

UNIVERSIDAD DE TARAPACÁ

FACULTAD DE INGENIERÍA



Departamento de Ingeniería en Computación e
Informática



Informe III

“EstacionaFlash”

**Autores: Álvaro Guarachi
Rafael Nakata
Erik Correa
Ángel Cabezas**

**Asignatura: Proyecto II
Profesor: Diego Aracena**

ARICA, 24/11/2025

Tabla I. Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
15/10/2025	1.0	Versión preliminar Informe I	Álvaro Guarachi Rafael Nakata
19/10/2025	1.1	Avance del informe I	Álvaro Guarachi
21/10/2025	1.2	Avance del informe I	Rafael Nakata Álvaro Guarachi
26/10/2025	1.3	Avance del informe I	Álvaro Guarachi
27/10/2025	2.0	Término informe I	Álvaro Guarachi Rafael Nakata Erik Correa Ángel Cabezas
22/11/2025	2.1	Corrección informe I: objetivos específicos	Álvaro Guarachi
23/11/2025	2.2	Corrección Informe I: Carta gantt, adición de índice de tablas y figuras.	Álvaro Guarachi
23/11/2025	2.3	Avance informe II	Álvaro Guarachi Rafael Nakata Erik Correa
24/12/2025	3.0	Término informe final	Álvaro Guarachi Erik Correa Rafael Nakata Ángel Cabezas

Índice

Índice	3
Índice de tablas	4
Índice de Ilustraciones	4
Panorama General	4
Resumen del Proyecto	5
Propósito, alcance, objetivos	5
Suposiciones y restricciones	6
Entregables del Proyecto	7
Organización del Proyecto	8
Mecanismos de Comunicación	9
Planificación de los procesos de gestión	10
Planificación inicial del proyecto	10
Planificación de estimaciones	10
Planificación de Recursos Humanos	12
Lista de actividades (carta Gantt)	13
Actividades de trabajo	13
Asignación de tiempo	14
Planificación de la gestión de riesgos	15
Especificación de Requerimientos	18
Requerimientos funcionales	18
Requerimientos no funcionales	19
Modelo de Diseño (Caso de uso general)	20
Diagramas de secuencias	27
Diagrama de clases del sistema	33
Descripción de la Arquitectura (vista del modelo diseño).	33
Documento de Diseño de Interfaz Usuario	34
Implementación	35
Plan de Integración	35
Módulos implementados	35
Relacionados con la interfaz	35
Relacionados con los sensores	35
Descripción de la Arquitectura (vista desde los módulos del caso de uso)	35
Modelo de Implementación	35
Reporte de Revisión	36
Aspectos generales	36
Avance de acuerdo a la Carta Gantt	36
Problemas Encontrados - Soluciones Propuestas	38
Trabajo Futuro	39
Conclusión	40
Referencias	41

Índice de tablas

Tabla I. Historial de Cambios	2
Tabla II. Personal y entidades internas	8
Tabla III. Roles y responsabilidades	8
Tabla IV. Costos de Software	10
Tabla V. Costos de Hardware	11
Tabla VI. Costo por hora de trabajo	12
Tabla VII. Costo por rol	12
Tabla VIII. Riesgos del proyecto	15
Tabla IX. requerimientos funcionales	18
Tabla X. requerimientos no funcionales	19
Tabla XI. caso de uso: Detectar vehículo	21
Tabla XII. caso de uso: Gestionar Ingreso Vehicular	22
Tabla XIII. caso de uso: Gestionar Salida Vehicular	23
Tabla XIV. caso de uso: Accionar barrera	24
Tabla XV. caso de uso: Actualizar disponibilidad	25
Tabla XVI. caso de uso: Detectar movimiento	26
Tabla XVII. Encargados módulos de la interfaz	35
Tabla XVIII. Encargados módulos de los sensores	36
Tabla XIX. Fase 1: Investigación y Prototipo Inicial	36
Tabla XX. Fase 2: Planificación y Definición	37
Tabla XXI. Fase 3: Desarrollo e implementación	37
Tabla XXII. Fase 4: Integración y Producto Final	38

Índice de Ilustraciones

Ilustración 1. Carta Gantt del proyecto	15
Ilustración 2. Diagrama de casos de uso	21
Ilustración 3: Diagrama de secuencia: detectar vehículo	28
Ilustración 4: Diagrama de secuencia: gestionar ingreso vehicular	29
Ilustración 5: Diagrama de secuencia: Gestionar salida vehicular	30
Ilustración 6: Diagrama de secuencia: Accionar barrera	31
Ilustración 7: Diagrama de secuencia: Actualizar disponibilidad	32
Ilustración 8: Diagrama de secuencia: Controlar iluminación	33
Ilustración 9. Diagrama de clases	34
Ilustración 10: Descripción de la arquitectura del proyecto (cliente-servidor)	34
Ilustración 11: Interfaz de usuario	35

Panorama General

Resumen del Proyecto

Propósito, alcance, objetivos

Propósito: El proyecto permitirá construir un software para la automatización de estacionamientos tales como supermercados, centros comerciales y malls.

Definición del Problema: La gestión actual de estacionamientos comerciales es ineficiente debido a su dependencia de controles manuales y tickets físicos. Esto genera tres problemas críticos:

1. **Congestión:** Cuellos de botella en accesos por tiempos lentos de validación.
2. **Desinformación:** Falta de datos en tiempo real sobre cupos disponibles para usuarios y administradores.
3. **Costos Elevados:** Gasto excesivo en personal para operación de barreras y resolución de incidencias.

Alcance: El proyecto comprende el desarrollo de un sistema integral de automatización que gestiona el control de acceso vehicular mediante reconocimiento de patentes(LPR) y el accionamiento de barreras físicas. El alcance incluye la detección de ocupación de sectores en tiempo real mediante sensores ultrasónicos, la visualización de disponibilidad a través de paneles LED, la gestión eficiente de iluminación mediante detección de movimiento y la validación lógica de pagos vinculada a la patente del vehículo.

Objetivo general: Desarrollar un sistema de automatización para estacionamientos comerciales basado en arquitectura IoT y visión artificial (LPR) sobre una Raspberry Pi 4B, con el fin de optimizar el control de acceso vehicular, la gestión de pagos y el monitoreo de disponibilidad en tiempo real.

Objetivos específicos:

- **Diseñar** la arquitectura integral de hardware y software del sistema "EstacionaFlash", definiendo los diagramas de conexión y la lógica de comunicación entre el Raspberry Pi 4B, los sensores y adicionales para garantizar el flujo de información en tiempo real.
- **Implementar** los algoritmos de visión artificial y lógica de control en el entorno del Raspberry Pi, desarrollando el reconocimiento de patentes y la gestión de señales de los sensores ultrasónicos para la automatización del acceso.
- **Integrar** los subsistemas físicos con la interfaz de software, sincronizando la detección vehicular con la respuesta de los actuadores (barreras) para lograr una operación autónoma.
- **Validar** el desempeño técnico mediante pruebas funcionales en escenarios

simulados, verificando el cumplimiento de métricas críticas como la latencia de respuesta y la precisión en la lectura de patentes.

- **Garantizar** la mantenibilidad y operatividad del sistema mediante la generación de manuales técnicos y de despliegue estandarizados, asegurando la escalabilidad futura del proyecto.

Suposiciones y restricciones

Suposiciones:

- **Sobre los Recursos:** Se supone que los recursos de hardware necesarios (como las cámaras, sensores ultrasónicos, motores de barrera y el Raspberry Pi) estarán disponibles a tiempo para el desarrollo e implementación del sistema.
- **Sobre la Experiencia:** Se espera que el equipo de desarrollo tenga la experiencia necesaria en la integración de hardware (cámaras, sensores, motores y Raspberry Pi) y el desarrollo de software en Python (incluyendo librerías de visión artificial) para completar el proyecto.
- **Sobre la validación :** Se asume que tanto los conductores (usuarios finales) como los administradores del estacionamiento estarán dispuestos a utilizar el sistema de automatización basado en LPR, y que lo encontrarán intuitivo y más eficiente que un sistema de tickets tradicional.
- **Sobre la Infraestructura del Cliente:** Se supone que la administración del mall, supermercado u otra institución proporcionará el acceso necesario a los espacios físicos (accesos, pasillos) donde se instalará el hardware, así como la infraestructura de red y electricidad requerida.
- **Sobre la Compatibilidad:** Se asume que los sensores , las cámaras, los motores, el Panel LED y el software desarrollado serán compatibles entre sí y con el sistema Raspberry Pi, garantizando un funcionamiento eficiente del sistema de automatización.
- **Sobre las Pruebas:** Se asume que habrá tiempo y recursos suficientes dentro del cronograma para realizar pruebas exhaustivas del sistema, y así, realizar los ajustes necesarios antes de la implementación final.

Restricciones:

- Se espera utilizar cámaras, sensores (ultrasónicos, PIR), motores y una Raspberry Pi como los componentes hardware principales para la construcción del prototipo.
- El proyecto debe ser realizado en un plazo definido de 3 meses.
- Debe ser construido sin exceder el presupuesto total asignado en la

planificación.

- El sistema prototipo deberá gestionar y monitorear un número máximo de 4 plazas de estacionamiento de forma simultánea, limitado por el hardware definido.
- Se deberán implementar medidas de seguridad adecuadas para proteger los datos sensibles de los vehículos (números de patente, registros de entrada/salida), cumpliendo con las normativas vigentes de protección de datos (como la Ley de Protección de Datos Personales de Chile).

