



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

Sistema Z-Parking

Documento de Arquitectura de Software

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

INDICE GENERAL

1. Introducción	4
1.1 Propósito	4
1.2 Alcance	4
1.3 Definición, siglas y abreviaturas	4
1.4 Referencias	4
1.5 Visión General	5
2. Representación Arquitectónica	5
2.1 Escenarios	5
2.2 Vista Lógica	6
2.3 Vista del Proceso	6
2.4 Vista del desarrollo	6
2.5 Vista Física	6
3. Objetivos y limitaciones arquitectónicas	7
3.1 Disponibilidad	7
3.2 Seguridad	7
3.3 Adaptabilidad	7
3.4 Rendimiento	7
4. Análisis de Requerimientos	8
4.1 Requerimientos funcionales	7
4.2 Requerimientos no funcionales	7
5. Vistas de Caso de Uso	9
6. Vista Lógica	17
6.1 Diagrama Contextual	17
7. Vista de Procesos	18
7.1 Diagrama de Proceso Actual	18
7.2 Diagrama de Proceso Propuesto	18
8. Vista de Despliegue	19
8.1 Diagrama de Contenedor	19
9. Vista de Implementación	20



9.1	Diagrama de Componentes	
	20	
10.	Vista de Datos	22
10.1	Diagrama Entidad Relación	22
11.	Calidad	24
11.1	Escenario de Seguridad	25
11.2	Escenario de Usabilidad	26
11.3	Escenario de Adaptabilidad	27
11.4	Escenario de Disponibilidad	28
11.5	Otro Escenario	28



1. 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.

1.3. Definición, siglas y abreviaturas

- **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

FD01: Informe de Factibilidad de Proyecto – Diseño y Arquitectura de Software, Universidad Privada de Tacna, 2025

FD02: Informe de Visión de Proyecto – Diseño y Arquitectura de Software, Universidad Privada de Tacna, 2025.

FD03: Informe SRS de Proyecto – Diseño y Arquitectura de Software, Universidad Privada de Tacna, 2025

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.

2. Representación Arquitectónica



2.1. Escenarios

Los escenarios describen cómo los usuarios interactúan con el sistema en situaciones típicas. Incluyen acciones como iniciar sesión, realizar búsquedas, gestionar información o generar reportes. Cada escenario muestra el flujo básico y las respuestas esperadas del sistema.

2.2. Vista Lógica

La vista lógica presenta la estructura del sistema en términos de módulos o componentes. Aquí se identifican las funciones principales, cómo se organizan y cómo se comunican entre sí. Su objetivo es mostrar la organización interna sin entrar en detalles técnicos de implementación.

2.3. Vista del Proceso

Esta vista explica cómo se ejecutan las operaciones dentro del sistema. Incluye el manejo de procesos simultáneos, tiempos de respuesta, sincronización y cualquier actividad que dependa del rendimiento o la secuencia de pasos internos.

2.4. Vista del desarrollo

La vista del desarrollo muestra cómo está estructurado el código durante la construcción del sistema. Se describen paquetes, directorios, convenciones de programación y herramientas utilizadas. Ayuda a entender cómo se mantiene y evoluciona el proyecto.

2.5. Vista Física

La vista física describe cómo se despliega el sistema en el entorno real. Incluye servidores, dispositivos, redes y cualquier recurso físico o virtual necesario para que el sistema funcione. Representa la distribución del software en la infraestructura disponible.

3. Objetivos y limitaciones arquitectónicas

3.1. Disponibilidad

El sistema debe mantener una alta disponibilidad debido a que el cobro de peaje funciona de forma continua. Para ello, se contemplan las siguientes consideraciones:

- *Operación 24/7*: El sistema debe estar activo permanentemente para permitir el registro, clasificación y cobro sin interrupciones.
- *Mínimos tiempos de inactividad*: Cualquier falla debe resolverse rápidamente para evitar congestión vehicular y pérdida de datos.



- *Respaldo básico de datos:* La información registrada (entradas, salidas, categorías y pagos) debe guardarse de forma segura para evitar pérdidas ante cortes eléctricos o fallas del sistema.
- *Monitoreo de sensores:* Los sensores de ubicación, humedad y detección de fuego deben mantenerse operativos para garantizar un funcionamiento automático y seguro del entorno.
- *Continuidad del servicio:* Si algún componente falla (sensor, módulo de pago o base de datos), el sistema debe contar con alternativas temporales que permitan mantener el flujo vehicular

3.2. Seguridad

La seguridad del sistema es importante porque se manejan datos personales, información de pagos y procesos automáticos conectados a sensores. Por eso, se consideran los siguientes puntos:

- **Control de accesos:** El sistema contará con roles (administrador y usuario) para evitar que cualquier persona pueda ver o modificar información sensible.
- **Protección de datos personales:** Los datos registrados (conductores, vehículos, categorías, movimientos, etc.) deben mantenerse seguros dentro de la base de datos y no mostrarse sin autorización.
- **Integridad de la información:** Las operaciones de registro, cobro y control de sensores deben guardarse correctamente para evitar alteraciones o pérdidas.
- **Seguridad ante fallas externas:** En caso de un incendio o humedad detectada por los sensores, el sistema debe reaccionar automáticamente y notificar al administrador para prevenir daños.
- **Comunicación confiable con sensores:** Los sensores deben enviar información segura y estable para evitar errores que comprometan el funcionamiento del sistema.

