de vehículos YOLOv8s. 51
4.4. Curvas del entrenamiento final del detector de patentes YOLO11s. 52
4.5. Prueba de presentación aislada de una patente. Al no detectarse un vehículo asociado, el flujo no continuó hacia OCR ni apertura. 56
4.6. Salida del cliente durante la verificación de una patente sin autorización vigente. La detección no derivó en una orden de apertura. 56
4.7. Prueba con una reproducción dibujada de patente instalada sobre un vehículo. . 57
4.8. Caso de una misma patente asociada a vehículos de apariencia distinta. 57
4.9. Montaje experimental del nodo de control en el acceso vehicular. 61

Glosario

Para la correcta comprensión de la presente memoria, se definen los siguientes términos técnicos utilizados en el desarrollo del sistema *ResiPark*:

API REST (Representational State Transfer): Arquitectura de software que impone condiciones sobre cómo debe funcionar una API. En este proyecto, se utiliza para la comunicación entre el cliente web/móvil y el servidor backend, gestionando recursos como usuarios, vehículos y reservas mediante métodos HTTP (GET, POST, PUT, DELETE).

Backend: Capa del sistema encargada de procesar la lógica de negocio, gestionar la base de datos y autenticar usuarios. En este proyecto, fue desarrollado utilizando el framework *FastAPI* en Python.

Dataset: Conjunto de datos estructurados utilizado para entrenar y validar modelos de inteligencia artificial. En el prototipo final se emplearon conjuntos publicados en Roboflow para detección de vehículos y patentes.

Endpoint: Punto final de comunicación en una API. Corresponde a una URL específica donde el servicio puede ser accedido por una aplicación cliente para realizar una operación (ej. `/auth/login` o `/reservas`).

JWT (JSON Web Token): Estándar abierto (RFC 7519) que define un método compacto y autónomo para transmitir información de forma segura entre partes como un objeto JSON. Se utiliza en el sistema para gestionar la autenticación y mantener la sesión de los usuarios de forma segura y sin estado (stateless).

LoRa (Long Range): Tecnología de modulación de radio para redes de baja potencia y largo alcance (LPWAN). Se seleccionó para la comunicación de los sensores de estacionamiento debido a su capacidad para atravesar estructuras de concreto en subterráneos donde la señal Wi-Fi es deficiente.

OCR (Optical Character Recognition): Tecnología que permite convertir diferentes tipos de documentos, como imágenes capturadas por una cámara digital, en datos editables y buscables. En el sistema, se utiliza el modelo *Florence-2* para extraer el texto alfanumérico de las patentes vehiculares.

ORM (Object-Relational Mapping): Técnica de programación para convertir datos entre sistemas de tipos incompatibles en lenguajes de programación orientados a objetos. Se utilizó *SQLAlchemy* para interactuar con la base de datos SQL desde código Python.

Shadow IT: Uso de sistemas, dispositivos, software, aplicaciones y servicios de TI sin la aprobación explícita del departamento de TI. Aunque el sistema busca mitigar riesgos físicos,

el diseño seguro del mismo busca evitar vulnerabilidades digitales asociadas a dispositivos IoT mal gestionados.

YOLO (You Only Look Once): Familia de modelos de detección de objetos en tiempo real basados en redes neuronales convolucionales (CNN). Se utilizó la versión YOLOv11/v8 para la detección y clasificación instantánea de vehículos en el acceso.

Capítulo 1

Introducción

1.1. Contexto

En la última década, el concepto de *Smart Cities* y comunidades inteligentes ha transformado la gestión urbana, integrando tecnologías avanzadas para mejorar la calidad de vida y la eficiencia operativa [5, 6, 7]. Sin embargo, en ciudades densamente pobladas como Santiago de Chile, la seguridad residencial sigue siendo un desafío crítico. Los edificios residenciales enfrentan problemas recurrentes relacionados con el control de acceso vehicular: ingresos no autorizados, dependencia del error humano en conserjerías y una gestión ineficiente de los recursos comunes, como los estacionamientos de visita [8, 9].

Actualmente, una parte importante de los controles de acceso depende de procesos manuales o de sistemas desconectados que no ofrecen trazabilidad ni validación en tiempo real. Esta situación genera brechas de seguridad física y dificultades operativas que afectan tanto a la administración como a los residentes [9, 10, 11].

1.2. Problemática

La validación manual de vehículos por parte del personal de conserjería presenta vulnerabilidades inherentes. La falta de herramientas tecnológicas adecuadas permite la suplantación de identidad, el ingreso de vehículos no registrados y el uso abusivo de estacionamientos de visita. Además, desde una perspectiva de ingeniería telemática, la implementación de soluciones basadas en Internet de las Cosas (IoT) en entornos subterráneos presenta desafíos de conectividad que las soluciones tradicionales basadas puramente en Wi-Fi no logran resolver eficazmente [6, 5, 12]. A esto se suma el riesgo de ciberseguridad: implementar dispositivos conectados y cámaras inteligentes introduce nuevos vectores de ataque que deben ser mitigados desde el diseño del software [11, 10]. En los modelos de aprendizaje profundo aparecen brechas adicionales: ejemplos adversarios que inducen clasificaciones erróneas, degradación ante cambios de dominio respecto de los datos de entrenamiento, sesgos asociados a la cobertura del conjunto de datos y limitada explicabilidad al auditar decisiones automáticas [13, 3, 14].

1.3. Objetivos del Proyecto

1.3.1. Objetivo General

Diseñar e implementar una plataforma segura para el control de accesos vehiculares y la gestión de estacionamientos en edificios residenciales (*ResiPark*), integrando mecanismos de autenticación, trazabilidad, validación de eventos físicos y digitales, y medidas de protección en backend, red e interfaces de operación.

1.3.2. Objetivos Específicos

Considerando la naturaleza multidisciplinaria del proyecto, esta memoria de título se enfoca específicamente en los siguientes hitos desarrollados por el autor:

1. **Identificar amenazas, vulnerabilidades y superficies de ataque** presentes en sistemas de control de acceso vehicular residenciales, considerando riesgos de suplantación, abuso de API, intercepción de comunicaciones y manipulación de dispositivos conectados.
2. **Diseñar e implementar mecanismos de seguridad en backend y capa de aplicación** para proteger la autenticación, autorización, trazabilidad y gestión de eventos del sistema, resguardando los datos sensibles y las decisiones de acceso.
3. **Desarrollar e integrar modelos de visión por computador orientados a una validación segura del acceso**, incorporando detección vehicular, localización de patentes y validación contextual por etapas como medida de mitigación parcial frente a intentos de suplantación.
4. **Evaluar la factibilidad técnica de la integración de los componentes de sensado, comunicación y operación** de ResiPark mediante pruebas funcionales en un entorno real, e identificar las brechas y requisitos de seguridad pendientes para un despliegue productivo.

1.4. Estructura de la Memoria

El presente documento se organiza de la siguiente manera: el **Capítulo 2** presenta el Estado del Arte y el Marco Teórico, abordando la evolución de los sistemas de acceso vehicular hacia enfoques ciberfísicos, así como los fundamentos de IoT, visión por computador y seguridad aplicados al problema. El **Capítulo 3** describe la metodología y el diseño de la plataforma segura, partiendo desde la arquitectura IoT/ciberfísica general y profundizando luego en el modelo de amenazas, los mecanismos de protección por capa y la integración de los componentes desarrollados. El **Capítulo 4** expone los resultados obtenidos, su validación experimental y una discusión crítica de las limitaciones del prototipo, organizada de acuerdo con los objetivos específicos. Finalmente, el **Capítulo 5** presenta las conclusiones y el trabajo futuro.

Capítulo 2

Estado del Arte y Marco Teórico

Este capítulo presenta una revisión del estado del arte en sistemas de control de acceso vehicular, gestión de estacionamientos y comunidades inteligentes, abarcando tanto soluciones industriales como aportes académicos recientes. Asimismo, se desarrolla el marco teórico que sustenta las decisiones técnicas adoptadas en el diseño e implementación del sistema *ResiPark*.

2.1. Estado del Arte

2.1.1. Control de acceso vehicular: contexto general y motivación tecnológica

El control de acceso vehicular ha sido implementado históricamente en múltiples contextos: recintos industriales, aeropuertos, campus universitarios, centros logísticos, hospitales, estacionamientos comerciales y, más recientemente, comunidades residenciales. En todos estos escenarios, el objetivo operativo es similar: decidir en tiempo acotado si un vehículo puede ingresar, registrar el evento de manera trazable y activar un mecanismo físico de barrera o portón.

La diferencia crítica entre contextos no está en la existencia del control, sino en su nivel de automatización, en el perfil de riesgo y en la calidad de los datos disponibles para tomar la decisión. En sistemas tradicionales, la validación descansa en personal humano, listas estáticas y observación visual directa. En sistemas modernos, la decisión se apoya en cámaras, sensores, servicios backend y modelos de inteligencia artificial capaces de identificar vehículos y patentes en tiempo real [15, 14, 16].

Desde el punto de vista de ingeniería, esta evolución responde a una necesidad concreta: reducir errores de operación, aumentar trazabilidad y disminuir tiempos de espera en puntos de acceso. No obstante, trasladar el proceso hacia arquitecturas digitalizadas amplía la superficie de ataque: cada sensor, API y canal de comunicación agrega un nuevo elemento que debe ser protegido [11]. Por lo tanto, el estado del arte no puede analizarse solo en términos de precisión de detección, sino también en términos de seguridad integral del sistema ciberfísico.

2.1.2. Del control vehicular convencional al enfoque ciberfísico residencial

Diversos autores en el área de *smart communities* y *smart buildings* describen una transición progresiva desde soluciones aisladas (domótica puntual) hacia ecosistemas conectados con capacidad de monitoreo y actuación coordinada [5, 6, 8]. Bajo este paradigma, el control de acceso deja de ser un proceso exclusivamente humano para convertirse en un flujo ciberfísico compuesto por captura de datos, procesamiento inteligente, decisión de autorización y actuación mecánica.



Figura 2.1: Comparación conceptual entre un sistema de control vehicular convencional y un sistema ciberfísico residencial.

Como se observa en la Figura 2.1, el tránsito desde un control convencional hacia un enfoque ciberfísico no solo implica agregar automatización, sino reorganizar el proceso de decisión alrededor de captura de evidencia, comunicación entre subsistemas y actuación controlada. La figura sintetiza este cambio: a la izquierda se representa un esquema centrado en conserjería y registro manual, mientras que a la derecha se muestra una arquitectura con sensores, backend, validación y trazabilidad integrada.

En términos funcionales, un sistema ciberfísico de acceso vehicular combina al menos cuatro bloques: (i) percepción del entorno físico (cámaras/sensores), (ii) comunicación confiable entre nodos de borde y backend, (iii) lógica de negocio para validar identidad y reglas de acceso, y (iv) actuación segura sobre infraestructura crítica (portones o barreras). Esta integración también ha sido reportada en soluciones de estacionamiento inteligente donde visión por computador e IoT convergen para habilitar monitoreo y control casi en tiempo real [17, 18, 12].

Sin embargo, una parte importante de las implementaciones publicadas, como varias propuestas de estacionamiento inteligente centradas en ocupación, transmisión de telemetría o precisión de detección visual [17, 18, 12, 16], prioriza el rendimiento técnico de un subsistema puntual sin discutir con suficiente profundidad la seguridad transversal entre capas. Esta fragmentación genera brechas prácticas: sistemas precisos en laboratorio pueden ser frágiles en operación si no existen controles de autenticación, segmentación de red, protección de API y validación cruzada de eventos físicos/digitales [11, 10, 19].

En el contexto residencial, estas brechas adquieren una relevancia particular, ya que la gestión de accesos se vincula con residentes, visitas, vehículos autorizados y la operación cotidiana del recinto. Por ello, la evaluación de estas soluciones debe considerar no solo continuidad operativa,

sino también trazabilidad, protección de datos y controles para prevenir autorizaciones indebidas.
.

En estacionamientos de centros comerciales, el objetivo principal de la automatización suele ser la continuidad de flujo y la validación de cobro, por lo que las arquitecturas tienden a priorizar ticketing, conteo y lectura de patente como soporte operacional. En cambio, en el contexto residencial el requisito dominante es la autorización de acceso y la protección física de personas y bienes, lo que obliga a incorporar validaciones semánticas, controles de identidad y trazabilidad de eventos [17, 20].

Tabla 2.1: Comparación funcional entre control vehicular convencional y enfoque ciberfísico residencial

Criterio	**Control convencional**	**Enfoque ciberfísico residencial**
Decisión de acceso	Basada principalmente en observación humana y listas manuales.	Basada en evidencia física, validación digital y reglas de negocio.
Trazabilidad	Limitada o dependiente de registros manuales.	Registro automático y auditable de eventos operativos.
Tiempo de respuesta	Variable según carga de trabajo del personal.	Potencialmente menor al apoyarse en sensado y procesamiento automatizado.
Superficie de ataque	Menor digitalización, pero alta dependencia humana.	Mayor exposición digital, mitigable mediante controles por capa.
Escalabilidad	Baja, fuertemente ligada a operación presencial.	Mayor capacidad de integración, monitoreo y automatización.

En la Tabla 2.1 se resume esta diferencia funcional. Los sistemas convencionales dependen en mayor medida de observación humana y registros manuales, mientras que los enfoques ciberfísicos trasladan parte relevante de la decisión a una infraestructura conectada capaz de registrar, validar y correlacionar evidencia en tiempo real. Esta transición mejora la trazabilidad y la escalabilidad, pero también exige diseñar controles de seguridad acordes con la mayor exposición digital.

2.1.3. Revisión de plataformas residenciales y brechas de integración seguridad-operación

En el mercado chileno existen plataformas ampliamente adoptadas para administración de comunidades, como **Edifito**[1] y **Comunidad Feliz**[2]. Estas herramientas han demostrado valor en digitalización administrativa: pagos, reservas de espacios comunes, comunicaciones internas y trazabilidad documental. Su aporte al ecosistema residencial es relevante, ya que consolidan información de gestión en una interfaz accesible para residentes y administración [21, 22].

[1] `https://edifito.com`
[2] `https://www.comunidadfeliz.cl`

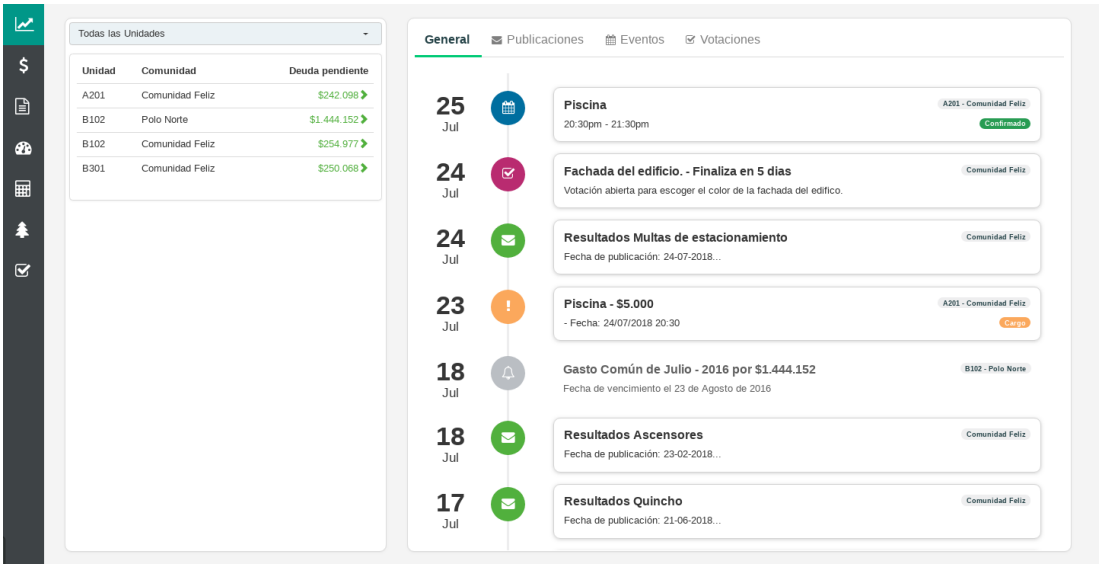


Figura 2.2: Ejemplo de dashboard de una plataforma de administración residencial. La interfaz prioriza indicadores de gestión comunitaria y operación administrativa, más que funciones nativas de control de acceso vehicular seguro en tiempo real.

En la Figura 2.2, tomada de la plataforma Comunidad Feliz [22], se observa un ejemplo de interfaz orientada principalmente a administración comunitaria. La figura resulta útil para distinguir entre una solución centrada en indicadores operativos y otra orientada a control de acceso vehicular seguro en tiempo real.

No obstante, al comparar su propuesta de valor con los requerimientos de un sistema de control de acceso vehicular inteligente, se observa una diferencia estructural: su núcleo funcional está orientado al proceso administrativo y no a la automatización ciberfísica de acceso en tiempo real. En la práctica, la decisión de apertura vehicular suele seguir dependiendo de verificación manual por conserjería o de subsistemas externos no integrados de forma nativa [21, 22, 9].

Esta separación entre “software de gestión” e “infraestructura física de acceso” limita tres capacidades que son críticas para esta memoria. Primero, la validación consistente entre evidencia física (vehículo detectado) y datos digitales (vehículo autorizado). Segundo, la trazabilidad unificada de eventos de seguridad para auditoría posterior. Tercero, la aplicación de controles de ciberseguridad extremo a extremo desde el dispositivo de borde hasta la Interfaz de Programación de Aplicaciones (API).

Tabla 2.2: Vulnerabilidades frecuentes en plataformas residenciales y subsistemas de acceso no integrados

Componente observado	**Vulnerabilidad o limitación**	**Impacto en control de acceso seguro**
Plataforma administrativa residencial	Foco principal en gestión documental, pagos y reservas, sin integración nativa con evidencia física del acceso.	La autorización vehicular sigue dependiendo de procesos manuales o de sistemas externos desacoplados.
Cámara con ALPR cerrado	Lectura de patente sin suficiente contexto semántico ni reglas de negocio personalizadas.	Mayor exposición a spoofing básico y menor capacidad de auditoría del proceso de autorización.
Conserjería manual	Dependencia de observación humana, carga cognitiva y registro manual de eventos.	Aumenta probabilidad de error operativo, omisiones y pérdida de trazabilidad.
Infraestructura no segmentada	Sensores, clientes y backend comparten dominios de confianza poco diferenciados.	Mayor riesgo de movimiento lateral, intercepción y propagación de fallos entre capas.

La Tabla 2.2 sintetiza estas limitaciones frecuentes. En ella puede observarse que la brecha principal no proviene de una única tecnología, sino de la falta de integración entre plataformas administrativas, cámaras con reconocimiento automático de patentes (ALPR), operación manual e infraestructura de red. En conjunto, estas debilidades dificultan implementar una autorización vehicular segura y auditable.

Adicionalmente, al revisar soluciones comerciales de automatización basadas en hardware cerrado (por ejemplo, cámaras IP con ALPR embebido), se observa una limitación relevante desde ciberseguridad aplicada: en arquitecturas orientadas solo a OCR, la lectura de texto puede activarse sin verificar previamente la presencia de un vehículo en la escena, aumentando la exposición ante ataques de presentación [14, 23, 24].

Estas limitaciones tienen impacto directo en contexto residencial. A diferencia de escenarios de cobro en estacionamientos comerciales, donde el error suele traducirse en incidencia operativa o económica, en un condominio la consecuencia puede comprometer seguridad física de residentes. Por ello, en esta memoria la comparación de plataformas no se hace solo por funcionalidades administrativas, sino por capacidad de articular seguridad física, red y aplicación en una cadena auditable [11, 23].

La revisión anterior muestra, por tanto, una oportunidad técnica clara: articular en una misma arquitectura la capa física (cámara/sensor/actuación), la capa lógica (backend/API) y la capa de operación (interfaces), incorporando autenticación, trazabilidad y validación coherente del evento de acceso.

Tabla 2.3: Amenazas relevantes y mitigaciones propuestas/analizadas para la arquitectura.

Ataque	**En qué consiste**	**Mitigación propuesta/analizada en ResiPark**	**Soporte en la literatura**
Spoofing de presentación	Mostrar una foto o pantalla con una patente válida para forzar apertura.	Detección vehicular previa, ROI y validación cruzada con backend.	[23, 25, 24].
Intercepción en red local	Capturar o modificar tráfico entre sensores y backend.	Segmentación de red recomendada (VLAN dedicada), LoRa para telemetría y política de bloqueo por defecto.	[11, 19, 26].
Abuso de API o sesión	Reutilizar sesiones o explotar endpoints sin autorización.	JWT *stateless*, validación de entradas y consultas parametrizadas.	[27, 28, 29].
Oclusión / baja visibilidad	Reducir la confiabilidad de visión en subterráneos.	Arquitectura propuesta con sensores IoT para plazas y uso de LoRa en niveles -1/-2.	[30, 20, 12].

La Tabla 2.3 resume amenazas plausibles para una arquitectura de acceso vehicular inteligente y las líneas de mitigación más reportadas en la literatura. Su propósito no es afirmar que cada control elimina completamente el riesgo, sino mostrar que la reducción de exposición depende de combinar validaciones físicas, controles de red y protección de la lógica de aplicación.

2.1.4. Control de acceso y seguridad en edificios inteligentes

La incorporación de IoT y servicios API en infraestructuras residenciales transforma un problema físico local en un sistema distribuido con exposición digital. De acuerdo con [11], la heterogeneidad de dispositivos, la diversidad de protocolos y las restricciones computacionales de nodos de borde hacen que los mecanismos de seguridad clásicos no sean directamente transferibles a entornos IoT.

En esta línea, [9] evidencian que la percepción de seguridad es el factor dominante para la adopción de sistemas inteligentes de acceso, mientras que [10] señalan que la autenticación y autorización deben diseñarse para escalabilidad y dinamismo operativo, evitando dependencias rígidas que dificulten la administración de múltiples dispositivos y usuarios.

Una implicancia directa para sistemas de acceso vehicular es que la seguridad debe modelarse de forma transversal por capas. En la capa física, se requiere asegurar la integridad de la captura y de la actuación; en la capa de comunicación, proteger la confidencialidad e integridad del tránsito de datos; en la capa de aplicación, reforzar autenticación, autorización y validación de entradas. Esta descomposición por capas será utilizada en los apartados siguientes para comparar tecnologías y justificar decisiones del sistema propuesto.

Siguiendo dicha lógica, diversas arquitecturas recientes implementan autenticación *stateless* con JWT en backend y validación de reglas de negocio orientadas a reducir suplantación y accesos indebidos [27, 28, 29]. La justificación de este enfoque no se limita a conveniencia de implementación, sino a un criterio de seguridad operativa: desacoplar identidad, trazabilidad y

autorización en componentes auditables y escalables.

Tabla 2.4: Resumen de seguridad transversal por capas en la arquitectura analizada.

Capa	**Riesgo principal**	**Control aplicado**	**Aporte de *ResiPark***
Física	Suplantación visual, manipulación del acceso o captura no autorizada.	Detección vehicular previa al OCR y verificación de ROI.	Evita que un texto aislado active la apertura del portón.
Red	Intercepción, spoofing local y pérdida de disponibilidad en subterráneos.	Segmentación de red recomendada, uso de LoRa para telemetría y restricciones de tráfico.	Propone separar el canal de sensores del flujo de video y de la aplicación.
Backend	Inyección de entradas, abuso de endpoints y autorización débil.	FastAPI, consultas parametrizadas y JWT *stateless*.	Centraliza la validación y deja trazabilidad auditable de cada acceso.
Aplicación	Exposición de eventos sin contexto o errores en la gestión operativa.	Interfaz web con polling controlado y reglas de negocio asociadas a la operación.	Vincula la operación diaria con evidencias verificables del sistema.

2.1.5. Sistemas ciberfísicos (CPS) y arquitectura por capas

La literatura reciente en hogares inteligentes y sistemas IoT converge en una idea clave: para pasar de monitoreo pasivo a control confiable del entorno físico, la arquitectura debe modelarse explícitamente como un sistema ciberfísico (CPS), donde percepción, comunicación, analítica y actuación cierran un ciclo de retroalimentación en tiempo real [1, 2, 31]. En este marco, IoT aporta la conectividad, sensado distribuido y transporte de datos, mientras que el enfoque CPS incorpora además la coordinación entre monitoreo, decisión y acción sobre el entorno físico; en otras palabras, no todo sistema IoT constituye por sí mismo un CPS, pero los CPS contemporáneos suelen apoyarse fuertemente en infraestructura IoT para cerrar ese ciclo de control [1, 6].

Aunque existen variaciones, los modelos más citados en implementaciones reales se organizan en tres a cinco capas. En arquitecturas de tres capas predomina la abstracción *percepción-red-aplicación*; en cuatro capas se separa la comunicación local de la comunicación remota; y en cinco capas se desacopla con mayor granularidad la adquisición física de la lógica analítica y de visualización [1]. Esta evolución no es solo formal: permite delimitar responsabilidades de seguridad, disponibilidad y latencia para cada componente.

En implementaciones reportadas para acceso inteligente, esta discusión orienta la preferencia por arquitecturas donde la percepción en borde, la red, la lógica de negocio y la interfaz de operación se modelan como bloques acoplados funcionalmente, pero desacoplados en responsabilidades técnicas. Este enfoque facilita auditoría de incidentes, resiliencia frente a fallos y control más preciso de superficies de ataque, aunque también incrementa la necesidad de coordinación entre capas y de políticas de seguridad homogéneas [2, 31, 19].

