UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Desarrollo de un Sistema Web de Administración de Cocheras para la Mutual Ingeniería de Santa Fe

Proyecto Final de Ingeniería en Informática

Alumno: Gaspar Ezequiel Oberti
Director: Ing. Gustavo Doldán
Codirectora: Ing. Josefina Morais

Índice

Índice de Tablas

Índice de Figuras

Glosario

1. Tarifa: Cantidad de tiempo en la que se cobran distintos precios de acuerdo a si es hora pico o no. Un estacionamiento tiene una o más tarifas.
2. Tipo de tarifa: Subdivisión de tarifa en la que se cobra distintos precios de acuerdo a la fracción de tiempo de estadía. Una tarifa tiene uno o más tipos de tarifas.
3. Estadía: Estancia o permanencia durante cierto tiempo en el estacionamiento.
4. Registro: Acción de registrar una estadía en el estacionamiento.
5. Registro medido: Registro de un vehículo que permanecerá un período de tiempo.
6. Estado abierto: Estado de un registro medido en el que el vehículo permanece en el estacionamiento. Solo posee fecha de ingreso.
7. Estado cerrado: Estado de un registro medido en el que el vehículo ya se retiró del estacionamiento. Posee fecha de ingreso y de egreso y total cobrado.
8. Registro mensualizado: Registro de un vehículo perteneciente a un socio que abona mensualmente.

Capítulo 1

1. Introducción

1.1 Contexto y Justificación

La Caja de los Profesionales de la Ingeniería es una es una entidad con personería jurídica de derecho público no estatal, sin fines de lucro. Tiene como fines esenciales proporcionar a todos los profesionales inscriptos en los Colegios Profesionales, que se encuadren en las disposiciones de las leyes provinciales 4889 y 6729, los beneficios de la cooperación mutua para asegurarles asistencia y seguridad social en condiciones dignas y justas. A diferencia de otros regímenes jubilatorios estatales o privados es administrada por sus propios afiliados y desde 1958, presta servicios asistenciales para el profesional y sus familiares con el fin de una mejor convivencia basada en la seguridad de no quedar en el desamparo en momentos de adversidad (Caja de Ingeniería, 2019).

La afiliación es obligatoria para todos los profesionales que se hallen inscriptos en los siguientes Colegios Profesionales de la Provincia de Santa Fe:

- Colegio de Arquitectos – Distritos 1, 5 y 6 (ley 10.653).
- Colegio de Ingenieros Agrónomos – 1º y 3º Circunscripción (ley 10.780).
- Colegio de Profesionales de la Agrimensura – Distrito Norte (ley 10.781).
- Colegio Profesional de MMO y Técnicos – Distrito 1 (ley 10.946).
- Colegio de Profesionales de la Ingeniería Civil – Distrito 1 (ley 11.008).
- Colegio de Ingenieros Especialistas - Distrito 1

En el proceso de evolución emprendido por la Caja desde el año 2009, para lograr reformular el sentido de la misma y en pos de lograr otorgar mayores y mejores servicios y beneficios, surge el emprendimiento de creación de un nuevo ente que integre todas aquellas actividades que desde el seno de la Caja no se podían realizar. Es así como empezó a madurar la idea de la creación de una Mutual, que sea distinta a las que ya existían en el entorno y que dé respuesta a las necesidades que tenían muchos de los afiliados. Siempre pensando en una única consigna: lograr un mejor bienestar de los afiliados.

El 18 de diciembre de 2012 se logró obtener la habilitación para el funcionamiento de la ASOCIACIÓN MUTUAL AFILIADOS A LA CAJA DE INGENIERÍA (A.M.A.C.I.), otorgada por el Instituto Nacional de Asociativismo y Economía Social –INAES- y regida por la Ley Orgánica para Asociaciones Mutuales N° 20321.

Avanzando en el tiempo, la institución logró dar su primer y gran paso: la creación de **Farmacia Mutual**. El 16 de abril de 2012 se adquirió la habilitación para el funcionamiento de este servicio.

En abril de 2014 quedó inaugurado, junto al Complejo Integrado de Caja de Ingeniería, el **Salón de Eventos**. Un espacio pensado para asociados y sus momentos más importantes. El 16 de junio de 2014 se dio apertura a los **Consultorios Odontológicos**, que

dotaron a todos los asociados de una prestación odontológica de excelente calidad, con un cuerpo de profesionales de primer nivel y un equipamiento tecnológico de vanguardia.

En octubre de 2015 lograron la apertura del departamento de **Turismo Mutual**. A partir de octubre de 2016, se comenzó con el Servicio de Cochera por el cual la Mutual se hizo cargo de la administración de las playas de estacionamiento propiedad de la Caja de Ingeniería a cambio de un alquiler mensual.

Gracias a todos los servicios brindados por Mutual Ingeniería hoy en día cuenta con un padrón de 10.683 asociados activos y adherentes.

En la actualidad las cocheras se encuentran distribuidas en el centro de la ciudad. Poseen un movimiento diario constante y tienen distintas tarifas dependiendo el tipo de servicio que se les brinde, ya sea mensualizado o estacionamiento temporario.

Son 5 playas de estacionamiento que hacen un total de 172 dársenas:

Cochera	**Dársenas**
25 de Mayo 2171	30
Obispo 2441	41
San Martín 1743	40
San Martin 3017	24
Crespo 2770	37

Tabla 1: Cocheras y sus respectivas dársenas.

La administración es llevada a cabo por un empleado que anota de forma manual el número de patente y el horario de ingreso. Al retirarse el vehículo, el precio a cobrar se calcula según si es estacionamiento medido o de un mensualizado.

La mayoría de los sistemas de administración de cocheras en el mercado actual son provistos por empresas privadas. Tienen las características de ser cerrados y garantizan un desempeño óptimo en la administración y gestión de usuarios, precios y facturación. Además brindan herramientas informativas para permitir optimizar el negocio.

Estos sistemas están compuestos de dos partes, un conjunto de dispositivos electrónicos y una solución informática.

Dispositivos electrónicos: Existen distintos dispositivos (Figura 1): barreras, columna de ingreso/egreso y sensores (dispositivos electrónicos de tipo 1), donde cada uno cumple una función específica. Al ingresar un vehículo y posarse por el primer sensor se activa la columna de ingreso, la cual registra la fecha y hora de entrada y emite el ticket con su correspondiente código de barra. Al retirarse el ticket se abre la barrera que permite la entrada a la cochera y cuando el vehículo pasa por el segundo sensor la barrera se baja. Al salir el dueño del vehículo debe presentar al empleado o en la columna de egreso el ticket para leer el código de barra y que este realice el cálculo del costo. Una vez realizado el pago se abre la barrera.

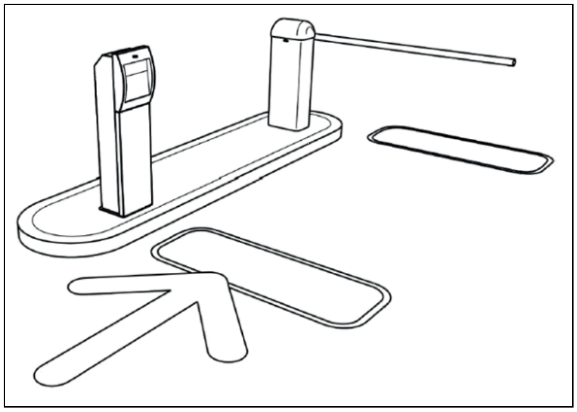

Figura 1: Dispositivos electrónicos (Gestión de Parking, 2019).

Solución informática: Consiste de un sistema relacionado con los dispositivos electrónicos de tipo 1. Al ingresar, el ticket emitido por la columna de ingreso actualiza automáticamente el sistema descontando en uno la cantidad de lugares disponibles en el estacionamiento y empezando a acumular el tiempo de estadía. Al realizar el egreso, la persona encargada de manipular el sistema solicita el ticket al dueño del vehículo, lo lee con el lector de código de barra y realiza el cobro manual en efectivo u otro medio de pago. En caso de que la cabina de cobro esté dentro de la playa, el dueño del vehículo deberá presentar el ticket en la columna de egreso para validar la salida pero anteriormente deberá realizar el pago. De lo contrario no se abrirá la barrera. El sistema incluye instalación y configuración de equipos electrónicos, administración de usuarios y sus permisos, configuración de precios según tiempo de permanencia (Gestión de Parking, 2019).

Para la Mutual Ingeniería, el principal problema es que no existe un control real de ingresos y egresos en las cocheras debido a que se realiza un registro manual. También se dificulta saber con certeza la disponibilidad de estacionamiento teniendo que hacer una revisión constante por parte del empleado lo que implica la ausencia en la cabina de ingreso/egreso. Todo esto conlleva a pérdida de tiempo, registros erróneos y actividad económica ineficiente.

Debido a que los sistemas de terceros poseen un número determinado de funcionalidades, no permiten a la institución cubrir necesidades particulares ni realizar mejoras futuras sin depender de un externo.

En este trabajo se pretende desarrollar una herramienta propia que sea escalable y automatice la administración de las cocheras teniendo la capacidad de la validar códigos de descuentos obtenidos por los socios de la mutual mediante una aplicación móvil existente. También se busca un sistema capaz de diferenciar entre tipos de asociados de la mutual, y que pueda verificar que el vehículo realmente ingresó, comparando si existe una imagen con la misma fecha y hora que el ticket del vehículo. La imagen es obtenida de las cámaras de seguridad las cuales fotografían los ingresos. Con escalable se refiere específicamente a que el sistema es capaz de ir incorporando nuevos módulos hasta poder, en un futuro, agregar y configurar dispositivos electrónicos de tipo 1, así como automatismos más complejos como por ejemplo apertura de barreras mediante llamadas telefónicas o detección automática de patentes.

Los usuarios serán los empleados de las cocheras que registrarán cada ingreso y se encargarán de gestionar el pago de los estacionamientos. Los administradores podrán realizar las altas, bajas y modificaciones de mensualizados, tarifas y promociones, además de poder emitir informes solicitados por la gerencia. Los desarrolladores de la institución serán los que cuenten con los permisos de administrador.

A partir de este proyecto se intentará solucionar los errores que provienen del accionar humano, ahorrar el dinero que implica tercerizar y llevar un control correcto en las transacciones monetarias con la implementación de tickets fiscales. A su vez, será de gran ayuda para la comunidad optimizando tiempo y espacio, así como también para Mutual Ingeniería debido a que el proyecto será implementado en producción y además se podrán obtener informes que servirán como bases para realizar mejoras en la organización.

1.2 Objetivos y Alcance

1.2.1 Objetivo General

Desarrollar un sistema web para la automatización de la administración de cocheras para la Mutual Ingeniería.

1.2.2 Objetivos Específicos

- Diseñar e implementar módulo de administración de distintos tipos de usuario.
- Diseñar e implementar módulo de administración de distintos tipos de tarifas y promociones.
- Diseñar e implementar módulo de administración de distintos tipos de estacionamientos (tipo afiliado, directores, empleados, proveedores) para asignarles horarios y tiempos de estacionamientos gratuitos.
- Diseñar e implementar módulo de integración con dispositivos electrónicos de tipo 2 (impresora fiscal, lector de código de barras) y sistemas externos (aplicación móvil, sistema de cámaras de seguridad).
- Diseñar e implementar módulo de informes estadísticos.
- Diseñar e implementar interfaz gráfica.

1.2.3 Alcance

Se desarrollará un sistema web versionado mediante gitlab para la administración de cocheras que será capaz de comunicarse con una WEB API de Mutual Ingeniería para la obtención de descuentos.

1.3 Metodología

En este proyecto se utilizará un modelo de desarrollo secuencial, el modelo en cascada, que considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y los representa como fases separadas del proceso tales como la especificación de requerimientos, el diseño del software, la implementación y las pruebas, cada una con su respectiva documentación. A estas se le agregará una etapa de investigación y capacitación en la cual se profundizará sobre el uso de la plataforma de desarrollo, la web API y la integración con sistemas externos y dispositivos electrónicos de tipo 2. Esto será necesario debido a que el alumno no ha trabajado aún con estas herramientas.

Si bien en la práctica estas etapas se superponen y proporcionan información unas a otras, la idea principal es que la siguiente fase no debe empezar hasta que la fase previa haya finalizado. El principal problema es que debido a su inflexibilidad al dividir el proyecto en distintas etapas se hace difícil responder a los cambios en los requerimientos del cliente. En este caso particular el cliente cuenta con un equipo de ingenieros en sistemas que entienden la problemática actual y son capaces de describir los requerimientos funcionales y no funcionales, por lo que se espera un claro y correcto informe de los mismos al inicio del proyecto a través de entrevistas personales. Es por esto que el modelo en cascada es una buena opción para la realización de este proyecto ya que es improbable que los requerimientos cambien radicalmente. De todas maneras, en caso de surgir errores y omisiones en los requerimientos originales se deberán repetir etapas previas del modelo (Sommerville, 2005).

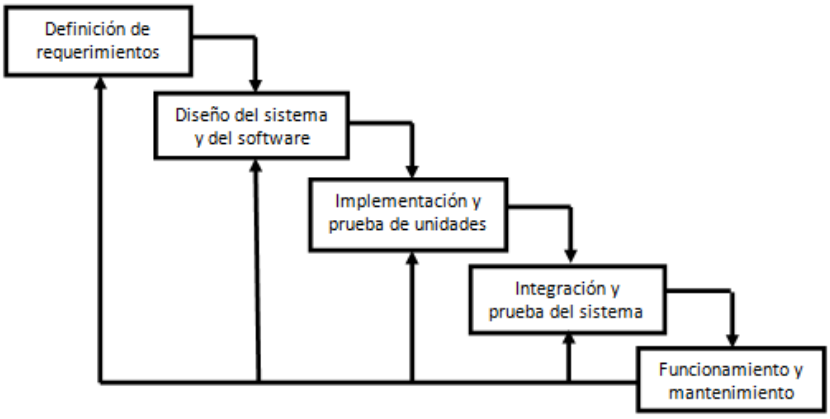

Figura 2: Modelo en cascada (Sommerville, 2005).

Debido a que se trata de un sistema web, se desarrollará mediante el método MVC (Modelo Vista Controlador). Este patrón de arquitectura separa la aplicación en estos tres componentes permitiendo lograr una separación de intereses y reutilización del código con el objetivo de facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Información general de ASP.NET Core MVC, 2019). El modelo contiene únicamente los datos sin la lógica que describe cómo puede presentarse los datos a un usuario. La vista

presenta al usuario los datos del modelo, pero sin saber que significan ni lo que el usuario puede hacer para manipularlos. El controlador está entre la vista y el modelo escuchando los sucesos desencadenados por la vista y ejecutando un método del modelo (Patrón de diseño de modelo-vista-controlador, 2019).

Cada una de las etapas contendrá la siguiente información:

- Investigación y capacitación: En esta etapa se llevará a cabo una interiorización sobre las herramientas a utilizar. Esto incluye un estudio del lenguaje C# y de la plataforma .net y cómo realizar la integración de dispositivos electrónicos de tipo 2.
- Análisis y diseño: En esta etapa se realizará el Documento de Requerimientos que contendrá el conjunto de necesidades que dieron lugar al proyecto. Además se realizarán todos los diagramas que detallarán los diagramas de base de datos, diagrama de clases y diagramas de casos de usos. Estos se utilizarán como guía para el desarrollo.
- Desarrollo: En esta etapa se transcribe a código todo lo plasmado en la etapa de diseño y por cada módulo realizado se llevan a cabo sus pruebas unitarias.
- Pruebas y puesta en funcionamiento: En esta etapa se integrarán todos los módulos individuales y se realizarán un conjunto de procesos para comprobar que el software está acorde a sus especificaciones y cumple las necesidades del cliente. Una vez concluidas las pruebas se lleva a cabo la redacción del informe final.

En cada una de estas etapas se irá documentando cada acción y sus respectivos resultados con el objetivo de ir conformando un documento que sea de utilidad para eventuales correcciones, mantenimiento futuro y ampliaciones al sistema. También se irá confeccionando un manual de usuario a medida que se vaya desarrollando cada módulo que será entregado a los encargados de manipular el sistema.

Capítulo 2

2. Análisis y Definición de Requerimientos

2.1 Obtención de los Requerimientos

En esta etapa se define qué es lo que el sistema debe hacer y sus propiedades deseables. Se establece un conjunto de objetivos que el sistema debe cumplir. Estos objetivos son los requerimientos y fueron obtenidos mediante entrevistas principalmente. También se llevó a cabo un análisis de documentación previa proveída por la empresa para entender el contexto y las necesidades y a partir de estos definir los requerimientos (Sommerville, 2005).

Las entrevistas se realizaron con miembros del área de sistemas de la empresa quienes fueron los encargados de comunicar los deseos de sus superiores, así como también de llegar a un consenso respecto al alcance del proyecto. Al ser de tipo abiertas, las entrevistas no tuvieron un programa predefinido sino que a partir de una serie de preguntas fueron surgiendo otras cuestiones que desembocaron en la comprensión general del sistema.

2.2 Casos de Uso

2.2.1 Diagrama de Casos de Uso

Los casos de uso son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente se utilizan para describir modelos de sistemas orientados a objetos identificando el tipo de interacción y los actores involucrados. El conjunto de casos de uso representa todas las posibles interacciones a representar en los requerimientos del sistema (Sommerville, 2005).

Para este proyecto se detectaron dos actores que intervienen en el sistema, administrador y playero. El administrador tiene permiso para todas las funcionalidades y es el que tiene a cargo la configuración inicial del sistema así como la modificación de ésta. El playero es la persona que se encarga de registrar los ingresos y egresos de las diferentes cocheras pero no puede modificar información que no sea relacionada a los registros que realiza.

En la figura 3 se muestra el diagrama de casos de uso.

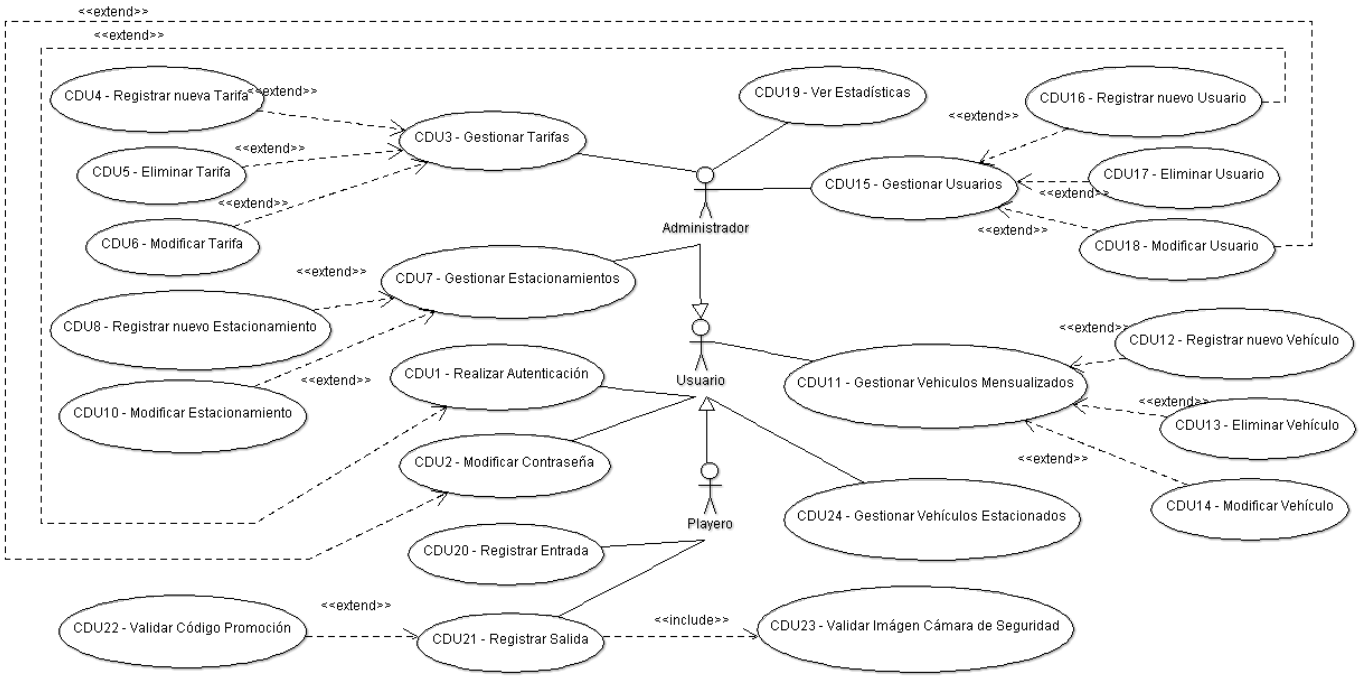

Figura 3: Diagrama de casos de uso.

2.2.2 Especificaciones de Casos de Uso

A modo de ejemplo a continuación se detalla la especificación del CU01. Este es el primer caso de uso con el cual el administrador o el playero inicia el sistema. Las demás especificaciones se encuentran en el anexo 8.1.

Nombre: Realizar Autenticación	**Nro 01**
Actores: Usuario	**Versión:** 1.0
Objetivo: Realizar la autenticación del usuario en el sistema y establecer el perfil de acceso en la aplicación.	
Precondiciones:	
Postcondiciones:	Éxito: ● Actor autenticado.
	Fracaso: ● Imposibilidad de autenticar el actor.
Flujo Principal	**Flujo Alternativo**
1. El actor desea iniciar sesión en la aplicación.	
2. El sistema solicita nombre de usuario y contraseña.	

3. El actor ingresa los campos y presiona el botón Iniciar Sesión.	
4. El sistema valida que sea un usuario registrado.	
5. La autenticación se realiza con éxito. El sistema habilita las funcionalidades del usuario específico.	5.A. El usuario no es válido. 5.A.1. El sistema muestra un mensaje indicando usuario o contraseña inválidos. 5.A.2. Se retorna al paso 2 del flujo principal.
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión: CDU16 - Alta Usuario. Se incorpora un nuevo usuario al sistema.	
Requerimientos Especiales: ● La contraseña debe mostrarse con *.	
Observaciones:	
Importancia: Alta.	
Urgencia: Alta.	

Tabla 2: Especificación CU1

2.3 Especificación de los Requerimientos

2.3.1 Requerimientos Funcionales

- El sistema deberá administrar distintos tipos de usuarios, estacionamientos, tarifas, promociones.
- El sistema deberá generar tickets fiscales.
- El sistema deberá realizar informes estadísticos.
- El sistema deberá verificar promociones obtenidas mediante aplicación móvil.
- El sistema deberá verificar la existencia de una imagen que asegure el ingreso de un vehículo.

2.3.2 No funcionales

- La sistema deberá ser web, en lenguaje de programación C# .net, utilizando base de datos SQL Server Express.
- El sistema se deberá vincular con una WEB API de Mutual Ingeniería mediante el intercambio de mensajes JSON o XML alojado en https://api.mutualingenieria.org.ar
- El sistema deberá estar documentado y versionado mediante gitlab.
- El sistema deberá ser escalable para poder incorporar nuevos módulos en el futuro.

2.3.3 Exclusiones

- No se realizará el reconocimiento de patentes mediante procesamiento de imágenes.
- No se realizará la vinculación con dispositivos electrónicos de tipo 1.
- No se vinculará con ninguna plataforma de pago online.
- No se contemplará otro medio de pago que no sea efectivo.
- No se realizarán automatismos para apertura de barreras mediante llamadas telefónicas.

2.3.4 Supuestos

- Se dispone de un conjunto de imágenes obtenidas por las cámaras de seguridad para verificación de ingreso.

2.3.5 Requerimientos Específicos

Definidos a partir de los casos de uso, los requerimientos específicos delinean las funcionalidades que deberá tener el sistema. Siguiendo con el mismo ejemplo, a continuación se describe el primer requerimiento funcional, la autenticación de usuario. Los demás se encuentran especificados en el anexo 8.2.

RF01 - Autenticación de usuario:
El sistema deberá permitir el logueo a usuarios que estén registrados con su respectivo nombre de usuario (DNI) y contraseña. Los posibles usuarios son Administrador o Playero. Si el usuario no está registrado se mostrará un mensaje notificando esa situación.

Capítulo 3

3. Diseño del Sistema

3.1 Arquitectura de Software

El proceso de diseño está relacionado con el establecimiento de un marco estructural básico que identifica los principales componentes de un sistema y las comunicaciones entre estos. Diseñar y documentar la arquitectura de software tiene las siguientes ventajas:

1. Comunicación con los stakeholders: La arquitectura constituye una presentación de alto nivel del sistema que puede usarse como punto de discusión por varios stakeholders.
2. Análisis del sistema: Hacer explícita la arquitectura del sistema en una etapa temprana del desarrollo del sistema requiere un análisis.
3. Reutilización a gran escala: La arquitectura del sistema es a menudo la misma para sistemas con requerimientos similares y por lo tanto pueden soportar reutilización a gran escala.

Las etapas del diseño de la arquitectura fuerzan a los diseñadores a considerar aspectos de diseño claves en etapas tempranas del proceso. La arquitectura afecta al rendimiento, solidez, grado de distribución y mantenibilidad de un sistema, es por esto que el estilo y la estructura pueden depender de los requerimientos no funcionales del sistema rendimiento, protección, seguridad, disponibilidad y mantenibilidad.

El resultado de un proceso de diseño de la arquitectura es un documento que puede incluir varias representaciones gráficas del sistema junto con un texto descriptivo asociado. Algunos ejemplos de estas representaciones pueden ser modelo estructural estático (muestra los subsistemas y componentes como unidades separadas), modelo de interfaz (define los servicios ofrecidos para cada subsistema a través de una interfaz), modelo de relaciones (muestra el flujo de datos entre subsistemas), entre otros.