3.3. Adaptabilidad

El sistema está diseñado para poder ajustarse a futuros cambios sin tener que rehacer todo desde cero. Esto incluye:

- **Nuevas categorías de vehículos:** Se puede agregar o modificar categorías de cobro sin afectar el funcionamiento principal.



- **Integración con más sensores:** Si en el futuro se quiere incluir cámaras, sensores de peso u otros dispositivos, el sistema puede ampliarse sin grandes cambios.
- **Actualización de reglas de negocio:** La tarifa, el cálculo del cobro o las funciones de seguridad pueden adaptarse fácilmente conforme cambien las necesidades.
- **Escenarios de crecimiento:** El sistema puede ajustarse para manejar más usuarios, más espacios de estacionamiento o incluso un segundo punto de control si el proyecto crece.

3.4. Rendimiento

El sistema está pensado para funcionar de manera rápida y fluida, incluso cuando hay varios vehículos entrando o saliendo al mismo tiempo. Las operaciones principales —como registrar ingresos, calcular tarifas o detectar señales de los sensores— se procesan casi al instante. Además, se busca que el sistema pueda manejar un crecimiento moderado sin volverse lento, ajustándose bien a un número mayor de vehículos, usuarios o dispositivos conectados.

4. Análisis de Requerimientos

4.1. Requerimientos funcionales



Código	Nombre del requerimiento	Caso de uso	Prioridad	Actor
RF1	El sistema debe permitir registrar vehículos al ingresar.	Registrar ingreso de vehículo (automático por sensor o manual).	Alta	Usuario / Sensor
RF2	El sistema debe asignar automáticamente un espacio disponible.	Asignar espacio de estacionamiento.	Alta	Usuario
RF3	El sistema debe mostrar los espacios libres y ocupados.	Visualizar estado del estacionamiento.	Mediana	Usuario
RF4	El sistema debe registrar la salida del vehículo.	Registrar salida y liberar espacio.	Mediana	Usuario / Sensor
RF5	El sistema debe calcular la tarifa según el tiempo y categoría del vehículo.	Generar cobro de peaje.	Alta	Administrador
RF6	El sistema debe permitir registrar y gestionar conductores.	Mantener datos de conductores (añadir, modificar, eliminar).	Alta	Administrador
RF7	El sistema debe generar reportes estadísticos automáticos.	Ver reportes de ocupación, ingresos y tiempos promedio.	Mediana	Administrador
RF8	El sistema debe permitir acceder con diferentes roles (admin / usuario).	Iniciar sesión y controlar permisos.	Alta	Administrador / Usuario
RF9	El sistema debe detectar automáticamente la entrada o salida mediante sensores.	Leer información de sensores en tiempo real.	Alta	Sensor
RF10	El sistema debe recibir alertas de sensores de humedad.	Activar techo de protección automática.	Mediana	Sensor
RF11	El sistema debe recibir alertas de sensores de fuego o llamas.	Enviar notificación de emergencia al administrador.	Alta	Administrador
RF12	El sistema debe permitir consultar el historial de ingresos y salidas.	Mostrar historial por fecha, placa o conductor.	Mediana	Administrador
RF13	El sistema debe permitir configurar las tarifas por categoría.	Gestionar categorías y precios.	Mediana	Administrador
RF14	El sistema debe asegurar los datos personales mediante validaciones.	Validar y proteger datos de usuarios registrados.	Alta	Administrador

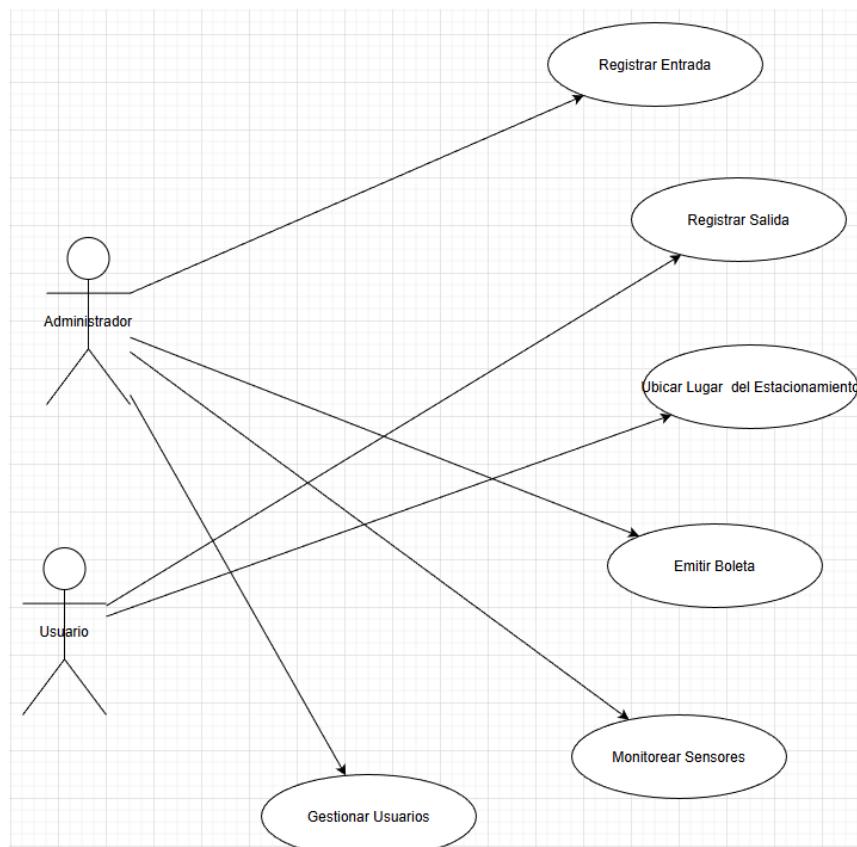
4.2. Cuadro de requerimientos no funcionales