Entregables del Proyecto

1. Presentación del problemática de proyecto
2. Presentación de la solución de la problemática
3. Informe I del proyecto
4. Presentación informe I
5. Informe II del proyecto
6. Presentación informe II
7. Wiki de Proyecto
8. Carta Gantt
9. Bitácoras semanales.
10. Poster o afiche Promocional
11. Manual de usuario
12. Sistema “EstacionaFlash”

Organización del Proyecto

Tabla II. Personal y entidades internas

Se identifica al personal interno designado y se describe la función general de sus cargos dentro de la organización del proyecto

Tipo Jefe	Personal designado	Descripción
Jefe Analista	Álvaro Guarachi	El analista será quien deberá evaluar e investigar los requerimientos (ej. cámaras o sensores) y restricciones que tengan relación con el proyecto, documentando los hallazgos para el equipo.
Jefe Arquitecto	Erik Correa	El arquitecto se encargará de definir el diseño de la arquitectura del proyecto (diagramas de hardware, prototipos de interfaz), buscando que el sistema sea funcional y entregando las directrices para el programador.
Jefe implementación	Ángel Cabezas	El implementador se encargará de decidir la lógica del programa para el Raspberry Pi (integración de cámaras, sensores, motores), y también de encontrar y corregir fallos y posibles errores de software que surjan.
Jefe de Proyecto	Rafael Nakata	El jefe de proyecto se encargará de ir realizando consultas al equipo para evaluar cómo se va desarrollando el proyecto, gestionar el cronograma (Carta Gantt) y entregar una solución a los problemas y riesgos identificados.

Tabla III. Roles y responsabilidades

Se definen las responsabilidades técnicas y tareas específicas asignadas a cada rol operativo del equipo de desarrollo.

Rol	Descripción de Responsabilidades
Técnico de redes	Responsable de configurar y monitorear la red (Wi-Fi/Ethernet) que conecta la Raspberry Pi con las cámaras

	y el sistema central, garantizando una conectividad estable.
Documentador	Encargado de producir la documentación requerida del proyecto. Deberá describir el desarrollo mediante bitácoras (actas de reunión), informes de avance y un manual de usuario/técnico de manera adecuada.
Programador	Encargado de producir, mantener y regular el código Python en Raspberry Pi. Deberá depurar el código de las cámaras LPR y sensores para asegurar el correcto funcionamiento.
Técnico de hardware	Encargado de implementar los componentes electrónicos requeridos para el proyecto (LPR, sensores, motores) y realizar las pruebas necesarias para asegurar el correcto funcionamiento y evitar fallas.

Mecanismos de Comunicación

- **Medio principal (Whatsapp):** Para coordinación diaria, consultas rápidas y envío de material de referencia.
- **Google Drive:** Nube para el trabajo colaborativo (documentos, borradores) donde se suben avances que pueden estar sujetos a cambios.
- **Redmine:** Medio oficial para la gestión de tareas, control de versiones y almacenamiento de material final que no está sujeto a cambios (entregables formales).
- **Discord:** Para llamadas en línea y reuniones de equipo, en caso de que se requiera trabajar de forma remota.

Planificación de los procesos de gestión

Planificación inicial del proyecto

Planificación de estimaciones

Tabla IV. Costos de Software

Se detalla las herramientas de software seleccionadas para el desarrollo, indicando su propósito específico y condiciones de licencia.

Nombre	Uso	Condiciones de Uso
Visual Studio Code	Editor y depurador de código Python.	Libre de uso bajo los Términos de licencia de Microsoft.
Python	Lenguaje de programación principal para el Raspberry Pi.	Libre de uso (Open Source).
OpenCV	Librería para el procesamiento de imágenes y LPR en tiempo real.	Libre de uso (Open Source).
Google Drive	Almacenamiento de archivos y documentos colaborativos.	Proporcionado por la institución UTA.
Redmine	Organización de actividades y almacenamiento de documentación para ser accedido por el profesor.	Proporcionado por la institución UTA.
Discord	Comunicación por voz y texto para reuniones extensas.	Libre uso bajo los Términos de servicio de Discord.
Whatsapp	Comunicación por texto para mensajes breves.	Libre uso bajo las Condiciones de servicio de WhatsApp.
Balena Etcher	Grabador de sistema operativo para la Raspberry Pi.	Libre de uso (Open Source).

Tabla V. Costos de Hardware

Se detalla el presupuesto del equipamiento físico, la cual desglosa los componentes requeridos, cantidades y costos estimados de adquisición.

Cabe destacar que este presupuesto corresponde a la construcción de un **prototipo funcional a escala** para pruebas de validación. La implementación comercial final requeriría hardware de grado industrial cuyo costo sería superior al aquí detallado.

Nombre del componente	Cantidad	Precio aproximado
Raspberry Pi 4 B+ (placa controladora)	1	\$95.000
Sensor ultrasónico(detector de vehículo)	4	\$10.400
Pantalla LCD 16×2 (visualización de datos)	2	\$36.000
Fuente de alimentación (5V 3A)	1	\$12.000
Cables de conexión y protoboard	1 set	\$5.000
Carcasa para Raspberry Pi	1	\$2.490
Material para estructura de maqueta (cartón, acrílico o madera)	1 set	\$15.000
Sensores PIR (movimiento para luces) (opcional)	4	\$12.000
Servomotor (control de barrera) (opcional)	2	\$6.600
Cámara LPR	2	\$60.000
Laptop(programación lógica del sistema)	4	\$2.400.000
Total		\$2.654.490

Planificación de Recursos Humanos

Tabla VI. Costo por hora de trabajo

Se establece la valoración económica por hora de trabajo asignada a cada perfil técnico involucrado en el proyecto.

Roles	Costo/Hora	Personal
Programador	\$45000	Erik Correa
Técnico de Hardware	\$35000	Rafael Nakata
Técnico de Redes	\$35000	Angel Cabezas
Especialista en Documentación	\$25000	Álvaro Guarachi

Tabla VII. Costo por rol

Se presenta el presupuesto total de Recursos Humanos calculado en base a la carga horaria semanal y la duración de las fases.

Roles	Semanas	Horas Semanales	Horas Totales	Costo total
Programador	8	6 hrs./sem.	48	\$2.160.000
Técnico de Hardware	7	3 hrs./sem.	21	\$735.000
Técnico de Redes	8	4 hrs./sem.	32	\$1.120.000
Especialista en Documentación	12	2 hrs./sem.	24	\$600.000
Total			125	\$4.615.000

Lista de actividades (carta Gantt)

Actividades de trabajo

- **Selección de la idea del proyecto:** Se discutieron varias ideas mediante la metodología de lluvia de ideas y se seleccionó una, la cual fue para desarrollar el sistema de automatización de estacionamientos.
- **Análisis de la problemática:** Definición de los requisitos funcionales (gestión de accesos, monitoreo de sectores) y no funcionales (tiempo de respuesta, seguridad) del sistema. Se planteó una solución general a la problemática del proyecto.
- **Establecimiento de objetivos y roles del proyecto:** Definición de los objetivos generales y específicos (LPR, sensores ultrasónicos, etc.). Asignación de los roles de los integrantes del equipo (Jefe de Proyecto, Programador, Técnico Hardware, Técnico de Redes, Documentador).
- **Planificación del sistema de sensores y componentes electrónicos:** Identificación de los sensores (ultrasónicos, PIR), actuadores (motores) y cámaras (LPR) que se utilizarán en el proyecto con Raspberry Pi.
- **Establecimiento y corrección de riesgos del proyecto:** Identificación de los posibles riesgos relacionados con la implementación del sistema (compatibilidad de LPR, tiempos de entrega de hardware, etc.).
- **Preparación del Informe 1:** Elaboración de la documentación de planificación del proyecto (este informe).

Asignación de tiempo

Se presenta el cronograma de trabajo del proyecto (Carta Gantt), detallando la duración estimada de las fases de investigación, planificación, desarrollo e implementación, así como la interdependencia entre las actividades críticas."

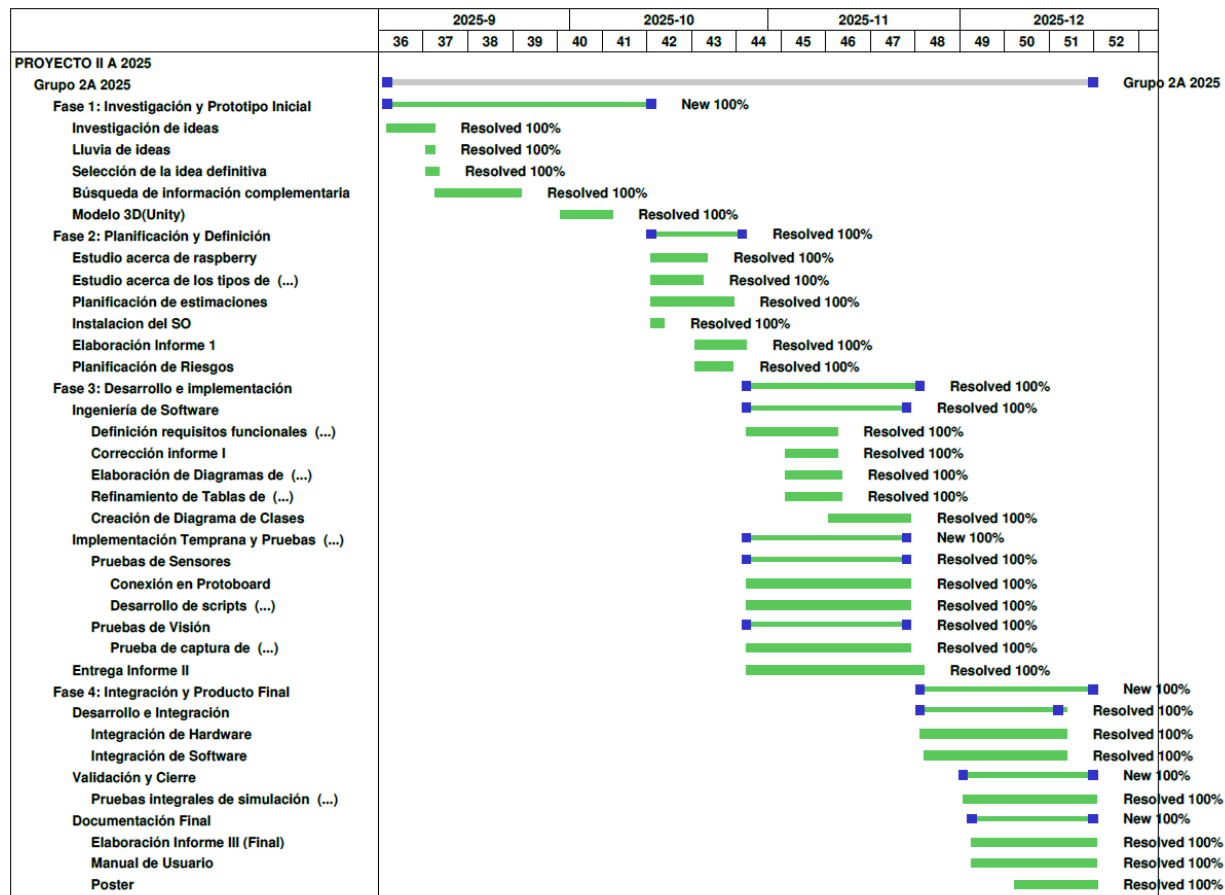


Ilustración 1. Carta Gantt del proyecto

Planificación de la gestión de riesgos

Estos son los riesgos identificados por parte del equipo que desarrollará el proyecto. Las categorías que existen son de la más grave a más leve:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

Tabla VIII. Riesgos del proyecto

Expone la matriz de gestión de riesgos, clasificándolos según su probabilidad de ocurrencia, nivel de impacto y las acciones remediales definidas.

