

DIGITALIZACIÓN DE PLAYAS DE ESTACIONAMIENTO CON TECNOLOGÍA SERVERLESS Y BASES DE DATOS EN TIEMPO REAL



Francisco Alfredo Jarma

Ingeniería en Informática

Facultad de Ingeniería

Universidad Católica de Salta

2021

Título

Ingeniería en Informática

Profesor guía

Nombre: Lic. Patricio Arredes Iway

Firma: _____

Tribunal evaluador:

Nombre: _____

Firma: _____

Nombre: _____

Firma: _____

Nombre: _____

Firma: _____

Fecha de exposición:

____/____/____

Dedicatorias y agradecimientos

A mi **mamá**, Olga, sin ella nada de esto hubiese sido posible.

A mis **hermanos**, Eduardo, Nicolás y Martín por el constante apoyo incondicional que recibo de ellos todos los días.

A mis **tíos**, Andrés y Nilda, por estar siempre, en los buenos y malos momentos.

A mi **novia**, Micaela, por haberme acompañado durante varios años y por ser mi compañera en todo.

A mis **amigos y compañeros**, con los cuales compartí muchos momentos en el transcurso de esta carrera.

Al **Lic. Patricio Arredes**, que supo guiarme y aconsejarme con sabiduría a lo largo del desarrollo de este trabajo.

Al resto de los profesores de la carrera, que con su conocimiento he aprendido muchas cosas que me servirán tanto en mi vida personal como en mi vida profesional.

Índice del trabajo

Abstract	10
Capítulo I: Introducción	11
1.1 Descripción del problema	11
1.2 Importancia del problema	11
1.3 Motivación para abordarlo	12
1.4 Pasos a realizar	12
1.5 Criterios de éxito	13
Capítulo II: Estado de la cuestión	14
2.1 Conceptos teóricos	14
2.1.1 Desarrollo a medida	14
2.1.2 API	14
2.1.3 SPA	14
2.1.4 La nube	15
2.1.5 Patrones de diseño web	15
2.1.6 Diseño web adaptable	15
2.1.7 Software como Servicio (SAAS)	15
2.1.8 E-commerce	16
2.1.10 Bases de datos en tiempo real	18
2.1.11 Bases de datos en la nube	18
2.1.12 Bases de datos no relacionales	19
2.1.13 Geolocalización	20
2.1.14 Seguridad en la capa de transporte (SSL – TSL - HTTPS)	20
2.2 Antecedentes relacionados con el problema	22
2.2.1 Multas y secuestro de vehículos	22
2.2.2 Infracciones por jurisdicción	23
2.3 Soluciones similares	25
2.3.1 Parqueo App	25
2.3.2 Westmart Park	25
2.3.4 Ingesys	25
2.4 Temas pendientes de resolución	26
2.4.1 Alcance a futuro	26
2.4.2 Implementación de servicios adicionales	26
2.4.3 Adaptar la solución a todo tipo de lugares	26
2.4.4 Aplicación móvil	26
2.4.5 Implementación de código QR con lectores	26
2.4.6 Reconocimiento de patentes	27
2.4.7 Gestión de convenios y descuentos	27

2.4.8	Suscripciones por planes	27
Capítulo III: Definición del problema		28
3.1	Objetivo general	28
3.2	Objetivos específicos	28
3.3	Alcance del proyecto	28
3.3.1	Descripción del alcance del proyecto	28
3.3.2	Exclusiones del proyecto	30
3.3.3	Restricciones o límites del trabajo	30
3.4	Alternativas tecnológicas	31
Capítulo IV: Solución propuesta		33
4.1	Herramientas utilizadas	33
4.1.1	Windows 10	33
4.1.2	React JS	33
4.1.3	Firebase	34
4.1.4	Google Maps	34
4.1.5	Mercadopago	34
4.1.6	Node JS	34
4.1.7	Express JS	34
4.1.8	Visual Studio Code	35
4.1.9	Git	35
4.1.10	Netlify	35
4.1.11	Heroku	35
4.1.12	Github	35
4.1.13	Material UI	35
4.2	Metodología de gestión de proyectos: Guía del PMBOK	36
4.3	Metodologías de desarrollo: Iterativo Incremental y Prototipado	36
4.4	Aplicación de metodologías de desarrollo	37
4.4.1	Esbozo de requerimientos	37
4.4.2	Incrementos	39
4.4.2	Historias de usuario	39
4.5	Planificación del proyecto	57
4.6	Arquitectura y funcionamiento del sistema	60
4.7	Seguridad del sistema	73
4.7.1	Roles y permisos	73
4.7.2	Manejo de rutas	74
4.7.3	Reglas de seguridad de firebase	75
4.7.4	Sesión del usuario	76
4.7.5	HTTPS - TSL	76
4.8	Análisis de factibilidad	78

4.8.1	Factibilidad técnica	78
4.8.2	Factibilidad operativa	78
4.8.3	Factibilidad económica-financiera	78
4.8.4	Factibilidad ambiental	83
4.8.5	Factibilidad legal	83
4.9	Análisis FODA	83
4.9.1	Análisis interno	83
4.9.2	Análisis externo	83
4.9.3	Estrategias a implementar	84
4.10	Análisis de Riesgos	85
Capítulo V: Resultados		88
5.1	Resultados obtenidos con uso de sistema	88
5.2	Dificultades no previstas encontradas y sus soluciones	89
5.2.1	Vulneración de credenciales del proyecto	90
5.2.2	Reglas no seguras en la base de datos	91
5.3.3	Acceso no autorizado de una dirección IP al proyecto	91
Capítulo VI: Conclusiones y futuras líneas de trabajo		93
Capítulo VII: Bibliografía.		94
Capítulo VIII: Anexos		97
7.1	Normativas	97
7.1.1	Facturación en guarderías de vehículos y playas de estacionamiento	97
7.1.2	Obligación para playas de estacionamiento de emitir tickets o comprobantes de ingreso de vehículos	98
7.1.3	Playas de Estacionamiento y Garage. Ciclomotores y motocicletas. Se establece la obligación de fijar una tarifa	99
7.1.4	Publicación de leyenda en carteles o tickets en playas de estacionamiento vehicular	99
7.1.5	Registro de playas de estacionamiento	100
7.2	Credenciales de usuarios de prueba	102
7.3	Cómo instalar el proyecto localmente	102

Índice de Tablas

TABLA 1: Ventajas y desventajas de SAAS.	16
TABLA 2: Esbozo de requerimientos.	39
TABLA 3: Historia de Usuario: Dar de alta una reserva.	42
TABLA 4: Historia de Usuario: Generar pago de una reserva.	42
TABLA 5: Historia de Usuario: Listar reservas del día en tiempo real.	43
TABLA 6: Historia de Usuario: Ver playas de estacionamiento.	44
TABLA 7: Historia de Usuario: Ver mis reservas.	45
TABLA 8: Historia de Usuario: Ver lugares de mi estacionamiento.	46
TABLA 9: Historia de Usuario: Modificar datos del estacionamiento y tarifas.	46
TABLA 10: Historia de Usuario: Administrar horarios y días de apertura.	47
TABLA 11: Historia de Usuario: Ver usuarios.	47
TABLA 12: Historia de Usuario: Modificar datos de usuario.	47
TABLA 13: Historia de Usuario: Dar de alta usuario.	48
TABLA 14: Historia de Usuario: Dar de baja usuario.	48
TABLA 15: Historia de Usuario: Dar de alta usuario.	49
TABLA 16: Historia de Usuario: Dar de alta playa de estacionamiento.	49
TABLA 17: Historia de Usuario: Dar de baja playa de estacionamiento.	50
TABLA 18: Historia de Usuario: Autenticar mi usuario.	50
TABLA 19: Historia de Usuario: Cerrar sesión.	50
TABLA 20: Historia de Usuario: Cambiar fecha de reserva.	51
TABLA 21: Historia de Usuario: Recuperar contraseña y recibir email de confirmación.	51
TABLA 22: Historia de Usuario: Modificar mis datos personales.	51
TABLA 23: Historia de Usuario: Generar reportes.	52
TABLA 24: Historia de Usuario: Cambiar contraseña.	52
TABLA 25: Historia de Usuario: Buscar reservas.	52
TABLA 26: Historia de Usuario: Registrar nuevo comentario y valoración.	52
TABLA 27: Historia de Usuario: Votar un comentario.	53
TABLA 28: Historia de Usuario: Listar mis vehículos.	53
TABLA 29: Historia de Usuario: Dar de alta mi vehículo.	53
TABLA 30: Historia de Usuario: Modificar datos de mi vehículo.	53
TABLA 31: Historia de Usuario: Dar de baja mi vehículo.	54
TABLA 32: Historia de Usuario: Dar de baja mi vehículo.	54
TABLA 33: Historia de Usuario: Ver mis mensualidades.	54
TABLA 34: Historia de Usuario: Dar de baja mensualidad.	55
TABLA 35: Historia de Usuario: Ver usuarios mensualizados.	55
TABLA 36: Historia de Usuario: Ver reservas de un día en particular.	55
TABLA 37: Historia de Usuario: Abrir estacionamiento.	55

TABLA 38: Costo de RR.HH.	78
TABLA 39: Costo de Software necesario.	79
TABLA 40: Costo de Hardware necesario	79
TABLA 41: Costo total del proyecto.	80
TABLA 42: Análisis de flujo de efectivo del proyecto.	82
TABLA 43: Listado de riesgos del proyecto.	86
TABLA 44: Datos sobre margen de ventas de playas de estacionamiento relevadas.	88
TABLA 45: Datos sobre margen de gastos de playas de estacionamiento relevadas.	89

Índice de Ilustraciones

FIGURA 1: Ejemplo de diseño web adaptable.	15
FIGURA 2: Convergencia de los programas de seguridad de cada una de las marcas en el estándar PCI DSS.	17
FIGURA 3: Ejemplo de Certificado SSL del Sistema de Autogestión de UCASAL	21
FIGURA 4: Encabezado de noticia.	23
FIGURA 5: Encabezado de noticia.	23
FIGURA 6: Gráfico estadístico de infracciones por jurisdicción.	24
FIGURA 7: Encabezado de noticia.	25
FIGURA 8: Estadísticas sobre herramientas basadas en Javascript.	33
FIGURA 9: Proceso de ciclo de vida incremental.	36
FIGURA 10: Formato de Historia de Usuario.	40
FIGURA 11: Diagrama de Gantt del Proyecto.	58
FIGURA 12: Arquitectura del sistema desarrollado.	60
FIGURA 13: Formato de las funciones de firebase authentication.	61
FIGURA 14: Dashboard de firebase authentication.	62
FIGURA 15: Dashboard de firestore database.	63
FIGURA 16: Pantalla de login del sistema.	63
FIGURA 17: Pantalla de recuperar contraseña del sistema.	64
FIGURA 18: Formato de email enviado automáticamente por el sistema.	64
FIGURA 19: Formato de funciones de firestore.	65
FIGURA 20: Pantalla de listar estacionamientos.	66
FIGURA 21: Pantalla de detalles de estacionamientos.	67
FIGURA 22: Pantalla de registrar nueva reserva.	67
FIGURA 23: Pantalla de listar mis reservas.	67
FIGURA 24: Pantalla de listar reservas del día en tiempo real.	68
FIGURA 25: Pantalla de ver tickets.	69
FIGURA 26: Pantalla de validar reserva.	69
FIGURA 27: Pantalla de concluir reserva.	69
FIGURA 28: Pantallas de reportes.	70
FIGURA 29: Checkout de mercadopago.	71
FIGURA 30: Checkout de mercadopago – Elección de medio de pago.	71
FIGURA 31: Funcionalidad de checkout.	72
FIGURA 32: Barra lateral del sistema.	74
FIGURA 33: Alertas del sistema.	75
FIGURA 34: Certificados SSL del proyecto.	77
FIGURA 35: Gráfico de costos por mes del proyecto.	80

FIGURA 36: Gráfico de Retorno de la inversión inicial.....	82
FIGURA 37: Matriz de Probabilidad - Impacto.....	85
FIGURA 38: Gráfico de aumento de ventas de playas de estacionamiento relevadas.....	88
FIGURA 39: Gráfico de disminución de gastos de playas de estacionamiento relevadas.....	89
FIGURA 40: Correo electrónico enviado por google por alerta de vulneración de credenciales del proyecto.....	90
FIGURA 41: Correo electrónico enviado por google por alerta de reglas de seguridad no seguras.	91
FIGURA 42: Regla de seguridad implementada.....	91
FIGURA 43: Correo electrónico enviado por google por alerta de acceso no autorizado al sistema.....	92
FIGURA 44: Normativa sobre facturación en playas de estacionamiento.....	97
FIGURA 45: Normativa sobre obligación de emisión de tickets.....	98
FIGURA 46: Normativa sobre tarifas en playas de estacionamiento.....	99
FIGURA 47: Normativa sobre leyendas de tickets de playas de estacionamiento.	100
FIGURA 48: Normativa registros de playas de estacionamiento.	101
FIGURA 49: Mensaje de verificación de instalación correcta de node.	102
FIGURA 50: Mensaje de verificación de instalación correcta de servidor de pagos.....	103

Abstract

Este proyecto consiste en el *desarrollo de un sistema web a medida* para la reserva y gestión de estacionamientos pertenecientes a diferentes playas ubicadas en Argentina. Se implementará como un *Software como Servicio (SAAS)*, con diferentes tecnologías web bastante novedosas y utilizadas hoy en día, de tal manera que no se necesitarán analizar costos de equipamiento para su implementación y posterior alojamiento, por ej: servidores.

Estará alojado en la nube y se desarrollará de tal forma que cualquier persona que posea algún dispositivo y con acceso a Internet y con cualquier Sistema Operativo pueda utilizarlo sin problemas, y esto debido a que se lo diseñará cumpliendo diferentes *patrones de diseño adaptable* que permite que pueda visualizarse en cualquier dispositivo, independientemente si es un teléfono celular, una computadora, una Tablet, etc.

Capítulo I: Introducción

1.1 Descripción del problema

En la actualidad, particularmente en la zona norte de Argentina, la administración de las playas de estacionamiento se realiza manualmente, existen pocas soluciones informatizadas relacionadas con el problema de encontrar un lugar en el cual estacionar cualquier tipo de vehículo, de tal forma que los usuarios puedan agilizar los procesos de búsqueda, reserva y posterior pago del mismo.

Esto genera varios problemas tanto para las personas que realizan dichos procesos como para los encargados de las playas de estacionamiento, entre los cuales podemos distinguir:

Para las personas que requieren realizar una reserva:

- Tiempos de espera excesivos en algunos casos.
- Gastos adicionales de combustible al momento de querer encontrar una playa de estacionamiento libre.
- Tiempo perdido en otras actividades, por ejemplo: demoras en trámites en el centro, demoras para llegar al trabajo, etc.
- Multas o secuestros del vehículo, todo esto generado por mal estacionamiento del mismo en la vía pública.

Para los encargados de los estacionamientos:

- Muchos gastos de papel para los comprobantes, tickets, facturas, talonarios, etc.
- Mala gestión financiera en la mayoría de los estacionamientos, es decir, una mala administración del dinero que ingresa a la misma.
- Se genera un desgaste en el personal de trabajo.

Y, además, todos estos problemas al mismo tiempo generan disgusto en los clientes y también deja una mala imagen en el servicio prestado por las playas de estacionamiento.

1.2 Importancia del problema

Es muy importante tener una solución informática de este tipo para poder evitar o en todo caso, disminuir todos los problemas mencionados anteriormente.

Además, mediante la implementación del mismo, se garantiza que se reduzcan considerablemente costos relacionados con tiempos de espera en el usuario y principalmente combustible de los vehículos.

También, es importante mencionar la facilidad y comodidad que se le ofrece al usuario, ya que solamente necesita un dispositivo con acceso a Internet para poder realizar la reserva y el pago desde el lugar en el que se encuentre, de tal manera que se evite la búsqueda innecesaria de playas de estacionamiento.

Otro aspecto muy importante a tener en cuenta, es que la informatización es inevitable, en varios aspectos de la vida, la tecnología, particularmente las *tecnologías de la información* (de ahora en más *TI*) están presentes para dar solución a problemas de cualquier naturaleza.

Por lo cual, es importante tratar de reducir la *resistencia al cambio* a través de la implementación de soluciones como estas.

1.3 Motivación para abordarlo

Los motivos que justifican el desarrollo de este proyecto son varios, primero están relacionados con el uso de tecnologías web de última generación, para dar solución a problemas como este.

Por otro lado, en Salta y en varias otras provincias del país no existen muchas soluciones informatizadas que respondan a las necesidades de este tipo, por lo que conlleva un reto bastante importante. Primero por el lado del usuario, para garantizar una mejor calidad de servicio y por otro lado para los trabajadores de las playas de estacionamiento, para que puedan gestionar y administrar de una manera más eficiente la forma de operar sus lugares de trabajo.

Otra cosa a tener en cuenta es el gran crecimiento que sufre día a día el mercado automotor, lo cual provoca una gran necesidad por parte de los conductores al querer encontrar soluciones como éstas.

Hay que aclarar que, en Salta, particularmente, se tiene una cultura muy asociada al modelo relacional de datos y a la utilización de servidores propios, por lo tanto, se pretende demostrar que existen tecnologías innovadoras muy utilizadas en el mundo, que pueden dar las mismas soluciones a problemas del día a día.

Por último y no menos importante, la situación vivida mundialmente estos años con la pandemia de COVID 19, la implementación de una solución como ésta permitiría evitar varios escenarios que se presentan en un contexto como este, por ejemplo: el distanciamiento social, el evitar el contacto con objetos que hayan sido manipulados por otra persona, es decir, en un futuro, de ser necesario, podrá implementarse para ir en conjunto con estos lineamientos.

1.4 Pasos a realizar

El trabajo estará estructurado de la siguiente manera:

1. **Capítulo II: “Estado de la cuestión”**, en este capítulo se explicará de forma detallada las soluciones existentes en el mercado, de tal forma que se pueda ver qué aspectos coinciden con lo planteado en este proyecto y, además, que aspectos faltan en estos y de qué manera este proyecto puede generar un valor agregado respecto a los mismos.
2. **Capítulo III: “Definición del problema”**, en este capítulo se detallará de forma completa el problema a solucionar con la implementación del proyecto, también los objetivos del mismo, el alcance, los límites del trabajo, las alternativas tecnológicas a utilizar. Todo esto teniendo en cuenta la metodología de gestión de proyecto que se utilizará a lo largo del trabajo.
3. **Capítulo IV: “Solución propuesta”**, en este capítulo se detallará la solución propuesta, es decir, se realizarán todos los pasos a seguir para el diseño y desarrollo del sistema para la reserva de estacionamientos, teniendo en cuenta los problemas planteados anteriormente.

4. **Capítulo V: “Resultados”**, en este capítulo se detallarán los resultados obtenidos a partir del capítulo anterior, es decir, en este punto describiremos todos los datos obtenidos a través del uso del sistema desarrollado.
5. **Capítulo VI: “Conclusiones”**, en este capítulo se formulará una conclusión respecto al proyecto desarrollado, teniendo en cuenta el impacto que tuvo en los usuarios y de qué manera se pudieron concretar los objetivos planteados inicialmente.
6. **Capítulo VII: “Bibliografía”**, se listarán todos los libros, artículos, documentos, etc, que fueron utilizados a lo largo del desarrollo del proyecto, asimismo con los autores correspondientes, el año de publicación y la editorial si es necesario.
7. **Capítulo VIII:” Anexos”**, acá se incluye todo el material extra que no se puso en el desarrollo del proyecto, pero se hace una referencia implícita.

1.5 Criterios de éxito

Los criterios que delimitarán el éxito del trabajo a realizar son los siguientes:

1. Que el sistema desarrollado permita cumplir los objetivos planteados en el trabajo y que pueda mejorar la calidad de los servicios prestados por las playas de estacionamientos de diferentes partes del país.
2. Aplicación correcta de las metodologías mencionadas en el trabajo, tanto para el desarrollo como para la gestión del mismo.
3. Que el sistema pueda ser utilizado por cualquier persona con un dispositivo tecnológico que tenga acceso a Internet. Y que, además, pueda visualizar el mismo contenido sin importar el que utilice.
4. Demostrar que es posible desarrollar una solución de negocio que normalmente es realizada con bases de datos relacionales y servidores locales utilizando bases de datos no relacionales y tecnologías en la nube.

Capítulo II: Estado de la cuestión

2.1 Conceptos teóricos

2.1.1 Desarrollo a medida

El *desarrollo de software a medida* es una solución adoptada cuando no es posible disponer de una herramienta o aplicación estándar que ayude a sistematizar las operaciones de una empresa. Esto debido a que cada empresa es diferente, y todas tienen un modelo de negocio diferente al resto. El desarrollo a medida busca que el software se adapte a los procesos específicos de cada organización.

2.1.2 API

API es una abreviatura de *Application Program Interface*, que significa interfaz de programación de aplicaciones, y no es nada más que un conjunto de protocolos y estándares para desarrollar y mantener integrados el software de las aplicaciones web. Permite la comunicación de diferentes soluciones en la web, sin la necesidad de conocer cómo están desarrollados. Esto simplifica notablemente el desarrollo de las mismas y permite ahorrar tiempo y costos.

Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita del consumidor (la llamada) y el que requiere el productor (la respuesta), ambos utilizan el protocolo HTTP como medio de comunicación, específicamente, una serie de verbos que hacen referencia a operaciones que se pueden realizar (get, post, put, etc.).

2.1.3 SPA

SPA es un acrónimo de *Single Page Application*, traducido como aplicación web de una sola página, la interacción del usuario con dicha página web es en una sola página, por lo tanto, la carga de HTML, CSS y Javascript se hace de una sola vez. Estas aplicaciones pueden ejecutarse en cualquier dispositivo que tenga navegador web y que cuente con acceso a Internet.

Este tipo de solución no necesita realizar peticiones al servidor constantemente para poder renderizar su contenido HTML, es decir, es un tipo de sitio web que se recarga de forma dinámica a partir de las interacciones que realiza el usuario con su contenido.