Código	Categoría	Requisito No Funcional	Descripción
RNF01	Seguridad	Control de acceso	Solo usuarios registrados pueden ingresar al sistema.
RNF02	Seguridad	Manejo de credenciales	Las contraseñas deben estar almacenadas de forma segura (idealmente hash).
RNF03	Usabilidad	Interfaz intuitiva	La vista debe ser sencilla, de colores claros y entendible sin capacitación.
RNF04	Usabilidad	Retroalimentación inmediata	Cambios de color, mensajes emergentes y alertas claras.
RNF05	Rendimiento	Tiempo de respuesta	Las acciones principales deben ejecutarse en menos de 2 segundos.
RNF06	Rendimiento	Ligereza del sistema	El uso de CPU y memoria debe ser bajo en equipos estándar.
RNF07	Disponibilidad	Fallo de sensor	El sistema permite asignación manual sin detener operaciones.
RNF08	Disponibilidad	Conexión a BD	Debe mantener conexión estable con SQL Server o BD en la nube.
RNF09	Adaptabilidad	Escalabilidad de espacios	El sistema debe soportar agregar más sensores o espacios.
RNF10	Mantenibilidad	Patrón de arquitectura	El sistema debe usar arquitectura por capas y patrón Factory donde corresponda.
RNF11	Interoperabilidad	Integración con servicio de correo	Se usa SMTP externo sin necesidad de reprogramar componentes internos.
RNF12	Confiabilidad	Manejo de errores	Los errores deben ser capturados y mostrados sin afectar la operación.
RNF13	Portabilidad	Soporte de despliegue	Posibilidad de publicar en servidor local o nube (API + SQL Cloud).
RNF14	Privacidad	Protección de datos	Los datos personales (DNI, correo) deben manejarse con confidencialidad.
RNF15	Respaldo	Copias de seguridad	La base de datos debe permitir respaldos automáticos en caso de despliegue en la nube.

