



EstacionaFlash

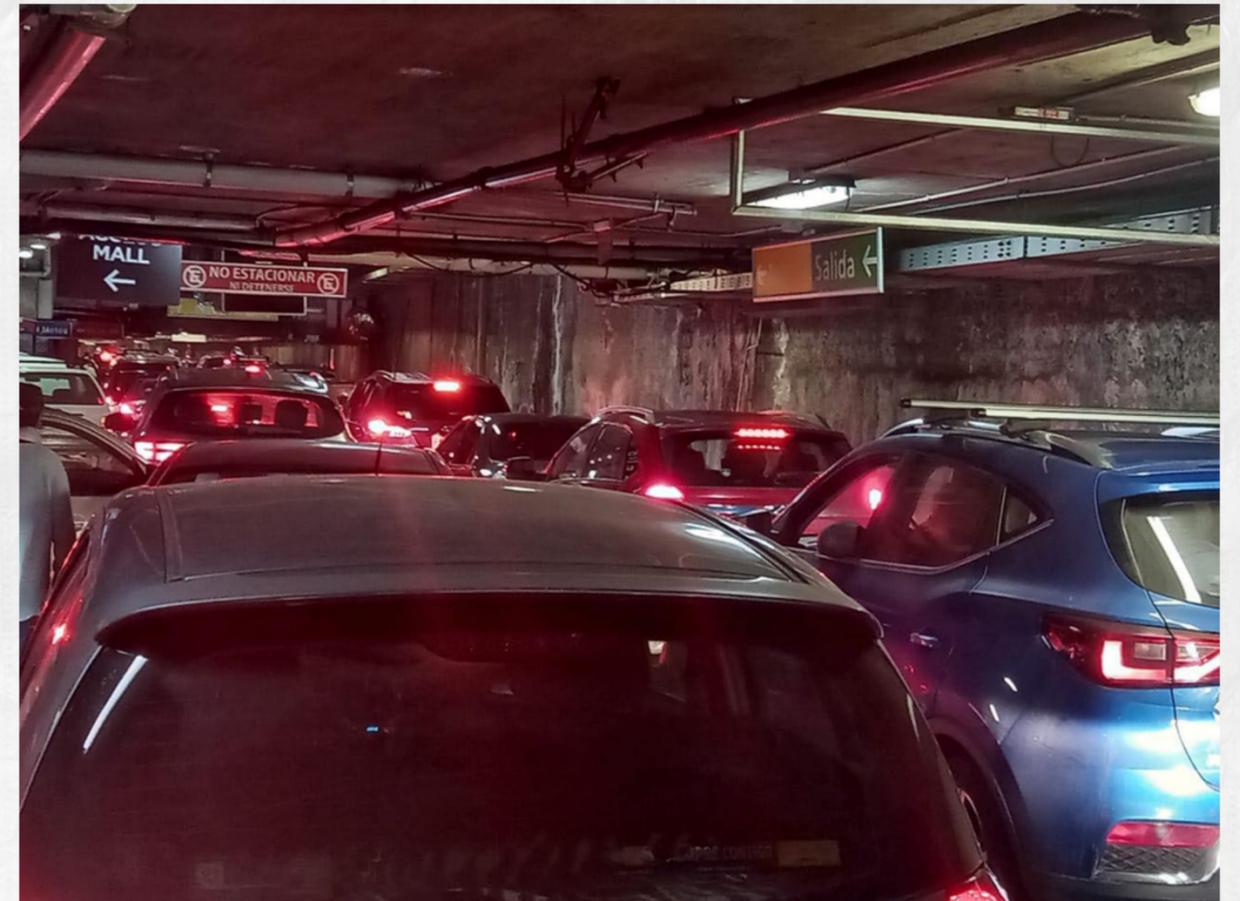


Problemática

- Falta de control y monitoreo en estacionamientos comerciales.
- Congestión y dificultad para encontrar espacios disponibles.
- Procesos manuales ineficientes (tickets, cobros).
- Gasto energético en iluminación sin control.

Consecuencias

- Pérdida de tiempo y mala experiencia del usuario.
- Costos operativos elevados por iluminación ineficiente.
- Errores humanos en cobros o control de accesos.
- Baja satisfacción de administradores y clientes.



Solución Propuesta

Un sistema inteligente que:

- Detecta vehículos y plazas libres en tiempo real.
- Controla accesos automáticamente.
- Muestra disponibilidad en pantallas LED.
- Optimiza la iluminación según movimiento.
- Reconoce patentes de autos.





