



**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**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

**Z-Parking**  
**Documento de Visión**

**Versión {1.0}**



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

## ÍNDICE GENERAL

<b>Introducción</b>	<b>4</b>
1.1. Propósito	4
1.2. Alcance	4
1.3. Definición y siglas	5
1.4. Referencias	5
1.5. Visión General	5
<b>Posicionamiento</b>	<b>5</b>
1.6. Oportunidad de negocio	5
1.7. Definición del problema	6
<b>Descripción de interesados y usuarios</b>	<b>6</b>
1.8. Interesados	6
1.9. Usuarios	6
1.10. Entorno de usuario	7
1.11. Necesidades clave	7
<b>Vista general de producto</b>	<b>7</b>
1.12. Perspectiva	7
1.13. Capacidades	7
1.14. Costos y precios	8
1.15. Licenciamiento e instalación	9
<b>Características del producto</b>	<b>9</b>
<b>Restricciones</b>	<b>9</b>
<b>Rangos de calidad</b>	<b>9</b>
<b>Precedencia y prioridad</b>	<b>10</b>
<b>Alta prioridad (esencial):</b>	<b>10</b>
● Registrar entrada/salida de autos.	10
● Calcular tarifas automáticamente.	10
● Control de usuarios y permisos.	10
<b>Media prioridad (importante):</b>	<b>10</b>
● Envío de boletas por correo.	10



• Reportes estadísticos.	10
• Detección de humedad/incendios.	10
<b>Baja prioridad (futuras versiones):</b>	<b>10</b>
• App móvil para reservas.	10
• Pago con tarjeta desde la app.	10
• Reconocimiento de placas con cámara.	10
<b>Otros requerimientos</b>	<b>10</b>
1.16. Estándares legales	10
1.17. Estándares de comunicación	10
1.18. Estándares de plataforma	11
<b>Conclusiones</b>	<b>12</b>
<b>Recomendaciones</b>	<b>12</b>



## Introducción

### 1.1. Propósito

El propósito de este proyecto es desarrollar un sistema de control de estacionamiento que sea totalmente intuitivo y fácil de usar, incluso para personas que no tienen experiencia con software. La idea principal es automatizar el registro de entrada y salida de los vehículos, calcular la tarifa de manera automática y mantener un control más ordenado del estacionamiento.

El sistema está conectado a sensores que ayudan a que el proceso sea todavía más automático. Por ejemplo, los sensores detectan cuándo un vehículo entra o sale para registrar el movimiento sin necesidad de hacerlo manualmente. También se incluyen sensores ambientales, como humedad o detección de fuego, para alertar sobre posibles riesgos y activar mecanismos de seguridad dentro del estacionamiento.

Además, el sistema considera la seguridad de la información, ya que maneja datos personales y contará con roles de usuario y administrador. A futuro, también se incluirá un módulo de reportes estadísticos para tener un mejor control del uso del estacionamiento y apoyar la toma de decisiones.

### 1.2. Alcance

El alcance del proyecto incluye tanto la parte lógica del software como la interacción con sensores que permiten que varios procesos sean automáticos y no dependan del operador.

#### El sistema contempla:

- Registro automático de vehículos al momento de ingresar, gracias a los sensores instalados en la entrada del estacionamiento, los cuales detectan la presencia del vehículo y registran la hora correspondiente.
- Asignación y control de espacios, mostrando qué espacios están ocupados y cuáles están libres de manera visual e intuitiva.
- Detección automática de salida, también mediante sensores, lo que permite registrar el tiempo total que el vehículo estuvo dentro.
- Cálculo de tarifas automáticamente, basándose en el tiempo estacionado y la tarifa por hora establecida.
- Generación y visualización de la boleta, mostrando todos los datos del conductor, el vehículo y el monto a pagar.
- Envío de boletas al correo del conductor.
- Módulo de seguridad y usuarios, donde solo el administrador puede ver la información personal de los usuarios registrados, mientras que los operadores tienen funciones limitadas.
- Posibilidad de generar reportes estadísticos, como proyección a mejoras del sistema.