Riesgos Posibles	Probabilidad de Ocurrencia	Nivel de Impacto	Acción Remedial
Integración LPR (Tiempo)	90%	1 (Catastrófico)	Validar librerías antes de comprar las cámaras.
Incompatibilidad de software(LPR) con Raspberry Pi	80%	1 (Catastrófico)	Investigar y realizar pruebas de concepto de las librerías (ej. OpenCV para cámara LPR) antes de la compra. Se buscarán proveedores alternativos con compatibilidad confirmada.
Daño físico del hardware	20%	1 (Catastrófico)	Informar al personal correspondiente (ayudante o profesor) sobre el daño al hardware para su renovación. Se debe tener hardware de respaldo (si es posible) o buscar equipos alternativos.

Cambios en los requisitos del proyecto	40%	2 (Crítico)	Se asigna un plazo para que el cliente (profesor) notifique de cambios. Se detiene el desarrollo del proyecto y se evalúa la integración de los nuevos requisitos.
Complicaciones del montaje (Sensores y motores)	50%	2 (Crítico)	Se buscará la forma de solventar la complicación, a través de información encontrada en línea (tutoriales, foros) o mediante la ayuda del profesor y/o ayudantes encargados.
Poca disponibilidad del equipo respecto a las reuniones de proyecto	50%	2(Crítico)	El Jefe de Proyecto debe realizar un "check-in" de disponibilidad al inicio de cada semana. Se debe asegurar que todo el código y documentación esté centralizado (Redmine/Drive) y no en equipos locales. Se reasignan tareas de la Carta Gantt para adelantar trabajo crítico en semanas de alta disponibilidad (ej. post-pruebas).
Falta de componentes durante el desarrollo	40%	2 (Crítico)	Se investigarán opciones alternativas o equivalentes. Se gestionará la compra de los componentes críticos con la mayor anticipación posible.
Sobrecarga de roles en personal	80%	2(Crítico)	Definir "backups" y centralizar todo el trabajo en Redmine y Google Drive.

Superación del presupuesto	50%	3 (Marginal)	Reasignar los recursos dentro del proyecto. Evaluar la reducción de costos de funcionalidades no esenciales (ej. reducir de 4 a 2 sensores PIR).
Documentación atrasada	15%	3(Marginal)	Establecer un horario fijo semanal, de carácter prioritario, para la actualización de la documentación.
Fallos en la conectividad (Wi-Fi/Red del RPi)	30%	4 (Despreciable)	Verificar la red local y restablecer la conexión. Si el problema persiste, cambiar a una conexión de respaldo (cable Ethernet) o usar un método offline hasta que se restablezca.

Especificación de Requerimientos

Requerimientos funcionales

Tabla IX. requerimientos funcionales

Número	Requerimiento	Descripción
RF1	Detección de Vehículo	El sistema debe detectar la presencia de un vehículo mediante sensores ultrasónicos en las entradas y salidas.
RF2	Captura y reconocimiento de patente	El sistema debe capturar la imagen de la patente utilizando cámaras y procesarla para obtener su número.
RF3	Cálculo y Validación de Tarifa	El sistema debe calcular el tiempo de estadía basado en la hora de entrada y verificar si la patente ha cubierto el pago correspondiente antes de autorizar la salida.
RF4	Control automático de barrera	El sistema debe enviar una señal al motor para abrir o cerrar la barrera solo si la validación fue exitosa.
RF5	Actualización de disponibilidad	El sistema debe actualizar la información en el panel LED indicando disponibilidad del sector del estacionamiento o estado del acceso.
RF6	Detectar movimiento	El sistema debe detectar movimientos mediante sensores PIR para encender o apagar las luces de los sectores.
RF7	Gestión de Estados Lógicos	El sistema debe validar la consistencia lógica durante el proceso de egreso, impidiendo el registro de una salida (y la apertura de barrera) a cualquier vehículo que no figure con el estado 'Dentro' o que carezca de un registro de entrada activo en la base de datos.

Requerimientos no funcionales

Tabla X. requerimientos no funcionales

Número	Requerimiento	Descripción
RFN1	Tiempo de respuesta	El sistema debe responder en tiempo real, con una latencia máxima de 2–3 segundos para la validación de la patente y la apertura de la barrera.
RFN2	Escalabilidad	El sistema debe permitir agregar más sensores o cámaras sin alterar su funcionamiento central.
RFN3	Seguridad	Los datos como registros de patentes y accesos deben mantenerse protegidos y no ser accesibles por usuarios no autorizados.
RFN4	Eficiencia energética	El uso de iluminación automatizada debe reducir el consumo energético cuando no se detecten vehículos.
RFN5	Mantenibilidad	El sistema debe estar documentado para facilitar su actualización o reparación, para facilitar su reparación o actualización.
RFN6	Fiabilidad	La detección y reconocimiento de patentes tendrá una precisión mínima del 90% en condiciones normales de iluminación, para garantizar un funcionamiento confiable.

Modelo de Diseño (Caso de uso general)