Una SPA tiene tres elementos fundamentales:

- **Location API:** es una API que sirve para manipular información en base al contenido de la URL de la página web. Se utiliza la propiedad `window.location`, que permite interactuar con las diferentes partes de dicha URL (hostname, href, pathname, origin, protocol).
- **History API:** es una API que contiene una serie de métodos utilizados para interactuar con la barra de direcciones y el historial de navegación. Sirve para manejar diferentes movimientos de navegación.
- **Web Storage API:** es una API que provee diferentes funcionalidades para almacenar información dentro del navegador a través de un formato clave-valor. Existen dos mecanismos fundamentales: Local Storage y Session Storage. Ambos son propiedades del estándar HTML5, que permiten almacenar datos en nuestro navegador de forma

similar a como lo hacen las cookies, con la diferencia que nos permite almacenar mucha más cantidad de información y de diferente naturaleza (multimedia, texto, etc).

2.1.4 La nube

Es una manera para poder almacenar y acceder a diferentes datos y recursos alojados a través de Internet. Con el uso de la nube, se tiene que pensar en una forma totalmente diferente de almacenar datos, por ejemplo: con el uso de un disco duro. Hablamos de nube o computación en la nube cuando accedemos a datos o programas a través de Internet, de tal forma que se los pueda tener sincronizados a través de la Web.

Un ejemplo muy claro es el de Google Drive, que permite que los usuarios guarden su información sin la necesidad de correr el riesgo de perderla en el caso de que se rompa o estropee su dispositivo, la información seguirá intacta debido a que se encuentra almacenada en una de las miles de computadoras que tiene Google.

2.1.5 Patrones de diseño web

Son buenas prácticas utilizadas en el mundo del desarrollo web, que permiten aumentar considerablemente la experiencia del usuario al momento de utilizar una aplicación y, por otro lado, reducir o minimizar la probabilidad de que el usuario cometa errores. Por ejemplo: la disposición de los controles utilizados, la cantidad de elementos mostrados, los colores de la aplicación, la transición en las pantallas, la carga de los elementos, etc.

2.1.6 Diseño web adaptable

Mejor conocido como *diseño web adaptativo o responsivo*, es una técnica de diseño y maquetación web utilizado en el mundo del desarrollo web, que permite mostrar correctamente la aplicación sea cual sea el dispositivo utilizado para visualizarla. Esto quiere decir que, sin importar el dispositivo que se utilice, la aplicación se tiene que adaptar a este, mostrando el mismo contenido.

El objetivo de esta técnica es conseguir que todo el contenido de una página web se adapte al tamaño de una pantalla cualquiera, por ejemplo: un celular, una Tablet, etc. Y, además, que el usuario pueda utilizarla de forma cómoda, garantizando una experiencia satisfactoria.



FIGURA 1: Ejemplo de diseño web adaptable. Fuente: <https://www.pixabay.com>

2.1.7 Software como Servicio (SAAS)

El software como servicio es una manera de ofrecer y distribuir software a los usuarios de tal forma que todos los datos se encuentren alojados en servidores de las compañías que

desarrollan las soluciones, es decir, el software y los datos manejados son alojados en un único servidor externo a la empresa. Esto permite que los usuarios no se vean en la necesidad de adquirir sus propios servidores, lo cual conlleva a un gasto mucho menor por parte de éstos.

Las empresas proveedoras de software se ocupan del alojamiento de la solución, y esto se hace utilizando el concepto de Computación en la Nube, que se mencionó anteriormente. La empresa contratante accede a las funcionalidades del software y todos los datos a través de un navegador web desde cualquier tipo de dispositivo.

La principal ventaja es que no hace falta instalar el software de forma local en cada uno de los dispositivos, con una conexión a Internet y un navegador web es suficiente para utilizar este tipo de solución.

El uso de Software como Servicio tiene varias ventajas y desventajas:

SAAS	
Ventajas	Desventajas
Menos tiempos de implementación.	Desconfianza por parte de las empresas que adquieren estos servicios, debido a que los datos dejan de estar en manos de la misma y pasan a ser propios de la empresa proveedora del software.
Menos costos de adquisición.	Depender mucho de la empresa proveedora de software.
Mucha escalabilidad e integración con los procesos de negocio propios de cada empresa.	Requiere conexión a internet en ambas empresas para poder utilizarlo.
Software actualizado.	Los miembros de la empresa no pueden realizar modificaciones al software una vez implementado.
Pueden ser utilizados en cualquier dispositivo con acceso a Internet, independientemente de su Sistema Operativo. No hace falta realizar instalaciones locales.	

TABLA 1: Ventajas y desventajas de SAAS.

2.1.8 E-commerce

Mejor conocido como comercio electrónico o contratación electrónica es una forma para realizar diferentes transacciones relacionadas con la compra-venta de productos y servicios por medio de dispositivos que tengan acceso a Internet.

2.1.9 Pago electrónico

Es una forma de realizar los pagos de dinero a través de Internet, sin la necesidad de que el comprador tenga que realizarlo de forma presencial con el vendedor del producto o servicio.

Un aspecto fundamental de esta forma de pago es la seguridad que debe tener. Uno de los conceptos a tener en cuenta es el de *Estándar de Seguridad de Datos de la Industria de Tarjetas de Pago*, que es un estándar de seguridad publicado por el PCI SSC, y orientado a la protección de datos del titular de la tarjeta y/o datos confidenciales de autenticación durante su procesamiento, almacenamiento y transmisión. Cuenta con más de 250 controles de seguridad física, lógica y administrativa. Antes que se publique la primera versión de este estándar, cada una de las marcas de tarjetas de pago contaban con un programa propio de seguridad, lo cual acarreó una serie de problemas como: incongruencias, duplicidades y solapamientos, los cuáles PCI DSS vino a unificar:

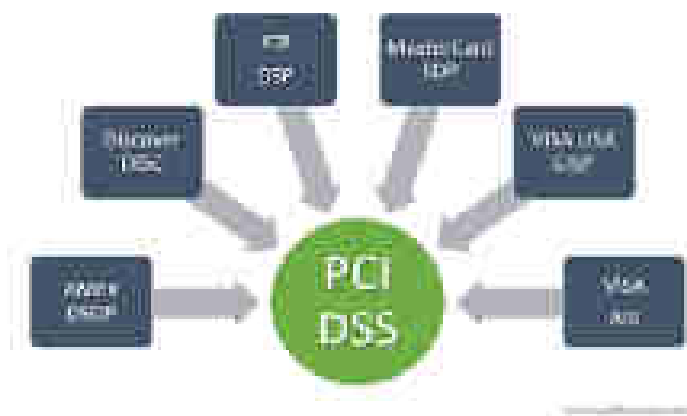


FIGURA 2: Convergencia de los programas de seguridad de cada una de las marcas en el estándar PCI DSS.

Fuente: pcihispano.com

El estándar PCI DSS está orientado a la protección de los datos del titular de la tarjeta y/o datos confidenciales de autenticación. Dentro de la primera clasificación tenemos: número primario de cuenta, nombre del titular, fecha de expiración y código de servicio, y en la segunda clasificación tenemos: código de seguridad y datos completos de la banda magnética.

Posee más de 250 controles de seguridad física, lógica y administrativa, divididos en 6 grupos principales que se dividen en 12 requerimientos:

- 1. Desarrollar y mantener redes y sistemas seguros**
 - 1.1.** Instalación y configuración de firewall para proteger los datos del titular de la tarjeta.
 - 1.2.** No usar valores predeterminados para las contraseñas y otros parámetros de seguridad.
- 2. Proteger los datos del titular de la tarjeta**
 - 2.1.** Proteger los datos del titular de la tarjeta que fueron almacenados.
 - 2.2.** Cifrar la transmisión de los datos del titular de la tarjeta en las redes públicas.
- 3. Mantener un programa de administración de vulnerabilidad**
 - 3.1.** Proteger sistemas contra malware y actualizar antivirus.
 - 3.2.** Desarrollar y mantener sistemas seguros.

4. Implementar medidas sólidas de control de acceso

- 4.1. Restringir el acceso a los datos del titular de la tarjeta según la necesidad de saber que tenga la empresa.
- 4.2. Identificar y autenticar el acceso a los componentes del sistema.
- 4.3. Restringir acceso físico a los datos del titular de la tarjeta.

5. Supervisar y evaluar las redes

- 5.1. Rastrear y supervisar accesos a los recursos de red y a los datos del titular de la tarjeta.
- 5.2. Probar periódicamente los sistemas y procesos de seguridad.

6. Mantener una política de seguridad de la información

- 6.1. Mantener una política que aborde la seguridad de la información para todo el personal.

Este estándar se aplica hacia todas las entidades que manejan procesos de almacenamiento, procesamiento y transmisión de datos del titular de la tarjeta y/o datos confidenciales de autenticación de tarjetas de pago. Entre los cuales tenemos: comerciantes, entidades emisoras, proveedores de servicios de pagos, proveedores de desarrollo de software, proveedores de infraestructura tecnológica, etc.

El cumplimiento de este estándar es obligatorio, aunque dependiendo del tipo de entidad habrá más o menos requerimientos y tipos de evaluación. Su incumplimiento implica:

- Limitaciones por parte de las marcas de tarjetas de pago.
- Asumir costes de demandas e indemnizaciones, costes de fraudes de transacciones, costes de renovación de tarjetas vulneradas, multas por dichas marcas de tarjetas, multas legales por afectación de datos personales, costes de investigación forense, costes de controles post incidentes y costes de pérdida de imagen de cara al público.

2.1.10 Bases de datos en tiempo real

Una *base de datos en tiempo real* es un tipo de base de datos que procesa la información en tiempo real, a medida que van ocurriendo los sucesos, de tal forma que se puedan administrar procesos que cambian frecuentemente. Estas bases de datos se utilizan en todas aquellas aplicaciones que necesitan actualizaciones instantáneas.

Son utilizadas ampliamente por varias empresas relacionadas con: derecho, contabilidad, registros médicos, bancos, sistemas de reserva, etc.

2.1.11 Bases de datos en la nube

Una *base de datos en la nube* es un tipo de base de datos que no se encuentra alojada en el dispositivo del usuario, sino que se encuentra alojada en Internet, de tal forma que ofrece todas las ventajas de la computación en la nube. Se utiliza como un servicio, es decir, el usuario paga una cuota que usualmente es mensual para poder utilizarlo y poder alojar su información en el mismo.

Este tipo de base de datos le ahorra el trabajo al usuario de tener que comprar hardware dedicado para almacenar su información, y puede utilizarla u ofrecer dicha información a otros. Tiene varias ventajas, entre las cuales podemos mencionar:

- **Alta disponibilidad:** debido a que los proveedores de estos servicios se encargan de mantener sus servidores disponibles las 24 horas del día, de tal forma que no se comprometa la experiencia del usuario. Utilizan conceptos como replicación y bases de datos distribuidas, para asegurarle al usuario que pueda realizar sus tareas sin riesgo de perder información.
- **Menor tiempo de salida al mercado:** lo cual, puede ser bastante útil cuando se necesita una solución que se tiene que implementar rápido.
- **Menos riesgos asociados:** debido a que este tipo de soluciones contienen una gran variedad de reglas de seguridad y automatización que reducen los riesgos asociados.
- **Costos más bajos:** ya que la gran mayoría cobran según la política de uso por servicio. En el cual se les cobra a los usuarios teniendo en cuenta la cantidad de transacciones que realicen diariamente, la cantidad de información generada, etc. Y para todo esto ofrecen ciertos planes que permiten mayor o menor tráfico de información. Esto es muy conveniente ya que los servidores no son muy económicos teniendo en cuenta que se tienen que estar actualizando cada cierto tiempo.

Este concepto va relacionado con el de **informática sin servidor o arquitectura serverless**, en el cual les permite a los usuarios en general poder administrar información sin necesidad de gestionar servidores de datos.

2.1.12 Bases de datos no relacionales

Las *bases de datos no relacionales*, mejor conocidas como NoSQL, son un tipo de base de datos que no sigue los estándares del Modelo Relacional ni el modelo Entidad - Relación, utilizado por las bases de datos SQL. Es decir, son bases de datos que no utilizan el lenguaje SQL para manipular la información, y por lo consiguiente, no utilizan estructuras de datos en formas de tabla, sino que utilizan otras estructuras como: clave-valor, grafos, objetos, etc.

El uso de este tipo de bases de datos conlleva ciertas ventajas:

- Las pueden utilizar dispositivos con pocos recursos.
- Tienen escalabilidad horizontal, mediante el agregado de nodos.
- Pueden manejar una gran cantidad de datos, utilizando una arquitectura distribuida de información.
- Las consultas están optimizadas para grandes volúmenes de datos.
- Se pueden hacer cambios en las bases de datos sin necesidad de pararlas.
- Son mucho más flexibles.

Asimismo, ciertas desventajas:

- No todas las bases de datos soportan propiedades ACID (atomicidad, consistencia, aislamiento y durabilidad).
- No existe un estándar como el Modelo Entidad Relación, ya que existe una gran variedad de bases de datos NoSQL diferentes, cada una hecha de forma diferente y siguiendo distintas metodologías.

2.1.13 Geolocalización

También llamada *georreferenciación* o *geoposicionamiento*, es una tecnología que utiliza la información generada en una computadora o dispositivo móvil y que permite obtener la ubicación física real de la persona o individuo, teniendo en cuenta su latitud y longitud. Dicha información siempre tiene un margen de error que varía según diferentes factores.

Esta tecnología es muy utilizada en aplicaciones modernas que requieren este tipo de funcionalidades. La información se obtiene en tiempo real, y se puede usar para una gran variedad de cosas: desde prevención de fraudes hasta administración de incidentes en tiempo real.

Los datos de geolocalización se pueden generar y recopilar de dos formas:

- **Modo activo:** que está basado en el dispositivo de cada uno de los usuarios, en este modo se utiliza directamente el software del mismo, y la ubicación se determina por medio de GPS, WIFI, 3G/4G.
- **Modo pasivo:** que realiza una búsqueda de datos que se basa en la dirección IP de cada dispositivo, que es independiente de la ubicación del mismo, y también en los identificadores de conjunto de servicios (SSID) para redes inalámbricas. Se realiza un proceso para relacionar la información de ambas bases de datos asociadas, sacándola de registros de compras, análisis de redes y nombres de dominio.

2.1.14 Seguridad en la capa de transporte (SSL – TSL - HTTPS)

SSL es un acrónimo de *Secure Sockets Layer*, es una tecnología estándar de seguridad digital que permite la comunicación cifrada entre un sitio web y un navegador web, para proteger cualquier información confidencial que se envía entre dos sistemas e impedir el acceso indebido a estos datos personales y cualquier tipo de modificación. La forma de hacerlo es cifrando dichos datos mediante algoritmos de cifrado, de tal forma que no se permita su lectura externa.

Un certificado SSL se instala en el servidor, pero se puede observar en el navegador la presencia o ausencia del mismo. Para empezar, si en el sitio está presente SSL, los usuarios verán `https://` al principio de la dirección web en lugar de `http://` (la "s" extra significa "seguro").

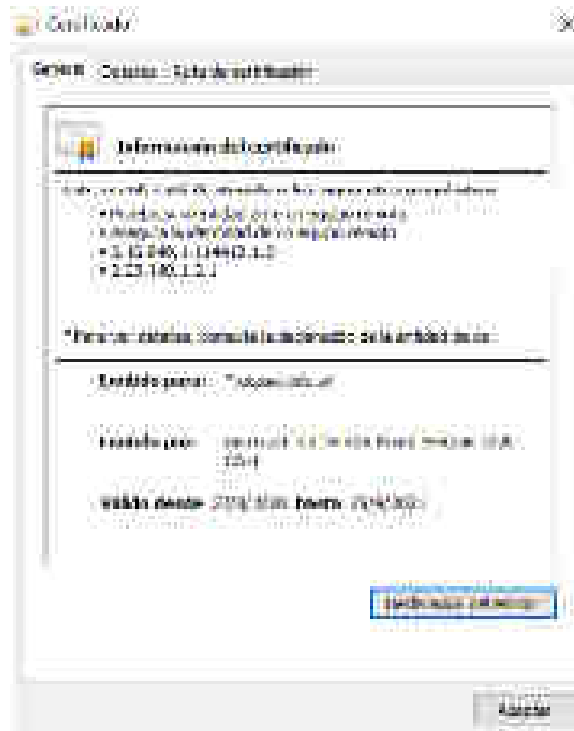


FIGURA 3: Ejemplo de Certificado SSL del Sistema de Autogestión de UCASAL. Fuente: <https://sistemas.ucasal.edu.ar/SAG/>

La creación de dicho certificado tiene un procedimiento asociado:

1. La persona o entidad solicitante genera una *clave pública* y *clave privada* que son necesarias para la otorgación del certificado. La clave pública será visible para cualquier persona que acceda a la información del certificado, la clave privada no ya que solamente la persona podrá acceder a su contenido. Estas claves pueden generarse mediante diferentes métodos: software, consola de windows o linux, etc.
2. Una vez generada el par de claves, se tiene que enviar esta información a una *autoridad de registración*, junto a datos de la persona o entidad. Esta autoridad será la encargada de autenticar esta información de acuerdo a diferentes políticas y estándares.
3. La *autoridad de registración* envía la información validada a una *autoridad de certificación*, que será la encargada de emitir o no dicho certificado. Es un documento electrónico, con un formato estándar y contiene atributos como: nombre de la persona, correo electrónico, copia de la clave pública, período de validez y nombre de la autoridad de certificación.

En el caso del sistema de autogestión de UCASAL los datos son:

- **Nombre de la persona o entidad:** *.ucasal.edu.ar (el asterisco hace referencia a que todas las URL con esa ruta tendrán el certificado SSL).

- **Emitido por:** GeoTrust TLS DV RSA Mixed SHA256 2020 CA-1
- **Válido desde:** 27/8/2020
- **Hasta:** 29/8/2021
- **Clave pública:** 30 82 01 0a 02 82 01 01 00 c6 6b c0 45 f8 53 02 9a cb 45 1b 16 35 33 36 96 3d ae 48 00 39 48 41 b3 7f 79 a1 fb 07 56 15 8d d6 21 7a 90 5b 7d f2 69 33 70 e6 dc fc a9 08 5e c9 02 ea 0c db 24 c2 5d 88 9d bc 8a f5 d8 41 2c 4e ad d0 90 6c 3c d8 93 8b 44 15 cc 4d a2 43 7b 61 a3 29 cd 16 cf bf ec 77 a4 e6 c2 69 92 db c0 39 dd 74 c6 e8 3f 6d 26 ad ad b7 f7 e4 89 d3 ed f3 32 ba 68 35 99 96 5c 5e f6 da 0a 82 be e1 bc ba 90 0a 76 5e 37 ba 86 50 13 da c3 fc c0 15 9d 31 28 35 a1 ec 37 39 0e bb 94 4b 20 48 95 bb 6b 30 59 fb 3a df 49 bd d0 d4 b1 c2 13 62 9e 77 8a 31 f1 ab 0e 52 77 f3 dd 7d aa 3c 42 fb 54 54 df 9f 0b 18 23 5b 22 41 ed 61 08 bb 24 c6 b7 5c 0e d8 b9 4a 29 e8 fa d2 e9 30 f5 7b 97 04 a6 64 2f 74 5c f1 a5 f7 af 80 0d d2 a5 3c 9d 2c 91 86 7c e7 52 51 75 db 92 20 e9 67 4a 7a f7 e9 e3 2d 53 02 03 01 00 01

El protocolo *TLS* (Transport Layer Security), o mejor conocido como seguridad de la capa de transporte, es una versión actualizada de *SSL*, que garantiza la privacidad de la misma forma que este. Se utiliza *TLS* ya que *SSL* está en desuso. Los más conocidos son *TLS2* y *TLS3*. Permite la encriptación entre dos puntos y todos los paquetes que se envían.

HTTPS es una extensión segura de *HTTP*, los sitios web que instalan y configuran un certificado *SSL/TSL* pueden usar el protocolo *HTTPS* para establecer una conexión segura con el servidor. No tener *HTTPS* tiene ciertas desventajas:

- No se puede saber si el sitio web es el verdadero.
- Los datos ingresados en el sitio no están encriptados.
- Se puede interceptar la comunicación, por ej: con el método de envenenamiento de máscara de red.
- Robo de datos (usuario, contraseña, n° de tarjeta), esto debido a que con *HTTP* los datos están en formato plano, no están encriptados.

2.2 Antecedentes relacionados con el problema

A continuación, se explicarán algunos aspectos vinculados a la problemática que se plantea en este proyecto, asimismo, se mostrarán ciertas noticias que los sustentan:

2.2.1 Multas y secuestro de vehículos

Uno de los principales problemas que se generan es cuando una persona no encuentra una playa de estacionamiento disponible al momento de querer guardar su vehículo de forma segura, en un tiempo determinado. Esto produce en la mayoría de los casos que la gente no estacione en los lugares correspondientes, lo cual conduce al desarrollo de diferentes multas y secuestros de los respectivos vehículos.

Este es un problema bastante grande, debido a que afecta no solo económicamente a



los conductores, sino también emocionalmente.

FIGURA 4: Encabezado de noticia. Fuente: <https://www.eltribuno.com/salta/nota/2019-5-10-0-0-0-el-50-de-las-multas-de-transito-se-labran-por-mal-estacionamiento>



FIGURA 5: Encabezado de noticia. Fuente: <https://pulsonoticias.com.ar/39250/los-autos-mal-estacionados-al-tope-de-las-estadisticas-de-infracciones/>

2.2.2 Infracciones por jurisdicción

El Sugit¹ (Sistema unificado de gestión de infracciones de tránsito) tiene en su página web una gráfica estadística que muestra la relación entre la cantidad de infracciones de tránsito cometidas en diferentes jurisdicciones del país.

