



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

## **Informe Final**

### **Proyecto Z-Parking**

**Curso: Diseño y Arquitectura de Software**

**Docente: Mag. Alberto Flor Rodriguez**

**Integrantes:**

***Flores Navarro, Eduardo Gino (2023076793)***

**Tacna – Perú  
2025**



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MPV	ELV	ARV	10/10/2020	Versión Original

## INDICE GENERAL

Antecedentes	1
Planteamiento del Problema	4
Problema	
Justificación	
Alcance	
Objetivos	6
Marco Teórico	
Desarrollo de la Solución	9
Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental)	
Tecnología de Desarrollo	
Metodología de implementación (Documento de VISION, SRS, SAD)	
Cronograma	11
Presupuesto	12
Conclusiones	13
Recomendaciones	14
Bibliografía	15
Anexos	16
Anexo 01 Informe de Factibilidad	
Anexo 02 Documento de Visión	
Anexo 03 Documento SRS	
Anexo 04 Documento SAD	
Anexo 05 Manuales y otros documentos	



## 1.- Antecedentes

Los estacionamientos en entornos universitarios y empresariales han operado tradicionalmente con sistemas manuales o semiautomáticos, lo que genera ineficiencias en el registro de vehículos, cálculo de tarifas y control de espacios. Con el avance de la tecnología, surgen oportunidades para optimizar estos procesos mediante soluciones accesibles que integren hardware y software. El proyecto Z-Parking nace en el contexto académico de la Universidad Privada de Tacna, como parte del curso Diseño y Arquitectura de Software, con el objetivo de aplicar conocimientos en un entorno real y resolver una problemática cotidiana mediante innovación tecnológica.

## 2.- Planteamiento del Problema

### 2.1.- Problema

La gestión ineficiente de estacionamientos mediante métodos manuales conlleva una serie de dificultades críticas:

- Registro manual de vehículos: Propenso a errores, omisiones o duplicidad de información.
- Falta de información en tiempo real: Los usuarios desconocen la disponibilidad de espacios, generando congestión y pérdida de tiempo.
- Cálculo manual de tarifas: Origina inconsistencias en los cobros y conflictos con los conductores.
- Falta de seguridad preventiva: No existen sistemas integrados que alerten sobre incendios o condiciones de riesgo.
- Pérdida de datos clave: No se generan reportes automatizados que apoyen la toma de decisiones.

### 2.2.- Justificación

La implementación de un sistema automatizado como Z-Parking se justifica mediante los siguientes aspectos:

- Económico: Reduce costos operativos al minimizar la dependencia de personal para tareas repetitivas.
- Operativo: Agiliza los procesos de entrada y salida, mejorando la experiencia del usuario.
- Tecnológico: Aprovecha herramientas accesibles (Arduino, .NET, SQL Server) para crear una solución escalable.
- Seguridad: Integra monitoreo ambiental para prevenir emergencias.



- Educativo: Ofrece a los estudiantes experiencia práctica en desarrollo de sistemas reales.

### 2.3.- Alcance

Incluye:

- Desarrollo de una aplicación desktop en C# .NET para gestión del estacionamiento.
- Integración con sensores Arduino (ultrásonicos, RFID, humedad y llama).
- Registro automatizado de entrada y salida de vehículos.
- Asignación y visualización en tiempo real de espacios disponibles.
- Cálculo automático de tarifas según tiempo de estacionamiento.
- Generación y envío de boletas electrónicas vía correo.
- Roles de usuario (administrador y operador) con permisos diferenciados.
- Módulo de reportes estadísticos (ocupación, ingresos, tendencias).
- Monitoreo ambiental con alertas tempranas (incendios, humedad).
- Base de datos en SQL Server para almacenamiento seguro de datos.

No incluye:

- Desarrollo de aplicación móvil para reservas o pagos.
- Integración con pasarelas de pago electrónico.
- Soporte para múltiples sedes o centralización de datos.
- Mantenimiento o instalación física de hardware externo.
- Reconocimiento de placas mediante visión artificial.
- Sistema de reservas en línea.
- Soporte para otros sistemas operativos (Linux, macOS).