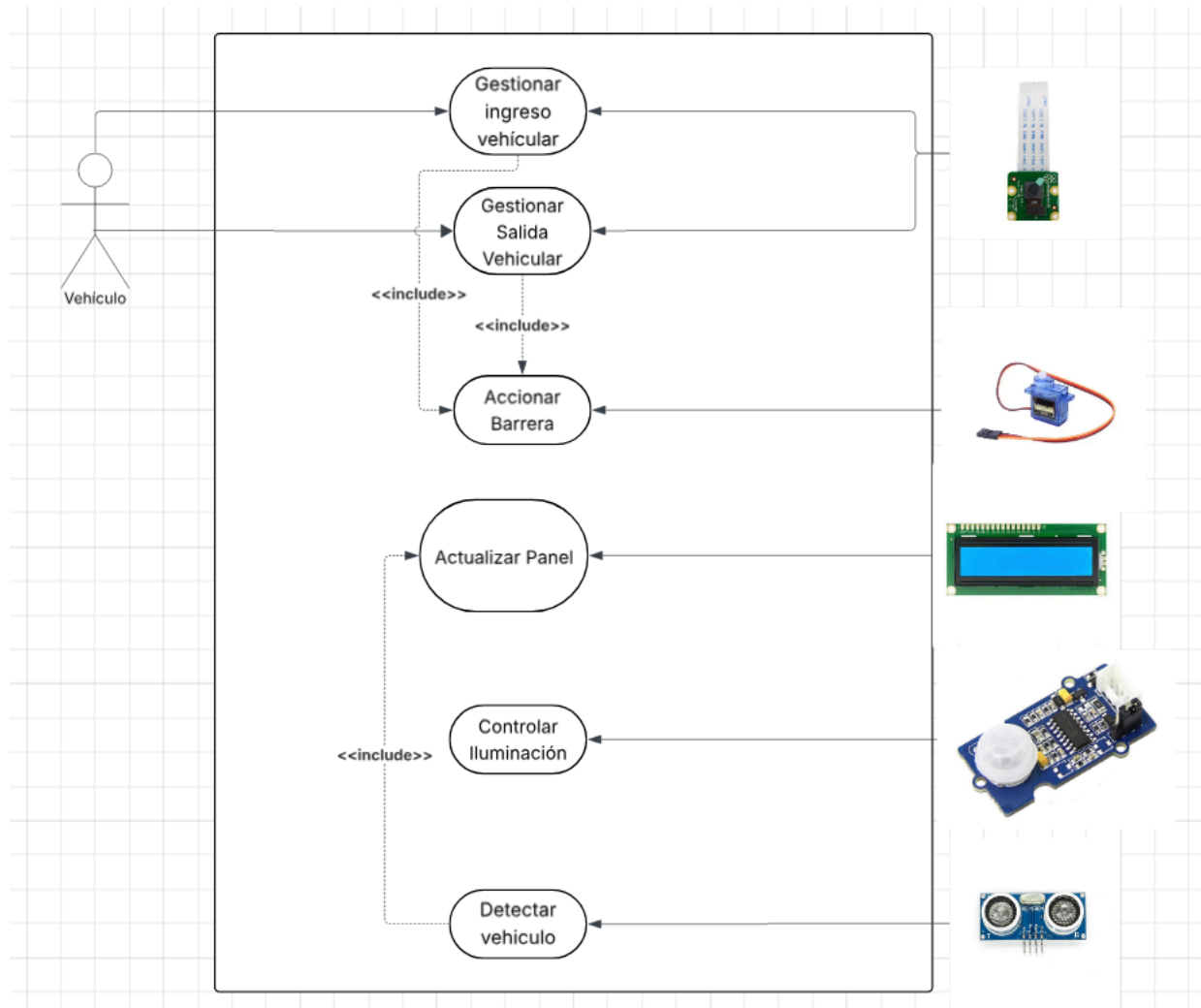


Ilustración 2. Diagrama de casos de uso

Tabla XI. caso de uso: Detectar vehículo

ID: CU-001	
Nombre del CU: Detectar Vehículo	
Actor(es): Sensor Ultrasónico	
Requisitos Relacionados: RF01, RF05	
Descripción: El sensor ultrasónico detecta el paso de un vehículo por un sector y el sistema actualiza el contador lógico de plazas disponibles.	
Precondiciones: El sistema está encendido y los sensores calibrados.	
Actor(Sensor ultrasónico)	Sistema
Flujo Principal: 1. Detecta la presencia de un vehículo a una distancia menor al umbral configurado. 2.Envía una señal de detección al sistema.	Flujo Principal: 3.Recibe la señal y confirma la presencia del vehículo. 4. Determina si la señal corresponde a una Entrada o Salida. 5. Incrementa (Entrada) o Decrementa (Salida) el contador de ocupación en la Base de Datos. 6. <<include>> CU-005 (Actualizar Disponibilidad).
Flujo Alternativo:	Flujo Alternativo: 3.1 La señal dura menos de 1 seg. El sistema la descarta como ruido.
Postcondiciones: La base de datos refleja la cantidad real de vehículos dentro.	
CU Relacionados : CU-005	

Tabla XII. caso de uso: Gestionar Ingreso Vehicular

ID: CU-002	
Nombre del CU: Gestionar Ingreso Vehicular	
Actor(es): Vehículo, Cámara, Servomotor	
Requisitos Relacionados: RF2, RF4	
Descripción: Gestiona el proceso completo de entrada: detecta el vehículo, lee la patente, valida disponibilidad, registra la entrada y abre la barrera.	
Precondiciones: El estacionamiento tiene plazas disponibles y la cámara está operativa.	
Actor(Conductor/Cámara)	Sistema
Flujo Principal: 1. El conductor posiciona el vehículo frente a la barrera. 2. La cámara detecta presencia y captura imagen.	Flujo Principal: 3. Procesa la imagen (LPR) para obtener la cadena de texto de la patente. 4. Verifica disponibilidad de estacionamiento > 0. 5. Registra la patente y hora de entrada en BD. 6. <<include>> CU-004 (Accionar Barrera - Abrir).
Flujo Alternativo:	Flujo Alternativo: 3.1. No se reconoce la patente. El sistema no acciona la barrera y alerta del error. 4.1. No se permite el ingreso hasta que haya disponibilidad en el estacionamiento.
Postcondiciones: Se crea un registro de sesión abierto para el vehículo.	
CU Relacionados : CU-004 (Accionar barrera).	

Tabla XIII. caso de uso: Gestionar Salida Vehicular

ID: CU-003	
Nombre del CU: Gestionar Salida Vehicular	
Actor(es): Vehiculo, Cámara, Servomotor	
Requisitos Relacionados: RF2, RF3, RF4, RF7	
Descripción: El sistema verifica que la patente capturada tenga su deuda saldada antes de liberar el paso.	
Precondiciones: La patente del vehículo debe estar en la base de datos del sistema.	
Actor(Cámara)	Sistema
Flujo Principal: 1. El conductor posiciona el vehículo frente a la barrera. 2. Captura imagen del vehículo en zona de salida.	Flujo Principal: 3. Procesa imagen (LPR) y obtiene patente. 4. Busca el registro de entrada activo para esa patente. 5. Calcula el tiempo total y verifica el estado del pago en BD. 6. Si estado == "PAGADO", cierra el ticket en la BD. 7. <<include>> CU-004 (Accionar Barrera - Abrir).
Flujo Alternativo:	Flujo Alternativo: 4.1 Vehículo no tiene entrada registrada (posible intrusión). Alerta seguridad. 5.1 El sistema detecta estado "IMPAGO". Muestra monto a pagar en pantalla y mantiene barrera cerrada.
Postcondiciones: El vehículo es autorizado a salir y se cierra su ciclo de facturación.	
CU Relacionados : CU-004 (Accionar barrera).	

Tabla XIV. caso de uso: Accionar barrera

ID: CU-004	
Nombre del CU: Accionar barrera	
Actor(es): Servomotor	
Requisitos Relacionados: RF04	
Descripción: Envía señales físicas al servomotor para permitir el paso y cerrarse automáticamente.	
Precondiciones: El sistema debe haber validado la entrada o el pago para la salida exitosamente.	
Actor (Servomotor)	Sistema
Flujo Principal: 3.El motor recibe una señal y el brazo se levanta. 6. El brazo baja.	Flujo Principal: 1. Recibe petición de apertura desde Ingreso o Salida. 2. Envía señal PWM para rotar servomotor 90°. 4. Inicia temporizador de seguridad (ej. 10s) o espera señal de paso de sensor. 5. Envía señal para cerrar la barrera (0°).
Flujo Alternativo:	Flujo Alternativo:
Postcondiciones: La barrera vuelve a su estado inicial	
CU Relacionados :	

Tabla XV. caso de uso: Actualizar disponibilidad

ID: CU-005	
Nombre del CU: Actualizar disponibilidad	
Actor(es): LED	
Requisitos Relacionados: RF05	
Descripción: Visualiza en tiempo real la cantidad de cupos disponibles calculados por el sistema.	
Precondiciones: Ha ocurrido un evento de entrada o salida.	
Actor LED	Sistema
Flujo Principal: 5. Actualiza la visualización.	Flujo Principal: 1. Recibe notificación de cambio en el conteo (desde CU-001). 2. Consulta el valor actual 3. Formatea el mensaje (ej: PLAZAS DISPONIBLES = 2). 4. El sistema envía el mensaje actualizado
Flujo Alternativo: 5.1 El sector está lleno, no se actualiza la pantalla LED	Flujo Alternativo: 2. Si el cálculo es 0, envía el mensaje "LLENO" o "SIN CUPO"
Postcondiciones: Información pública actualizada.	
CU Relacionados:	

Tabla XVI. caso de uso: Detectar movimiento

ID: CU-006	
Nombre del CU: Detectar movimiento	
Actor(es): Sensor PIR	
Requisitos Relacionados: RF06	
Descripción: Activa la iluminación de un sector específico solo al detectar presencia para ahorro energético.	
Precondiciones: El sistema está en "Modo Nocturno" o el sensor de luminosidad indica oscuridad.	
Actor(Sensor PIR)	Sistema
Flujo Principal: 1.Detecta radiación infrarroja (movimiento) en el sector. 2. Envía señal al sistema.	Flujo Principal: 3.El sistema recibe la señal y activa el controlador de las luces de ese sector específico. 4. Inicia un temporizador de apagado. 5. Al finalizar el temporizador, verifica si hay nuevo movimiento. Si no hay, apaga las luces.
Flujo Alternativo:	Flujo Alternativo: 4.1 Si el sensor sigue enviando señales durante el conteo del temporizador, el sistema reinicia el temporizador a 0, manteniendo la luz encendida sin interrupciones.
Postcondiciones: Las luces se apagan automáticamente tras un periodo de inactividad, ahorrando energía.	
CU Relacionados :	