La organización del sistema refleja la estrategia básica usada para estructurar dicho sistema. Existen varias estrategias como: modelo de repositorio, modelo cliente-servidor, modelo de capas, etc. En este proyecto se utilizó la segunda opción, el modelo cliente-servidor porque así lo dispuso la empresa interesada.

Este modelo se organiza como un conjunto de servicios y servidores asociados, más unos clientes que acceden y usan los servicios. La ventaja más importante del modelo cliente-servidor es que es una arquitectura distribuida, es decir que el procesamiento de la información se distribuye en varias computadoras en vez de en una sola. En la figura 4 se muestra una aplicación estructurada en tres capas. La capa de presentación está relacionada con la presentación de la información al usuario y con la interacción con éste. La capa de procesamiento está relacionada con la implementación de la lógica de la aplicación, y la capa de gestión de datos está relacionada con las operaciones sobre la base de datos. El uso de la arquitectura de tres capas permite optimizar la transferencia de información

entre el servidor web y el servidor de la base de datos ya que para la comunicación entre estos sistemas se pueden usar protocolos de bajo nivel muy rápidos (Sommerville, 2005).

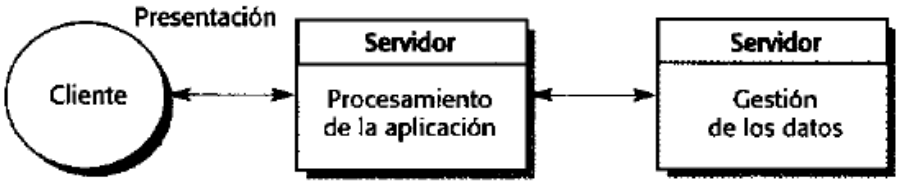

Figura 4: Modelo cliente-servidor de tres capas (Sommerville, 2005).

En la actualidad las técnicas y lenguajes de programación de servidores han avanzado de tal forma de permitir desarrollos orientados a objetos y modulares (MVC).

MVC (modelo, vista, controlador) es un patrón de arquitectura que separa la aplicación en estos tres componentes permitiendo lograr una separación de intereses y reutilización del código con el objetivo de facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento (Información general de ASP.NET Core MVC, 2019). El modelo contiene únicamente los datos sin la lógica que describe cómo puede presentarse los datos a un usuario. La vista presenta al usuario los datos del modelo, pero sin saber que significan ni lo que el usuario puede hacer para manipularlos. El controlador está entre la vista y el modelo escuchando los sucesos desencadenados por la vista y ejecutando un método del modelo (Figura 5) (Patrón de diseño de modelo-vista-controlador, 2019).

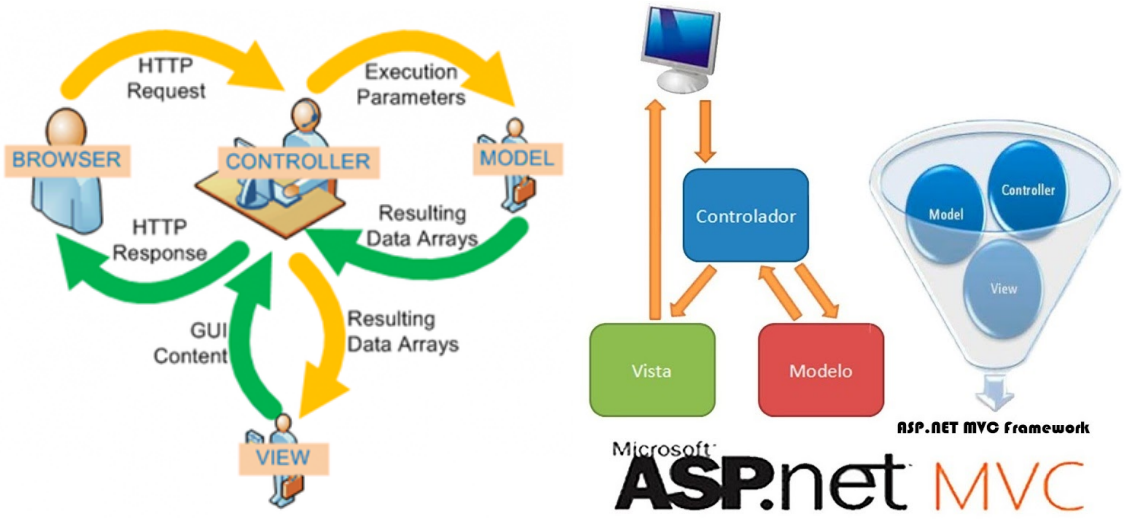

Figura 5: Patrón MVC.

3.3 Análisis y Diseño Orientado a Objetos

Los requerimientos deberían redactarse en lenguaje natural debido a que tienen que ser comprendidos por personas que no son técnicos expertos. Sin embargo, pueden expresarse de forma más técnica documentando la especificación del sistema como un conjunto de modelos. Estos modelos son representaciones gráficas que son a menudo más comprensibles que las descripciones en lenguaje natural. Pueden desarrollarse distintos modelos para representar el sistema desde diferentes perspectivas. Por ejemplo:

- Perspectiva externa: modela el entorno del sistema.
- Perspectiva del comportamiento: modela el comportamiento del sistema.
- Perspectiva estructural: modela la arquitectura del sistema.

Una abstracción simplifica y resalta las características más relevantes del sistema. Es por esto que existen distintos modelos que se basan en distintas aproximaciones de abstracción. En este proyecto se usó el modelado de objetos, el cual combina el modelado de comportamiento y de la estructura. Se optó por los modelos de objetos porque pueden utilizarse para representar tanto los datos del sistema como su procesamiento y porque son útiles para mostrar cómo se clasifican las entidades y cómo se componen de otras entidades. Pero sobre todo porque normalmente simplifican la transición entre el diseño y la programación.

En el diseño orientado a objetos se modelan entidades del mundo real utilizando clases de objetos. No deben incluirse detalles de los objetos individuales en el sistema (instanciaciones de la clase) (Sommerville, 2005).

3.3.1 Diagrama de Clases

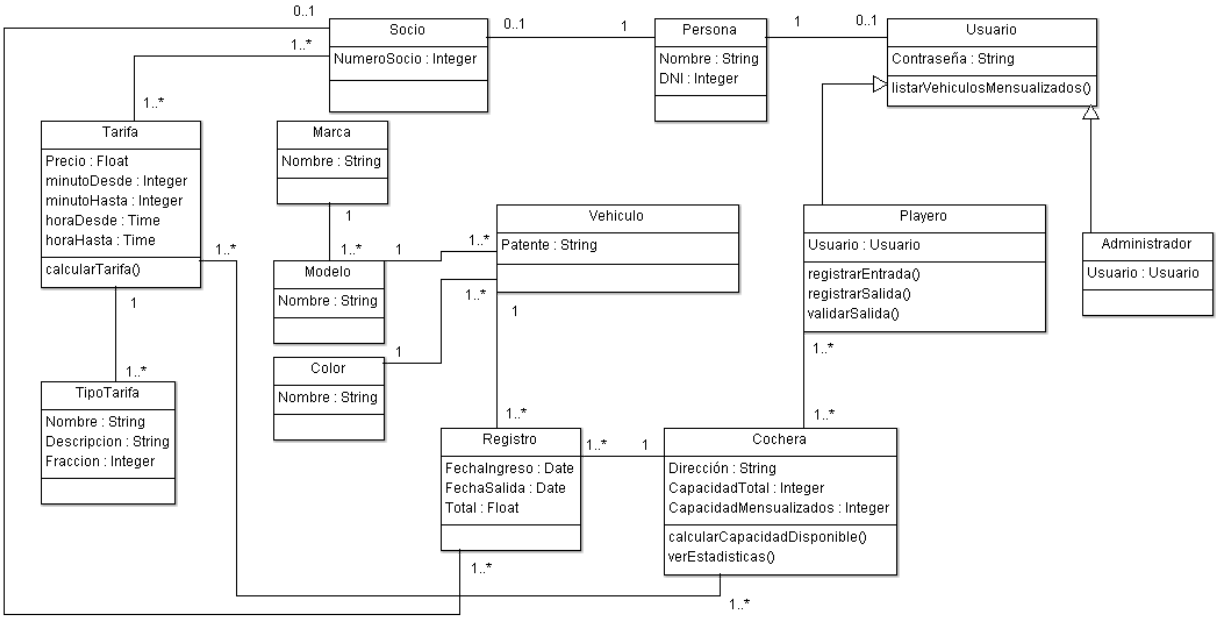

Figura 6: Diagrama de clases.

3.4 Base de Datos

Una base de datos relacional es un tipo de bases de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Esta es una forma intuitiva y directa de representar datos en tablas.

En el modelo relacional las estructuras de datos lógicas están separadas de las estructuras de almacenamiento físicas. Por ejemplo se puede cambiar el nombre de un archivo de base de datos pero el nombre de las tablas almacenadas en el no se cambian.

Las ventajas del modelo relacional, además de que proporciona una forma estándar de representar y consultar datos, son:

1. Coherencia de los datos: Garantiza que varias instancias de bases de datos tengan los mismos datos todo el tiempo.
2. Confirmación y atomicidad: Maneja políticas muy estrictas respecto a los cambios permanentes. No se pueden eliminar elementos que dependen de otros.
3. Procedimientos almacenados: Poseen bloques de códigos a los que se puede acceder con una simple llamada favoreciendo la reutilización de código.
4. Bloqueo y simultaneidad: Reduce la posibilidad de conflictos al impedir que otros usuarios y aplicaciones accedan a los datos mientras estos se actualizan. (¿Qué es una base de datos relacional?, (2020))

Una base de datos SQL Server es de tipo relacional y consta de una colección de tablas en las que se almacena un conjunto específico de datos estructurados. Una tabla contiene una colección de filas, también denominadas tuplas o registros, y columnas, también denominadas atributos. En cada columna se almacena un tipo determinado de información (Bases de datos, (2020)).

3.4.1 Diagrama de Base de Datos

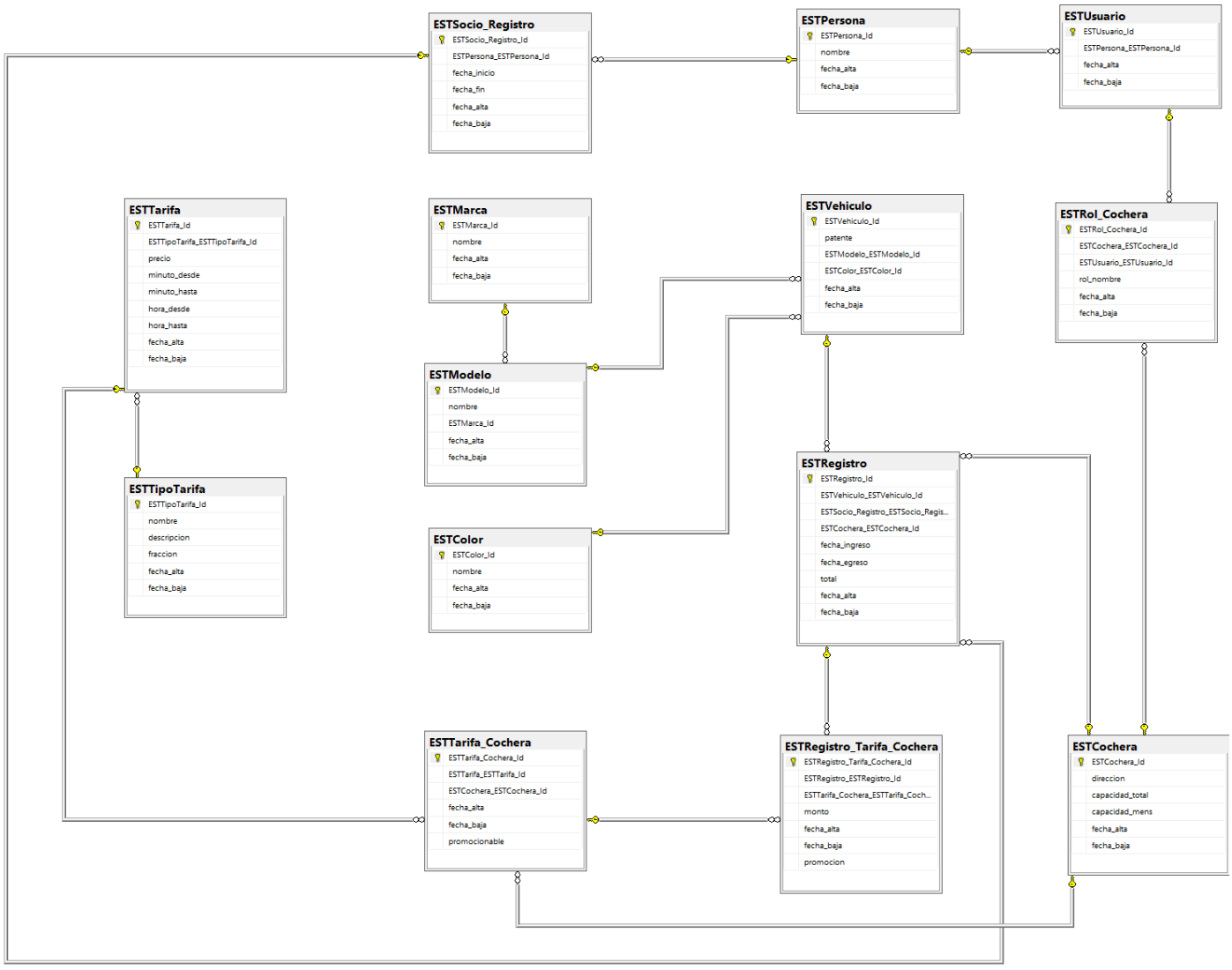

Figura 7: Diagrama de base de datos.

3.4.2 Diccionario de Datos

El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema en el que se definen con precisión los campos y relaciones. El diccionario de datos es un buen complemento del diagrama de base de datos, y el objetivo principal es dar la precisión suficiente para evitar malas interpretaciones o ambigüedades.

A continuación se muestra un ejemplo con la tabla ESTCochera. Los demás se especifican en el anexo 8.3.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTCochera_Id	int	x		Id único de la cochera.	
direccion	varchar(45)			Nombre de la cochera.	

capacidad_total	int			Cantidad máxima de cocheras.	
capacidad_mens	int			Cantidad máxima de cocheras disponible para mensualizados.	
fecha_alta	datetime			Fecha alta cochera.	
fecha_baja	datetime			Fecha baja cochera.	

Tabla 3: Base de Datos. Tabla ESTCochera.

3.5 Diseño de la Interfaz Gráfica

El diseño de sistemas informáticos abarca varias actividades que van desde el diseño de hardware hasta el de la interfaz de usuario. Un diseño cuidadoso de la interfaz de usuario es parte fundamental del proceso de diseño y es crítico para la confiabilidad del sistema ya que muchos de los errores del usuario son causados por el hecho de que las interfaces no consideran las habilidades de los usuarios reales. Al no poder acceder a algunas características del sistema los usuarios cometerán errores y sentirán que el sistema les dificulta en vez de ayudarlos.

Al momento de diseñar las interfaces de usuario se deben tener en cuenta las capacidades físicas y mentales de las personas que utilizarán el software (memoria limitada a corto plazo, amplio rango de capacidades, diferentes preferencias de interacción, etc). Estos factores humanos son la base de los principios de diseño de la tabla 4, los cuales se aplican a todos los diseños de interfaces de usuario.

Principio	Descripción
Familiaridad del usuario	La interfaz debe utilizar términos y conceptos obtenidos de la experiencia de las personas que más utilizan el sistema. Los usuarios no deben ser obligados a adaptarse a una interfaz solo porque sea conveniente implementarla.
Uniformidad	Siempre que sea posible la interfaz debe ser uniforme en el sentido de que las operaciones comparables se activen de la misma forma. Los comandos y menús deben tener el mismo formato, los parámetros deben pasarse a todos los comandos

	de la misma forma, etc. Esto reducirá el tiempo de aprendizaje de los usuarios.
Mínima sorpresa	Si una acción en algún contexto provoca un cambio particular, es razonable esperar que la misma acción en un contexto diferente cause un cambio similar. Si sucede algo completamente diferente el usuario se sorprende y confunde.
Recuperabilidad	La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores. Se deben incluir recursos como por ejemplo confirmación de acciones destructivas, posibilidad para deshacer, etc.
Guía de usuario	Cuando ocurren errores la interfaz debe proporcionar retroalimentación significativa y características de ayuda sensible al contexto.
Diversidad de usuarios	La interfaz debe proporcionar características de interacción apropiadas para los diferentes tipos de usuarios del sistema. Los usuarios casuales necesitan interfaces que los guíen, pero los usuarios potenciales requieren métodos abreviados que permitan una interacción más rápida con el sistema.

Tabla 4: Principios de diseño de Shneiderman

Una interfaz de usuario coherente debe integrar la interacción del usuario y la presentación de la información. Existen distintos tipos de interacción como selección de menús, rellenado de formularios, lenguaje de comandos, etc. Cada uno tiene sus ventajas y desventajas y según a quién se dirige el sistema conviene usar uno u otro. La presentación de la información puede ser de forma directa en formato texto o gráficamente, pero una buena pauta de diseño es mantener por separado el software requerido para presentar la información. El enfoque MVC es una forma efectiva de permitir representaciones múltiples de datos.

Por último se debe pensar detenidamente en los colores utilizados. El color puede mejorar las interfaces ayudando a los usuarios a comprender y manejar la complejidad. Algunas pautas para la utilización efectiva del color son:

1. Limitar el número de colores utilizados y ser conservador en la forma de utilizarlos: No deben utilizarse más de cuatro o cinco colores en una ventana y no más de siete en una interfaz del sistema.
2. Utilizar un código de colores para apoyar la tarea que el usuario esta tratando de llevar a cabo: Si los usuarios tienen que identificar instancias anómalas se deben resaltar esas instancias.
3. Utilizar el código de colores de forma consistente y uniforme: Si en una parte del sistema muestra los errores en rojo en todas las demás partes debe mostrarlos igual (Sommerville, 2005).

Para realizar el diseño de una interfaz existen varias maneras de hacerlo, en este proyecto se decidió por el uso de wireframes que son representaciones visuales de una interfaz que esquematizan el contenido y el ordenamiento del sistema.

A continuación se mostrarán a modo de ejemplo los wireframes pertenecientes a la gestión de estacionamientos, los demás se detallan en el anexo 8.4.

Interfaz principal en la cual se inicia el sistema. Aquí se detallan los estacionamientos que existen. Contiene las siguientes partes:

- Vehículos: Al seleccionar esta pestaña aparecen el listado con todos los vehículos que se encuentran estacionados en un estacionamiento o cochera específico.
- Tarifas: Esta pestaña contiene las distintas tarifas que existen para un cierto estacionamiento.
- Usuarios: Aquí se especifican los usuarios que están relacionados con un estacionamiento particular.
- Nuevo Estacionamiento: Este botón permite agregar un nuevo estacionamiento al sistema. Agregando los datos solicitados del lado derecho de la figura 8 y presionando guardar si pasa las validaciones se crea con éxito.
- Modificar Estacionamiento: Presionando el botón de la columna acciones en el listado se abre el formulario con la información correspondiente para editar. Al presionar guardar, del mismo modo que al agregar uno nuevo, si pasa las validaciones modifica con éxito.
- Eliminar Estacionamiento: Al clickear el segundo botón en la columna acción se abre un alert pidiendo confirmación. Si la cochera no tiene ningún vehículo estacionado al momento de querer realizar la acción se elimina con éxito.

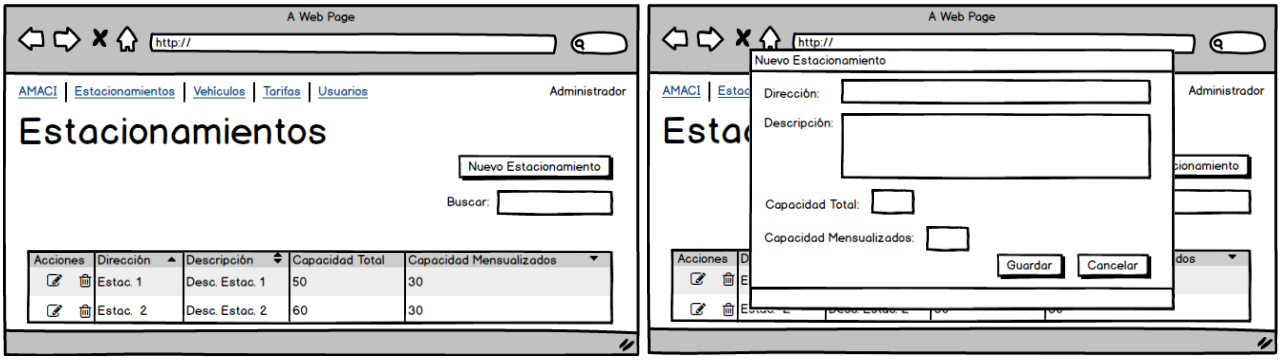

Figura 8: Pantalla gestionar estacionamientos y registrar nuevo estacionamiento.

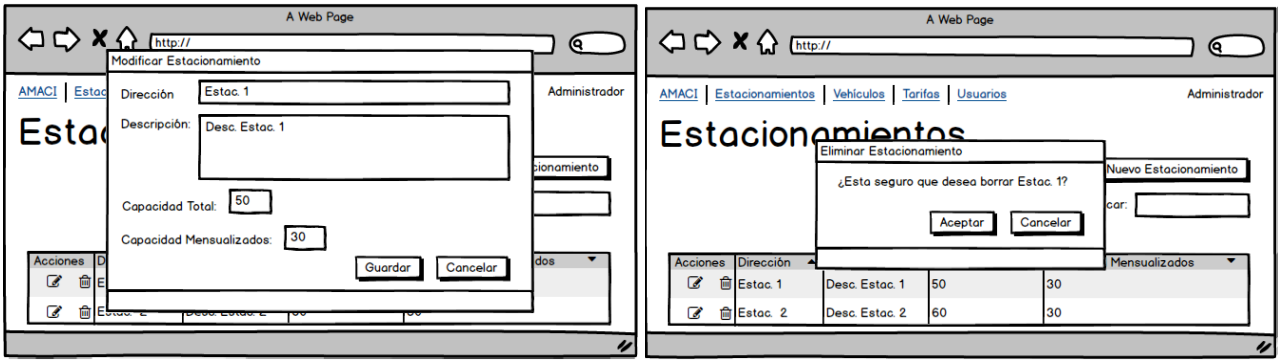

Figura 9: Pantalla modificar estacionamientos y eliminar estacionamientos.

Capítulo 4

4. Análisis de Tecnologías

4.1 Investigación de la plataforma de desarrollo y lenguaje

Debido a que las tecnologías fueron propuestas por el cliente, no se llevará a cabo una estudio de alternativas sino que se realizará una explicación resumida de cada una de ellas con el objetivo de dejar en claro los conceptos más importantes.

- HTML:
  Significa en inglés HyperText Markup Language y es un lenguaje de marcas de hipertexto que hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que define una estructura básica y un código para la definición de contenido de una página web (HTML Tutorial, 2019).

- JavaScript:
  Es un lenguaje de programación interpretado que se utiliza principalmente del lado del cliente implementado como parte del navegador permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Todos los navegadores modernos interpretan el código JS integrado en las páginas web (JavaScript Introduction, W3schools, 2019).

- Bootstrap:
  Creado inicialmente para como una solución interna para Twitter y posteriormente liberado al público, es el framework más popular de HTML, CSS y JavaScript para desarrollo responsive tanto móvil como web. Facilita la maquetación y ofrece herramientas para los sitios web se vean bien en toda clase de dispositivos (¿Qué es Bootstrap?, 2019).

- AJAX:
  Significa en inglés Asynchronous JavaScript And XML. Es una técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan en el cliente,es decir en el navegador mientras se comunica de forma asíncrona con el servidor en segundo plano. Esto permite que se puedan realizar cambios sobre las páginas sin necesidad de recargarlas (AJAX Introduction, 2019).

- JQuery:

Es una librería multiplataforma de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, y agregar interacción con la técnica AJAX. Es software libre y de código abierto y su principal ventaja es la reducción a la hora de escribir código (JQuery Tutorial, 2019).

- C#:
  Es un lenguaje de programación Orientada a Objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza un modelo de objetos similar al de Java (Introducción C#, 2019).

- Framework .NET:
  Es una creación de Microsoft para satisfacer el creciente mercado de los negocios en entornos web para poder competirle a la plataforma de Java y a los diferentes frameworks de PHP. El objetivo de .NET es ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones (¿Qué es el .NET? ¿Para qué sirve?, 2019).

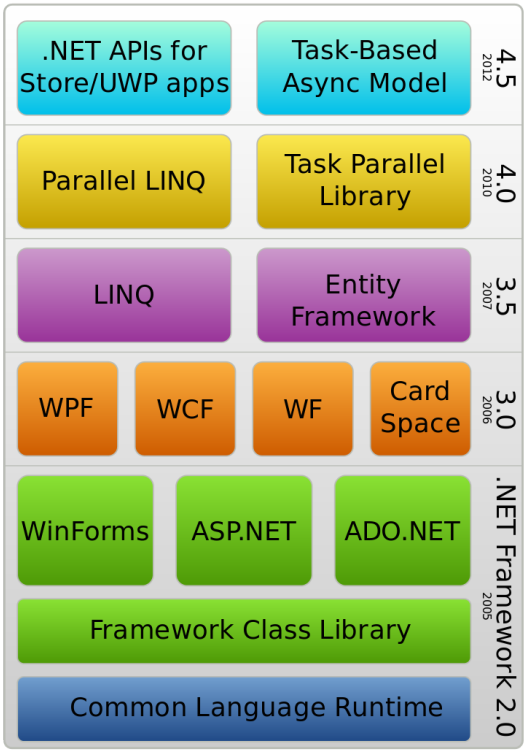

Figura 10: Estructura de .NET Framework.