## 3.- Objetivos

### 3.1- Objetivos de Negocios

Automatizar procesos para reducir errores y tiempos de espera.

1. Optimizar el uso de espacios mediante control en tiempo real.
2. Mejorar la experiencia del usuario con interfaz intuitiva y boletas electrónicas.
3. Generar reportes clave para la toma de decisiones operativas y estratégicas.
4. Reducir costos operativos mediante la eliminación de tareas manuales repetitivas.
5. Fortalecer la seguridad con monitoreo ambiental y protección de datos.
6. Garantizar escalabilidad para adaptarse a futuras expansiones o necesidades.

### 3.2.- Objetivos de Diseño

1. Interfaz intuitiva que permita su uso con mínima capacitación.
2. Arquitectura modular para facilitar el mantenimiento y la escalabilidad.
3. Integración eficiente entre hardware (sensores) y software.
4. Respuesta en tiempo real para gestión de entradas, salidas y alertas.
5. Alta disponibilidad con tolerancia a fallos de sensores o conexión.
6. Seguridad integrada en acceso a datos y operaciones críticas.



7. Bajo consumo de recursos para funcionar en equipos estándar.
  
  
  
8. Documentación clara para desarrolladores y usuarios finales.

#### 4.- Marco Teórico

El desarrollo del sistema Z-Parking se sustenta en un marco teórico que integra principios de ingeniería de software, tecnologías de automatización y fundamentos de gestión de estacionamientos. En el ámbito de los sistemas de gestión de estacionamientos, la evolución desde métodos manuales hacia soluciones tecnológicas integradas representa un avance significativo en la optimización de recursos espaciales y temporales. La automatización de estacionamientos, según estudios contemporáneos, se fundamenta en tres pilares esenciales: control de acceso mediante sistemas de identificación vehicular, gestión inteligente de espacios mediante tecnologías de detección y procesamiento eficiente de datos a través de software especializado.

En el campo de la sensórica aplicada, Z-Parking incorpora principios de la Internet de las Cosas (IoT) mediante la implementación de sensores ultrasónicos HC-SR04 que operan bajo el principio de tiempo de vuelo para medición de distancias, módulos RFID RC522 que funcionan en la frecuencia estándar de 13.56 MHz según la normativa ISO 14443, y microcontroladores Arduino Uno R3 con arquitectura RISC de 8 bits. Esta combinación tecnológica permite crear una red de detección precisa y confiable para el monitoreo en tiempo real de los espacios de estacionamiento.

La arquitectura de software del sistema se construye sobre patrones de diseño establecidos en la ingeniería de software moderna. Se implementa una arquitectura en capas que separa claramente las responsabilidades de presentación, lógica de negocio y acceso a datos, facilitando el mantenimiento y la escalabilidad. El patrón Repository proporciona una abstracción eficiente para el acceso a datos, mientras que el patrón Factory optimiza la creación de objetos sensores, permitiendo una gestión flexible del hardware.

El framework de desarrollo seleccionado (.NET Framework 4.8) ofrece un entorno robusto para aplicaciones Windows, complementado con Entity Framework para el mapeo objeto-relacional y Windows Forms para la implementación de interfaces gráficas intuitivas. En el ámbito de bases de datos, se aplican principios de modelado entidad-relación y normalización hasta la tercera forma normal, garantizando la integridad y consistencia de la información en SQL Server 2019.

La comunicación entre hardware y software se establece mediante protocolos seriales UART (Universal Asynchronous Receiver-Transmitter), utilizando interfaces físicas RS-232 para la transmisión asíncrona de datos. Este enfoque garantiza una comunicación estable y eficiente entre los componentes físicos y la aplicación de gestión.

Los principios de usabilidad, establecidos por Jakob Nielsen en la década de 1990, guían el diseño de la interfaz de usuario, priorizando la consistencia visual, la



visibilidad del estado del sistema, la prevención de errores y el reconocimiento antes que la memorización. Estos fundamentos aseguran que el sistema sea accesible para operadores con diversos niveles de habilidad técnica.