Diagramas de secuencias

Diagrama de secuencia CU-001

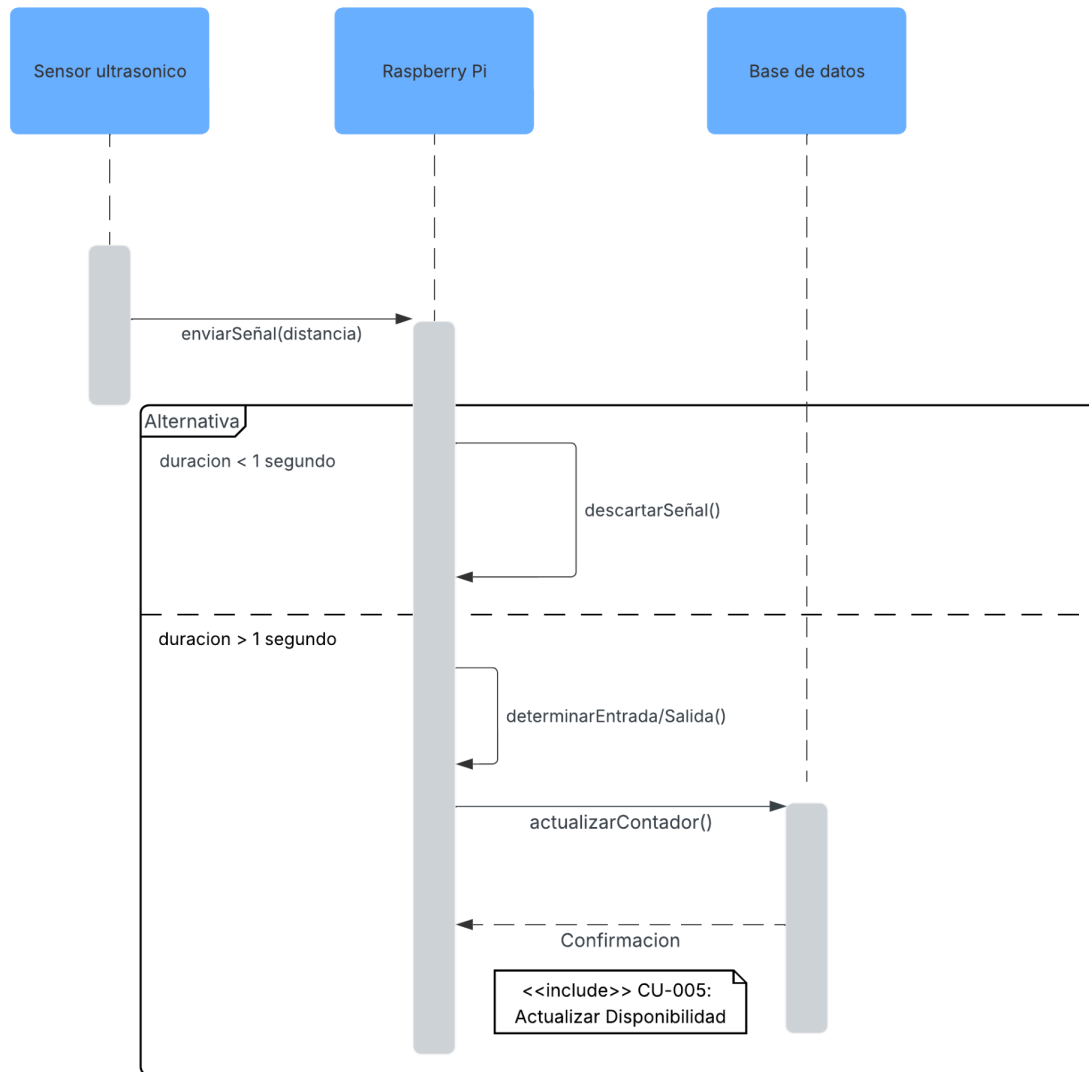


Ilustración 3: Diagrama de secuencia: detectar vehículo

Diagrama de secuencia CU-002

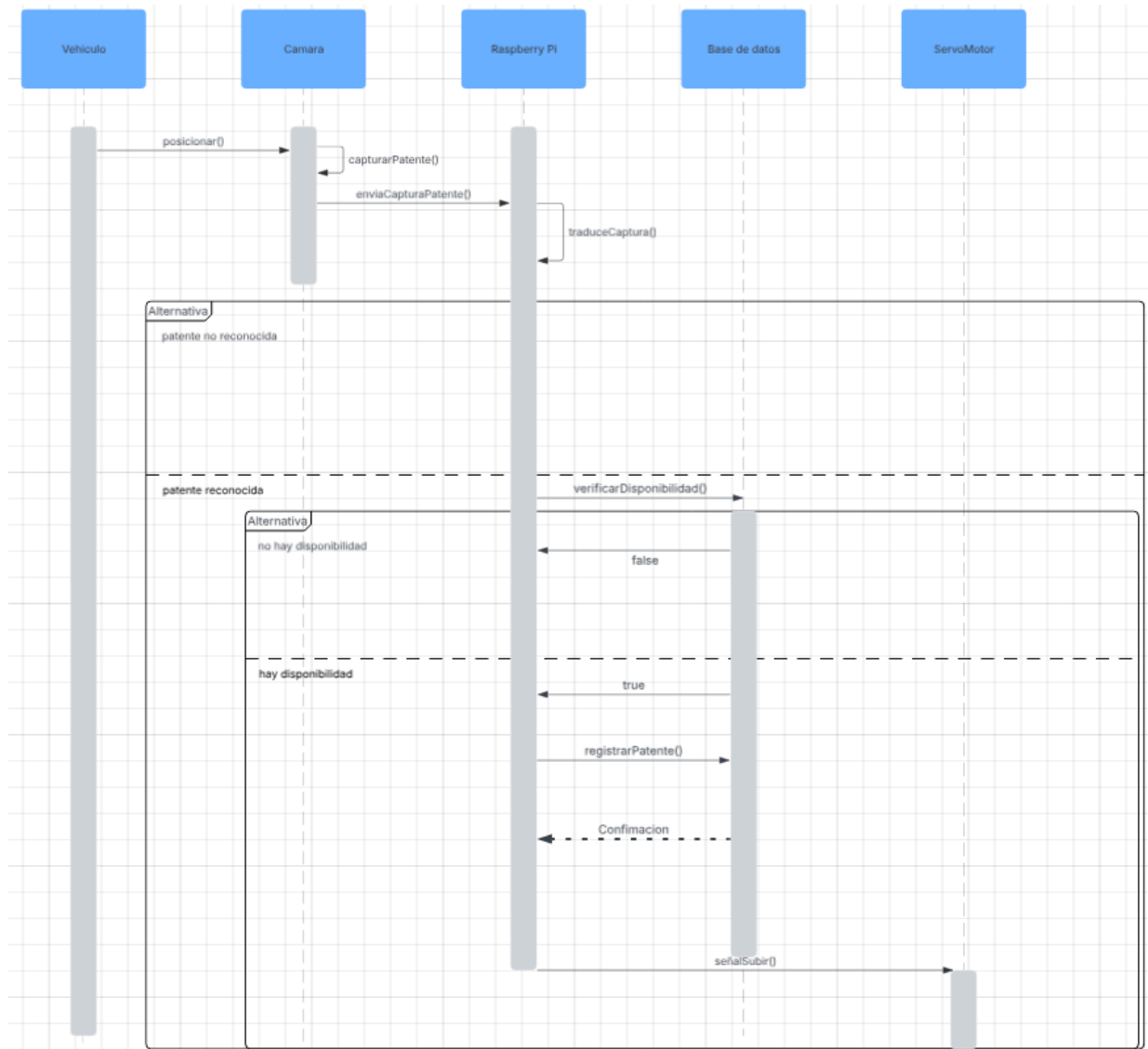


Ilustración 4: Diagrama de secuencia: gestionar ingreso vehicular

Diagrama de secuencia CU-003

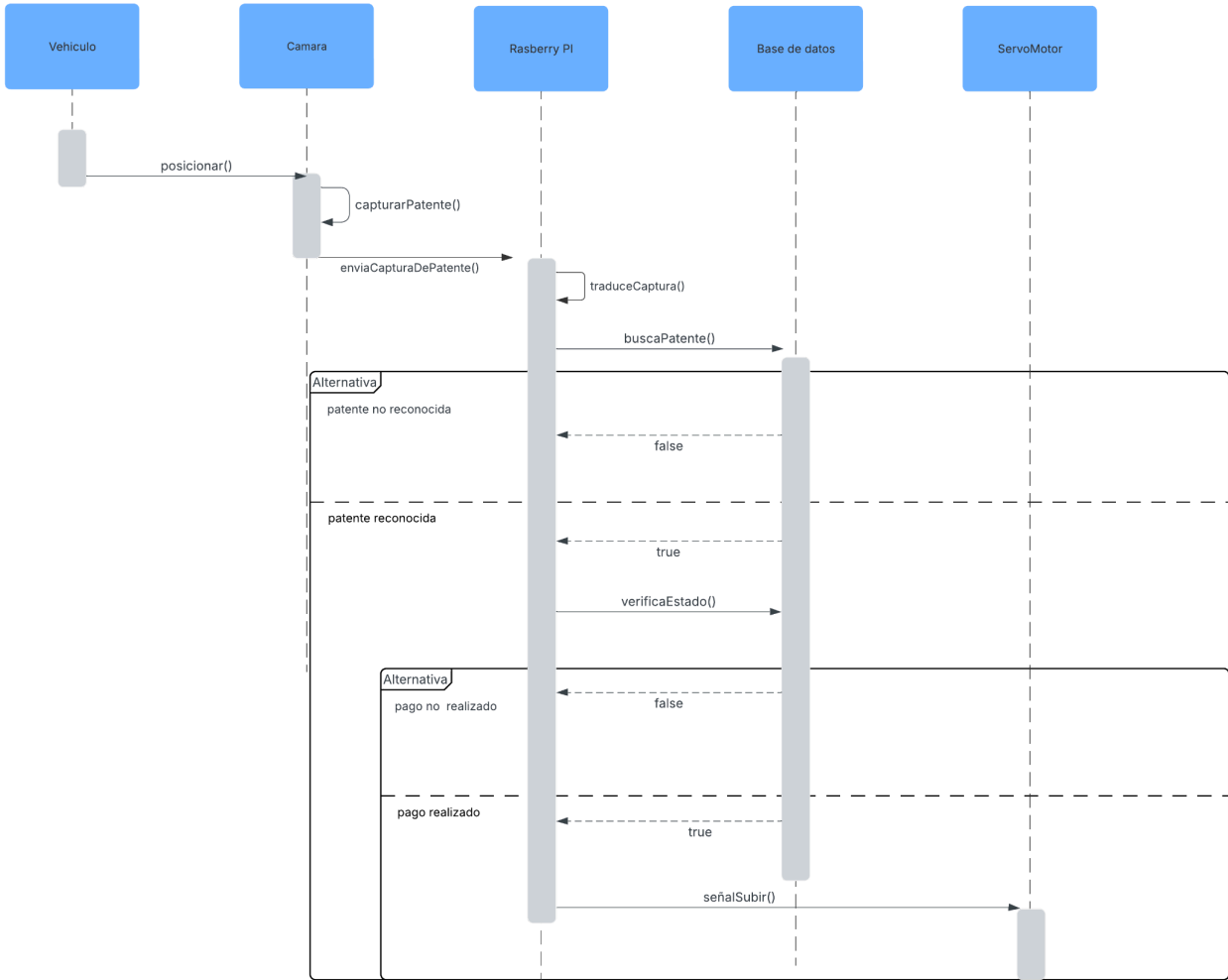


Ilustración 5: Diagrama de secuencia: Gestionar salida vehicular

Diagrama de secuencia CU-004

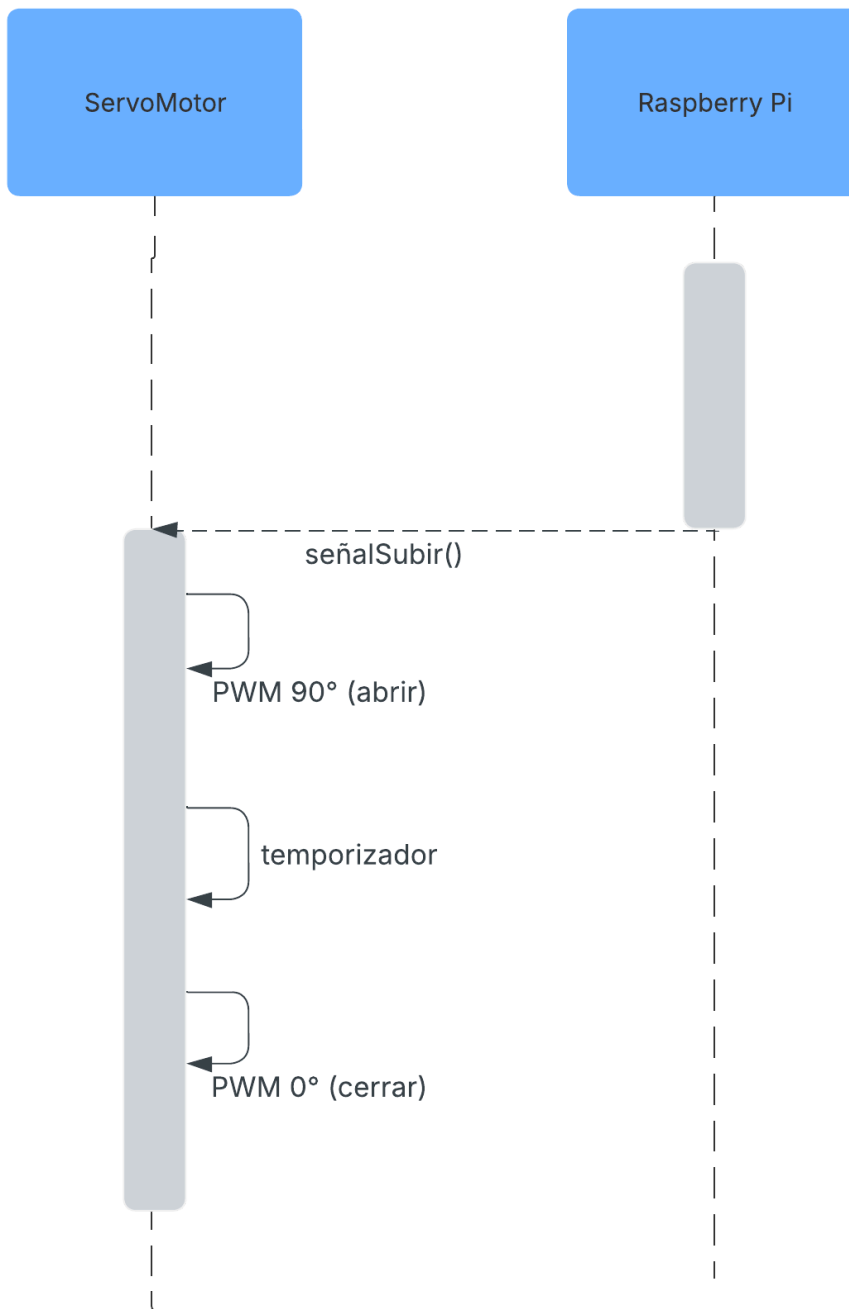


Ilustración 6: Diagrama de secuencia: Accionar barrera

Diagrama de secuencia CU-005

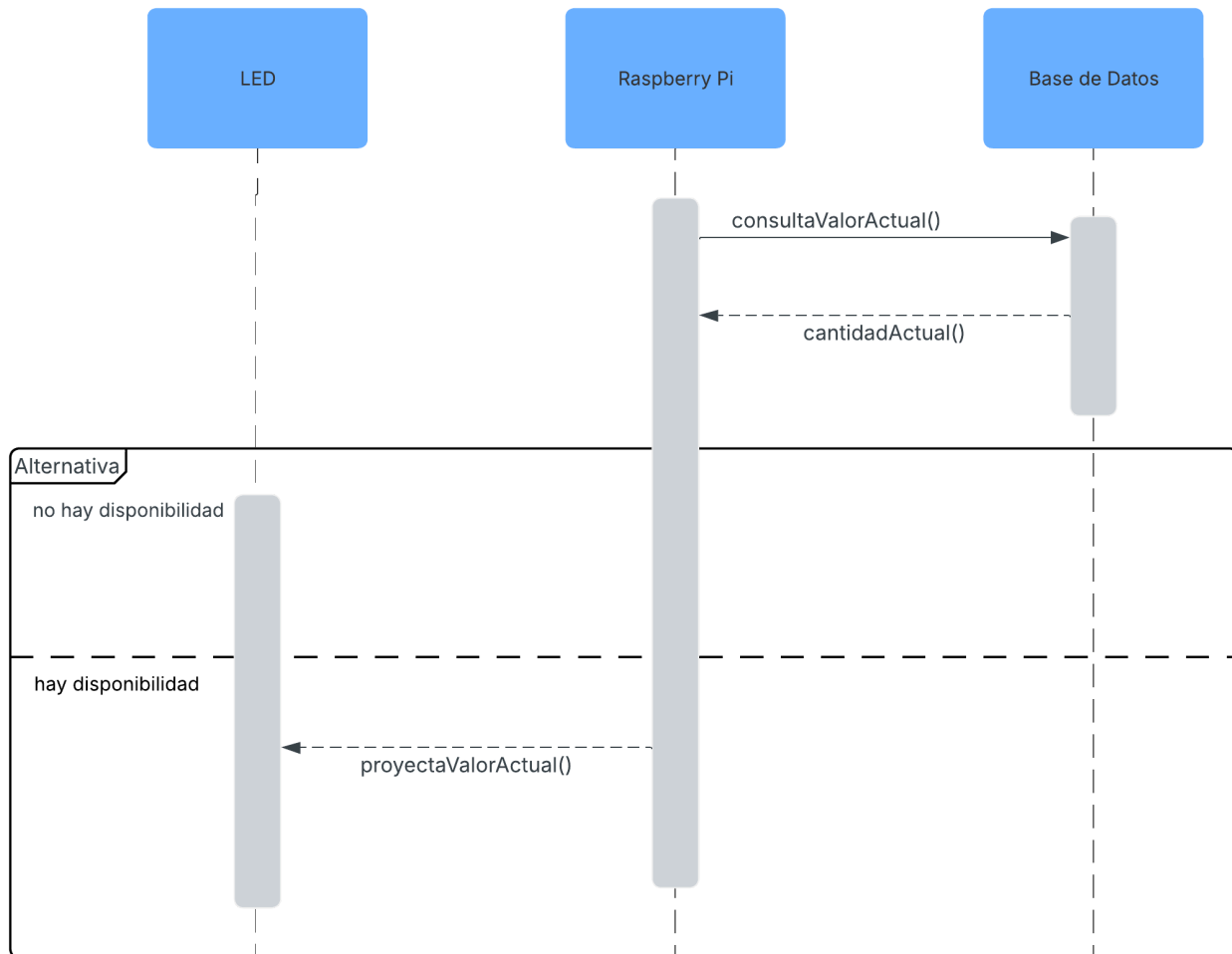


Ilustración 7: Diagrama de secuencia: Actualizar disponibilidad

Diagrama de secuencia CU-006

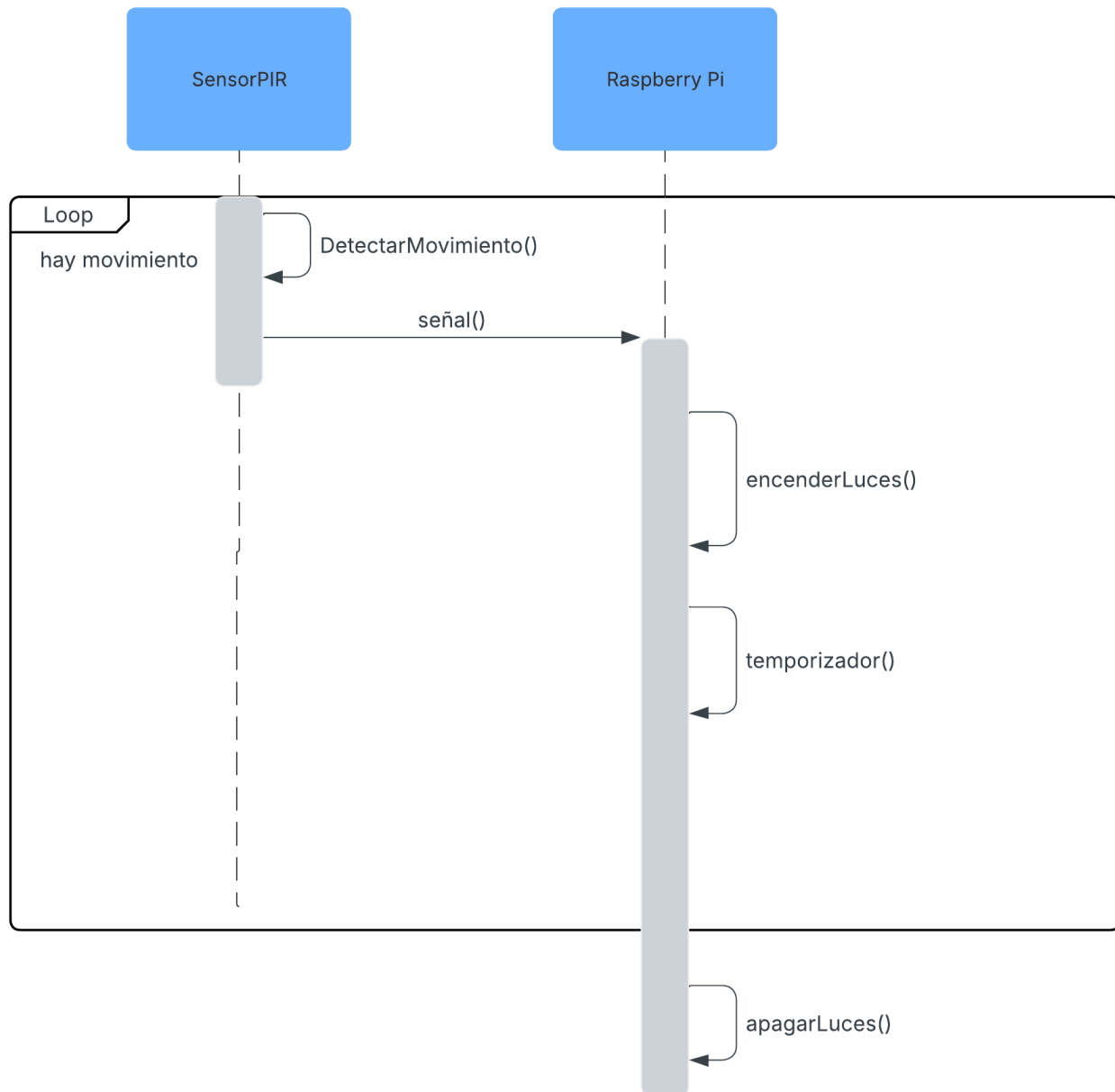


Ilustración 8: Diagrama de secuencia: Controlar iluminación

Diagrama de clases del sistema

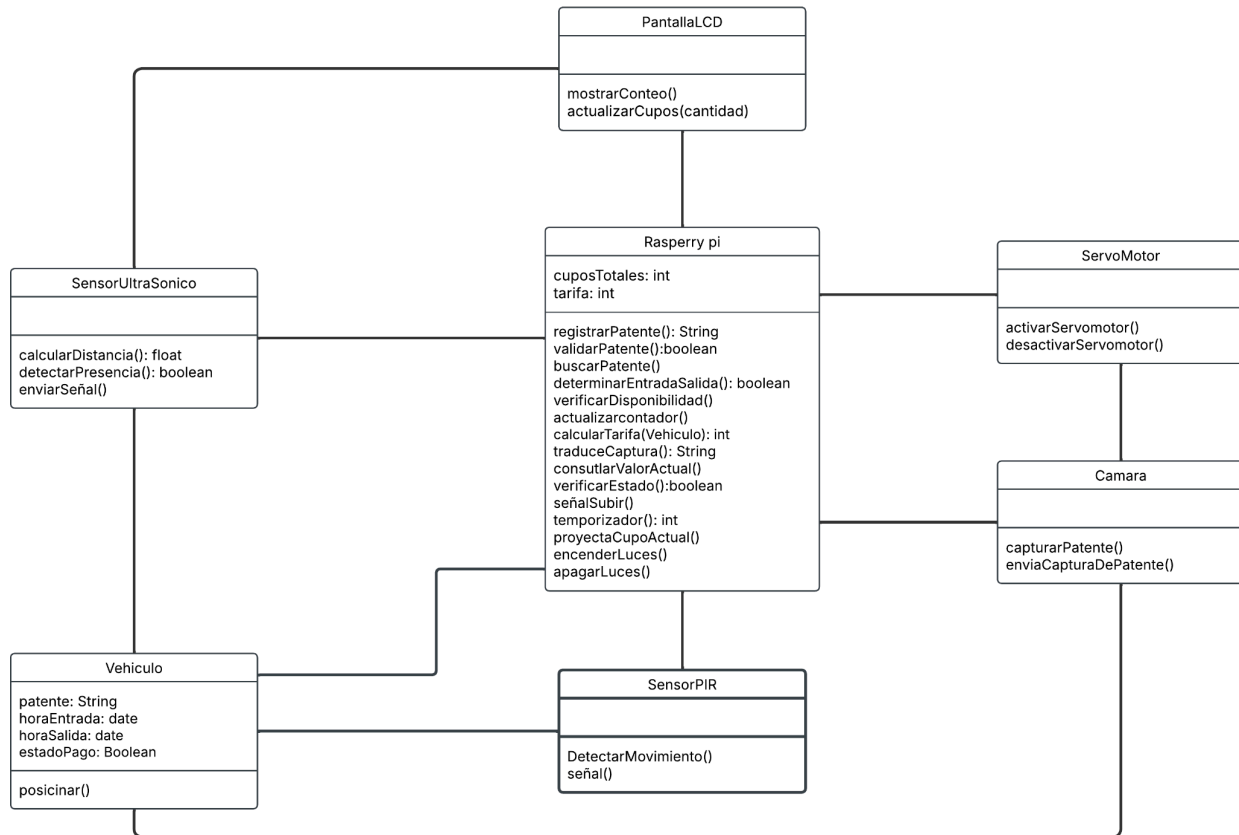


Ilustración 9. Diagrama de clases

Descripción de la Arquitectura (vista del modelo diseño).