### 1.3. Definición y siglas

- **BD:** Base de Datos
- **SQL:** Structured Query Language (Lenguaje de Consulta Estructurado)
- **SMTP:** Simple Mail Transfer Protocol (Protocolo para envío de correos)
- **UI:** User Interface (Interfaz de Usuario)
- **C# / C Sharp:** Lenguaje de programación utilizado en el sistema
- **.NET:** Framework usado para construir la aplicación
- **ID:** Identificador único de un registro
- **GUI:** Graphical User Interface (Interfaz gráfica de usuario)

### 1.4. Referencias

- Microsoft Documentation: *.NET Framework y Windows Forms*.
- SQL Server Express – *Microsoft Official Documentation*.
- Sommerville, I. (2011). *Ingeniería del Software* (9<sup>a</sup> edición). Pearson Educación.
- Material de clase y guías proporcionadas por la universidad.
- Apuntes del curso de *Programación Orientada a Objetos y Bases de Datos*.

### 1.5. Visión General

Esta documentación describe el funcionamiento general del sistema de gestión de estacionamiento desarrollado. El proyecto busca automatizar procesos como el registro de entrada y salida de vehículos, el control de espacios y el cálculo de tarifas, manteniendo siempre una interfaz sencilla y fácil de usar.

Además, el sistema considera el uso de sensores (vehículo, humedad y fuego) para que ciertas acciones se ejecuten de manera automática, como detectar ingresos, activar un techo protector cuando llueve o alertar sobre posibles emergencias. También se contempla un módulo de administración para proteger los datos de los usuarios y tener un mejor control del sistema.

### Posicionamiento

### 1.6. Oportunidad de negocio

Hoy en día, la mayoría de estacionamientos todavía funcionan de forma manual: anotan las placas en una libreta, calculan el precio a pagar mentalmente y no saben con seguridad cuántos espacios están libres en tiempo real. Eso hace que todo sea más lento, tanto para el dueño del estacionamiento como para los conductores.

Con Z-Parking, queremos modernizar ese proceso. La idea es simple: usar sensores que detecten cuándo un carro entra o sale, una aplicación que muestre los espacios disponibles al instante, y que el sistema solo calcule automáticamente lo que se debe pagar. Además, incluye funciones extras como detectar humedad o fuego para mayor seguridad.



Esto no solo ahorra tiempo, sino que también hace todo más ordenado y confiable. Es una gran oportunidad porque cada vez más lugares buscan sistemas así: fáciles de usar, automáticos y que crezcan con el negocio.

### 1.7. Definición del problema

Imagina que llegas a un estacionamiento y no sabes si hay espacio. O que al salir, tardan mucho en calcular tu pago. O peor: que no hay manera de saber si hubo un incendio o una fuga de agua a tiempo.

- Eso es lo que pasa en muchos estacionamientos hoy:
- Se usan métodos manuales (papel y lápiz) que dan lugar a errores.
- No hay forma de ver en una pantalla qué espacios están libres.
- A veces se pierde información de los clientes o no se protege bien.
- Casi no hay sistemas que avisen sobre emergencias como incendios.

Z-Parking nace para resolver esos problemas. Queremos que administrar un estacionamiento sea fácil, rápido y seguro, tanto para el que trabaja ahí como para el que estaciona su auto.

### Descripción de interesados y usuarios

#### 1.8. Interesados

Los interesados son las personas o grupos a los que les importa el proyecto o se ven afectados por él. En Z-Parking, los principales interesados son:

- Dueños del estacionamiento: Les interesa que el sistema funcione bien para ganar más dinero y organizar mejor su negocio.
- Administradores: Son los que usan el sistema a diario para controlar entradas, salidas y cobros.
- Clientes o conductores: Quieren estacionar rápido, pagar fácil y sentirse seguros.
- Desarrolladores (nosotros): Queremos crear un sistema que sea útil, fácil de mantener y que funcione sin problemas.