En materia de seguridad informática, se implementan medidas basadas en control de acceso por roles (RBAC), cifrado de contraseñas con algoritmos SHA-256 y el principio de mínimo privilegio, garantizando la protección de datos sensibles y la integridad del sistema.

Este marco teórico integrado proporciona los cimientos conceptuales y tecnológicos que validan técnicamente el diseño e implementación de Z-Parking, estableciendo las bases para el desarrollo de una solución eficiente, escalable y confiable para la gestión moderna de estacionamientos.

## 5.- Desarrollo de la Solución

### 5.1.- Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental)

#### - *Factibilidad Técnica:*

El sistema Z-Parking es técnicamente viable mediante el uso de tecnologías establecidas como C# .NET Framework 4.8 para el desarrollo de la aplicación desktop, SQL Server para la gestión de base de datos, y Arduino Uno para la integración de sensores. La arquitectura en capas garantiza modularidad y mantenibilidad, mientras que la comunicación serial mediante protocolo UART asegura una integración estable entre hardware y software. Los componentes electrónicos seleccionados (sensores ultrasónicos, módulos RFID) son de bajo costo y amplia disponibilidad en el mercado local.

#### - *Factibilidad Económica:*

La inversión inicial requerida es de S/ 684.00, con un retorno de inversión estimado en 2.3 meses. Los indicadores financieros demuestran rentabilidad: Relación Beneficio/Costo de 3.89, VAN positivo de S/ 7,386.15 y TIR del 488%. Los costos operativos mensuales son mínimos (S/ 20.00), considerando que no se requieren licencias de software comerciales y el mantenimiento es sencillo.

#### - *Factibilidad Operativa:*

El sistema está diseñado para ser operado por personal con conocimientos básicos de informática. La interfaz intuitiva de Windows Forms reduce la curva de aprendizaje a menos de 2 horas de capacitación. Los procesos automatizados eliminan el 90% de los errores manuales y reducen en 70% el tiempo de registro de vehículos. El sistema incluye modos de operación manual como respaldo para casos de falla en sensores.

#### - *Factibilidad Social:*

Z-Parking mejora significativamente la experiencia del usuario al reducir tiempos de espera y proporcionar transparencia en los cobros



mediante boletas electrónicas. El sistema promueve la inclusión tecnológica al ser accesible para usuarios con diferentes niveles de habilidad digital. Como proyecto académico, fortalece las competencias profesionales de los estudiantes involucrados.

- **Factibilidad Legal:**

El sistema cumple con la Ley Peruana 29733 de Protección de Datos Personales, implementando consentimiento informado y almacenamiento seguro de información. Todas las herramientas de software utilizadas cuentan con licencias gratuitas para uso educativo (.NET Framework, SQL Server Express, Arduino IDE). Los componentes electrónicos cumplen con estándares internacionales de seguridad.

- **Factibilidad Ambiental:**

La solución reduce el consumo de papel mediante la emisión digital de boletas, minimiza las emisiones de CO<sub>2</sub> al agilizar el flujo vehicular, e incorpora monitoreo ambiental para prevención de incidentes. Los componentes electrónicos tienen bajo consumo energético y son reciclables al final de su vida útil.

## 5.2.- Tecnología de Desarrollo

**Frontend:**

- Lenguaje: C# .NET Framework 4.8
- Interfaz: Windows Forms con controles personalizados
- Reportes: Crystal Reports para documentos formales

**Backend:**

- Base de datos: Microsoft SQL Server 2019 Express
- ORM: Entity Framework 6.0 para mapeo objeto-relacional
- Comunicación: System.IO.Ports para comunicación serial

**Hardware:**

- Microcontrolador: Arduino Uno R3 (ATmega328P)
- Sensores: HC-SR04 (ultrásónico), RC522 (RFID), DHT11 (humedad)
- Comunicación: Protocolo UART a 9600 baudios
- Herramientas de Desarrollo:
- IDE: Visual Studio 2022 Community Edition
- Control de versiones: Git con repositorio en GitHub
- Documentación: Microsoft Word y [draw.io](#)