.NET Framework es una plataforma de desarrollo para compilar aplicaciones que proporciona muchos servicios, API y estructuras de datos fáciles de usar que abstraen el sistema operativo (.NET Framework Documentation, 2019). El Framework .NET consta de dos componentes principales: Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable.

Los servicios que ofrece son:

- Administración de la memoria: CLR lo proporciona para que no lo tengan que realizar los desarrolladores.
- Sistema de tipos comunes: Define el sistema de tipos (comunes a todos los lenguajes) en vez de que lo haga el compilador.
- Biblioteca de clases extensa: Evita tener que escribir cantidades extensas de código.
- Marcos y tecnologías de desarrollo: Ofrece bibliotecas para determinadas áreas de desarrollo (ASP.NET para aplicaciones web, ADO.NET para acceso a datos, etc.).
- Interoperabilidad entre lenguajes: Se emite un código intermedio que se compila en tiempo de ejecución para que las rutinas escritas en un lenguaje sean accesibles para otros lenguajes.
- Compatibilidad de versiones: La mayoría de las aplicaciones se ejecutan sin modificaciones en una versión posterior.

- Ejecución en paralelo: Resuelve conflictos entre versiones por lo que pueden coexistir varias versiones de las aplicaciones al mismo tiempo (Introducción a .NET Framework, 2019).

- ASP.NET:
  Es un marco web gratuito para crear sitios y aplicaciones web mediante HTML, CSS y JavaScript. También puede crear API web y usar tecnologías en tiempo real como Web Sockets (Información general de ASP.NET, 2019).
  ASP.NET ofrece tres alternativas estables de trabajo:
  - Formularios Web Forms: Permite crear sitios web dinámicos de manera sencilla basado en eventos de arrastrar y colocar.
  - ASP.NET MVC: Permite crear sitios web dinámicos basándose en patrones y proporciona control total sobre el marcado para un desarrollo más agradable y ágil.
  - ASP.NET Web Pages: Con la sintaxis Razor provee una manera rápida y accesible de combinar código con HTML para crear contenido web dinámico.

  Los tres marcos, al basarse en .NET Framework comparten las funcionalidades básicas de .NET y ASP.NET, por ejemplo ofrecen un modelo de seguridad de inicio de sesión, comparten las mismas funciones para administrar solicitudes, etc. Además los marcos no son totalmente independientes sino que elegir uno no impide el uso del otro. Por ejemplo, las partes orientadas al cliente pueden desarrollarse en MVC para optimizar el marcado mientras que el acceso a datos y las partes administrativas pueden desarrollarse en formularios web para aprovechar los controles de datos y el acceso a datos simples.

- ASP.NET MVC:
  - Modelo: Representa el estado de la aplicación y cualquier lógica de negocios. Las vistas fuertemente tipadas normalmente usan tipos ViewModel para que contengan los datos para mostrar en esa vista. El controlador crea y rellena estas instancias del ViewModel desde el modelo.

- Vistas con Razor: Las vistas presentan el contenido a través de una interfaz de usuario. Razor es un motor de vistas que posee sintaxis de marcado para insertar código en servidor en páginas web que combina marcado de Razor, C# y HTML.
- Controlador: Controlan la interacción del usuario, trabajan con el modelo y selecciona una vista para representarla (Referencia de sintaxis de Razor para ASP.NET Core, 2019) (Información general de ASP.NET Core MVC, 2019).

- SQL Server Express:
  Es un sistema de gestión de base de datos relacional desarrollado por Microsoft y posee su propio lenguaje llamado Transact-SQL. Esta es una implementación del estándar ANSI del lenguaje SQL. Posee soporte de transacciones, procedimientos almacenados y permite trabajar en modo cliente-servidor.
  Existen varias ediciones donde cada una posee distintas características: Enterprise, Developer, Standard, Express, SQL Azure. En el caso de este proyecto se utilizará la Express que es una versión gratuita y posibilita la creación de bases de datos limitadas con características básicas pero que servirán perfectamente para el objetivo que se persigue (Microsoft SQL Server, 2019).

- Gitlab:
  Es una plataforma libre donde se puede alojar código fuente de forma privada o pública que permite trabajar a nivel colaborativo mediante un controlador de versiones GIT. La ventaja por sobre GiTHub es que cuenta con una versión gratuita donde se pueden alojar proyectos privados. También posee planes privados que permite disponer de todo el poder de GitLab y sus herramientas sin invertir tiempo en configuración (Todo lo que debes saber sobre Gitlab, la mejor alternativa a Github, 2019).

- Visual Studio:
  Es un conjunto de herramientas de desarrollo para la generación de aplicaciones web, móviles y de escritorio mediante distintos lenguajes. Estos lenguajes utilizan las funciones de .NET Framework para simplificar el desarrollo. Todos utilizan el mismo IDE (entorno de desarrollo integrado) facilitando la creación de soluciones en varios lenguajes.
  Existen diferentes versiones, para este proyecto se utilizará la versión 2019 Community por ser eficiente y gratis para estudiantes, colaboradores de código abierto y usuarios particulares (¿Qué es y para qué sirve Visual Studio 2017?, 2019) (Visual Studio 2019, 2019).

4.2 Investigación de Web API de Mutual Ingeniería

Para la comunicación entre la aplicación y los datos se utilizará una API. Esta representa una comunicación entre componentes de software. Se realiza mediante un conjunto de llamadas a ciertas bibliotecas que ofrecen determinados servicios. Estos acceden a distintos objetos o recursos mediante peticiones HTTP.
Específicamente la API que se utilizará en este proyecto (API de Mutual Ingeniería, 2019) permite integrar los sistemas existentes de Mutual Ingeniería. Se divide en dos partes, en una se detalla el encabezado que deben tener las consultas así como también las respuestas de error que se podrían manejar. En la otra parte se describen los distintos componentes que la conforman, como por ejemplo Usuario, Socio, Cochera, Tarifa, etc. De cada una de estos se ofrecen información de las distintas solicitudes.

4.3 Investigación dispositivos de tipo 2 y sistemas externos

- Impresora térmica:
  Es un dispositivo que no utiliza cartuchos de ningún tipo sino que en su lugar utiliza un papel especial termosensible que al contacto se vuelve de color negro. Al no consumir

tinta resulta más económica que una impresora común pero la impresión dura menos tiempo.
En este proyecto este tipo de impresora se utilizará para emitir el comprobante con el que el usuario será capaz de constatar que su vehículo se encuentra dentro del lugar (¿Qué diferencia hay entre impresora térmica y una matricial?, 2019).

- Impresora fiscal:
  Es un dispositivo que sirve para emitir y guardar documentos fiscales con el objetivo de evitar la evasión de impuestos presentados a la Dirección General de Impuestos. El programa de facturación del negocio se comunica con la impresora mediante un micro código que interpreta y guarda los datos de la factura en la placa fiscal y luego imprime el ticket con todas las regulaciones del país. La impresora guarda la información de las ventas gravables y extensas en una memoria que está protegida por un precinto de seguridad que solo puede ser abierta por un técnico autorizado.
  La ventaja que ofrecen estas impresoras es que la información fiscal queda almacenada en la memoria de la impresora y no hace falta usar facturas preimpresas (¿Qué es una impresora fiscal y cómo funciona?, 2019). En este proyecto se utilizarán para emitir la factura una vez que se realiza el pago de la estadía.

- Lector de código de barra:
  Es un dispositivo óptico-electrónico que emite rayos de luz láser (simples u omni-direccionales) los cuales al ser reflejados hacia la fuente de orígen son detectados por un receptor especializado que interpreta códigos de barra que se procesan como datos relevantes contenidos en una imagen. Un código de barra es una secuencia de líneas de distintos grosores y separación que almacena información.
  El dispositivo es un periférico de entrada y los datos que escanea los envía por medio de un cable usb a la computadora (El lector de código de barras, 2019).

Capítulo 5

5. Desarrollo

5.1 Metodología

La metodología se basa en el patrón Modelo, Vista, Controlador (MVC) pero con la particularidad de que se trabajan en proyectos separados. En primer lugar existe proyecto que contiene las clases y por otro parte existen dos proyectos independientes (API y FRONT) que utilizan el proyecto de clases.

En el proyecto de clases se definen las clases identificadas en la etapa de diseño con sus respectivas propiedades y métodos que seguirán el paradigma orientado a objetos.

En el proyecto API se define la lógica de negocio, en la que se delinean las reglas mediante las cuales el sistema interacciona con la base de datos. Esta interacción puede llevarse a cabo de dos maneras según las reglas internas de la empresa:

- Consulta directa: Utilizada para los request de tipo GET.
- Procedimiento almacenado: Utilizada para los request de tipo POST (agregar y editar) y DELETE.

Los servicios HTTP, se gestionan usando el protocolo homónimo, proporcionan una respuesta a las solicitudes del cliente y muestran los datos de forma sencilla en formatos como JSON o XML. Para las pruebas se utiliza la herramienta Postman que sirve para, entre otras cosas, realizar testing exploratorio en API REST mediante el envío de peticiones asíncronas con el objetivo de analizar las respuestas y verificar que todo funcione correctamente.

FRONT es el proyecto MVC de la página web que maneja solo la lógica de visualización de la información. Se gestionan las vistas y controladores con las que interactúa el usuario, estas a su vez se relacionan con el proyecto API solicitando los datos que mostrará al usuario final. En la figura 11 se muestra el camino que se recorre en los distintos proyectos al momento de realizar una solicitud HTTP en el sistema. En la figura 12 se muestra el mismo camino pero entre los directorios específicos del proyecto.

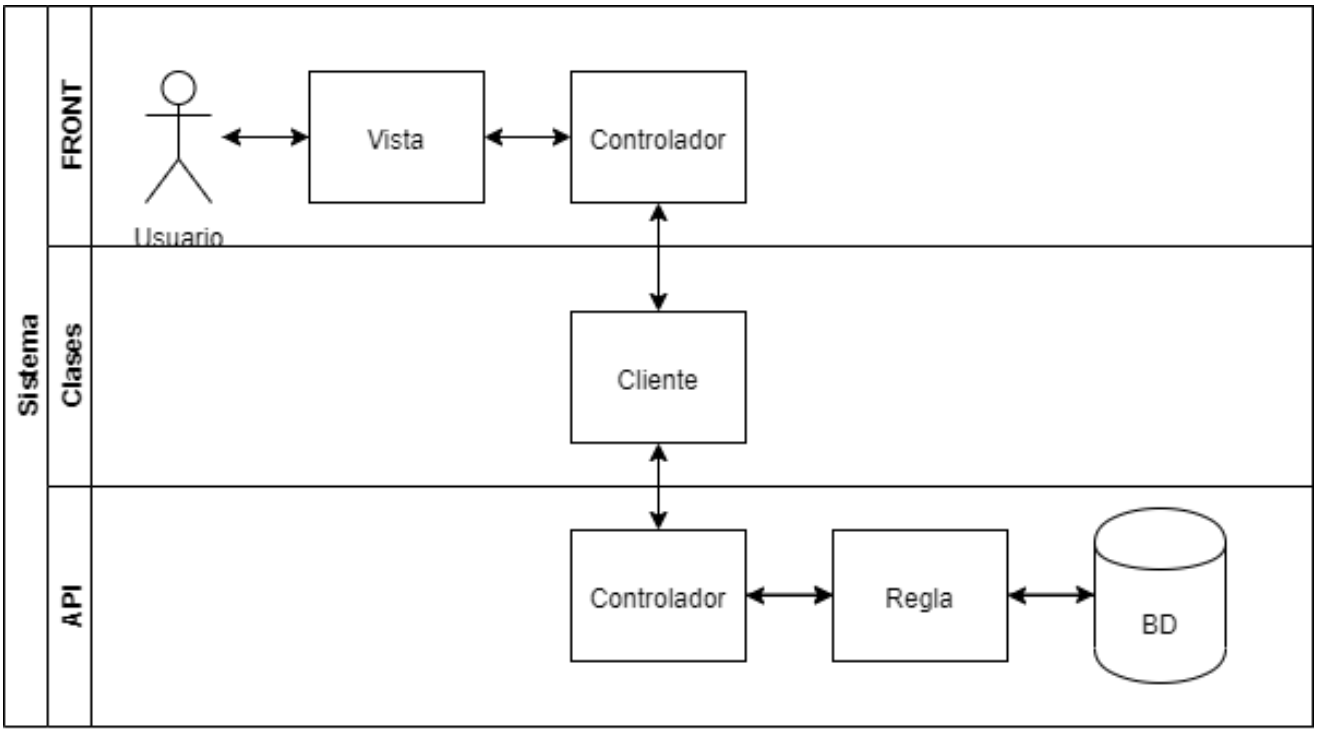

Figura 11: Ejemplo de una solicitud HTTP en el sistema.

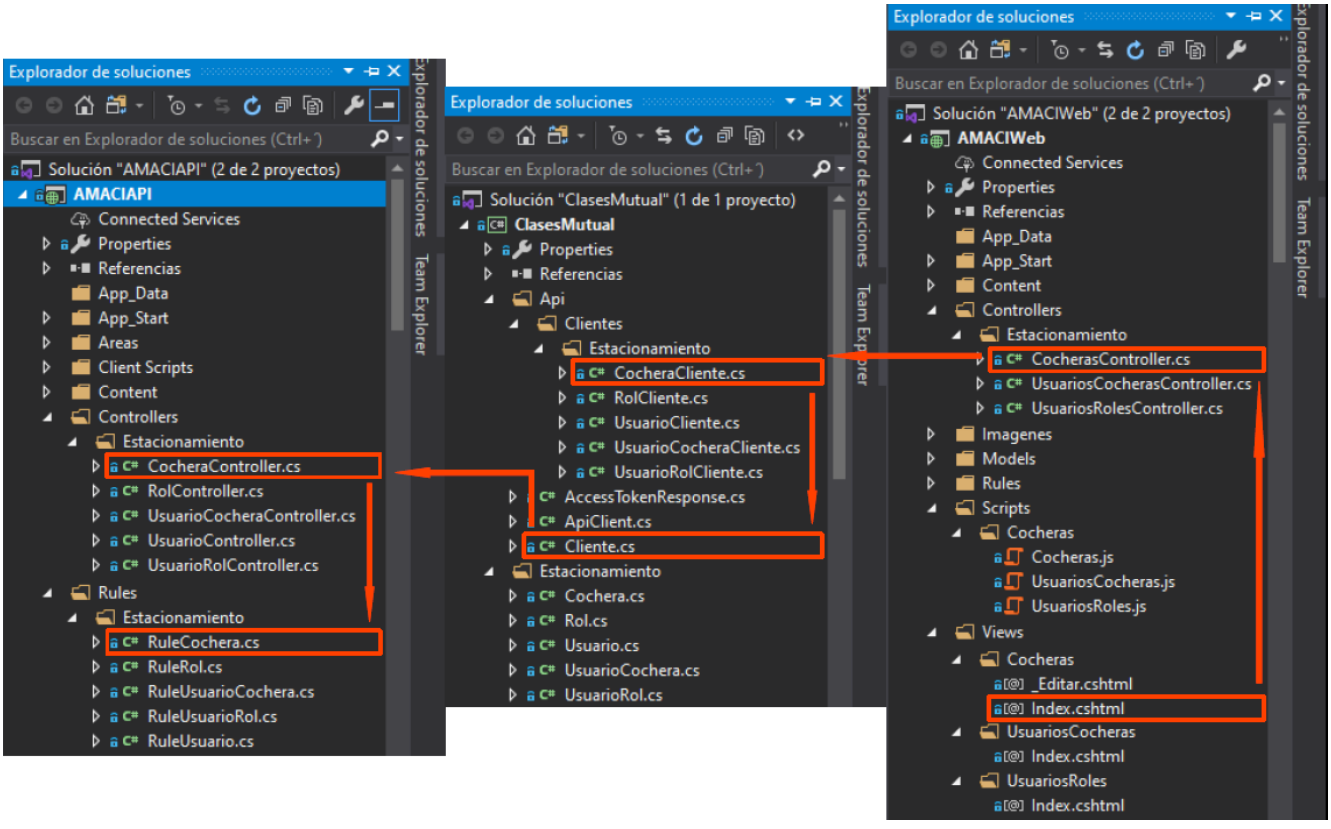

Figura 12: Camino de los directorios en los distintos proyectos para el módulo de Administración de Cocheras. API, Clases y FRONT de izquierda a derecha.

5.2 Administración de Estacionamientos

En este módulo se realiza alta, baja y modificación (ABM) de cocheras. Para la creación de una nueva cochera se solicitan, mediante una ventana modal, la dirección de la cochera, la cantidad de dársenas para vehículos mensualizados y la cantidad para vehículos en total. Una vez creada una cochera se pueden realizar 4 acciones (botones) sobre esta:

1. Modificar: Seleccionando una cochera del listado se recarga el modal con la información perteneciente a esa cochera para poder actualizar la información.
2. Eliminar: Seleccionando una cochera del listado se da de baja una cochera.
3. Listar: Se listan los administradores y playeros que están asociados a esta cochera. Se pueden eliminar si se desea.
4. Agregar: Se listan los administradores y playeros que aún no han sido seleccionados por esta cochera. Se pueden asignar si se quiere.

Figura 13: Captura de pantalla del módulo de estacionamientos.

El módulo se encarga del manejo de las cocheras y de sus relaciones (muchos a muchos) con los tipos de usuarios que tiene el sistema, administrador y playero. Las pruebas se hicieron a través de debugs con puntos de interrupción para ir siguiendo el funcionamiento. Los resultados se verifican en las mismas listas que se van modificando o directamente en la base de datos.

5.3 Administración de Vehículos

En este módulo se realizó solo la parte de Backend debido a que es un módulo que será utilizado por otro que sí posee Frontend (módulo de Administración de Registros). Este módulo se encarga de gestionar los ABM de un vehículo con sus respectivas propiedades, por ejemplo marca, o las demás clases que se especifican en la Figura 14.

Debido a la falta de interfaz, en este módulo se utilizó solo Postman con la ayuda de puntos de interrupción y los resultados se fueron chequeando en la base de datos. En Postman se realizaron principalmente solicitudes GET mediante los cuales se obtenían los listados en formato xml y solicitudes POST enviando objetos en formato JSON como se detalla en la Figura 15.

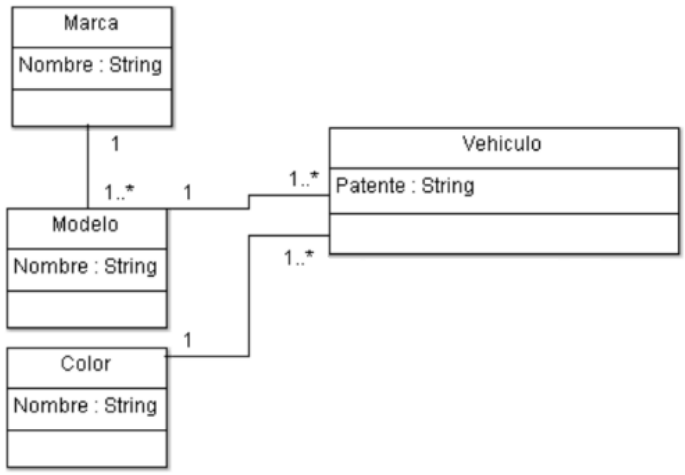

Figura 14: Diagrama de clases módulo Administración de Vehículos.

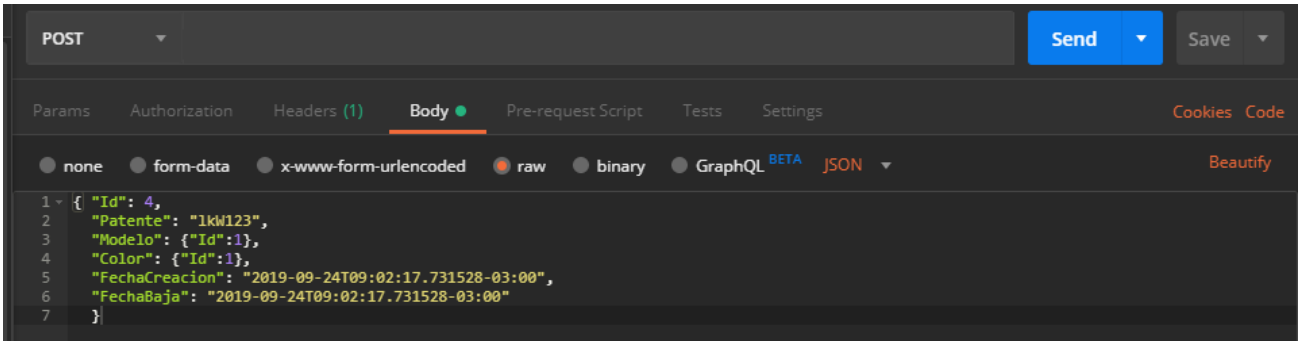

Figura 15: Ejemplo de objeto en formato JSON para pruebas.

5.4 Administración de Usuarios

Para la administración de usuarios se manejaron dos tipos de roles definidos por la empresa, administrador y playero. De la misma forma que en el AMB de cocheras, en la pestaña de usuarios se pueden crear usuarios, los cuales irán apareciendo en un listado en la misma vista. Luego se pueden editar o eliminar, así como también asignar uno o más roles. En la tabla 5 se detalla qué tipos de acciones puede realizar cada rol.

Las pruebas se realizan con debug y puntos de interrupciones para seguir el curso correcto de los datos y se termina de verificar haciendo las consultas requeridas en la base de datos. En la Figura 16 se muestran usuarios de la base de datos creados a modo de prueba.

Actividad/Rol	**Administrador**	**Playero**
Estacionamientos		
Listar	X	X

Agregar	X	
Editar	X	
Eliminar	X	
Asignar Usuario	X	
Liberar Usuario	X	
Vehículos		
Listar	X	X
Agregar	X	X
Eliminar	X	X
Usuarios		
Iniciar sesión	X	X
Listar	X	
Agregar	X	
Editar	X	X
Eliminar	X	
Tarifas		
Listar	X	X
Agregar	X	
Editar	X	
Eliminar	X	
Asignar Tipo de Tarifa	X	
Liberar Tipo de Tarifa	X	
Registros		
Listar	X	X
Agregar	X	X
Cerrar	X	X

Tabla 5: Acciones que puede realizar un usuario según su rol.

Results Messages

	id	Doc_Nro	fecha_alta	fecha_baja	Nombre	Clave
1	1	11111111	2019-10-08 20:13:28.977	NULL	JUAN PEREZ	0x0200AF5C4D577AE4C57D680EDC8E9040035D6FAD0CA946...
2	2	22222227	2019-10-08 20:13:28.980	NULL	MIGUEL JUAREZ	0x02002E1983B653C552F07E8752579F16AF56A238C8E93F266...
3	3	33333333	2019-10-08 20:38:58.070	NULL	PEPE MONJE	0x0200DA13B965F6D923C496D1D0F9A5D25BA7C14C1555F47...

Figura 16: Tabla con usuarios de prueba.

5.5 Administración de Tarifas

Haremos una breve explicación del razonamiento a la hora de crear un registro y calcular la tarifa para poder entender cómo trabaja el sistema. Como se puede observar en la figura 17, tenemos relaciones muchos a muchos entre cochera y tarifa, y uno a muchos entre cochera y registro. Es decir que una cochera puede contener varias tarifas y una tarifa puede pertenecer a varias cocheras, y una cochera puede tener muchos registros pero un registro pertenece a una cochera. Las relaciones muchos a muchos en base de datos se representan mediante una tabla intermedia en la que se alojan los identificadores de ambas tablas.

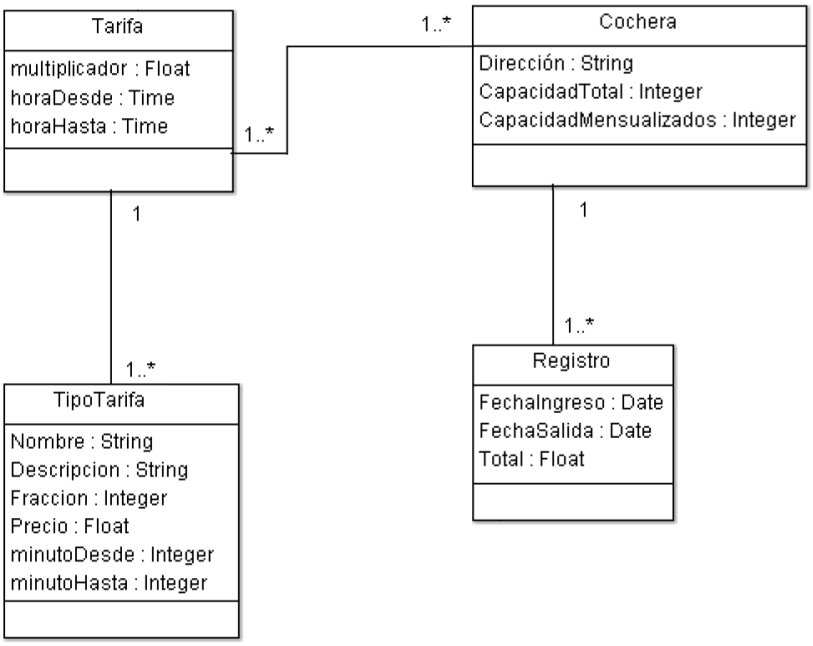

Figura 17: Clases y relaciones que intervienen en los módulos Adm. de Tarifas y Adm. de Registros.