Desde seguridad, varias revisiones recomiendan tratar los controles como propiedad transversal y no como módulo final de endurecimiento [11, 2, 32]. Esto implica que controles como autenticación de dispositivos, segmentación de red, cifrado de transporte y validación contextual de eventos deben definirse por capa y verificarse en conjunto durante la operación.

En esa misma línea, documentos orientados a smart buildings y gestión de amenazas enfatizan que un compromiso lógico en la red puede convertirse rápidamente en una afectación física si no existe segmentación, monitoreo y control de efectividad de medidas [19, 26].

Tabla 2.5: Comparación de arquitecturas por capas en sistemas IoT/CPS [1, 2].

Arquitectura	**Capas típicas**	**Ventaja principal**	**Riesgo si no hay seguridad transversal**
3 capas	Percepción, Red, Aplicación.	Simplicidad y menor sobrecarga de integración.	Controles demasiado agregados y baja trazabilidad por dominio.
4 capas	Percepción, Red local, Red remota, Aplicación.	Mejor separación de enlaces y dominios de comunicación.	Brechas entre borde y nube si no se validan identidades extremo a extremo.
5 capas	Capa física, percepción, comunicación, analítica, visualización.	Máximo desacople funcional y gobernanza por componente.	Mayor complejidad operativa si no existen políticas homogéneas.

La Tabla 2.5 permite comparar por qué las arquitecturas de cuatro y cinco capas suelen ser preferidas en aplicaciones donde la seguridad debe analizarse de forma diferenciada entre percepción, comunicación y aplicación. A medida que aumenta la granularidad arquitectónica, mejora la trazabilidad de responsabilidades, pero también se vuelve más exigente la integración de controles coherentes entre capas.

2.1.6. Reconocimiento Automático de Patentes y Detección Vehicular

Los sistemas de reconocimiento automático de patentes (ALPR) han evolucionado desde pipelines clásicos de procesamiento de imagen (segmentación por bordes, umbralización y reglas geométricas) hacia arquitecturas de aprendizaje profundo integradas. Los enfoques clásicos mantienen utilidad en escenarios controlados, pero su desempeño cae cuando existen variaciones de iluminación, ruido en sensores, inclinación de placa, suciedad oclusiva o ángulos no frontales [33].

La literatura reciente converge en dos observaciones. Primero, los detectores de una etapa como YOLO permiten resolver localización y clasificación de objetos en una sola inferencia, reduciendo latencia y simplificando el despliegue en tiempo real [34, 14]. Segundo, en aplicaciones de control de acceso, la métrica de precisión debe analizarse junto con estabilidad operacional (falsos rechazos y falsos aceptos), ya que ambos tipos de error impactan directamente en seguridad y experiencia de usuario [15, 16].

Trabajos recientes refuerzan que la familia YOLO sigue siendo competitiva para escenarios complejos de ALPR, especialmente cuando se combina con módulos específicos para lectura de placas y optimización por contexto. En particular, métodos basados en CSCM-YOLOv8 y CSM-LPRNet muestran que la descomposición del problema sigue siendo efectiva en ambientes complejos, mientras que estudios sobre YOLOv11 destacan ventajas en la detección de objetos pequeños y en la robustez del detector para escenarios de acceso [24, 25].

En términos de diseño de sistema, el problema ALPR en acceso residencial no es equivalente a 'detectar texto en una imagen'. En la práctica, el flujo incluye una secuencia acoplada de tareas: detectar vehículo, validar que su posición corresponde a una zona de acceso real, detectar placa, reconocer caracteres y correlacionar resultado con reglas de negocio. Esta secuencia explica por qué múltiples trabajos reportan mejores resultados con arquitecturas de etapas especializadas

que con una única red monolítica para todo el flujo [14, 35].

Sobre esa base, la literatura muestra que los sistemas con dos modelos diferenciados (por ejemplo, uno para detección vehicular y otro para lectura de patente) ofrecen ventajas concretas frente a aproximaciones monolíticas. Entre ellas destacan la separación de errores entre etapas, la posibilidad de optimizar cada modelo con datos y umbrales específicos y la habilitación de validaciones cruzadas entre contexto vehicular y texto reconocido [14, 35, 23]. Estas propiedades resultan especialmente pertinentes en aplicaciones de acceso, donde la robustez operativa y la trazabilidad del fallo son tan relevantes como la precisión puntual. .

Investigaciones como [36] muestran el trade-off entre eficiencia computacional y exactitud en hardware restringido. Este punto es crítico para nodos de borde en acceso vehicular: minimizar cómputo no puede comprometer la tasa de decisión correcta en un sistema que controla un activo físico. Por eso, la estrategia seguida prioriza robustez de inferencia y consistencia de decisión por sobre la reducción extrema de recursos. Esta decisión es coherente con el paradigma de *edge computing*, donde la ejecución local mejora resiliencia y reduce dependencia de nube en aplicaciones sensibles a latencia [37, 38].

Tabla 2.6: Comparación de enfoques para reconocimiento de patentes y detección vehicular.

Enfoque	**Ventajas reportadas**	**Limitaciones reportadas**	**Implicancia para ResiPark**
ALPR clásico (reglas + morfología)	Bajo costo computacional inicial; implementación simple en escenarios controlados [33].	Alta sensibilidad a iluminación, ángulo y ruido; menor robustez en operación real [33].	No suficiente para acceso residencial 24/7 con variabilidad ambiental.
Detector de una etapa (YOLO)	Inferencia rápida; buena relación precisión/latencia para tiempo real [34, 14].	Requiere dataset bien curado y ajuste fino por dominio [15].	Base adecuada para detección vehicular y localización de placa en borde.
Pipeline multietapa (vehículo + placa + OCR)	Mayor control por etapa; facilita validaciones cruzadas de seguridad [14, 16].	Más componentes a integrar y monitorear.	Se adopta por robustez operacional y trazabilidad de errores.
Optimización extrema para hardware mínimo	Reduce consumo y costo en borde [36].	Puede degradar precisión en contextos críticos [36].	Se privilegia precisión y estabilidad sobre ahorro computacional extremo.

La Figura 2.3 resume una arquitectura de referencia para control de acceso vehicular inteligente, donde se distinguen la percepción en borde, la red de comunicaciones, el backend encargado de la lógica de negocio y las interfaces de operación. Su incorporación en este apartado permite anticipar, de forma conceptual, cómo convergen los subsistemas revisados en el estado del arte antes de abordar la propuesta particular en el capítulo siguiente.

2.1.7. Amenazas de suplantación en ALPR y estrategias de mitigación

Un riesgo recurrente en sistemas ALPR de acceso es el *presentation spoofing*: el atacante presenta una imagen o pantalla con una patente autorizada para inducir una apertura indebida. En arquitecturas que dependen de OCR aislado, la falta de contexto visual y semántico incrementa la probabilidad de aceptación errónea.

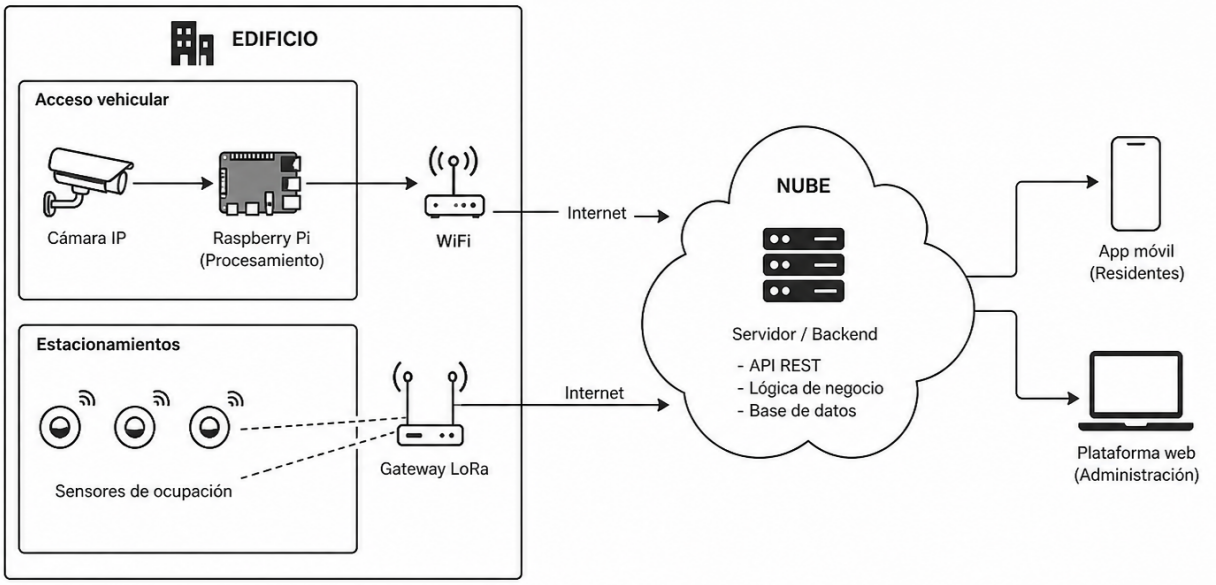


Figura 2.3: Arquitectura propuesta para el sistema de control de acceso vehicular: componentes de borde, comunicaciones y backend en la nube.

La Figura 2.4 resume la relación entre suplantación visual, abuso de interfaces de programación, manipulación de comunicaciones y compromiso de nodos conectados. Estas amenazas no son independientes: una debilidad en una capa puede amplificar el impacto de otra si no existen controles de defensa en profundidad.

Los enfoques recientes de mitigación combinan múltiples señales de validación (detección de vehículo, reconocimiento de patente, atributos adicionales y verificación cruzada en backend), reduciendo el impacto de ataques basados en evidencia bidimensional [23, 25]. Este principio es coherente con la lógica de defensa en profundidad para sistemas ciberfísicos: evitar que una sola evidencia determine una acción física irreversible.

En términos generales, esta línea de mitigación aporta tres ventajas relevantes: (i) reduce la dependencia de una única evidencia textual o visual, (ii) permite introducir verificaciones semánticas adicionales antes de una acción física crítica y (iii) mejora la trazabilidad del proceso de decisión para auditoría posterior. Aunque no elimina completamente el riesgo, sí dificulta ataques básicos de suplantación y fortalece la defensa en profundidad del sistema [23, 25].

2.1.8. Arquitectura de detección desacoplada: justificación técnica y de seguridad

Diversos trabajos recientes en ALPR y visión aplicada a control vehicular evitan los modelos monolíticos y dividen el flujo en etapas desacopladas, típicamente detección/clasificación vehicular y extracción de caracteres [14, 25, 35]. Esta decisión responde tanto a razones de rendimiento como de seguridad. En términos computacionales, la separación permite ejecutar filtros tempranos y reservar el costo mayor del OCR para casos con alta probabilidad de validez. En términos de seguridad, evita que una evidencia textual aislada dispare por sí sola una acción física.

En la primera etapa, el detector vehicular actúa como anillo de verificación física. En lugar de buscar texto en toda la escena, el sistema confirma presencia y clase de un objeto vehicular antes de continuar. En la segunda etapa, el OCR opera sobre una región recortada y contextualizada, reduciendo ruido y falsos positivos. En algunos trabajos, esta lógica se refuerza mediante zonas

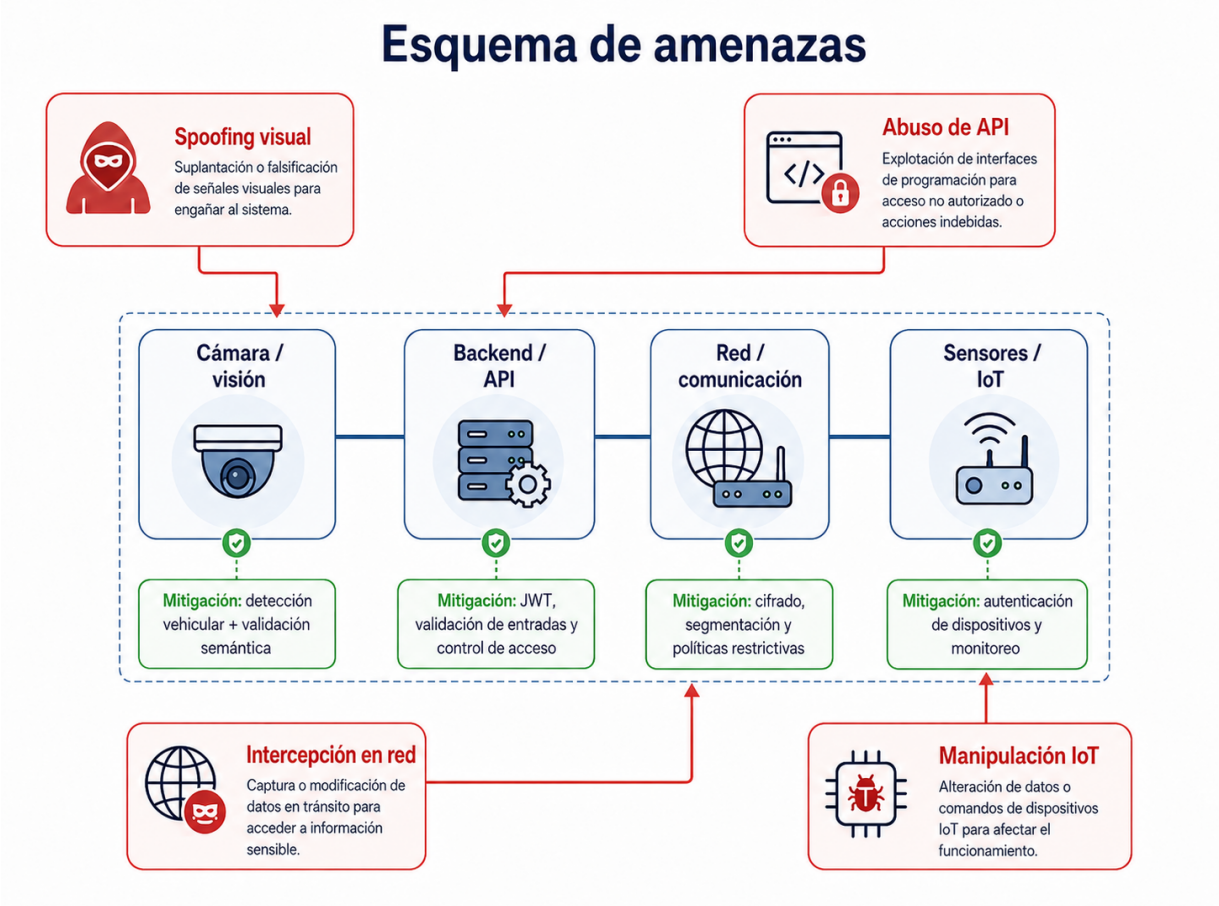


Figura 2.4: Representación conceptual de amenazas relevantes en sistemas de acceso vehicular ciberfísicos.

de interés (ROI), activando el OCR solo cuando la detección intersecta la zona operativa del acceso [14, 24].

La literatura reciente sobre ALPR robusto respalda este enfoque por etapas para equilibrar precisión y latencia en escenarios reales [14, 25]. Complementariamente, enfoques anti-spoofing multimodales reportan que combinar más de una señal de validación reduce la exposición frente a ataques de presentación [23]. Como contrapartida, estas arquitecturas exigen una integración más cuidadosa, monitoreo por etapa y criterios explícitos de consistencia entre módulos para evitar que el aumento de complejidad introduzca nuevas fallas operativas.

2.1.9. Gestión de Estacionamientos: Visión por Computador versus Sensores IoT

La detección de ocupación de estacionamientos ha sido abordada principalmente mediante dos enfoques: sistemas basados en visión por computador y sistemas basados en sensores dedicados.

Según la revisión sistemática de [17], los sistemas basados en cámaras presentan ventajas económicas en espacios abiertos, pero sufren limitaciones significativas en entornos complejos debido a oclusiones y condiciones de iluminación adversas. Estas limitaciones se acentúan en estacionamientos subterráneos, caracterizados por pilares de concreto y luz artificial tenue.

En contraste, los sistemas basados en sensores IoT ofrecen una detección más robusta y

modular cuando el entorno físico introduce oclusiones persistentes o variabilidad lumínica alta. Según [5], las arquitecturas IoT distribuidas permiten desacoplar adquisición y procesamiento, aumentando tolerancia a fallos y adaptabilidad ante cambios de topología. Esta característica es particularmente útil en subterráneos residenciales, donde la disposición de pilares, rampas y ángulos de estacionamiento cambia significativamente entre edificios.

Estudios aplicados que combinan IoT y estacionamientos inteligentes reportan mejoras operativas en monitoreo de disponibilidad y gestión de flujo vehicular [18, 12]. Sin embargo, también muestran que la selección tecnológica no es universal: en espacios abiertos y bien iluminados, visión por computador puede ser suficiente; en espacios cerrados con múltiples obstrucciones, sensores por plaza logran mayor confiabilidad de ocupación [17].

Desde la perspectiva de seguridad, esta decisión también tiene efectos de diseño. Un sistema basado exclusivamente en cámara concentra riesgo en un único punto de percepción; un sistema distribuido por sensores introduce más nodos, pero permite redundancia y validación cruzada de estados. En consecuencia, separar los dominios funcionales entre ocupación y visión para acceso reduce el acoplamiento, facilita la aplicación de controles por capa y permite ajustar cada subsistema a sus propias restricciones operativas.

Tabla 2.7: Comparación de enfoques para detección de ocupación de estacionamientos.

Criterio	**Visión por computador**	**Sensores IoT por plaza**	**Decisión en Resi-Park**
Robustez ante oclusiones	Sensible a bloqueos visuales [17].	Alta robustez al medir presencia local.	Se privilegia sensores en subterráneo.
Dependencia de iluminación	Alta en escenarios no controlados [17].	Baja (depende de tipo de sensor).	Se evita visión para ocupación en -1/-2.
Escalabilidad física	Menor hardware por zona amplia.	Requiere dispositivo por plaza.	Se acepta costo para ganar confiabilidad.
Integración con IoT distribuido	Requiere pipeline de video y cómputo.	Integración natural con nodos IoT [5].	Arquitectura modular por telemetría.
Seguridad operativa	Riesgo de punto único de percepción.	Mayor superficie de nodos, pero más redundancia.	Seguridad por capas y segmentación de red.

La Tabla 2.7 permite comparar ambos enfoques para la detección de ocupación. Se aprecia que la visión por computador ofrece ventajas de cobertura amplia con menor hardware por zona, mientras que los sensores IoT por plaza proporcionan mayor robustez ante oclusiones y variabilidad lumínica, aspecto especialmente relevante en subterráneos residenciales.

2.1.10. Análisis comparativo de trabajos relacionados (2021–2026)

Para consolidar el posicionamiento del aporte, la Tabla 2.8 compara enfoques recientes de ALPR y control de acceso con la arquitectura propuesta en esta memoria. El objetivo no es solo contrastar métricas de detección, sino evidenciar diferencias en validación de seguridad, integración IoT y madurez de despliegue.

Para fortalecer la trazabilidad académica del estado del arte, se adoptó un criterio de selección que prioriza publicaciones recientes (2021–2026) en temas de visión por computador aplicada, IoT/CPS y seguridad de acceso, complementadas con trabajos fundacionales cuando resultan necesarios para definir conceptos base (por ejemplo, IoT, edge computing y OAuth 2.0) [5, 6, 37, 38, 39].

Tabla 2.8: Comparativa de trabajos relacionados en control de acceso inteligente y ALPR (2021–2026).

Referencia	**Enfoque de detección/OCR**	**Seguridad y anti-spoofing**	**Integración IoT/arquitectura**	**Brecha frente al aporte de esta memoria**
[16]	ALPR con enfoque profundo orientado a reconocimiento de placa.	No reporta validación contextual entre presencia vehicular y autorización en capa de negocio.	Enfoque de software; integración ciberfísica no central.	Menor articulación con backend seguro y actuación física auditada.
[36]	Priorización de eficiencia para recursos limitados.	Seguridad no es eje principal del diseño reportado.	Edge básico sin capa de gestión comunitaria completa.	Optimiza cómputo, pero no aborda seguridad transversal de extremo a extremo.
[35]	YOLOv8 para ocupación/estacionamiento visual.	Mitigación de falsos positivos por umbral; anti-spoofing limitado.	Dependencia de visión artificial para ocupación.	En subterráneos, la oclusión favorece arquitectura con sensores dedicados.
[25]	Pipeline desacoplado con YOLO y estrategia temporal para ALPR.	Buen desempeño algorítmico, sin foco principal en identidad de usuarios.	Sistema orientado a procesamiento visual.	No integra de forma explícita autenticación API y reglas de acceso comunitario.
Esta memoria (2026)	Arquitectura desacoplada: detección vehicular + OCR contextual + ROI.	Validación contextual por etapas y autenticación *stateless*; segmentación de red definida como medida recomendada de despliegue.	Diseño CPS con edge y backend API; segmentación de red tratada a nivel de análisis y propuesta.	Integra desempeño y análisis de seguridad en un flujo operativo de acceso residencial; la validación experimental fue mayoritariamente diurna.

La revisión diferencia tres tipos de evidencia: (i) artículos científicos y conferencias con resultados experimentales, utilizados para fundamentar decisiones técnicas de arquitectura y desempeño; (ii) revisiones o papers de síntesis para identificar brechas y tendencias; y (iii) documentación técnica/industrial para precisar diferencias de implementación (como LoRa versus LoRaWAN) y consideraciones prácticas de despliegue [1, 2, 12, 25, 40, 41].

La Tabla 2.9 resume esta clasificación y su rol en el capítulo. Esta estrategia permite equilibrar rigor académico con aplicabilidad de ingeniería, evitando depender exclusivamente de fuentes comerciales y, al mismo tiempo, sin ignorar documentación técnica útil para la implementación real.

2.1.11. Síntesis de brechas y posicionamiento del aporte

La revisión anterior permite identificar una brecha recurrente: numerosos trabajos optimizan un subsistema particular, como detección visual, OCR o conectividad IoT, pero reportan con

Tabla 2.9: Tipología de fuentes y rol metodológico en el estado del arte.

Tipo de fuente	**Ventana temporal priorizada**	**Nivel de evidencia**	**Uso en esta memoria**
Artículo científico (journal/conference)	2021–2026	Alto	Comparar arquitecturas, métricas y brechas; justificar decisiones de diseño.
Revisión/survey técnico-científica	2020–2026	Alto/Medio	Estructurar el estado del arte e identificar tendencias y vacíos.
Estándar/RFC	Sin restricción temporal	Alto (normativo)	Sustentar mecanismos de autenticación/autorización (OAuth/JWT).
Documentación técnica industrial	2024–2026	Medio (implementación)	Aterrizar diferencias de protocolo y consideraciones de despliegue real.

menor detalle la integración de seguridad extremo a extremo entre percepción física, comunicación y lógica de autorización [11, 10]. En aplicaciones de acceso residencial, esta brecha es especialmente crítica porque una autorización errónea puede comprometer tanto la operación como la protección de personas, bienes y rutinas asociadas al recinto.

El aporte planteado en esta memoria se posiciona, por tanto, en la integración de mecanismos de percepción, comunicación, validación y trazabilidad, en lugar de basarse en una única métrica aislada. La evidencia revisada muestra que la automatización aporta valor cuando se articula con controles de ciberseguridad y reglas de autorización coherentes entre capas. En consecuencia, para un condominio una falla de autorización no debe entenderse solo como una pérdida operativa, sino como un potencial incidente de seguridad; este criterio orienta las decisiones de diseño, implementación y evaluación desarrolladas en los capítulos siguientes.

2.2. Marco Teórico

El presente marco teórico establece los fundamentos conceptuales y tecnológicos que sustentan el diseño e implementación del sistema *ResiPark*. Dado que la solución corresponde a una plataforma ciberfísica, se aborda primero el rol de IoT, comunicación y seguridad por capas, para luego profundizar en visión por computador, aprendizaje profundo y reconocimiento óptico de caracteres. Este orden permite avanzar desde la plataforma general hacia los subsistemas específicos que la componen.

2.2.1. Sistemas Ciberfísicos, IoT y Comunicación

Los sistemas ciberfísicos integran percepción del entorno, procesamiento de información, comunicación entre componentes y capacidad de modificar el ambiente físico mediante actuación. En control de acceso vehicular, esto implica que la decisión de autorizar un ingreso no surge de un único software aislado, sino de la coordinación entre sensores, cámaras, servicios backend, red de comunicaciones e interfaces de operación.

Dentro de esta lógica, el Internet de las Cosas (IoT) aporta la infraestructura de sensado y

conectividad necesaria para capturar estados del entorno y transmitirlos a otros componentes del sistema. La literatura reciente muestra que, en aplicaciones residenciales, el valor del IoT no se limita al monitoreo pasivo: su integración con analítica y reglas de negocio habilita plataformas capaces de observar, decidir y actuar sobre infraestructura crítica [6, 5, 2].

Desde el punto de vista de diseño, esta integración exige considerar simultáneamente cobertura inalámbrica, latencia aceptable, restricciones energéticas y controles de seguridad. Por esta razón, en esta memoria las tecnologías IoT no se estudian solo como medios de transmisión, sino como parte estructural del sistema ciberfísico y de su superficie de ataque.