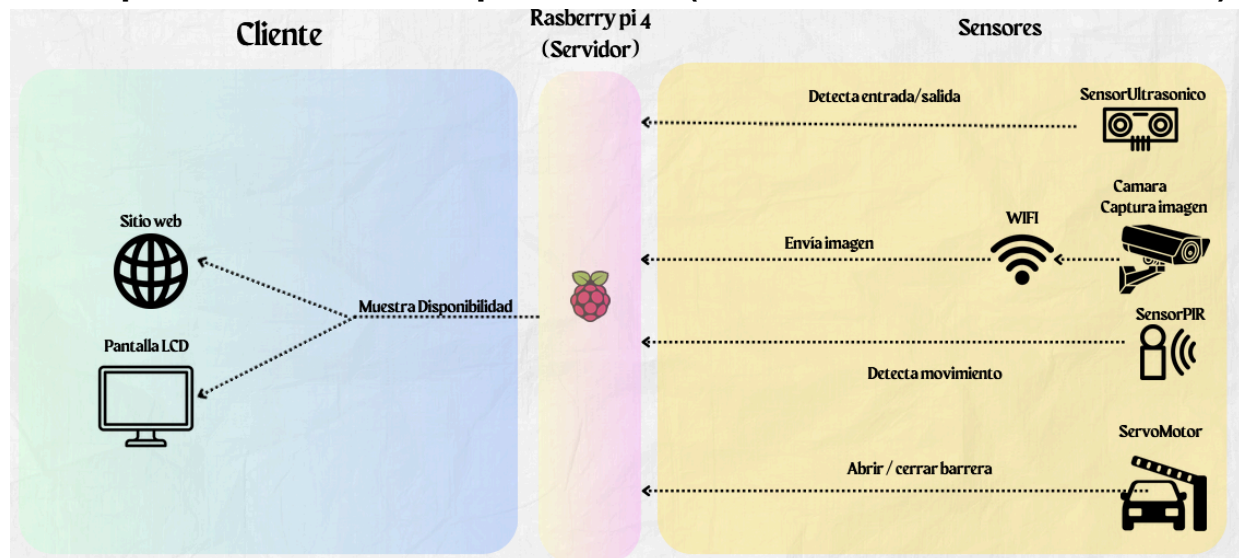


Ilustración 10: Descripción de la arquitectura del proyecto

Documento de Diseño de Interfaz Usuario



Ilustración 11: Interfaz de usuario

Se presenta el diseño de la interfaz de visualización principal (Cliente de Software), optimizada para su despliegue en paneles LED de gran formato. Su estructura se centra en la legibilidad inmediata para el conductor en movimiento.

Los elementos clave del diseño son:

- **Barra de Estado Global (Superior):** Centraliza los datos críticos del sistema. Muestra el indicador de ocupación total (Total: 38/95) para una evaluación rápida de la capacidad del recinto, junto con la fecha y hora sincronizadas con el servidor para la validación temporal de los accesos.
- **Indicadores de Sector (Cuerpo):** Se implementa un diseño modular de tarjetas que desglosa la disponibilidad por zonas físicas (Sector A y B).
 - **Codificación Semántica de Color:** Utiliza el verde para alta disponibilidad y el ámbar/naranja para disponibilidad media o precaución, permitiendo una interpretación visual instantánea sin necesidad de lectura detallada.
 - **Tipografía de Alto Contraste:** Los contadores numéricos ("25", "13") se presentan con un tamaño de fuente dominante para asegurar la visibilidad a distancia.

Implementación

Plan de Integración

Módulos implementados

Relacionados con la interfaz

- **Dashboard Web:** La pantalla que muestra "Total: 38/95" y los sectores. StreamLIT
- **Controlador de Panel LED:** El código que envía el texto al letrero físico LED.

Relacionados con los sensores

Son los scripts que corren "por detrás" en la Raspberry Pi 4B.

- **Módulo de Visión (LPR):** El script que usa la cámara y OpenCV. pytesseract para transformar la imagen a texto
- **Módulo de Distancia:** El script que lee los sensores ultrasónicos.
- **Módulo de Actuadores:** El script que mueve los servomotores (barreras).
- **Módulo de Iluminación:** El script que lee el PIR y enciende luces.
- **Gestor de Base de Datos:** El archivo que conecta con SQLite.