Una tarifa tiene un tiempo de duración con el objetivo de diferenciar los montos que se cobran considerando si se trata de hora pico o no en cada uno de los estacionamientos, debido a que el flujo no es el mismo de 8 a 12 que de 18 a 20 por ejemplo. Por lo tanto, una estadía puede llevarse a cabo en más de uno de estos lapsos de tiempo. Esto se determinará comparando los límites de tiempo del registro con cada tarifa. A su vez, cada tarifa tiene distintos tipos de tarifa, en los cuales según la fracción de tiempo (media hora, más de media hora, etc.) se calcula un monto específico. Luego se suman los montos de la/las tarifas. En la figura 18 se detalla un pseudocódigo con los pasos que se deben realizar al momento de calcular el total de un registro.

```
lista-tarifa: conjunto de tarifas que posee un estacionamiento.
fecha_ingreso: horario en que ingresa un vehículo.
fecha_egreso: horario en que egresa un vehículo.
horaDesde: límite inferior de una tarifa.
horaHasta: lìmite superior de una tarifa.
multiplicador: valor que modifica el precio según si es hora pico o no.

for i=1 hasta fin lista-tarifa
     case

      fecha_ingreso >= lista-tarifa(i).horaDesde and fecha_egreso <= lista-tarifa(i).horaHasta
        minutos = fecha_egreso - fecha_ingreso
       total += lista-tarifa(i).multiplicador * calcular_precio_tipos_tarifas(minutos)

      fecha_ingreso >= lista-tarifa(i).horaDesde and fecha_egreso > lista-tarifa(i).horaHasta
          minutos = lista-tarifa(i).horaHasta - fecha_ingreso
         total += lista-tarifa(i).multiplicador * calcular_precio_tipos_tarifas(minutos)

      fecha_ingreso < lista-tarifa(i).horaDesde and fecha_egreso <= lista-tarifa(i).horaHasta
          minutos = fecha_egreso - lista-tarifa(i).horaDesde
          total += lista-tarifa(i).multiplicador * calcular_precio_tipos_tarifas(minutos)

       fecha_ingreso < lista-tarifa(i).horaDesde and fecha_egreso > lista-tarifa(i).horaHasta
         minutos = lista-tarifa(i).horaHasta - lista-tarifa(i).horaDesde
         total += lista-tarifa(i).multiplicador * calcular_precio_tipos_tarifas(minutos)

     fin case
fin for
return total
```

Figura 18: Pseudocódigo para el cálculo del costo de una estadía para un registro.

En la siguiente figura (figura 19) podemos apreciar un diagrama en el cual se ejemplifica un posible registro. Supongamos que tenemos 4 tarifas en los siguientes períodos de tiempo, donde cada una tiene un multiplicador de precio según si es hora pico o no:

- 8:01 hs - 12 hs, multiplicador: 1.5
- 12:01 hs -16 hs, multiplicador: 1
- 16:01 hs - 20 hs, multiplicador: 1.2
- 20:01 hs - 8 hs, multiplicador: 1

Un registro se lleva a cabo desde las 10 hs hasta las 14 hs, por lo tanto la estadía se realiza dos horas en la primer tarifa y dos horas en la segunda. Esta información se obtiene de la siguiente manera:

Para cada tarifa se debe calcular el tiempo:

1. tiempo = horaHasta tarifa 1 - fecha_ingreso = 12 - 10 = 2
2. tiempo = fecha_egreso - horaDesde tarifa 2 = 14 - 12 = 2

De esta manera obtendremos los tiempos para cada tarifa. En cada uno de estos se calcula el monto individual según los tipos de tarifas y luego se suman todos los montos.

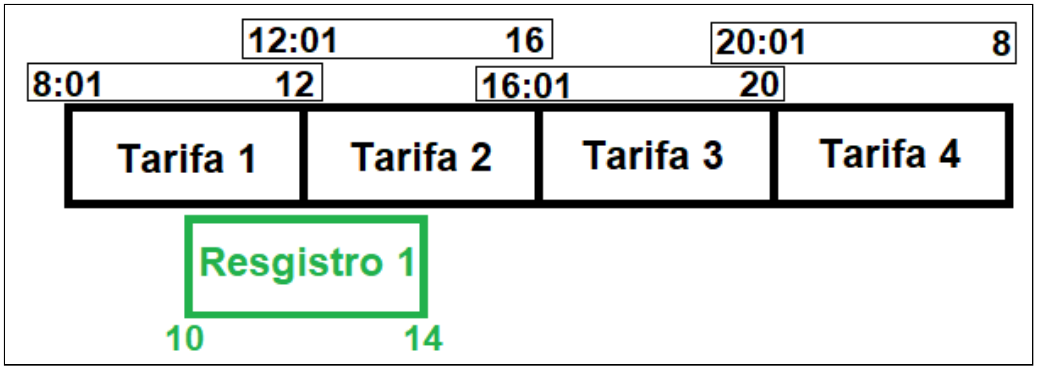

Figura 19: Ejemplificación de un posible registro.

En cada tarifa podemos encontrar uno o más tipos de tarifas con delimitadores de tiempo que permiten diferenciar los rangos a los que pertenece cada tipo. La tabla 6 y 7 detallan los rangos para las tarifas 1 y 2 respectivamente. **Observación**: Para los mensualizados los límites y la fracción permanecen en NULL. Para este rango no se aplica la ecuación. Solo se asigna el precio.

Rango	**Límite inferior (minutos)**	**Límite superior (minutos)**	**Fracción**	**Precio ejemplificador**	**Descripción**
1	0	30	30	50	La fracción entera se cobra 50 pesos.
2	31	240	30	30	240 minutos son 4 Hs. En este rango cada fracción de media hora se cobra 30 pesos.

Tabla 6: Tipos de tarifas de la tarifa 1.

Rango	**Límite inferior (minutos)**	**Límite superior (minutos)**	**Fracción**	**Precio ejemplificador**	**Descripción**
1	0	60	60	90	La fracción entera se cobra 90 pesos.

Tabla 7: Tipos de tarifas de la tarifa 2.

A partir de estos rangos, se determina a qué tipo/tipos de tarifa pertenece la cantidad de minutos. En la función `calcular_precio_tipos_tarifas(minutos)`, a partir de la variable `minutos` se obtienen los `minutos_tipotarifa` y con estas nuevas variables se calculará el costo por cada tipo de tarifa según la siguiente ecuación:

costo = precio * [techo((minutos_tipotarifa - descuento) / fracción)]

minutos_tipotarifa: tiempo que permaneció el vehículo en un tipo de tarifa específico.
descuento: minutos que se descuentan si la tarifa admite promoción.

fracción: minutos que dura la fracción según las tablas 1 y 2.
precio: valor en pesos de lo que sale cada fracción.

Obtención de **minutos_tipotarifa**:
Para 1. tenemos los primeros 30 minutos (A) pertenecen al rango o tipo de tarifa 1, y los 90 minutos (B) siguientes pertenecen a 3 fracciones del tipo de tarifa 2.
Para 2. tenemos que los 120 minutos (C) pertenecen a 2 fracciones del tipo de tarifa 1.
Luego con una simple ecuación se calcula cuánto se debe cobrar en cada tipo de tarifa:

Obtención del **costo**:

A. Para una estadía de 30 minutos con descuento:
   costo = precio*techo((30-dto)/fracción)
   = 50*techo((30-30)/30)
   = 50*0/30
   = 50*0
   = $0.
B. Para una estadía de 90 minutos sin descuento:
   costo = precio*techo((90-0)/fracción)
   = precio*techo((90-0)/30)
   = precio*techo(3)
   = 30*3
   = $90.
C. Para una estadía de 120 minutos sin descuento:
   costo = precio*techo((120-0)/fracción)
   = precio*techo((120-0)/60)
   = precio*techo(2)
   = 90*2
   = $180.

Cálculo totales para cada tarifa:
A y B pertenecen a tarifa 1:
total += 0 + [$0 * 1.5 + $90 * 1.5]
= $135
C pertenece a la tarifa 2:
total += 135 + [$180 * 1]
= $315
En el registro aparecerá un total a cobrar de $315.

Para cada tarifa se puede realizar las siguientes tareas: listar, editar o eliminar. El orden de creación es primero se crean las tarifas y luego los tipos de tarifas pertenecientes a cada una. Al momento de eliminar no se permitirá realizar la acción si la tarifa tiene tipos asociados. Tampoco se podrá editar o eliminar si se tiene registros asociados.

5.6 Administración de Registros

Un registro es el objetivo principal del sistema, todos los demás módulos deben estar desarrollados para que se pueda crear una instancia de este. Como se observa en la figura 20, un registro contiene:

- Una cochera.
- Un vehículo.
- Una tarifa o más.

Al momento de crear el registro se debe asociar una cochera y un vehículo. La cochera debe existir en el sistema, mientras que el vehículo si no existe se crea al momento de abrir el registro. La información Marca, Modelo y Color no es imprescindible para los registros de tipo medido, en cambio, sí lo es para los de tipo mensualizado. Un registro se crea con fecha de egreso y tarifa nulos, los cuales se completarán al momento de cerrarlo.

Figura 20: Captura de pantalla del formulario del módulo Adm. de Registros.

En la pestaña de registros encontramos un listado con los registros abiertos ordenados por fecha al tope de la lista. En cada uno de ellos además de cerrar, se puede editar si es necesario. Estas acciones son realizables tanto por los administradores, como por los playeros. En la figura 20 se pueden observar los botones que permiten realizar estas acciones:

1. Editar Registro. Abre el modal con la información de ese ítem para actualizar.
2. Cerrar Registro Específico. Agrega fecha de egreso al ítem seleccionado, calcula el monto a cobrar y lo muestra en pantalla.
3. Cerrar Registro General: Abre un modal para ingresar el código de registro o la patente del vehículo y cierra el registro abierto que coincida con la información ingresada.
4. Nuevo Registro: Abre el modal de la figura 20 con los campos para completar. Al presionar guardar si cumple las validaciones se crea el registro.

Cuando se crea el registro se emite un ticket con información básica de la estadía. Los datos indispensables son la fecha y hora de entrada, un código y la patente. Este es el

comprobante que el usuario deberá presentar a la salida para que el playero dé por terminada la estadía en el sistema. Esto se puede realizar de dos maneras:

1. Presionando el botón Cerrar en el registro específico del listado.
2. Presionando el botón general Cerrar Registro e introduciendo el código emitido en el comprobante de entrada.

Al cerrar el registro se genera otro archivo de texto a modo de recibo que contendrá:

- Fecha y hora de entrada.
- Fecha y hora de salida.
- Patente.
- Total a cobrar.

Cada vez que se genera un ticket se lo muestra en la pantalla para que el playero corrobore que sea correcto.

5.7 Administración de Dispositivos

El sistema va a ser manipulado por los playeros que podrán registrar entradas y salidas. El administrador deberá en primera instancia configurar las respectivas cocheras, tarifas y precios para cada cochera. Para el funcionamiento del sistema es necesario contar con dispositivos básicos como gabinete, monitor, teclado, mouse y red. Sin estos resulta imposible el funcionamiento del sistema. Por otra parte, también se necesita una impresora térmica en la cual poder llevar a cabo las impresiones de tickets de ingreso y egreso del estacionamiento. La comunicación de la impresora es mediante una conexión usb. En el sistema al momento de registrar el ingreso o el egreso se confecciona el ticket que cuenta con la siguiente información:

- Estacionamiento
- Fecha y hora de ingreso y/o egreso
- Patente
- Código/Monto a cobrar

Con esta información se realiza la impresión del ticket como comprobante para el usuario.

Estacionamiento: **25 de Mayo 2171** Ingreso: **30/1/20, 10:00** Patente: **AB123CD**	Estacionamiento: **25 de Mayo 2171** Ingreso: **30/1/20, 10:00** Egreso: **30/1/20 14:00** Patente: **AB123CD**
Código: **1345**	Total: **$315**
GRACIAS POR SU ESTADÍA	GRACIAS POR SU ESTADÍA

Figura 21: Tickets que genera el sistema.

5.8 Administración de Estadísticas

Las estadísticas pueden ser accedidas por los administradores. Para obtenerlas se debe seleccionar la pestaña en el navbar llamada Estadísticas. Al realizar esta acción aparecerán dos fechas para ingresar (Desde-Hasta) que serán utilizadas para filtrar los ingresos registrados. Para calcular las estadísticas se hace un recorrido por los registros que existen, y a partir de estos se recopila la información necesaria para el análisis que desee la empresa. Los informes analizan las siguientes variables para cada cochera:

1. Vehículos medidos por hora
2. Vehículos medidos por día
3. Cantidad de registros (medidos y mensualizados)

Supongamos que se quieren las estadísticas desde el 1/2/2020 hasta el 12/3/2020. Una consulta sin ningún tipo de filtro a nuestra base de datos de prueba arrojaría las tuplas de la figura 22, un total de 30 registros. Para obtener los vehículos medidos por hora y por día es necesario agrupar los registros y contarlos. Para el agrupamiento se deben realizar conversiones en el formato fecha para luego poder contar los registros que se realizaron en una misma hora en cada cochera. De la misma forma, pero con otro tipo de conversión, se pueden agrupar por día (figura 23). Luego se grafican los resultados a través de la librería Google Charts (figura 24).

	direccion	fecha_ingreso
1	SAN MARTIN 1743	2020-02-11 18:35:06.633
2	CRESPO 2770	2020-02-15 12:16:08.170
3	OBISPO 2441	2020-02-15 17:33:13.780
4	SAN MARTIN 1743	2020-02-15 19:21:30.373
5	25 DE MAYO 2171	2020-02-16 14:53:29.623
6	25 DE MAYO 2171	2020-02-16 14:59:12.777
7	25 DE MAYO 2171	2020-02-16 15:34:00.363
8	SAN MARTIN 3017	2020-02-16 15:36:51.230
9	SAN MARTIN 3017	2020-02-16 15:39:35.143
10	SAN MARTIN 3017	2020-02-16 15:43:17.320
11	SAN MARTIN 3017	2020-02-16 15:47:10.917
12	SAN MARTIN 1743	2020-02-17 00:01:20.730
13	OBISPO 2441	2020-02-17 14:54:39.250
14	25 DE MAYO 2171	2020-02-18 18:12:25.710
15	SAN MARTIN 3017	2020-02-19 10:08:11.893
16	SAN MARTIN 3017	2020-02-19 10:09:50.777
17	SAN MARTIN 1743	2020-02-19 10:10:44.893
18	CRESPO 2770	2020-02-19 10:13:42.930
19	SAN MARTIN 3017	2020-02-19 21:24:29.663
20	SAN MARTIN 1743	2020-02-24 12:52:11.760
21	OBISPO 2441	2020-02-25 14:46:05.120
22	OBISPO 2441	2020-02-25 15:01:00.823
23	SAN MARTIN 1743	2020-02-27 20:02:03.890
24	SAN MARTIN 1743	2020-02-27 21:57:58.843
25	OBISPO 2441	2020-02-27 23:58:48.813
26	OBISPO 2441	2020-02-28 00:12:31.870
27	OBISPO 2441	2020-02-28 09:08:24.190
28	OBISPO 2441	2020-03-06 08:53:19.807
29	OBISPO 2441	2020-03-06 10:00:47.360
30	OBISPO 2441	2020-03-06 10:09:09.290

Figura 22: Registros sin filtrar.

	direccion	hora	registros
1	SAN MARTIN 1743	2020-02-11 18	1
2	CRESPO 2770	2020-02-15 12	1
3	OBISPO 2441	2020-02-15 17	1
4	SAN MARTIN 1743	2020-02-15 19	1
5	25 DE MAYO 2171	2020-02-16 14	2
6	25 DE MAYO 2171	2020-02-16 15	1
7	SAN MARTIN 3017	2020-02-16 15	4
8	SAN MARTIN 1743	2020-02-17 00	1
9	OBISPO 2441	2020-02-17 14	1
10	25 DE MAYO 2171	2020-02-18 18	1
11	CRESPO 2770	2020-02-19 10	1
12	SAN MARTIN 1743	2020-02-19 10	1
13	SAN MARTIN 3017	2020-02-19 10	2
14	SAN MARTIN 3017	2020-02-19 21	1
15	SAN MARTIN 1743	2020-02-24 12	1
16	OBISPO 2441	2020-02-25 14	1
17	OBISPO 2441	2020-02-25 15	1
18	SAN MARTIN 1743	2020-02-27 20	1
19	SAN MARTIN 1743	2020-02-27 21	1
20	OBISPO 2441	2020-02-27 23	1
21	OBISPO 2441	2020-02-28 00	1
22	OBISPO 2441	2020-02-28 09	1
23	OBISPO 2441	2020-03-06 08	1
24	OBISPO 2441	2020-03-06 10	2

	direccion	dia	registros
1	SAN MARTIN 1743	2020-02-11	1
2	CRESPO 2770	2020-02-15	1
3	OBISPO 2441	2020-02-15	1
4	SAN MARTIN 1743	2020-02-15	1
5	25 DE MAYO 2171	2020-02-16	3
6	SAN MARTIN 3017	2020-02-16	4
7	OBISPO 2441	2020-02-17	1
8	SAN MARTIN 1743	2020-02-17	1
9	25 DE MAYO 2171	2020-02-18	1
10	CRESPO 2770	2020-02-19	1
11	SAN MARTIN 1743	2020-02-19	1
12	SAN MARTIN 3017	2020-02-19	3
13	SAN MARTIN 1743	2020-02-24	1
14	OBISPO 2441	2020-02-25	2
15	OBISPO 2441	2020-02-27	1
16	SAN MARTIN 1743	2020-02-27	2
17	OBISPO 2441	2020-02-28	2
18	OBISPO 2441	2020-03-06	3

Figura 23: Resultados de agrupamiento por hora (izquierda) y por día (derecha).

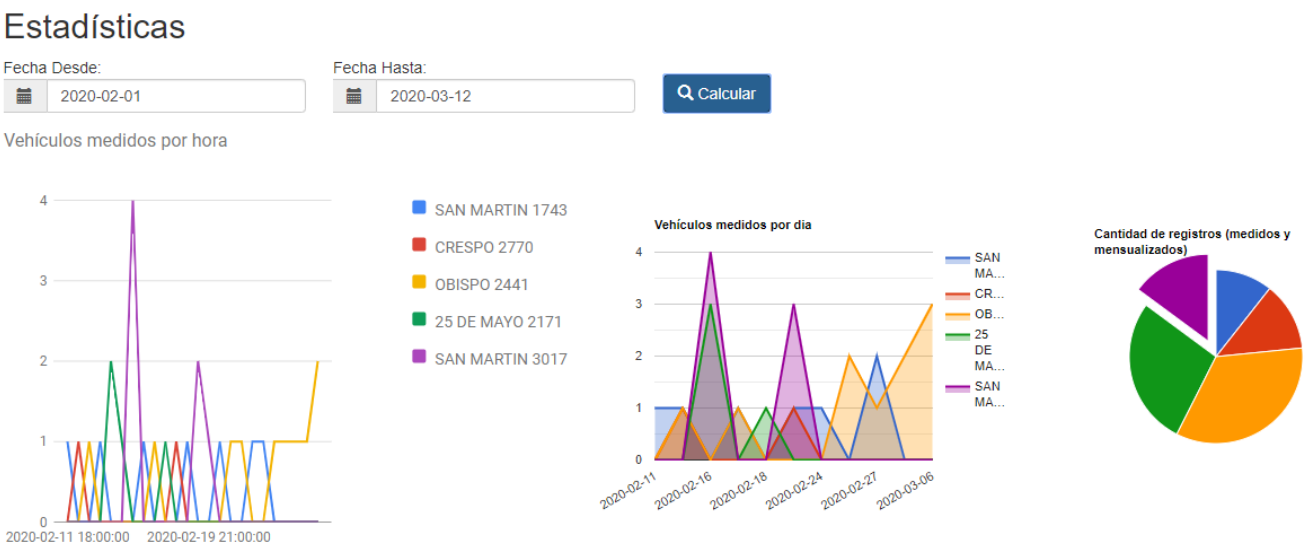

Figura 24: Estadísticas en el sistema.

Capítulo 6

6. Pruebas de software

Las pruebas integrales se realizan con el objetivo de verificar que todos los elementos unitarios que componen el sistema funcionan correctamente en conjunto.

- Objetivos: Se debe comprobar que las funcionalidades brindadas por el sistema cumplen con sus especificaciones y sin errores. También se debe verificar que la información sea accedida por usuarios con los respectivos permisos.
- Alcance: Las pruebas deben abarcar los requerimientos funcionales, específicamente los que manipulan datos (administración de usuarios, de estacionamientos y de tarifas).
- Estrategias y criterios: Se realizan pruebas de caja negra al sistema en las cuales nos enfocamos en las entradas y salidas del sistema sin tomar en cuenta la estructura interna de código.

Los casos de prueba deben generarse a partir de los casos de uso realizados en la etapa de requerimientos. Para cada caso de prueba se identifican los valores de los datos con los cuales se lleva a cabo la prueba. (sOMMERVILLE, 2005)

A continuación se muestra un ejemplo con los casos de prueba para el CU01. Los demás se especifican en el anexo 8.5.

Casos de prueba de CU1: Realizar Autenticación

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Autenticación exitosa	Usuario: “37261984” Contraseña: “*******” Botón Entrar	El sistema inicia sesión.
CP2	Autenticación cancelada	Botón Cerrar Navegador	El sistema se cierra.
CP3	Autenticación fallida ● Usuario no registrado.	Usuario: “37261985” Contraseña: “*******” Botón Entrar	El sistema no inicia sesión y muestra el mensaje: El usuario no es válido.
CP4	Autenticación fallida ● Campos obligatorios.	Usuario: “ ” Contraseña: “ ” Botón Entrar	El sistema no inicia sesión y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 8: Casos de prueba CU1.

Capítulo 7

7. Conclusión y trabajos futuros

Inicialmente se destaca que el proyecto cumple con los objetivos planteados en un principio facilitando y beneficiando la labor de los playeros de la empresa que son quienes harán uso del sistema. También es importante remarcar que al almacenar información importante que antes no se consideraba permite un mejor seguimiento del funcionamiento de cada una de las cocheras.

Además provee un mecanismo seguro al momento de realizar el cobro de una estadía permitiendo tener en cuenta nuevos factores como individualizar precios según la zona de la cochera así como considerar las franjas de hora pico del día. Es por esto que el proyecto fue aceptado por los miembros del área de sistemas de la empresa debido a que será una herramienta clave en una de las áreas de la empresa, la de gestión de estacionamientos. Cabe destacar que la implementación en producción del sistema necesitará tiempo, actualmente las pruebas fueron llevadas a cabo de manera local.

Respecto a la ingeniería de software, se sabe que es una parte importante en el desarrollo de un producto de software y es por esto que, debido a que esta parte se llevó a cabo de manera adecuada, se pudo delimitar desde un principio los pasos a seguir para la obtención de un producto que cumpla las necesidades requeridas. A partir de los requerimientos y casos de uso identificados se pudo definir el diseño de los diferentes diagramas del sistema así como también de las distintas interfaces para interactuar con los usuarios. También es de gran importancia destacar que la obtención de requerimientos se facilitó ya que desde un principio hubo mucha información disponible por parte de la empresa sumado a una buena predisposición para realizar las reuniones que sean necesarias para responder cualquier tipo de inquietud.

En el desarrollo, al usar MVC se logró separar las vistas de la lógica. Combinado con la programación orientada a objetos se pudo capitalizar todas las ventajas de estos enfoques. Con respecto a la planificación surgieron retrasos en un principio debido a una estimación que no fue suficiente para asimilar los conceptos en su totalidad. Gracias a esto se logró aprender de los errores cometidos, para mejorar las planificaciones y los tiempos de trabajo. Tras una reprogramación se logró cumplir con los objetivos planeados.

En cuanto a las tecnologías usadas se puede decir que no hubo más inconvenientes en el backend más que en la comprensión de la forma de trabajo. Esto demandó más tiempo de lo planificado, pero una vez comprendida se pudo trabajar con comodidad. Respecto al frontend hubo más problemas. A pesar de no usar ningún framework y trabajar con javascript puro surgieron contratiempos con la manipulación de la estructura de datos DataTable y la librería Google Chart para las estadísticas. También es posible que existan tecnologías que mejoren el rendimiento del sistema, pero debido a que las tecnologías eran especificadas por el cliente no se llevó a cabo ningún análisis de estos factores.

Como se especificó anteriormente el único riesgo que se efectivizó fue el de la indisponibilidad de un recurso humano para cumplir con el cronograma. Esto se debió a la

falta de conocimiento previo por parte del autor del proyecto que desembocó en un inclumplimiento del cronograma planificado inicialmente para dedicar más tiempo a capacitaciones.

El desarrollo del sistema permitió materializar todos los conocimientos y herramientas obtenidas a lo largo de la carrera universitaria. Debido a que se trata del primer proyecto que engloba todas las etapas del desarrollo de software se puede afirmar que se ha transitado por una experiencia agradable y positiva en la cual se aprendieron diversos temas que serán de gran utilidad para el desempeño profesional. Esto no hubiera sido posible sin el seguimiento constante de los directores del proyecto quienes en base a sus conocimientos y experiencia en sus respectivas áreas aportaron sus correcciones y consejos para poder llevar a cabo la mejor versión posible del sistema.

Para trabajos futuros la idea es poder sumar módulos que potencien las funcionalidades actuales del sistema. Algunos de estos pueden ser vincular el sistema con formas de pago alternativas como tarjetas de crédito o débito, o herramientas como Mercado Pago. Además podrían implementarse automatismos de barreras y sensores de peso que a través de mecanismos como mensaje de texto o aplicaciones móviles puedan ser manipuladas. Por último sería de gran ventaja poder llevar a cabo un módulo que permita el reconocimiento de patentes mediante procesamiento de imágenes para acelerar y minimizar aún más posibles errores de tipeo cometidos por empleados.