Objetivos

General:

Desarrollar un sistema de automatización basado en sensores con Raspberry Pi para solucionar la falta de gestión, control y monitoreo en tiempo real de los estacionamientos comerciales (malls, supermercados)





Objetivos

Específicos:

- Investigar acerca de los componentes software y hardware que se utilizarán en el proyecto.
- Implementar un sistema de software que identifique y valide patentes para autorizar pagos automáticamente.
- Realizar pruebas de integración entre cámaras, sensores, motores y Raspberry para asegurar respuesta en tiempo real.
- Implementar sensores para detectar vehículos y controlar eficientemente la iluminación.
- Implementar actuadores para el control sincronizado de apertura y cierre de barreras.
- Asegurar la coordinación del Raspberry entre cámaras, sensores, motores y panel LED para una respuesta precisa.



Restricciones

Restricciones (Lo que nos limita):

- **Hardware (Prototipo):** El sistema se construirá usando cámaras LPR, sensores (ultrasónicos, PIR), motores de barrera y una Raspberry Pi.
- **Plazo de Ejecución:** El proyecto tiene una duración total definida de 3 meses.
- **Capacidad del Prototipo:** El sistema gestionará y monitoreará un máximo de 4 plazas de estacionamiento simultáneamente.
- **Seguridad y Normativa:** Se implementarán medidas para proteger datos sensibles (patentes, registros) cumpliendo con la Ley de Protección de Datos Personales de Chile.
- **Presupuesto:** Estamos limitados a los componentes definidos(Raspberry).



Suposiciones

Suposiciones (Lo que damos por hecho):

Disponibilidad de Hardware: Se contará a tiempo con todos los componentes físicos (cámaras LPR, sensores, barreras y Raspberry Pi).

Experiencia del Equipo: El equipo de desarrollo posee las habilidades necesarias en integración de hardware y desarrollo en Python.

Adopción del Usuario: Los conductores y administradores aceptarán el sistema LPR, encontrándolo intuitivo y eficiente.



Suposiciones

Infraestructura del Cliente: El cliente (mall/supermercado) proporcionará el acceso físico, los puntos de red y la energía eléctrica requerida.

Compatibilidad Técnica: Se asume una integración y compatibilidad total entre el hardware (sensores, cámaras, Raspberry Pi) y el software desarrollado.

Tiempo de Pruebas: Habrá tiempo y recursos suficientes asignados en el cronograma para realizar pruebas exhaustivas y los ajustes necesarios.



Roles

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



Organización del Proyecto

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



Planificación y Costos

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

Hardware (La Arquitectura Física)

Raspberry Pi 4.

Detección de Acceso (Entrada/Salida):

- 2x Cámaras LPR (Lector de Patentes).
- 2x Motores (para las barreras de acceso).

Detección de Plazas:

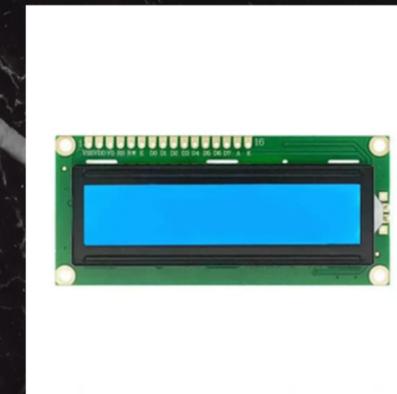
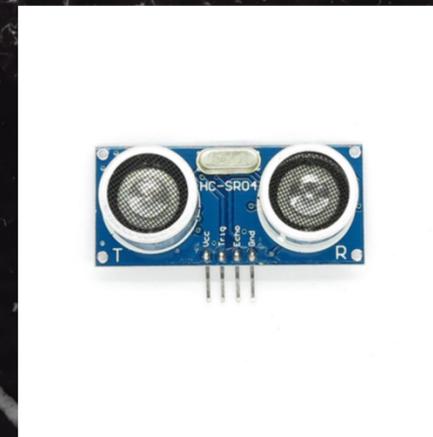
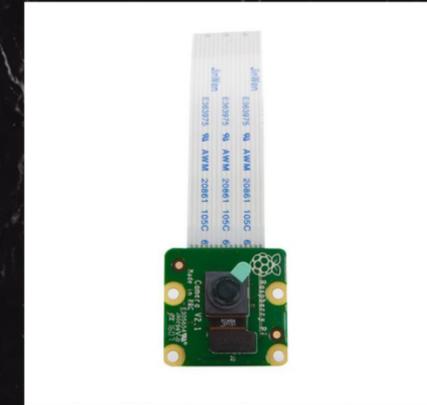
- 4x Sensores Ultrasónicos (para verificar si la plaza está ocupada).

Eficiencia Energética:

- 4x Sensores PIR (para activar luces solo cuando hay movimiento).

Información al Usuario:

- 2x Panel LCD (para mostrar plazas disponibles).