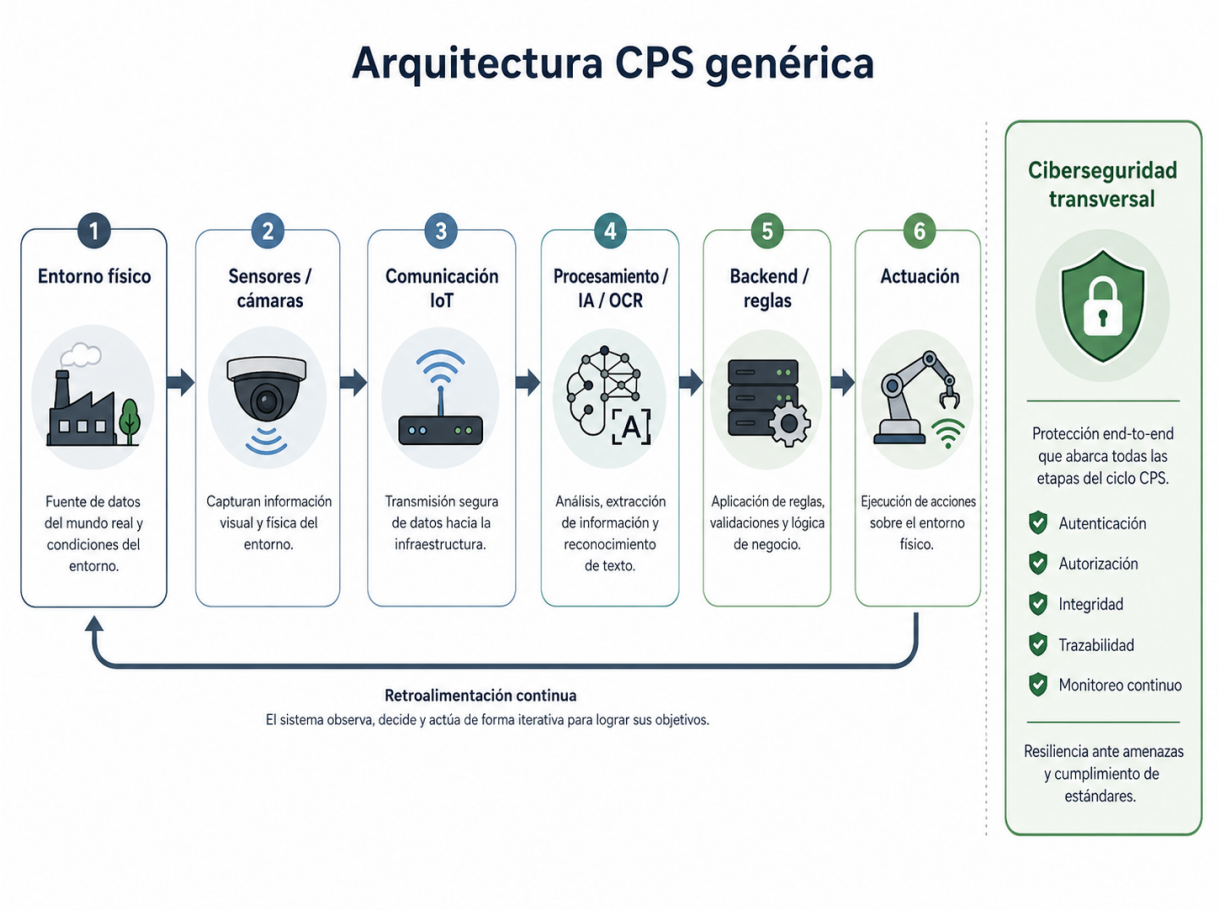


Figura 2.5: Esquema genérico de un sistema ciberfísico orientado a monitoreo, procesamiento, decisión y actuación.

Como se observa en la Figura 2.5, un sistema ciberfísico no se limita a recolectar datos: articula sensado, comunicación, análisis y actuación en un ciclo cerrado. Esta relación entre monitoreo y respuesta resulta especialmente útil para contextualizar por qué, en aplicaciones de acceso vehicular, la seguridad debe analizarse sobre toda la cadena y no solo sobre un módulo aislado.

2.2.2. Visión por Computador y Aprendizaje Profundo

La visión por computador es una disciplina de la Inteligencia Artificial que busca dotar a los sistemas computacionales de la capacidad de interpretar y comprender información visual proveniente del entorno [3, 4]. En aplicaciones de control de acceso vehicular, esta tecnología

permite automatizar procesos tradicionalmente manuales, como la identificación de vehículos y la validación de patentes.

En sistemas de acceso, el objetivo no se limita a detectar objetos con alta precisión media, sino a sostener una tasa de decisión estable bajo variabilidad operacional: cambios de iluminación, oclusiones parciales, perspectivas oblicuas y ruido en captura de video. Por esta razón, la literatura reciente enfatiza evaluar la robustez en condiciones reales y no solo en *benchmarks* controlados [14, 16, 25].

Por tanto, la literatura suele separar la detección vehicular y la lectura de patente en etapas dedicadas, con el fin de aislar errores, reducir el área de búsqueda del OCR y facilitar la trazabilidad de fallos en el pipeline [14, 25, 35]. Esta separación permite aplicar controles de seguridad específicos en cada etapa y evita que una evidencia textual aislada determine por sí sola una acción física crítica.

Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales constituyen la base de los sistemas modernos de reconocimiento visual. A diferencia de las redes neuronales densas, las CNN emplean capas convolucionales capaces de extraer características espaciales jerárquicas, tales como bordes, texturas y formas complejas, reduciendo el número de parámetros entrenables y mejorando la eficiencia computacional [3, 4].

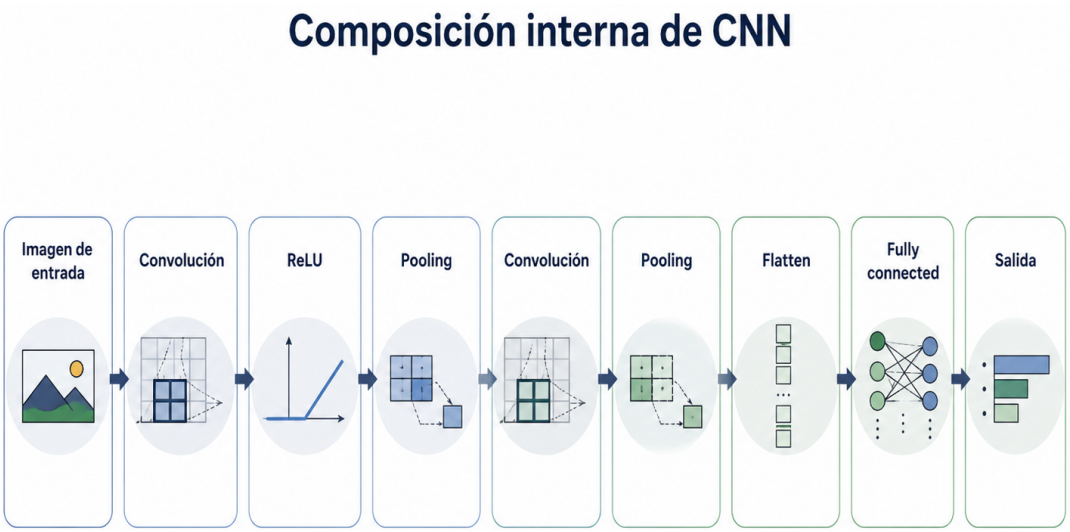


Figura 2.6: Composición interna general de una red neuronal convolucional, desde la extracción jerárquica de rasgos hasta la salida de clasificación/detección. Elaboración propia basada en [3, 4].

La Figura 2.6 permite visualizar este principio de operación: las primeras capas extraen patrones simples, mientras que las capas más profundas integran rasgos de mayor complejidad semántica. Esta propiedad explica por qué las CNN se consolidaron como base de los detectores modernos utilizados en visión por computador.

Esta capacidad las hace especialmente adecuadas para tareas de detección de objetos en imágenes capturadas en entornos no controlados, como accesos vehiculares con variaciones de iluminación y perspectiva.

Adicionalmente, el uso de CNN en inferencia de borde permite reducir el tiempo entre per-

cepción y acción, propiedad crítica en control de acceso físico. El beneficio operativo no es solo velocidad, sino resiliencia: si la conectividad remota se degrada, una parte del procesamiento puede mantenerse localmente, preservando continuidad parcial del servicio [37, 38, 2].

Evolución de enfoques de visión por computador

La detección visual aplicada a control de acceso ha evolucionado desde técnicas convencionales basadas en reglas geométricas y segmentación manual hacia arquitecturas profundas capaces de aprender representaciones robustas directamente desde los datos. En enfoques previos al aprendizaje profundo, la detección dependía fuertemente de umbrales, bordes, morfología matemática y supuestos relativamente rígidos sobre la geometría de los objetos observados. Si bien estas estrategias podían funcionar en entornos controlados, su desempeño se degradaba frente a cambios de iluminación, ángulos de captura, oclusiones y ruido [33, 14, 4].

La irrupción de arquitecturas profundas cambió esta situación al permitir una extracción jerárquica de características, reduciendo la necesidad de diseñar manualmente cada descriptor. Dentro de esta evolución, distintas familias de modelos ofrecieron compromisos diferentes entre precisión, costo computacional y adecuación a tiempo real.

Tabla 2.10: Evolución de enfoques de visión por computador relevantes para detección vehicular

Enfoque	**Rasgo principal**	**Ventaja**	**Limitación para este caso**
Técnicas clásicas	Segmentación, bordes, umbralización y reglas geométricas.	Bajo costo computacional inicial y fácil interpretación.	Alta sensibilidad a iluminación, perspectiva y ruido en operación real.
CNN de clasificación	Aprendizaje automático de características visuales.	Mejor generalización que descriptores manuales.	No resuelven por sí solas localización precisa en escenas completas.
Detectores de dos etapas (R-CNN y variantes)	Separan propuesta de regiones y clasificación.	Alta precisión en múltiples benchmarks.	Mayor costo computacional y menor adecuación a tiempo real en borde.
Redes residuales (ResNet y afines)	Profundidad elevada con conexiones residuales.	Mejor extracción de rasgos complejos y base para modelos posteriores.	Requieren integrarse con detectores más amplios para tareas completas de acceso.
Detectores de una etapa (YOLO)	Localización y clasificación en una sola inferencia.	Relación favorable entre velocidad, precisión y despliegue operativo.	Exigen curación de datos y ajuste fino por dominio de aplicación.

La Tabla 2.10 resume esta evolución y permite justificar por qué los detectores de una etapa, especialmente la familia YOLO, resultan particularmente atractivos para escenarios de acceso vehicular en tiempo real. Frente a alternativas clásicas o detectores de dos etapas, ofrecen un equilibrio más favorable entre latencia operativa y precisión utilizable.

2.2.3. Arquitectura YOLO (You Only Look Once)

La detección de objetos en tiempo real requiere arquitecturas capaces de procesar imágenes completas con baja latencia. En este contexto, la familia de modelos YOLO se clasifica como detectores de una sola etapa (*single-stage detectors*), ya que realizan simultáneamente la localización y clasificación de objetos en una única pasada por la red neuronal.

A diferencia de enfoques de dos etapas, como R-CNN, YOLO divide la imagen de entrada en una grilla y predice directamente las cajas delimitadoras (*bounding boxes*) junto con las probabilidades de clase. Esta estrategia permite alcanzar velocidades de inferencia compatibles con aplicaciones en tiempo real, evitando congestiones en puntos críticos como portones de acceso vehicular.

La evidencia reciente sobre ALPR basado en YOLO indica que su ventaja principal no es exclusivamente la latencia, sino la relación entre precisión utilizable y costo de despliegue cuando se integra con pipelines de posprocesamiento y OCR [14, 25]. En términos prácticos, esto habilita arquitecturas híbridas donde un detector rápido filtra regiones de interés y un segundo módulo especializado refina la decisión final.

La Figura 2.7 representa de manera conceptual la estructura de un detector tipo YOLO, mostrando la secuencia de extracción de características y detección multiescala. Su inclusión permite visualizar por qué este tipo de arquitectura resulta adecuado cuando se requieren decisiones rápidas y consistentes sobre escenas completas, como ocurre en el acceso vehicular.

Selección de la Arquitectura YOLO

Dentro de la familia YOLO, las variantes más compactas favorecen velocidad y menor consumo computacional, mientras que las versiones medianas o grandes suelen ofrecer mayor capacidad para detectar objetos pequeños o escenas visualmente complejas. En aplicaciones de acceso vehicular seguro, esta diferencia es relevante porque la pérdida de sensibilidad sobre motocicletas, placas parcialmente visibles o vehículos en condiciones adversas puede traducirse en falsos rechazos o validaciones incompletas.

No existe una variante YOLO universalmente óptima: los modelos compactos son preferibles cuando el hardware y la latencia disponible son la restricción dominante, mientras que las variantes grandes pueden mejorar la sensibilidad frente a objetos pequeños o escenas complejas a cambio de mayor costo computacional. Para sistemas de acceso vehicular, una variante intermedia suele constituir un punto de partida razonable cuando se busca equilibrar capacidad de detección, latencia y viabilidad de despliegue; la selección definitiva debe validarse con el conjunto de datos y el entorno de operación específicos.

2.2.4. Detección vehicular y clasificación supervisada

Los detectores de propósito general entrenados con conjuntos como COCO incluyen categorías vehiculares amplias. En cambio, el prototipo final emplea un detector especializado en una única clase, *car*, cuya función es verificar la presencia de un automóvil antes de activar la localización y lectura de patente. Esta decisión simplifica el flujo de autorización, pero no permite inferir atributos como marca, modelo, color o subtipo de vehículo.

Por esta razón, la autorización se basa en la presencia del vehículo, la lectura de patente y la consulta de registros vigentes en el backend. La verificación de correspondencia entre atributos visuales detallados y el vehículo autorizado queda fuera del alcance del detector integrado.

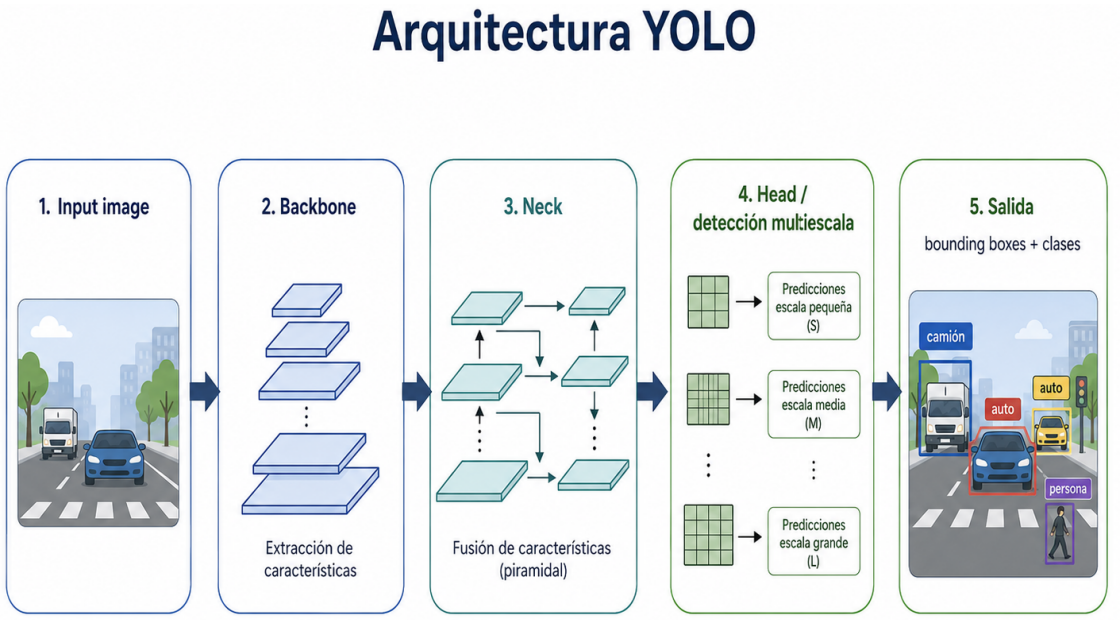


Figura 2.7: Esquema conceptual de una arquitectura tipo YOLO para detección de objetos en tiempo real.

2.2.5. Métricas de Evaluación de Modelos

Para evaluar el desempeño de los modelos de detección empleados en esta memoria, se utilizaron métricas estándar en visión por computador. Su definición en este capítulo permite interpretar posteriormente los resultados sin necesidad de reiterar el significado de cada indicador.

La **precisión** (*Precision*) mide la proporción de detecciones positivas que efectivamente corresponden a objetos correctos:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.1}$$

El **recall** mide la capacidad del modelo para recuperar los objetos relevantes presentes en el conjunto evaluado:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.2}$$

Donde TP corresponde a verdaderos positivos, FP a falsos positivos y FN a falsos negativos [3].

En tareas de detección, además, resulta fundamental considerar la calidad simultánea de localización y clasificación. Por ello se emplea la métrica **Average Precision** (AP), que resume el comportamiento de la curva precisión–recall para una clase determinada. A partir de ella se calcula el **mean Average Precision** (mAP), promediando el desempeño entre clases o umbrales de solapamiento.

En esta memoria se reportan principalmente dos variantes:

- **mAP@0.5:** considera correcta una detección cuando la intersección sobre unión (*Intersection over Union*, IoU) entre la predicción y la anotación real es al menos 0.5.

- **mAP@0.5:0.95:** promedia el desempeño del detector sobre múltiples umbrales de IoU entre 0.5 y 0.95, ofreciendo una evaluación más estricta de la calidad de localización.

La medida de IoU se define como:

$$\text{IoU} = \frac{\text{Área de intersección}}{\text{Área de unión}} \tag{2.3}$$

Estas métricas son particularmente relevantes en control de acceso vehicular, ya que no basta con detectar la presencia de un objeto: también es necesario ubicarlo con suficiente precisión para habilitar procesos posteriores, como la localización de la patente y la validación segura del evento.

2.2.6. Reconocimiento Óptico de Caracteres (OCR)

El reconocimiento de patentes vehiculares consta de dos etapas fundamentales: la detección de la región que contiene la patente y el reconocimiento de los caracteres alfanuméricos presentes en ella.

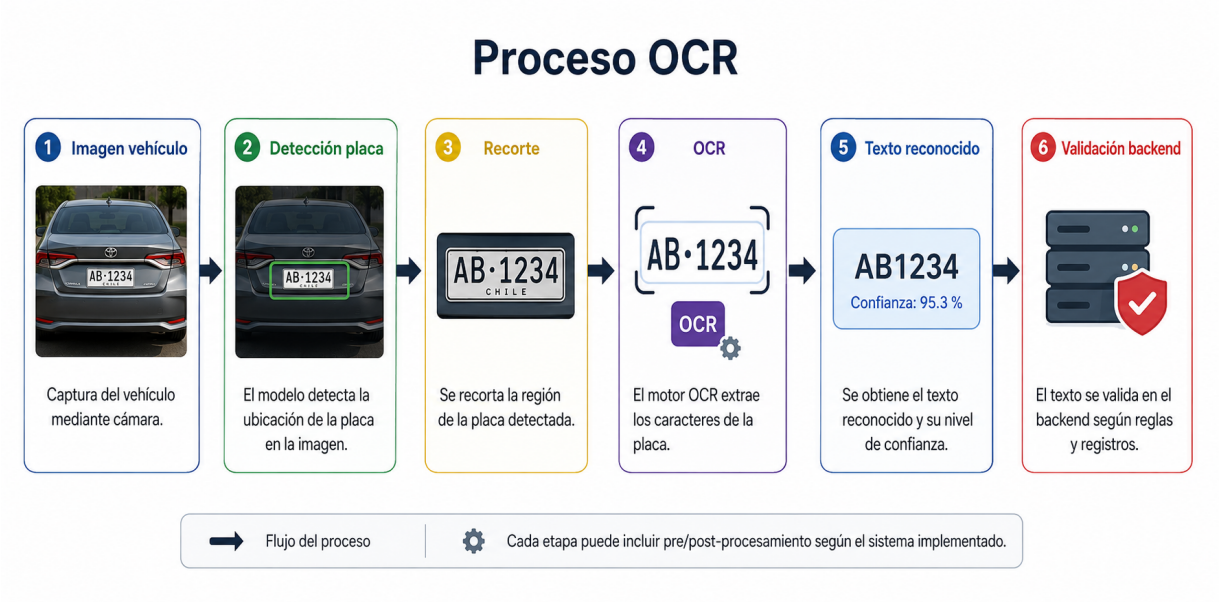


Figura 2.8: Flujo general de detección, recorte de región, reconocimiento de caracteres y validación posterior en un pipeline OCR. Elaboración propia.

Como se observa en la Figura 2.8, el OCR se inserta en una secuencia de procesamiento y validación, no como una decisión aislada. Esta organización permite restringir el reconocimiento a regiones pertinentes y contrastar posteriormente el resultado con reglas de negocio.

La literatura reciente muestra que errores de OCR en ALPR no se distribuyen aleatoriamente: aumentan en condiciones de bajo contraste, desenfoque de movimiento y ángulos de incidencia altos. Por ello, un sistema robusto no debe evaluar OCR en aislamiento, sino dentro de un flujo de validación contextual que considere presencia vehicular y reglas de negocio [14, 23, 25].

Modelos de Atención Visual

Los modelos modernos basados en mecanismos de atención visual, como Florence-2, constituyen modelos fundacionales de visión que emplean instrucciones basadas en texto para resolver diversas tareas visuales y de visión-lenguaje [42]. Su uso en reconocimiento de caracteres debe evaluarse en el dominio de aplicación, pues los resultados publicados del modelo no reemplazan una validación específica sobre patentes vehiculares locales [14].

Estos modelos permiten focalizar dinámicamente regiones relevantes de la imagen, mejorando la precisión del reconocimiento en escenarios reales de acceso vehicular, donde las patentes pueden presentar suciedad, inclinación o variaciones de contraste.

Desde la ingeniería del sistema, su potencial valor no se reduce a la exactitud puntual: un modelo fundacional puede simplificar la integración de capacidades visuales heterogéneas mediante una interfaz de instrucciones. Sin embargo, la conveniencia frente a un OCR especializado depende de métricas reproducibles, compatibilidad de dependencias y recursos computacionales disponibles. En este trabajo, la decisión práctica de integración se documenta en el Capítulo 3 y su comparación cuantitativa queda identificada como una limitación de validación.

Tabla 2.11: Comparación conceptual de enfoques OCR aplicables a lectura de patentes

Enfoque	**Ventaja principal**	**Limitación principal**	**Pertinencia para esta memoria**
OCR clásico basado en reglas	Implementación simple en escenarios muy controlados.	Alta sensibilidad a ruido, inclinación, contraste y calidad del recorte.	Insuficiente para un acceso residencial con variabilidad ambiental.
OCR tradicional generalista	Disponibilidad amplia y bajo costo de integración inicial.	Dependencia de preprocesamiento y comportamiento irregular según dispositivo o entorno.	Útil como línea base, pero limitado para operación robusta.
OCR especializado profundo	Mejor adaptación a variabilidad visual y escenarios complejos.	Mayor costo de integración y dependencia de recursos computacionales.	Adecuado cuando se prioriza robustez de lectura sobre simplicidad extrema.
Modelos con atención visual	Mejor contextualización de regiones relevantes y mayor tolerancia a heterogeneidad del dato.	Requieren integración cuidadosa y validación dentro del pipeline completo.	Se alinean con la estrategia de validación contextual y robustez operacional de *ResiPark*.

La Tabla 2.11 resume esta evolución conceptual. En ella se aprecia que los enfoques más robustos no se distinguen únicamente por una mayor precisión puntual, sino por su capacidad de tolerar heterogeneidad visual y de integrarse en un flujo de validación más amplio.

2.2.7. Internet de las Cosas (IoT) y Comunicación

El monitoreo de ocupación de estacionamientos requiere una red de sensores distribuida capaz de operar de forma confiable en entornos subterráneos. En este contexto, la selección del protocolo

de comunicación resulta crítica debido a la atenuación de señales provocada por estructuras de concreto armado.

Análisis Comparativo de Tecnologías Inalámbricas

La comunicación inalámbrica en edificios residenciales presenta restricciones particulares: atenuación por losas de hormigón armado, multipath en subterráneos, interferencia en bandas congestionadas y disponibilidad energética limitada en nodos remotos. Por ello, la selección de protocolo debe considerar simultáneamente alcance efectivo, consumo energético, latencia aceptable para la operación y seguridad de transporte [6, 5].

En términos prácticos, Wi-Fi ofrece facilidad de integración con infraestructura existente y buen ancho de banda, pero su desempeño se degrada en subterráneos con múltiples obstáculos estructurales. Zigbee mejora consumo y permite redes malladas, aunque su planificación de cobertura puede complejizarse en recintos con geometría irregular. LoRa, en cambio, prioriza bajo consumo y mayor alcance en enlaces de baja tasa de datos, propiedad útil para telemetría binaria de ocupación (libre/ocupado) en estacionamientos [7, 12, 40].