Capítulo 8

8. Anexo

8.1 Especificaciones de Casos de Uso

<table>
<tr><td colspan="2">Nombre: Realizar Autenticación</td><td>Nro 01</td></tr>
<tr><td colspan="2">Actores: Usuario</td><td>Versión: 1.0</td></tr>
<tr><td colspan="3">Objetivo: Realizar la autenticación del usuario en el sistema y establecer el perfil de acceso en la aplicación.</td></tr>
<tr><td colspan="3">Precondiciones:</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td colspan="2">Éxito:
● Actor autenticado.</td></tr>
<tr><td colspan="2">Fracaso:
● Imposibilidad de autenticar el actor.</td></tr>
<tr><td>Flujo Principal</td><td colspan="2">Flujo Alternativo</td></tr>
<tr><td>1. El actor desea iniciar sesión en la aplicación.</td><td colspan="2"></td></tr>
<tr><td>2. El sistema solicita nombre de usuario y contraseña.</td><td colspan="2"></td></tr>
<tr><td>3. El actor ingresa los campos y presiona el botón Iniciar Sesión.</td><td colspan="2"></td></tr>
<tr><td>4. El sistema valida que sea un usuario registrado.</td><td colspan="2"></td></tr>
<tr><td>5. La autenticación se realiza con éxito. El sistema habilita las funcionalidades del usuario específico.</td><td colspan="2">5.A. El usuario no es válido.
5.A.1. El sistema muestra un mensaje indicando usuario o contraseña inválidos.
5.A.2. Se retorna al paso 2 del flujo principal.</td></tr>
<tr><td>6. El caso de uso termina.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="3">Asociaciones de Extensión:
CDU16 - Alta Usuario. Se incorpora un nuevo usuario al sistema.</td></tr>
</table>

Requerimientos Especiales: ● La contraseña debe mostrarse con *.
Observaciones:
Importancia: Alta.
Urgencia: Alta.

Tabla 9: Especificación CU1.

Nombre: Modificar Contraseña	**Nro 02**
Actores: Usuario	**Versión:** 1.0
Objetivo: Realizar la modificación de la contraseña del usuario.	
Precondiciones: ● Estar autenticado en el sistema.	
Postcondiciones:	Éxito: ● Contraseña modificada.
	Fracaso: ● Imposibilidad de modificar la contraseña.
Flujo Principal	**Flujo Alternativo**
1. El actor desea modificar la contraseña de usuario.	1.A. El sistema da la opción de modificar la información del usuario. Se invoca al CDU18 - Modificar Usuario. 1.A.1. Se retorna al paso 1 del flujo principal.
2. El sistema muestra una pantalla para que el usuario ingrese la contraseña anterior, la nueva contraseña y una confirmación de la nueva contraseña.	
3. El actor ingresa los campos.	
4. El sistema valida los campos con éxito.	4.A. La contraseña no cumple con la sintaxis requerida. 4.A.1. El sistema muestra un mensaje indicando el error. 4.A.2. El caso de uso retorna al paso 2 del flujo principal. 4.B. La contraseña y su confirmación no coinciden.

	4.B.1. El sistema muestra un mensaje indicando el error. 4.B.2. El caso de uso retorna al paso 2 del flujo principal. 4.C. La contraseña anterior no coincide. 4.C.1. El sistema muestra un mensaje indicando el error. 4.C.2. El caso de uso retorna al paso 2 del flujo principal.
5. El sistema registra la nueva contraseña y muestra un mensaje exitoso.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión: CDU18 - Modificar Usuario. Se modifica un usuario del sistema.	
Requerimientos Especiales:	
Observaciones:	
Importancia: Alta	
Urgencia: Media	

Tabla 9: Especificación CU2.

Nombre: Gestionar Tarifas	**Nro 03**
Actores: Administrador	**Versión:** 1.0
Objetivo: Obtener el listado de Tarifas según preferencias.	
Precondiciones: ● Estar autenticado en el sistema como administrador.	
Postcondiciones:	Éxito: ● Listado de Tarifas según preferencias.
	Fracaso: ● No existen Tarifas para mostrar.
Flujo Principal	**Flujo Alternativo**
1. El actor desea obtener el listado de Tarifas según los siguientes criterios de búsqueda: ● Estacionamiento ● Fracción de tiempo (minutos)	

• Estadía	
2. El sistema muestra el listado de Tarifas en la pantalla.	
3. El actor abandona la pantalla.	3.A. El actor selecciona Registrar Nueva Tarifa. 3.A.1. Se invoca al CDU4 - Registrar Nueva Tarifa. 3.A.2. Se retorna al paso 3 del flujo principal. 3.B. El actor selecciona Baja Tarifa. 3.B.1. Se invoca al CDU5 - Eliminar Tarifa. 3.B.2. Se retorna al paso 3 del flujo principal. 3.C. El actor selecciona Modificar Tarifa. 3.C.1. Se invoca al CDU6 - Modificar Tarifa. 3.C.2. Se retorna al paso 3 del flujo principal.
4. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión: CDU4 - Registrar Nueva Tarifa. Se incorpora una nueva tarifa al sistema. CDU5 - Eliminar Tarifa. Se elimina una tarifa del sistema. CDU6 - Modificar Tarifa. Se modifica una tarifa del sistema.	
Requerimientos Especiales:	
Observaciones:	
Importancia: Alta	
Urgencia: Media	

Tabla 10: Especificación CU3.

Nombre: Registrar Nueva Tarifa	**Nro 04**
Actores: Administrador	**Versión:** 1.0
Objetivo: Crear una nueva tarifa en el sistema.	
Precondiciones: • Estar autenticado en el sistema como administrador.	

<table>
<tr><td colspan="2">● Haber seleccionado el CDU3 - Gestionar Tarifas.</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td>Éxito:
● Nueva Tarifa creada.</td></tr>
<tr><td>Fracaso:
● Imposibilidad de crear una Tarifa.</td></tr>
<tr><td>Flujo Principal</td><td>Flujo Alternativo</td></tr>
<tr><td>1. El actor desea crear una nueva Tarifa.</td><td></td></tr>
<tr><td>2. El sistema presenta la pantalla crear Tarifa para que el usuario ingrese:
● Nombre
● Descripción
● Precio
● Minutos</td><td></td></tr>
<tr><td>3. El actor ingresa los campos.</td><td></td></tr>
<tr><td>4. El sistema valida los campos ingresados.</td><td>4.A. Ya existe una Tarifa con el mismo nombre.
4.A.1. El sistema muestra un mensaje de error.
4.A.2. Se retorna al paso 2 del flujo principal.

4.B. Se dejó algún campo vacío.
4.B.1. El sistema muestra un mensaje de error.
4.B.2. Se retorna al paso 2 del flujo principal.</td></tr>
<tr><td>5. El sistema registra la nueva Tarifa.</td><td></td></tr>
<tr><td>6. El caso de uso termina.</td><td></td></tr>
<tr><td colspan="2">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="2">Asociaciones de Extensión:</td></tr>
<tr><td colspan="2">Requerimientos Especiales:</td></tr>
<tr><td colspan="2">Observaciones:
● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.</td></tr>
<tr><td colspan="2">Importancia: Media</td></tr>
<tr><td colspan="2">Urgencia: Media</td></tr>
</table>

Tabla 11: Especificación CU4.

Nombre: Eliminar Tarifa	**Nro 05**
Actores: Administrador	**Versión:** 1.0
Objetivo: Eliminar una Tarifa del sistema.	
Precondiciones: ● Estar autenticado en el sistema como administrador. ● Haber seleccionado una Tarifa en el CDU3 - Gestionar Tarifas.	
Postcondiciones:	Éxito: ● Tarifa eliminada lógicamente.
	Fracaso: ● Imposibilidad de eliminar la Tarifa.
Flujo Principal	**Flujo Alternativo**
1. El actor desea eliminar una Tarifa del sistema.	
2. El sistema muestra una pantalla Baja Tarifa mostrando los siguientes campos: ● Nombre ● Descripción ● Precio ● Minutos	
3. El actor confirma que quiere eliminar la Tarifa.	
4. El sistema elimina la Tarifa.	
5. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Baja	
Urgencia: Media	

Tabla 12: Especificación CU5.

<table>
<tr><td colspan="2">Nombre: Modificar Tarifa</td><td>Nro 06</td></tr>
<tr><td colspan="2">Actores: Administrador</td><td>Versión: 1.0</td></tr>
<tr><td colspan="3">Objetivo: Realizar la modificación de datos existentes en el sistema pertenecientes a una Tarifa.</td></tr>
<tr><td colspan="3">Precondiciones:
● Estar autenticado en el sistema como administrador.
● Haber seleccionado una Tarifa en el CDU3 - Gestionar Tarifas.</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td colspan="2">Éxito:
● Datos de la Tarifa modificada.</td></tr>
<tr><td colspan="2">Fracaso:
● Imposibilidad de modificar la Tarifa.</td></tr>
<tr><td>Flujo Principal</td><td colspan="2">Flujo Alternativo</td></tr>
<tr><td>1. El actor desea modificar una Tarifa.</td><td colspan="2"></td></tr>
<tr><td>2. El sistema presenta la pantalla Modificar Tarifa con los siguientes campos:
● Nombre
● Descripción
● Precio
● Minutos</td><td colspan="2"></td></tr>
<tr><td>3. El actor modifica los campos que desee.</td><td colspan="2"></td></tr>
<tr><td>4. El sistema valida los campos ingresados.</td><td colspan="2">4.A. Ya existe una Tarifa con el mismo nombre.
4.A.1. El sistema muestra un mensaje de error.
4.A.2. Se retorna al paso 2 del flujo principal.

4.B. Se dejó algún campo vacío.
4.B.1. El sistema muestra un mensaje de error.
4.B.2. Se retorna al paso 2 del flujo principal.</td></tr>
<tr><td>5. El sistema guarda los cambios.</td><td colspan="2"></td></tr>
<tr><td>6. El caso de uso termina.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="3">Asociaciones de Extensión:</td></tr>
<tr><td colspan="3">Requerimientos Especiales:</td></tr>
<tr><td colspan="3">Observaciones:
● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda,</td></tr>
</table>

terminando el Caso de Uso.
Importancia: Media
Urgencia: Media

Tabla 13: Especificación CU6.

Nombre: Gestionar Estacionamientos	**Nro 07**
Actores: Administrador	**Versión:** 1.0
Objetivo: Obtener el listado de Estacionamientos.	
Precondiciones: ● Estar autenticado en el sistema como administrador.	
Postcondiciones:	Éxito: ● Listado de Estacionamientos.
	Fracaso: ● No existen Estacionamientos para mostrar.
Flujo Principal	**Flujo Alternativo**
1. El actor desea obtener el listado de Estacionamientos.	
2. El sistema muestra el listado de Estacionamientos en la pantalla.	
3. El actor abandona la pantalla.	3.A. El actor selecciona Registrar Nuevo Estacionamiento. 3.A.1. Se invoca al CDU8 - Registrar Nuevo Estacionamiento. 3.A.2. Se retorna al paso 3 del flujo principal. 3.B. El actor selecciona Eliminar Estacionamiento. 3.B.1. Se invoca al CDU9 - Eliminar Estacionamiento. 3.B.2. Se retorna al paso 3 del flujo principal. 3.C. El actor selecciona Modificar Estacionamiento. 3.C.1. Se invoca al CDU10 - Modificar Estacionamiento. 3.C.2. Se retorna al paso 3 del flujo principal.

4. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión: CDU8 - Registrar Nuevo Estacionamiento. Se incorpora un nuevo Estacionamiento al sistema. CDU9 - Eliminar Estacionamiento. Se elimina un Estacionamiento del sistema. CDU10 - Modificar Estacionamiento. Se modifica un Estacionamiento del sistema.	
Requerimientos Especiales:	
Observaciones:	
Importancia: Alta	
Urgencia: Media	

Tabla 14: Especificación CU7.

Nombre: Registrar Nuevo Estacionamiento	**Nro 08**
Actores: Administrador	**Versión:** 1.0
Objetivo: Crear un nuevo Estacionamiento en el sistema.	
Precondiciones: ● Estar autenticado en el sistema como administrador. ● Haber seleccionado el CDU7 - Gestionar Estacionamientos.	
Postcondiciones:	Éxito: ● Nuevo Estacionamiento creado.
	Fracaso: ● Imposibilidad de crear un Estacionamiento.
Flujo Principal	**Flujo Alternativo**
1. El actor desea crear un nuevo Estacionamiento.	
2. El sistema presenta la pantalla crear Estacionamiento para que el usuario ingrese: ● Dirección ● Descripción ● Capacidad total ● Capacidad mensualizados	
3. El actor ingresa los campos.	

4. El sistema valida los campos ingresados.	4.A. Ya existe un Estacionamiento con el mismo nombre. 4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal. 4.B. Se dejó algún campo vacío. 4.B.1. El sistema muestra un mensaje de error. 4.B.2. Se retorna al paso 2 del flujo principal.
5. El sistema registra el nuevo Estacionamiento.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso. ● La capacidad de dársenas medidas se calcula a partir de las capacidades total y mensualizados.	
Importancia: Media	
Urgencia: Media	

Tabla 15: Especificación CU8.

Nombre: Modificar Estacionamiento	**Nro 10**
Actores: Administrador	**Versión:** 1.0
Objetivo: Realizar la modificación de datos existentes en el sistema pertenecientes a un Estacionamiento.	
Precondiciones: ● Estar autenticado en el sistema como administrador. ● Haber seleccionado un Estacionamiento en el CDU7 - Gestionar Estacionamientos.	
Postcondiciones:	Éxito: ● Datos del Estacionamiento modificado.
	Fracaso:

	• Imposibilidad de modificar el Estacionamiento.
Flujo Principal	**Flujo Alternativo**
1. El actor desea modificar un Estacionamiento.	
2. El sistema presenta la pantalla Modificar Estacionamiento con los siguientes campos: • Nombre • Descripción • Capacidad total • Capacidad mensualizados	
3. El actor modifica los campos que desee.	
4. El sistema valida los campos ingresados.	4.A. Ya existe un Estacionamiento con el mismo nombre. 4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal. 4.B. Se dejó algún campo vacío. 4.B.1. El sistema muestra un mensaje de error. 4.B.2. Se retorna al paso 2 del flujo principal.
5. El sistema guarda los cambios.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: • En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Media	
Urgencia: Media	

Tabla 16: Especificación CU10.

Nombre: Gestionar Vehículos Mensualizados	**Nro 11**
Actores: Usuario.	**Versión:** 1.0

<table>
<tr><td colspan="2">Objetivo: Obtener el listado de Vehículos según preferencias.</td></tr>
<tr><td colspan="2">Precondiciones:
● Estar autenticado en el sistema.</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td>Éxito:
● Listado de Vehículos según preferencias.</td></tr>
<tr><td>Fracaso:
● No existen Vehículos para mostrar.</td></tr>
<tr><td>Flujo Principal</td><td>Flujo Alternativo</td></tr>
<tr><td>1. El actor desea obtener el listado de Vehículos según los siguientes criterios de búsqueda:
● Estacionamiento
● Fecha
● Patente
● Socio
● Color
● Modelo</td><td></td></tr>
<tr><td>2. El sistema muestra el listado de Vehículos en la pantalla.</td><td></td></tr>
<tr><td>3. El actor abandona la pantalla.</td><td>3.A. El actor selecciona Registrar Nuevo Vehículo.
3.A.1. Se invoca al CDU12 - Registrar Nuevo Vehículo.
3.A.2. Se retorna al paso 3 del flujo principal.

3.B. El actor selecciona Eliminar Vehículo.
3.B.1. Se invoca al CDU13 - Eliminar Vehículo.
3.B.2. Se retorna al paso 3 del flujo principal.

3.C. El actor selecciona Modificar Vehículo.
3.C.1. Se invoca al CDU14 - Modificar Vehículo.
3.C.2. Se retorna al paso 3 del flujo principal.</td></tr>
<tr><td>4. El caso de uso termina.</td><td></td></tr>
<tr><td colspan="2">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="2">Asociaciones de Extensión:</td></tr>
</table>

CDU12 - Registrar Nuevo Vehículo. Se incorpora un nuevo Vehículo al sistema. CDU13 - Eliminar Vehículo. Se elimina un Vehículo del sistema. CDU14 - Modificar Vehículo. Se modifica un Vehículo del sistema.
Requerimientos Especiales:
Observaciones:
Importancia: Alta
Urgencia: Media

Tabla 17: Especificación CU11.

Nombre: Registrar Nuevo Vehículo	**Nro 12**
Actores: Usuario	**Versión:** 1.0
Objetivo: Crear un nuevo Vehículo en el sistema.	
Precondiciones: ● Estar autenticado en el sistema. ● Haber seleccionado el CDU11 - Gestionar Vehículos.	
Postcondiciones:	Éxito: ● Nuevo Vehículo creado.
	Fracaso: ● Imposibilidad de crear un Vehículo.
Flujo Principal	**Flujo Alternativo**
1. El actor desea crear un nuevo Vehículo.	
2. El sistema presenta la pantalla crear Vehículo para que el usuario ingrese: ● Patente ● Socio ● Estacionamiento ● Color ● Modelo	
3. El actor ingresa los campos.	
4. El sistema valida los campos ingresados.	4.A. Ya existe un Vehículo con el mismo nombre. 4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal. 4.B. Se dejó algún campo vacío. 4.B.1. El sistema muestra un mensaje

	de error. 4.B.2. Se retorna al paso 2 del flujo principal. 4.C. Se sobrepasó la capacidad de de vehículos mensualizados. 4.C.1. El sistema muestra un mensaje de error. 4.C.2. Se retorna al paso 2 del flujo principal.
5. El sistema registra el nuevo Vehículo.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso. ● El campo socio será nulo si el vehículo no pertenece a un mensualizado.	
Importancia: Media	
Urgencia: Media	

Tabla 18: Especificación CU12.

Nombre: Eliminar Vehículo	**Nro 13**
Actores: Usuario	**Versión:** 1.0
Objetivo: Eliminar un Vehículo del sistema.	
Precondiciones: ● Estar autenticado en el sistema. ● Haber seleccionado un Vehículo en el CDU11 - Gestionar Vehículos.	
Postcondiciones:	Éxito: ● Vehículo eliminado lógicamente.
	Fracaso: ● Imposibilidad de eliminar el Vehículo.
Flujo Principal	**Flujo Alternativo**
1. El actor desea eliminar un Vehículo del	

sistema.	
2. El sistema muestra una pantalla Baja Vehículo mostrando los siguientes campos: ● Patente ● Socio ● Estacionamiento ● Color ● Modelo	
3. El actor confirma que quiere eliminar el Vehículo.	
4. El sistema elimina el Vehículo.	
5. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Baja	
Urgencia: Media	

Tabla 19: Especificación CU13.

Nombre: Modificar Vehículos	**Nro 14**
Actores: Usuario	**Versión:** 1.0
Objetivo: Realizar la modificación de datos existentes en el sistema pertenecientes a un Vehículo.	
Precondiciones: ● Estar autenticado en el sistema. ● Haber seleccionado un Vehículo en el CDU11 - Gestionar Vehículos.	
Postcondiciones:	Éxito: ● Datos del Vehículo modificado.
	Fracaso: ● Imposibilidad de modificar el Vehículo.
Flujo Principal	**Flujo Alternativo**
1. El actor desea modificar un Vehículo.	

2. El sistema presenta la pantalla Modificar Vehículo con los siguientes campos: ● Patente ● Socio ● Estacionamiento ● Color ● Modelo	
3. El actor modifica los campos que desee.	
4. El sistema valida los campos ingresados.	4.A. Ya existe un Vehículo con el mismo nombre. 4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal. 4.B. Se dejó algún campo vacío. 4.B.1. El sistema muestra un mensaje de error. 4.B.2. Se retorna al paso 2 del flujo principal.
5. El sistema guarda los cambios.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Media	
Urgencia: Media	

Tabla 20: Especificación CU14.

Nombre: Gestionar Usuarios	**Nro 15**
Actores: Administrador	**Versión:** 1.0
Objetivo: Obtener el listado de Usuarios.	
Precondiciones: ● Estar autenticado en el sistema como administrador.	
Postcondiciones:	Éxito:

	• Listado de Usuarios según preferencias.
	Fracaso: • No existen Usuarios para mostrar.
Flujo Principal	**Flujo Alternativo**
1. El actor desea obtener el listado de Usuarios.	
2. El sistema muestra el listado de Usuarios en la pantalla.	
3. El actor abandona la pantalla.	3.A. El actor selecciona Registrar Nuevo Usuario. 3.A.1. Se invoca al CDU16 - Registrar Nuevo Usuario. 3.A.2. Se retorna al paso 3 del flujo principal. 3.B. El actor selecciona Eliminar Usuario. 3.B.1. Se invoca al CDU17 - Eliminar Usuario. 3.B.2. Se retorna al paso 3 del flujo principal. 3.C. El actor selecciona Modificar Usuario. 3.C.1. Se invoca al CDU18 - Modificar Usuario. 3.C.2. Se retorna al paso 3 del flujo principal.
4. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión: CDU16 - Registrar Nuevo Usuario. Se incorpora un nuevo Usuario al sistema. CDU17 - Eliminar Usuario. Se elimina un Usuario del sistema. CDU18 - Modificar Usuario. Se modifica un Usuario del sistema.	
Requerimientos Especiales:	
Observaciones:	
Importancia: Alta	
Urgencia: Media	

Tabla 21: Especificación CU15.

<table>
<tr><td colspan="2">Nombre: Registrar Nuevo Usuario</td><td>Nro 16</td></tr>
<tr><td colspan="2">Actores: Administrador</td><td>Versión: 1.0</td></tr>
<tr><td colspan="3">Objetivo: Crear un nuevo Usuario en el sistema.</td></tr>
<tr><td colspan="3">Precondiciones:
● Estar autenticado en el sistema como administrador.
● Haber seleccionado el CDU15 - Gestionar Usuarios.</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td colspan="2">Éxito:
● Nuevo Usuario creado.</td></tr>
<tr><td colspan="2">Fracaso:
● Imposibilidad de crear un Usuario.</td></tr>
<tr><td>Flujo Principal</td><td colspan="2">Flujo Alternativo</td></tr>
<tr><td>1. El actor desea crear un nuevo Usuario.</td><td colspan="2"></td></tr>
<tr><td>2. El sistema presenta la pantalla crear Usuario para que el usuario ingrese:
● DNI
● Contraseña</td><td colspan="2"></td></tr>
<tr><td>3. El actor ingresa los campos.</td><td colspan="2"></td></tr>
<tr><td>4. El sistema valida los campos ingresados.</td><td colspan="2">4.A. Ya existe un Usuario con el mismo nombre.
4.A.1. El sistema muestra un mensaje de error.
4.A.2. Se retorna al paso 2 del flujo principal.

4.B. Se dejó algún campo vacío.
4.B.1. El sistema muestra un mensaje de error.
4.B.2. Se retorna al paso 2 del flujo principal.</td></tr>
<tr><td>5. El sistema registra el nuevo Usuario.</td><td colspan="2"></td></tr>
<tr><td>6. El caso de uso termina.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="3">Asociaciones de Extensión:</td></tr>
<tr><td colspan="3">Requerimientos Especiales:
● La contraseña debe mostrarse con *.</td></tr>
<tr><td colspan="3">Observaciones:
● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.</td></tr>
</table>

Importancia: Media
Urgencia: Media

Tabla 22: Especificación CU16.

Nombre: Eliminar Usuario	**Nro 17**
Actores: Administrador	**Versión:** 1.0
Objetivo: Eliminar un Usuario del sistema.	
Precondiciones: ● Estar autenticado en el sistema como administrador. ● Haber seleccionado un Usuario en el CDU15 - Gestionar Usuarios.	
Postcondiciones:	Éxito: ● Usuario eliminado lógicamente.
	Fracaso: ● Imposibilidad de eliminar el Usuario.
Flujo Principal	**Flujo Alternativo**
1. El actor desea eliminar un Usuario del sistema.	
2. El sistema muestra una pantalla Baja Usuario mostrando los siguientes campos: ● DNI ● Contraseña	
3. El actor confirma que quiere eliminar el Usuario.	
4. El sistema elimina el Usuario.	
5. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales: ● La contraseña debe mostrarse con *.	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Baja	
Urgencia: Media	

Tabla 23: Especificación CU17.

<table>
<tr><th colspan="2">Nombre: Modificar Usuario</th><th>Nro 18</th></tr>
<tr><td colspan="2">Actores: Administrador</td><td>Versión: 1.0</td></tr>
<tr><td colspan="3">Objetivo: Realizar la modificación de datos existentes en el sistema pertenecientes a un Usuario.</td></tr>
<tr><td colspan="3">Precondiciones:
● Estar autenticado en el sistema como administrador.
● Haber seleccionado un Usuario en el CDU15 - Gestionar Usuarios.</td></tr>
<tr><td rowspan="2">Postcondiciones:</td><td colspan="2">Éxito:
● Datos del Usuario modificado.</td></tr>
<tr><td colspan="2">Fracaso:
● Imposibilidad de modificar el Usuario.</td></tr>
<tr><td>Flujo Principal</td><td colspan="2">Flujo Alternativo</td></tr>
<tr><td>1. El actor desea modificar un Usuario.</td><td colspan="2"></td></tr>
<tr><td>2. El sistema presenta la pantalla Modificar Usuario con los siguientes campos:
● Nombre
● Contraseña</td><td colspan="2"></td></tr>
<tr><td>3. El actor modifica los campos que desee.</td><td colspan="2"></td></tr>
<tr><td>4. El sistema valida los campos ingresados.</td><td colspan="2">4.A. Ya existe un Usuario con el mismo nombre.
4.A.1. El sistema muestra un mensaje de error.
4.A.2. Se retorna al paso 2 del flujo principal.

4.B. Se dejó algún campo vacío.
4.B.1. El sistema muestra un mensaje de error.
4.B.2. Se retorna al paso 2 del flujo principal.</td></tr>
<tr><td>5. El sistema guarda los cambios.</td><td colspan="2"></td></tr>
<tr><td>6. El caso de uso termina.</td><td colspan="2"></td></tr>
<tr><td colspan="3">Asociaciones de Inclusión:</td></tr>
<tr><td colspan="3">Asociaciones de Extensión:</td></tr>
<tr><td colspan="3">Requerimientos Especiales:</td></tr>
</table>

● La contraseña debe mostrarse con *.
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.
Importancia: Media
Urgencia: Media

Tabla 24: Especificación CU18.