#### 1.9. Usuarios

Los usuarios son quienes usarán directamente el sistema. En nuestro caso:

- **Administrador:** Puede ver todos los datos, generar reportes, configurar tarifas y manejar la información de los conductores.
- **Usuario general:** Registra entradas y salidas, asigna espacios y genera boletas, pero con permisos limitados.
- **Conductor:** Interactúa con el sistema de forma indirecta, ya que recibe boletas por correo y ve los espacios disponibles.



## 1.10. Entorno de usuario

El sistema se usará principalmente dentro de la cabina de control del estacionamiento, en una computadora con Windows. El operador estará sentado frente al monitor, viendo en tiempo real los espacios ocupados y libres. También recibirá alertas de los sensores (como humedad o fuego) y podrá actuar rápido.

Es un entorno tranquilo, pero requiere atención para no equivocarse al ingresar datos o al asignar espacios.

## 1.11. Necesidades clave

Lo más importante que necesitan los usuarios es:

- Simplicidad: Que el sistema sea fácil de entender y usar, incluso para alguien sin experiencia.
- Rapidez: Que las acciones como registrar un auto o generar una boleta sean casi instantáneas.
- Confiabilidad: Que el sistema no falle y que los datos no se pierdan.
- Seguridad: Que solo las personas autorizadas puedan entrar al sistema y ver información sensible.
- Alertas útiles: Que avise a tiempo si hay una emergencia o si un sensor dejó de funcionar.

## Vista general de producto

### 1.12. Perspectiva

Z-Parking es un sistema moderno y automático para gestionar estacionamientos. Reemplaza los métodos antiguos como los registros en papel o los sistemas complicados, por uno fácil de usar que combina una aplicación en Windows con sensores inteligentes.

Imagine que el estacionamiento “piensa” y “actúa” solo: detecta cuándo llega un carro, asigna un espacio libre, calcula lo que se debe pagar y hasta envía la boleta al correo. Además, avisa si hay algún riesgo como humedad o fuego. Todo esto hace que la experiencia sea rápida, ordenada y segura para todos.

### 1.13. Capacidades

El sistema puede hacer cosas como:

- Registro automático de autos que entran y salen, gracias a sensores.
- Mostrar en tiempo real los espacios libres y ocupados en una pantalla.
- Calcular solo el costo del estacionamiento según el tiempo.
- Enviar la boleta al correo del conductor sin necesidad de imprimir.
- Alertar al administrador si hay un incendio o mucha humedad.
- Guardar toda la información de manera segura en internet (en la nube).



- Permitir que solo usuarios autorizados entren al sistema.

## 1.14. Costos y precios

A continuación, se presenta un desglose estimado de los componentes electrónicos necesarios para implementar el sistema Z-Parking en un estacionamiento pequeño (ej. 10-20 espacios):

Cant.	Descripción	Precio Unit. (S/)	Importe (S/)
2	Display LCD 16x2	25	50
5	Sensor ultrasónico HC-SR04	8	40
1	Módulo RFID RC522	17	17
2	Arduino Uno R3	80	160
10	Cable jumper Macho-Hembra	0.5	5
1	Fuente de poder 12V 2A	25	25
1	Servomotor SG90	12	12
1	Sensor de humedad DHT11	12	12
1	Sensor de llama	15	15
1	Módulo Bluetooth HC-05	25	25
1	Protopboard	10	10
1	Cable Ethernet UTP 5m	10	10
<b>Total Inversión en Componentes</b>			<b>S/ 381.00</b>

*Explicación de los costos:*

Componentes electrónicos: Incluyen sensores, Arduino, displays y cables para automatizar entradas, salidas, detección de espacios y monitoreo ambiental.

Recursos de la universidad: No se incluyen costos de computadora ni servicios en la nube, ya que se usan los laboratorios de la facultad y el convenio con Azure para estudiantes.



## 1.15. Licenciamiento e instalación

- **Licencia:** El sistema se entrega con licencia gratuita de uso educativo. Para uso comercial, se requiere una licencia única de S/ 250.00 que incluye actualizaciones por 1 año.
  