¹ <http://www.sugitweb.com.ar/>

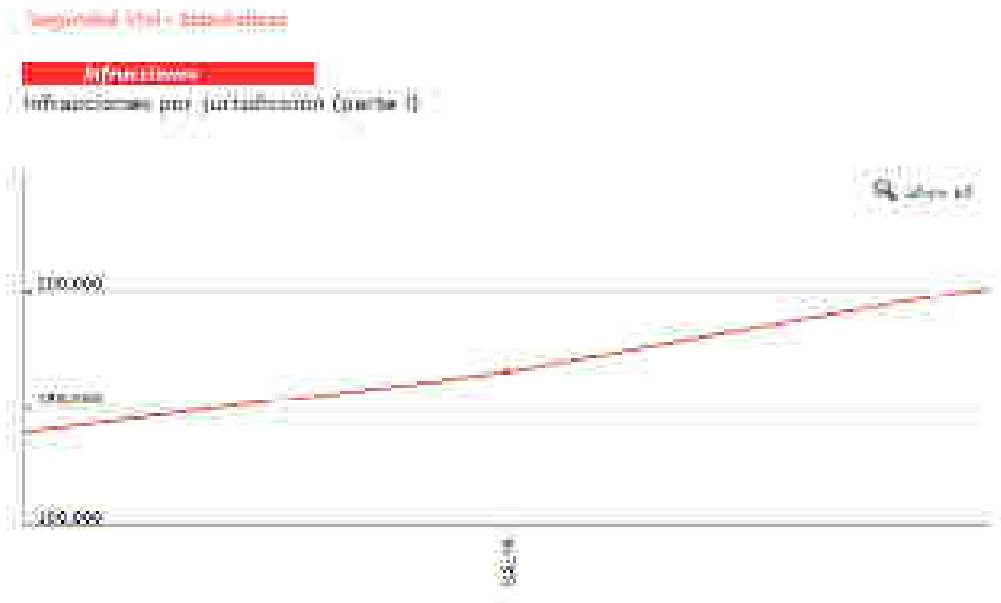


FIGURA 6: Gráfico estadístico de infracciones por jurisdicción. Fuente: http://www.sugitweb.com.ar/secciones/seguridad_vial/index.php?id=estadisticas

Se puede ver que, en Salta particularmente la curva tiene un carácter creciente, lo cual denota un aumento notable de infracciones. 2.2.3 Excesiva cantidad de vehículos – Mal funcionamiento de las playas de estacionamiento

En este caso, una mala administración de las playas de estacionamiento provoca una saturación en su capacidad para guardar vehículos, lo que provoca un mal funcionamiento.

Comenzó un reordenamiento de las playas de estacionamiento

18 de agosto de 2014



El objetivo es incidir para prevenir y combatir entre la mala conducta al estacionar en lugares indebidos y el acantonamiento de vehículos en espacios donde deberían circular unidades de tránsito frente a un accidente.

FIGURA 7: Encabezado de noticia. Fuente: <http://www.noticias.unsl.edu.ar/25/09/2019/comenzo-un-reordenamiento-de-las-playas-de-estacionamiento/>

Todos estos problemas se podrían solucionar con la implementación de una plataforma que administre eficientemente los procesos que se realizan en las playas de estacionamiento.

2.3 Soluciones similares

En este apartado se mencionan aquellas soluciones similares que fueron creadas por diferentes compañías para satisfacer problemáticas parecidas que están planteadas en este trabajo. Cabe aclarar que, algunas son aplicaciones móviles, por lo tanto, se las puede encontrar en las tiendas de *Android* y *Apple*, y también hay aplicaciones web.

2.3.1 Parkeo App²

Es una aplicación que conecta en tiempo real a los conductores con las playas de estacionamiento, permitiendo realizar las reservas y los pagos asociados. Esta solución está implementada para la provincia de Córdoba, Argentina. También se pudo ver que pretenden extenderse hacia Mendoza y Santa Fé, pero en Salta no se puede utilizar.

2.3.2 Westmart Park³

Es un servicio de estacionamiento colaborativo, originado en Barcelona en 2013. A diferencia del anterior, Westmart Park permite reservar estacionamientos en diferentes ciudades del mundo, entre las cuales se pueden mencionar: Buenos Aires, Barcelona, Madrid, Santiago de Chile, Lima, Lleida, Sao Paulo, Montevideo, Bogotá y Ciudad de México.

WeSmartPark funciona a través de una App gratuita alojada en Playstore y AppStore, de la misma forma que Parkeo App. Cabe mencionar que, en Salta, no se puede utilizar tampoco debido a que no hay estacionamientos que utilicen esta solución.

Esta aplicación es más compleja, tiene asociada muchas más funcionalidades que Parkeo App. Entre las cuales se destaca el de Movimientos, que hace referencia a la cantidad de dinero que se mueve en la cuenta del usuario registrado.

También, utilizan un concepto bastante novedoso, el de stickertag, que es una etiqueta identificadora de un automóvil que permite reconocer a los dueños de las playas de estacionamiento aquellos usuarios que utilizan esta aplicación de los que no. Dicha etiqueta la pueden enviar por medio de una carta a domicilio o el usuario puede comprarla en las tiendas asociadas. Esta etiqueta se necesita sí o sí para poder utilizar la aplicación y realizar las reservas,

2.3.4 Ingesys⁴

Ingesys es una empresa proveedora de sistemas para control de estacionamientos y condominios. Proveen diferentes tipos de soluciones, desde software hasta barreras de acceso. Una de las más importantes es el sistema de control de parking SICOA que es un conjunto de herramientas desarrolladas para cubrir todas las necesidades de gestión y control de estacionamiento de vehículos, registrando todas las entradas y salidas y brindando informes al momento.

² <https://parkeo.app/>

³ <https://www.wesmartpark.com/>

⁴ <https://www.ingesys.com.ar/>

2.4 Temas pendientes de resolución

En este apartado se mencionan todos aquellos que quedarán pendientes de resolución a lo largo del desarrollo del proyecto, es decir, todas aquellas cosas que se podrían implementar en un futuro, pero que en este trabajo no se considerarán debido a la complejidad asociada con las mismas.

2.4.1 Alcance a futuro

En un futuro, se planea aumentar el alcance del proyecto desarrollado, más que nada con permitir que la plataforma web pueda utilizarse en cualquier parte del mundo. Esto es posible debido a la naturaleza de la solución, ya que solamente se necesita un dispositivo con acceso a Internet y un navegador web para poder utilizarse, pero por otro lado, existen ciertas normativas asociadas con cada país que tendrían que respetarse antes de hacer esto.

2.4.2 Implementación de servicios adicionales

Algo que se tendría que implementar en un futuro sería que la plataforma permita al usuario acceder a diferentes servicios adicionales que una playa de estacionamiento pueda ofrecer. Existen diferentes playas que ofrecen estos servicios, con costos asociados en algunos casos. Algunos ejemplos son:

- Lavado completo de vehículo.
- Calibrado de gomas.
- Reparaciones.
- Cambio de aceite, líquido refrigerante, etc.

Actualmente el sistema permite a los encargados realizar una breve descripción de los servicios que ofrecen, pero, no está contemplada la posibilidad de añadir dichos servicios a la reserva realizada, impactando en el precio final.

2.4.3 Adaptar la solución a todo tipo de lugares

Otra cosa bastante interesante que se podría implementar con esta solución es la posibilidad de asociar los estacionamientos de diferentes entidades privadas en las cuales se tenga que pagar para obtener un lugar para estacionar, no solamente a playas de estacionamiento.

También podría aplicarse a cines, shoppings, aeropuertos, etc. Lo importante, es que esta solución se pueda adaptar a cada una de estas entidades sin problema.

2.4.4 Aplicación móvil

Más adelante, en futuras líneas de implementación, se tiene pensado realizar una nueva versión del sistema que se pueda utilizar mediante una aplicación móvil, para cualquier tipo de Sistema Operativo como Android ó IOS. Más que nada porque, hoy en día, la mayoría de las personas tienen más facilidad para utilizar este tipo de dispositivos para realizar sus tareas diarias.

2.4.5 Implementación de código QR con lectores

Una idea a futuro podría ser el utilizar el *código QR* que se genera con cada reserva para poder validar el ingreso de los clientes a los estacionamientos, utilizando un lector de código QR que automáticamente se asocia con la información generada por el sistema. Se

podría vincular esta información con algún sistema de barreras para manejar mejor la entrada y salida de vehículos desde el estacionamiento.

2.4.6 Reconocimiento de patentes

Se podría implementar un módulo de reconocimiento de patentes, utilizando técnicas de *aprendizaje automático*, específicamente de *visión artificial*, para que cada vez que ingrese un vehículo al establecimiento, se reconozca automáticamente la patente del mismo, y que esta información impacte directamente en el sistema. De esta manera no haría falta realizar ningún tipo de reserva, sería otra forma para administrar el ingreso y la salida de los vehículos de las playas de estacionamiento.

Podría implementarse como un módulo nuevo dentro del sistema, independiente del módulo de reservas, Además, se tendrían que implementar sensores o cámaras que detectan el movimiento de los vehículos para poder reconocer la patente.

2.4.7 Gestión de convenios y descuentos

El sistema actualmente no contempla la posibilidad de gestionar convenios con otras empresas o entidades, por ej: hoteles, librerías, etc. En un futuro se piensa poder administrar esto para poder asignar el descuento correspondiente al precio final a los clientes que estén afiliados a dichas entidades.

2.4.8 Suscripciones por planes

El sistema está pensado para que pueda utilizarse por cualquier usuario, siempre y cuando pague una licencia cuyo monto se detalla en el análisis de factibilidad. A futuro se podría implementar una serie de suscripciones con diferentes precios, cada una otorgando ciertos beneficios. Se podría ofrecer un plan gratuito, un plan premium y un plan vip.

Capítulo III: Definición del problema

3.1 Objetivo general

Diseñar y desarrollar un sistema informático que permita, por un lado, la gestión y administración de diferentes playas de estacionamiento ubicadas en el país, y, por otro lado, la reserva, gestión y pago de lugares en dichas playas mencionadas.

3.2 Objetivos específicos

- Realizar una tarea de relevamiento de datos mediante entrevistas y encuestas en diferentes playas de estacionamiento en Salta, Capital.
- Desarrollar la interfaz gráfica que el usuario verá al momento de utilizar la página web.
- Diseñar y crear modelos de datos que reflejen la realidad de los procesos propios de estos negocios.
- Vincular alguna plataforma de pago online cómo mercadopago al sistema.
- Realizar el despliegue del sistema en Internet, de tal forma que cualquier usuario que esté autorizado pueda acceder a los servicios definidos en el mismo.

3.3 Alcance del proyecto

Para definir el alcance del proyecto, se basará en lo propuesto en la guía del PMBoK® Quinta Edición. En el capítulo IV se explicará con detalle esta metodología.

3.3.1 Descripción del alcance del proyecto

El sistema informático a desarrollar consta de tres roles bien diferenciados:

- De la persona que realizará la reserva y el pago, de ahora en más: *usuario común*.
- Del *encargado* de la playa de estacionamiento.
- Del *administrador* del sistema.

Por lo que, siguiendo esta idea, se puede decir que el sistema en su totalidad está formado por los siguientes módulos:

- **Módulo de reservas:** en el cual un usuario común podrá registrar, modificar y cancelar una reserva de un lugar en una playa de estacionamiento de su preferencia. Además, se podrá realizar el seguimiento de cada una de las mismas, asimismo, los encargados de las playas de estacionamiento podrán observar y modificar ciertos datos, no todos, de las reservas que se realizaron en el establecimiento.
- **Módulo de pagos:** este módulo permitirá, a los usuarios comunes, realizar el pago del mismo a través de alguna de las siguientes formas de pago provistas por la plataforma de mercadopago:
 - **Tarjeta de crédito.**
 - **Tarjeta de débito.**
 - **Rapipago, pago fácil, etc.**

Y generar el comprobante de pago correspondiente.

- **Módulo de usuarios y autenticación:** este módulo permitirá, por un lado, para el usuario administrador, gestionar a todos los usuarios en el sistema, pudiendo dar de alta, modificar sus datos, asignarles permisos o eliminarlos. Por otro lado, permitirá a los usuarios comunes y encargados acceder al sistema pudiendo autenticarse con correo electrónico y contraseña. Cabe aclarar que, a los usuarios comunes el sistema permite que se den de alta al mismo, pero no a los encargados, por cuestiones de seguridad.
- **Módulo de cobros:** este módulo permitirá, a los usuarios encargados de las playas de estacionamiento, realizar los cobros a los usuarios en caso de que realicen sus pagos de forma presencial. Asimismo, permitirá crear los comprobantes de cobro correspondientes.
- **Módulo de playas de estacionamiento:** este módulo permitirá, por un lado, a los encargados de las playas de estacionamiento gestionar toda la información de sus playas de estacionamiento. Por ej: datos jurídicos, ubicación, tarifas, horarios, lugares, etc. Por otro lado, permitirá a un usuario común ver todos los estacionamientos registrados en el sistema, de tal forma que los pueda filtrar según diferentes criterios (más valorados, más cercanos, más comentados, horario corrido etc), y poder así continuar con el proceso de reserva.
- **Módulo de comentarios y valoraciones:** este módulo permite a los usuarios comunes registrar comentarios y valoraciones a las playas de estacionamientos, para que luego la información generada ayude a los encargados a la toma de decisiones.
- **Módulo de mensualidades:** este módulo permite a los usuarios comunes reservar un lugar en una playa de estacionamiento en una determinada franja horaria y días, y pagar una mensualidad para poder acceder a ella.
- **Módulo de reportes:** dependiendo del usuario que acceda, este módulo proporcionará información, a través de gráficos y estadísticas, respecto a las reservas realizadas, a los pagos registrados por el usuario, a los cobros realizados por los encargados de las playas de estacionamiento. Además, debe permitir que dicha información se pueda discriminar por diferentes parámetros, por ejemplo: por unidad de tiempo, por fechas, por medios de pago.

Asimismo, se detallan los siguientes aspectos que estarán asociados con el desarrollo del proyecto:

- **Capacitación:** se realizará una capacitación a los encargados de las playas de estacionamiento que estén interesados en obtener el servicio. Esta capacitación se realizará por vía zoom una vez implementado el sistema. Por otro lado, la capacitación a los usuarios que realicen las reservas en las playas de estacionamiento será vía web, es decir, en el mismo sistema web, los usuarios podrán acceder a tutoriales y manuales de usuario que les servirán como guía al momento de querer utilizar los servicios que provee el sistema.
- **Mantenimiento de los sistemas en las playas de estacionamiento:** en caso de que se requiera, se aplicará un mantenimiento correctivo del sistema informático en cada una de las playas de estacionamiento. Dicho mantenimiento estará destinado a corregir posibles fallos que puedan surgir con el uso de este y tendrá una duración de 6 meses.

- **Atención al cliente:** ante posibles dudas e inconvenientes que puedan llegar a surgir con el uso del sistema, se brindará un servicio de atención al usuario en horario laboral, es decir, de 9:00 am a 17:00 pm, en el cual se proveerá toda la información que el usuario necesite acerca de las funcionalidades desarrolladas.
- **Entrega de documentación:** una vez finalizado el proyecto, se realizará la entrega de la documentación relacionada con el uso del sistema desarrollado. La misma contendrá toda la información que necesitarán los usuarios para poder utilizarlo y tutoriales guiados con videos sobre el funcionamiento del sistema.

3.3.2 Exclusiones del proyecto

Según la guía del **PMBOK®**, se definen exclusiones como: *todo aquello que esté excluido fuera del alcance del proyecto*. Entonces se pueden mencionar los siguientes aspectos que no se incluirán dentro del desarrollo del proyecto:

- **Carga de datos en los sistemas informáticos:** una vez desarrollado el sistema, no se realizará la carga de los datos en cada una de las playas de estacionamiento en donde estén aplicados, eso corre por cuenta de los encargados de estas. Solamente se proporcionará el usuario y contraseña a cada uno de los encargados vía mail.
- **Migración de sistemas antiguos:** no se realizará ningún proceso de migración de sistemas antiguos en el caso de que los encargados de las playas de estacionamiento posean alguno.
- **Compra de dispositivos tecnológicos y conectividad para las playas de estacionamiento:** no se realizará la compra de hardware ni de conectividad para cada una de las playas de estacionamientos donde esté aplicado el sistema desarrollado.
- **Reparación, reposición de dispositivos en las playas de estacionamiento:** no se realizará ni la reparación de los dispositivos utilizados ni la reposición de los mismos. Estos corren por cuenta del encargado.
- **La implementación inicial del proyecto no estará destinada a aplicarse fuera del país:** más adelante, en futuras líneas de implementación podría aplicarse en otros territorios.

3.3.3 Restricciones o límites del trabajo

Según la guía del **PMBOK®**, se definen restricciones como: *factores limitantes que afectan a la ejecución de un proyecto o proceso*. Dicha guía propone las siguientes restricciones a tener en cuenta:

3.3.3.1 Restricciones de alcance

Si existen distintas restricciones de alcance debido a que, por ejemplo, si se quiere implementar esta solución fuera de la Argentina, se tienen que respetar normativas que rigen en cada uno de los países: transferencia transfronteriza de datos, protección de datos personales, normativas propias de las playas de estacionamiento, etc. Es por eso que, inicialmente, está pensado para aplicarse solamente en el territorio argentino.

3.3.3.2 Restricciones de tiempo o cronograma

No existen restricciones de tiempo para el proyecto, ya que está pensado para realizarse en un lapso de tiempo de 6 meses aproximadamente, en caso de ser necesario, puede extenderse más.

3.3.3.3 Restricciones legales

El desarrollo del proyecto tiene que respetar las normativas establecidas por el país en torno a las playas de estacionamiento. En el presente trabajo, particularmente en el capítulo VIII: Anexos se mostrará con detalle el contenido de cada una de ellas. Como referencia se utilizó el *Sistema Argentino de Información Jurídica (SAIJ* ⁵), la más grande base de datos jurídica en español.

- **Ley N°7464. Salta:** Facturación en guarderías de vehículos y playas de estacionamiento en la provincia de Salta.
- **Decreto N°454/2014. Ciudad de BS AS:** Playas de Estacionamiento y Garage. Ciclomotores y motocicletas. Se establece la obligación de fijar una tarifa.
- **Ley 7492. Mendoza:** Obligación para playas de estacionamiento de emitir tickets o comprobantes de ingreso de vehículos.
- **Ley 7772. San Juan:** Registro de playas de estacionamiento.
- **Ley N°5007. Ciudad de BS AS:** Publicación de leyenda en carteles o tickets en playas de estacionamiento vehicular.

El sistema desarrollado tiene que respetar cada una de estas normativas en caso de querer implementarse, para evitar que los encargados puedan tener problemas legales.

3.3.3.4 Restricciones tecnológicas o de recursos

La única restricción tecnológica para el funcionamiento del proyecto sería que los usuarios tengan dispositivos con conexión a internet para poder utilizarlo.

3.3.3.5 Restricciones de costo o de presupuesto

No existen restricciones de costo para el desarrollo del proyecto, en el capítulo IV se explica con detalle el análisis económico - financiero necesario para poder llevarlo adelante.

3.3.3.6 Restricciones de riesgos

Más adelante, en el capítulo IV se detalla de forma completa los posibles riesgos a los cuáles podría estar sometido el proyecto, por lo cual sí existen ciertas restricciones relacionadas con dichos riesgos que se tienen que manejar o disminuir en caso de una posible ocurrencia.

3.4 Alternativas tecnológicas

Para el desarrollo de este proyecto se tuvieron en cuenta diferentes alternativas tecnológicas, más adelante en el capítulo IV se explica con detalle la alternativa elegida en base a los conocimientos y la afinidad del autor del trabajo:

- **Tecnologías del lado del servidor:**
 - Firebase
 - Google Cloud
 - Node Js
 - Express Js

⁵ <http://www.saij.gob.ar/>

- .NET Framework
- **Tecnologías del lado del cliente**
 - Vainilla Js
 - Angular Js
 - React JS
- **Bases de datos**
 - SQL Server
 - PostgreSQL
 - MongoDB
 - Firestore
- **Plataformas de Pago**
 - Mercadopago
 - Stripe
 - Paypal
- **Herramientas de manejo de mapas**
 - Google Maps
 - Leaflet
 - React-maps
- **Editores de texto**
 - Sublime text
 - Visual studio code
 - Atom
- **Sistemas operativos**
 - Windows 10
 - GNU/Linux
- **Hosting web**
 - Netlify
 - Heroku
 - Firebase
- **Software de control de versiones**
 - Git
 - SVN
- **Software de gestión de proyectos**
 - Gitlab
 - Github
 - Bitbucket

Capítulo IV: Solución propuesta

4.1 Herramientas utilizadas

4.1.1 Windows 10⁶

Windows 10 es un sistema operativo creado por Microsoft en el año 2015. Se lo elige debido a la facilidad de uso que tiene y a la afinidad que tiene el autor de este trabajo con este tipo de herramienta.

4.1.2 React JS⁷

React JS es una biblioteca de Javascript de código abierto para crear interfaces de usuario, creado y mantenido por Facebook. Es declarativo, basado en componentes y facilita mucho la creación de aplicaciones de diferentes características.

Tiene la característica de vincular directamente Javascript con HTML5, a través de un estándar denominado JSX, de esta forma se facilita notablemente la construcción de los componentes a utilizar.

Se utiliza React JS debido a que el autor tiene conocimientos de desarrollo con esta herramienta y además por la gran comunidad que posee, ya que es una de las herramientas más utilizadas en todo el mundo, según una página web que evalúa varias soluciones, se posiciona a React dentro de los mejores puestos con un porcentaje de satisfacción de casi el 88% y con casi 18000 usuarios en todo el 2020, muy por encima de frameworks de JavaScript



como Vue.js o Angular.

FIGURA 8: Estadísticas sobre herramientas basadas en Javascript. Fuente: <https://2020.stateofjs.com/es-ES/technologies/>

⁶ <https://www.microsoft.com/es-ar/windows/windows-10-specifications>

⁷ <https://es.reactjs.org/>

4.1.3 Firebase⁸

Firestore es una plataforma web creada por Google, cuya función principal es facilitar la creación de aplicaciones, proveyendo diferentes funcionalidades, permitiendo un aumento en la productividad. Tiene una amplia gama de herramientas para utilizar: autenticación de usuarios, bases de datos en tiempo real, almacenamiento en la nube, hosting, etc.