En arquitecturas recientes de smart parking, LoRaWAN aparece como una opción robusta cuando se necesita atravesar estructuras densas y mantener nodos energéticamente austeros, mientras que el procesamiento en fog o edge evita enviar video o telemetría innecesaria a un servidor central [20, 30]. En general, esta lógica favorece reservar la visión por computador para aquellos puntos donde la imagen aporta contexto crítico, y utilizar sensores distribuidos cuando basta con estados binarios o telemetría simple.

Para evitar ambigüedades terminológicas, es importante distinguir entre *LoRa* (modulación/-capa física) y *LoRaWAN* (protocolo MAC y arquitectura de red). Esta separación conceptual resulta relevante al diseñar seguridad, ya que la robustez del enlace radio no reemplaza controles de autenticación, políticas de red y validación de aplicación [41, 40].

Desde seguridad, ningún protocolo es seguro por defecto en ausencia de diseño de arquitectura. La protección efectiva exige segmentación de red, control de rutas permitidas, autenticación de dispositivos y validación en backend de la coherencia de eventos recibidos [11, 10]. En este trabajo, la comparación tecnológica se realizó bajo ese criterio integral y no solo bajo métricas de conectividad.

La Tabla 2.12 sintetiza esta comparación tecnológica. Puede observarse que Wi-Fi ofrece ventajas de integración y throughput, Zigbee se adapta a escenarios mallados de consumo reducido y LoRa resulta especialmente atractivo cuando se privilegia alcance, bajo consumo y telemetría simple en entornos estructuralmente complejos.

2.2.8. Arquitectura de Software y Seguridad

La exposición de servicios mediante plataformas web y APIs requiere un diseño de seguridad robusto, especialmente en sistemas que interactúan con dispositivos físicos y datos sensibles de usuarios. Adicionalmente, en arquitecturas que incorporan dispositivos de borde y nodos IoT, la seguridad no puede depender exclusivamente de mecanismos software. Según [43], el uso de módulos de seguridad hardware, como Trusted Platform Modules (TPM), permite establecer raíces de confianza, proteger credenciales criptográficas y garantizar la integridad del firmware, mitigando ataques de suplantación o manipulación física de dispositivos. Si bien el sistema *ResiPark* implementa actualmente mecanismos de autenticación y autorización a nivel de software, la integración de hardware seguro en dispositivos de borde se identifica como una evolución natural del sistema, especialmente en escenarios de despliegue a gran escala o con mayores requerimientos

Tabla 2.12: Comparación de tecnologías inalámbricas para monitoreo de estacionamientos.

Tecnología	**Fortalezas**	**Debilidades**	**Ajuste al caso residencial subterráneo**	**Implicancia de seguridad**
Wi-Fi	Alta disponibilidad y throughput.	Mayor consumo; cobertura limitada en hormigón denso.	Adecuado en superficie; débil en subterráneos extensos.	Requiere segmentación estricta y hardening de AP.
Zigbee	Bajo consumo; malla cooperativa.	Diseño y mantenimiento de malla más complejos.	Viable en densidades medias de nodos.	Gestión de llaves y control de emparejamiento crítico.
LoRa	Largo alcance y muy bajo consumo para telemetría.	Baja tasa de datos; no apto para video.	Muy adecuado para estado de ocupación por plaza.	Debe complementarse con autenticación en backend y red aislada.

de seguridad. En este contexto, los estándares de autorización modernos promueven arquitecturas desacopladas entre autenticación y lógica de negocio, reduciendo la superficie de ataque y facilitando la escalabilidad del sistema.

En paralelo, la literatura de CPS advierte que la seguridad efectiva requiere alinear controles entre mundo digital y físico: una API correctamente autenticada no compensa una capa de percepción vulnerable, y una red segmentada no compensa reglas de autorización débiles en backend [11, 2, 31]. Esta visión transversal fundamenta la estrategia adoptada en este trabajo, donde cada decisión de software se evalúa por su impacto en la actuación física final.

Autenticación Stateless con JSON Web Tokens

El estándar JSON Web Token (JWT) permite la transmisión segura de información mediante tokens firmados digitalmente, y se enmarca dentro del modelo de autorización definido por OAuth 2.0 [39]. Este enfoque permite delegar la autenticación a una autoridad central, emitiendo credenciales temporales que pueden ser verificadas sin mantener estado en el servidor. Al ser un mecanismo *stateless*, el backend no requiere almacenar sesiones activas, facilitando la escalabilidad horizontal del sistema y reduciendo el impacto de ataques orientados a la gestión de sesiones, como la fijación o secuestro de sesión [27, 28, 29].

En *ResiPark*, los JWT se emplean en el backend móvil para autenticar usuarios y verificar la vigencia de sus solicitudes. La autenticación criptográfica de dispositivos IoT no fue implementada y se mantiene como una medida de fortalecimiento futura.

Seguridad en APIs RESTful

La seguridad en APIs RESTful se fundamenta en principios como la validación estricta de entradas, el control de acceso basado en privilegios mínimos y la segmentación de responsabilidades. Estas prácticas son especialmente relevantes en sistemas IoT, donde un compromiso de un único dispositivo puede propagarse rápidamente si no existen controles adecuados [11]. En este proyecto, estas medidas se implementan mediante esquemas de validación de datos y mecanismos de autorización a nivel de recurso.

Adicionalmente, para aplicaciones de control de acceso, se recomienda complementar la autenticación con validación contextual: presencia de una entidad visual esperada, identidad declarada y regla operativa vigente. Este principio se aplica en la lógica de backend al encadenar detección vehicular, lectura de patente y consulta de autorización [23].

Capítulo 3

Metodología y Diseño de la Plataforma Segura ResiPark

Este capítulo describe la metodología de ingeniería empleada en el desarrollo de *ResiPark*, con énfasis en la definición de controles de ciberseguridad sobre un sistema ciberfísico de acceso vehicular. Siguiendo un enfoque *top-down*, se parte desde la plataforma IoT/ciberfísica general y desde la protección transversal de sus capas, para luego profundizar en amenazas, backend seguro, visión por computador, sensado, comunicaciones e interfaces de operación.

3.1. Metodología de Desarrollo

La metodología de desarrollo seguida en esta memoria se estructuró en cinco etapas sucesivas e iterativas, alineadas con los objetivos específicos del trabajo y con la necesidad de incorporar seguridad desde las primeras decisiones de diseño:

1. **Levantamiento de requerimientos y contexto de riesgo:** se identificaron necesidades funcionales del sistema de acceso vehicular residencial y, en paralelo, restricciones de seguridad, conectividad, trazabilidad y operación en subterráneos.

2. **Diseño de arquitectura y delimitación por capas:** se definió una arquitectura ciberfísica de cuatro capas para separar percepción, comunicaciones, soporte lógico y aplicación, facilitando el análisis de amenazas y la asignación de controles.

3. **Implementación de componentes críticos:** se desarrollaron los módulos de backend, autenticación, lógica de validación, prototipos de visión por computador y mecanismos de sensado/comunicación necesarios para materializar la arquitectura.

4. **Integración progresiva y prueba de concepto:** antes del despliegue final, se construyó un prototipo funcional que permitió validar decisiones de detección, reconocimiento, correlación de eventos y operación de interfaz.

5. **Validación técnica y análisis de seguridad:** finalmente, se evaluó el comportamiento de los modelos, la factibilidad de comunicación y la respuesta de la plataforma frente a amenazas priorizadas, con énfasis en accesos indebidos, abuso de API e inconsistencias entre evidencia física y datos digitales.

Este encuadre metodológico permite distinguir con mayor claridad entre el proceso seguido para llegar a la solución y la descripción de la solución misma, que se desarrolla en las secciones posteriores.

3.2. Arquitectura General de la Plataforma Segura

La plataforma propuesta se concibe como un sistema ciberfísico de cuatro capas: percepción del entorno físico, comunicaciones, soporte lógico y aplicación. Esta organización permite describir el sistema desde lo general hacia lo particular y, al mismo tiempo, identificar con claridad dónde se ubican los mecanismos de protección implementados.

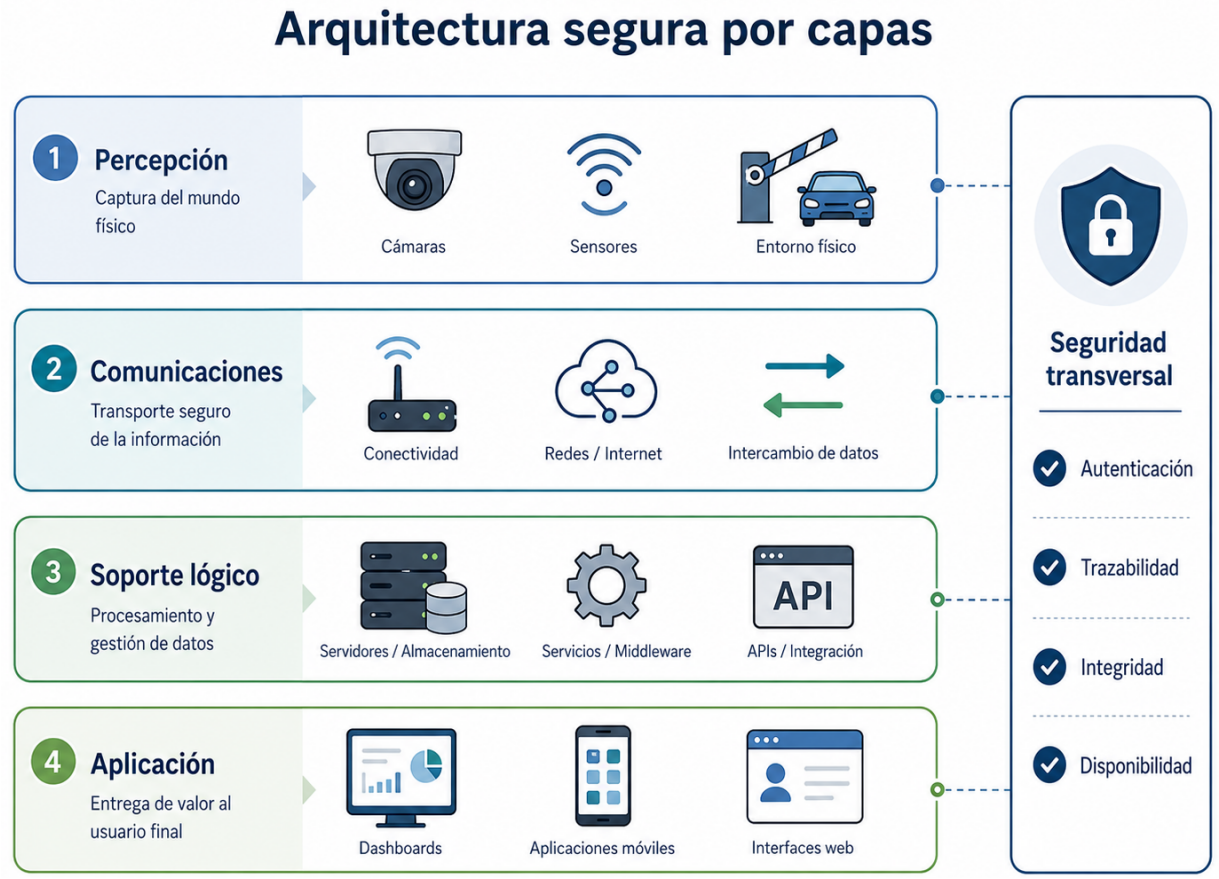


Figura 3.1: Esquema conceptual de la plataforma segura propuesta para *ResiPark*, mostrando las cuatro capas del sistema y la ciberseguridad como propiedad transversal.

Como se observa en la Figura 3.1, la plataforma se organiza en cuatro capas funcionales sobre las cuales se superpone una dimensión transversal de ciberseguridad. En este enfoque, la capa de percepción integra la captura de video y el sensado físico del entorno; la capa de comunicaciones transporta telemetría y eventos entre nodos; la capa de soporte concentra la lógica de negocio, persistencia y validación; y la capa de aplicación expone la operación a residentes y personal de administración. La ciberseguridad se entiende aquí no como un módulo aislado, sino como un conjunto de controles que protege identidad, integridad, trazabilidad y disponibilidad en todo el sistema.

La arquitectura anterior establece el orden de análisis que sigue el capítulo: primero se identifican activos, puntos de entrada y amenazas; a continuación se justifican los controles seleccionados; finalmente se describe cómo dichos controles se materializaron en los componentes de la plataforma.

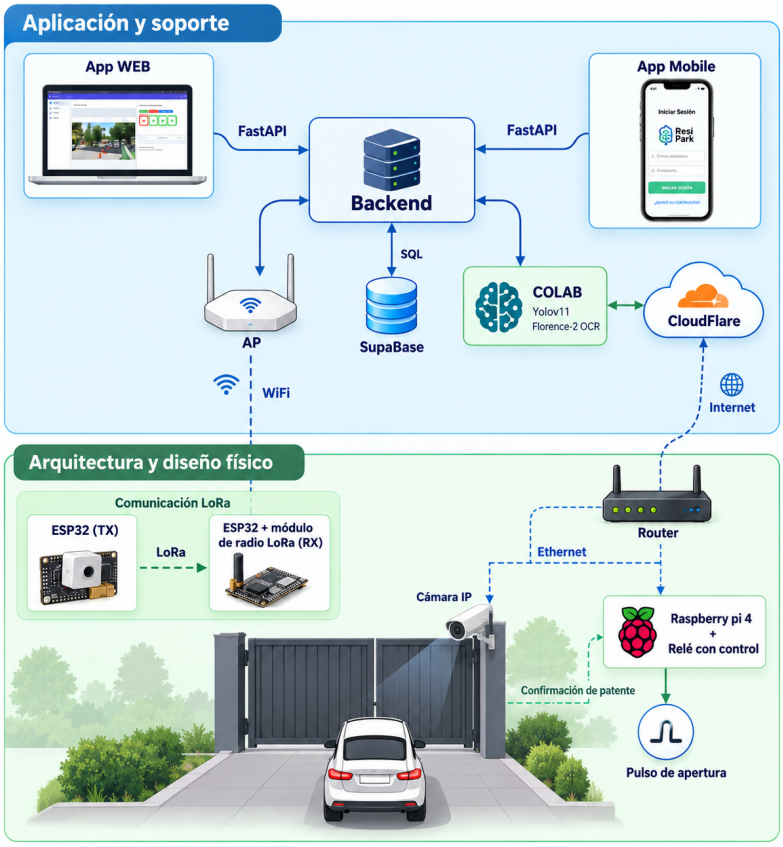


Figura 3.2: Flujo de interacción entre los componentes físicos, de comunicación y de aplicación.

La Figura 3.2 traduce la arquitectura conceptual a los componentes desplegados en el prototipo. En el edificio, los sensores de ocupación se conectan a nodos ESP32 y transmiten telemetría mediante módulos LoRa; en el acceso vehicular, una cámara entrega la captura a una Raspberry Pi que coordina la comunicación con el servicio de inferencia y la activación del relé. La capa de soporte reúne el backend FastAPI, la base de datos PostgreSQL gestionada mediante Supabase y el servicio remoto que ejecuta la detección vehicular y el reconocimiento de patentes. Las interfaces web y móvil consumen estos servicios para operación y consulta de eventos.

El flujo comienza con un evento físico, como el arribo de un vehículo o un cambio en la ocupación de una plaza. Luego, los datos se transmiten y procesan en la capa de soporte, donde se registra el estado o se consulta la autorización correspondiente. El resultado se refleja en las interfaces o genera una acción sobre el entorno, como la apertura del portón. Esta separación permite delimitar los dominios de confianza y reconocer los límites entre dispositivos, red local, servicios externos y aplicaciones cliente como puntos relevantes para el análisis de seguridad.

3.3. Modelo de Amenazas y Vulnerabilidades Consideradas

Como parte del enfoque metodológico individual de esta memoria, se definió explícitamente una superficie de ataque para el sistema antes de consolidar la arquitectura final. El procedimiento consistió en: (i) identificar los activos cuya alteración afectaría la autorización, la integridad de los datos o la disponibilidad del servicio; (ii) determinar los puntos plausibles de entrada asociados a cada activo; (iii) agrupar las amenazas según su efecto sobre confidencialidad, integridad, disponibilidad y autorización; y (iv) realizar una priorización cualitativa según el impacto potencial sobre la seguridad física y la operación residencial. No se aplicó de manera rígida un marco formal único ni una puntuación cuantitativa de riesgo; el análisis propio tomó como referencia categorías compatibles con STRIDE y recomendaciones generales de OWASP para servicios web y dispositivos conectados.

3.3.1. Amenazas Priorizadas

Se priorizaron cuatro familias de ataque por su impacto potencial sobre la seguridad física y digital del sistema:

1. **Suplantación de acceso mediante evidencia visual falsa:** presentación de imágenes impresas o pantallas con patentes autorizadas para inducir la apertura del portón.
2. **Abuso de API y sesión:** reutilización de credenciales, explotación de endpoints sin autenticación suficiente o manipulación de parámetros de entrada.
3. **Intercepción o manipulación de comunicaciones:** ataques *Man-in-the-Middle* en la red local o alteración de tráfico entre dispositivos de borde, backend y clientes.
4. **Manipulación o compromiso de nodos IoT:** uso indebido de sensores o dispositivos conectados para alterar estados operativos o degradar la disponibilidad del sistema.

El esquema conceptual de estas cuatro familias de ataque se presentó en la Figura 2.4 del Capítulo 2. En esta sección, el análisis se profundiza sobre la arquitectura concreta mediante la superficie de ataque y la relación entre activos, entradas y controles.

Tabla 3.1: Activos críticos y puntos de entrada asociados al modelo de amenazas

Activo crítico	**Punto de entrada asociado**	**Amenazas relacionadas**
Evento de autorización de acceso	Cámara, flujo visual y zona de captura	Suplantación mediante evidencia visual falsa
Credenciales y sesiones	Endpoint de autenticación, token y vistas protegidas	Abuso de API y sesión
Telemetría y mensajes internos	Red local, enlace entre nodos y backend	Intercepción o manipulación de comunicaciones
Estado de sensores y ocupación	Nodo IoT, gateway y backend de ingestión	Manipulación o compromiso de nodos IoT
Registro histórico de eventos	Base de datos y backend de negocio	Alteración de trazabilidad y pérdida de evidencia operativa

La Tabla 3.1 complementa el modelo de amenazas al vincular cada riesgo con activos concretos y con sus principales puntos de entrada.

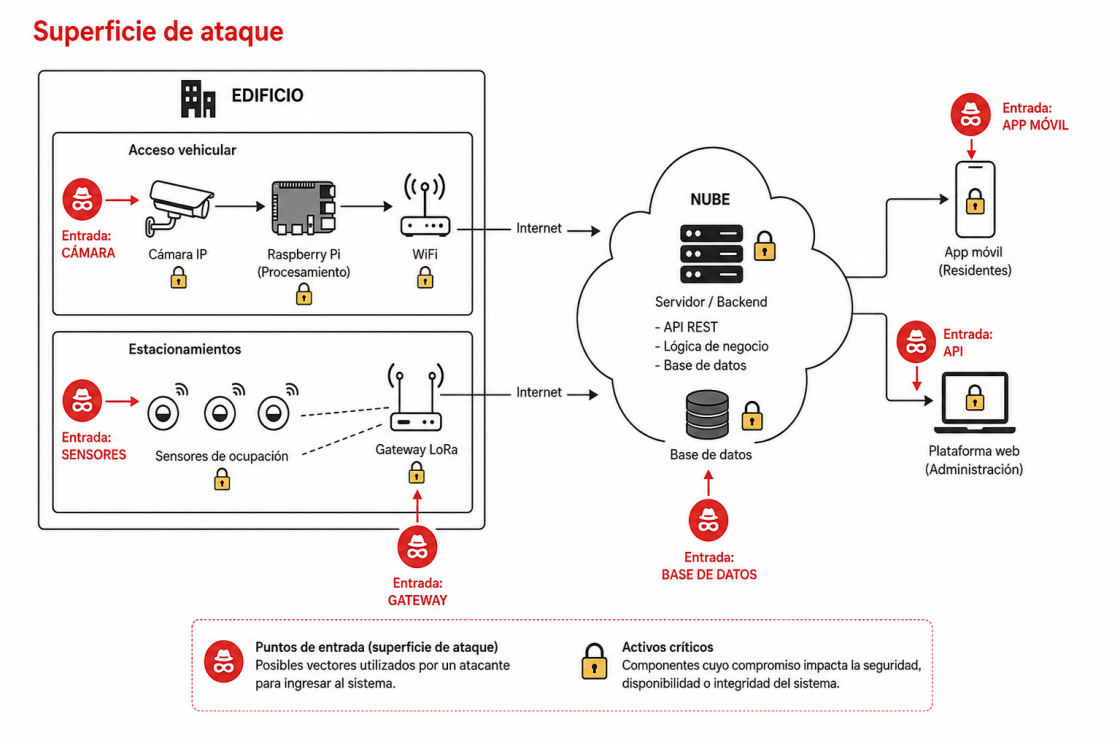


Figura 3.3: Superficie de ataque del sistema, mostrando activos críticos y puntos plausibles de entrada.

La Figura 3.3 aterriza este análisis sobre la arquitectura del sistema, permitiendo visualizar dónde se ubican los activos críticos y por qué ciertos puntos de entrada concentran mayor exposición desde la perspectiva de ciberseguridad.

3.3.2. Criterios de Mitigación Adoptados

Para responder a estas amenazas se definieron criterios de diseño que guiaron las decisiones técnicas desarrolladas en el resto del sistema:

- **Validación por múltiples evidencias:** ninguna acción física crítica debía depender de una única evidencia textual o visual aislada.
- **Autenticación y autorización explícitas:** toda operación sensible del backend debía estar protegida mediante credenciales verificables y controles de acceso asociados al usuario o departamento autenticado.
- **Segmentación por capas:** los dominios de percepción, red, backend y operación debían mantenerse desacoplados para reducir propagación de fallos y facilitar auditoría.
- **Trazabilidad de eventos:** cada decisión de acceso debía quedar registrada para análisis posterior y validación operativa.

La correspondencia entre amenazas y criterios de mitigación puede resumirse de la siguiente forma: la suplantación visual se enfrenta mediante validación por múltiples evidencias; el abuso de API y sesión mediante autenticación y autorización explícitas; la intercepción o manipulación de comunicaciones mediante segmentación por capas; y la manipulación de nodos IoT mediante segmentación y trazabilidad, de modo que un comportamiento anómalo pueda detectarse y auditarse con mayor facilidad.

3.4. Ciberseguridad Transversal por Capas

Con base en el modelo de amenazas, se definieron controles específicos para cada capa de la plataforma. La Tabla 3.2 relaciona explícitamente los activos, amenazas y mecanismos de mitigación que justifican las decisiones de implementación descritas en las secciones posteriores.

Tabla 3.2: Relación entre capa, activo a proteger, amenaza identificada y mecanismo de mitigación

Capa	**Activo a proteger**	**Amenaza identificada**	**Mecanismo de mitigación**
Percepción física	Evidencia visual y activación del acceso	Suplantación mediante imágenes o pantallas con patentes válidas	Detección vehicular previa al OCR, ROI y validación contextual por etapas
Comunicaciones	Integridad y confidencialidad de eventos y telemetría	Intercepción, manipulación o propagación lateral en red local	HTTPS, separación funcional, restricciones de tráfico y segmentación propuesta para despliegue productivo
Soporte lógico	Identidad, sesiones, datos y reglas de autorización	Abuso de API, uso indebido de credenciales e inyección de entradas	JWT, validación de datos, consultas parametrizadas y trazabilidad de operaciones
Aplicación y operación	Historial de eventos y acciones de usuarios	Errores operativos, acceso indebido a vistas críticas o pérdida de trazabilidad	Autorización por usuario o departamento, visualización auditable y separación de perfiles

La Tabla 3.2 permite seguir la trazabilidad desde cada amenaza priorizada hasta el control correspondiente. Las secciones siguientes describen la implementación de estos controles en backend, visión por computador, comunicación IoT e interfaces de operación.

3.5. Desarrollo del Backend y Seguridad

El núcleo lógico del sistema fue desarrollado utilizando el framework **FastAPI** en Python. La selección no se fundamentó en un *benchmark* de rendimiento frente a Flask o Django, sino en el ajuste de cada alternativa a los requerimientos observados durante el prototipo: exponer servicios HTTP para clientes web, móviles y de inferencia; validar solicitudes estructuradas; reutilizar lógica de sesiones y autorización; y mantener los módulos de negocio próximos al ecosistema Python de los modelos de visión. FastAPI ofrece validación basada en modelos Pydantic, documentación OpenAPI y un sistema de inyección de dependencias apropiado para dichos requerimientos [44].