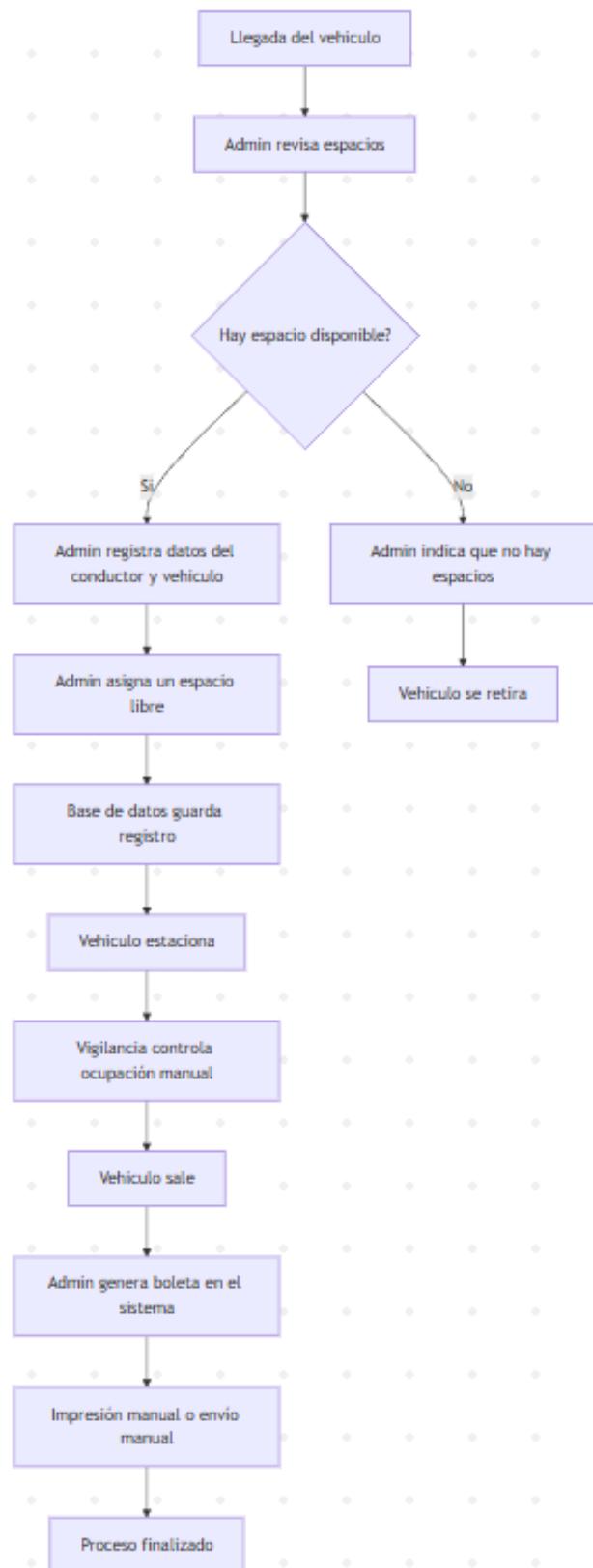


5. Vistas de Caso de Uso

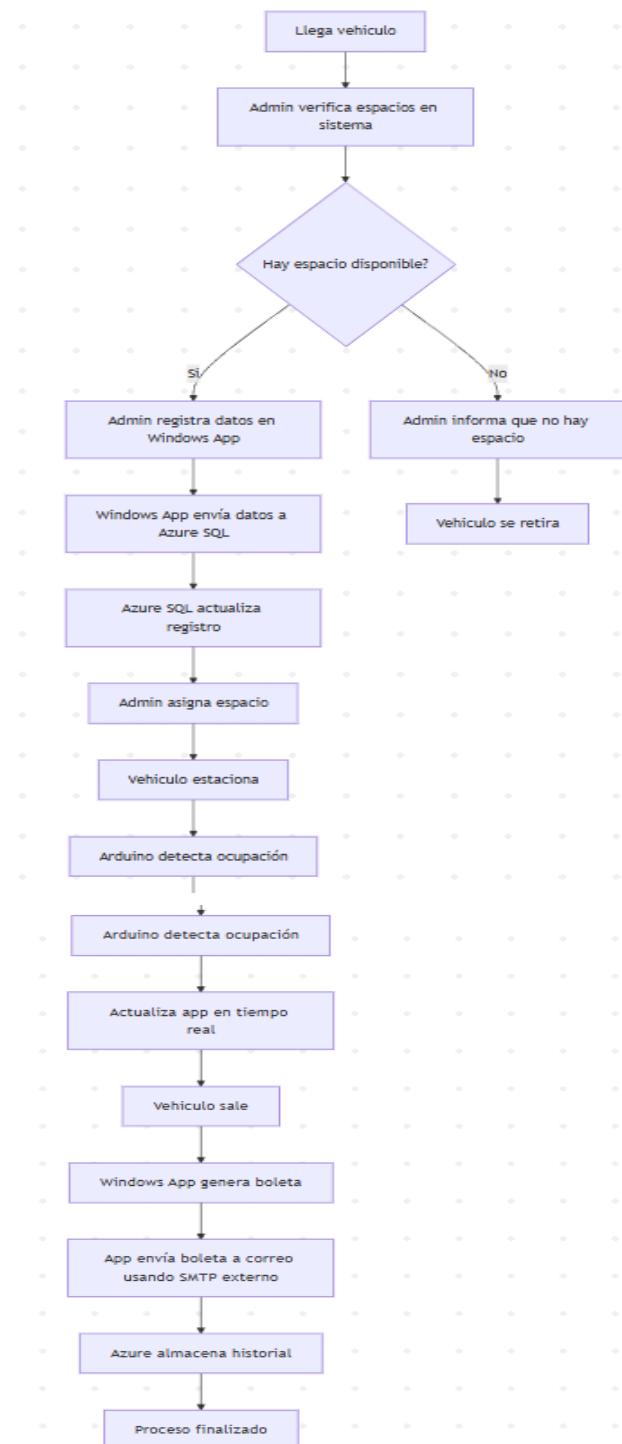


6. Vista de Procesos

6.1. Diagrama de Proceso Actual



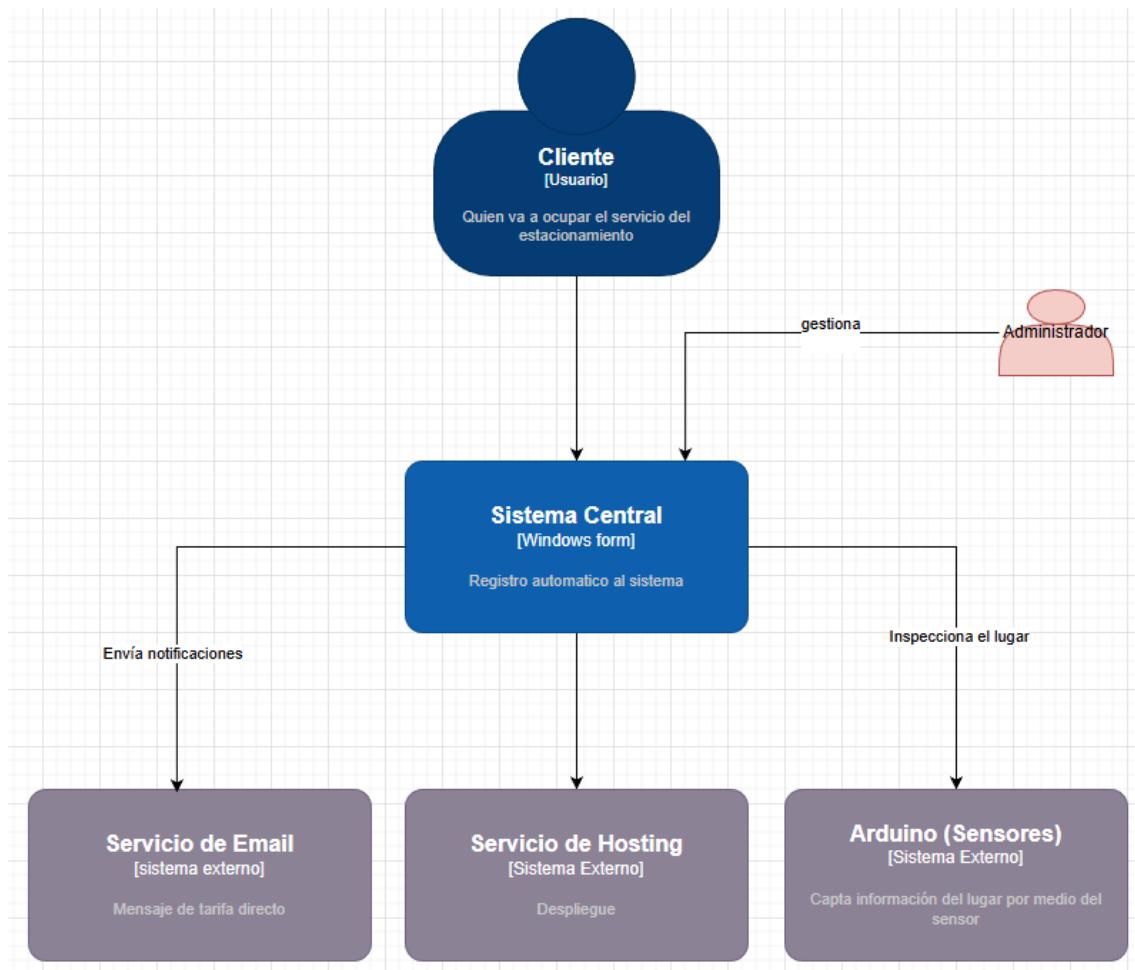