Se utiliza *Firestore* debido a la naturaleza de sus soluciones, ya que provee servicios de bases de datos No SQL en tiempo real que se almacenan en la nube.

4.1.4 Google Maps⁹

Google maps es un servicio de mapas de Google que funciona a través de la web. Provee diferentes funcionalidades como imágenes de mapas desplazables, fotos satelitales, rutas y recorridos, etc.

Se utiliza *Google Maps* para mejorar la experiencia de usuario en el desarrollo del sistema, de tal forma que puedan encontrar los estacionamientos de forma mucho más fácil, y también ya que se integra fácilmente con Javascript.

4.1.5 Mercadopago¹⁰

Mercadopago es una plataforma que provee diferentes funcionalidades a los usuarios para realizar el cobro y los pagos asociados a la compra-venta de productos o servicios. Se pueden utilizar diferentes medios de pago como tarjeta de crédito, tarjeta de débito, efectivo, rapipago, pago fácil, código QR, etc.

Se utiliza *Mercadopago* debido a que es de las plataformas más utilizadas en latinoamérica para el procesamiento de pagos, debido a su alta seguridad para mantener la integridad de las transacciones bancarias realizadas.

4.1.6 Node JS¹¹

NodeJS es un entorno multiplataforma de código abierto, basado en Javascript y que permite utilizar dicho lenguaje para el lado del servidor. Permite crear aplicaciones, API, y muchas cosas más.

Se utiliza esta herramienta debido a la gran facilidad que tiene para integrarse con *ReactJS* y con *MercadoPago*.

4.1.7 Express JS¹²

ExpressJS es una infraestructura de aplicaciones web *NodeJS* mínima y flexible que proporciona un conjunto sólido de características para aplicaciones web y móviles.

Se utiliza *ExpressJS* debido a que facilita notablemente la realización de APIS, en este caso, se utilizará para procesar los pagos, vinculando con *Mercadopago*.

⁸ <https://firebase.google.com/?hl=es>

⁹ <https://developers.google.com/maps?hl=es>

¹⁰ <https://www.mercadopago.com.ar/developers/es/guides>

¹¹ <https://nodejs.org/es/>

¹² <https://expressjs.com/es/>

4.1.8 Visual Studio Code¹³

Visual Studio Code es un editor de código fuente, abierto y optimizado que permite trabajar con múltiples lenguajes de programación para la creación de diferentes aplicaciones (web, escritorio, móvil, etc). Creado por Microsoft en 2015, ofrece una amplia gama de funcionalidades en el mundo del desarrollo de software.

Se utiliza esta herramienta debido a la gran cantidad de herramientas que ofrece para el desarrollo de aplicaciones web.

4.1.9 Git¹⁴

Git es un software gratuito y distribuido de control de versiones, que permite manejar desde pequeños hasta grandes proyectos con gran rapidez y eficiencia.

Se utiliza Git debido a la facilidad de instalación y por su gran comunidad, y debido a que se integra perfectamente con las herramientas que se mencionan a continuación.

4.1.10 Netlify¹⁵

Netlify es una plataforma que sirve para realizar despliegues gratuitos en la nube de diferentes aplicaciones web estáticas, es decir, sin la lógica del servidor. Ofrece varias licencias de hosting para poder alojar diferentes sitios web.

Se utiliza Netlify debido a la facilidad que provee para realizar los despliegues, y para evitar incurrir en gastos de servidores.

4.1.11 Heroku¹⁶

Heroku es una plataforma que sirve para realizar despliegues gratuitos en la nube de la lógica del servidor de diferentes aplicaciones web. Se vincula con varios lenguajes de programación como NodeJS, Python, Java, Ruby, PHP, etc.

Se utiliza Heroku debido a que se vincula perfectamente con NodeJS y con Github.

4.1.12 Github¹⁷

Github es un sistema de control de proyectos y control de versiones en la nube, que utiliza Git. Permite administrar repositorios en donde se pueden alojar y administrar diferentes proyectos. Tiene una gran comunidad y permite que cualquier persona acceda al código fuente de un proyecto en particular.

Se utiliza Github debido a que se vincula perfectamente con Heroku y Netlify, permitiendo realizar despliegues mucho más fácilmente que con otras herramientas del mercado.

4.1.13 Material UI¹⁸

Material UI es un framework de interfaces de usuario para React, basado en los estándares de Material Design. Provee una serie de componentes estandarizados que se integran fácilmente con React y sus nuevas actualizaciones.

¹³ <https://code.visualstudio.com/>

¹⁴ <https://git-scm.com/>

¹⁵ <https://www.netlify.com/>

¹⁶ <https://www.heroku.com/>

¹⁷ <https://github.com/>

¹⁸ <https://material-ui.com/>

Se utiliza esta herramienta ya que tiene una gran facilidad para integrarse con React y provee muchos componentes funcionales.

4.2 Metodología de gestión de proyectos: Guía del PMBOK

La metodología que se utilizará para gestionar el proyecto que se desarrollará a lo largo de este trabajo será la Guía de los Fundamentos para la Dirección de Proyectos, de ahora en más, *Guía del PMBOK®*, que identifica una serie de conceptos y fundamentos clave para la dirección de proyectos, reconocidas por varios profesionales como buenas prácticas.

Se utilizará dicha guía para explicar con detalle aspectos como: factibilidad del proyecto, análisis de riesgos, exclusiones del proyecto, restricciones del proyecto, etc.

4.3 Metodologías de desarrollo: Iterativo Incremental y Prototipado

Para llevar adelante el desarrollo del presente proyecto se utilizarán dos metodologías: Iterativo - Incremental y Prototipado.

El *ciclo de vida Iterativo - Incremental* es una metodología de desarrollo de software que combina características tanto del modelo clásico de cascada como de desarrollo evolutivo. Proviene del concepto de desarrollo iterativo, en el cual, los requerimientos del sistema cambian y donde la especificación se desarrolla junto con el software.

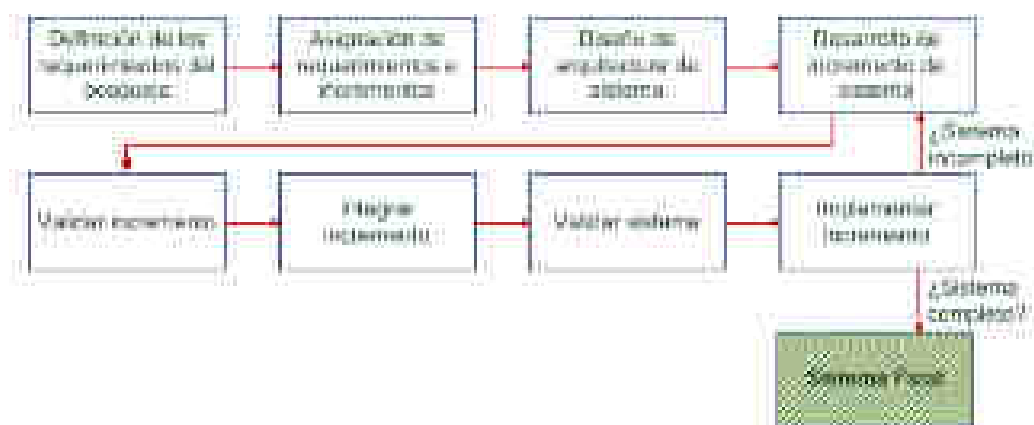


FIGURA 9: Proceso de ciclo de vida incremental. Fuente: Ingeniería del Software Ian Sommerville

En este tipo de desarrollo, lo primero que se hace es una definición de los requerimientos de los clientes, o sea, identificar las funcionalidades más importantes del sistema y las que no lo son tanto, es decir, se define un *esbozo de requerimientos* que tiene que cumplir el sistema a grandes rasgos. Después se definen varios *incrementos*, estos incrementos serán porciones de funcionalidad que forman parte de un sistema final, y que se le irán entregando al cliente de forma iterativa. La especificación, el diseño y la implementación del software se realizan por cada incremento, es decir, en cada uno, se aplica un ciclo de vida en cascada que después se va refinando.

Una vez definido el incremento con el cual se trabajará en esa iteración y las funcionalidades que tendrá, se delimita la arquitectura que se utilizará para llevar adelante el desarrollo del mismo. En este punto, se inicia el desarrollo del prototipo, para poder mostrárselo al cliente y que él lo valide en base a sus preferencias.

Al realizar la entrega y posterior instalación del producto perteneciente a un incremento, los clientes pueden empezar a utilizarlo sin ningún problema. Esto significa que

tienen una entrega temprana de la funcionalidad del sistema, por lo cual pueden probarlo y definir futuros requerimientos para los próximos incrementos. A medida que se van desarrollando, se van agregando en los incrementos existentes, de tal forma que el sistema va teniendo cada vez más y más funcionalidades, y de esta forma se va validando hasta tener un sistema final.

Esta metodología tiene varias ventajas:

- Los clientes no tienen que esperar hasta que el sistema esté completo para poder utilizarlo. El primer incremento contiene los requerimientos más críticos de tal forma que pueden utilizar el software inmediatamente.
- Los clientes pueden utilizar los incrementos iniciales como *prototipos* para probarlos y poder analizar requerimientos para futuros incrementos.
- Existe un muy bajo riesgo de fallo total del proyecto.
- Al realizarse al principio los incrementos de más alta prioridad, son a los que se hacen más pruebas, por lo cual, es muy poco probable que se encuentren fallos en el funcionamiento del software en estas partes.

Pero, además conlleva ciertas desventajas:

- Los incrementos deben ser relativamente pequeños (no más de 20.000 líneas de código).
- Dificultad de adaptar requerimientos del cliente a incrementos de este tamaño.
- Muchos sistemas requieren un conjunto de recursos que se utilizan en diferentes partes del sistema.
- Dificultad para identificar los recursos comunes que requieren todos los incrementos.

Como se dijo anteriormente, también se utilizará la metodología de *desarrollo de prototipos*. Un prototipo es un primer modelo que sirve como representación o modelo del producto final, que permite identificar el diseño y confirmar que cuenta con los requerimientos necesarios. En este caso particular, los prototipos serán los incrementos que se vayan mostrando al cliente, de tal forma que puedan servir como guía para futuros incrementos que se vayan implementando. Antes de realizar la validación del incremento, se tiene un prototipo perfectamente funcional que puede integrarse fácilmente a los ya existentes.

4.4 Aplicación de metodologías de desarrollo

4.4.1 Esbozo de requerimientos

Como se mencionó anteriormente, el esbozo de requerimientos contiene todas las funcionalidades del sistema a implementar priorizadas según algún criterio. En este caso, la prioridad se asignará implementando una *serie de Fibonacci*¹⁹, asignando una prioridad de 1 a 8, teniendo en cuenta que, a mayor nivel de priorización, más importante es a nivel de negocio

¹⁹ <https://quantdare.com/numeros-de-fibonacci/>

dicha funcionalidad y también requerirán mucho más grado de esfuerzo. Aquellas funcionalidades que tengan el número 2 requerirán el doble de esfuerzo que las que lleven el número 1, las que tengan el 3 el triple, y así sucesivamente.

El grado de esfuerzo necesario se puede determinar de muchas formas, pero usualmente es recomendable utilizar técnicas como *Planning Poker*²⁰, que es una técnica para realizar la estimación de requerimientos, que se basa principalmente en el *consenso*, es decir, el acuerdo mediante el consentimiento de las personas que forman parte del desarrollo en este caso. Básicamente lo que se hace es utilizar las funcionalidades priorizadas, y cada uno de los integrantes realizan estimaciones de esfuerzo en forma secreta, sin decir nada al resto del equipo, y a partir de ahí se hace una estimación grupal para llegar a un consenso entre todos. Esto se hace teniendo en cuenta las capacidades del equipo de trabajo, en este caso se la definirá con un máximo de 16 puntos, de tal forma que en cada iteración no se pueda sobrepasar este número.

Esta información servirá como guía para poder realizar la planificación de tareas correspondiente, con todos los recursos necesarios para llevarlas adelante.

Esbozo de Requerimientos		
N°	Nombre	Prioridad
1	Dar de alta una reserva	8
2	Generar pago de reserva	8
3	Listar reservas del día en tiempo real	8
4	Ver playas de estacionamiento	8
5	Ver mis reservas	5
6	Listar lugares de mi estacionamiento	5
7	Modificar datos del estacionamiento y tarifas	5
8	Administrar horarios y días de apertura	5
9	Listar usuarios	5
10	Modificar datos de usuario	5
11	Dar de alta usuario	5
12	Dar de baja usuario	5
13	Modificar datos de playa de estacionamiento	5
14	Dar de alta playa de estacionamiento	5
15	Dar de baja playa de estacionamiento	5
16	Autenticar mi usuario	2
17	Cerrar sesión	2
18	Cambiar fecha de reserva	2
19	Recuperar contraseña y recibir email de recuperación	2
20	Modificar mis datos personales	2
21	Generar reportes	2
22	Cambiar contraseña	2

²⁰ <https://samuelcasanova.com/2016/01/estimacion-agil-con-la-tecnica-planning-poker/>

23	Buscar reservas	2
24	Registrar nuevo comentario y valoración	2
25	Votar un comentario	1
26	Dar de alta mi vehículo	1
27	Modificar datos de mi vehículo	1
28	Dar de baja mi vehículo	1
29	Mensualizar lugar de estacionamiento	1
30	Listar mis mensualidades	1
31	Dar de baja mensualidad	1
32	Listar usuarios mensualizados	1
33	Listar reservas de un día en particular	1
34	Abrir estacionamiento	1

TABLA 2: Esbozo de requerimientos.

4.4.2 Incrementos

Se puede ver en el esbozo de requerimientos que hay un total de 34 funcionalidades con diferentes niveles de priorización, teniendo en cuenta una duración de 3 semanas por incremento y una capacidad de desarrollo de 16 puntos, se podría realizar un total aproximado de 3 funcionalidades por cada incremento, para dar un tiempo de desarrollo del sistema aproximado de 6 meses.

Cabe recalcar que, a medida que se vaya avanzando, el grado de dificultad irá disminuyendo, ya que las funcionalidades más complejas se realizarán al principio. De todas formas, esta es una estimación muy inexacta que se explicará con mucho más nivel de detalle en la parte de planificación del proyecto.

4.4.2 Historias de usuario

A cada uno de estos requerimientos hay que detallarlos para que se puedan identificar toda aquella información que sea relevante para el negocio, el ciclo de vida iterativo no proporciona ninguna herramienta para poder hacer esto, por lo cual, se utilizarán Historias de Usuario, que son una de las herramientas propuestas en las metodologías ágiles de desarrollo, particularmente *SCRUM*. Se sabe que *SCRUM* no es una metodología en sí, sino más bien un marco de trabajo que proporciona buenas prácticas para el desarrollo de software.

Según la bibliografía de Proyectos ágiles con Scrum de Martín Alaimo, una *historia de usuario* es una especificación funcional que invita a la conversación entre sus participantes. Es una solución al problema de realizar especificaciones muy detalladas de funcionalidades al momento de desarrollar software. Se compone de tres elementos:

- **Carta (ficha):** toda historia de usuario tiene que poder describirse en una ficha de papel pequeña, si nos pasamos de ese tamaño estamos comunicando demasiada información.
- **Conversación:** toda historia de usuario se tiene que poder conversar, a partir del cual se intercambia información, pensamientos, opiniones y sentimientos.
- **Confirmación:** toda historia de usuario debe estar lo suficientemente explicada para que el o los desarrolladores de software sepan que construir, conocido como *criterios de aceptación*.

Se redactan de la siguiente manera:

Cómo (rol) → Necesito (funcionalidad) → Para (beneficio) → Criterios de Aceptación

En el cual, el rol hace referencia al usuario que utilizará el sistema y que necesita que este le proporcione una determinada funcionalidad para alcanzar un beneficio, al mismo tiempo, tiene que cumplir ciertos requisitos, que como se dijo anteriormente, son los criterios de aceptación.

Ejemplo 1 - Comercializar Eventos				
Prior.	Cómo...	Necesito...	Para...	Criterios de Aceptación
1	Comercial	Crear un evento confirmado	Hacer el seguimiento del mismo	- Debe tener Nombre, Fecha, Descripción, Dominarios, Programa, Testimonios, Lugar, Ciudad, País, Capacidad, Precios y Promociones SII= \$ (Super Early Bird), FH (Early Bird), día en % para 2 personas a día, en % para 3 o más personas. - Las promociones son opcionales. - Las fechas de SII y FH deben ser anteriores a la fecha del evento. - Por defecto SII=30 días antes, FH=10 días antes, 2 personas=10%, 3+ personas = 15%. - Un evento puede ser público o privado.
2	Comercial	Ver listado de eventos confirmados	No superponer eventos	- Mostrar Nombre, Ciudad y País - Mostrar solo las fechas - Ordenado por fecha secundaria
3	Comercial	Modificar evento confirmado	Cambiar cualquier dato o se programado	- Permite modificar todos los campos.
4	Comercial	Cancelar evento confirmado	Dejar de seguirlo	- Desaparace del listado de eventos confirmados.
5	Comercial	Listar los eventos en un sitio web	Que los muestren	- Solo se listan los eventos públicos - Listado por fechas (a futuro)

Las historias de usuario tienen el siguiente formato:

FIGURA 10: Formato de Historia de Usuario. Fuente: Proyectos ágiles con scrum - Martin Alaimo

Teniendo en cuenta todo esto, a continuación, se realizará el listado de cada una de las historias de usuario que se identificaron en el proyecto:

N°1: Dar de alta una reserva				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
8	Conductor	Dar de alta una reserva	Poder estacionar mi vehículo en una playa de estacionamiento de mi preferencia	<ul style="list-style-type: none"> - Se tiene que validar que el usuario esté autenticado. - Una vez seleccionada una playa de estacionamiento, se tiene que solicitar la siguiente información al usuario: <ul style="list-style-type: none"> ● Datos de la persona: <ul style="list-style-type: none"> ○ Nombre completo ○ Correo electrónico ○ DNI ○ Teléfono ● Datos del vehículo <ul style="list-style-type: none"> ○ Tipo de vehículo ○ Marca del vehículo ○ Número de Patente ● Datos del estacionamiento: <ul style="list-style-type: none"> ○ Nombre completo ○ Dirección ○ Provincia ○ Teléfono ● Datos adicionales: <ul style="list-style-type: none"> ○ Fecha ○ Hora ○ Observaciones - Los datos de la persona y del estacionamiento deben cargarse por defecto. - Se tiene que mostrar una lista desplegable con los vehículos del usuario si es que tiene registrado. En este caso, los campos de tipo, marca y patente de vehículo tienen que estar deshabilitados y solo deben mostrar la información del vehículo. - En caso de que el usuario no tenga vehículos registrados, el tipo y marca del vehículo tienen que cargarse en dos listas desplegables para que el usuario pueda seleccionar, en la lista de tipo tiene que mostrarse el precio por hora de cada vehículo. - Se tiene que validar que todos los campos menos observaciones sean requeridos. - El texto del campo Patente tiene que estar enmascarado con el formato de patentes de Argentina. Por ej: AB123AA. - Se tiene que validar que la fecha y hora sean coherentes con los horarios de apertura de la playa de estacionamiento. - Al apretar el botón verde "Siguiente" se tiene que pasar a una pestaña en donde se muestren todos los datos ingresados. - Al apretar el botón blanco "Anterior" se tiene que volver a la pestaña de datos de Reserva. - Al apretar el botón verde "Pagar" se tiene que

				abrir una pantalla para realizar el pago de la reserva con MercadoPago.
--	--	--	--	---

TABLA 3: Historia de Usuario: Dar de alta una reserva.

N°2: Generar pago de reserva				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
8	Usuario común	Generar pago de reserva	Poder confirmar la reserva realizada	<ul style="list-style-type: none"> - Se tiene que validar que el usuario que realiza el pago no sea un encargado de playa de estacionamiento. - Se tiene que permitir ingresar como invitado o con cuenta de mercado pago. - Se tienen que permitir los siguientes medios de pago: <ul style="list-style-type: none"> ● Tarjetas <ul style="list-style-type: none"> ○ De Crédito ○ De Débito ● Efectivo <ul style="list-style-type: none"> ○ Rapipago ○ Pago fácil ○ Kioscos - En caso de pagar con tarjeta de crédito o débito se tienen que pedir los siguientes datos: <ul style="list-style-type: none"> ● Número de tarjeta ● Fecha de vencimiento ● CVC ● Nombre del titular - En caso de pagar con efectivo, se tienen que pedir los siguientes datos: <ul style="list-style-type: none"> ● DNI ● Email - Se tienen que validar que sean requeridos todos los datos anteriores. - Al apretar el botón “Pagar” se tiene que realizar el pago correspondiente. - Se tienen que mostrar notificaciones en caso de que el pago sea rechazado o aceptado.

TABLA 4: Historia de Usuario: Generar pago de una reserva.

N°3: Listar reservas del día en tiempo real

Prioridad	Cómo	Necesito	Para	Criterios de aceptación
8	Encargado de playa de estacionamiento	Ver reservas del día	Poder controlar el ingreso y salida de vehículos de mi establecimiento	<p>- Se tiene que mostrar la siguiente información:</p> <ul style="list-style-type: none"> ● Código ● Nombre de la persona ● Marca del vehículo ● Patente del vehículo ● Tipo de vehículo ● Observaciones ● Lugar ● Hora de Ingreso ● Hora de Salida <p>- Si la reserva no está validada, se tiene que mostrar un botón azul con el siguiente texto: "Validar". Al apretar este botón, se tiene que abrir un pop-up en el cual se solicite el horario de ingreso y el lugar que ocupará el vehículo. Se tiene que validar que los lugares no estén ocupados.</p> <p>- Si la reserva está validada, se tiene que mostrar un botón verde con el siguiente texto: "Concluir". Al apretar este botón, se tiene que abrir un pop-up en el cual se solicite el horario de salida del vehículo, además, en dicho pop-up se tiene que calcular automáticamente el monto adicional cada 15 minutos de demora que tendría que pagar el conductor en caso de retirar tarde el vehículo. Dicho monto no puede superar el 25% del monto final. Una vez apretado el botón, se tiene que liberar el lugar que estaba ocupado.</p> <p>- Los cambios tienen que impactar en tiempo real, de tal forma que, al registrarse una nueva reserva, se refleje automáticamente en este listado, sin la necesidad de recargar la página.</p>

TABLA 5: Historia de Usuario: Listar reservas del día en tiempo real.