La Tabla 3.3 muestra que la elección se basó en adecuación funcional y mantenibilidad dentro del alcance del prototipo. No permite concluir una superioridad general de FastAPI sobre otros

Tabla 3.3: Criterios de selección de FastAPI para el alcance del prototipo

Requerimiento	**Evidencia en la implementación**	**Implicancia para la selección**
API modular para distintos clientes	El repositorio organiza servicios web, móviles, reconocimiento de patentes y disponibilidad de estacionamientos; el backend web agrupa rutas mediante `APIRouter`.	FastAPI permite estructurar rutas y dependencias reutilizables sin imponer una capa de presentación servidor-renderizada.
Validación de solicitudes	Endpoints como `/auth/login` declaran cuerpos de solicitud mediante `BaseModel` de Pydantic.	La validación tipada integrada reduce código manual de conversión y facilita rechazar estructuras de entrada malformadas [44].
Gestión compartida de recursos y seguridad	Las sesiones de base de datos y la verificación de token se inyectan en endpoints mediante `Depends`.	La inyección de dependencias se adapta al uso repetido de conexiones, autenticación y controles de acceso [44].
Integración con visión por computador	Los servicios de inferencia y lógica de negocio se implementaron en Python.	Evita incorporar otra tecnología de servidor entre los modelos y la API. Flask era viable, pero su enfoque ligero requiere seleccionar e integrar manualmente componentes adicionales; Django provee ORM, administración y plantillas integradas que no eran requisitos centrales, pues la persistencia e interfaces se resolvieron con otros componentes [45, 46].

frameworks ni atribuirle por sí sola propiedades de seguridad; estas dependen de los controles concretos implementados en autenticación, validación, autorización y persistencia.

3.5.1. Diseño Modular y Gestión de Rutas

Para mantener separadas las responsabilidades del backend, se organizó la aplicación mediante `APIRouter`. La autenticación, la gestión de conserjes, departamentos, patentes e historial se exponen como rutas independientes, mientras que la sesión de base de datos se entrega a cada operación a través de una dependencia y se libera al finalizar la solicitud. Esta organización reduce el acoplamiento entre funcionalidades y facilita aplicar controles comunes de autenticación y validación en los endpoints que los requieren.

3.5.2. Implementación de Autenticación con JWT

La autenticación del backend se implementó mediante un mecanismo *stateless* basado en **JSON Web Tokens (JWT)**. Este control contribuye a proteger endpoints definidos como privados, pero no se presenta como garantía absoluta de seguridad: su efectividad depende de la protección de la clave de firma, la expiración, la verificación del token y los controles de autorización aplicados en cada recurso.

El siguiente código muestra la lógica desarrollada para el endpoint de inicio de sesión. En el fragmento se distinguen tres mecanismos con propósitos diferentes: primero, la **normalización del RUT**, que elimina separadores y uniforma el formato de entrada; segundo, el uso de **consultas parametrizadas**, que mitiga el riesgo de inyección SQL al evitar concatenar directamente valores ingresados por el usuario; y tercero, la verificación de credenciales mediante `crypt()`, que compara la contraseña recibida con el valor protegido en PostgreSQL. La normalización no equivale a una validación completa del RUT, pues este fragmento no verifica su dígito verificador; esa comprobación se reconoce como mejora pendiente.

```python
@auth.post("/login")
def login(req: LoginRequest, db: Session = Depends(get_db)):
    # 1. Normalizacion de formato de entrada
    rut_limpio = req.rut.replace(".", "").replace("-", "").strip().upper()

    # 2. Consulta parametrizada segura
    row = db.execute(
        text("""
            SELECT rut, nombre
            FROM conserje
            WHERE rut = :rut AND contrasena = crypt(:password, contrasena)
        """),
        {"rut": rut_limpio, "password": req.contrasena},
    ).mappings().first()

    if not row:
        raise HTTPException(status_code=401, detail="Credenciales invalidas")

    # 3. Generacion de Token con expiracion controlada (60 min)
    token = create_access_token(
        data={"sub": row["rut"]},
        expires_delta=timedelta(minutes=60)
    )

    return {"access_token": token, "token_type": "bearer", ...}
```

Listing 3.1: Lógica de autenticación y generación de tokens seguros

El JWT emitido contiene el identificador del usuario autenticado en el *claim* `sub` y una expiración de 60 minutos en `exp`. El backend lo firma con el algoritmo `HS256`; la clave se obtiene desde la variable de entorno `SECRET_KEY`. Los endpoints protegidos del backend móvil extraen el token *Bearer* del encabezado `Authorization` y verifican firma y expiración antes de continuar. El código del prototipo incluye un valor por defecto de desarrollo para `SECRET_KEY`; dicho valor debe eliminarse y reemplazarse por un secreto gestionado fuera del código para cualquier despliegue productivo.

3.6. Ingeniería de Modelos de Inteligencia Artificial

La ingeniería de modelos de inteligencia artificial se desarrolló siguiendo un enfoque iterativo de experimentación, validación y optimización, considerando restricciones reales de hardware y latencia propias de un sistema de control de acceso.

3.6.1. Conjuntos de datos y detectores integrados

El prototipo final utiliza dos detectores especializados de una sola clase, entrenados con conjuntos publicados en Roboflow Universe. El detector vehicular YOLOv8s se ajustó mediante *fine-tuning* sobre el conjunto *huaung77*, versión 1, cuya clase objetivo es *car* [47]. El detector de patentes YOLO11s se entrenó sobre el conjunto *License Plate Recognition*, versión 11, dirigido a la clase *License_Plate* [48].

Ambos conjuntos utilizan una partición nominal de 70 % para entrenamiento, 20 % para validación y 10 % para prueba. Sin embargo, las métricas preservadas de los entrenamientos integrados corresponden solo a las particiones de validación: 1.936 imágenes y 2.418 instancias para vehículos, y 2.048 imágenes y 2.195 instancias para patentes. Por esta razón, el Capítulo 4 reporta los resultados como métricas de validación y no como una evaluación independiente sobre *test*.

3.6.2. Entrenamiento y Selección de Arquitectura

La selección final de arquitecturas respondió a las necesidades diferenciadas del pipeline. Para detectar presencia vehicular se utilizó **YOLOv8s** ajustado mediante *fine-tuning* sobre el conjunto *huaung77* de Roboflow. Para localizar patentes se utilizó **YOLO11s** con pesos preentrenados y ajuste sobre *License Plate Recognition* de Roboflow. Esta decisión no se presenta como el resultado de un *benchmark* final reproducible entre todas las variantes de YOLO, sino como una selección acotada a los modelos que se integraron y para los cuales se cuenta con resultados de validación.

Ambos entrenamientos finales se ejecutaron durante 50 épocas en Google Colab sobre una GPU Tesla T4. Las curvas de pérdidas de entrenamiento y validación disponibles muestran una disminución sostenida y una estabilización de las métricas hacia las últimas épocas, sin una divergencia persistente entre ambas pérdidas dentro de la partición de validación. Esta observación es consistente con un ajuste estable durante el entrenamiento, pero no sustituye una evaluación independiente de generalización.

Tabla 3.4: Configuración documentada de los entrenamientos finales integrados al prototipo

Modelo	**Inicialización**	**Épocas**	**Tamaño de imagen**	**Hardware**	**Evidencia disponible**
Detector de vehículos (YOLOv8s)	*Fine-tuning* sobre *huaung77* V1	50	No documentado	Tesla T4	Curvas y resumen de validación final
Detector de patentes (YOLO11s)	Pesos preentrenados `yolo11s.pt`	50	640 px	Tesla T4	Curvas y resumen de validación final

La Tabla 3.4 resume la configuración documentada de los entrenamientos finales. El tamaño

de imagen, *batch size*, optimizador y tasa de aprendizaje del detector vehicular no quedaron disponibles en el resumen de entrenamiento; por ello, no se reportan como valores confirmados. En el servicio de inferencia desplegado en Colab se utilizó `FastAPI`, `Uvicorn`, `Cloudflared` y modelos YOLO cargados desde Google Drive, lo que confirma la separación entre procesamiento remoto y captura física en el borde.

3.7. Prototipado y Validación (Prueba de Concepto)

Antes de implementar la arquitectura final basada en servicios web, se desarrolló un prototipo funcional de escritorio como prueba de concepto (PoC).

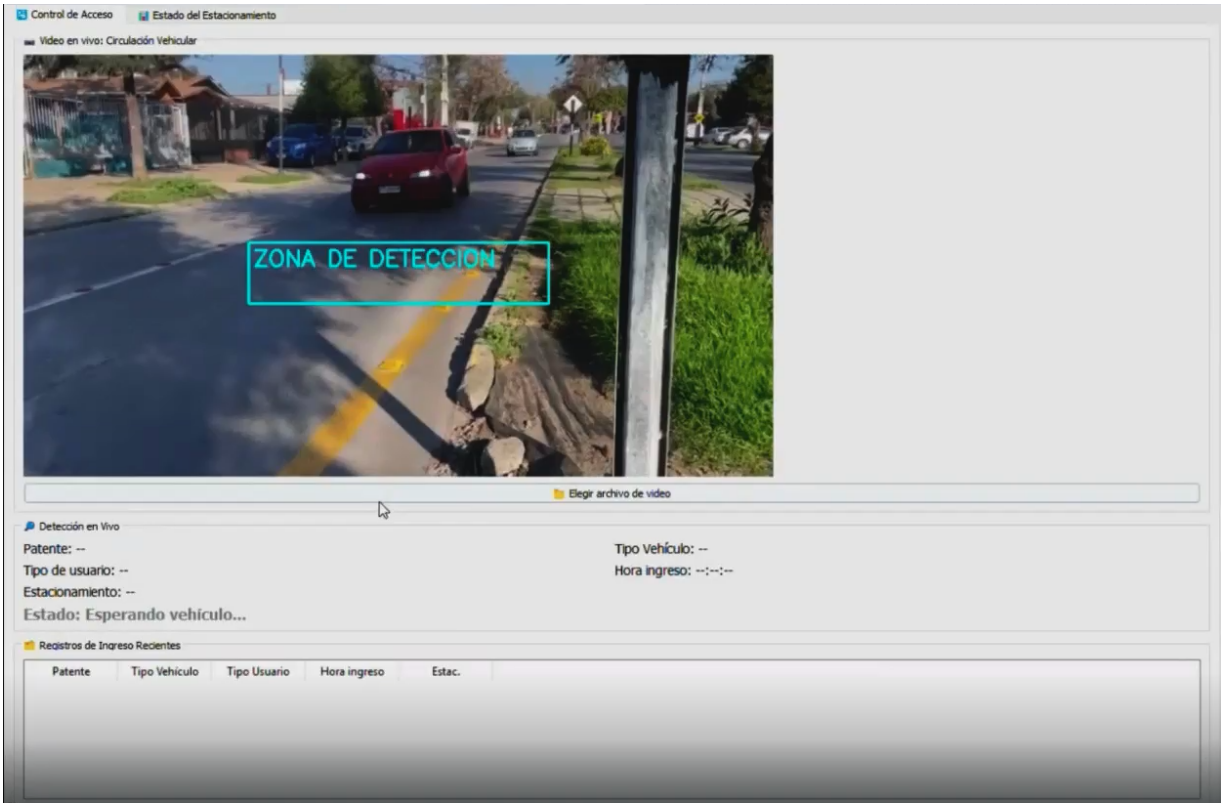


Figura 3.4: Interfaz del prototipo de escritorio (PyQt5) desarrollado para la validación de modelos y lógica de acceso.

La prueba de concepto se utilizó como una etapa de integración funcional previa a los servicios web y al despliegue físico. Su propósito fue comprobar que el flujo de acceso podía encadenar detección vehicular, delimitación de una zona de interés, recorte de patente, reconocimiento OCR, consulta de autorización y actualización del estado de estacionamiento dentro de una misma aplicación. Se consideró satisfactorio que la interfaz procesara un video de entrada, mostrara la clase vehicular y la patente detectada, resolviera la autorización contra la base de datos local y reflejara el resultado en el historial y el estado de la plaza. Como se observa en la Figura 3.4, la interfaz PyQt5 concentraba la visualización del video, la zona de detección, las etiquetas de resultado y el registro de ingresos; por ello permitió detectar fallas de integración entre etapas antes de migrar el flujo a la arquitectura distribuida. Esta etapa no constituye una evaluación formal de rendimiento, seguridad o usabilidad, sino una verificación funcional temprana del

pipeline.

3.7.1. Optimización Computacional y Evolución del OCR

Para reducir invocaciones de OCR fuera del área relevante de la escena, se definió una lógica de activación condicionada por una zona de interés (ROI). El reconocimiento de patente se ejecuta únicamente cuando la caja delimitadora de un vehículo intersecta la zona configurada para el acceso. Esta decisión limita el análisis a eventos visualmente pertinentes y disminuye procesamiento innecesario en cuadros sin posibilidad de ingreso.

La implementación concreta de la intersección geométrica y del recorte se conserva en el repositorio del proyecto. En el cuerpo de la memoria se privilegia su efecto funcional: la secuencia detector vehicular–ROI–recorte–OCR evita procesar indiscriminadamente la imagen completa y facilita identificar en qué etapa se produce una falla.

3.7.2. Evolución del Módulo OCR

Durante el desarrollo del prototipo se utilizó inicialmente *PaddleOCR*. La migración posterior a **Microsoft Florence-2** no se fundamenta en una superioridad cuantitativa demostrada sobre lectura de patentes, ya que no se conservó un *benchmark* reproducible con métricas comunes. La evidencia disponible corresponde a comparaciones visuales exploratorias y a problemas de integración observados con *PaddleOCR*, asociados a dependencias, versiones de CUDA y comportamiento heterogéneo entre entornos con GPU. En consecuencia, Florence-2 se incorporó como una decisión de compatibilidad e integración operativa para el prototipo; la comparación cuantitativa de exactitud OCR permanece fuera del alcance de esta memoria individual.

Como se aprecia en la Figura 2.8, el reconocimiento de caracteres opera como una etapa intermedia entre el recorte de región y la validación posterior. En esta memoria, la decisión entre motores OCR se interpreta como una evaluación de factibilidad de integración, no como una comparación de desempeño. La elección permitió consolidar el flujo de detección, OCR y respuesta HTTP en el servicio remoto de inferencia; una comparación con exactitud de patente completa o tasa de error por carácter deberá realizarse con un conjunto de evaluación preservado y común a ambos motores.

3.8. Integración Frontend

Las interfaces web y móvil materializan los controles de operación definidos para la plataforma: separan las funciones disponibles para residentes y personal administrativo, exponen el historial de eventos y permiten que las acciones de gestión queden asociadas a usuarios autenticados. De este modo, su aporte a la seguridad no radica en la presentación visual, sino en apoyar la trazabilidad de decisiones y la aplicación de permisos diferenciados sobre funciones críticas.

3.8.1. Actualización en Tiempo Real

La interfaz web se desarrolló en **React** e incorpora actualización periódica del historial mediante *polling* cada 10 segundos. Esta estrategia resultó suficiente para el objetivo operativo del prototipo, que era mantener visibilidad reciente de los accesos sin requerir infraestructura adicional de comunicación persistente. A diferencia de WebSockets o Server-Sent Events, el *polling* simplifica la integración inicial y el manejo de reconexiones, pero introduce una latencia de visualización acotada por el intervalo configurado y solicitudes periódicas al backend. Por esta

razón, no se presenta como la alternativa óptima para alta frecuencia de eventos; una migración a comunicación dirigida por eventos queda como mejora para un despliegue con mayor carga. El código de la rutina se conserva en el repositorio del proyecto.

3.9. Implementación de los Componentes por Capas

Una vez definidos la arquitectura conceptual, el modelo de amenazas y los controles asociados, esta sección describe cómo se materializaron los componentes físicos, de red, backend y aplicación que soportan la operación de *ResiPark*. La figura siguiente no reemplaza la arquitectura conceptual de la Figura 3.1; presenta la topología técnica de implementación y las interconexiones entre componentes.



Figura 3.5: Topología técnica de implementación de ResiPark, organizada por componentes de hardware, red y software.

Como se observa en la Figura 3.5, el flujo de información es bidireccional: los sensores en el borde envían telemetría hacia la nube, mientras que las aplicaciones de usuario consumen estos datos y envían comandos de actuación validados por el backend. A diferencia de la Figura 3.1, que representa una vista conceptual centrada en capas y seguridad transversal, esta figura busca mostrar la materialización técnica de los componentes y sus interconexiones.

3.10. Capa de Obtención de Datos (Edge Computing)

Esta capa es responsable de la digitalización del entorno físico. Se compone de dos subsistemas principales: el módulo de control de acceso y el módulo de monitoreo de estacionamientos.

3.10.1. Módulo de Visión y Acceso (Raspberry Pi)

Para el control del portón, se implementó un nodo de borde basado en **Raspberry Pi 4**. En la arquitectura desarrollada, este dispositivo se encarga de la captura de video, del preprocesamiento local básico y de la comunicación con el servicio de inferencia, mientras que la lógica de validación y autorización final se consolida en la capa de soporte.

El nodo de borde se destinó a la captura de video, el preprocesamiento básico y la actuación física sobre el relé del portón. El entrenamiento y la inferencia de los modelos YOLO y Florence-2 se ejecutaron de forma remota en Google Colab, expuestos mediante un servicio `FastAPI` con `Uvicorn` y publicado a través de `Cloudflared Tunnel`. Así, la Raspberry Pi envía la información necesaria al servicio de inferencia y recibe la respuesta de validación para continuar el flujo de acceso, sin asumir la carga computacional de los modelos.

- **Función:** Captura de video en tiempo real y comunicación con el servicio de inferencia.
- **Hardware:** Cámara CSI conectada directamente al bus de la Raspberry Pi para minimizar la latencia.
- **Actuación:** Conexión GPIO directa al relé del motor del portón, permitiendo la apertura física tras recibir una respuesta positiva de autorización desde el servidor.

La Figura 3.6 muestra este flujo de operación: la cámara CSI entrega imágenes a la Raspberry Pi, el nodo prepara la información necesaria para el análisis y, una vez validado el evento por la capa de soporte, se acciona el relé conectado al portón mediante GPIO.

Alcance de la orden de apertura

En el prototipo, la respuesta positiva del backend habilita la actuación del relé sobre el portón. No se implementó ni evaluó un mecanismo criptográfico específico que vincule la orden de apertura con un identificador de evento, una vigencia breve y un único consumo; por tanto, la resistencia frente a la reutilización de una autorización previa no forma parte de las garantías validadas.

Para un despliegue productivo, la orden debería emitirse como una autorización firmada y dirigida al dispositivo de borde, incorporando como mínimo un identificador único de evento, la identidad de la Raspberry Pi destinataria, una expiración de corta duración y un indicador de uso único. El nodo debería verificar estos atributos antes de accionar el GPIO y el backend registrar el consumo de la orden, de modo que una autorización expirada, destinada a otro nodo o ya utilizada sea rechazada.

3.10.2. Red de Sensores de Ocupación

Para la gestión de estacionamientos, donde la visión artificial fue descartada por las oclusiones (ver Capítulo 2), se desplegaron sensores ultrasónicos de proximidad en cada plaza. La implementación física utilizó sensores HC-SR04; no se incorporaron sensores infrarrojos de manera simultánea ni como mecanismo redundante. La elección respondió a la necesidad de obtener un

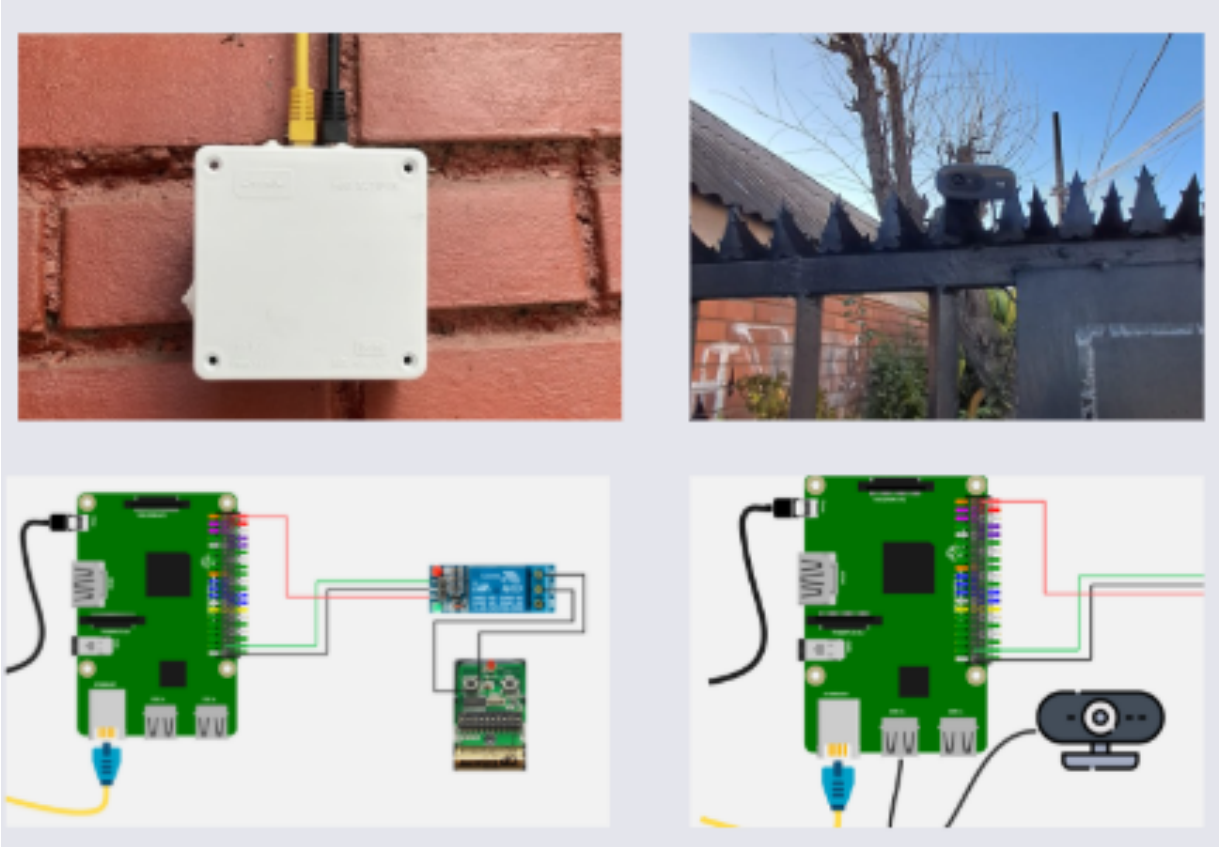


Figura 3.6: Esquema de conexión del nodo de borde para el control de acceso.

estado binario de ocupación mediante medición de distancia a corta escala, con una integración simple con el ESP32 y transmisión de eventos discretos por LoRa.

- **Sensor:** Se utilizaron sensores ultrasónicos HC-SR04 para discriminar ocupación vehicular en distancias cortas.
- **Precisión operativa:** Fueron calibrados para detectar presencia vehicular a una distancia menor a 60 cm.
- **Microcontrolador:** Cada nodo es gestionado por un ESP32 con enlace LoRa mediante módulos SX1276/78, seleccionado por su bajo costo, disponibilidad y capacidad de operación en modos de bajo consumo.

3.11. Capa de Red y Comunicaciones

Dada la naturaleza híbrida del recinto (acceso en superficie y estacionamientos subterráneos), se diseñó una estrategia de red mixta para garantizar la conectividad en todas las zonas.

3.11.1. Topología LoRa para Subterráneos

Los estacionamientos ubicados en los niveles -1 y -2 presentan un desafío de propagación para las señales de 2.4 GHz (Wi-Fi). Para mitigar esto, se implementó una red **LoRa (Long Range)** en topología estrella.

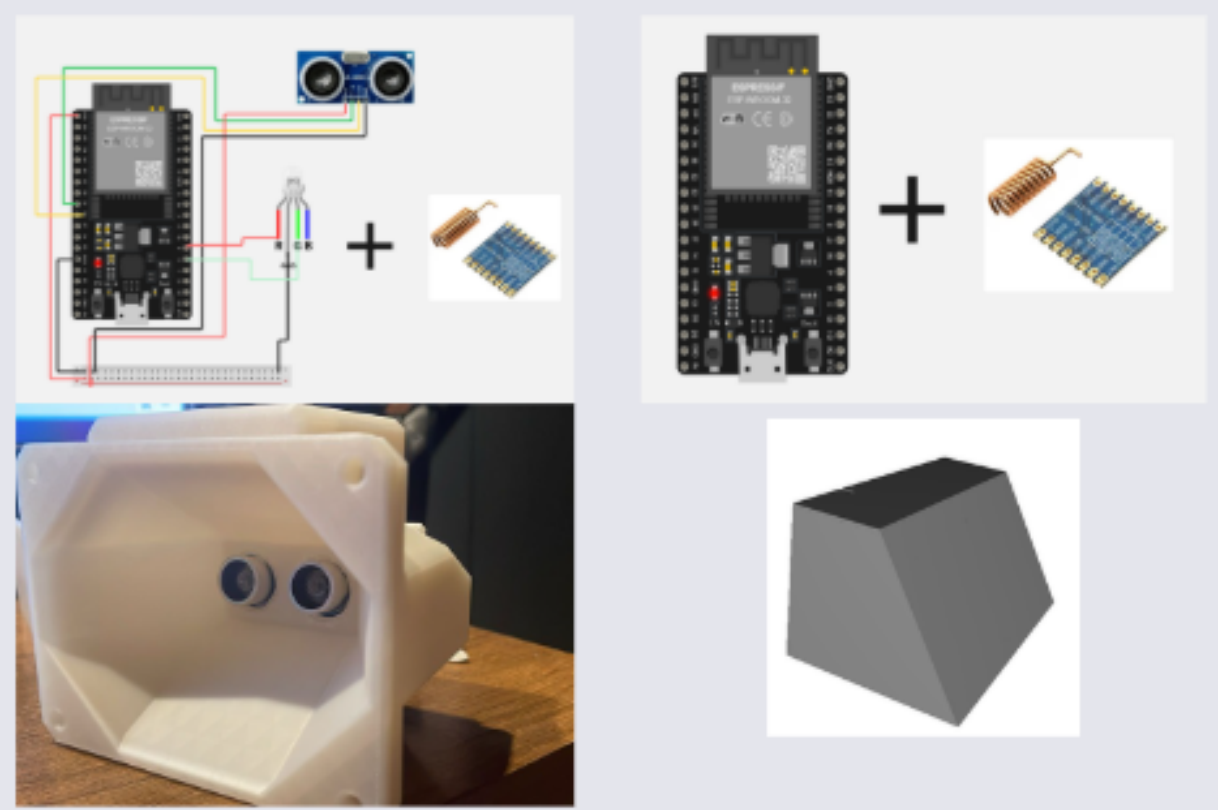


Figura 3.7: Topología de red híbrida: LoRa para sensores subterráneos y Wi-Fi para el control de acceso en superficie.