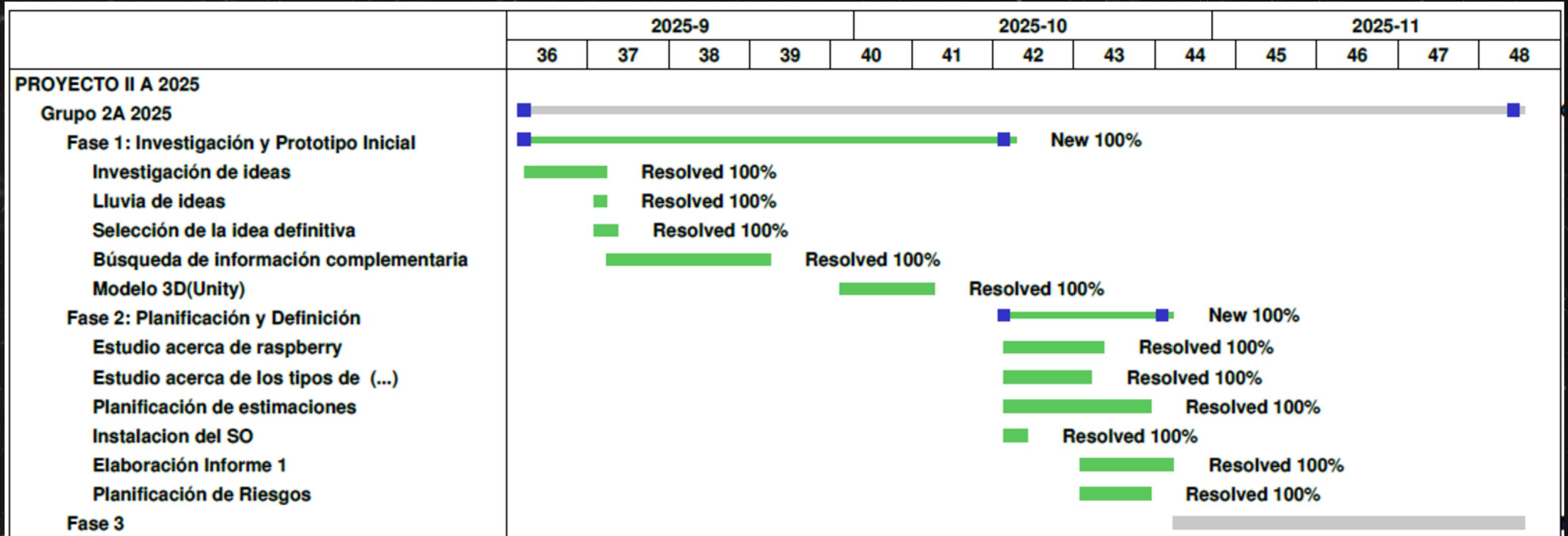
Herramientas de Software

Mecanismos de Comunicación:

-  • Whatsapp: Coordinación diaria.
-  • Discord: Reuniones remotas.
-  • Google Drive: Documentos en borrador.
-  • Redmine: Gestión de tareas y entregables finales.



Carta gantt



Riesgos Identificados

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.



Gestión de Riesgos (Nivel 1 - Catastrófico)

Identificamos 3 riesgos principales que pueden detener el proyecto:

1. Integración LPR (Tiempo): (90% Probabilidad)

Riesgo: Subestimar el tiempo que toma hacer funcionar el LPR.

Acción: Priorizar pruebas de concepto (PoC) en la Semana 1.

2. Incompatibilidad LPR (Software): (80% Probabilidad)

Riesgo: Que la librería (OpenCV) no funcione con nuestro RPi o Python.

Acción: Validar librerías ANTES de comprar las cámaras.

3. Daño físico del hardware: (20% Probabilidad)

Riesgo: Que hardware se malogre y por ende quede inutilizable (ej: Raspberry)

Acción: Tener hardware de respaldo (si es posible) o buscar equipos alternativos.



Conclusión

- Se completó exitosamente la fase de planificación del proyecto “EstacionaFlash”.
- Se definieron el propósito, alcance, roles, costos y riesgos del sistema.
- El documento sirve como base para la fase de implementación.
- En las siguientes etapas se avanzará hacia la programación e integración de sensores, cámaras y barreras controladas por Raspberry Pi.



Referencias

- Arduino.cl. (s.f.). Micro Servo Motor SG90 9G. Arduino.cl. <https://arduino.cl/producto/micro-servo-motor-sg90-9g/>
- MCI Electronics. (s.f.). Cámara V2 para Raspberry Pi. MCI Electronics. <https://mcielectronics.cl/shop/product/camara-v2-para-raspberry-pi-raspberry-pi-21481/>
- MCI Electronics. (s.f.). Sensor de Proximidad de Ultrasonido HC-SR04. MCI Electronics. <https://mcielectronics.cl/shop/product/sensor-de-proximidad-de-ultrasonido-hc-sr04-23582/>
- Mediatech. (s.f.). Mediatech | Tienda de Electrónica y Robótica en Chile. Mediatech. <https://mediatech.cl/>
- Diego Aracena, N. (s.f.). Material compartido por el profesor. Google Drive. <https://drive.google.com/drive/u/1/folders/13I54IW9OofTCxDqKDI1iKrSNhdGpzIt0>



Gracias •