Nº4: Ver playas de estacionamiento

Prioridad	Cómo	Necesito	Para	Criterios de aceptación
8	Usuario común	Ver playas de estacionamiento	Elegir una de mi preferencia y realizar la reserva	<p>- Se tiene que mostrar la siguiente información:</p> <ul style="list-style-type: none"> ● Nombre de la playa de estacionamiento ● Valoración General ● Ubicación: Provincia Dirección ● Teléfono ● Imagen descriptiva del estacionamiento <p>- Se tiene que permitir filtrar las playas de estacionamiento según los siguientes criterios:</p> <ul style="list-style-type: none"> ● Todos ● Horario Corrido ● Abiertos todos los días ● Más valorados ● Cercanos a mi ubicación ● Precios más bajos <p>- Se tiene que mostrar un campo de texto que permita buscar una playa de estacionamiento por su nombre.</p> <p>- Al hacer clic en una playa de estacionamiento, se tiene que abrir una nueva pantalla con los detalles del estacionamiento. Se tiene que mostrar la siguiente información:</p> <ul style="list-style-type: none"> ● Mapa con ubicación exacta ● Tarifas ● Horarios ● Comentarios <p>- Se tiene que mostrar un botón blanco con letras verdes con el texto “Reservar Aquí”, el cual al apretarlo me lleva a la pantalla de Dar de Alta una reserva.</p>

TABLA 6: Historia de Usuario: Ver playas de estacionamiento.

N°5: Ver mis reservas

Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Usuario común	Ver mis reservas	Poder realizar un seguimiento de mis movimientos y gastos	<p>- Se tiene que mostrar la siguiente información:</p> <ul style="list-style-type: none"> ● Nombre de la playa de estacionamiento ● Precio ● Día ● Hora <p>- Se tiene que mostrar una lista desplegable con todas las playas de estacionamiento, al lado un botón de color verde claro con el texto “Filtrar”. Al seleccionar una playa de estacionamiento y apretar dicho botón, se tienen que filtrar todas las reservas realizadas por el usuario a esa playa.</p> <p>- Se tiene que mostrar un botón de color azul con el texto “Ver Ticket”, al apretarlo se tiene que abrir un pop-up con el ticket asociado a la reserva anterior. Dicho ticket tiene que tener la siguiente leyenda exclusivamente: “Comprobante generado el: dd/mm/aaaa a las hh:mm:ss por: “nombre usuario. No válido como factura”. Este ticket tiene que poder guardarse en formato PDF y también imprimirse.</p>

TABLA 7: Historia de Usuario: Ver mis reservas.

N°6: Ver lugares de mi estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Encargado de playa de estacionamiento	Ver lugares de mi estacionamiento	Permitir a los usuarios comunes realizar reservas y mejorar su experiencia	<p>- Se tiene que mostrar un listado con todos los lugares registrados. En cada uno de los ítems de la lista se tiene que mostrar un botón:</p> <ul style="list-style-type: none"> ● Si el texto del botón es “Deshabilitar”, al apretarlo se deshabilita el lugar y no será accesible para realizar reservas. El color de dicho botón debe ser rojo con letras blancas. ● Si el texto del botón es “Habilitar” al apretarlo se habilita de nuevo. El color de dicho botón debe ser verde con letras blancas. ● Si el texto del botón es: ”Desocupar”, al apretarlo se desocupa el lugar. El color de

				<p>dicho botón debe ser azul con letras blancas.</p> <p>- Se tiene que mostrar un botón verde circular a la derecha de la pantalla, que al apretarlo automáticamente dé de alta un lugar y lo posicione al final del listado.</p>
--	--	--	--	---

TABLA 8: Historia de Usuario: Ver lugares de mi estacionamiento.

N°7: Modificar datos del estacionamiento y tarifas				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Encargado de playa de estacionamiento	Modificar datos del estacionamiento y tarifas	Permitir a los usuarios comunes realizar reservas y mejorar su experiencia	<p>-Se tienen que mostrar diferentes campos de texto con la siguiente información:</p> <ul style="list-style-type: none"> ● Datos del estacionamiento <ul style="list-style-type: none"> ○ Nombre Completo ○ CUIT ○ Número de teléfono ○ Descripción ○ Imagen ● Tarifas <ul style="list-style-type: none"> ○ Auto ○ Camioneta ○ Motocicleta ○ Traffic ● Ubicación <ul style="list-style-type: none"> ○ Dirección ○ Provincia ○ Ciudad <p>- Se tiene que validar que los campos sean requeridos excepto el de descripción.</p> <p>- Se tiene que mostrar un botón blanco con letras azules:</p> <ul style="list-style-type: none"> ● Si el texto del botón es “Mostrar Mapa” se tiene que mostrar el mapa con la ubicación exacta de la playa de estacionamiento. ● Si el texto del botón es “Ocultar Mapa” se tiene que ocultar el mapa.

TABLA 9: Historia de Usuario: Modificar datos del estacionamiento y tarifas.

N°8: Administrar horarios y días de apertura				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Encargado de playa de estacionamiento	Administrar horarios y días de apertura	Reflejar esta información con la realidad, y mantenerla actualizada	-Se tienen que mostrar siete checkbox que hacen referencia a los siete días de la semana. Al apretar uno, se tienen que mostrar automáticamente dos campos de texto en formato de fecha para el horario de apertura y de cierre. - Se tiene que validar que el horario de cierre de cierto día no sea menor al de apertura.

TABLA 10: Historia de Usuario: Administrar horarios y días de apertura.

N°9: Ver usuarios				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Ver usuarios	Administrar la información de cada uno de ellos	-Se tiene que mostrar un listado con todos los usuarios registrados con la siguiente información: <ul style="list-style-type: none"> ● Nombre completo ● Usuario ● DNI ● Email ● Teléfono

TABLA 11: Historia de Usuario: Ver usuarios.

N°10: Modificar datos de usuario				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Modificar datos de usuario	Administrar la información de cada uno de ellos	- Al apretar el botón azul con el texto: "Modificar datos" se tiene que abrir un pop-up con campos de texto que contengan la información del usuario. Se debe poder modificar todos los datos menos el e-mail.

TABLA 12: Historia de Usuario: Modificar datos de usuario.

N°11: Dar de alta usuario				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Dar de alta usuario	Administrar la información de cada uno de ellos	<p>- Al apretar el botón circular verde se tiene que abrir un pop-up con campos de texto vacíos:</p> <ul style="list-style-type: none"> ● Nombre completo ● Usuario ● DNI ● Email ● Teléfono ● Contraseña ● Repita contraseña <p>Y con dos checkbox:</p> <ul style="list-style-type: none"> ● Es admin ● Es encargado <p>- Se tiene que validar que todos los campos sean requeridos.</p> <p>- Las contraseñas deben coincidir.</p>

TABLA 13: Historia de Usuario: Dar de alta usuario.

N°12: Dar de baja usuario				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Dar de baja usuario	Mantener actualizada la información del sistema	<p>- Al apretar el botón rojo con el texto "Dar de baja" se tiene que abrir un pop-up con un campo de texto de validación y un botón deshabilitado para eliminar la información del usuario. En dicho campo de texto se tiene que ingresar el nombre de usuario, de tal forma que se active el botón para eliminar al usuario.</p>

TABLA 14: Historia de Usuario: Dar de baja usuario.

N°13: Modificar datos de playa de estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Modificar datos de playa de estacionamiento	Administrar la información de cada uno de ellos	<p>- Se tiene que mostrar un listado con todas las playas de estacionamiento registradas con la siguiente información:</p> <ul style="list-style-type: none"> ● Nombre completo ● Teléfono ● Dirección ● Provincia ● Ciudad <p>- Cada ítem tiene que tener dos botones: uno azul para modificar los datos del usuario y otro rojo para dar de baja.</p> <p>- Se tiene que mostrar un botón verde circular que al apretarlo abra un pop-up para dar de alta una playa de estacionamiento.</p>

TABLA 15: Historia de Usuario: Dar de alta usuario.

N°14: Dar de alta playa de estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Dar de alta playa de estacionamiento	Administrar la información de cada uno de ellas	<p>- Al apretar el botón circular verde se tiene que abrir un pop-up con campos de texto vacíos:</p> <ul style="list-style-type: none"> ● Nombre completo ● CUIT ● Teléfono ● Ubicación ● Cantidad de lugares ● N° de sucursal <p>- Se tiene que mostrar una lista desplegable para elegir el encargado del estacionamiento.</p> <p>- Se tiene que validar que todos los campos sean requeridos.</p> <p>- El campo de texto Ubicación tiene que ser un campo autocompletado para cargar las ubicaciones de Google Maps.</p> <p>- Se tiene que validar que la lista desplegable solo contenga usuarios con el rol encargado y que no tengan estacionamiento asignado.</p> <p>- Las contraseñas deben coincidir.</p>

TABLA 16: Historia de Usuario: Dar de alta playa de estacionamiento.

N°15: Dar de baja playa de estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
5	Administrador	Dar de baja playa de estacionamiento	Mantener actualizada la información del sistema	- Al apretar el botón rojo con el texto “Dar de baja” se tiene que abrir un pop-up con un campo de texto de validación y un botón deshabilitado para eliminar la información del estacionamiento.

TABLA 17: Historia de Usuario: Dar de baja playa de estacionamiento.

N°16: Autenticar mi usuario				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
2	Encargado de playa de estacionamiento, usuario común y Administrador	Autenticar mi usuario	Poder utilizar el sistema	-Se tiene que validar que el email y la contraseña ingresadas en la pantalla de login se encuentren registradas en el sistema. - En caso de que la contraseña sea incorrecta se tiene que mostrar una notificación con el texto: ‘Contraseña incorrecta.’ - En caso de que el email sea incorrecto se tiene que mostrar una notificación con el texto: ‘Ingrese un correo electrónico válido’ - En caso de que el email no se encuentre registrado en el sistema se tiene que mostrar una notificación con el texto: ‘El correo electrónico no se encuentra registrado’

TABLA 18: Historia de Usuario: Autenticar mi usuario.

N°17: Cerrar sesión				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
2	Encargado de playa de estacionamiento, usuario común y Administrador	Cerrar sesión	Poder poder proteger mi información	- Se tiene que validar que haya una sesión activa en el sistema. - Al apretar el botón “Cerrar sesión” se tiene que cerrar la sesión actual del usuario logueado.

TABLA 19: Historia de Usuario: Cerrar sesión.

N°18: Cambiar fecha de reserva				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
2	Encargado de playa de estacionamiento	Cambiar fecha de reserva	Poder resolver necesidades del usuario común	<ul style="list-style-type: none"> - Se tiene que mostrar un campo de texto para ingresar el código de la reserva, y un botón con el texto “Consultar” - Una vez ingresado el código de 20 dígitos y apretado el botón se tiene que mostrar la información de la reserva, esta información debe estar deshabilitada menos la fecha de la reserva.

TABLA 20: Historia de Usuario: Cambiar fecha de reserva.

N°19: Recuperar contraseña y recibir email de recuperación				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
2	Encargado de playa de estacionamiento, usuario común y Administrador	Recuperar contraseña y recibir email de recuperación	Poder utilizar el sistema en caso de olvidar mis credenciales	<ul style="list-style-type: none"> - Se tiene que mostrar un campo de texto para ingresar el correo electrónico de recuperación. - Se tiene que validar que el correo electrónico esté registrado en el sistema y que sea uno válido. - Se tiene que mostrar un botón verde con el texto “Recuperar contraseña” que al apretarlo automáticamente muestra una notificación con el texto “Se ha enviado un correo electrónico a la dirección ingresada. Por favor, siga los pasos para poder recuperar su contraseña”, y envía un email de recuperación con un enlace para cambiar la contraseña.

TABLA 21: Historia de Usuario: Recuperar contraseña y recibir email de confirmación.

N°20: Modificar mis datos personales				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento, usuario común y Administrador	Modificar mis datos personales	Mantener mis credenciales actualizadas	<ul style="list-style-type: none"> - Se tiene que mostrar diferentes campos de texto con la siguiente información: <ul style="list-style-type: none"> ● Nombre completo ● Nombre de usuario ● Número de teléfono ● DNI ● Correo electrónico - Se tiene que validar que los campos sean requeridos. - Se tiene que mostrar un botón con el texto “Modificar Datos”

TABLA 22: Historia de Usuario: Modificar mis datos personales.

N°21: Generar reportes				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Generar reportes	Ayudar a la toma de decisiones	- Se tienen que mostrar diferentes gráficos con información de la playa de estacionamiento.

TABLA 23: Historia de Usuario: Generar reportes.

N°22: Cambiar contraseña				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento, usuario común y Administrador	Cambiar contraseña	Mantener actualizadas mis credenciales	- Se tiene que mostrar dos campos de texto para ingresar la contraseña y repetir la contraseña. - Se tiene que validar que ambos campos sean iguales. - Se tiene que mostrar un botón con el texto "Modificar Datos"

TABLA 24: Historia de Usuario: Cambiar contraseña.

N°23: Buscar reservas				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Buscar reservas	Ver información de una reserva en particular	- Se tiene que mostrar un campo de texto en la barra de navegación con el texto "Buscar Reservas".

TABLA 25: Historia de Usuario: Buscar reservas.

N°24: Registrar nuevo comentario y valoración				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Registrar nuevo comentario y valoración	Dar un feedback al encargado de la playa de estacionamiento	- Se tiene que mostrar un campo de texto para ingresar el comentario y 5 estrellas para seleccionar la valoración. - Se tiene que mostrar un botón con el texto "Calificar"

TABLA 26: Historia de Usuario: Registrar nuevo comentario y valoración.

N°25: Votar un comentario				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Votar un comentario	Dar un feedback al encargado de la playa de estacionamiento	<ul style="list-style-type: none"> - En el listado de comentarios de cada playa de estacionamiento se tienen que mostrar dos botones: uno verde con un ícono de un pulgar hacia arriba y uno rojo con un ícono de un pulgar hacia abajo. - Se tiene que validar que el usuario que vote no sea el mismo que el que hizo el comentario.

TABLA 27: Historia de Usuario: Votar un comentario.

N°26: Listar mis vehículos				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Listar mis vehículos	Facilitar el proceso de reserva	<ul style="list-style-type: none"> - Se tiene que mostrar un listado con todos los vehículos registrados del usuario.

TABLA 28: Historia de Usuario: Listar mis vehículos.

N°26: Dar de alta mi vehículo				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Dar de alta mi vehículo	Facilitar el proceso de reserva	<ul style="list-style-type: none"> - En el listado de vehículos de cada usuario, se tiene que poder agregar un nuevo vehículo con los siguientes datos: <ul style="list-style-type: none"> ● Tipo de vehículo ● Marca ● Patente - Se tiene que enmascarar el campo de texto de Patente con el formato de patente de Argentina.

TABLA 29: Historia de Usuario: Dar de alta mi vehículo.

N°27: Modificar datos de mi vehículo				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Modificar datos de mi vehículo	Facilitar el proceso de reserva	<ul style="list-style-type: none"> - Se tiene que poder modificar los datos de un vehículo.

TABLA 30: Historia de Usuario: Modificar datos de mi vehículo.

N°28: Dar de baja mi vehículo				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Votar un comentario	Dar un feedback al encargado de la playa de estacionamiento	- A definir

TABLA 31: Historia de Usuario: Dar de baja mi vehículo.

N°29: Mensualizar lugar de estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Mensualizar lugar de estacionamiento	Facilitar la búsqueda de lugares fijos	- Se tiene que abrir un formulario para seleccionar usuario y lugar a mensualizar.

TABLA 32: Historia de Usuario: Dar de baja mi vehículo.

N°30: Listar mis mensualidades				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Usuario común	Ver mis mensualidades	Facilitar la búsqueda de lugares fijos	- Se tiene que mostrar un listado con todas las mensualidades, indicando su monto, playa de estacionamiento y lugar asignado. - En cada una, se tiene que mostrar el historial de pagos asociado.

TABLA 33: Historia de Usuario: Ver mis mensualidades.

N°31: Dar de baja mensualidad				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Dar de baja mensualidad	Mantener actualizada la información de mi playa de estacionamiento	- A definir

TABLA 34: Historia de Usuario: Dar de baja mensualidad.

N°32: Ver usuarios mensualizados				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Ver usuarios mensualizados	Analizar la situación de mi establecimiento	- Se tiene que mostrar un listado con los usuarios mensualizados, indicando monto a pagar, día de pago y lugar asignado.

TABLA 35: Historia de Usuario: Ver usuarios mensualizados.

N°33: Listar reservas de un día en particular				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Ver reservas de un día en particular	Realizar un seguimiento de mis operaciones	- Se tiene que poder elegir una fecha determinada y mostrar un listado con todas las reservas de ese día.

TABLA 36: Historia de Usuario: Ver reservas de un día en particular.

N°34: Abrir estacionamiento				
Prioridad	Cómo	Necesito	Para	Criterios de aceptación
1	Encargado de playa de estacionamiento	Abrir estacionamiento	Proveer información correcta a los demás usuarios	- Se tiene que mostrar un botón para poder abrir o cerrar el estacionamiento respectivamente.

TABLA 37: Historia de Usuario: Abrir estacionamiento.

Una vez definidas y priorizadas las historias de usuario, se puede realizar un plan de entregas con una duración de 3 semanas cada una, teniendo en cuenta una capacidad de desarrollo de 16 puntos. Cada entrega será un incremento en el proyecto, que será reflejado mediante una iteración en el Diagrama de Gantt:

Entrega 1 - Total: 16

- Dar de alta una reserva (8)
- Listar reservas del día en tiempo real (8)

Entrega 2 - Total: 16

- Generar pago de reserva (8)
- Administrar horarios y días de apertura (5)
- Autenticar mi usuario (2)
- Dar de alta mi vehículo (1)

Entrega 3 - Total: 16

- Ver playas de estacionamiento (8)
- Modificar datos del estacionamiento y tarifas (5)
- Cerrar sesión (2)
- Modificar datos de mi vehículo (1)

Entrega 4 - Total: 16

- Ver mis reservas (5)
- Ver lugares de mi estacionamiento (5)
- Recuperar contraseña y recibir email de recuperación (2)
- Modificar mis datos personales (2)
- Cambiar contraseña (2)

Entrega 5 - Total: 16

- Ver usuarios (5)
- Modificar datos de usuario (5)
- Dar de alta usuario (5)
- Ver reservas de un día en particular (1)

Entrega 6 - Total: 16

- Dar de baja usuario (5)
- Modificar datos de playas de estacionamiento (5)
- Dar de alta playa de estacionamiento (5)
- Abrir estacionamiento (1)

Entrega 7 - Total: 13

- Dar de baja de playa de estacionamiento (5)
- Cambiar fecha de reserva (2)
- Generar reportes (2)
- Buscar reservas (2)
- Registrar nuevo comentario y valoración (1)
- Votar un comentario (1)

Entrega 8 - Total: 4

- Dar de baja mi vehículo (1)
- Mensualizar lugar de estacionamiento (1)
- Ver mis mensualidades (1)
- Dar de baja mensualidad (1)
- Ver usuarios mensualizados (1)

4.5 Planificación del proyecto

La planificación del proyecto se representará mediante un Diagrama de Gantt en el cual se definirán las iteraciones a realizar, teniendo en cuenta la duración total del proyecto, el bosquejo de requerimientos que se definió con anterioridad y la capacidad del equipo. Según estos dos aspectos, se asignan los recursos correspondientes. Más adelante, en el análisis de factibilidad se estudiarán con detalle dichos recursos. Se consideran horarios de trabajo de 8 horas.

Capítulo IV: Solución propuesta

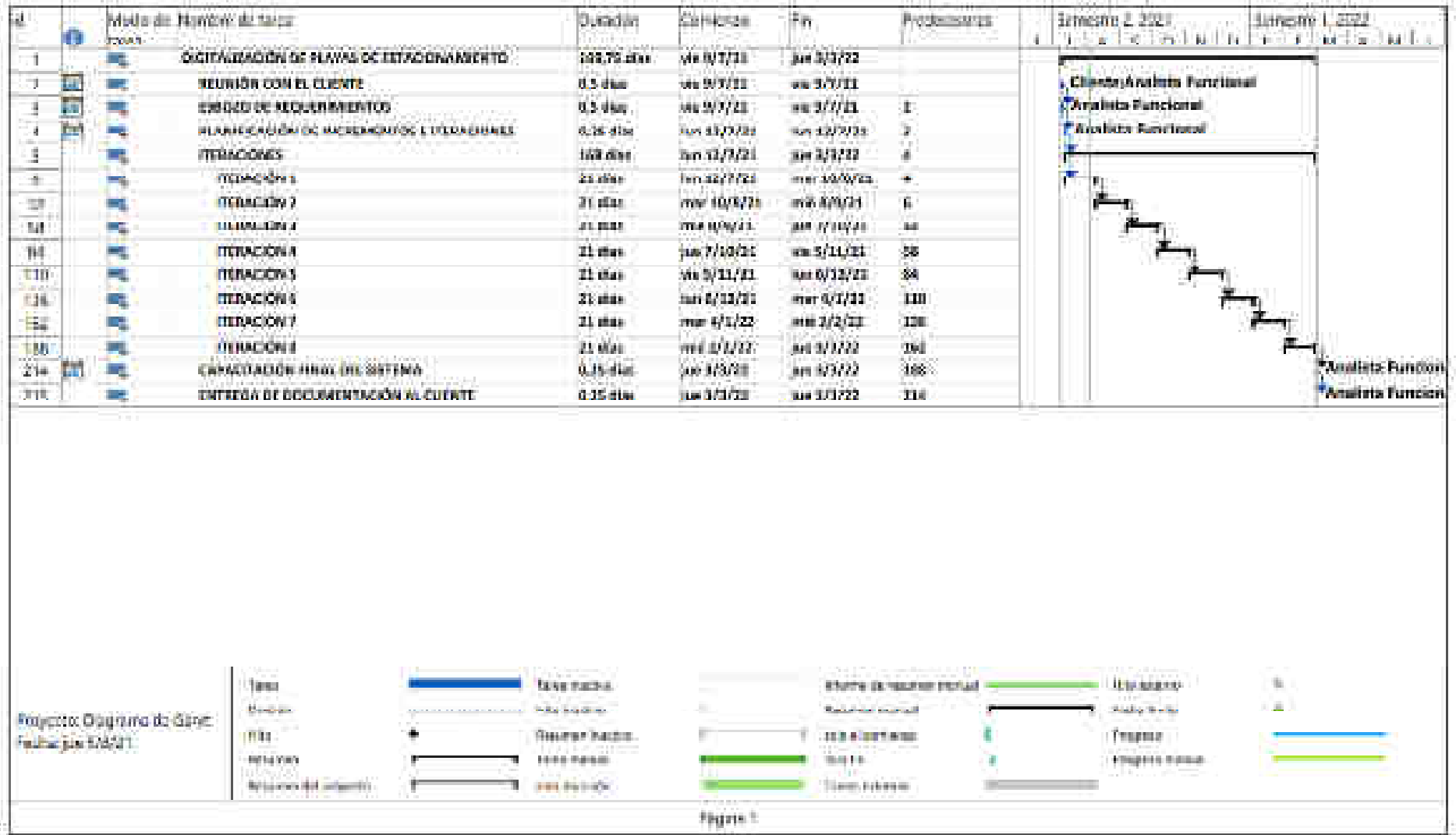


FIGURA 11: Diagrama de Gantt del Proyecto.