Descripción de la Arquitectura (vista desde los módulos del caso de uso)

Modelo de Implementación

Tabla XVII. Encargados módulos de la interfaz

Módulos(Interfaz)	Descripción	Encargado
Dashboard Principal	Interfaz gráfica desarrollada en (Tkinter/Flask) que visualiza la ocupación de los Sectores A y B, y el contador global en tiempo real.	Erik Correa
Actualizar panel	Script que convierte los datos numéricos de disponibilidad en señales seriales para proyectarlas en la matriz LED física de la entrada.	Erik Correa

Tabla XVIII. Encargados módulos de los sensores

Módulos(funcionamiento de los sensores)		Encargado
Módulo LPR (Entrada/Salida)	Script de Python que integra la cámara y la librería OpenCV/OCR para capturar la patente, convertirla a texto y validarla contra la base de datos.	Ángel Cabezas
Módulo de Sensores Ultrasónicos	Script que gestiona los pines GPIO (Trigger/Echo) de los 4 sensores HC-SR04 para detectar ocupación y actualizar el contador.	Rafael Nakata
Controlador de Barreras (Actuadores)	Módulo que envía señales PWM a los servomotores para abrir la barrera (90°) solo cuando el Módulo LPR valida el acceso.	Rafael Nakata
Módulo de Eficiencia Energética	Lógica que lee la señal digital del sensor PIR y activa el relé de iluminación por un tiempo temporizado.	Álvaro Guarachi
Gestor de Base de Datos	<i>(Módulo Faltante Agregado)</i> Clase que gestiona la conexión SQL, permitiendo guardar registros de entrada/salida y calcular tarifas.	Erik Correa

Aspectos generales

Avance de acuerdo a la Carta Gantt

Tabla XIX. Fase 1: Investigación y Prototipo Inicial

Tareas	Resultado final
Investigación de ideas	Logrado en su totalidad
Lluvia de ideas	Logrado en su totalidad

Selección de la idea definitiva	Logrado en su totalidad
Búsqueda de información complementaria	Logrado en su totalidad
Modelo 3D(Unity)	Logrado en su totalidad

Tabla XX. Fase 2: Planificación y Definición

Tareas	Resultado final
Estudio acerca de raspberry	Logrado en su totalidad
Estudio acerca de los tipos de sensores	Logrado en su totalidad
Planificación de estimaciones	Logrado en su totalidad
Instalacion del SO	Logrado en su totalidad
Elaboración Informe 1	Logrado en su totalidad
Planificación de Riesgos	Logrado en su totalidad

Tabla XXI. Fase 3: Desarrollo e implementación

<p>Ingeniería de Software</p> <ul style="list-style-type: none"> Definición requisitos funcionales y no funcionales Corrección informe I Elaboración de Diagramas de Casos de Uso Refinamiento de Tablas de Casos de Uso Creación de Diagrama de Clases 	Logrado en su totalidad
<p>Implementación Temprana y Pruebas Unitarias</p> <ul style="list-style-type: none"> Pruebas de Sensores <ul style="list-style-type: none"> Conexión en Protoboard Desarrollo de scripts de prueba en Python (lectura de distancia/movimiento) 	Logrado en su totalidad
<p>Implementación Temprana y Pruebas Unitarias 2</p> <ul style="list-style-type: none"> Pruebas de Visión <ul style="list-style-type: none"> Conexión de cámara a Raspberry Pi 	Logrado en su totalidad

<ul style="list-style-type: none"> ○ Prueba de captura de imagen y primeras pruebas de librerías LPR 	
Entrega Informe II	Logrado en su totalidad

Tabla XXII. Fase 4: Integración y Producto Final

Desarrollo e Integración <ul style="list-style-type: none"> ● Integración de Hardware ● Integración de Software 	Logrado en su totalidad
Validación y Cierre <ul style="list-style-type: none"> ● Pruebas integrales de simulación del flujo completo 	Logrado en su totalidad
Documentación Final	Logrado en su totalidad
Elaboración Informe III	Logrado en su totalidad
Manual de Usuario	Logrado en su totalidad
Poster	Logrado en su totalidad

Problemas Encontrados - Soluciones Propuestas

Problemas	Soluciones
Incompatibilidad de Librerías: Los sensores basados en <i>GrovePi</i> presentaron conflictos de comunicación con la arquitectura del sistema.	Migración a GPIO Nativo: Se reemplazaron por sensores ultrasónicos estándar gestionados directamente mediante librerías GPIO de Python, eliminando intermediarios.
Baja Precisión LPR: La cámara inicial no lograba distinguir los caracteres de la patente con nitidez suficiente.	Upgrade de Hardware y Software: Se adquirió una cámara de mayor resolución (V2) y se refinaron los parámetros del algoritmo de visión (LPR) para mejorar el enfoque.
Complejidad del cableado: La gran cantidad de conexiones físicas dificulta la identificación de fallos y el mantenimiento.	Gestión de Cableado: Se implementó un sistema de etiquetado y organización de cables (rotulación) para trazar cada conexión hacia la Raspberry Pi.
Saturación de CPU (Overload): El sistema operativo se saturaba al mantener el procesamiento de imagen activo el 100% del tiempo.	Optimización por Eventos: Se reprogramó la lógica para que la cámara permanezca en <i>standby</i> y solo se active cuando el sensor ultrasónico detecta un vehículo (Trigger).

Trabajo Futuro

Si bien el prototipo actual "EstacionaFlash" cumple con los objetivos de validación funcional y técnica, se han identificado las siguientes líneas de desarrollo para evolucionar el sistema hacia una implementación comercial robusta:

1. **Optimización del Modelo LPR con Deep Learning:**

- Se planea sustituir o complementar el actual algoritmo de visión artificial basado en procesamiento de imágenes tradicional por modelos de **Deep Learning (como YOLOv8)**. Esto mejorará drásticamente la precisión del reconocimiento de patentes en condiciones adversas (baja iluminación, lluvia, ángulos difíciles o patentes sucias), elevando la tasa de acierto por encima del 98%.

2. **Desarrollo de Aplicación Móvil para Usuarios:**

- Implementar una App (Android/iOS) que permita a los conductores consultar la disponibilidad en tiempo real antes de llegar al recinto, reservar plazas con anticipación y localizar su vehículo dentro del estacionamiento mediante un mapa interactivo.

3. **Integración de Pasarelas de Pago Reales:**

- Evolucionar el módulo de pagos lógico actual hacia una integración funcional con pasarelas de pago bancarias (WebPay, Stripe) o billeteras digitales. Esto permitirá el cobro automático vinculado a la tarjeta de crédito del usuario al momento de la lectura de patente en la salida, eliminando completamente la necesidad de cajeros físicos.

4. **Escalabilidad a Hardware Industrial:**

- Para la puesta en marcha en un entorno real, se contempla la sustitución de los componentes de prototipado (servomotores SG90 y sensores HC-SR04) por **hardware de grado industrial** (barreras electromecánicas de alto ciclo, sensores inductivos de piso y cámaras IP con certificación para exteriores), garantizando la durabilidad y resistencia del sistema ante el uso intensivo y condiciones climáticas.

Conclusión

La fase de diseño y desarrollo del proyecto "EstacionaFlash" ha permitido validar técnica y funcionalmente la solución propuesta, alcanzando los siguientes hitos críticos:

1. **Validación de la Arquitectura IoT:** Se ha confirmado que la arquitectura Cliente-Servidor centralizada en la Raspberry Pi 4B es la estrategia adecuada. El sistema ha demostrado capacidad para orquestar simultáneamente la lógica de visión artificial (LPR) y el control de sensores físicos sin latencia perceptible, garantizando la autonomía operativa requerida.
2. **Integración Hardware-Software Exitosa:** A diferencia de las pruebas iniciales, el prototipo final presenta una sincronización robusta entre la detección ultrasónica y la captura de imágenes. La implementación de disparadores por evento (triggers) optimizó el uso de CPU, solucionando los problemas de saturación evidenciados en las primeras iteraciones.
3. **Viabilidad Tecnológica y Económica:** El proyecto ha demostrado que es factible implementar un sistema de control de acceso vehicular eficiente y automatizado utilizando componentes de bajo costo. La sustitución de tickets físicos por reconocimiento de patentes no solo mejora la experiencia del usuario, sino que reduce significativamente los costos operativos y el impacto ambiental.
4. **Escalabilidad del Prototipo:** El diseño modular del software (basado en diagramas de clases y secuencias bien definidos) permite una futura migración hacia hardware industrial o servicios en la nube sin necesidad de reescribir el núcleo lógico del sistema, cumpliendo con los requisitos de mantenibilidad y escalabilidad planteados.

En definitiva, "EstacionaFlash" cumple con los objetivos específicos de automatización y gestión en tiempo real, entregando un producto de ingeniería funcional que resuelve la problemática de congestión y control en estacionamientos comerciales.

Referencias

- [1] Arduino.cl, "Micro Servo Motor SG90 9G," *Arduino.cl*, [En línea]. Disponible en: <https://arduino.cl/producto/micro-servo-motor-sg90-9g/>.
- [2] MCI Electronics, "Cámara V2 para Raspberry Pi," *MCI Electronics*, [En línea]. Disponible: <https://mcielectronics.cl/shop/product/camara-v2-para-raspberry-pi-raspberry-pi-21481/>.
- [3] MCI Electronics, "Sensor de Proximidad de Ultrasonido HC-SR04," *MCI Electronics*, [En línea]. Disponible: <https://mcielectronics.cl/shop/product/sensor-de-proximidad-de-ultrasonido-hc-sr04-23582/>.
- [4] Mediatech, "Tienda de Electrónica y Robótica en Chile," *Mediatech*, [En línea]. Disponible: <https://mediatech.cl/>.
- [5] D. Aracena, "Material compartido por el profesor - Asignatura Proyecto II," *Google Drive*, Universidad de Tarapacá, 2025. [En línea]. Disponible: <https://drive.google.com/drive/u/1/folders/13I54IW9OofTCxDqKDl1iKrSNhdGpzlt0>.
- [6] Equipo EstacionaFlash, "Diagramas de casos de uso y clases," *Lucidchart*, 2025. [En línea]. Disponible: https://lucid.app/lucidchart/edb1af02-4c03-4e51-a8c2-74a1f786ef9b/edit?beaconFlowId=C72FA5319B624A87&invitationId=inv_965c987c-aef3-48e8-a5ad-8e0d74eb2035&page=0_0#.
- [7] Equipo EstacionaFlash, "Diagramas de secuencias," *Lucidchart*, 2025. [En línea]. Disponible: https://lucid.app/lucidchart/e0672d13-0039-4a33-90ca-a4c60d857f69/edit?invitationId=inv_a cb7248a-15d4-4c13-a968-2797d384e685&page=0_0#.
- [8] Equipo EstacionaFlash, "Diseño de Interfaz de Usuario (Digital Signage)," *Figma*, 2025. [En línea]. Disponible: <https://www.figma.com/make/OzkXidTqwnWDRfjRvoqCyk/Digital-Signage-Interface-Design?node-id=0-1&p=f&t=6X7ESNJ22uxl3TRQ-0&fullscreen=>