Nombre: Ver Estadísticas	**Nro 19**
Actores: Administrador	**Versión:** 1.0
Objetivo: Consultar las estadísticas de un período específico.	
Precondiciones: ● Estar autenticado en el sistema como administrador.	
Postcondiciones:	Éxito: ● Estadísticas de un período mostradas.
	Fracaso: ● Imposibilidad de mostrar las Estadísticas.
Flujo Principal	**Flujo Alternativo**
1. El actor desea ver alguna de las siguientes Estadísticas: ● Vehículos promedio medido por hora, día, fecha. ● Cantidad promedio mensual medido y mensualizado. ● Tasa mensual de Vehículos mensualizados sobre capacidad total. Comparación entre todas las cocheras.	
2. El sistema muestra las Estadísticas según las preferencias seleccionadas por el actor.	
3. El actor abandona la pantalla.	
4. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	

Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.
Importancia: Media
Urgencia: Baja

Tabla 25: Especificación CU19.

Nombre: Registrar Entrada	**Nro 20**
Actores: Playero	**Versión:** 1.0
Objetivo: Registrar en el sistema una nueva entrada de un vehículo.	
Precondiciones: ● Estar autenticado en el sistema como playero.	
Postcondiciones:	Éxito: ● Entrada registrada.
	Fracaso: ● Imposibilidad de registrar una entrada.
Flujo Principal	**Flujo Alternativo**
1. El actor desea registrar una nueva entrada.	
2. El sistema presenta la pantalla de Registrar Entrada para que el actor ingrese: ● Patente ● Fecha ● Hora	
3. El actor ingresa los campos.	
4. El sistema valida los campos ingresados correctamente.	4.A. Ya existe un Vehículo con la misma patente. 4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal. 4.B. Se dejó algún campo vacío. 4.B.1. El sistema muestra un mensaje de error. 4.B.2. Se retorna al paso 2 del flujo principal.

5. El sistema registra la nueva Entrada.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Alta	
Urgencia: Media	

Tabla 26: Especificación CU20.

Nombre: Registrar Salida	**Nro 21**
Actores: Playero	**Versión:** 1.0
Objetivo: Registrar en el sistema la salida de un vehículo.	
Precondiciones: ● Estar autenticado en el sistema como playero.	
Postcondiciones:	Éxito: ● Salida registrada.
	Fracaso: ● Imposibilidad de registrar la Salida.
Flujo Principal	**Flujo Alternativo**
1. El actor desea registrar una salida en el sistema.	
2. El sistema muestra una pantalla Registrar Salida mostrando los siguientes campos: ● Patente ● Fecha ● Hora	
3. El actor confirma que quiere Registrar la Salida.	
4. El sistema calcula el precio de la estadía.	4.A. El actor selecciona Validar Código Promoción. 4.A.1. Se invoca al CDU22 - Validar Código Promoción. 3.A.2. Se retorna al paso 4 del flujo

	principal.
5. Se invoca al CDU23 - Validar Imagen Cámara de Seguridad.	
6. El caso de uso termina.	
Asociaciones de Inclusión: CDU23 - Validar Imagen Cámara de Seguridad. Se valida con las imágenes registradas si hubo una salida en un cierto período de tiempo.	
Asociaciones de Extensión: CDU22 - Validar Código Promoción. Se valida código para realizar descuento en el precio..	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Alta	
Urgencia: Media	

Tabla 27: Especificación CU21.

Nombre: Validar Código Promoción	**Nro 22**
Actores: Playero	**Versión:** 1.0
Objetivo: Validar en el sistema si el código de promoción es correcto.	
Precondiciones: ● Estar autenticado en el sistema como playero. ● Haber seleccionado el CDU21 - Registrar Salida.	
Postcondiciones:	Éxito: ● Código validado.
	Fracaso: ● Imposibilidad de validar el código.
Flujo Principal	**Flujo Alternativo**
1. El actor desea validar un código de promoción en el sistema.	
2. El sistema muestra una pantalla Validar Código para que el actor ingrese:: ● Código	
3. El actor ingresa los campos.	
4. El sistema valida los campos ingresados	4.A. El código no es válido.

correctamente.	4.A.1. El sistema muestra un mensaje de error. 4.A.2. Se retorna al paso 2 del flujo principal.
5. El sistema informa el descuento obtenido.	
6. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Media	
Urgencia: Baja	

Tabla 28: Especificación CU22.

Nombre: Validar Salida con Imagen de Cámara de Seguridad	**Nro 23**
Actores: Playero	**Versión:** 1.0
Objetivo: Validar si existe una imagen donde coincida la patente del vehículo con la que se registró su salida.	
Precondiciones: ● Estar autenticado en el sistema como playero. ● Haber seleccionado el CDU21 - Registrar Salida.	
Postcondiciones:	Éxito: ● Salida validada con imagen de cámara de seguridad.
	Fracaso: ● Imposibilidad de validar la salida.
Flujo Principal	**Flujo Alternativo**
1. El actor desea validar una salida.	
2. El sistema muestra una pantalla Validar Salida para que el actor seleccione la imagen que pertenece al vehículo.	
3. El actor selecciona la imagen.	
4. El sistema agrega la imagen a la salida	

registrada.	
5. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones: ● En cualquier paso del flujo normal el usuario puede cancelar la búsqueda, terminando el Caso de Uso.	
Importancia: Baja	
Urgencia: Baja	

Tabla 29: Especificación CU23.

Nombre: Gestionar Vehículos Estacionados	**Nro 24**
Actores: Usuario.	**Versión:** 1.0
Objetivo: Obtener el listado de Vehículos Estacionados según preferencias.	
Precondiciones: ● Estar autenticado en el sistema.	
Postcondiciones:	Éxito: ● Listado de Vehículos Estacionados según preferencias.
	Fracaso: ● No existen Vehículos Estacionados para mostrar.
Flujo Principal	**Flujo Alternativo**
1. El actor desea obtener el listado de Vehículos Estacionados según los siguientes criterios de búsqueda: ● Estacionamiento ● Fecha ● Patente ● Socio ● Color ● Modelo	
2. El sistema muestra el listado de Usuarios en la pantalla.	
3. El actor abandona la pantalla.	

4. El caso de uso termina.	
Asociaciones de Inclusión:	
Asociaciones de Extensión:	
Requerimientos Especiales:	
Observaciones:	
Importancia: Alta	
Urgencia: Media	

Tabla 30: Especificación CU24.

8.2 Documento de Requerimientos

RF01 - Autenticación de usuario:
El sistema deberá permitir el logueo a usuarios que estén registrados con su respectivo nombre de usuario (DNI) y contraseña. Los posibles usuarios son Administrador o Playero. Si el usuario no está registrado se mostrará un mensaje notificando esa situación.

RF02 - Modificación de contraseña:
El sistema deberá permitir la edición de contraseña a usuarios que estén registrados. Presionando el respectivo botón antes de loguearse en la primer pantalla. Se solicitarán la contraseña anterior, la nueva y una conformación de la nueva. La contraseña deberá ser alfanumérica con por lo menos 8 caracteres. Esta acción solo podrá ser realizada por un administrador. Al completar los campos si se realiza de manera correcta se cambiará la contraseña en el sistema.

RF03 - Módulo de usuarios:
Este módulo solo puede ser accedido por un administrador. La pantalla principal presenta un listado de los usuarios registrados mostrando su documento, fecha de alta, rol y estacionamiento al que pertenece. El listado tendrá un buscador. En la columna acciones habrá dos íconos que permitirán la edición o eliminación del usuario. Estas dos acciones aparecerán cada una con su respectiva ventana modal.

RF04 - Creación de usuario:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el botón nuevo usuario. Al abrirse la ventana modal se solicitarán ingresar los campos DNI, fecha alta, rol y estacionamiento al que pertenece. Todos los campos son obligatorios. Si se completan correctamente, al presionar el botón guardar se agrega un nuevo usuario al sistema. Si se presiona cancelar se vuelve a la pantalla principal de usuarios.

RF05 - Edición de usuarios:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para editar en la fila del usuario correspondiente. Al abrirse la ventana modal se mostrarán los campos DNI, fecha alta, rol y estacionamiento los cuales podrán modificarse. Todos los campos son obligatorios. Si se completan correctamente, al presionar el botón guardar sea modifica el usuario. Si se presiona cancelar se vuelve a la pantalla principal de usuarios.

RF06 - Eliminación de usuarios:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para eliminar en la fila del usuario correspondiente. Se abrirá una ventana de confirmación. Si se presiona aceptar se da de baja el usuario. Si se presiona cancelar se vuelve a la pantalla principal de usuarios.

RF07 - Módulo de estacionamiento:
Este módulo solo puede ser accedido por un administrador. La pantalla principal presenta un listado de los estacionamientos registrados mostrando su dirección, descripción, cantidad total de dársenas y cantidad de dársenas para mensualizados. El listado tendrá un buscador. En la columna acciones habrá dos íconos que permitirán la edición o eliminación del estacionamiento. Estas dos acciones aparecerán cada una con su respectiva ventana modal.

RF04 - Creación de estacionamientos:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el botón nuevo estacionamiento. Al abrirse la ventana modal se solicitarán ingresar los campos dirección, descripción, cantidad de dársenas total y de mensualizados. Todos los campos son obligatorios. Si se completan correctamente, al presionar el botón guardar se agrega un nuevo estacionamiento al sistema. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF05 - Edición de estacionamientos:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para editar en la fila del estacionamiento correspondiente. Al abrirse la ventana modal se mostrarán los mismos campos que para la creación, los cuales podrán modificarse. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF06 - Eliminación de estacionamientos:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para eliminar en la fila del estacionamiento correspondiente. Se abrirá una ventana de confirmación. Si se presiona aceptar se da de baja el estacionamiento. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF08 - Módulo de tarifa:

Este módulo solo puede ser accedido por un administrador. La pantalla principal presenta un listado de las tarifas registrados mostrando su nombre, descripción, precio y minutos que dura la tarifa. El listado tendrá un buscador en el que se podrá filtrar por estacionamiento, fracción de tiempo o estadía. En la columna acciones habrá dos íconos que permitirán la edición o eliminación del estacionamiento. Estas dos acciones aparecerán cada una con su respectiva ventana modal.

RF09 - Creación de tarifas:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el botón nueva tarifa. Al abrirse la ventana modal se solicitarán ingresar los campos nombre, descripción, precio y minutos que dura la tarifa. Todos los campos son obligatorios. Si se completan correctamente, al presionar el botón guardar se agrega una nueva tarifa al sistema. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF10 - Edición de tarifas:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para editar en la fila de la tarifa correspondiente. Al abrirse la ventana modal se mostrarán los mismos campos que para la creación, los cuales podrán modificarse. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF11 - Eliminación de tarifas:
El administrador solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para eliminar en la fila de la tarifa correspondiente. Se abrirá una ventana de confirmación. Si se presiona aceptar se da de baja la tarifa. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF12 - Módulo de vehículos estacionados:
Este módulo solo puede ser accedido por un playero. La pantalla principal presenta un listado de las vehículos estacionados registrados mostrando su estacionamiento, fecha de ingreso, patente, socio, tarifa, hora de ingreso, marca, modelo y color. El listado tendrá un buscador en el que se podrá filtrar por estacionamiento, fecha de ingreso, patente, socio, hora de ingreso, marca, modelo y color. En la columna acciones habrá dos íconos que permitirán la edición y registro de salida del vehículo. Estas dos acciones aparecerán cada una con su respectiva ventana modal.

RF13 - Creación de registro de entrada:
El playero solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el botón nuevo registro. Al abrirse la ventana modal se solicitarán ingresar los campos estacionamiento, fecha de ingreso, patente, socio, tipo de tarifa (medido, mensualizado), hora de ingreso, marca, modelo y color. Son obligatorios estacionamiento, fecha y hora de ingreso, patente, tipo de tarifa. Si se completan correctamente, al presionar el botón guardar se agrega un nuevo registro al sistema. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF14 - Edición de registros:

El playero solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para editar en la fila del registro correspondiente. Al abrirse la ventana modal se mostrarán los mismos campos que para la creación, los cuales podrán modificarse. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF15 - Registro de salida:
El playero solo podrá acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para registro de salida en la fila del registro correspondiente. Se abrirá una ventana de confirmación. Si se presiona aceptar se da de baja el registro. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF16 - Módulo de vehículos mensualizados:
Este módulo puede ser accedido por un administrador o un playero. La pantalla principal presenta un listado de las vehículos mensualizados mostrando su estacionamiento, fecha de ingreso, patente, socio, tarifa, hora de ingreso, marca, modelo y color. El listado tendrá un buscador en el que se podrá filtrar por estacionamiento, fecha de ingreso, patente, socio, hora de ingreso, marca, modelo y color. En la columna acciones habrá dos íconos que permitirán la edición y eliminación del vehículo. Estas dos acciones aparecerán cada una con su respectiva ventana modal.

RF17 - Creación de vehículos mensualizados:
El playero o el administrador podrán acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el botón nuevo mensualizado. Al abrirse la ventana modal se solicitarán ingresar los campos estacionamiento, fecha de ingreso, patente, socio, tipo de tarifa (medido, mensualizado), hora de ingreso, marca, modelo y color. Son obligatorios estacionamiento, fecha y hora de ingreso, patente, tipo de tarifa. Si se completan correctamente, al presionar el botón guardar se agrega un nuevo vehículo mensualizado al sistema. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF18 - Edición de vehículos mensualizados:
El playero o el administrador podrán acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para editar en la fila del vehículo correspondiente. Al abrirse la ventana modal se mostrarán los mismos campos que para la creación, los cuales podrán modificarse. Si se presiona cancelar se vuelve a la pantalla principal de estacionamientos.

RF19 - Eliminación de vehículos mensualizados:
El playero o el administrador podrán acceder a esta acción. Para esto deberá estar logueado en el sistema y presionar el ícono para eliminar en la fila del vehículo correspondiente. Se abrirá una ventana de confirmación. Si se presiona aceptar se da de baja el registro. Si se presiona cancelar se vuelve a la pantalla principal de tarifas.

RF20 - Módulo de estadísticas:
Este módulo puede ser accedido por un administrado. La pantalla principal presenta un distintos gráficos que son de utilidad para la observación del desempeño del

estacionamiento. Estas son vehículos promedio medido por hora, día, fecha; cantidad promedio mensual medido y mensualizado; tasa mensual de Vehículos mensualizados sobre capacidad total; comparación entre todas las cocheras.

8.3 Especificación del Diccionario de Datos

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTCochera_Id	int	x		Id único de la cochera.	
direccion	varchar(45)			Nombre de la cochera.	
capacidad_total	int			Cantidad máxima de cocheras.	
capacidad_mens	int			Cantidad máxima de cocheras disponible para mensualizados.	
fecha_alta	datetime			Fecha alta cochera.	
fecha_baja	datetime			Fecha baja cochera.	

Tabla 31: Base de Datos. Tabla ESTCochera.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTColor_Id	int	x		Id único del color del vehículo.	
nombre	varchar(45)			Nombre del color del vehículo.	
fecha_alta	datetime			Fecha alta color.	
fecha_baja	datetime			Fecha baja color.	

Tabla 32: Base de Datos. Tabla ESTColor.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTMarca_Id	int	x		Id único de la marca del vehículo.	
nombre	varchar(45)			Nombre de la marca del vehículo.	

fecha_alta	datetime			Fecha alta marca.	
fecha_baja	datetime			Fecha baja marca.	

Tabla 33: Base de Datos. Tabla ESTMarca.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTModelo_Id	int	x		Id único del modelo del vehículo.	
ESTMarca_Id	int		x	Id único de la marca del vehículo.	Marca.
nombre	varchar(45)			Nombre del modelo del vehículo.	
fecha_alta	datetime			Fecha alta modelo.	
fecha_baja	datetime			Fecha baja Modelo.	

Tabla 34: Base de Datos. Tabla ESTModelo.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTVehiculo_Id	int	x		Id único del vehículo.	
ESTModelo_ESTModelo_Id	int		x	Id único del modelo del vehículo.	Modelo.
ESTColor_ESTColor_Id	int		x	Id único del color del vehículo.	Color.
patente	varchar(10)			Nombre del vehículo.	
fecha_alta	datetime			Fecha alta vehículo.	
fecha_baja	datetime			Fecha baja vehículo.	

Tabla 35: Base de Datos. Tabla ESTVehiculo.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTPersona_id	int	x		Id único de la persona.	
nombre	varchar(45)			Nombre de la persona.	
fecha_alta	datetime			Fecha alta persona.	

fecha_baja	datetime			Fecha baja persona.	

Tabla 36: Base de Datos. Tabla ESTPersona.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTUsuario_id	int	x		Id único del usuario.	
ESTPersona_ESTPersona_Id	varchar(45)		x	Documento del usuario.	Persona.
fecha_alta	datetime			Fecha alta usuario.	
fecha_baja	datetime			Fecha baja usuario.	

Tabla 37: Base de Datos. Tabla ESTUsuario.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTRol_Cochera_Id	int	x		Id único de la relación.	
ESTCochera_ESTCochera_Id	int		x	Id de la cochera.	Cochera.
ESTUsuario_ESTUsuario_Id	int		x	Id del usuario.	Usuario.
fecha_alta	datetime			Fecha alta marca.	
fecha_baja	datetime			Fecha baja marca.	

Tabla 38: Base de Datos. Tabla ESTRol_Cochera.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTTarifa_Id	int	x		Id único de la tarifa.	
precio	float			Precio de la tarifa.	
minuto_desde	int			Minuto en el que comienza la tarifa.	
minuto_hasta	int			Minuto en el que finaliza la tarifa.	
hora_desde	time(7)			Hora en la que comienza la tarifa.	

hora_hasta	time(7)			Hora en la que finaliza la tarifa.	
fecha_alta	datetime			Fecha alta tarifa.	
fecha_baja	datetime			Fecha baja tarifa.	

Tabla 39: Base de Datos. Tabla ESTTarifa.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTTipoTarifa_Id	int	x		Id único del tipo de tarifa.	
nombre	float			Nombre del tipo de tarifa.	
descripcion	int			Descripción del tipo de tarifa.	
fraccion	int			Fracción de minutos que indica cada cuánto debe cobrarse.	
fecha_alta	datetime			Fecha alta tipo de tarifa.	
fecha_baja	datetime			Fecha baja tipo de tarifa.	

Tabla 40: Base de Datos. Tabla ESTTipoTarifa.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTSocio_Registro_Id	int	x		Id único del socio.	
ESTPersonaESTPersona_Id	int		x	Id de la persona socia.	Persona.
fecha_inicio	datetime			Fecha inicio socio.	
fecha_fin	datetime			Fecha fin socio.	
fecha_alta	datetime			Fecha alta socio.	
fecha_baja	datetime			Fecha baja socio.	

Tabla 41: Base de Datos. Tabla ESTSocio_Registro.

Campo	Tipo	PK	FK	Descripción	Valores posibles

ESTTarifa_Cochera_Id	int	x		Id único de la relación.	
ESTTarifa_ESTTarifa_Id	int		x	Id de la tarifa.	Tarifa.
ESTCochera_ESTCochera_Id	int		x	Id de la cochera.	Cochera.
promocionable	bit			Indicador de si admite promoción o no.	0: No admite promoción. 1: Admite promoción.
fecha_alta	datetime			Fecha alta relación.	
fecha_baja	datetime			Fecha baja relación.	

Tabla 42: Base de Datos. Tabla ESTTarifa_Cochera.

Campo	Tipo	PK	FK	Descripción	Valores posibles
ESTRegistro_Id	int	x		Id único del registro.	
ESTVehiculo_ESTVehiculo_Id	int		x	Id del vehículo.	Vehículo.
ESTSocio_Registro_ESTSocio_Registro_Id	int		x	Id del socio.	Socio.
ESTCochera_ESTCochera_Id	int		x	Id de la cochera.	Cochera.
promocionable	bit			Indicador de si admite promoción o no.	0: No admite promoción. 1: Admite promoción.
fecha_ingreso	datetime			Fecha ingreso del vehículo.	
fecha_egreso	datetime			Fecha egreso del vehículo.	
total	float			Total a cobrar por la estadía en la cochera.	
fecha_alta	datetime			Fecha alta registro.	
fecha_baja	datetime			Fecha baja registro.	

Tabla 43: Base de Datos. Tabla ESTRegistro.

8.4 Especificación de Wireframes del Sistema

8.4.1 Autenticación

Interfaz inicial que se muestra al abrir el sistema para los dos tipos de usuarios. Aquí se encuentran los campos usuario y contraseña para completar si se desea iniciar sesión. Contiene las siguientes partes:

- Iniciar Sesión: Al presionar este botón si se cumplen las validaciones el sistema inicia sesión para el usuario que ingresó.
- Modificar Contraseña: Al seleccionar este botón se abre un formulario para ingresar los datos perteneciente a un usuario y se solicita la nueva contraseña. Si pasa las validaciones se modifica la contraseña con éxito.

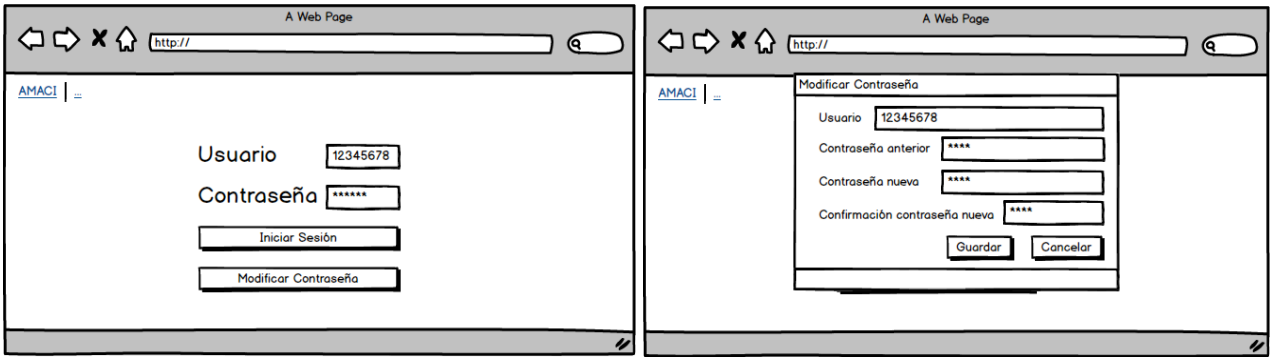

Figura 25: Pantalla inicio de sesión y modificar contraseña.

8.4.2 Gestionar Vehículos Estacionados

Interfaz en la cual se encuentran los vehículos que están dentro de un estacionamiento específico. Puede ser accedida por los dos tipos de usuarios y contiene las siguientes partes:

- Nuevo Registro: Este botón permite agregar un nuevo registro al sistema. Agregando los datos solicitados del lado derecho de la figura 26 y presionando guardar si pasa las validaciones se crea con éxito.
- Modificar Registro: Presionando el botón de la columna acciones en el listado se abre el formulario con la información correspondiente para editar. Al presionar guardar, del mismo modo que al agregar uno nuevo, si pasa las validaciones modifica con éxito.
- Registrar Salida: Al clickear el segundo botón en la columna acción se abre un alert pidiendo confirmación para registrar el horario de salida y calcular el

precio a cobrar. Si la cochera en la que se encuentra el registro tiene una tarifa asociada se realiza la acción de manera exitosa.

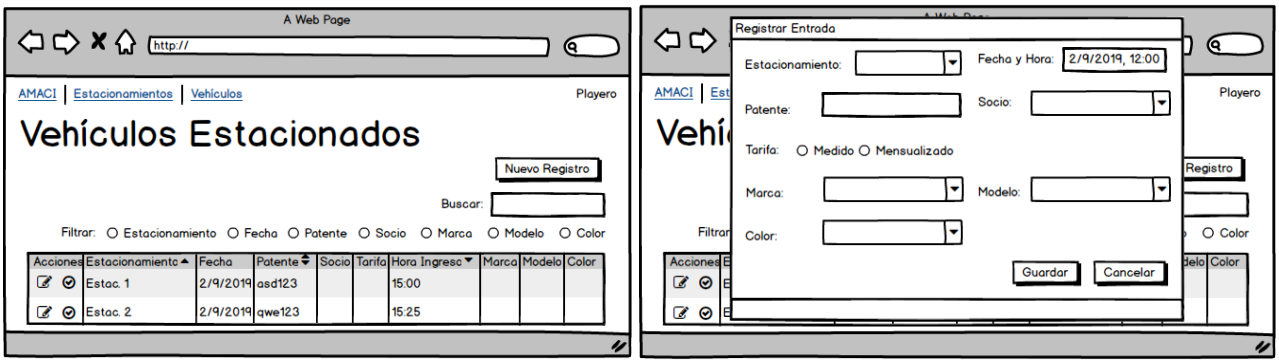

Figura 26: Pantalla 2 gestionar vehículos estacionados.

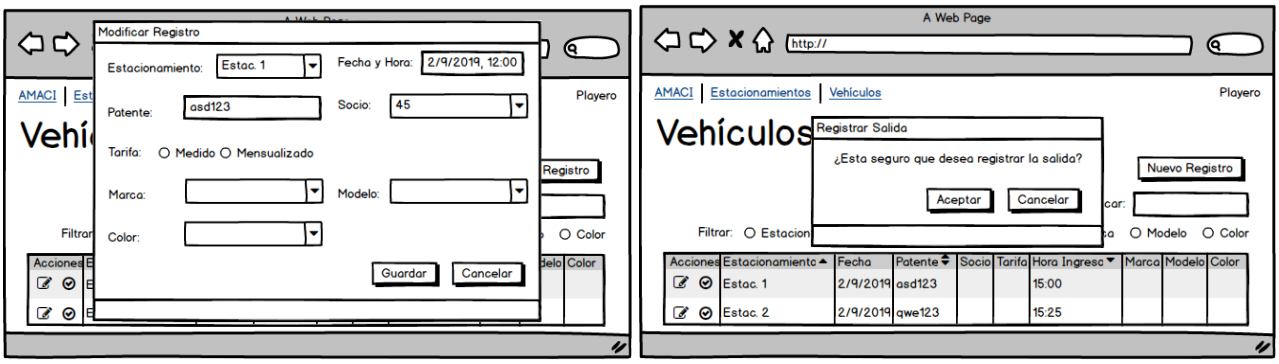

Figura 27: Pantalla 2 gestionar vehículos estacionados.