Capítulo IV: Solución propuesta

	# Iteración 1	100 días	feb 22/21	ago 30/21	6
	# Iteración 2	11 días	feb 23/21	mar 06/21	6
	# Análisis y diseño	10 días	feb 23/21	feb 26/21	6
	Análisis de requerimientos e impacto	0,25 días	feb 23/21	feb 23/21	6
	Análisis de requerimiento	0,05 días	feb 27/21	feb 13/21	6
	Diseño de arquitectura	2 días	feb 25/21	feb 25/21	6
	Diseño de arquitectura por capas	2 días	feb 25/21	feb 25/21	10
	Evaluación del prototipo de interfaces al cliente	0,75 días	feb 23/21	feb 23/21	10
	Documentación de prototipos de datos	0,25 días	feb 26/21	feb 26/21	10
	# Codificación	4 días	feb 26/21	mar 03/21	
	Evaluación de la implementación de prototipo de datos	0,7 días	feb 26/21	feb 26/21	10
	Desarrollo del código	0,2 días	feb 26/21	mar 2/21	10
	Desarrollo del prototipo de datos	1 día	mar 17/21	feb 26/21	10
	# Pruebas	2 días	mar 03/21	mar 07/21	
	# Pruebas funcionales	1 día	mar 03/21	mar 04/21	
	Pruebas unitarias	1 día	mar 03/21	feb 28/21	17
	Pruebas de integración	1 día	feb 27/21	mar 03/21	10
	Pruebas de aceptación	1 día	mar 02/21	mar 04/21	21
	# Pruebas no funcionales	1 día	mar 04/21	mar 04/21	
	Pruebas de confiabilidad	1 día	mar 04/21	feb 28/21	10
	Pruebas de carga	1 día	mar 03/21	mar 04/21	19
	Pruebas de rendimiento	1 día	mar 04/21	feb 28/21	19
	# Despliegue	1 día	mar 04/21	mar 04/21	
	Construcción de testing	0,75 días	feb 28/21	feb 28/21	26
	Despliegue e instalación de producción de frontend	0,75 días	feb 28/21	feb 28/21	19
	Despliegue e instalación de producción de backend	0,25 días	feb 28/21	feb 28/21	29
	Pruebas de sistema en producción	0,75 días	mar 03/21	mar 03/21	30

Cada iteración tiene el siguiente formato:

Teniendo en cuenta que para distribuir cada una de las fases de cada iteración se basó en la regla de 40-20-40:

Análisis y diseño: 40-50% → 10 días **Codificación:** 15-20% → 4 días **Pruebas y despliegue:** 30-40% → 7 días

4.6 Arquitectura y funcionamiento del sistema

A continuación, se muestra mediante un gráfico ilustrativo la arquitectura utilizada para el desarrollo del sistema que permitirá vincular a cada uno de los clientes que realicen consultas al servidor donde estará alojado el sistema.

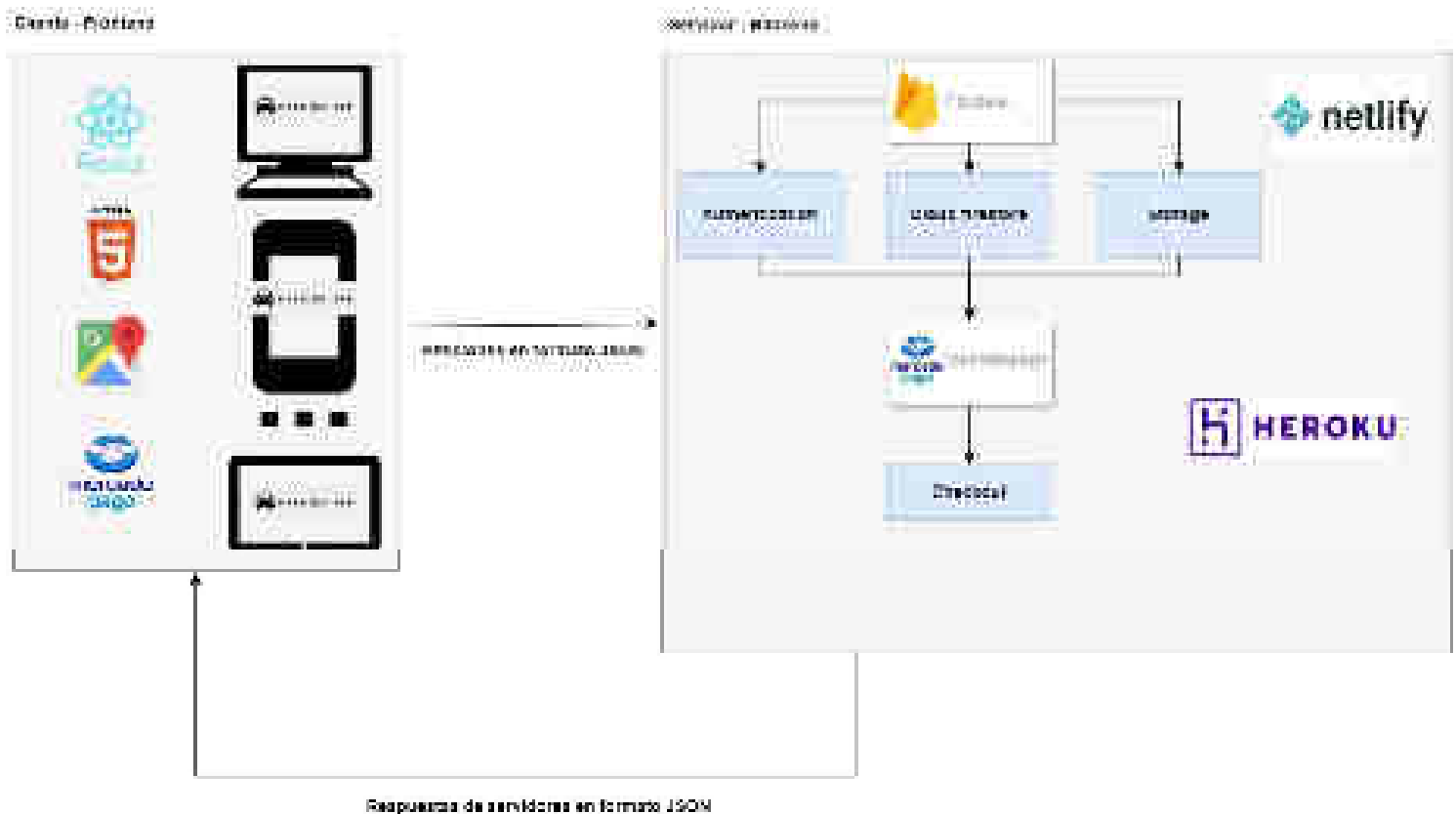


FIGURA 12: Arquitectura del sistema desarrollado.

Este esquema explica el funcionamiento del sistema con un nivel de detalle no tan alto, para que se pueda tener una idea de cómo sería una vez implementado.

Se tiene un cliente, cuyas tecnologías estarán desarrolladas con las siguientes tecnologías: React JS, HTML5, Google Maps y Mercadopago, y que hace referencia a todos los usuarios que van a interactuar con el sistema. mediante funciones de cloud firestore en tiempo real, Firebase Authentication y Cloud Storage:

*Cloud Firestore*²¹ es un servicio de base de datos NoSQL, flexible, escalable y en la nube para permitir la recuperación y sincronización de datos tanto del lado del cliente como del servidor.

*Firebase Authentication*²² por otro lado es un servicio de backend provisto por firebase para permitir la autenticación de usuarios en mi sistema. Permite la autenticación mediante contraseñas, números de teléfono y proveedores de identidad como Google, Twitter, Facebook, Github, etc. Este servicio se relaciona directamente con el resto de soluciones de firebase.

²¹ <https://firebase.google.com/docs/firestore>

²² <https://firebase.google.com/docs/auth?hl=es>

*Cloud Storage*²³ es un servicio de firebase que permite almacenar objetos de forma potente, simple y rentable para el escalamiento de Google. Permite almacenar imágenes, audio y video y otros tipos de contenido generado por el usuario. Este servicio agrega la seguridad de Google a las operaciones de carga y descarga de archivos, sin importar la calidad de la red. Para realizar esto, se tiene que utilizar una clase firebase que tiene ciertos métodos asociados para realizar estas operaciones mencionadas:

Por ejemplo, las funciones de firebase authentication tienen la siguiente estructura:

```

async function iniciarSesion() {
  0. PREVENT default();
  try {
    if(email === "" || contraseña === "") {
      Toast(CGeneral);
      COMPLETE TODOS LOS CAMPOS;
    }
    else {
      await firebase.login(email, contraseña);
      maxTransparencia(0);
      history.push('/home');
    }
  }
  catch (error) {
    Toast(CGeneral);
  }
}

const firebase = firebase.initializeApp({
  async login(email, contraseña) {
    return await this.auth.signInWithEmailAndPassword(
      email, contraseña);
  }
});

```

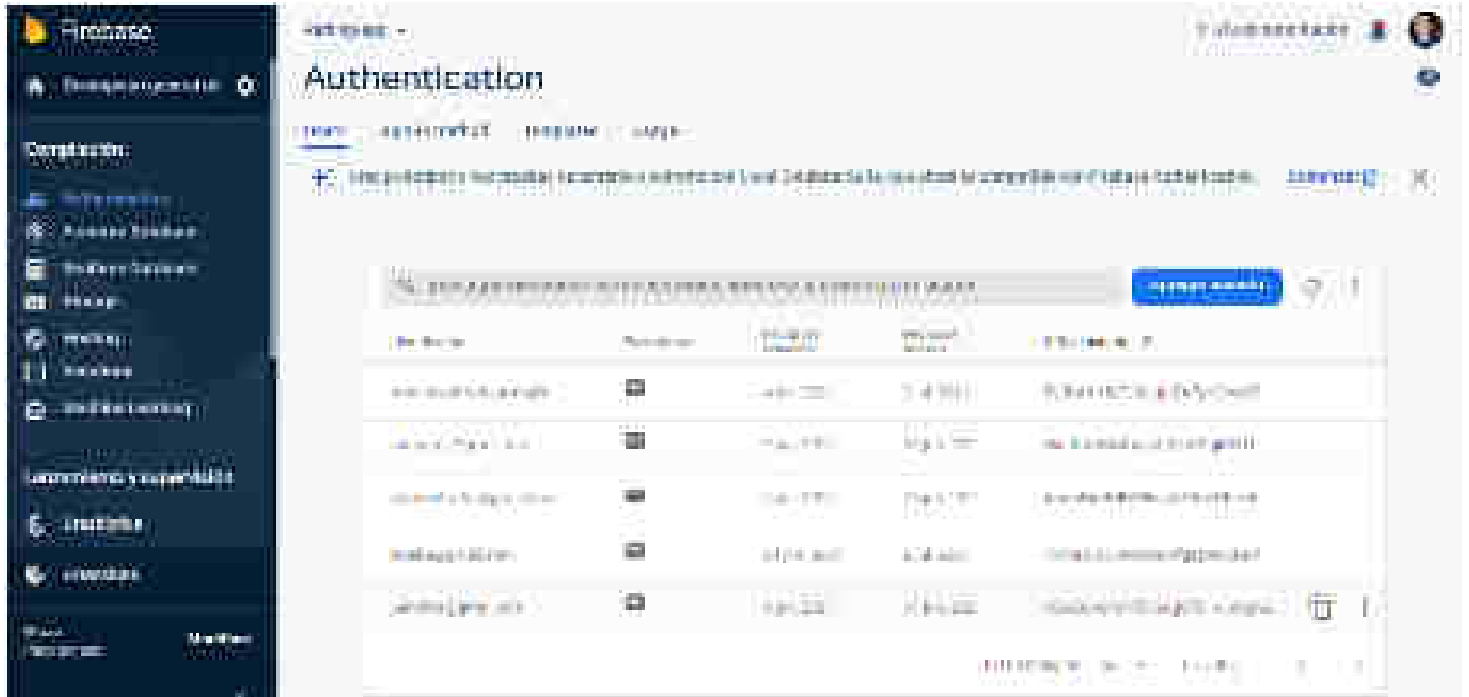
FIGURA 13: Formato de las funciones de firebase authentication.

Se puede ver que tienen una estructura `async - await`, ya que se tratan de peticiones asíncronas no bloqueantes hacia el servidor, esto quiere decir que, mediante esta expresión, se evita que se bloquee la ejecución hasta que se haya completado dicha tarea, en este caso, el inicio de sesión del usuario. Este tipo de instrucciones permite tener una mejora en las respuestas del servidor, y reducir así el tiempo de espera de los clientes. Estas funciones se definen en el componente JSX de React, y se comunica directamente con diferentes métodos de firebase definidos en otra carpeta específica del mismo:

²³ <https://firebase.google.com/docs/storage?hl=es>

SignInWithEmailAndPassword es una función específica de firebase authentication, que permite a un usuario acceder y autenticarse mediante email y contraseña que se encuentran almacenadas en el servicio de backend.

Si vamos a la página de firebase de nuestro proyecto, se pueden ver todos los usuarios que se encuentran registrados en el servicio con los siguientes datos: email identificador,



proveedor de autenticación, fecha de creación, fecha de último acceso y UID de usuario que es un identificador único de cada uno.

FIGURA 14: Dashboard de firebase authentication. Fuente: <https://firebase.google.com/?hl=es>

Desde acá se pueden manejar todos los usuarios, pero solamente esos datos, por lo que al realizar el desarrollo del sistema se relacionó esta información con Firestore Database para poder tener información adicional de cada uno de los usuarios, vinculando mediante el UID de cada uno, de la siguiente manera:

Todos estos datos: dni, nombreCompleto, teléfono son datos necesarios al momento



de realizar las reservas, por lo tanto, es conveniente tenerlos registrados en el sistema.

FIGURA 15 – Dashboard de firestore database. Fuente: <https://firebase.google.com/?hl=es>

Las pantallas del sistema que se comunican con firebase authentication tienen el siguiente formato:



FIGURA 16: Pantalla de login del sistema.



FIGURA 17: Pantalla de recuperar contraseña del sistema.

En ambos casos obtenemos una respuesta del servidor ante la petición, pero en el segundo caso, firebase también permite enviar un correo electrónico para recuperar la contraseña, con el siguiente formato:



Es el formato de las contraseñas al momento de enviarlas.

segundo lugar, el formato de las funciones de algunas funciones.



FIGURA 19: Formato de funciones de firestore.

Se puede observar que tienen el mismo formato delimitado en firebase authentication, con la diferencia principal que el método para agregar un nuevo registro en una colección de firestore tiene la siguiente forma:

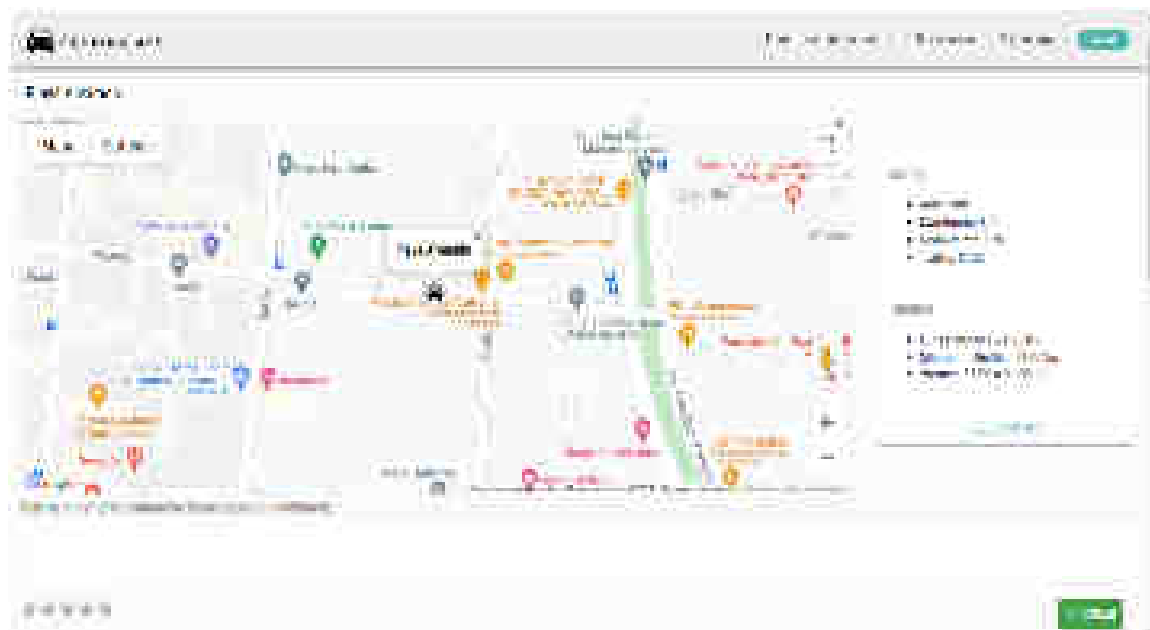
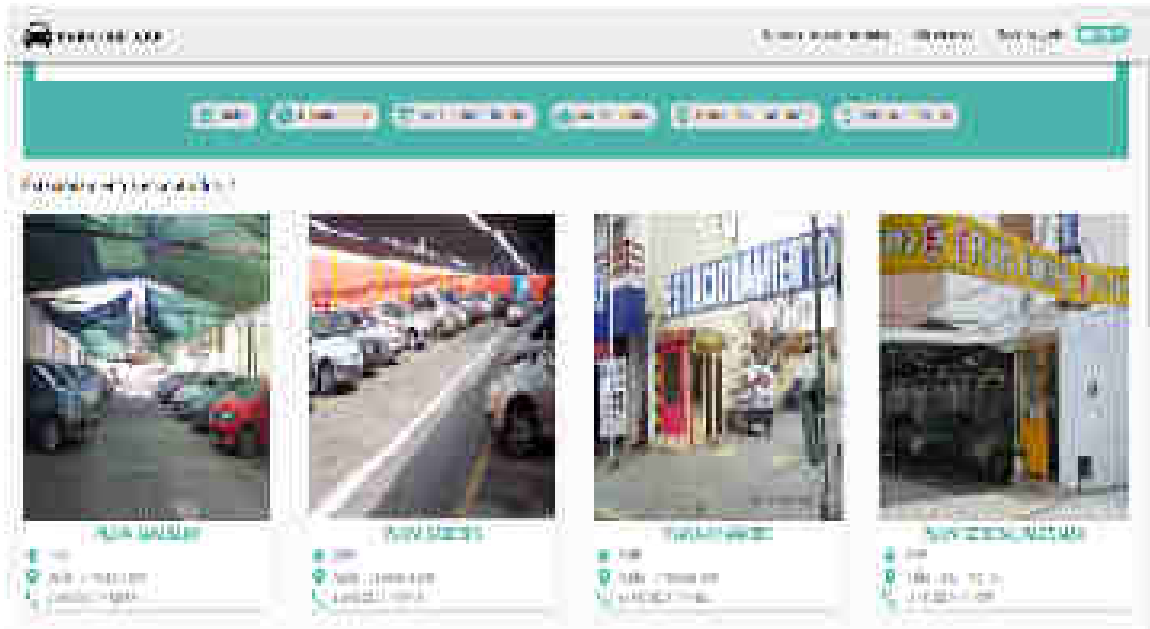
collection('nombre de la colección').nombreOperación

Donde el nombre de la colección puede hacer referencia a cualquier colección que tengamos registrada en nuestra base de datos, y el nombre de la operación puede ser alguno de los siguientes:

- **add**: para agregar una colección o documento a la base de datos.

- **update:** para actualizar una colección o documento a la base de datos.
- **delete:** para eliminar una colección o documento a la base de datos.
- **get:** para obtener una colección o documento a la base de datos.
- y varios más

Las pantallas principales del sistema que utilizan este tipo de operaciones son las



siguientes:

FIGURA 20: Pantalla de listar estacionamientos.

FIGURA 21: Pantalla de detalles de estacionamientos.