- **Instalación:** Es simple y rápida:
  - Conectar los sensores Arduino a la computadora.
  - Instalar la aplicación Windows (La mayoría lo tiene por defecto).
  - Configurar la conexión a internet (solo para envío de correos).
  - 30 minutos de capacitación al personal.
  - No se necesita técnico especializado para mantenerlo.

## Características del producto

- Registro automático de autos con sensores.
- Pantalla en tiempo real de espacios libres/ocupados.
- Cálculo automático de tarifas.
- Envío de boletas por correo.
- Alertas de seguridad (fuego/humedad).
- Interfaz colorida y fácil de entender.
- Funciona en computadoras normales con Windows.

## Restricciones

- El sistema necesita Windows 10 o superior.
- Requiere conexión a internet para enviar correos.
- Los sensores deben estar dentro del rango de 50 metros de la computadora.
- Máximo 50 espacios por estacionamiento (por temas de rendimiento visual).
- No funciona en celulares ni tablets (solo computadora).

## Rangos de calidad

- Tiempo de respuesta: Menos de 2 segundos en cualquier acción.
- Disponibilidad: Funciona 24/7, con menos de 1 hora de mantenimiento al mes.
- Seguridad: Solo usuarios registrados pueden ingresar.
- Facilidad de uso: Cualquier persona lo puede usar sin manual.
- Tolerancia a fallos: Si un sensor falla, el sistema sigue funcionando en modo manual.



## Precedencia y prioridad

*Alta prioridad (esencial):*

- Registrar entrada/salida de autos.
- Calcular tarifas automáticamente.
- Control de usuarios y permisos.

*Media prioridad (importante):*

- Envío de boletas por correo.
- Reportes estadísticos.
- Detección de humedad/incendios.

*Baja prioridad (futuras versiones):*

- App móvil para reservas.
- Pago con tarjeta desde la app.
- Reconocimiento de placas con cámara.

## Otros requerimientos

### 1.16. Estándares legales

El sistema Z-Parking debe cumplir con las leyes y normas que protegen a los usuarios y sus datos. Por ejemplo, en Perú existe la Ley de Protección de Datos Personales, que dice que no podemos guardar información personal (como DNI, teléfono o correo) sin permiso. Por eso, el sistema pedirá consentimiento antes de registrar cualquier dato.

También, cuando se emitan boletas o recibos de pago, estos deben seguir las recomendaciones de SUNAT, aunque sea un proyecto universitario, para que en el futuro pueda adaptarse fácilmente a un entorno real.

Otra regla importante es que solo las personas mayores de 18 años podrán registrarse como administradores, ya que son responsables de manejar información sensible.

### 1.17. Estándares de comunicación

Para que todas las partes del sistema se entiendan entre sí, usamos estándares de comunicación conocidos y seguros:

- **Comunicación con sensores:** Los sensores Arduino envían información mediante protocolo serial (UART), un lenguaje simple y confiable para dispositivos electrónicos.



- **Envío de correos:** Para mandar boletas al cliente, el sistema se conecta a servicios como Gmail o Outlook mediante SMTP con TLS, que es como enviar una carta certificada: segura y privada.
- **Conexión a la base de datos:** La aplicación en Windows se comunica con la base de datos usando SQL estándar, el idioma universal para gestionar datos.
- **Contraseñas seguras:** Todas las contraseñas se guardan convertidas en hash (SHA-256), es decir, en forma de código irreconocible, para que ni siquiera los administradores puedan verlas.
- 

## 1.18. Estándares de plataforma

El sistema Z-Parking está desarrollado usando tecnologías estables y ampliamente utilizadas en el mercado, garantizando compatibilidad, facilidad de mantenimiento y posibilidad de crecimiento futuro. A continuación, detallamos los estándares técnicos adoptados:

- **Lenguaje de programación:**

El sistema está desarrollado en C# utilizando .NET Framework 4.8. Este entorno es ideal para crear aplicaciones de escritorio en Windows, ya que ofrece herramientas modernas, alta compatibilidad y gran cantidad de bibliotecas que facilitan tareas como la conexión a bases de datos, el manejo de interfaces gráficas y la comunicación con dispositivos externos.