6.2. Diagrama de Proceso Propuesto



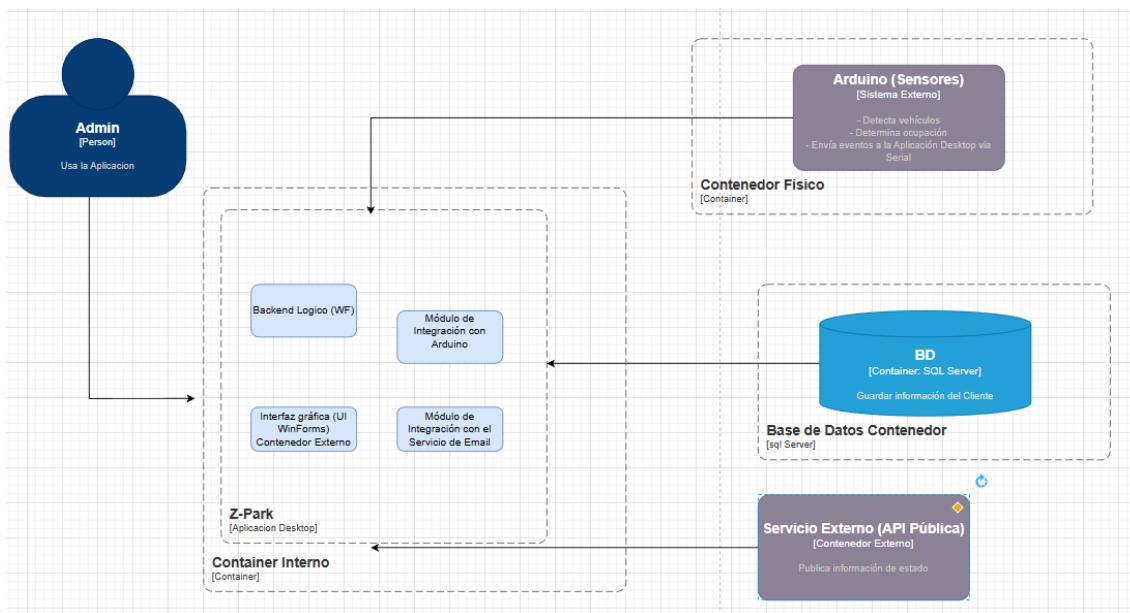


7. Vista de Despliegue

7.1. Diagrama de contextos

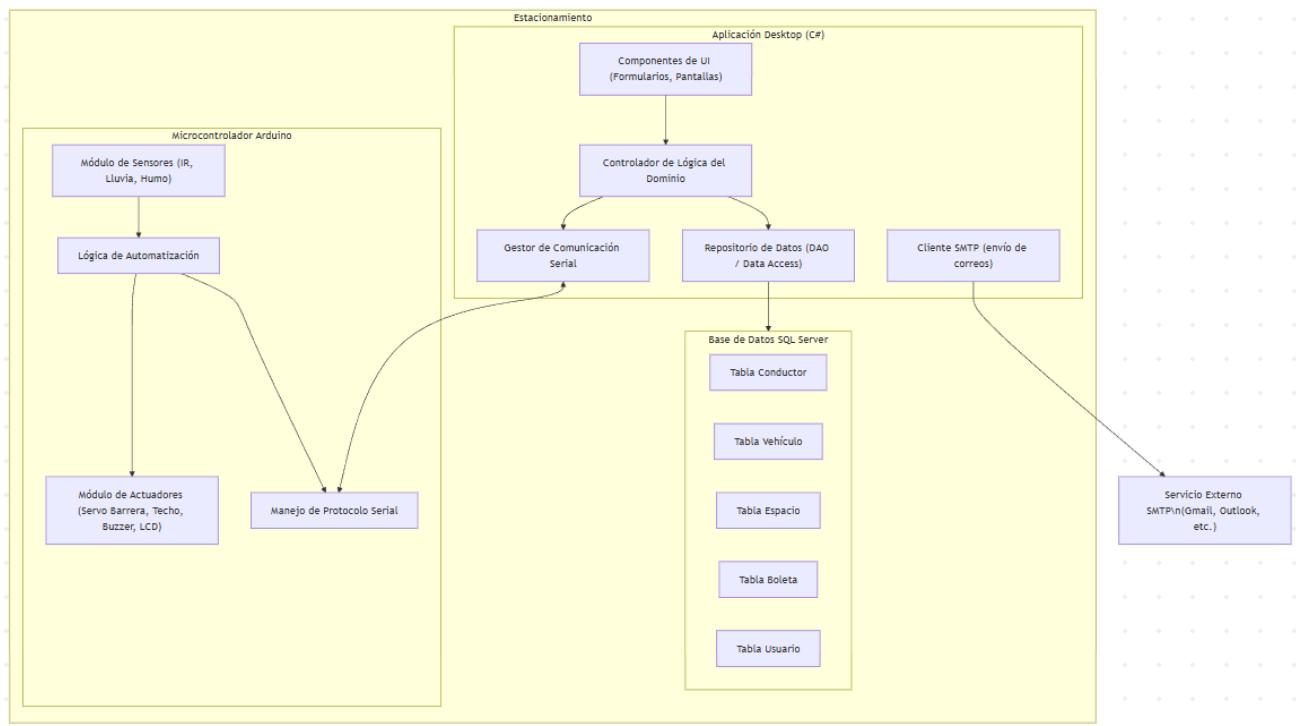