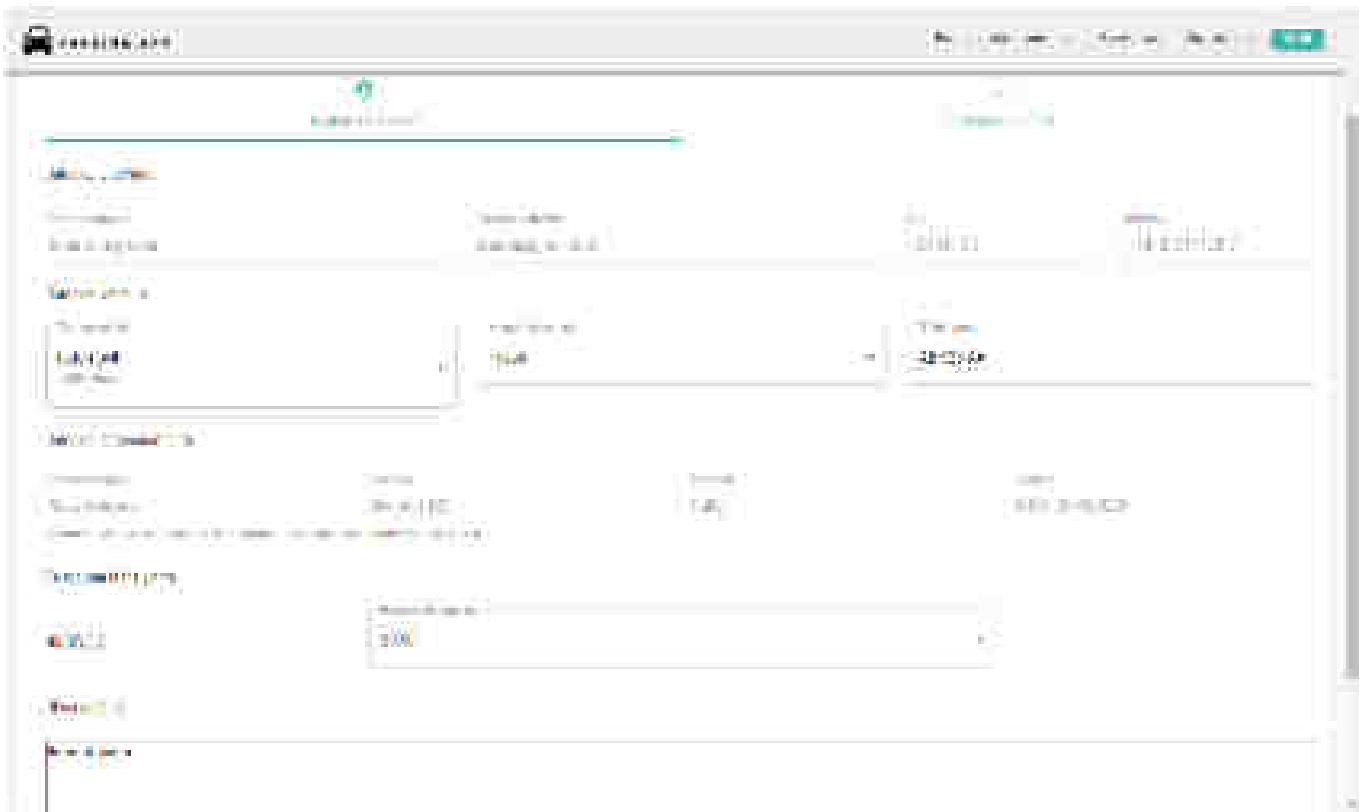


FIGURA 22: Pantalla de registrar nueva reserva.



FIGURA 23: Pantalla de listar mis reservas.



FIGURA 24: Pantalla de listar reservas del día en tiempo real.

FIGURA 25: Pantalla de ver tickets



FIGURA 26: Pantalla de validar reserva.



FIGURA 27: Pantalla de concluir reserva.

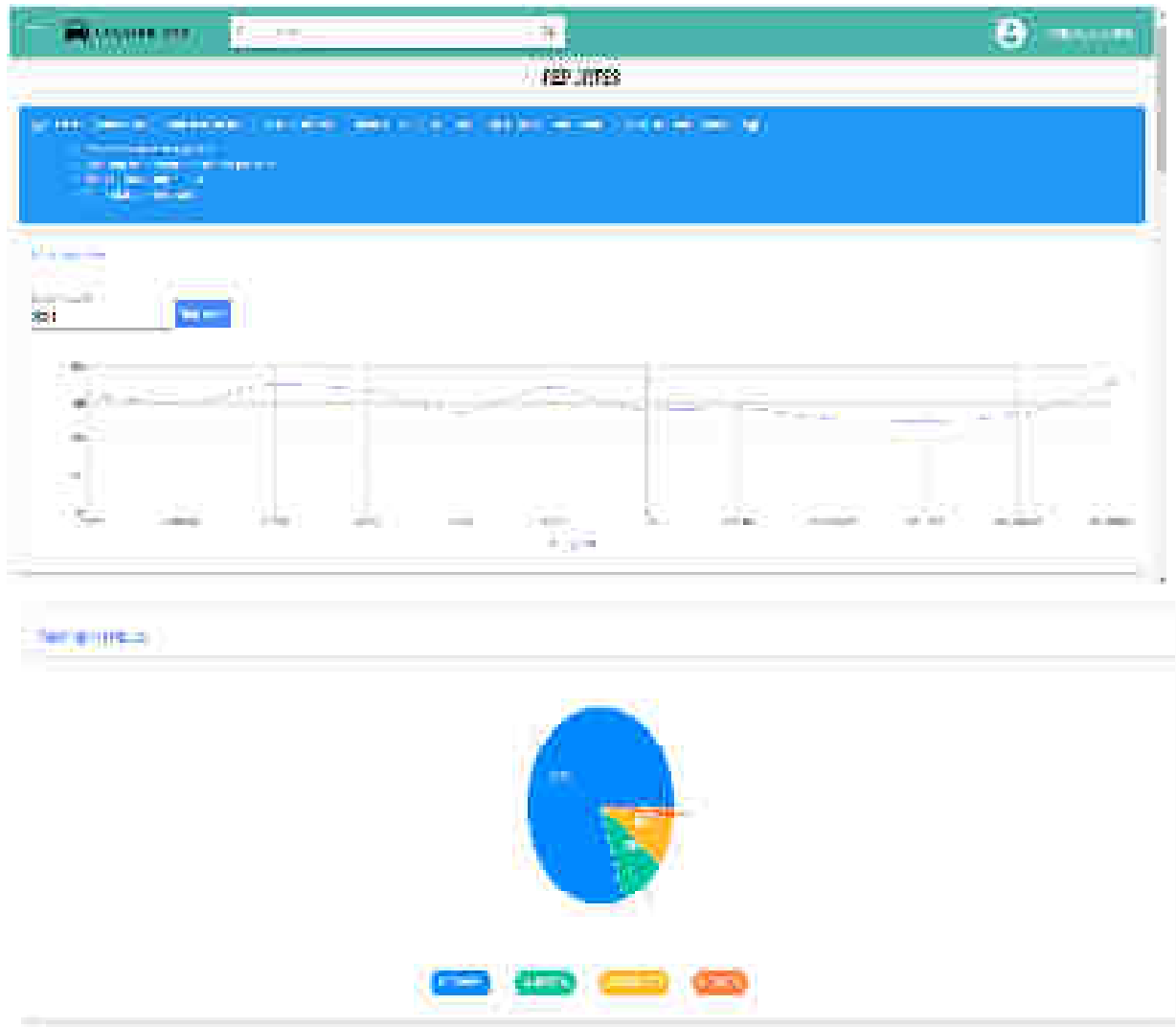
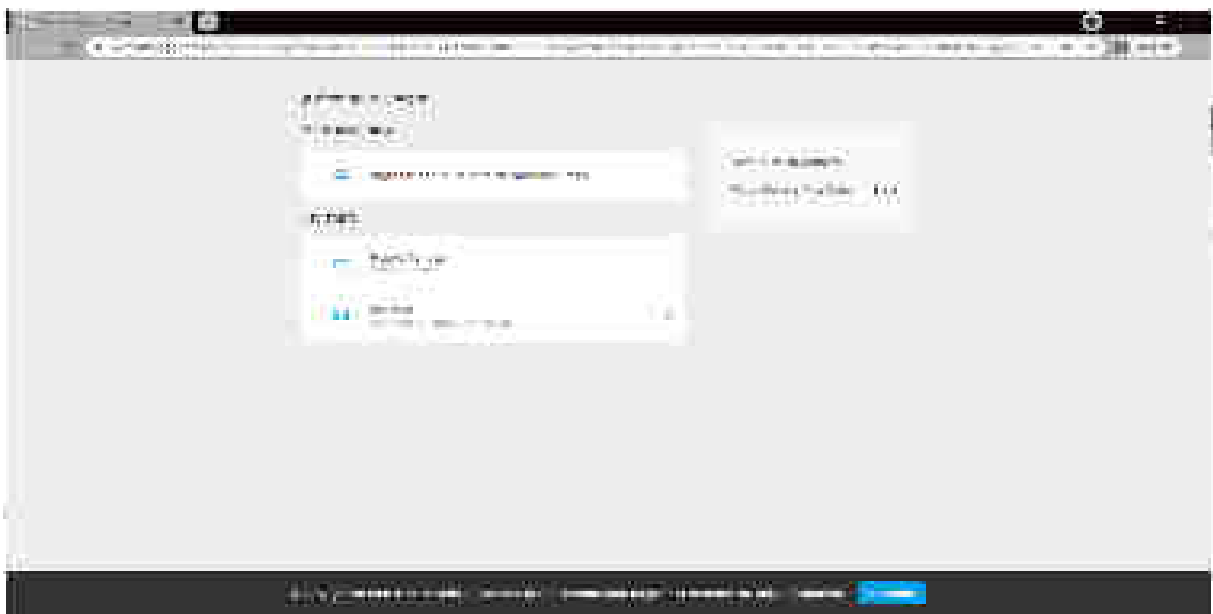


FIGURA 28: Pantallas de reportes.

El servidor del sistema está alojado en Netlify, un servicio de hosting gratuito que se integra con Javascript. Entonces todas las peticiones se realizan hacia la URL²⁴ del sistema:

<https://parking-app-ucasal.netlify.app/>

Al momento de realizar un pago de una reserva, se abre una ventana similar a la

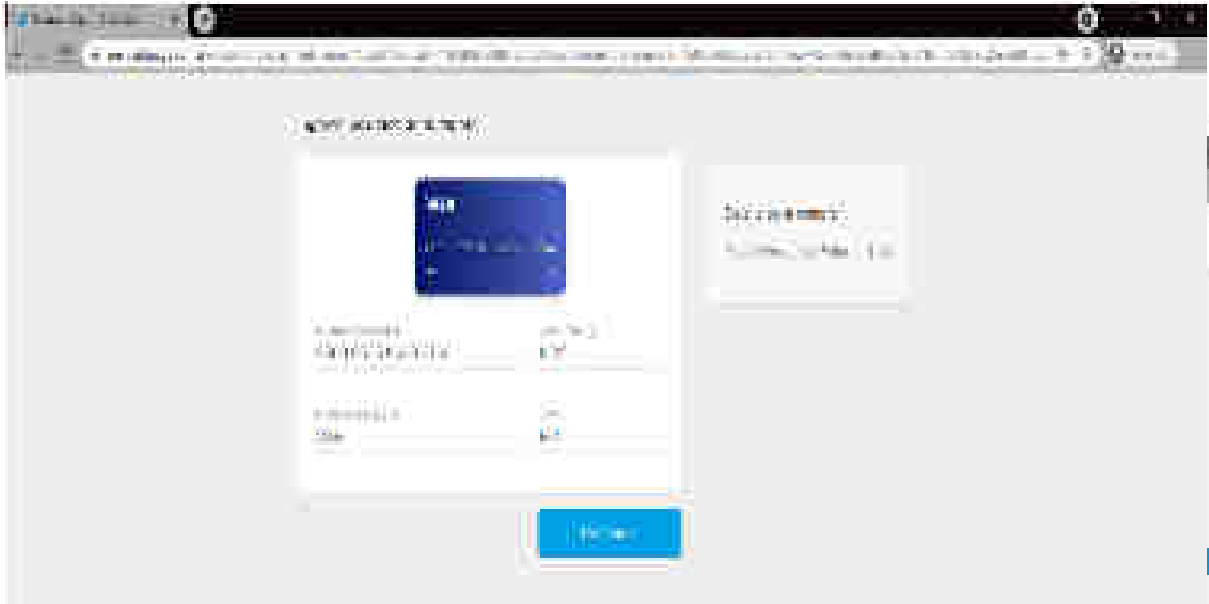


siguiente:

FIGURA 29: Checkout de mercadopago. Fuente: mercadopago.com.ar

Se puede ver que se puede pagar de diferentes maneras:

- Con Cuenta de MercadoPago
- Como invitado con tarjeta de crédito, débito o efectivo en diferentes puntos.



Al elegir tarjeta, nos pedirán los datos de la misma de la siguiente manera:

FIGURA 30: Checkout de mercadopago – Elección de medio de pago. Fuente: mercadopago.com.ar

Cabe aclarar que, mercadopago protege la información en el trayecto desde la computadora, teléfono celular o tablet hasta sus servidores, ya que encriptan los datos en forma segura, mediante certificados TLS y protocolo HTTPS, y además respetan el estándar PCI-DSS, conceptos que se explicaron con anterioridad.

La funcionalidad de checkout de mercadopago requiere que se arme una preferencia, que según la documentación son los datos del producto o servicio que se va a vender. En este caso, solamente se necesita el nombre, el precio unitario de la reserva y la cantidad que por defecto es 1.

Después, se tiene que hacer una petición POST enviando los datos de la preferencia, para que la API de mercadopago automáticamente redirija hacia la URL del formulario para realizar el pago. El código para realizar la petición a la API es la siguiente:


```

const mongoose = require('mongoose');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const bodyParser = require('body-parser');

// Mongoose connection / database
app.use(bodyParser.urlencoded({ extended: false }));

require('dotenv').config({
  path: './.env'
});

const token = process.env.TOKEN_SECRET || 'SECRET_TOKEN';

const app = express();

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(passport.initialize());
app.use(passport.session());

// Routes
app.get('/', (req, res) => {
  res.render('index');
});

app.post('/login', (req, res) => {
  const { email, password } = req.body;

  // Find user by email
  User.findOne({ email }, (err, user) => {
    if (err) return res.status(500).send(err);
    if (!user) return res.status(401).send('User not found');

    // Compare password
    bcrypt.compare(password, user.password, (err, isMatch) => {
      if (err) return res.status(500).send(err);
      if (!isMatch) return res.status(401).send('Invalid password');

      // Generate token
      const token = jwt.sign({ user }, token, { expiresIn: '1h' });

      // Return token and user
      res.json({ token, user });
    });
  });
});

// Middleware
app.use(passport.authenticate('bearer', { session: false }));

// Error handling
app.use((err, req, res, next) => {
  console.log(err);
  res.status(500).send('Internal server error');
});

// Start server
app.listen(4000, () => {
  console.log('Server is running on port 4000');
});

```

FIGURA 31: Funcionalidad de checkout.

Se puede ver que se necesita un token de acceso, según la documentación es una clave privada de la aplicación que se utilizará normalmente en el backend para generar pagos. Es muy importante que este dato quede protegido en los servidores y no sea accesible por ningún usuario del sistema o atacante.

Para crear este pequeño servidor se utilizó Express JS. Dicho servidor localmente escucha al puerto 4000, se encuentra alojado en Heroku, un servicio de hosting de backend que ya se explicó con anterioridad para que el usuario lo pueda utilizar sin necesidad de acceder a dicho puerto. En este servidor está la funcionalidad para realizar el checkout de MercadoPago, que permitirá realizar el pago correspondiente a una reserva realizada. Las peticiones para los pagos se realizan a la siguiente URL:

<https://checkout-mp-parkingapp.herokuapp.com/>

Para realizar la petición se utilizó como endpoint de la URL lo siguiente: /checkpoint, pero esto es a elección del autor, no tiene que ser así obligatoriamente.

Si cualquier usuario quiere acceder desde localhost:4000 o desde la URL de arriba en nuestro navegador, se redirigirá automáticamente hacia la URL del sistema desarrollado y alojado en Netlify, para que el usuario tenga una experiencia mucho más cómoda.

Cabe destacar que, si se quiere tener localmente al servidor, primero debe ejecutarse en la computadora del usuario y por este motivo se debe clonar el repositorio de github en donde se encuentra subido el proyecto. Este proceso se encuentra explicado en el capítulo VIII: Anexos.

4.7 Seguridad del sistema

4.7.1 Roles y permisos

El sistema se desarrolló teniendo en cuenta tres roles diferenciados:

- Administrador del sistema
- Encargado de playa de estacionamiento
- Usuario común

Cada uno de estos roles tiene ciertos permisos asignados, que permiten en mayor o menor medida acceder a ciertas funcionalidades. El sistema valida el tipo de rol que tiene el usuario logueado, consultando a la base de datos de Firestore. De esta forma, las vistas del sistema serán diferentes dependiendo de dicho rol.

Por ejemplo, analizando la barra lateral del sistema se puede observar lo siguiente:

Dependiendo del rol del usuario que se encuentre autenticado en el sistema, el contenido de la misma cambiará, el administrador del sistema no puede ver información relacionada con las reservas ni reportes. Solamente puede dar de alta usuarios y playas de estacionamiento. Por otro lado, un encargado de una playa de estacionamiento solamente puede ver información de su playa, de las reservas que le realizan a la misma y datos personales. No puede dar de alta usuarios ni playas de estacionamiento ni realizar reservas.

Y, por último, un usuario común no puede modificar información de usuarios ni playas de estacionamiento, sólo puede realizar reservas, pagos, generar comprobantes y ver sus reservas realizadas.

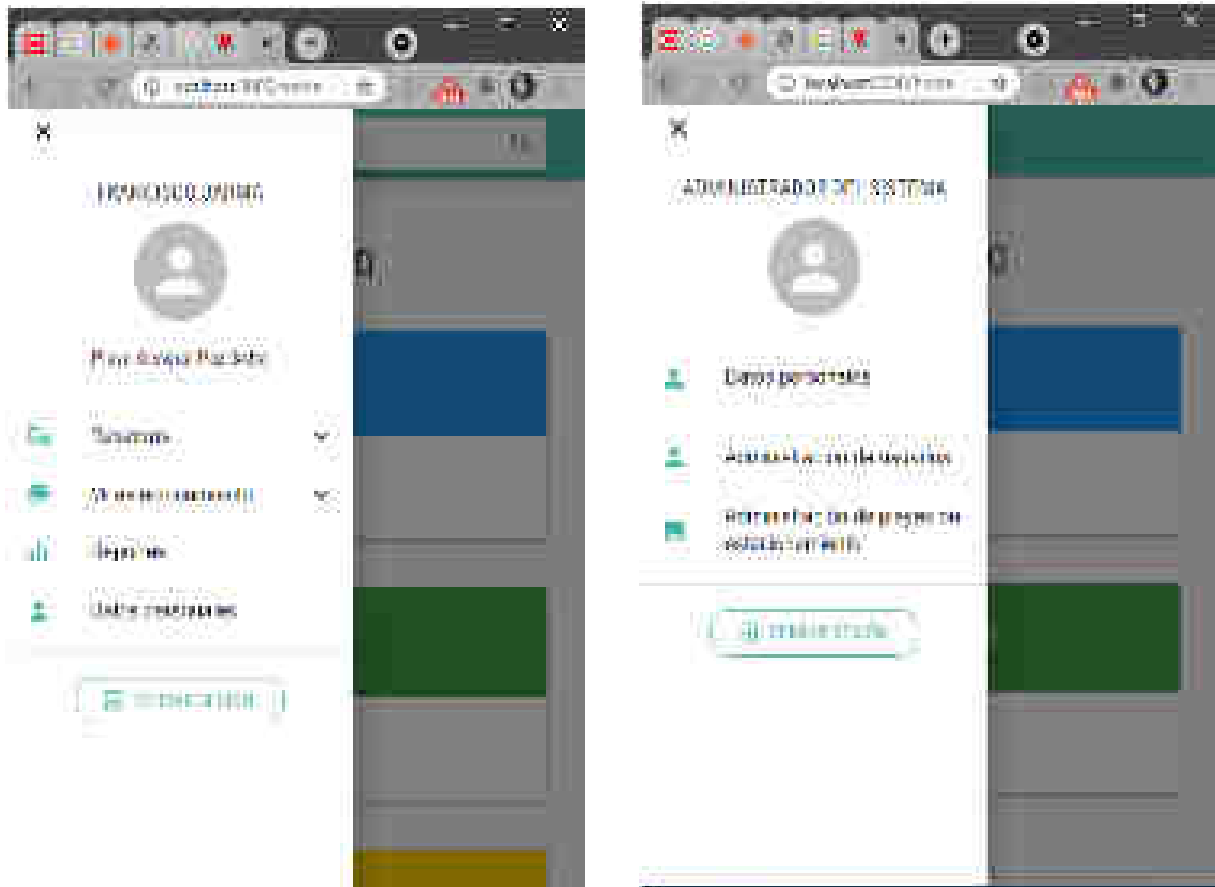


FIGURA 32: Barra lateral del sistema.

4.7.2 Manejo de rutas

React implementa mediante una librería denominada *React Router*²⁵ todo lo relacionado con el manejo de rutas del sistema, incluyendo rutas públicas, privadas y restringidas.