## 5.3.- Metodología de implementación (Documento de VISION, SRS, SAD)

**Fase 1: Documentación de Requisitos**

- Documento de Visión: Define objetivos de negocio, stakeholders y visión del producto



- SRS (Software Requirements Specification): Especifica 14 requerimientos funcionales y 15 no funcionales detallados
- SAD (Software Architecture Document): Describe arquitectura en 5 paquetes principales y patrones de diseño

#### **Fase 2: Desarrollo Iterativo**

- Sprint 1: Módulo de gestión de usuarios y autenticación (2 semanas)
- Sprint 2: Integración con sensores y comunicación serial (3 semanas)
- Sprint 3: Módulo de cálculo de tarifas y generación de boletas (2 semanas)
- Sprint 4: Sistema de reportes y alertas de seguridad (2 semanas)

#### **Fase 3: Pruebas y Validación**

- Pruebas unitarias: Cobertura del 80% del código lógico
- Pruebas de integración: Validación hardware-software
- Pruebas de aceptación: Con usuarios finales del estacionamiento

#### **Fase 4: Despliegue y Capacitación**

- Instalación: Configuración en computadora del estacionamiento
- Capacitación: 2 sesiones de 2 horas para administradores y operadores
- Soporte: Mantenimiento por 6 meses post-implementación

#### **Fase 5: Documentación Final**

- Manual técnico: Especificaciones de instalación y mantenimiento
- Manual de usuario: Guías ilustradas para operadores
- Informe final: Resultados y lecciones aprendidas

## **6- Cronograma**



Fase	Actividad	Sem 1-2	Sem 3-4	Sem 5-6	Sem 7-8	Sem 9-10	Sem 11-12
Planificación	Definición de requisitos	■■■■■					
	Diseño de arquitectura	■■■■■	■■■■■				
Desarrollo	Módulo de gestión de usuarios		■■■■■	■■■■■			
	Integración con sensores			■■■■■	■■■■■		
	Módulo de tarifas y boletas				■■■■■	■■■■■	
Pruebas	Sistema de reportes y alertas					■■■■■	■■■■■
	Pruebas unitarias			■■■■■	■■■■■	■■■■■	
	Pruebas de integración					■■■■■	■■■■■
Implementación	Pruebas de aceptación						■■■■■
	Instalación y configuración						■■■■■
	Capacitación de usuarios						■■■■■
Documentación	Manuales y informes finales					■■■■■	■■■■■

## 7- Presupuesto

Rubro	Descripción	Cantidad	Costo Unitario (S/)	Subtotal (S/)
<b>HARDWARE</b>				
	Arduino Uno R3	2	80	160
	Sensor ultrasónico HC-SR04	5	8	40
	Módulo RFID RC522	1	17	17
	Display LCD 16x2	2	25	50
	Sensor DHT11 (Humedad)	1	12	12
	Sensor de llama	1	15	15
	Servomotor SG90	1	12	12
	Cables y conectores	Varios	-	50
	Fuente de poder 12V	1	25	25
	<b>Subtotal Hardware</b>			<b>381</b>



Rubro	Descripción	Cantidad	Costo Unitario (S/)	Subtotal (S/)
<b>MATERIALES</b>				
	Material de apoyo (impresiones, documentos, carpetas)	1 lote	95	95
	Almacenamiento (memorias USB 32GB, servicio nube)	2 unidades	27.5	55
	Transporte y comunicaciones (movilidad local, llamadas)	3 meses	38.33	115
	Otros gastos menores (papelería, toner, material de oficina)	1 lote	38	38
	<b>Subtotal Materiales</b>			<b>303</b>

#### Notas del Presupuesto:

- Los costos de hardware están basados en precios de mercado local
- Los recursos humanos no tienen costo por ser un proyecto académico
- La universidad proporciona infraestructura y servicios sin costo
- No se incluyen costos de mantenimiento post-implementación
- El presupuesto incluye un 10% de imprevistos en los rubros de materiales