7.2. Diagrama de Contenedor

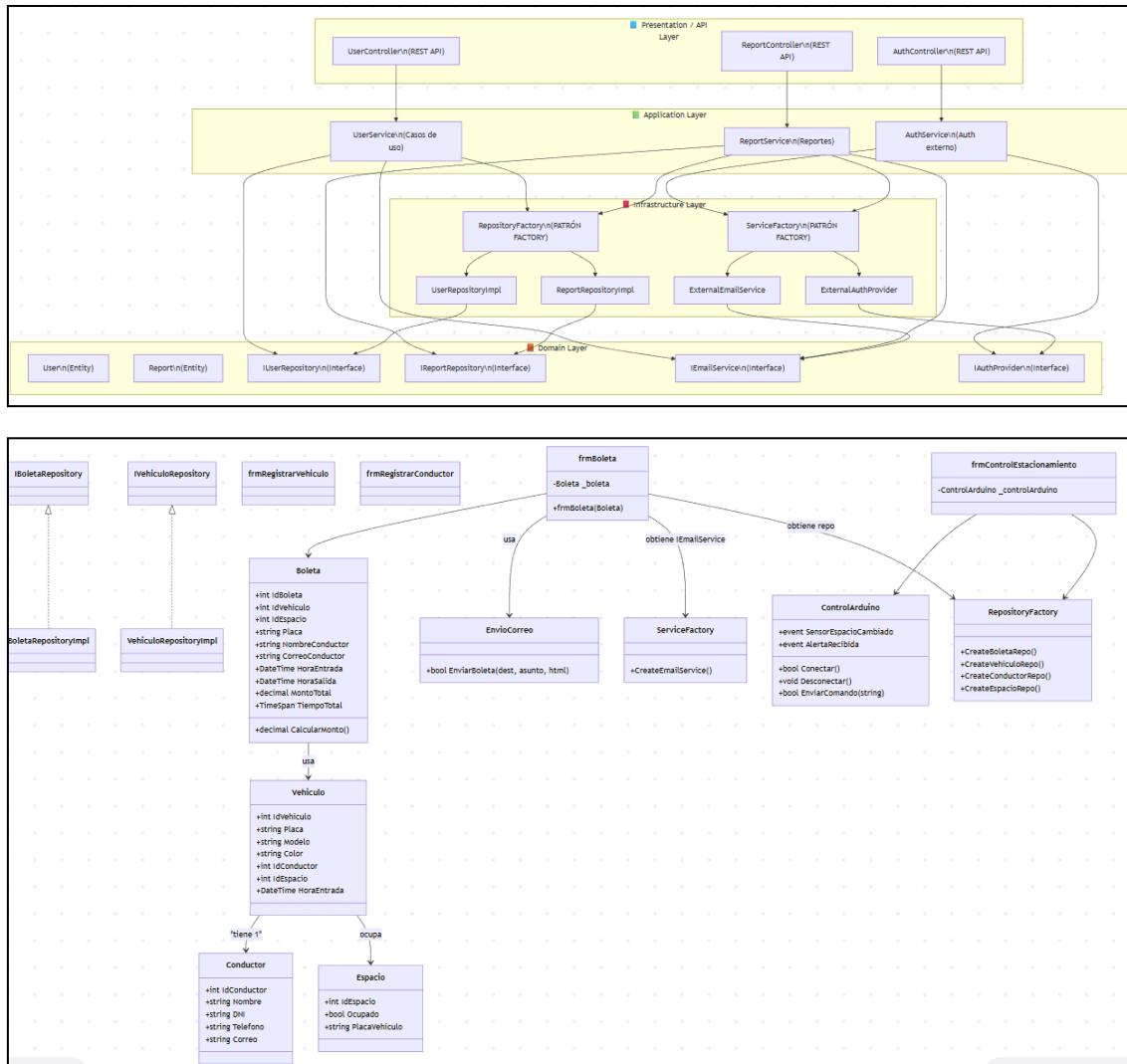


8. Vista de Implementación

8.1. Diagrama de Componentes



8.2. Diagrama de Codificación



9. Vista de Datos