8.4.3 Gestionar Tarifas

Interfaz en la cual se encuentran las tarifas del sistema. Solo puede ser accedida por los administradores y contiene las siguientes partes:

- Nueva Tarifa: Este botón permite agregar una nueva tarifa al sistema. Agregando los datos solicitados del lado derecho de la figura 28 y presionando guardar si pasa las validaciones se crea con éxito.
- Modificar Tarifa: Presionando el botón de la columna acciones en el listado se abre el formulario con la información correspondiente para editar. Al presionar guardar, del mismo modo que al agregar uno nuevo, si pasa las validaciones modifica con éxito.
- Eliminar Tarifa: Al clickear el segundo botón en la columna acción se abre un alert pidiendo confirmación. Si la tarifa no tiene ningún vehículo asociado al momento de querer realizar la acción se elimina con éxito.

Figura 28: Pantalla gestionar tarifas y registrar nueva tarifa.

Figura 29: Pantalla modificar tarifa y eliminar tarifa.

8.4.4 Gestionar Usuarios

Interfaz en la cual se encuentran los usuarios del sistema. Solo puede ser accedida por los administradores y contiene las siguientes partes:

- Nuevo Usuario: Este botón permite agregar un nuevo usuario al sistema. Agregando los datos solicitados del lado derecho de la figura 30 y presionando guardar si pasa las validaciones se crea con éxito.
- Modificar Usuario: Presionando el botón de la columna acciones en el listado se abre el formulario con la información correspondiente para editar. Al presionar guardar, del mismo modo que al agregar uno nuevo, si pasa las validaciones modifica con éxito.
- Eliminar Usuario: Al clickear el segundo botón en la columna acción se abre un alert pidiendo confirmación. Si el usuario no tiene ninguna cochera asociada al momento de querer realizar la acción se elimina con éxito.

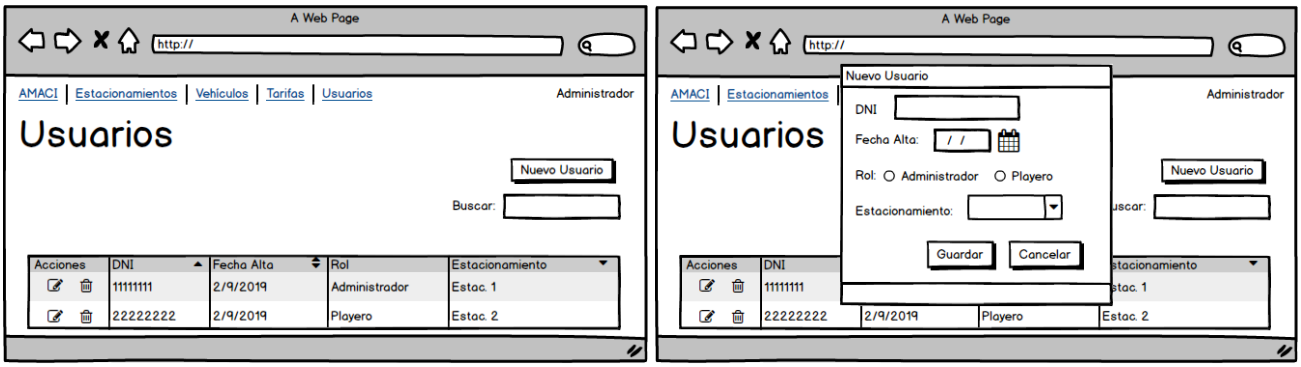

Figura 30: Pantalla gestionar usuarios y registrar nuevo usuario.

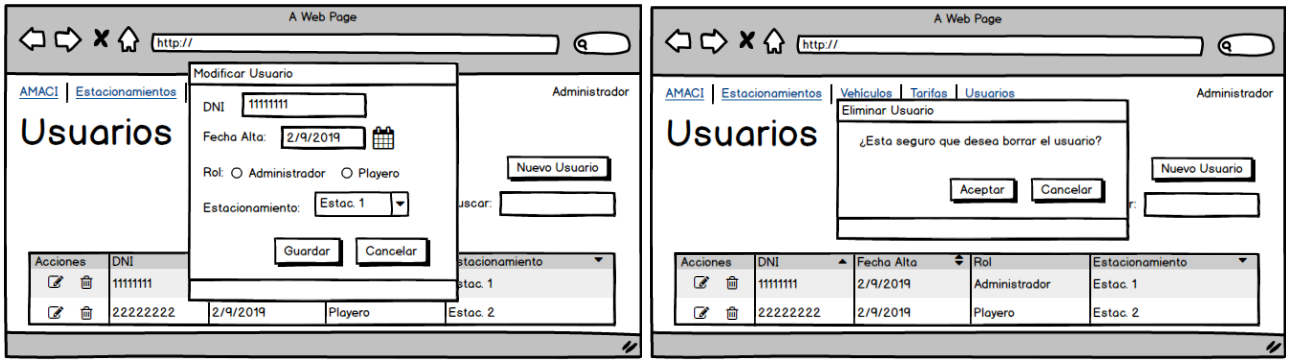

Figura 31: Pantalla modificar usuario y eliminar usuario.

8.4.5 Ver Estadísticas

Interfaz en la cual se encuentran las estadísticas del sistema. Solo puede ser accedida por los administradores y contiene las siguientes partes:

- Fecha Desde: Este botón permite introducir la fecha que será usada como límite inferior al momento de calcular las estadísticas.
- Fecha Hasta: Este botón permite introducir la fecha que será usada como límite superior al momento de calcular las estadísticas.
- Calcular: Al presionar este botón si las fechas son válidas se realiza la obtención de estadísticas con éxito.

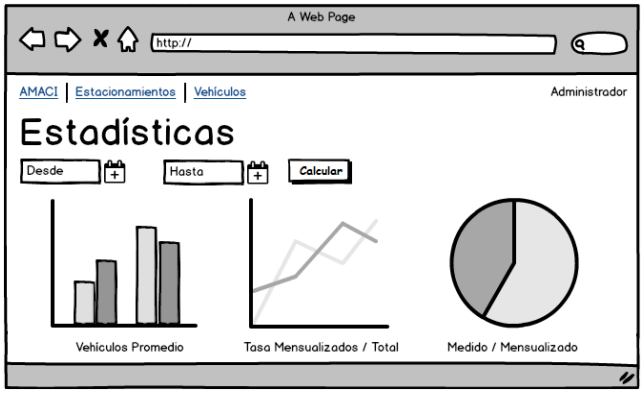

Figura 32: Pantalla ver estadísticas.

8.5 Especificación de los Casos de Prueba

8.5.1 Administración de usuarios

Casos de prueba de CU1: Realizar Autenticación

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Autenticación exitosa	Usuario: “37261984” Contraseña: “*******” Botón Entrar	El sistema inicia sesión.
CP2	Autenticación cancelada	Botón Cerrar Navegador	El sistema se cierra.
CP3	Autenticación fallida ● Usuario no registrado.	Usuario: “37261985” Contraseña: “*******” Botón Entrar	El sistema no inicia sesión y muestra el mensaje: El usuario no es válido.
CP4	Autenticación fallida ● Campos obligatorios.	Usuario: “ ” Contraseña: “ ” Botón Entrar	El sistema no inicia sesión y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 44: Casos de prueba CU1.

Casos de prueba de CU2: Modificar Contraseña

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Modificación exitosa	Usuario: “37261984” Contraseña: “*******” Botón Entrar	El sistema inicia sesión.
CP2	Modificación cancelada	Botón Cerrar Navegador	El sistema se cierra.
CP3	Modificación fallida ● Usuario no registrado.	Usuario: “37261985” Contraseña: “*******” Botón Entrar	El sistema no modifica contraseña y muestra el mensaje: El usuario no está registrado.
CP4	Modificación fallida	Usuario: “37261985”	El sistema no modifica

	• Sintaxis requerida.	Contraseña: “ ” Botón Entrar	contraseña y muestra el mensaje: El usuario o la contraseña no cumplen la sintaxis requerida.

Tabla 45: Casos de prueba CU2.

Casos de prueba de CU16: Registrar Nuevo Usuario

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro del usuario exitoso	DNI: “32145698” Contraseña: “*****” Botón Guardar	El sistema guarda el usuario.
CP2	Registro del usuario cancelado	DNI: N/A Contraseña: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Registro del usuario fallido • usuario existente.	DNI: “32145698” Contraseña: “*****” Botón Guardar	El sistema no inicia sesión y muestra el mensaje: Ya existe un usuario con esos datos.
CP4	Registro del usuario fallido • Campos requeridos.	DNI: “ ” Contraseña: “ ” Botón Guardar	El sistema no guarda el usuario y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 46: Casos de prueba CU16.

Casos de prueba de C18: Modificar Usuario

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Modificación del usuario exitoso	DNI: “32145666” Contraseña: “*****” Botón Guardar	El sistema guarda el usuario.
CP2	Modificación del usuario cancelado	DNI: N/A Contraseña: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Modificación del usuario fallido • Usuario existente.	DNI: “32145666” Contraseña: “*****” Botón Guardar	El sistema no guarda el usuario y muestra el mensaje: Ya existe un usuario con esos

			datos.
CP4	Modificación del usuario fallido ● Campos requeridos.	DNI: “ ” Contraseña: “ ” Botón Guardar	El sistema no inicia sesión y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 47: Casos de prueba CU18.

8.5.2 Administración de tarifas

Casos de prueba de CU4: Registrar Nueva Tarifa

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro de la tarifa exitoso	Nombre: “Por hora” Descripción: “Tarifa que se cobra por hora durante 4 hs” Precio: 80 Minutos: 240 Botón Guardar	El sistema guarda la tarifa.
CP2	Registro de la tarifa cancelado	Nombre: N/A Descripción: N/A Precio: N/A Minutos: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Registro de la tarifa fallido ● Tarifa existente.	Nombre: “Por hora” Descripción: “Tarifa que se cobra por hora durante 4 hs” Precio: 80 Minutos: 240 Botón Guardar	El sistema no guarda la tarifa y muestra el mensaje: El usuario no está registrado.
CP4	Registro de la tarifa fallido ● Campos requeridos.	Nombre: “ ” Descripción: “Tarifa que se cobra por hora durante 4 hs” Precio: “ “ Minutos: “ “ Botón Guardar	El sistema no guarda la tarifa y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 48: Casos de prueba CU4.

Casos de prueba de CU6: Modificar Tarifa

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Modificación de la tarifa exitosa	Nombre: “Por hora” Descripción: “Tarifa que se cobra por hora durante 2 hs” Precio: 80 Minutos: 120 Botón Guardar	El sistema modifica la tarifa existente.
CP2	Modificación de la tarifa cancelada	Nombre: N/A Descripción: N/A Precio: N/A Minutos: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Modificación de la tarifa fallido ● Tarifa existente.	Nombre: “Por hora” Descripción: “Tarifa que se cobra por hora durante 4 hs” Precio: 80 Minutos: 240 Botón Guardar	El sistema no guarda la tarifa y muestra el mensaje: Ya existe una tarifa con esos datos.
CP4	Modificación de la tarifa fallido ● Campos requeridos.	Nombre: “ ” Descripción: “Tarifa que se cobra por hora durante 4 hs” Precio: “ “ Minutos: “ “ Botón Guardar	El sistema no guarda la tarifa y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 49: Casos de prueba CU6.

8.5.3 Administración de estacionamientos

Casos de prueba de CU8: Registrar Nuevo Estacionamiento

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro del estacionamiento exitoso	Dirección: “25 de Mayo 2171” Descripción: “Cochera principal” Capacidad Total: “30” Capacidad Mensualizados: “5” Botón Guardar	El sistema guarda el estacionamiento.

CP2	Registro del estacionamiento cancelado	Dirección: N/A Descripción: N/A Capacidad Total: N/A Capacidad Mensualizados: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Registro del estacionamiento fallido ● Estacionamiento existente.	Dirección: “25 de Mayo 2171” Descripción: “Cochera principal” Capacidad Total: “30” Capacidad Mensualizados: “5” Botón Guardar	El sistema no guarda el estacionamiento y muestra el mensaje: Ya existe un estacionamiento con esos datos.
CP4	Registro del estacionamiento fallido ● Campos requeridos.	Dirección: “ ” Descripción: “Cochera principal” Capacidad Total: “ ” Capacidad Mensualizados: “ ” Botón Guardar	El sistema no guarda el estacionamiento y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 50: Casos de prueba CU8.

Casos de prueba de C10: Modificar Estacionamiento

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Modificación del estacionamiento exitoso	Dirección: “25 de Mayo 2171” Descripción: “Cochera principal” Capacidad Total: “90” Capacidad Mensualizados: “15” Botón Guardar	El sistema guarda el estacionamiento.
CP2	Modificación del estacionamiento cancelado	Dirección: N/A Descripción: N/A Capacidad Total: N/A Capacidad Mensualizados: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Modificación del estacionamiento fallido ● Estacionamiento existente.	Dirección: “25 de Mayo 2171” Descripción: “Cochera principal” Capacidad Total: “90”	El sistema no guarda el estacionamiento y muestra el mensaje: Ya existe un estacionamiento con

		Capacidad Mensualizados: “15” Botón Guardar	esos datos.
CP4	Modificación del estacionamiento fallido ● Campos requeridos.	Dirección: “ ” Descripción: “Cochera principal” Capacidad Total: “ ” Capacidad Mensualizados: “ ” Botón Guardar	El sistema no guarda el estacionamiento y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 51: Casos de prueba CU10.

Casos de prueba de C12: Registrar Nuevo Vehículo

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro del vehículo exitoso	Patente: “lkj789” Socio: “32145698” Estacionamiento: 1 Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema guarda el estacionamiento.
CP2	Registro del vehículo cancelado	Patente: N/A Socio: N/A Estacionamiento: N/A Color: N/A Modelo: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Registro del vehículo fallido ● Vehículo existente.	Patente: “lkj789” Socio: “32145698” Estacionamiento: 1 Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema no guarda el vehículo y muestra el mensaje: Ya existe un vehículo con esos datos.
CP4	Registro del vehículo fallido ● Campos requeridos.	Patente: “ ” Socio: “ ” Estacionamiento: “ ” Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema no guarda el vehículo y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 52: Casos de prueba CU12.

Casos de prueba de C14: Modificar Vehículo

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Modificación del vehículo exitoso	Patente: “lkj777” Socio: “32145698” Estacionamiento: 1 Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema guarda el vehículo.
CP2	Modificación del vehículo cancelado	Patente: N/A Socio: N/A Estacionamiento: N/A Color: N/A Modelo: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Modificación del vehículo fallido ● Vehículo existente.	Patente: “lkj777” Socio: “32145698” Estacionamiento: 1 Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema no guarda el vehículo y muestra el mensaje: Ya existe un vehículo con esos datos.
CP4	Modificación del vehículo fallido ● Campos requeridos.	Patente: “ ” Socio: “ ” Estacionamiento: “ “ Color: “Negro” Modelo: “Ka” Botón Guardar	El sistema no guarda el vehículo y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 53: Casos de prueba CU14.

Casos de prueba de C20: Registrar Entrada

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro de entrada exitoso	Patente: “loi520” Fecha: “2020-12-03” Hora: “09:00” Botón Guardar	El sistema guarda la entrada.
CP2	Registro de entrada cancelado	Patente: N/A Fecha: N/A Hora: N/A Botón Cancelar	El sistema cierra la ventana.
CP3	Registro de entrada fallido ● Usuario existente.	Patente: “loi520” Fecha: “2020-12-03” Hora: “09:00” Botón Guardar	El sistema no guarda la entrada y muestra el mensaje: Ya existe un usuario con esos datos.

CP4	Registro de entrada fallido ● Campos requeridos.	Patente: “ ” Fecha: “ ” Hora: “ ” Botón Guardar	El sistema no guarda la entrada y muestra el mensaje: Los campos requeridos no pueden estar vacíos.

Tabla 54: Casos de prueba CU20.

Casos de prueba de C21: Registrar Salida

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Registro de salida exitoso	Patente: “loi520” Fecha: “2020-12-03” Hora: “10:00” Botón Guardar	El sistema guarda la salida.
CP2	Registro de salida cancelado	Patente: N/A Fecha: N/A Hora: N/A Botón Cancelar	El sistema cierra la ventana.

Tabla 55: Casos de prueba CU21.

Casos de prueba de C22: Validar Código Promoción

ID	Caso de prueba	Datos ingresados	Resultado esperado
CP1	Validación exitosa	Código: “15236” Botón Guardar	El sistema valida.
CP2	Validación cancelada	Código: N/A Botón Cancelar	El sistema cierra la ventana.

Tabla 56: Casos de prueba CU22.

8.6 Anteproyecto

8.6.1 Justificación

La Caja de los Profesionales de la Ingeniería es una es una entidad con personería jurídica de derecho público no estatal, sin fines de lucro. Tiene como fines esenciales proporcionar a todos los profesionales inscriptos en los Colegios Profesionales, que se encuadren en las disposiciones de las leyes provinciales 4889 y 6729, los beneficios de la

cooperación mutua para asegurarles asistencia y seguridad social en condiciones dignas y justas. A diferencia de otros regímenes jubilatorios estatales o privados es administrada por sus propios afiliados y desde 1958, presta servicios asistenciales para el profesional y sus familiares con el fin de una mejor convivencia basada en la seguridad de no quedar en el desamparo en momentos de adversidad [Caja de Ingeniería, 2019].

La afiliación es obligatoria para todos los profesionales que se hallen inscriptos en los siguientes Colegios Profesionales de la Provincia de Santa Fe:

- Colegio de Arquitectos – Distritos 1, 5 y 6 (ley 10.653).
- Colegio de Ingenieros Agrónomos – 1º y 3º Circunscripción (ley 10.780).
- Colegio de Profesionales de la Agrimensura – Distrito Norte (ley 10.781).
- Colegio Profesional de MMO y Técnicos – Distrito 1 (ley 10.946).
- Colegio de Profesionales de la Ingeniería Civil – Distrito 1 (ley 11.008).
- Colegio de Ingenieros Especialistas - Distrito 1

En el proceso de evolución emprendido por la Caja desde el año 2009, para lograr reformular el sentido de la misma y en pos de lograr otorgar mayores y mejores servicios y beneficios, surge el emprendimiento de creación de un nuevo ente que integre todas aquellas actividades que desde el seno de la Caja no se podían realizar. Es así como empezó a madurar la idea de la creación de una Mutual, que sea distinta a las que ya existían en el entorno y que dé respuesta a las necesidades que tenían muchos de los afiliados. Siempre pensando en una única consigna: lograr un mejor bienestar de los afiliados.

El 18 de diciembre de 2012 se logró obtener la habilitación para el funcionamiento de la ASOCIACIÓN MUTUAL AFILIADOS A LA CAJA DE INGENIERÍA (A.M.A.C.I.), otorgada por el Instituto Nacional de Asociativismo y Economía Social –INAES- y regida por la Ley Orgánica para Asociaciones Mutuales N° 20321.

Avanzando en el tiempo, la institución logró dar su primer y gran paso: la creación de **Farmacia Mutual**. El 16 de abril de 2012 se adquirió la habilitación para el funcionamiento de este servicio.

En abril de 2014 quedó inaugurado, junto al Complejo Integrado de Caja de Ingeniería, el **Salón de Eventos**. Un espacio pensado para asociados y sus momentos más importantes. El 16 de junio de 2014 se dio apertura a los **Consultorios Odontológicos**, que dotaron a todos los asociados de una prestación odontológica de excelente calidad, con un cuerpo de profesionales de primer nivel y un equipamiento tecnológico de vanguardia.

En octubre de 2015 lograron la apertura del departamento de **Turismo Mutual**.

A partir de octubre de 2016, se comenzó con el Servicio de Cochera por el cual la Mutual se hizo cargo de la administración de las playas de estacionamiento propiedad de la Caja de Ingeniería a cambio de un alquiler mensual.

Gracias a todos los servicios brindados por Mutual Ingeniería hoy en día cuenta con un padrón de 10.683 asociados activos y adherentes.

En la actualidad las cocheras se encuentran distribuidas en el centro de la ciudad. Poseen un movimiento diario constante y tienen distintas tarifas dependiendo el tipo de servicio que se les brinde, ya sea mensualizado o estacionamiento temporario.

Son 5 playas de estacionamiento que hacen un total de 172 dársenas:

Cochera	Dársenas
25 de Mayo 2171	30
Obispo 2441	41
San Martín 1743	40
San Martin 3017	24
Crespo 2770	37

Tabla 57: Cocheras y sus respectivas dársenas.

La administración es llevada a cabo por un empleado que anota de forma manual el número de patente y el horario de ingreso. Al retirarse el vehículo, el precio a cobrar se calcula según si es estacionamiento medido o de un mensualizado.

La mayoría de los sistemas de administración de cocheras en el mercado actual son provistos por empresas privadas. Tienen las características de ser cerrados y garantizan un desempeño óptimo en la administración y gestión de usuarios, precios y facturación. Además brindan herramientas informativas para permitir optimizar el negocio.

Estos sistemas están compuestos de dos partes, un conjunto de dispositivos electrónicos y una solución informática.

Dispositivos electrónicos: Existen distintos dispositivos (Figura 33): barreras, columna de ingreso/egreso y sensores (dispositivos electrónicos de tipo 1), donde cada uno cumple una función específica. Al ingresar un vehículo y posarse por el primer sensor se activa la columna de ingreso, la cual registra la fecha y hora de entrada y emite el ticket con su correspondiente código de barra. Al retirarse el ticket se abre la barrera que permite la entrada a la cochera y cuando el vehículo pasa por el segundo sensor la barrera se baja. Al salir el dueño del vehículo debe presentar al empleado o en la columna de egreso el ticket para leer el código de barra y que este realice el cálculo del costo. Una vez realizado el pago se abre la barrera.

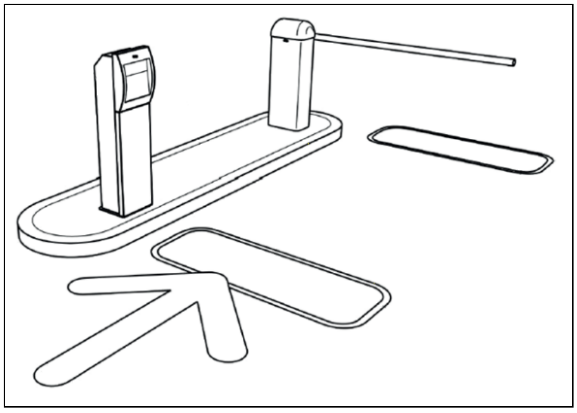

Figura 33: Dispositivos electrónicos [Gestión de Parking, 2019].

Solución informática: Consiste de un sistema relacionado con los dispositivos electrónicos de tipo 1. Al ingresar, el ticket emitido por la columna de ingreso actualiza automáticamente el sistema descontando en uno la cantidad de lugares disponibles en el estacionamiento y empezando a acumular el tiempo de estadía. Al realizar el egreso, la persona encargada de manipular el sistema solicita el ticket al dueño del vehículo, lo lee con el lector de código de barra y realiza el cobro manual en efectivo u otro medio de pago. En caso de que la cabina de cobro esté dentro de la playa, el dueño del vehículo deberá presentar el ticket en la columna de egreso para validar la salida pero anteriormente deberá realizar el pago. De lo contrario no se abrirá la barrera. El sistema incluye instalación y configuración de equipos electrónicos, administración de usuarios y sus permisos, configuración de precios según tiempo de permanencia [Gestión de Parking, 2019].

Para la Mutual Ingeniería, el principal problema es que no existe un control real de ingresos y egresos en las cocheras debido a que se realiza un registro manual. También se dificulta saber con certeza la disponibilidad de estacionamiento teniendo que hacer una revisión constante por parte del empleado lo que implica la ausencia en la cabina de ingreso/egreso. Todo esto conlleva a pérdida de tiempo, registros erróneos y actividad económica ineficiente.

Debido a que los sistemas de terceros poseen un número determinado de funcionalidades, no permiten a la institución cubrir necesidades particulares ni realizar mejoras futuras sin depender de un externo.

En este trabajo se pretende desarrollar una herramienta propia que sea escalable y automatice la administración de las cocheras teniendo la capacidad de la validar códigos de descuentos obtenidos por los socios de la mutual mediante una aplicación móvil existente. También se busca un sistema capaz de diferenciar entre tipos de asociados de la mutual, y que pueda verificar que el vehículo realmente ingresó, comparando si existe una imagen con la misma fecha y hora que el ticket del vehículo. La imagen es obtenida de las cámaras de seguridad las cuales fotografían los ingresos. Con escalable se refiere específicamente a que el sistema es capaz de ir incorporando nuevos módulos hasta poder, en un futuro, agregar y configurar dispositivos electrónicos de tipo 1, así como automatismos más complejos como por ejemplo apertura de barreras mediante llamadas telefónicas o detección automática de patentes.

Los usuarios serán los empleados de las cocheras que registrarán cada ingreso y se encargarán de gestionar el pago de los estacionamientos. Los administradores podrán realizar las altas, bajas y modificaciones de mensualizados, tarifas y promociones, además de poder emitir informes solicitados por la gerencia. Los desarrolladores de la institución serán los que cuenten con los permisos de administrador.