Los nodos sensores (ESP32) transmiten paquetes de telemetría pequeños (ocupado/libre) mediante modulación LoRa hacia un Gateway central ubicado en la recepción. Bajo las condiciones evaluadas para este prototipo y de acuerdo con la literatura revisada, esta tecnología resultó adecuada para telemetría simple en subterráneos de hormigón armado, por su alcance efectivo y bajo consumo [12, 40]. El piloto físico no incorporó instrumentación temporal suficiente para reportar latencia o pérdida de paquetes en terreno; la caracterización cuantitativa mediante simulación se presenta en el Capítulo 4.

3.11.2. Medidas Recomendadas de Seguridad en la Transmisión

El piloto no incluyó una configuración administrada de red que permita demostrar la implementación de segmentación, filtrado de tráfico o protección ARP. Por ello, las siguientes medidas se documentan como recomendaciones de endurecimiento para un despliegue productivo, y no como controles implementados o validados experimentalmente:

- **Segmentación de red:** separar los dispositivos IoT de la red de residentes mediante VLAN o una red dedicada, reduciendo la exposición a movimiento lateral desde equipos no administrados.
- **Filtrado por defecto:** aplicar reglas de firewall de denegación por defecto y habilitar solo los flujos necesarios entre sensores, gateway, borde y API.
- **Protección frente a suplantación local:** evaluar entradas ARP estáticas o mecanismos

equivalentes en segmentos controlados. Esta medida solo reduce ataques de *ARP spoofing* en la red local y no protege de forma general frente a escenarios *Man-in-the-Middle*; debe complementarse con canales cifrados y autenticación de los extremos.

En una siguiente iteración, estas medidas deben configurarse y verificarse junto con el uso de HTTPS para las comunicaciones con la API. La evidencia disponible permite describir la topología y los componentes del prototipo, pero no acredita una política de red productiva; por tanto, no se atribuye a estas recomendaciones una mitigación ya demostrada.

3.12. Capa de Soporte y Lógica de Negocio (Backend)

La inteligencia central del sistema reside en la nube. Se diseñó una arquitectura de microservicios utilizando **FastAPI** (Python), framework seleccionado por su capacidad de manejo asíncrono, lo cual es crítico para evitar el bloqueo del hilo de ejecución durante el procesamiento de peticiones de inferencia de inteligencia artificial.

3.12.1. Persistencia, Seguridad y Tiempo Real

Para la persistencia de datos se utilizó **Supabase** basado en PostgreSQL. Esta decisión arquitectónica permitió habilitar las siguientes capacidades dentro del alcance del prototipo:

- **Persistencia:** almacenamiento de usuarios, vehículos autorizados, reservas, eventos e información de ocupación sobre PostgreSQL.
- **Actualización de estado:** soporte para consultar y reflejar el estado de ocupación en las interfaces, sin que ello constituya una evaluación de escalabilidad o de latencia en producción.
- **Diseño de RLS:** definición de políticas de seguridad a nivel de fila (*Row Level Security*) como línea de endurecimiento para restringir el acceso según el rol del usuario.

La integración entre Supabase y FastAPI se resolvió separando responsabilidades. FastAPI actúa como capa de lógica de negocio, autenticación, validación y exposición de endpoints, mientras que Supabase provee la persistencia sobre PostgreSQL y servicios auxiliares de tiempo real utilizados por la interfaz. De esta forma, la API desarrollada controla las decisiones sensibles del sistema y la base de datos opera como soporte de almacenamiento y consulta.

No se conservó evidencia versionada de políticas RLS aplicadas ni fue posible acceder al proyecto Supabase para verificarlas. En consecuencia, RLS no se presenta como un mecanismo implementado o evaluado en esta memoria. Como recomendación de despliegue, las políticas deberían limitar las reservas de cada residente a su propio departamento, reservar la administración de vehículos autorizados a perfiles administrativos y permitir la inserción de eventos automáticos solo al backend autenticado. Estas restricciones complementarían, pero no reemplazarían, la autorización aplicada por FastAPI.

3.12.2. API RESTful

El backend expone una API REST segura que actúa como intermediario entre los dispositivos IoT y las aplicaciones de usuario.

La Tabla 3.5 resume los endpoints más relevantes de la API. Desde la perspectiva de seguridad, `/auth/login` corresponde al punto de emisión de credenciales; `/detect/plate` y `/parking/live`

Tabla 3.5: Endpoints principales de la API de ResiPark.

Método	Ruta	Función
POST	`/auth/login`	Autenticación y emisión de JWT.
POST	`/detect/plate`	Procesamiento de imagen y validación de acceso.
GET	`/parking/live`	Estado actual de sensores IoT.
POST	`/reservas`	Creación de solicitudes de estacionamiento.

intervienen directamente en el flujo operativo del sistema y deben quedar restringidos a clientes autorizados; mientras que `/reservas` forma parte de la interacción de usuarios autenticados. En consecuencia, el acceso a estos recursos no es homogéneo y debe gobernarse mediante autenticación, autorización y validación contextual.

La autenticación se expone mediante la ruta `/auth/login` tanto para la aplicación web como para la aplicación móvil. En el backend móvil, el token emitido corresponde a un JWT firmado con algoritmo `HS256` y una expiración de 60 minutos; las rutas que aplican la dependencia de verificación responden con código HTTP 401 cuando el token es inválido o no se presenta correctamente. Esta protección no se encontraba aplicada de forma uniforme en las rutas administrativas del backend web. Antes de un despliegue productivo, la misma dependencia de autenticación debe extenderse a dichas rutas y complementarse con autorizaciones por rol y por recurso, de manera que cada operación administrativa verifique tanto la identidad del usuario como sus privilegios.

3.13. Capa de Aplicación y Experiencia de Usuario

La interacción humana con *ResiPark* se realiza a través de dos interfaces diferenciadas según el perfil del usuario: una aplicación móvil para residentes y una plataforma web para la administración del recinto.

3.13.1. Aplicación Móvil (Residentes)

La aplicación móvil permite a los residentes gestionar su acceso y el de sus visitas de forma autónoma.

Autenticación y Seguridad

El acceso requiere credenciales validadas contra la base de datos de residentes activos. Al iniciar sesión, el token JWT emitido por el backend se conserva localmente para mantener la sesión del usuario durante su vigencia. En la implementación verificada, la aplicación móvil almacena el token en `AsyncStorage` bajo la clave `accessToken`, junto con el identificador del departamento autenticado. Este mecanismo, aunque funcional para el prototipo, debe entenderse como una decisión de implementación con alcance acotado; en un despliegue productivo convendría migrar a almacenamiento seguro del sistema operativo y complementar el flujo con renovación controlada de sesión.

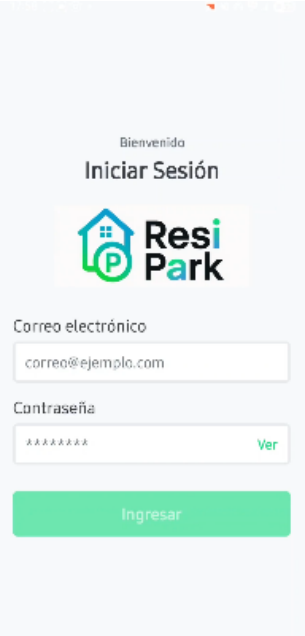
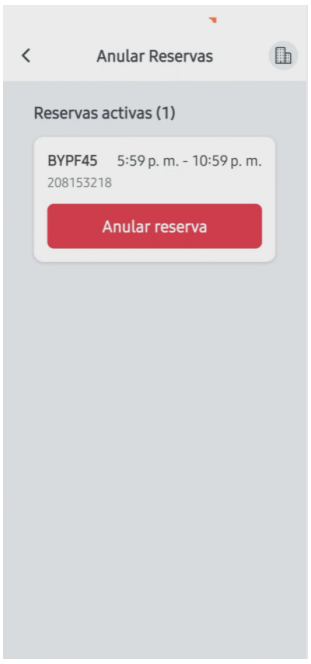


Figura 3.8: Pantalla de inicio de sesión de la aplicación móvil.

Gestión de Reservas

El flujo de reservas fue diseñado para ser intuitivo: el residente selecciona la fecha, ingresa la patente de la visita y el sistema asigna automáticamente un estacionamiento disponible, o lo deja en estado "Pendiente"para asignación manual si el recinto está lleno.



(a) Historial de reservas.



(b) Creación de nueva visita.

Figura 3.9: Interfaces de gestión de visitas en la aplicación móvil.

3.13.2. Plataforma Web (Administración)

Diseñada para pantallas de escritorio, esta interfaz es la herramienta principal del conserje. Permite visualizar en tiempo real el estado de los portones y la ocupación de los estacionamientos subterráneos.

Dashboard de Monitoreo en Tiempo Real

Esta es la vista principal del sistema. Muestra una representación visual de los estacionamientos. Los cuadros cambian de color (Verde/Rojo) en tiempo real según la telemetría enviada por los sensores LoRa, permitiendo al conserje detectar anomalías visualmente.

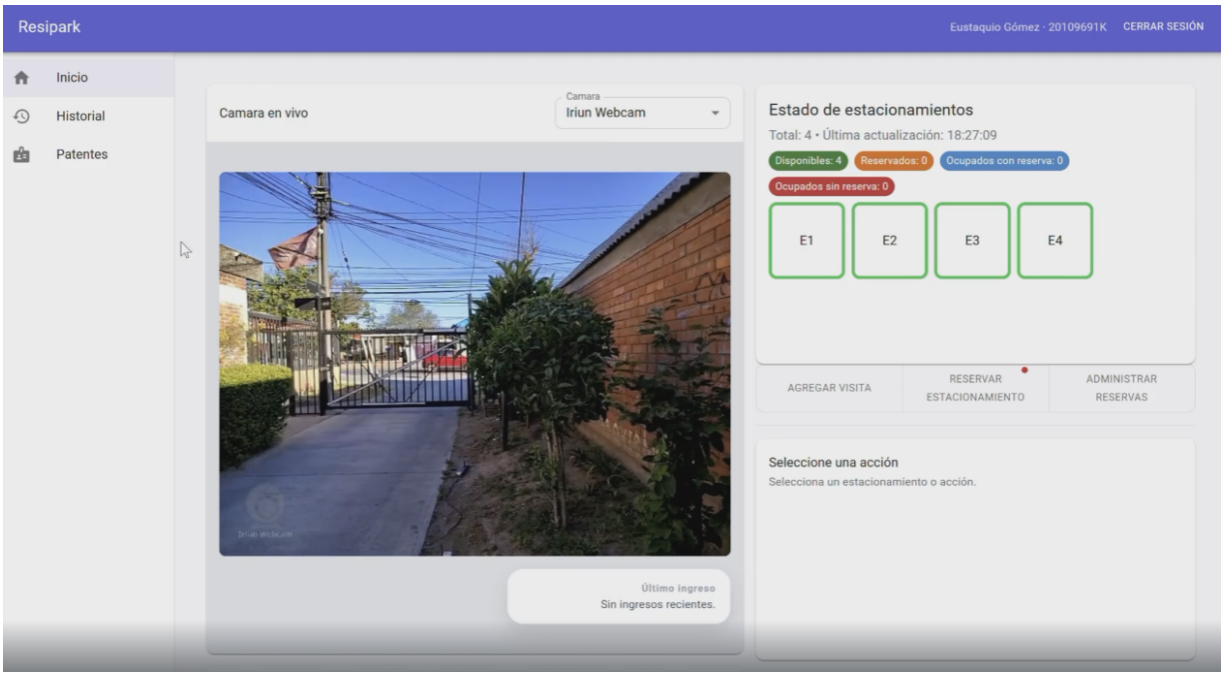


Figura 3.10: Dashboard principal: Visualización en tiempo real de la ocupación de estacionamientos.

Historial y Trazabilidad

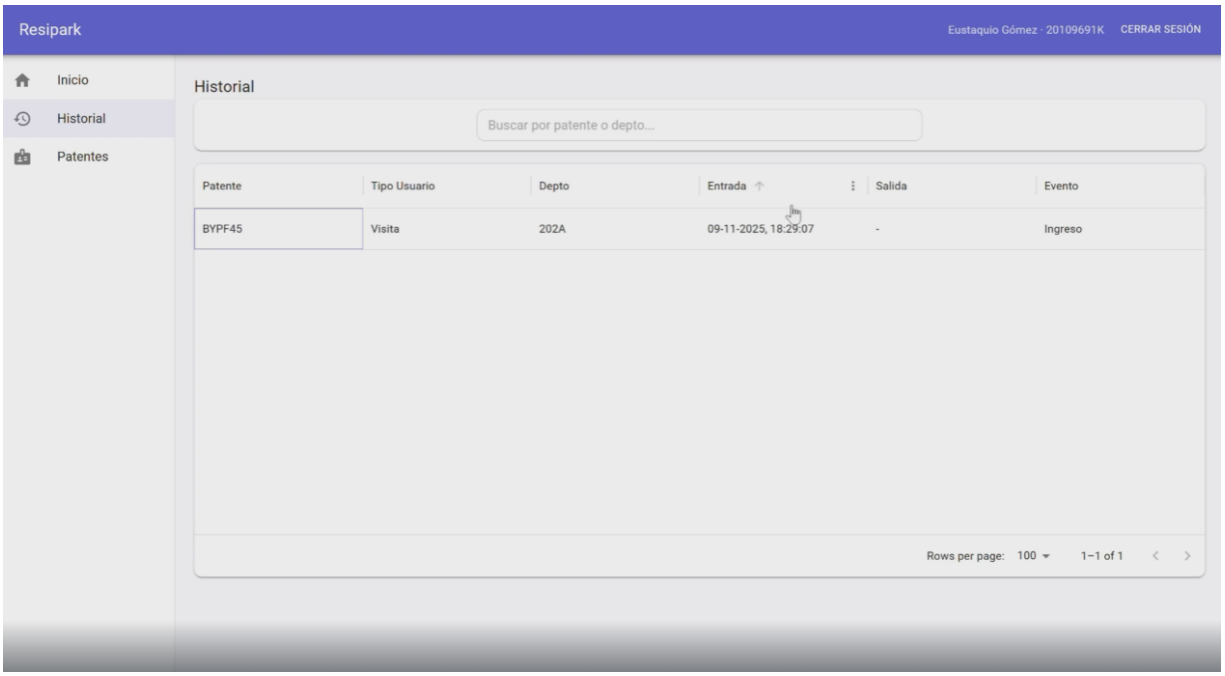


Figura 3.11: Módulo de historial de accesos con evidencia fotográfica de las patentes detectadas.

Para apoyar la auditoría operativa, el sistema mantiene un historial de los eventos relevantes de acceso. No se implementó un mecanismo específico de inmutabilidad o integridad criptográfica, por lo que el término se limita a *registro histórico* y no a un registro a prueba de modificaciones. La Figura 3.11 muestra el módulo donde se visualizan la patente capturada, la hora exacta y el resultado de la validación (Acceso Concedido/Denegado).

Gestión de Residentes (CRUD)

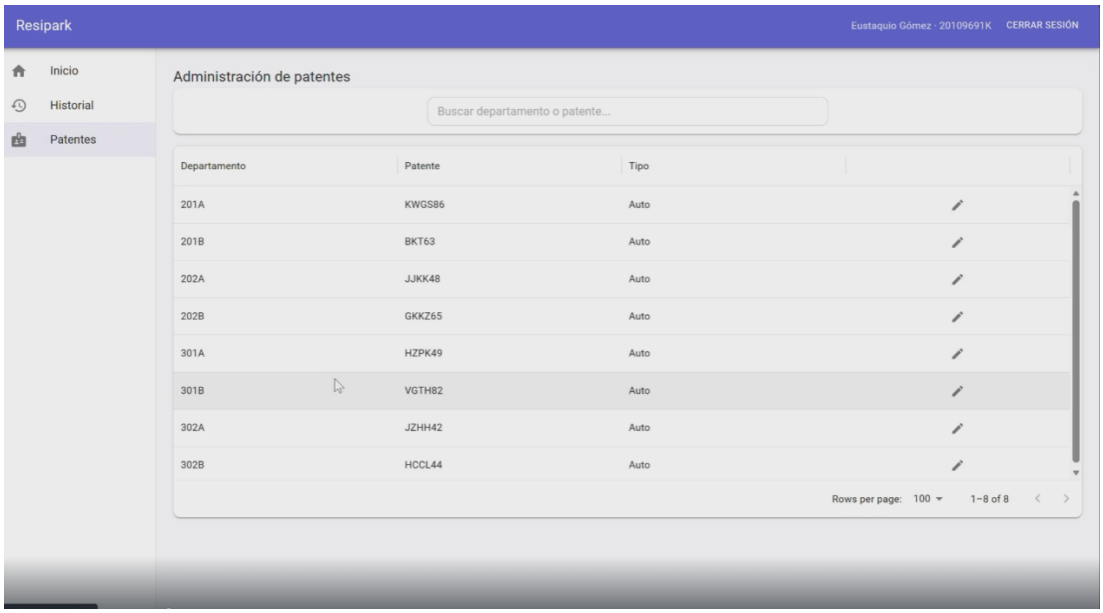


Figura 3.12: Interfaz de administración de residentes y vehículos autorizados.

El administrador posee privilegios para gestionar la base de datos de usuarios. Esta interfaz permite registrar nuevos residentes, actualizar sus vehículos autorizados o revocar accesos en caso de mudanza, impactando directamente en las reglas de validación del modelo de IA.

Capítulo 4

Resultados, Validación Experimental y Análisis de Seguridad

En este capítulo se presentan los resultados cuantitativos y cualitativos obtenidos tras la implementación de *ResiPark*. La exposición sigue el orden de los objetivos específicos: identificación de amenazas, mecanismos de backend y aplicación, integración de visión por computador, y factibilidad de la plataforma en sus capas físicas y de comunicación. Como no se documentó un conjunto de prueba independiente junto con los entrenamientos finales, las métricas reportadas para los modelos se interpretan como resultados sobre validación.

4.1. Alcance y Criterios de Evaluación

La evidencia disponible combina resultados de entrenamiento, verificaciones funcionales del prototipo, observaciones de integración y pruebas de campo. Por esta razón, cada resultado se interpreta según su nivel de respaldo: *implementado* cuando el mecanismo está presente en los artefactos del proyecto; *verificado funcionalmente* cuando se observó su operación en el prototipo o piloto; y *propuesto* cuando corresponde a una medida de endurecimiento no configurada ni evaluada. Esta distinción evita atribuir a la plataforma propiedades de seguridad que no fueron demostradas experimentalmente.

4.2. Resultado del Objetivo 1: Identificación de Amenazas y Superficie de Ataque

El modelo de amenazas presentado en el Capítulo 3 permitió identificar como riesgos prioritarios la suplantación visual, el acceso no autorizado a la API, la manipulación de entradas, la intercepción de comunicaciones y la alteración de dispositivos IoT. El resultado de este objetivo es la definición de los activos y puntos de entrada que orientan la evaluación posterior; la Tabla 4.4 consolida la correspondencia entre estas amenazas, los controles considerados y el nivel de evidencia disponible.

4.3. Resultado del Objetivo 2: Backend y Capa de Aplicación

4.3.1. Verificación Funcional de la Plataforma Web

La plataforma web permitió centralizar la gestión de seguridad y visualizar eventos operativos de manera consistente con los requerimientos definidos para conserjería y administración.

Panel de Historial y Gestión

La vista de historial de la Figura 4.1 incorpora una consulta periódica al backend cada 10 segundos. Este intervalo no representa la latencia de procesamiento del evento, sino que define la ventana máxima de actualización visual: un registro ya persistido puede aparecer en la interfaz entre el ciclo inmediatamente posterior y, como máximo, el siguiente intervalo de consulta. Por tanto, la interfaz ofrece trazabilidad operativa con una actualización visual acotada por 10 segundos; no se levantó una serie temporal independiente para estimar la latencia extremo a extremo del frontend.

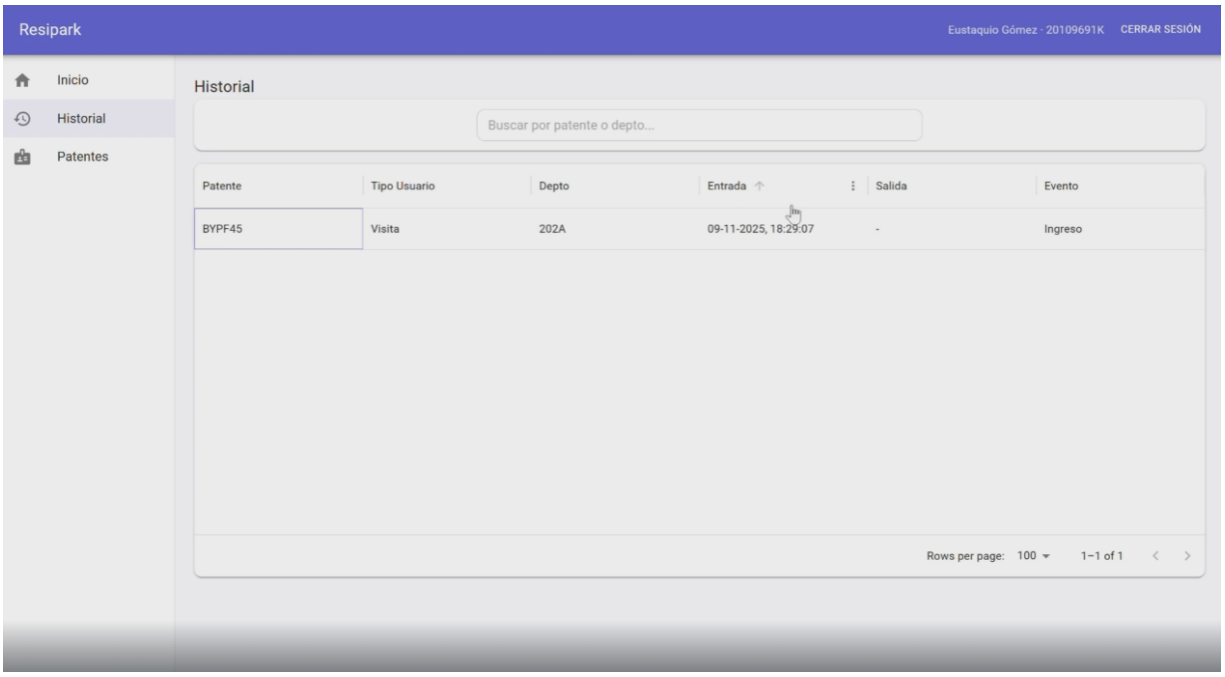


Figura 4.1: Módulo de historial de accesos desarrollado en React, visualizando datos desde Supabase.

Por otro lado, el módulo de Administración de Patentes (Figura 4.2) permite la gestión CRUD (*Create, Read, Update, Delete*) de los vehículos autorizados, impactando directamente en las reglas de negocio que validan el ingreso automático.

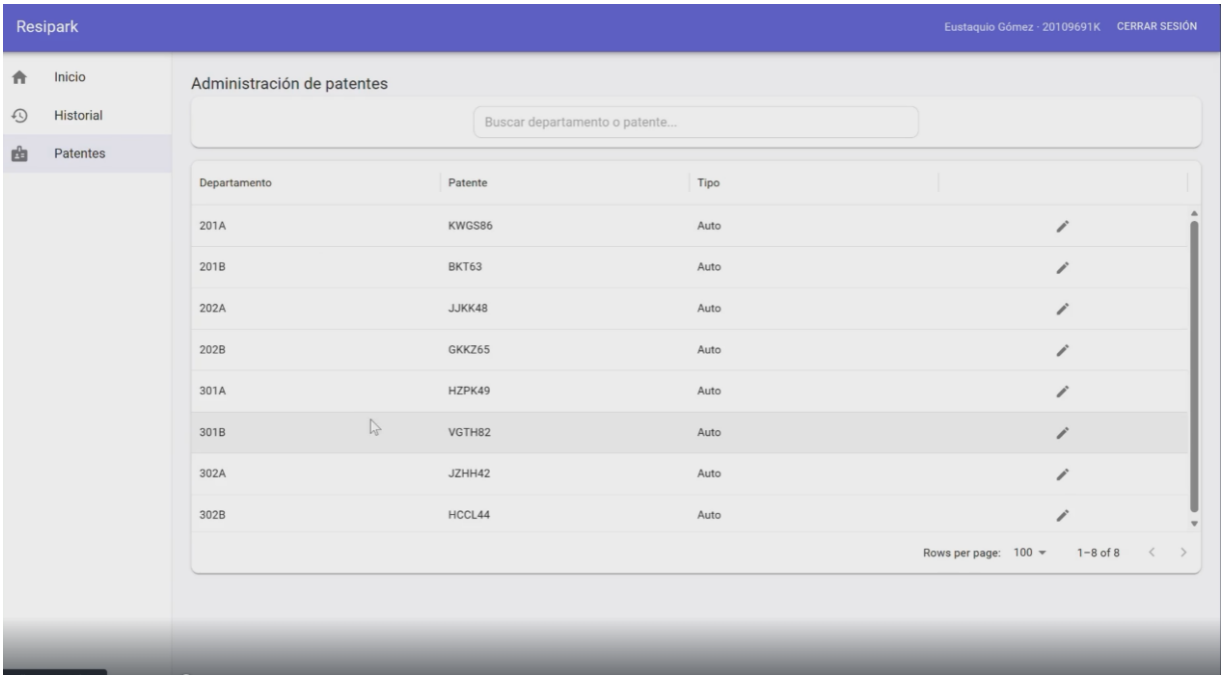


Figura 4.2: Interfaz de gestión de vehículos y residentes.