9.1. Diagrama Entidad Relación

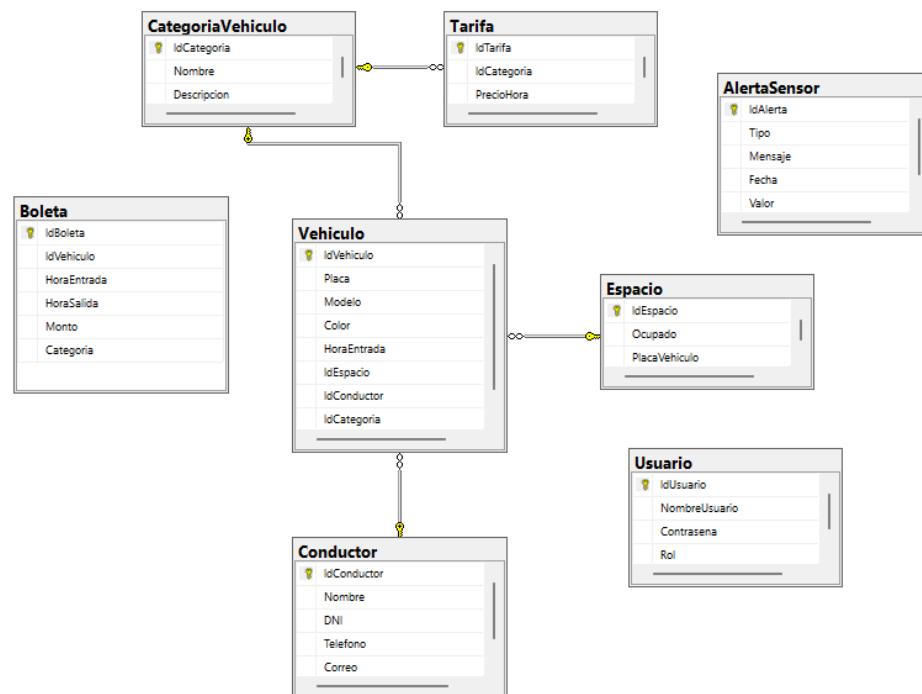


Diagrama Lógico (Entidad - Relacion)

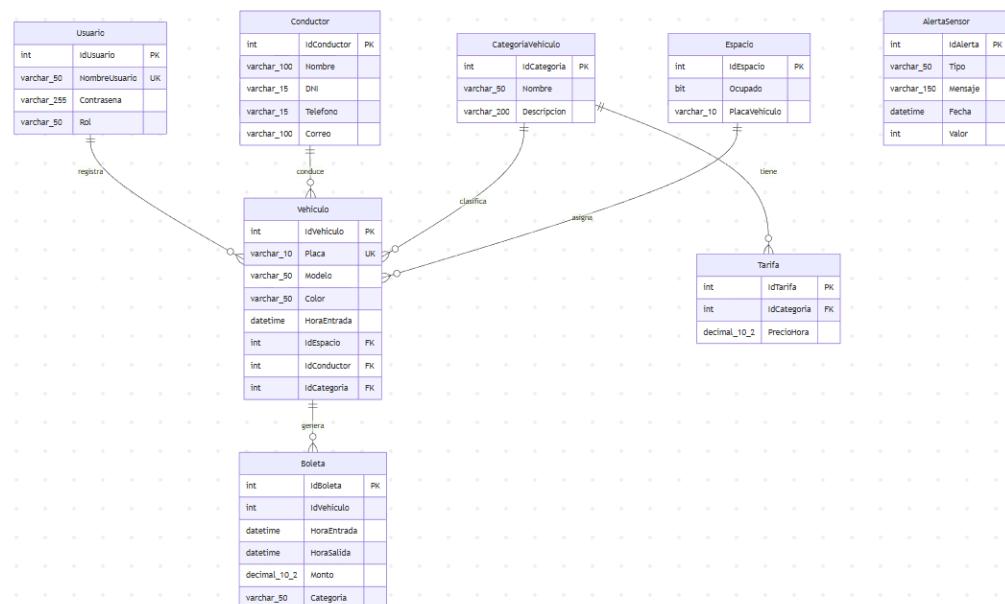
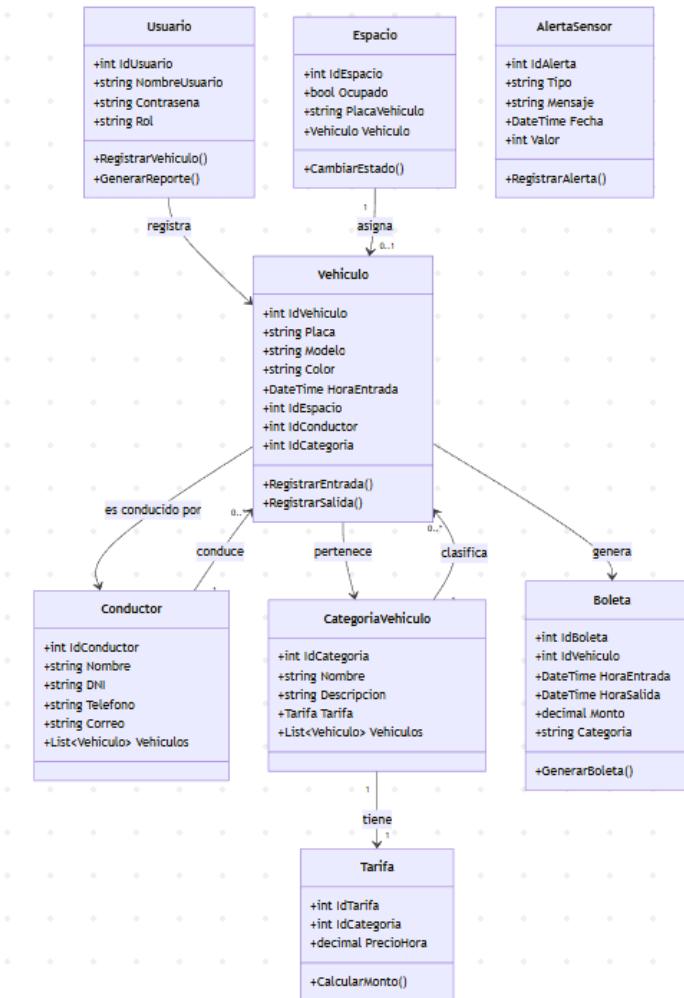