A partir de este proyecto se intentará solucionar los errores que provienen del accionar humano, ahorrar el dinero que implica tercerizar y llevar un control correcto en las transacciones monetarias con la implementación de tickets fiscales. A su vez, será de gran ayuda para la comunidad optimizando tiempo y espacio, así como también para Mutual Ingeniería debido a que el proyecto será implementado en producción y además se podrán obtener informes que servirán como bases para realizar mejoras en la organización.

8.6.2 Objetivos

Objetivo General

Desarrollar un sistema web para la automatización de la administración de cocheras para la Mutual Ingeniería.

Objetivos Específicos

- Diseñar e implementar módulo de administración de distintos tipos de usuario.
- Diseñar e implementar módulo de administración de distintos tipos de tarifas y promociones.
- Diseñar e implementar módulo de administración de distintos tipos de estacionamientos (tipo afiliado, directores, empleados, proveedores) para asignarles horarios y tiempos de estacionamientos gratuitos.
- Diseñar e implementar módulo de integración con dispositivos electrónicos de tipo 2 (impresora fiscal, lector de código de barras) y sistemas externos (aplicación móvil, sistema de cámaras de seguridad).
- Diseñar e implementar módulo de informes estadísticos.
- Diseñar e implementar interfaz gráfica.

8.6.3 Alcance

Se desarrollará un sistema web versionado mediante gitlab para la administración de cocheras que será capaz de comunicarse con una WEB API de Mutual Ingeniería para la obtención de descuentos.

Funcionales

- El sistema deberá administrar distintos tipos de usuarios, estacionamientos, tarifas, promociones.
- El sistema deberá generar tickets fiscales.
- El sistema deberá realizar informes estadísticos.
- El sistema deberá verificar promociones obtenidas mediante aplicación móvil.
- El sistema deberá verificar la existencia de una imagen que asegure el ingreso de un vehículo.

No funcionales

- La sistema deberá ser web, en lenguaje de programación C# .net, utilizando base de datos SQL Server Express.
- El sistema se deberá vincular con una WEB API de Mutual Ingeniería mediante el intercambio de mensajes JSON o XML alojado en https://api.mutualingenieria.org.ar
- El sistema deberá estar documentado y versionado mediante gitlab.
- El sistema deberá ser escalable para poder incorporar nuevos módulos en el futuro.

Exclusiones

- No se realizará el reconocimiento de patentes mediante procesamiento de imágenes.
- No se realizará la vinculación con dispositivos electrónicos de tipo 1.
- No se vinculará con ninguna plataforma de pago online.
- No se contemplará otro medio de pago que no sea efectivo.
- No se realizarán automatismos para apertura de barreras mediante llamadas telefónicas.

Supuestos

- Se dispone de un conjunto de imágenes obtenidas por las cámaras de seguridad para verificación de ingreso.

8.6.4 Metodología

En este proyecto se utilizará un modelo de desarrollo secuencial, el modelo en cascada, que considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y los representa como fases separadas del proceso tales como la especificación de requerimientos, el diseño del software, la implementación y las pruebas, cada una con su respectiva documentación. A estas se le agregará una etapa de investigación y capacitación en la cual se profundizará sobre el uso de la plataforma de desarrollo, la web API y la integración con sistemas externos y dispositivos electrónicos de tipo 2. Esto será necesario debido a que el alumno no ha trabajado aún con estas herramientas.

Si bien en la práctica estas etapas se superponen y proporcionan información unas a otras, la idea principal es que la siguiente fase no debe empezar hasta que la fase previa haya finalizado. El principal problema es que debido a su inflexibilidad al dividir el proyecto en distintas etapas se hace difícil responder a los cambios en los requerimientos del cliente. En este caso particular el cliente cuenta con un equipo de ingenieros en sistemas que entienden la problemática actual y son capaces de describir los requerimientos funcionales y no funcionales, por lo que se espera un claro y correcto informe de los mismos al inicio del proyecto a través de entrevistas personales. Es por esto que el modelo en cascada es una buena opción para la realización de este proyecto ya que es improbable que los requerimientos cambien radicalmente. De todas maneras, en caso de surgir errores y omisiones en los requerimientos originales se deberán repetir etapas previas del modelo [Sommerville, 2005].

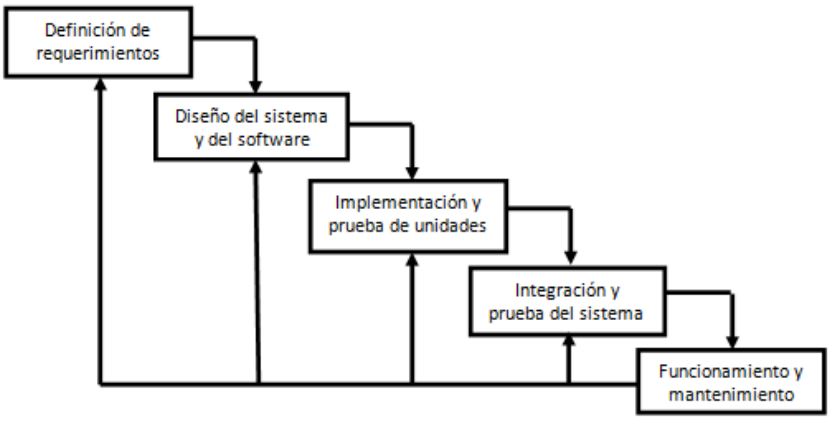

Figura 34: Modelo en cascada [Sommerville, 2005].

Cada una de las etapas contendrá la siguiente información:

- Investigación y capacitación: En esta etapa se llevará a cabo una interiorización sobre las herramientas a utilizar. Esto incluye un estudio del lenguaje C# y de la plataforma .net y cómo realizar la integración de dispositivos electrónicos de tipo 2.
- Análisis y diseño: En esta etapa se realizará el Documento de Requerimientos que contendrá el conjunto de necesidades que dieron lugar al proyecto. Además se realizarán todos los diagramas que detallarán los diagramas de base de datos, diagrama de clases y diagramas de casos de usos. Estos se utilizarán como guía para el desarrollo.
- Desarrollo: En esta etapa se transcribe a código todo lo plasmado en la etapa de diseño y por cada módulo realizado se llevan a cabo sus pruebas unitarias.
- Pruebas y puesta en funcionamiento: En esta etapa se integrarán todos los módulos individuales y se realizarán un conjunto de procesos para comprobar que el software está acorde a sus especificaciones y cumple las necesidades del cliente. Una vez concluidas las pruebas se lleva a cabo la redacción del informe final.

En cada una de estas etapas se irá documentando cada acción y sus respectivos resultados con el objetivo de ir conformando un documento que sea de utilidad para eventuales correcciones, mantenimiento futuro y ampliaciones al sistema. También se irá confeccionando un manual de usuario a medida que se vaya desarrollando cada módulo que será entregado a los encargados de manipular el sistema.

8.6.5 Plan de Tareas

Se estima que al proyecto se le dedicarán 20 horas semanales durante un total aproximado de 18 semanas (345 horas). La estimación de horas se lleva a cabo mediante juicio de expertos, por lo cual se consultó a los directores según su experiencia. Cabe aclarar que, si bien en la etapa de desarrollo todos los módulos tendrán la misma

complejidad, hay una diferencia de cantidad de horas debido a que para los módulos posteriores al primero podrá realizarse reutilización de código.

La fecha de inicio será el Lunes 13 de Agosto de 2019 mientras que se pretende finalizar a mediados del mes de Diciembre del mismo año.

El plan de tareas del proyecto será el siguiente:

Actividad	**Duración**
1. **Investigación y capacitación**	
1.1 Investigación de la plataforma de desarrollo y lenguaje	20 hs
1.2 Investigación de Web API de Mutual Ingeniería	20 hs
1.3 Investigación dispositivos de tipo 2 y sistemas externos	20 hs
Total	**60 hs**
2. Análisis y diseño	
2.1 Creación de Documento de Requerimientos	5 hs
2.2 Diseño de Casos de Usos	5 hs
2.3 Diseño de Diagrama de Clases	10 hs
2.4 Diseño de Diagrama de Base de Datos	10 hs
2.5 Diseño de Interfaz Gráfica	10 hs
Total	**40 hs**
3. **Desarrollo**	
1. Desarrollo de la Base de Datos	10 hs
2. Módulo de administración de usuarios	
a. Desarrollo de modelos	15 hs
b. Desarrollo de vistas	15 hs
c. Desarrollo de controladores	15 hs
d. Prueba unitaria del módulo	5 h
3. Módulo de administración de tarifas y promociones	
a. Desarrollo de modelos	10 hs
b. Desarrollo de vistas	10 hs
c. Desarrollo de controladores	10 hs
d. Prueba unitaria del módulo	5 hs
4. Módulo de administración de estacionamientos	
a. Desarrollo de modelos	10 hs
b. Desarrollo de vistas	10 hs
c. Desarrollo de controladores	10 hs
d. Prueba unitaria del módulo	5 hs
5. Módulo de integración de dispositivos tipo 2 y sistemas externos	
a. Desarrollo de modelos	10 hs
b. Desarrollo de vistas	10 hs
c. Desarrollo de controladores	10 hs
d. Prueba unitaria del módulo	5 hs
Total	**165 hs**
4. **Pruebas y puesta en funcionamiento**	

4.1 Pruebas integrales de todos los módulos juntos 4.2 Redacción de informe final **Total**	20 hs 60 hs **80 hs**
Total	**345 hs**

Tabla 58: Cronograma.

8.6.6 Cronograma

En la Figura 35 se observa el cronograma de las actividades del proyecto mediante un Diagrama de Gantt.

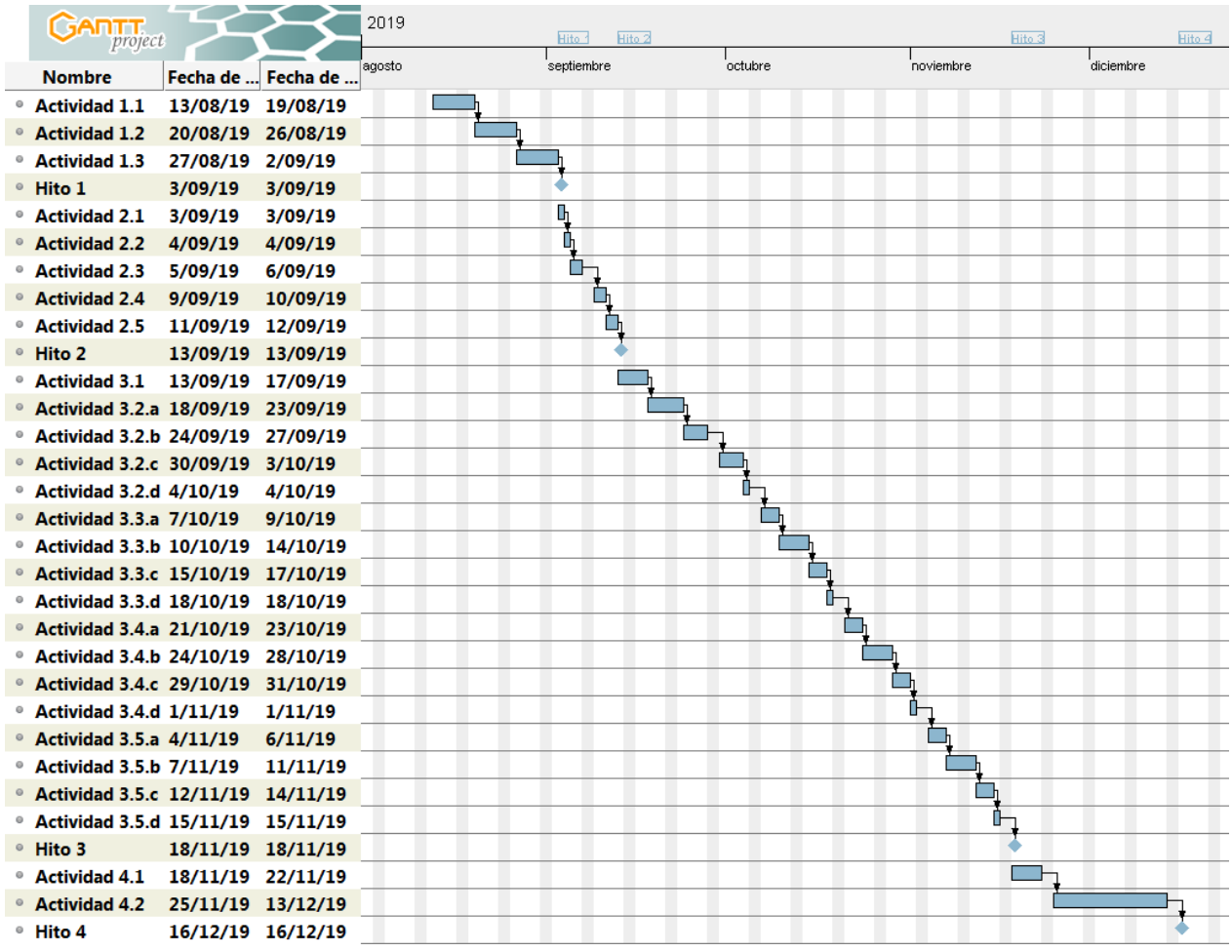

Figura 35: Diagrama de Gantt de las actividades.

8.6.7 Entregables

Etapa 1: Investigación y capacitación

- Entregable: Se entregará un documento a manera de resumen sobre los nuevos conceptos que servirán para la comprensión e interiorización de las herramientas, dispositivos electrónicos de tipo 2 y sistemas que se van a utilizar en el proyecto.
    - Criterio de aceptación: El documento debe estar aprobado por los directores.
- Hito 1: 3/09/2019. Documento resumen realizado.

Etapa 2: Análisis y diseño
- Entregable: Se entregará el documento de requerimientos obtenido a través de entrevistas, así como también los diagramas de base de datos, de clases, de casos de uso y diseño de interfaz que serán de vital importancia para la elaboración de los modelos, vistas y controladores.
    - Criterio de aceptación: El documento debe estar aprobado por los directores.
- Hito 2: 13/09/2019. Documento de requerimientos y diagramas diseñados.

Etapa 3: Desarrollo
- Entregable: Se entregarán las implementaciones de los módulos de administración de usuarios, tarifas, promociones, estacionamientos y dispositivos electrónicos de tipo 2.
    - Criterio de aceptación: La implementación debe estar aprobada por los directores.
- Hito 3: 18/11/2019. Implementación de módulos realizada y funcionando.

Etapa 4: Pruebas y puesta en funcionamiento
- Entregable: Se entregará un informe de estado obtenido a partir de las pruebas. También se entregará el informe final terminado.
    - Criterio de aceptación: El funcionamiento del sistema y el informe final deben estar aprobados por los directores.
- Hito 4: 16/12/2019. Sistema validado funcionando e informe final realizado.

Informe de avance 1:
- Contenido: Este informe contendrá un resumen de C# .net, todos los diagramas de la etapa de diseño, el Documento de Requerimientos, unas consultas básicas a la base de datos desarrollada y una conclusión de cómo fueron los resultados en la realización de las etapas 1 y 2, y las sección 3.1.
- Fecha de entrega estimada: 19/09/2019.

Informe de avance 2:
- Contenido: Este informe contendrá capturas de las pruebas unitarias a los módulos desarrollados y una conclusión de cómo fueron los resultados en la realización de de las secciones 3.2, 3.3 y 3.4.
- Fecha de entrega estimada: 5/11/2019.

Informe de avance 3:
- Contenido: Este informe contendrá capturas de las pruebas integrales del sistema y una conclusión de cómo fueron los resultados en la realización de la sección 3.5 y la etapa 4.

- Fecha de entrega estimada: 19/12/2019.

8.6.8 Riesgos

Los riesgos que se detallan en esta etapa son los identificados al inicio del proyecto, en caso de que aparezcan nuevos riesgos en el transcurso del proyecto serán agregados y justificados.

Los riesgos se clasifican por orden de prioridad de acuerdo con sus implicaciones potenciales sobre los objetivos del proyecto. Se priorizan mediante una tabla de búsqueda o una matriz de probabilidad e impacto que establecen importancia “alta”, “media” o “baja” y de acuerdo a esto se decidirá la estrategia que se efectuará para cada riesgo. [Guía del PMBOOK, 2009]

Probabilidad/Impacto	**Bajo (1)**	**Medio (2)**	**Alto (3)**
Baja (1)	Aceptar	Aceptar	Transferir o Mitigar
Media (2)	Aceptar	Transferir o Mitigar	Evitar
Alta (3)	Transferir o Mitigar	Evitar	Evitar

Tabla 59: Matriz de estrategias de respuesta al riesgo.

Los riesgos identificados fueron los siguientes:

1. **Caída del servidor web donde se aloja el sistema**
   El servidor debe estar disponible en todo momento para poder realizar las tareas especificadas en el cronograma.

Probabilidad	Baja
Impacto	Alto
Estrategia	Mitigar: Realizar backups cada cierto período de tiempo para tener siempre disponible los archivos del sistema.
Contingencia	Continuar trabajando en el servidor local con la última versión hasta que se recupere el servidor web.

Tabla 60: Riesgo Nro 1.

2. **Pérdida de recursos de hardware**
   El hardware que se utiliza para llevar a cabo el proyecto puede sufrir problemas inesperados que retrasen el cronograma o provoquen pérdida de información.

Probabilidad	Baja
Impacto	Medio
Estrategia	Aceptar: Reemplazar el hardware temporal o definitivamente para continuar con la realización del proyecto.
Contingencia	Continuar con el cronograma hasta conseguir otro recurso.

Tabla 61: Riesgo Nro 2.

3. **Indisponibilidad de un recurso humano para cumplir con el cronograma**
   Debido al surgimiento de ofertas laborales o de problemas de salud el alumno no puede llegar a cumplir con las condiciones del cronograma.

Probabilidad	Baja
Impacto	Medio
Estrategia	Aceptar
Contingencia	Se incrementarán las horas de trabajo diarias para compensar y cumplir con el cronograma en caso de que el recurso sea el alumno. Si el recurso humano es alguno de los directores se incrementará la frecuencia de la reuniones o se cambiará alguno de los directores.

Tabla 62: Riesgo Nro 3.

8.6.9 Recursos

Todos los recursos que el alumno necesitará para realizar el proyecto se encuentran disponibles al inicio del mismo.

- Recursos humanos:
  - Alumno: Encargado de llevar a cabo las etapas investigación y capacitación, análisis y diseño, desarrollo, pruebas y puesta en funcionamiento.
  - Director y codirectora: Encargados de revisar y corregir el contenido del proyecto y los informes.

- Hardware:
  - Notebook Acer Aspire 4738z, procesador Intel Pentium, CPU P6100 2 GHz, 2 Gb ram, 320 disco.
  - Impresora láser.

- Software
  - Windows 7 Home Basic 64 bits.

- Visual Studio 2019.
- SQL Server Express.
- Control de versiones gitlab.
- Herramientas de Google Drive para la redacción de informes.

- Equipamiento e insumos
  - Insumos de oficina: Lapicera, hojas, toners, fibron, tinta para fibron.
  - Conexión a internet.
  - Luz eléctrica.
  - Habitación con escritorio, pizarra,

8.6.10 Presupuesto

A continuación se detalla el presupuesto requerido para la realización del proyecto. La amortización se calcula mediante:
Amortización = [(Valor a nuevo - Valor residual) / Vida útil (hs)] * Horas uso requerido

Razón	**Detalle**	**Concepto**	**Tipo**	**Total**
Bienes de capital				
Notebook	Acer Aspire 4738z VN= $16000 VR= $7000 Vida útil= 10000 Hs. Uso requerido= 345 Hs.	Amortización	Costo directo	$310.5
Impresora láser	Brother HI-1212w VN= $7000 VR= $3000 Vida útil= 10000 Hs. Uso requerido= 345 Hs.	Amortización	Costo directo	$138
Recursos humanos				
Alumno	Programador. Valor hora: $1179 [*] Horas: 345	Gasto	Costo directo	$406755
Director	Project Manager. Valor hora: $1193 [*] Horas: 50	Gasto	Costo directo	$59650

Codirectora	Project Manager. Valor hora: $1193 [*] Horas: 50	Gasto	Costo directo	$59650
Viajes y viáticos				
Transporte	Transporte interurbano: $70 (por día) Días: 126	Gasto	Costo directo	$8820
Almuerzo	Comedor universitario: $48 (por día) Días: 126	Gasto	Costo directo	$6048
Otros costos				
Internet	Valor mensual: $1025 Meses: 4.5	Gasto	Costo indirecto	$4612.5
Luz	Valor mensual: $1500 Meses: 4.5	Gasto	Costo indirecto	$6750
Equipamiento e insumos				
Insumos de oficina	Lapicera: $20 Hojas: $200 Toner: $544 Fibron: $70 Tinta fibron: $300	Gasto	Costo indirecto	$1134
Total				$553868

Tabla 63: Presupuesto. [*] obtenidos de http://coprocier.org.ar/web/?page_id=53

Capítulo 9

9. Bibliografía

Caja de Ingeniería (2019). Santa Fe, Argentina. Recuperado de: http://www.cajaingenieria.org/la-caja

Gestión de Parking (2019). Proytec [Figura 1]. Córdoba, Argentina. Recuperado de: http://www.proytec.com.ar/pdf/g/Proytec%20-%20Descripcion%20del%20Sistema%20de%20Parking.pdf

Gestión de Parking (2019). Proytec. Córdoba, Argentina. Recuperado de: http://www.proytec.com.ar/pdf/g/Proytec%20-%20Descripcion%20del%20Sistema%20de%20Parking.pdf

Ian Sommerville (2005). Ingeniería del software. Madrid. España. PEARSON Addison Wesley.

Información general de ASP.NET Core MVC (2019). Microsoft. España. Resuperado de: https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-2.2

Patrón de diseño de modelo-vista-controlador (2019). IBM. Recuperado de: https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm

Project Management Institute (2009). Guía de los fundamentos para la dirección de proyectos (Guía del PMBOOK). Pennsylvania. EE.UU. PMI Publications.

¿Qué es una base de datos relacional?, (2020), Recuperado de: https://www.oracle.com/ar/database/what-is-a-relational-database/

Bases de datos, Microsoft, (2020), Recuperado de: https://docs.microsoft.com/es-es/sql/relational-databases/databases/databases?view=sql-server-ver15

HTML Tutorial, W3schools, (2019), Recuperado de: https://www.w3schools.com/html/default.asp

JavaScript Introduction, W3schools, (2019), Recuperado de: https://www.w3schools.com/js/js_intro.asp

¿Qué es Bootstrap?, DevCode, (2019), Recuperado de: https://devcode.la/blog/que-es-bootstrap/

AJAX Introduction, W3schools, (2019), Recuperado de: https://www.w3schools.com/js/js_ajax_intro.asp

JQuery Tutorial, W3schools, (2019), Recuperado de: https://www.w3schools.com/jquery/default.asp https://es.wikipedia.org/wiki/JQuery

Introducción C#, GoDotEngine, (2019), Recuperado de: https://docs.godotengine.org/es/latest/getting_started/scripting/c_sharp/c_sharp_basics.html https://es.wikipedia.org/wiki/C_Sharp

¿Qué es el .NET? ¿Para qué sirve?, Emagister, (2019), Recuperado de: https://www.emagister.com/blog/que-es-el-net-para-que-sirve/

.NET Framework Documentation, Microsoft, (2019), Recuperado de: https://docs.microsoft.com/es-es/dotnet/framework/

Introducción a .NET Framework, Microsoft, (2019), Recuperado de: https://docs.microsoft.com/es-es/dotnet/framework/get-started/index

Información general de ASP.NET, Microsoft, (2019), Recuperado de: https://docs.microsoft.com/es-es/aspnet/overview

Referencia de sintaxis de Razor para ASP.NET Core, Microsoft, (2019), Recuperado de: https://docs.microsoft.com/es-es/aspnet/core/mvc/views/razor?view=aspnetcore-2.2

Información general de ASP.NET Core MVC, Microsoft, (2019), Recuperado de: https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-2.2

Microsoft SQL Server, Wikipedia, (2019), Recuperado de: https://es.wikipedia.org/wiki/Microsoft_SQL_Server

Todo lo que debes saber sobre Gitlab, la mejor alternativa a Github, Tekcrispy, (2019), Recuperado de: https://www.tekcrispy.com/2018/06/04/gitlab-repositorio-github/

¿Qué es y para qué sirve Visual Studio 2017?, Msn noticias, (2019), Recuperado de: https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAnLZL9

Visual Studio 2019, Microsoft, (2019), Recuperado de: https://visualstudio.microsoft.com/es/vs/

API de Mutual Ingeniería, Mutual Ingeniería, (2019), Recuperado de: https://apiprueba.mutualingenieria.org.ar/

¿Qué diferencia hay entre impresora térmica y una matricial?, ComercialTPV, (2019), Recuperado de: https://www.comercialtpv.com/blog/wp_super_faq/que-diferencia-hay-entre-una-impresora-termica-y-una-matricial/

¿Qué es una impresora fiscal y cómo funciona?, Impulsa Popular, (2019), Recuperado de: https://www.impulsapopular.com/legal/que-es-una-impresora-fiscal-y-como-funciona/

El lector de código de barras, Informática moderna, (2019), Recuperado de: http://www.informaticamoderna.com/Lector_codigos.htm#defs

C# Asp.Net MVC (Ajax + Json +Bootstrap + JavaScript, Udemy, (2019), Recuperado de: https://www.udemy.com/course/c-aspnet-mvc-ajax-json-boostrap-javascript/learn/lecture/15570618?start=75#overview

C# Online Compiler, .NET Fiddler, (2019), Recuperado de: https://dotnetfiddle.net/

Online JavaScript Editor, PlayCode, (2019), Recuperado de: https://playcode.io/online-javascript-editor

Flowchart Maker and Online Diagram Software, Draw.io, (2019), Recuperado de: https://www.draw.io/

Quick Start, Google, (2019), Recuperado de: https://developers.google.com/chart/interactive/docs/quick_start