Esta sección presenta resultados de verificación funcional y no una evaluación formal de usabilidad. Durante la integración se comprobó que la interfaz muestra el historial de accesos, actualiza periódicamente la información disponible y permite gestionar los vehículos autorizados. La aceptación de usuarios, tiempos de tarea y facilidad de uso requieren un instrumento de evaluación específico y se tratan como trabajo futuro.

4.4. Resultado del Objetivo 3: Visión por Computador y Validación de Acceso

El subsistema de visión aporta evidencia para la decisión de autorización, la cual se completa con las reglas de backend y los controles de operación descritos en las secciones siguientes. Por ello, sus métricas se interpretan como un componente del flujo completo y no como una medida aislada de seguridad. A continuación se presentan los resultados de validación de los detectores de vehículos y patentes.

4.4.1. Métricas de Entrenamiento

Para evaluar el desempeño de los detectores finales se utilizó la partición de validación registrada por cada ejecución de entrenamiento. Las métricas presentadas no corresponden a validación cruzada ni a un conjunto de prueba independiente: describen el comportamiento de cada modelo sobre su respectiva partición de validación.

La Tabla 4.1 muestra que el detector vehicular alcanzó mAP50 de 0,995 y mAP50-95 de 0,973 sobre su conjunto de validación. El detector de patentes obtuvo mAP50 de 0,977; su mAP50-95 de 0,716 revela una mayor sensibilidad cuando se exige un solapamiento más estricto de la caja delimitadora, consistente con el menor tamaño relativo de la patente dentro del fotograma. Estos valores caracterizan la detección de objetos y no la exactitud completa del OCR.

Tabla 4.1: Métricas globales de validación de los detectores finales.

Detector	**Modelo**	**Clase evaluada**	**Validación**	**Precisión**	**Recall**	**mAP50**	**mAP50-95**
Vehículos	YOLOv8s	*car*	1.936 imágenes; 2.418 instancias	0,997	0,988	0,995	0,973
Patentes	YOLO11s	*License_Plate*	2.048 imágenes; 2.195 instancias	0,989	0,948	0,977	0,716

Los detectores finales se entrenaron para una única clase: *car* en el caso de vehículos y *License_Plate* en el caso de patentes. En consecuencia, no corresponde desagregar precisión, *recall* o AP entre tipos de vehículo.

Los resultados deben interpretarse considerando dos aspectos. Primero, ambos detectores alcanzan valores altos de mAP@0.5, pero el descenso de mAP@0.5:0.95 es más marcado en el detector de patentes, lo que evidencia una mayor sensibilidad al criterio de localización más estricto. Segundo, no se evaluó un conjunto de prueba independiente; por ello, los valores caracterizan el desempeño sobre validación y no constituyen una medición exhaustiva del comportamiento del sistema en un despliegue productivo. Además, las condiciones de baja iluminación, lluvia y otros escenarios ambientales requieren una evaluación sistemática adicional.

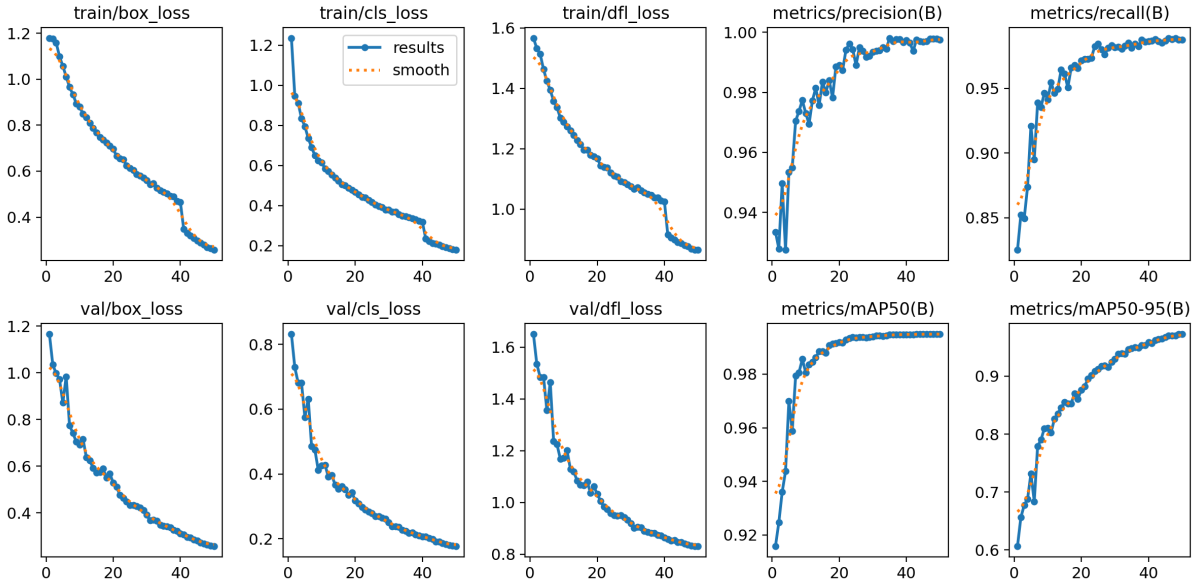


Figura 4.3: Curvas del entrenamiento final del detector de vehículos YOLOv8s.

La Figura 4.3 presenta las curvas de pérdida de entrenamiento y validación, además de la evolución de precisión, *recall*, mAP@0.5 y mAP@0.5:0.95 durante las 50 épocas del detector vehicular. La disminución conjunta de las pérdidas y la estabilización de las métricas hacia las últimas épocas indican que el ajuste alcanzó un régimen estable dentro de la partición utilizada. Esta observación se limita al proceso de entrenamiento y validación; la ausencia de métricas preservadas sobre un conjunto de prueba independiente restringe cualquier conclusión sobre

generalización fuera de los datos disponibles.

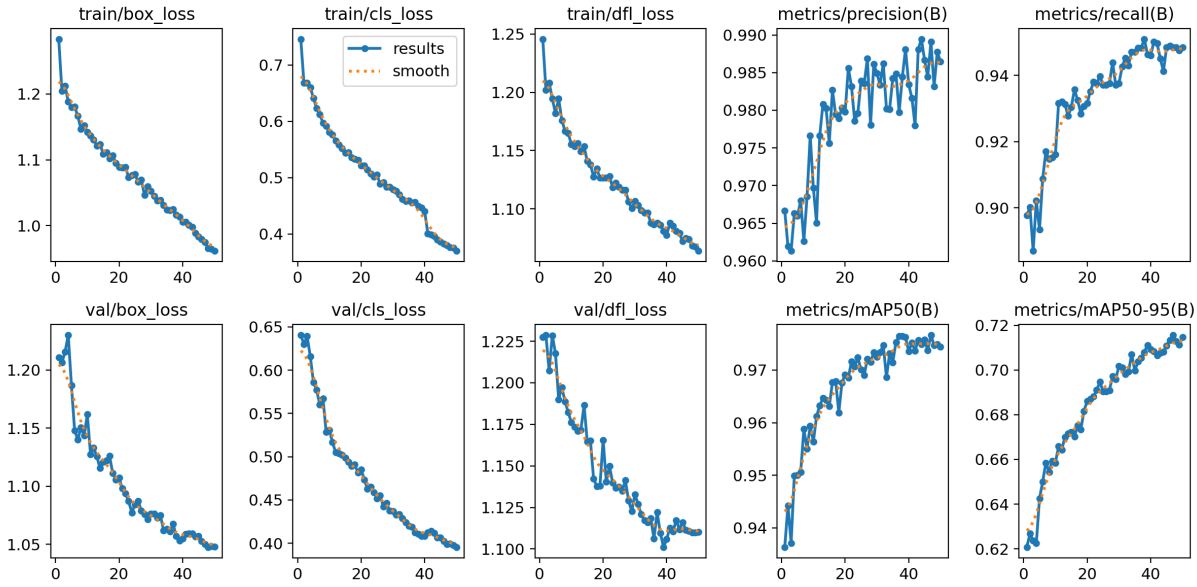


Figura 4.4: Curvas del entrenamiento final del detector de patentes YOLO11s.

En la Figura 4.4 las pérdidas de entrenamiento y validación también disminuyen durante las 50 épocas, mientras las métricas de precisión, *recall* y mAP se estabilizan al final del proceso. El valor de mAP50-95 se mantiene por debajo del mAP50, lo que refleja que localizar una patente con alta precisión geométrica es una tarea más exigente que detectar su presencia. La figura no reemplaza una evaluación de reconocimiento de caracteres ni una prueba independiente de generalización.

4.4.2. Comparativa de Motores OCR

Durante el desarrollo se evaluaron dos motores OCR: PaddleOCR y Microsoft Florence-2.

Tabla 4.2: Observaciones de integración que motivaron el cambio de motor OCR.

Motor OCR	**Observación durante la integración**	**Implicancia para el prototipo**
PaddleOCR	La integración inicial presentó conflictos entre dependencias, versiones de CUDA y entornos con GPU heterogéneos.	El servicio no pudo consolidarse con una configuración reproducible para el flujo remoto de inferencia.
Florence-2	Se integró con el servicio remoto que concentraba detección, OCR y respuesta HTTP en Google Colab.	Permitió mantener un flujo funcional de reconocimiento dentro de la configuración de cómputo disponible.

La Tabla 4.2 documenta una decisión de ingeniería de integración y no una comparación de exactitud OCR. PaddleOCR se utilizó inicialmente, pero los conflictos de dependencias y compatibilidad entre CUDA, GPU y librerías impidieron mantener una configuración reproducible

en el entorno de inferencia. Florence-2 se adoptó porque permitió consolidar el flujo remoto de detección, reconocimiento y respuesta HTTP en la infraestructura disponible. En consecuencia, la selección se fundamenta en factibilidad operativa del prototipo, sin atribuirle una superioridad cuantitativa en lectura de patentes.

No se preservó un conjunto común de patentes ni resultados comparables para estimar exactitud de patente completa o *Character Error Rate*. Una evaluación posterior deberá ejecutar ambos motores sobre el mismo conjunto independiente, informar los criterios de preprocesamiento y reportar métricas de reconocimiento a nivel de carácter y de patente completa.

4.5. Validación Transversal de Medidas de Ciberseguridad

Esta sección integra la evidencia de los objetivos 1, 2 y 3, ya que las amenazas identificadas, los controles de backend y las validaciones visuales forman parte de una misma decisión de autorización. Se presenta de forma transversal para evitar evaluar cada mecanismo de manera aislada y para distinguir entre controles implementados, evidencia experimental y medidas únicamente propuestas.

4.5.1. Protocolo de pruebas de autenticación, autorización y entradas

Para evaluar los mecanismos efectivamente disponibles se ejecutó una batería controlada sobre el backend móvil de FastAPI, utilizando el cliente de pruebas del propio framework y la configuración de autenticación del servicio. Las solicitudes con token ausente, token alterado o token expirado se dirigieron al endpoint protegido `GET /vehiculos/`. Para la autenticación se envió una solicitud de inicio de sesión con correo y contraseña inexistentes. La validación de entradas se verificó mediante un tipo no admitido en `tipo_vehiculo` y mediante un intervalo temporal inválido al crear una reserva. Estos casos fueron rechazados antes de realizar operaciones de escritura.

La autorización se evaluó sobre una reserva existente utilizando un token de prueba válido asociado a un departamento distinto del propietario de la reserva. La operación `PUT /reservas/{id}` respondió sin modificar registros, debido a que la consulta de actualización restringe el recurso mediante el identificador de departamento contenido en el token. Esta prueba corresponde a aislamiento por departamento; el prototipo no implementa un modelo de roles diferenciado. La Tabla 4.3 resume el procedimiento y los resultados observados.

Los resultados muestran que las rutas móviles evaluadas rechazan tokens ausentes, alterados y expirados, y que las reglas de pertenencia impiden modificar una reserva desde un departamento distinto. Asimismo, las validaciones de esquema y de consistencia temporal rechazaron los cuerpos de solicitud evaluados. La prueba no cubre ataques de inyección, limitación de tasa, revocación de tokens ni escalamiento por roles, por lo que esos aspectos permanecen fuera de la evidencia experimental disponible.

La Tabla 4.4 resume el vínculo entre amenaza, escenario y evidencia. Su propósito es distinguir controles validados en el backend móvil de aquellos que solo fueron observados durante el piloto o propuestos como endurecimiento futuro.

4.5.2. Pruebas funcionales preliminares frente a escenarios de suplantación visual

Se realizaron pruebas funcionales para observar cómo respondía el flujo de acceso ante evidencia visual insuficiente, patentes sin autorización y situaciones de posible suplantación. Cada

Tabla 4.3: Resultados de pruebas controladas sobre autenticación, autorización y validación de entradas del backend móvil.

Amenaza evaluada	**Prueba realizada**	**Resultado esperado**	**Resultado observado**
Acceso sin autenticación	Solicitud a `GET /vehiculos/` sin cabecera `Authorization`.	Rechazo antes de acceder al recurso.	**401 Unauthorized**.
Token alterado	Solicitud al mismo endpoint con JWT inválido.	Rechazo del token no verificable.	**401 Unauthorized**.
Token expirado	Solicitud al mismo endpoint con JWT cuyo vencimiento ya había ocurrido.	Rechazo de sesión expirada.	**401 Unauthorized**.
Credenciales incorrectas	Inicio de sesión con correo inexistente y contraseña inválida.	No emitir token de acceso.	**401 Unauthorized**.
Acceso a recurso ajeno	Intento de editar una reserva existente con token de otro departamento.	No modificar la reserva ajena.	**404 Not Found**; no se actualizó ninguna fila.
Entrada inválida	Tipo no entero en `tipo_vehiculo` y horario de término anterior al inicio en una reserva.	Rechazar la solicitud sin escritura.	**422 Unprocessable Entity** y **400 Bad Request**, respectivamente.

escenario se ejecutó una vez durante el desarrollo del prototipo; por tanto, los resultados se presentan como observaciones funcionales y no como tasas de efectividad frente a ataques de presentación.

Lógica escalonada de autorización

El flujo de autorización se organiza en etapas consecutivas. Primero, el detector debe identificar un objeto de la clase *car*. Solo entonces se localiza la patente y se ejecuta el OCR. Cuando se obtiene una lectura utilizable, el backend consulta si la patente corresponde a un vehículo residente o a una reserva vigente. El relé se activa únicamente después de una autorización positiva y el evento se registra para trazabilidad. Si una etapa falla, el proceso termina sin enviar una orden de apertura.

Escenarios evaluados

La evaluación consideró los siguientes escenarios:

- **Video sin vehículos:** se procesó una secuencia sin vehículos dentro de la zona de detección para verificar que no se iniciaran las etapas de OCR, autorización ni apertura.
- **Persona sosteniendo una patente:** se presentó una patente aislada frente a la cámara, sin un vehículo asociado, para observar si podía activar el flujo de acceso.
- **Vehículo con patente no registrada:** se presentó un vehículo real con una patente sin autorización vigente en el backend.

Tabla 4.4: Matriz de validación de seguridad alineada con las amenazas priorizadas.

Amenaza	**Escenario o revisión**	**Resultado esperado**	**Resultado observado**	**Estado de evidencia**
Suplantación visual básica	Video sin vehículo, persona sosteniendo una patente y vehículo con patente no registrada.	No accionar el portón sin vehículo o sin autorización vigente.	No se generó apertura ante ausencia de vehículo ni ante falta de autorización.	Prueba funcional parcial.
Acceso no autorizado a la API	Solicitudes sin JWT, con token alterado, expirado y credenciales inválidas contra rutas móviles.	Rechazar solicitudes no autenticadas.	Todas las solicitudes fueron rechazadas con **401**.	Validado en backend móvil.
Acceso a recurso ajeno	Edición de una reserva existente con token válido de otro departamento.	Impedir modificación de datos ajenos.	**404**; no se actualizó ninguna fila.	Validado como aislamiento por departamento.
Manipulación de entradas	Tipo no entero y horario de reserva inconsistente.	Rechazar la solicitud sin escritura.	Respuestas **422** y **400**.	Validado para los casos evaluados.
Exposición de red e IoT	Revisión de la topología y de las recomendaciones de endurecimiento.	Reducir acceso no necesario a componentes críticos.	VLAN, firewall administrado, protección ARP y RLS no fueron configurados ni evaluados en el piloto.	Propuesto.

- **Patente dibujada o adulterada:** se instaló una reproducción manual de patente sobre un vehículo para observar el comportamiento de detección y OCR ante tipografía, espaciado y bordes no estándar.
- **Misma patente en un vehículo de apariencia distinta:** se presentó una patente válida sobre un vehículo con color diferente para identificar el alcance de la validación implementada.

La Figura 4.5 muestra el escenario de presentación aislada. Aunque la patente se ubicó frente a la cámara, la ausencia de una detección vehicular impidió continuar con el OCR y la autorización.



Figura 4.5: Prueba de presentación aislada de una patente. Al no detectarse un vehículo asociado, el flujo no continuó hacia OCR ni apertura.

En el escenario de patente no registrada, el sistema detectó el vehículo y ejecutó el OCR, pero el backend rechazó la solicitud al no encontrar una autorización vigente. La Figura 4.6 muestra la salida del cliente durante esta verificación controlada.

```
x:\Carpetas\Universidad\Memoria\Github\resipark-pmm\ai\Client.py:14: SyntaxWarning: invalid escape sequence '\I'
  url = "ai\IMG_9880.mp4"
Iniciando procesamiento de camara. Presiona 'Ctrl+C' para detener.
No se detectan autos. Finalizando...
✅ Procesamiento terminado.
📊 Resumen:
   Detecciones encontradas: 0
   Patentes detectadas: []
PS X:\Carpetas\Universidad\Memoria\Github\resipark-pmm> & C:/Python313/python.exe x:/Carpetas/Universidad/Memoria/Github/resipark-pmm/ai/Client.py

x:\Carpetas\Universidad\Memoria\Github\resipark-pmm\ai\Client.py:14: SyntaxWarning: invalid escape sequence '\I'
  url = "ai\IMG_9880.mp4"
x:\Carpetas\Universidad\Memoria\Github\resipark-pmm\ai\Client.py:15: SyntaxWarning: invalid escape sequence '\V'
  url2= "Videos\Video prueba.mp4"
Iniciando procesamiento de camara. Presiona 'Ctrl+C' para detener.
[20:18:22] ✅ DETECTADO (Frame 10):
   🚗 Vehículo: car
   📊 Conf. Vehículo: 0.877
   🔠 Patente: BYPF75
   📈 Conf. Patente: 0.780
   🔍 Verificando patente...
   ❌ Error en verificación
--------------------------------------------------
[20:18:32] ✅ DETECTADO (Frame 30):
   🚗 Vehículo: car
   📊 Conf. Vehículo: 0.931
   🔠 Patente: BYPF45
   📈 Conf. Patente: 0.807
   🔍 Verificando patente...
   ❌ Error en verificación
--------------------------------------------------
[20:18:36] ✅ DETECTADO (Frame 40):
   🚗 Vehículo: car
   📊 Conf. Vehículo: 0.925
   🔠 Patente: BYPF45
   📈 Conf. Patente: 0.802
   🔍 Verificando patente...
   ❌ Error en verificación
--------------------------------------------------
[20:18:39] ✅ DETECTADO (Frame 50):
   🚗 Vehículo: car
   📊 Conf. Vehículo: 0.948
   🔠 Patente: BYPF45
```

Figura 4.6: Salida del cliente durante la verificación de una patente sin autorización vigente. La detección no derivó en una orden de apertura.

La Figura 4.7 ilustra una reproducción dibujada sobre un vehículo. En esta prueba se detectó el vehículo y se activó el OCR, pero la lectura no alcanzó una calidad utilizable para solicitar

autorización; en consecuencia, el portón permaneció cerrado.



Figura 4.7: Prueba con una reproducción dibujada de patente instalada sobre un vehículo.

La Figura 4.8 presenta una misma patente en vehículos de apariencia distinta. Este caso evidenció una limitación del prototipo: el detector integrado solo identifica la clase *car*, mientras que el backend autoriza según la patente registrada. Por ello, una diferencia de color, marca o modelo no participa en la decisión actual.



Figura 4.8: Caso de una misma patente asociada a vehículos de apariencia distinta.

Resultados observados

La Tabla 4.5 resume el comportamiento observado en cada etapa del flujo. Debido a que se realizó una ejecución por escenario, los resultados se expresan como decisiones funcionales y no como porcentajes de efectividad.

Tabla 4.5: Resultados observados en pruebas funcionales preliminares frente a escenarios de suplantación visual

Escenario	**Vehículo detectado**	**OCR ejecutado**	**Autorización**	**Apertura**	**Observación**
Video sin vehículos	No	No	No consultada	No	El flujo finalizó en la etapa de detección vehicular.
Persona sosteniendo una patente	No	No	No consultada	No	La patente aislada no produjo apertura.
Vehículo con patente no registrada	Sí	Sí	Denegada	No	El backend no encontró una autorización vigente.
Patente dibujada o adulterada	Sí	Sí, lectura no utilizable	No consultada	No	La lectura no aportó una patente válida para la consulta de autorización.
Misma patente en vehículo diferente	Sí	Sí	Concedida	Sí	El backend autorizó por patente; no se evaluaron color, marca ni modelo.

Los primeros cuatro escenarios mostraron una respuesta conservadora del flujo: la ausencia de vehículo, la patente aislada, una patente no registrada y una lectura no utilizable no generaron apertura. Sin embargo, el último caso también delimita claramente el alcance del prototipo. Una patente válida instalada sobre un vehículo distinto puede ser autorizada porque la implementación no compara atributos visuales adicionales con el registro del vehículo.

Alcance y limitaciones

Estas pruebas aportan evidencia funcional de que la autorización no depende solo de una lectura OCR aislada y que el sistema rechaza patentes sin autorización vigente. No obstante, la detección de un objeto de clase *car* no demuestra que la escena corresponda a un vehículo físico real, ni descarta fotografías o pantallas que contengan simultáneamente un vehículo y una patente válida. Tampoco se realizaron repeticiones sistemáticas por escenario ni variaciones controladas de distancia, iluminación o condiciones climáticas.

Como trabajo futuro se requiere incorporar mecanismos específicos de detección de ataques de presentación, validar sintácticamente la lectura de patente con criterios reproducibles y agregar atributos visuales, como color, marca o modelo. Ante una discrepancia o evidencia ambigua, el sistema debería detener la apertura automática y derivar el evento a revisión de conserjería, conservando el registro correspondiente.

4.5.3. Mitigación de Abuso de API y Sesión

Las pruebas descritas en la Tabla 4.3 validaron el comportamiento de autenticación y manejo de sesiones del backend móvil. La ausencia de JWT, la alteración de su contenido y su expiración fueron rechazadas con código **401**; del mismo modo, las credenciales inválidas no generaron un

token de acceso. La prueba de autorización entre departamentos verificó que una reserva no puede ser actualizada cuando el identificador contenido en el token no corresponde a su propietario.

En relación con las entradas, los modelos de solicitud rechazaron un valor no entero para `tipo_vehiculo` con código **422**, mientras que una reserva con término anterior al inicio recibió código **400**. Estas verificaciones aportan evidencia sobre los casos evaluados, pero no sustituyen una prueba exhaustiva de inyección, *fuzzing*, revocación de tokens o abuso por volumen de solicitudes. Además, las rutas administrativas del backend web no aplican actualmente la dependencia de verificación JWT; por ello, los resultados de esta subsección no se extrapolan a dicha API. Como fortalecimiento necesario, la protección debe extenderse de forma uniforme a las rutas administrativas y asociarse a controles de rol y recurso antes de un despliegue productivo.

4.5.4. Mitigación en Red e Infraestructura IoT

La segmentación mediante VLAN, el filtrado *deny-all*, la protección ARP y RLS se mantienen como recomendaciones para un despliegue productivo. No fueron configurados ni evaluados durante el piloto, por lo que no se atribuye a la red implementada una mitigación comprobada frente a interceptación, movimiento lateral o manipulación de telemetría. El uso de HTTPS debe formar parte de una configuración productiva, junto con autenticación explícita entre dispositivos y backend. En particular, el nodo de borde debería identificarse mediante una credencial individual verificable por el servidor; *Mutual TLS* (mTLS) es la alternativa preferente cuando se requiere autenticación criptográfica de ambos extremos. Como transición, podrían utilizarse claves de API únicas, rotables y transportadas exclusivamente sobre HTTPS, pero estas no reemplazan las garantías de identidad y protección de claves que aporta mTLS.