Diagrama Lógico



10. Calidad

10.1. Escenarios de Seguridad

- Nombre:** Seguridad ante acceso no autorizado
- Estímulo:** Un usuario intenta acceder al sistema utilizando credenciales inválidas o sin permisos.
- Fuente:** Actor externo malintencionado o usuario sin autorización.
- Ambiente:** Sistema en operación normal.
- Restricciones:** Los usuarios deben estar previamente registrados en la tabla Usuario y autenticados con credenciales válidas.

Respuesta esperada:

- El sistema bloquea el acceso.
- Se registra el intento fallido (log).
- Se muestra un mensaje indicando que las credenciales no son válidas.
- No se debe revelar información interna del sistema.



Métricas de aceptación:

- 100% de los intentos fallidos no deben permitir acceso.
- El sistema debe responder en menos de 2 segundos ante un intento de acceso no autorizado.

10.2. Escenario de Usabilidad

- **Nombre:** Facilidad de uso en la interfaz de control del estacionamiento
- **Estímulo:** El operador del estacionamiento requiere vincular una placa detectada por el sensor a un espacio.
- **Fuente:** Operador del sistema (usuario humano).
- **Ambiente:** Uso cotidiano con sensores activándose automáticamente.
- **Restricciones:** El operador puede no tener conocimientos técnicos avanzados.

Respuesta esperada:

- El sistema muestra una notificación clara cuando un sensor detecta un vehículo.
- La interfaz permite ingresar la placa rápidamente.
- El botón rojo o verde comunica el estado de manera visual inmediata.

Métricas de aceptación:

- La operación completa (clic → ingreso de placa → vinculación) debe realizarse en menos de 10 segundos.
- Los colores y mensajes deben ser identificables en una primera vista.
- Al menos el 90% de usuarios deben entender el flujo sin capacitación formal.

10.3. Escenario de Adaptabilidad

- **Nombre:** Ampliación del estacionamiento
- **Estímulo:** La empresa decide añadir 10 nuevos espacios físicos con sensores adicionales.
- **Fuente:** Administrador del sistema.
- **Ambiente:** Sistema funcionando en modo producción.
- **Restricciones:** No se debe interrumpir el funcionamiento de los espacios existentes.

Respuesta esperada:



- El sistema permite registrar nuevos espacios en la base de datos.
- La lógica de lectura de sensores soporta fácilmente más entradas.
- La interfaz puede escalar visualmente (nuevos botones).
- No es necesario reescribir completamente el código del backend.

Métricas de aceptación:

- Agregar nuevos espacios no debe tomar más de 5 minutos en configuración.
- El sistema debe tolerar un incremento del 50% de sensores sin afectar al rendimiento perceptible.

10.4. Escenario de Disponibilidad

- **Nombre:** Continuidad operativa ante falla del sensor
- **Estímulo:** Un sensor Arduino falla o deja de enviar datos.
- **Fuente:** Hardware del sistema.
- **Ambiente:** Horas de operación del estacionamiento.
- **Restricciones:** El operador debe poder continuar su trabajo.

Respuesta esperada:

- El sistema muestra un indicador de “sensor desconectado”.
- El operador puede asignar el espacio manualmente sin dependencia del sensor.
- No debe bloquearse la asignación ni liberación de espacios.

Métricas de aceptación:

- El sistema debe detectar la pérdida de comunicación en menos de 5 segundos.
- El operador debe poder usar el modo manual sin errores al 100%.

10.5. Otro

- **Nombre:** Generación y envío de boleta sin retrasos
- **Estímulo:** Un operador genera una boleta al momento que el vehículo abandona su espacio.
- **Fuente:** Operador del sistema.
- **Ambiente:** Fin de una sesión de estacionamiento, carga normal de uso.



- **Restricciones:** El correo depende de un servicio externo (SMTP / Gmail / Outlook).
El sistema no controla la disponibilidad del proveedor.

Respuesta esperada:

- La boleta se genera en menos de 1 segundo.
- El envío por correo se intenta inmediatamente.
- El sistema muestra un mensaje claro de éxito o fallo.
- Si el envío falla por el servicio externo, no afecta la generación de la boleta ni la salida del vehículo.

Métricas de aceptación:

- 95% de las boletas enviadas deben tardar menos de 5 segundos.
- La interfaz debe seguir siendo fluida y no congelarse durante el envío.