Duración Total del Proyecto: 12 semanas (3 meses)

Inversión Total Requerida: S/ 684.00

#### 8.- Conclusiones

El desarrollo del proyecto Z-Parking ha demostrado ser una solución tecnológica integral y viable para la gestión moderna de estacionamientos. A lo largo del proceso de diseño e implementación, se ha comprobado que la integración entre hardware accesible y software robusto permite automatizar eficientemente los procesos de registro vehicular, control de espacios y cálculo de tarifas. El sistema logra reducir significativamente los tiempos de atención y minimizar los errores humanos, mejorando sustancialmente la experiencia tanto para los administradores como para los usuarios finales.

Desde el punto de vista técnico, la arquitectura en capas implementada ha facilitado el desarrollo modular y el mantenimiento del sistema, mientras que la



comunicación estable entre los sensores Arduino y la aplicación desktop garantiza un funcionamiento confiable en tiempo real. Económicamente, el proyecto demuestra una notable rentabilidad con una inversión inicial moderada y un rápido retorno de la misma, respaldado por indicadores financieros sólidos.

Operativamente, Z-Parking se adapta perfectamente a las capacidades del personal existente, requiriendo mínima capacitación y ofreciendo mecanismos de respaldo que aseguran la continuidad del servicio. Socialmente, el proyecto no solo moderniza un servicio esencial sino que también contribuye al desarrollo de competencias técnicas en el ámbito académico. Ambientalmente, representa un avance hacia la sostenibilidad al reducir el consumo de papel y optimizar el flujo vehicular.

## 9.- Recomendaciones

Para maximizar el éxito en la implementación y evolución del sistema Z-Parking, se recomienda:

Iniciar con una fase piloto en un sector del estacionamiento que permita validar el funcionamiento del sistema en condiciones reales, identificar posibles ajustes y familiarizar gradualmente al personal con la nueva tecnología. Establecer un programa de capacitación continuo que incluye no solo el uso básico del sistema sino también

procedimientos para el manejo de situaciones excepcionales y mantenimiento preventivo de los componentes hardware.

Desarrollar un plan de mantenimiento periódico que contemple la calibración de sensores, actualización de software y verificación de la integridad de los datos. Considerar la implementación de un sistema de respaldo energético que garantice el funcionamiento continuo ante interrupciones del suministro eléctrico.

Para futuras versiones, se sugiere explorar la integración con pasarelas de pago electrónico para agilizar aún más el proceso de cobranza, desarrollar una aplicación móvil complementaria que permita a los usuarios consultar disponibilidad de espacios y realizar reservas previas, e implementar un módulo de reconocimiento de placas vehiculares mediante visión artificial para reforzar el sistema de identificación.

Es fundamental establecer canales de feedback continuo con los usuarios operativos y finales para detectar oportunidades de mejora y ajustar el sistema según las necesidades emergentes. Documentar rigurosamente todas las incidencias y soluciones implementadas durante la fase de operación inicial para construir un conocimiento institucional que facilite el mantenimiento y futuras expansiones.



## 10.- Bibliografía

- Arduino LLC. (2023). Arduino Technical Documentation. Recuperado de <https://www.arduino.cc/en/Main/Documentation>
- Microsoft Corporation. (2023). .NET Framework Documentation. Recuperado de <https://docs.microsoft.com/en-us/dotnet/framework/>
- Ministerio de Transportes y Comunicaciones del Perú. (2022). Reglamento Nacional de Estacionamientos.
- Ley N° 29733 - Ley de Protección de Datos Personales. (2011). Lima, Perú
- Sommerville, I. (2016). Software Engineering. Pearson Education.

## 11.- Anexos

- Anexo 01 Informe de Factibilidad
- Anexo 02 Documento de Visión
- Anexo 03 Documento SRS
- Anexo 04 Documento SAD
- Anexo 05 Manuales y otros documentos