4.6. Resultado del Objetivo 4: Factibilidad Técnica y Operación en Terreno

La red de sensores fue evaluada en un entorno subterráneo real.

4.6.1. Fiabilidad de la Detección Física

Las pruebas funcionales del nodo ESP32 con sensor ultrasónico HC-SR04 permitieron ajustar un umbral de ocupación inferior a **60 cm**. Bajo ese criterio, la detección de un vehículo generó el cambio de estado ocupado y su propagación hacia el dashboard mediante el enlace LoRa y el gateway. Esta evidencia confirma la integración del flujo físico–digital en una plaza de prueba, pero no permite estimar precisión, falsos positivos o falsos negativos, ya que no se preservó el número de intentos ni una campaña de medición controlada.

4.6.2. Desempeño de la Conectividad LoRa

La evaluación de conectividad combinó una comprobación funcional en el piloto con una campaña de simulación del tramo sensor–*gateway*. En terreno se verificó que los cambios de ocupación generados por el nodo ESP32 fueran recibidos por el receptor LoRa, publicados hacia el backend y reflejados en la interfaz. Esta comprobación confirma la integración del flujo de telemetría, mientras que la caracterización cuantitativa se realizó mediante simulación, evitando extrapolar sus resultados como mediciones directas del despliegue físico.

La campaña simuló enlaces LoRa P2P basados en módulos SX1276/78 y una topología tipo estrella para mensajes de 12 bytes, representativos de un identificador de plaza, una marca

temporal y el estado ocupado/libre. Se utilizó *LoRaEnergySim*, un marco de simulación para el análisis de nodos IoT y enlaces LoRa [**?**]. Se evaluaron tres condiciones: variación de distancia, incremento de nodos activos y aumento de la carga mediante intervalos de transmisión más reducidos. La Tabla 4.6 sintetiza los resultados más representativos.

Tabla 4.6: Resultados de la campaña de simulación del enlace LoRa P2P

Escenario simulado	**Configuración**	**Resultado observado**
Cobertura en función de distancia	Un nodo y un *gateway*; 10–100 m; SF7, SF9 y SF12.	A 100 m, la DER fue 0,8696 con SF7 y 1,0000 con SF9 y SF12. Los valores de SF más altos aumentaron la robustez del enlace.
Escalamiento de sensores	SF7; 10, 20 y 30 nodos con mensajes discretos.	La DER fue 1,0000, 0,9978 y 0,9809, respectivamente. La degradación fue gradual bajo tráfico de baja frecuencia.
Latencia bajo carga	Variación de intervalo entre transmisiones y *spreading factor*.	La latencia media de los paquetes recibidos se mantuvo cercana a 97,6 ms con SF7 y 233,5 ms con SF9. Al aumentar la carga, la degradación se manifestó principalmente en la DER y no en un alza pronunciada de la latencia media.

Los resultados de simulación respaldan la adecuación de LoRa P2P para transmitir mensajes breves y esporádicos de ocupación, donde la cobertura y el consumo son más relevantes que el caudal. Asimismo, muestran el compromiso entre robustez, tiempo en aire y energía: configuraciones de mayor *spreading factor* mejoran la entrega de paquetes, pero incrementan el costo de transmisión. No se realizó una comparación experimental equivalente con Wi-Fi; por lo tanto, la decisión se fundamenta en la adecuación de LoRa al perfil de telemetría y a las restricciones de cobertura del caso de uso, no en una demostración de superioridad general.

4.6.3. Pruebas de Integración y Campo (End-to-End)

Para verificar la integración del sistema completo se realizó un despliegue piloto en un acceso vehicular real automatizado. La comprobación se concentró en escenarios funcionales: vehículo autorizado, vehículo con patente no autorizada, video sin vehículo y presentación de una patente sostenida por una persona. En los casos de autorización vigente se observó el flujo de detección, consulta y apertura; en los escenarios sin vehículo o sin autorización no se accionó el portón. No se preservó una cantidad de repeticiones por escenario ni se evaluaron de forma controlada pérdida de conectividad, patentes proyectadas en pantalla o errores deliberados de OCR.

Escenario de Prueba

Se diseñó una arquitectura de doble cámara para separar las funciones de operación humana y detección automática. La cámara de detección se instaló en el punto de acceso y se conectó directamente a una Raspberry Pi 4 encargada de ejecutar el script `client.py`, capturar video y enviar las peticiones de validación al backend. En paralelo, una cámara IP independiente

se integró en la interfaz web para brindar confirmación visual al conserje sin interferir en el procesamiento de IA. Sobre esta configuración se realizaron pruebas de ingreso con vehículos registrados y no registrados.

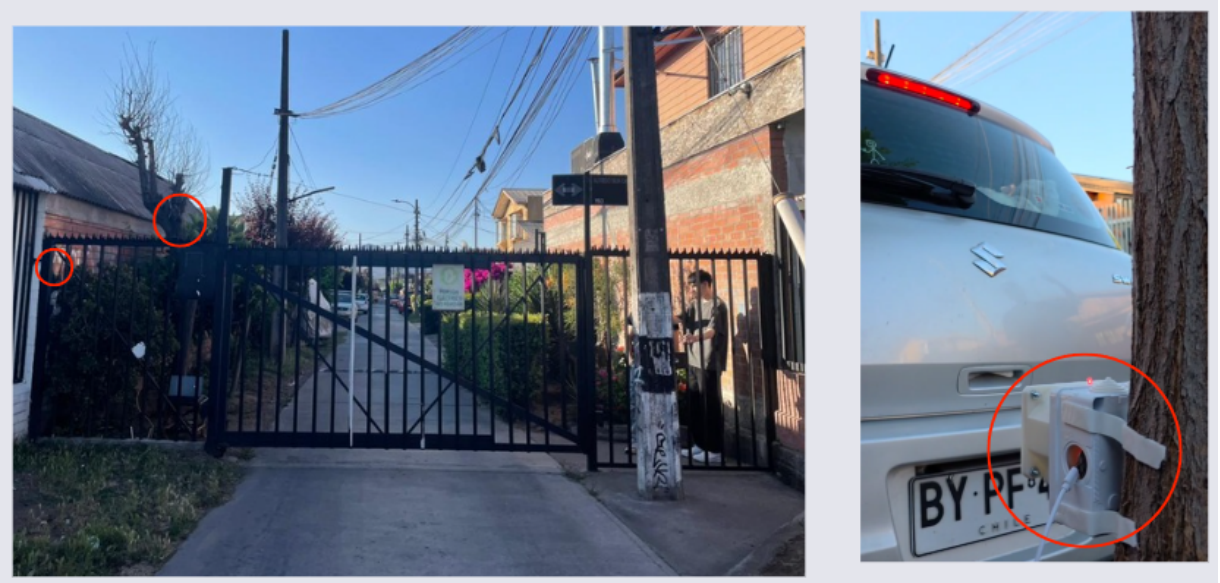


Figura 4.9: Montaje experimental del nodo de control en el acceso vehicular.

Resultados del Piloto

El piloto confirmó la ejecución del flujo detección de vehículo, lectura de patente, verificación en backend, activación del relé y registro del evento. La observación se realizó de forma funcional y no mediante una campaña cronometrada; por ello, no se informan promedios, desviación estándar, percentiles ni un tiempo total de apertura. En particular, no se definió una marca inicial y final homogénea para descomponer captura, preprocesamiento, comunicación, inferencia, consulta a la base de datos y actuación.

La evidencia del piloto permite sostener la factibilidad de la integración bajo los escenarios ejercitados, pero no una garantía de latencia para un despliegue productivo. La caracterización temporal del flujo completo requiere instrumentación por etapa y repeticiones controladas.

4.6.4. Retroalimentación Cualitativa de Usuarios

Durante la integración se recibieron observaciones informales de personas que interactuaron con las interfaces de conserjería y residentes. No se aplicó un instrumento, no se registró el número ni perfil de participantes y tampoco se levantaron métricas de satisfacción, tiempo de tarea o error. En consecuencia, estas observaciones no se reportan como resultados de usabilidad; su único aporte fue orientar ajustes visuales y de flujo durante el desarrollo.

4.7. Discusión de Resultados y Limitaciones

En conjunto, los resultados obtenidos muestran que el enfoque propuesto es técnicamente factible y que la incorporación de mecanismos de seguridad por capas aporta valor frente a soluciones centradas exclusivamente en detección visual o administración comunitaria. En comparación con

varios trabajos revisados en el Capítulo 2, esta memoria no se limita a reportar desempeño de visión por computador, sino que articula ese desempeño con autenticación, trazabilidad y criterios de segmentación entre subsistemas.

No obstante, también persisten brechas relevantes. La validación del OCR sigue siendo principalmente cualitativa; la resistencia frente a suplantación visual requiere pruebas anti-spoofing más específicas; y la evaluación de red e IoT, aunque cuenta con resultados de simulación y comprobación funcional en el piloto, todavía necesita mediciones controladas en terreno, una comparación con tecnologías alternativas y pruebas de escenarios de falla. Por ello, los resultados deben interpretarse como evidencia de factibilidad técnica y de mejora de robustez, más que como demostración definitiva de seguridad integral en un despliegue productivo.

4.7.1. Análisis Crítico de Seguridad

Desde la perspectiva de esta memoria, resulta fundamental evaluar no solo la funcionalidad alcanzada, sino también los vectores de ataque introducidos por la digitalización del control de acceso.

Mitigación de Suplantación (Spoofing)

Uno de los riesgos más comunes en sistemas ALPR es la presentación de patentes falsas (impresas o en pantallas). El sistema *ResiPark* aborda este riesgo mediante una validación contextual por etapas:

1. **Detección de contexto vehicular:** El modelo YOLO debe confirmar primero la presencia de un vehículo antes de proceder a la detección de la patente, reduciendo la dependencia de una única evidencia textual.

2. **Validación contextual:** el sistema verifica que la lectura de la patente ocurra después de detectar la presencia de un vehículo y contrasta la patente reconocida con los registros de autorización del backend. Este encadenamiento reduce la posibilidad de activar el flujo mediante la presentación aislada de una patente; sin embargo, el detector final utiliza una única clase vehicular (*car*), por lo que no permite comprobar la correspondencia entre distintos tipos de vehículo.

Seguridad en la Red y Dispositivos IoT

La incorporación de dispositivos IoT introduce nuevos vectores de ataque que fueron considerados durante el diseño del sistema.

Se identificó el riesgo de ataques *Man-in-the-Middle* en la red local. La segmentación por VLAN, el filtrado de tráfico, la protección frente a suplantación ARP y la autenticación robusta entre dispositivos se mantienen como recomendaciones de despliegue productivo, pues no fueron configurados ni validados en el piloto. El uso de canales cifrados y de autenticación entre extremos constituye una condición necesaria para abordar este riesgo en una siguiente iteración.

Como mejora futura para un despliegue productivo, se propone fortalecer la autenticación entre el nodo de borde y el backend mediante *Mutual TLS (mTLS)*, asegurando identidad criptográfica de ambos extremos.

4.7.2. Limitaciones del Prototipo

Si bien el sistema demostró un alto desempeño durante las pruebas piloto, se identificaron limitaciones propias del alcance del prototipo:

- **Dependencia de conectividad:** La validación de accesos depende actualmente de la conexión a internet. Una mejora futura es incorporar una operación degradada, limitada a autorizaciones previamente verificadas y vigentes, almacenadas localmente con expiración breve. Los eventos generados durante la desconexión deberían quedar registrados para sincronizarse al recuperar conectividad; las solicitudes nuevas, ambiguas o fuera de la caché deberían derivarse a revisión manual.
- **Condiciones de iluminación extrema:** Las pruebas se realizaron mayoritariamente en condiciones diurnas. Escenarios de contraluz severo o operación nocturna sin iluminación IR dedicada no fueron exhaustivamente evaluados.
- **Hardware de prototipado:** El uso de Raspberry Pi con almacenamiento en tarjeta SD puede presentar riesgos ante cortes de energía. Para un producto industrial se recomienda migrar a plataformas con almacenamiento eMMC.

4.7.3. Proyección y Transferencia Tecnológica

El sistema *ResiPark* fue concebido para responder a una necesidad real del mercado inmobiliario. Actualmente existe interés concreto por parte de la **Inmobiliaria Norte Verde** para una potencial implementación en proyectos residenciales.

Esta vinculación con la industria valida la propuesta de valor del sistema y establece una hoja de ruta clara para su escalamiento hacia entornos reales con usuarios finales.

Capítulo 5

Conclusiones y Trabajo Futuro

Esta memoria permitió diseñar e implementar un prototipo de control de acceso vehicular residencial que integra percepción visual, dispositivos IoT, una plataforma web y mecanismos de seguridad en la lógica de autorización. Bajo las condiciones funcionales evaluadas, el trabajo muestra la factibilidad de articular estos componentes para apoyar la gestión de accesos y estacionamientos. Su alcance corresponde a un prototipo: los resultados no permiten afirmar una seguridad absoluta ni extrapolar el desempeño a todo escenario residencial sin una validación adicional.

Identificación de amenazas y requisitos de seguridad

El primer objetivo específico se abordó mediante la definición de activos, puntos de entrada y amenazas relevantes para el contexto residencial. Este análisis permitió vincular riesgos de suplantación visual, abuso de API, intercepción de comunicaciones y manipulación de dispositivos con controles concretos de la arquitectura. Su principal resultado fue establecer una base de decisión para que la autorización no dependa exclusivamente de una lectura de patente ni de un único componente del sistema.

Diseño de la plataforma y controles de aplicación

Respecto del segundo objetivo, se implementó una capa de backend con validación de entradas, autenticación basada en JWT, reglas de autorización y registro de eventos. Estos mecanismos centralizan la decisión de acceso y entregan trazabilidad para la revisión posterior de eventos. Las pruebas sobre el backend móvil confirmaron el rechazo de credenciales inválidas, tokens ausentes, alterados y expirados, además de la restricción de modificación de reservas entre departamentos. La cobertura sigue siendo acotada: se requieren pruebas adicionales de inyección, revocación, control de tasa y protección uniforme de las rutas administrativas web antes de un despliegue productivo.

Integración de visión por computador en la autorización

El tercer objetivo se materializó en un flujo que combina detección vehicular, localización y reconocimiento de patente, y validación contextual de la información registrada. Sobre sus respectivas particiones de validación, los detectores finales alcanzaron mAP50 de 0,995 para vehículos y 0,977 para patentes. Las pruebas funcionales realizadas durante el piloto permitieron observar el comportamiento del sistema ante presencia de vehículo, ausencia de vehículo, patente no registrada y presentación aislada de una patente. En consecuencia, la autorización utiliza

más de una evidencia y la detección vehicular previa al OCR aporta una mitigación parcial ante una patente aislada; no obstante, no se implementó un mecanismo completo de *liveness* o anti-spoofing capaz de descartar fotografías, pantallas u otros ataques de presentación en todos los casos.

Factibilidad técnica de la integración IoT

El cuarto objetivo se evaluó mediante la integración del nodo de borde, el servicio remoto de inferencia, el backend, la plataforma web y el mecanismo de actuación del portón. La prueba de campo verificó de forma funcional el ciclo de detección, consulta de autorización y activación ante un vehículo autorizado, así como el rechazo de escenarios no autorizados considerados en el piloto. El Capítulo 4 presenta tanto las métricas de los modelos como una caracterización simulada del enlace LoRa; sin embargo, no se realizó una campaña instrumental en terreno que permita concluir valores estadísticos de latencia extremo a extremo, disponibilidad o pérdida de paquetes.

En la capa de comunicaciones se utilizó HTTPS para la interacción con la API. La segmentación mediante VLAN, las reglas adicionales de filtrado, la autenticación robusta de dispositivos IoT y la configuración de RLS se mantienen como medidas recomendadas para una siguiente iteración, puesto que no formaron parte de la configuración final validada del prototipo.

En síntesis, el aporte diferenciador de esta memoria consiste en organizar la automatización del acceso vehicular como un problema ciberfísico: la percepción visual, la comunicación, la autorización y la trazabilidad se analizan de manera integrada. Frente a soluciones residenciales centradas principalmente en administración y a trabajos que tratan visión, IoT o seguridad por separado, el prototipo propone una integración técnica orientada a reducir riesgos identificados en el acceso físico. Esta contribución demuestra factibilidad bajo las condiciones del piloto y delimita, al mismo tiempo, las validaciones de seguridad y despliegue que aún son necesarias para evolucionar hacia una operación productiva.

5.1. Trabajo Futuro

Las líneas de trabajo futuro se definen a partir de las principales limitaciones identificadas durante el desarrollo y la validación del prototipo:

- **Evaluación en condiciones ambientales adversas:** dado que las pruebas se realizaron principalmente en condiciones diurnas, resulta necesario extender la validación a operación nocturna, contraluz, lluvia o baja visibilidad, con el fin de determinar el comportamiento real del sistema fuera del escenario dominante del piloto.

- **Fortalecimiento frente a suplantación:** aunque la validación contextual por etapas aumenta la robustez frente a una patente aislada, aún se requiere incorporar y evaluar mecanismos más específicos de anti-spoofing o liveness para enfrentar fotografías, pantallas u otros ataques de presentación.

- **Autenticación robusta de dispositivos IoT y nodos de borde:** una limitación actual es que la identidad de los dispositivos no se apoya en mecanismos criptográficos fuertes. Por ello, una evolución natural consiste en integrar autenticación mutua mediante mTLS, credenciales individuales rotables o raíces de confianza hardware, de modo que el backend pueda identificar al dispositivo antes de aceptar telemetría o emitir una orden de apertura.

- **Operación ante pérdida de conectividad:** como el piloto depende de conectividad con backend para validar accesos, una línea futura importante es incorporar una política local limitada. Esta política debería autorizar únicamente vehículos previamente verificados y aún vigentes, registrar cada decisión para su sincronización posterior y derivar casos nuevos o ambiguos a revisión manual.
- **Validación prolongada y con más usuarios:** finalmente, se requiere una evaluación longitudinal con un mayor número de vehículos, usuarios y jornadas de operación, de manera de medir estabilidad, carga operativa, errores poco frecuentes y percepción de uso en condiciones más cercanas a un despliegue real.

Bibliografía

[1] P. Franco, J. M. Martinez, et al., "Cyber-physical approach for home energy management systems: Architecture, communications and data analytics," Sustainability, vol. 14, no. 8, p. 4639, 2022.

[2] P. Franco, F. Condon, J. M. Martinez, et al., "Smart energy data with activity recognition (sedar): An edge-iot platform for in-home monitoring," Sensors, vol. 23, no. 18, p. 7936, 2023.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, vol. 25, 2012.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[6] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, 2014.

[7] F. Li, H. Yang, X. Gao, and H. Han, "Towards iot-based sustainable digital communities," Intelligent and Converged Networks, vol. 3, no. 2, pp. 190–203, 2022.

[8] Q. Liu, Y. Zeng, and X. Zhou, "Design and implementation of smart community information management system," in 2023 International Seminar on Computer Science and Engineering Technology (SCSET), pp. 144–147, IEEE, 2023.

[9] N. Xue, L. Liang, J. Zhang, and X. Huang, "An access control system for intelligent buildings," in MobiMedia 2016: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, pp. 11–17, EAI, 2016.

[10] S. Bandara, T. Yashiro, N. Koshizuka, and K. Sakamura, "Access control framework for api-enabled devices in smart buildings," in 2016 22nd Asia-Pacific Conference on Communications (APCC), pp. 210–217, IEEE, 2016.

[11] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," Computer Networks, vol. 76, pp. 146–164, 2015.

[12] B. Reddy, J. Raj, and V. V. Aroulanandam, "Smartpark: Ai and iot-enabled automated parking system for urban efficiency," in 2024 International Conference on Sustainable Communication Networks and Application (ICSCNA), pp. 1–6, IEEE, 2024.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in International Conference on Learning Representations, 2015.

[14] R. Laroca, L. Zanlorensi, G. Gonçalves, et al., "A robust real-time automatic license plate recognition based on the yolo detector," Pattern Recognition Letters, vol. 138, pp. 37–43, 2021.

[15] S. Silva and C. Jung, "Real-time license plate detection and recognition using deep learning," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 11, pp. 3491–3500, 2018.

[16] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," IOP Conference Series: Materials Science and Engineering, vol. 1084, no. 1, p. 012027, 2021.

[17] M. Z. Abidin and R. Pulungan, "A systematic review of machine-vision-based smart parking systems," Scientific Journal of Informatics, vol. 7, no. 2, pp. 213–228, 2020.

[18] M. Dixit, C. Srimathi, R. Doss, S. Loke, and M. Saleemdurai, "Smart parking with computer vision and iot technology," in 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), pp. 170–174, IEEE, 2020.

[19] C. S. A. of Singapore, "Guide on conducting threat identification and assessing effectiveness of controls for smart buildings." Government Guide, mar 2026.

[20] J. J. Barriga, J. Sulca, J. León, A. Ulloa, D. Portero, J. García, and S. G. Yoo, "A smart parking solution architecture based on lorawan and kubernetes," Applied Sciences, vol. 10, no. 13, 2020.

[21] Edifito, "Edifito: Software para la administración de edificios y condominios." Sitio web oficial, 2026. Consultado: 2026-09-02.

[22] ComunidadFeliz, "Software para la administración de edificios y condominios." Sitio web oficial, 2026. Consultado: 2026-09-02.

[23] M. Alsultan, T. Alghonaim, A. Alorf, B. Alwazzan, F. Alsakakir, A. Alhassan, and Y. Hussain, "Tri-gate: A tri-modal anti-spoofing system for gate access using vehicle, license plate, and face recognition," International Journal of Advanced Computer Science and Applications, vol. 17, no. 1, 2026.

[24] W. Xiong, L. Cao, D. Yan, Y. Jiang, G. Zhang, Y. Wang, X. Huang, and L. Liu, "License plate recognition methodology in complex scenarios based on CSCM-YOLOv8 and CSM-LPRNet," PLOS ONE, vol. 21, p. e0339649, Jan. 2026. PMID: 41481637; PMCID: PMC12758793.

[25] H. Silva et al., "Efficient video-based alpr system using yolo and visual rhythm," arXiv preprint arXiv:2501.02270, jan 2025.

[26] C. B. International, "Iot security risks: Why smart buildings are hacker targets." Security Whitepaper, sep 2025.

[27] D. O. Security, "What is stateless authentication?." Security Wiki, 2025.

[28] B. Dayal, "Jwt authentication end-to-end: Token generation, validation, jwks." Medium, jan 2026.

[29] IBM, "What is api authentication? transport layer security and stateless implementations." IBM Think, feb 2026. Accessed: 2026-04-01.

[30] M. A. Ala'anzy, A. Abilakim, R. Zhanuzak, and L. Li, "Real time smart parking system based on IoT and fog computing evaluated through a practical case study," Scientific Reports, vol. 15, p. 33483, Sept. 2025.

[31] E. Ferruzola-Gomez, N. Beltrán Robayo, M. Barrera Rea, and J. Barros Naranjo, "Cyber-physical systems in the smart home: A review," International Journal of Computational Innovations, Intelligent Systems and AI, vol. 2, pp. 92–121, Jan. 2026.

[32] Fortinet, "What is cyber-physical systems (cps)? examples, applications and security." Fortinet Cyber Glossary, 2024. Accessed: 2026-04-01.

[33] Z. Selmi, M. B. Halima, and A. M. Alimi, "Deep learning system for automatic license plate detection and recognition," in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1132–1138, IEEE, 2017.

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[35] N. Gonthina, S. Katkam, and R. T. Pusuluri, "Parking slot detection using yolov8," in 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC), pp. 1–6, IEEE, 2023.

[36] N. Kharina, "Software for car license plates recognition with minimal computing resources," in 2022 24th International Conference on Digital Signal Processing and its Applications (DSPA), pp. 1–4, IEEE, 2022.

[37] W. Shi and S. Dustdar, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[38] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[39] D. Hardt, "The oauth 2.0 authorization framework." RFC 6749, 2012.

[40] Actility, "Understanding lorawan architecture and network server capabilities." Actility Tech Reports, 2024. Accessed: 2026-04-01.

[41] Mainlink, "Lora vs lorawan: Key differences explained for physical and mac layers." Mainlink IoT Documentation, jan 2026. Accessed: 2026-04-01.

[42] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4818–4829, 2024.

[43] B. Aziz, D. Arellanes, et al., "Trusted platform module (tpm) for iot security: A survey," Journal of Information Security and Applications, vol. 47, pp. 57–67, 2019.

[44] FastAPI, "Fastapi features and dependency injection." Documentación oficial, 2026. Consultado: 2026-09-02.

[45] Pallets Projects, "Flask documentation." Documentación oficial, 2026. Consultado: 2026-09-02.

[46] Django Software Foundation, "Django documentation." Documentación oficial, 2026. Consultado: 2026-09-02.

[47] huang77, "huaung77 object detection dataset." Roboflow Universe, 2024. Consultado: 2026-09-21.

[48] Roboflow Universe Projects, "License plate recognition dataset." Roboflow Universe, 2025. Consultado: 2026-09-21.