- **Base de datos:**

Se utiliza Microsoft SQL Server como sistema de gestión de bases de datos. Gracias al convenio educativo con Microsoft, accedemos de forma gratuita a Azure SQL Database, lo que permite almacenar y consultar datos de manera segura y eficiente desde cualquier lugar con acceso a internet.

- **Sistema operativo:**

La aplicación está diseñada específicamente para ejecutarse en Windows 10 o versiones superiores, ya que es el sistema operativo más utilizado en entornos de administración y control. Esto asegura que el software funcione sin problemas en la mayoría de equipos disponibles en el mercado.

- **Plataforma de sensores:**

Todos los sensores (ultrásónicos, RFID, humedad, llama, etc.) están conectados mediante placas Arduino Uno R3, una plataforma de hardware abierto, económica y fácil de programar. El firmware de los Arduino se desarrolla en C++ usando el entorno oficial de Arduino (IDE), lo que garantiza compatibilidad y estabilidad en la lectura de datos.

- **Arquitectura de la aplicación:**



Se sigue una arquitectura en capas (presentación, lógica de negocio y acceso a datos), lo que facilita el mantenimiento, las pruebas y la escalabilidad. Además, se aplica el patrón Factory para la creación de objetos sensores, permitiendo agregar nuevos tipos de sensores en el futuro sin afectar el código existente.

- ***Comunicación en red:***

La aplicación se comunica con los servicios en la nube mediante el protocolo TCP/IP, y con los sensores a través de USB o conexión serial. Para el envío de correos se utiliza el protocolo SMTP, integrado mediante bibliotecas estándar de .NET.

- ***Control de versiones:***

El código se gestiona con Git a través de GitHub, lo que permite el trabajo en equipo, el control de cambios y la documentación automática de cada versión del sistema.

## Conclusiones

El desarrollo del sistema Z-Parking ha permitido demostrar que es posible crear una solución tecnológica accesible y funcional para la gestión de estacionamientos, incluso en el contexto de un proyecto universitario. A lo largo del proceso, se logró integrar hardware y software de manera eficiente, utilizando sensores Arduino para automatizar tareas como el registro de vehículos, la detección de espacios y el monitoreo ambiental.

Se comprobó que el uso de tecnologías como C# .NET, SQL Server y Arduino ofrece una base sólida para construir sistemas estables y fáciles de mantener. Además, la arquitectura en capas y el uso de patrones de diseño facilitaron la organización del código y la escalabilidad del sistema.

Otro logro importante fue garantizar que la interfaz fuera intuitiva, de modo que cualquier persona, incluso sin experiencia técnica, pueda utilizarla sin dificultad. El sistema no solo responde a las necesidades básicas de un estacionamiento, sino que también incorpora características de seguridad y automatización que lo hacen innovador.

## Recomendaciones

Aunque el sistema funciona correctamente, se sugieren las siguientes mejoras y consideraciones para futuras versiones o implementaciones en un entorno real:



- Incluir un módulo de reservas en línea: Permitir que los usuarios reserven un espacio desde una aplicación móvil o página web antes de llegar al estacionamiento.
- Implementar pasarelas de pago electrónico: Integrar opciones como Yape, PayPal o tarjeta de crédito para hacer el proceso de pago aún más rápido y seguro.
- Ampliar la capacidad de sensores: Probar con cámaras para reconocimiento de placas o sensores de peso que ayuden a categorizar vehículos de forma automática.
- Realizar pruebas de estrés: Simular un alto volumen de vehículos para asegurar que el sistema responde correctamente en horas pico.
- Documentar el proceso de instalación y uso: Elaborar una guía visual o en video para que los administradores puedan configurar el sistema sin ayuda externa.
- Considerar el uso de energía renovable: En una versión a mayor escala, se podría integrar paneles solares para alimentar los sensores y reducir el consumo eléctrico.
- Mantener actualizado el software: Revisar periódicamente las actualizaciones de .NET, SQL Server y las bibliotecas de Arduino para garantizar seguridad y compatibilidad.