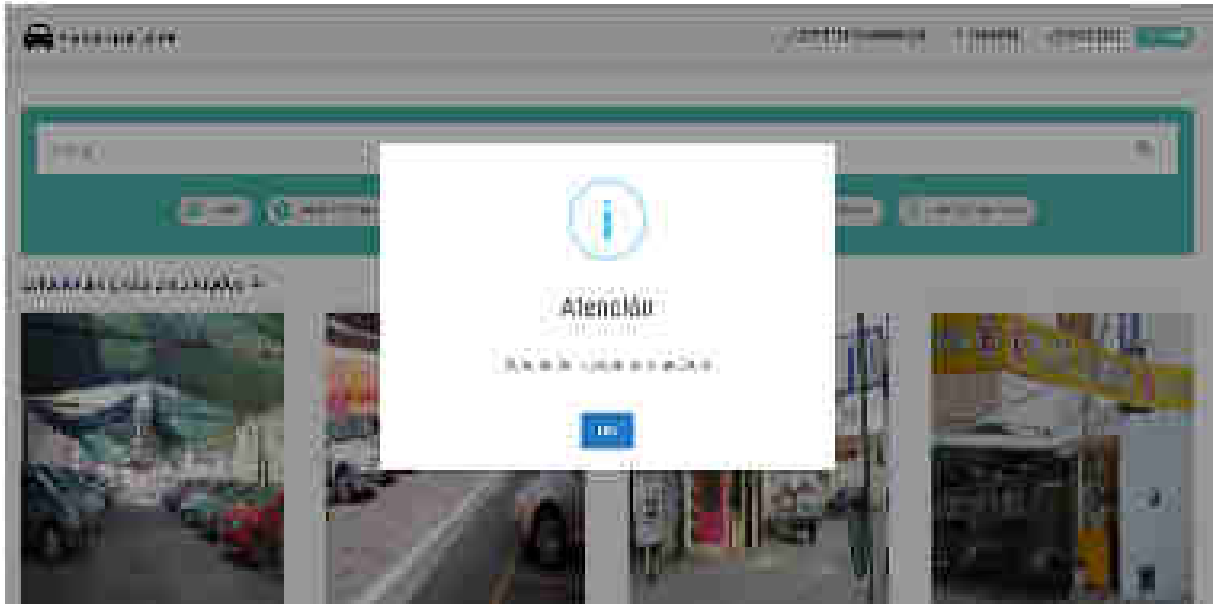
- Las **rutas públicas** son aquellas que pueden ser accedidas por cualquier usuario del sistema, independientemente si está autenticado o no. Por ejemplo: la página de Home.
- Las **rutas privadas** son aquellas que sólo pueden ser accedidas cuando un usuario está autenticado. Por ejemplo: un panel de control.
- Las **rutas restringidas** son aquellas que puede ver un usuario no autenticado, pero, una vez que se autentica, se redirige hacia otra página. Por ejemplo: si el usuario autenticado vuelve al inicio de sesión, se lo redirige a la página principal.

En el sistema desarrollado, la gran mayoría de las rutas son privadas, es decir, que se exige que el usuario esté autenticado para acceder. Existe una ruta pública que es la que tiene la siguiente estructura: /, y hace referencia a la página principal que es visible para todos. Por otro lado, se tienen dos rutas restringidas: /login-encargados y /recuperar contraseña, que no

²⁵ <https://reactrouter.com/>

serán accesibles una vez que el usuario se autentique, ya que el sistema automáticamente lo redirigirá hacia la pantalla de Home.

En caso de que el usuario quiera acceder a una ruta privada, el sistema le notifica con



una alerta de la siguiente forma:

FIGURA 33: Alertas del sistema.

4.7.3 Reglas de seguridad de firebase

Firebase proporciona ciertas reglas de seguridad para brindar protección personalizable hacia los datos almacenados en Cloud Firestore y Cloud Storage. Permite restringir el acceso de tres formas diferentes:

- **Sólo propietario del contenido:** restringe el acceso al contenido por usuario.
- **Acceso mixto:** restringe el acceso de escritura por usuario, pero permite el acceso de lectura público.
- **Acceso basado en atributos:** restringe el acceso a un grupo o tipo de usuario.

Este mecanismo de seguridad es indispensable para mantener protegidos los datos en caso de que se vulnere de alguna forma el método de manejo de rutas de React. Las reglas de seguridad de firebase utilizan una sintaxis muy similar a la de Javascript con el formato JSON, utilizan un lenguaje basado en Common Expression Language (CEL), que utilizan declaraciones match y allow para permitir el acceso condicional. Tienen la siguiente estructura:

```
service <<name>> {
  // Matchea la URL del recurso a validar
  match <<path>> {
    // Permitir la solicitud si las condiciones siguientes son verdaderas
    allow <<methods>> : if <<condition>>
```

```

}
}

```

Se puede observar que hay tres conceptos fundamentales:

- **Service:** declara el servicio de Firebase al que se le aplicarán las reglas: cloud firestore, cloud firestore, etc.
- **Bloque match:** define una ruta en la base de datos al que se le aplican las reglas.
- **Allow:** proporciona condiciones para otorgar acceso. Los métodos compatibles son: get, list, create, update y delete.

4.7.4 Sesión del usuario

Las sesiones de Firebase Authentication son de larga duración. Cada vez que un usuario accede, sus credenciales se envían al backend de Firebase Authentication y se cambian por un token de ID de Firebase y un token de actualización. Los tokens de ID de Firebase tienen una duración de una hora, y el token de actualización se puede usar para obtener tokens de ID nuevos. Los tokens de actualización vencen únicamente en los siguientes casos:

- Si se borró al usuario
- Si se inhabilitó al usuario
- Si se detecta un cambio importante en la cuenta del usuario (esto incluye eventos como actualizaciones de contraseña o de dirección de correo electrónico)

4.7.5 HTTPS - TSL

Tanto heroku como netlify ofrecen sitios seguros para que los usuarios puedan desplegar sus soluciones. Esto quiere decir que, nuestros proyectos tendrán estándar TSL cuando los subamos en cada una de estas plataformas, por lo cual, estarán protegidos ante accesos indebidos por parte de usuarios malintencionados. Ambos sitios crean y renuevan los certificados correspondientes automáticamente, pero si nosotros queremos, podemos agregar un certificado que ya hayamos adquirido con anterioridad, de igual forma con los nombres de dominio.

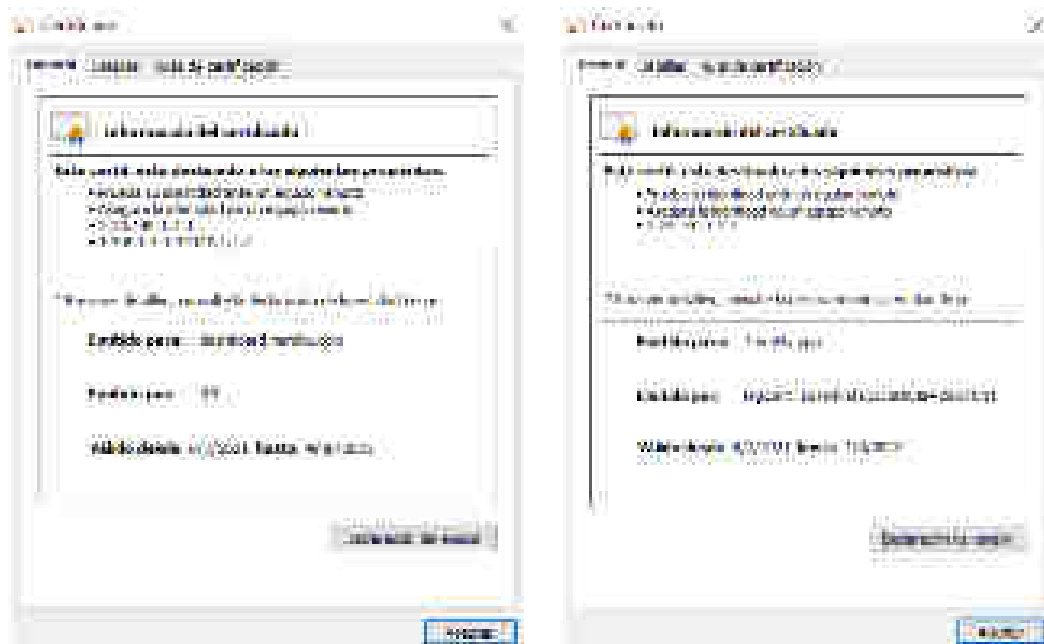


FIGURA 34: Certificados SSL del proyecto. Fuente: netlify.com y heroku.com

4.8 Análisis de factibilidad

Para determinar si es conveniente llevar adelante el proyecto, se tiene que hacer un análisis de factibilidad teniendo en cuenta diferentes criterios para la decisión. También para delimitar los beneficios que tendría su supuesta implementación.

4.8.1 Factibilidad técnica

La factibilidad técnica está relacionada con la infraestructura que se utilizará para llevar adelante el proyecto. En este caso, al tratarse de herramientas alojadas en la nube, el proyecto es factible técnicamente, siempre y cuando se paguen las licencias correspondientes.

En el análisis económico-financiero se detallan las herramientas necesarias para poder implementar satisfactoriamente la solución.

4.8.2 Factibilidad operativa

La factibilidad operativa está relacionada con los recursos humanos necesarios para llevar adelante el proyecto. En este caso, el proyecto sí es factible operativamente ya que sí se disponen de dichos recursos, y estos están debidamente capacitados para utilizar las herramientas necesarias y poder así llevar adelante el proyecto satisfactoriamente. Como se dijo anteriormente, los recursos humanos son los siguientes:

- Analista Funcional
- Analista Programador x 3

4.8.3 Factibilidad económica-financiera

Para ver si un proyecto es factible económicamente hay que analizar los costos iniciales para poder implementarlo, y después comparar con los beneficios esperados al momento de utilizarlo. Si los costos son mayores que los beneficios, entonces no conviene llevarlo adelante.

Cargo	Cantidad necesaria	Costo por hora	Total de horas en el proyecto	Subtotal
Analista Programador	3	\$ 600	1296	\$ 2.332.800
Analista Funcional	1	\$ 720	62	\$ 44.640
TOTAL RR.HH				\$ 2.377.440

Antes que nada, se tienen que definir los costos de recursos humanos:

TABLA 38: Costo de RR.HH.

Como referencia para determinar los costos de recursos humanos se utilizó la escala de honorarios de informática de COPAIPA²⁶, que es el CONSEJO PROFESIONAL DE AGRIMENSORES, INGENIEROS Y PROFESIONALES AFINES. Esta escala salarial está actualizada a agosto de 2021. Se consideró una carga laboral de 175 horas mensuales (4,3 semanas x 40hs semanales).

²⁶ <https://www.copaipa.org.ar/>

Cabe aclarar que, las Licencias de Software están cotizadas en divisa extranjera (dólar estadounidense), por lo tanto, para facilitar la redacción del trabajo se consideró el precio en pesos, teniendo en cuenta la siguiente conversión al mes de agosto de 2021: 1 dólar → 180 pesos argentinos. Además, estas licencias se abonan mensualmente, por lo cual, para facilitar la realización del presupuesto se toman estos valores y se los transforma a un costo por hora.

Después, se definen los costos de software, cuya referencia está en cada una de las páginas web de las aplicaciones, dichas páginas se detallan en el anexo:

Licencia	Cantidad necesaria	Costo por mes	Costo por hora (Costo por día/24)	Total de horas en el proyecto	Subtotal
Firestore Plan Blaze	1	\$ 32.400	\$ 44	1344	\$ 59.136,00
API de Google Maps	1	\$ 360	\$ 0,50	1344	\$ 672,00
API de Google Places	1	\$ 2.080	\$ 1,20	1344	\$ 5.611,80
Netflix Business	1	\$ 18.000	\$ 25	1344	\$ 33.600,00
Heroku Advanced	1	\$ 45.000	\$ 61	1344	\$ 81.084,00
Visual Studio Code	1	\$ 0	\$ 0	1344	\$ 0,00
TOTAL SOFTWARE					\$ 381.036,80

TABLA 39: Costo de Software necesario.

Y, por último, definen los costos de hardware asociados, cabe aclarar que, al tratarse de soluciones en la nube, no hay gastos vinculados con manejo de servidores. Solamente se tendrán en cuenta costos de conectividad y de equipos necesarios. El costo de las notebooks se tiene en cuenta según la cantidad necesaria. Por otro lado, el costo de conectividad depende

Concepto	Cantidad necesaria	Costo
Notebook 15 8gb RAM	3	\$ 225.000
Conectividad (fibra óptica 50mb, \$2800 por mes)	1	\$ 16.800
TOTAL HARDWARE		\$ 241.800

de la cantidad de meses de duración del proyecto, que son 6 en este caso.

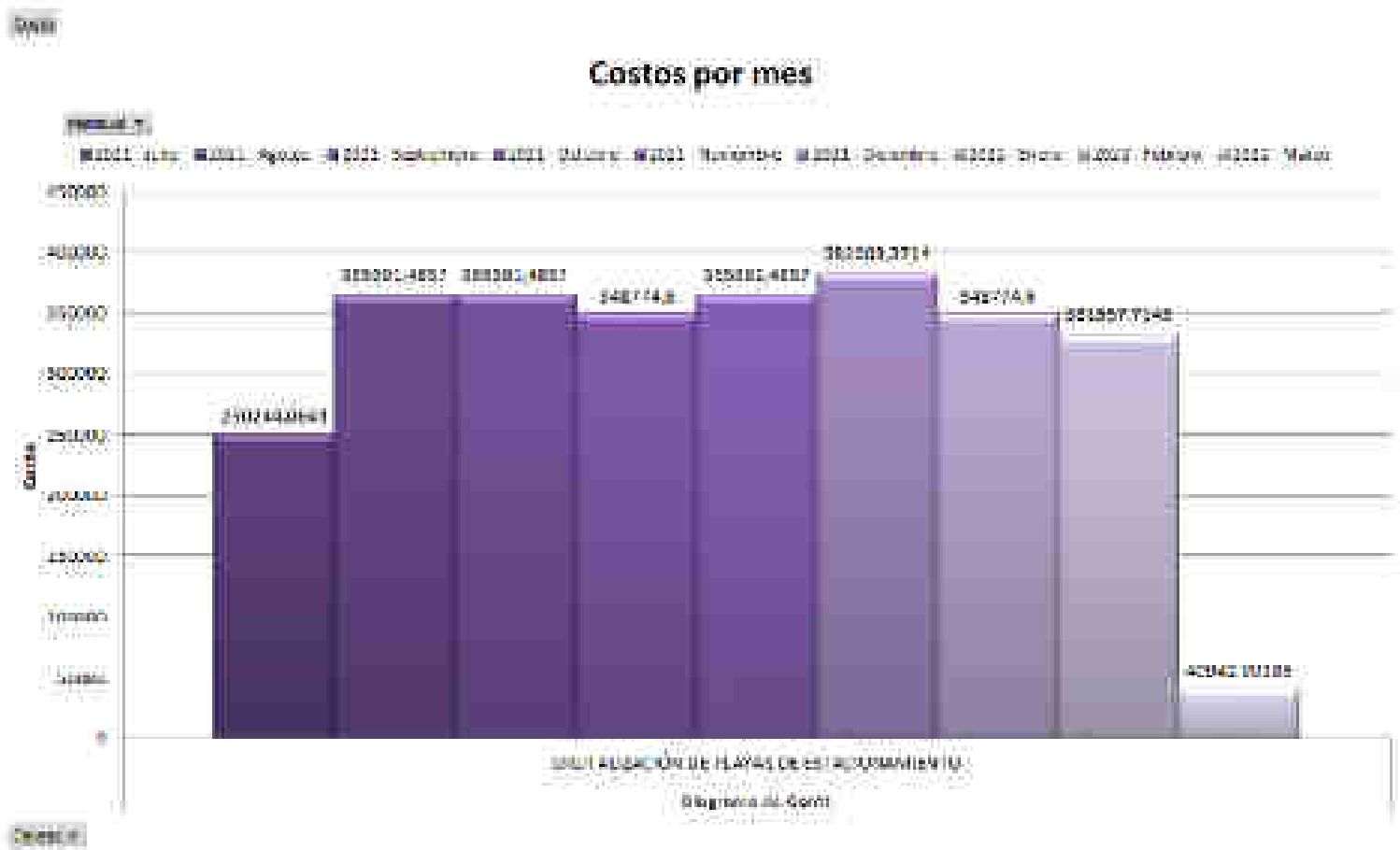
TABLA 40: Costo de Hardware necesario

Entonces, una vez tabulados todos estos datos, se tendría el costo total para llevar

Tipo de Costo	Total
Costo de Recursos Humanos	\$ 2.377.440
Costo de Software	\$ 381.036,80
Costo de Hardware	\$ 241.800
TOTAL DEL PROYECTO	\$ 2.800.276,80

adelante el proyecto:

TABLA 41: Costo total del proyecto.



A continuación, se muestran gráficamente los costos estimados por mes.

FIGURA 35: Gráfico de costos por mes del proyecto.

Cuando el sistema esté funcional y se proporcione las credenciales de acceso a cada uno de los clientes, se tendría que cobrar por cada uno un monto mensual por uso de servicio, para poder hacer frente a los costos totales definidos anteriormente y poder así generar ganancias. Este monto no está incluido en la inversión inicial, ya que se cobraría como una suscripción una vez implementada el sistema.

Suponiendo que, habiendo hecho un análisis de mercado, 30 playas de estacionamiento quieren acceder a este servicio, y se cobra una licencia de \$15.000 mensuales a cada uno, lo cual daría un total de \$450.000 netos mensuales. Aproximadamente en 7 meses se podría recuperar la inversión inicial y empezar a generar ganancias. A continuación, se muestra un gráfico simple que muestra el análisis de flujo de efectivo del proyecto.

Período	Ingresos	Gastos	Flujo neto	Acumulado
jul-21	\$00,00	\$250.244,06	-\$250.244,06	-\$250.244,06
ago-21	\$00,00	\$365.691,49	-\$365.691,49	-\$615.935,55
sep-21	\$00,00	\$365.691,49	-\$365.691,49	-\$981.627,04
oct-21	\$00,00	\$348.774,60	-\$348.774,60	-\$1.330.401,64
nov-21	\$00,00	\$365.691,49	-\$365.691,49	-\$1.696.093,12
dic-21	\$00,00	\$382.608,37	-\$382.608,37	-\$2.078.701,49
ene-22	\$00,00	\$348.774,60	-\$348.774,60	-\$2.427.476,09
feb-22	\$00,00	\$331.857,71	-\$331.857,71	-\$2.759.333,81
mar-22	\$00,00	\$40.942,99	-\$40.942,99	-\$2.800.276,80
abr-22	\$450.000,00	\$00,00	\$450.000,00	-\$2.350.276,80
may-22	\$450.000,00	\$00,00	\$450.000,00	-\$1.900.276,80
jun-22	\$450.000,00	\$00,00	\$450.000,00	-\$1.450.276,80
jul-22	\$450.000,00	\$00,00	\$450.000,00	-\$1.000.276,80
ago-22	\$450.000,00	\$00,00	\$450.000,00	-\$550.276,80
sep-22	\$450.000,00	\$00,00	\$450.000,00	-\$100.276,80
oct-22	\$450.000,00	\$00,00	\$450.000,00	\$349.723,20
nov-22	\$450.000,00	\$00,00	\$450.000,00	\$799.723,20

TABLA 42: Análisis de flujo de efectivo del proyecto.



FIGURA 36: Gráfico de Retorno de la inversión inicial.

Se puede ver gráficamente que, a partir del mes de octubre de 2022 (el séptimo mes de implementación) la curva tiene un punto de inflexión, en el cual hay un cambio en la pendiente de la curva del proyecto, y, por ende, se empieza a recuperar la inversión realizada inicialmente.

La implementación de este proyecto trae consigo asociados muchos beneficios, tanto tangibles como intangibles, se sabe que, los primeros son aquellos que se miden en términos monetarios, los segundos no se miden así, pero tienen un valor de negocio muy importante.

Algunos ejemplos de beneficios tangibles relacionado con la implementación de este sistema web pueden ser:

- **Reducción de costos**
 - De papel para los encargados de las playas de estacionamiento.
 - De combustible para los usuarios comunes.
- **Mejora en la productividad en los procesos**
- **Reducción de tiempos de espera para los usuarios comunes**
- **Automatización en el procedimiento de pagos**
- **Mejora en la respuesta, atención y servicio al cliente**

Y, por otro lado, algunos beneficios intangibles podrían ser:

- **Ahorro de tiempo**
- **Mejora en la imagen de las playas de estacionamiento**
- **Mejor y mayor control financiero**

Estos son algunos de varios beneficios que se esperan, por lo tanto, es rentable implementarlo, entonces decimos que sí es factible económicamente.

4.8.4 Factibilidad ambiental

La factibilidad ambiental hace referencia a si el uso del sistema tendrá algún tipo de impacto en el ambiente. En este caso, al tratarse de un producto intangible, su uso no tendrá ningún tipo de impacto, por lo cual es factible ambientalmente.

4.8.5 Factibilidad legal

La factibilidad legal hace referencia a si el uso del sistema no violará ninguna ley o normativa legal, a lo largo del desarrollo del mismo se utilizaron licencias oficiales de cada una de las herramientas de software mencionadas, y además se respetaron las normativas desarrolladas en el apartado 3.3.3.3, por lo cual, es factible legalmente.

4.9 Análisis FODA

A continuación, se muestra el análisis FODA realizado para explicar las fortalezas y debilidades internas asociadas al desarrollo del proyecto y las oportunidades y amenazas externas asociadas al mismo.

Esta herramienta permite tener una visión mucho más general de la solución propuesta, para poder identificar todos aquellos factores que puedan potenciar o poner en riesgo a la misma.

Este análisis también permite identificar el grado en el que las fortalezas del proyecto van a contrarrestar a las amenazas que puedan aparecer, asimismo, identifica las oportunidades que permitirán superar las debilidades que se encuentran. En base a esto, se establecerán ciertas estrategias para representar cada uno de estos casos.

4.9.1 Análisis interno

4.9.1.1 Fortalezas

- Ofrecer al mercado de Argentina una solución diferenciadora que no se ve mucho en la actualidad.
- Posibilidad de poder utilizarse desde cualquier dispositivo con acceso a Internet y que posea un navegador web.
- El uso de tecnologías muy novedosas en la actualidad, lo cual facilita el escalamiento y la integración de funcionalidades.

4.9.1.2 Debilidades

- Necesidad de conectividad para uso de sistema, es decir, que los usuarios tienen que estar conectados a Internet sí o sí para utilizarlo, esto en algunas zonas puede conllevar a un problema.
- Preferencia de aplicación móvil por parte de los usuarios.

4.9.2 Análisis externo

4.9.2.1 Oportunidades

- Gran crecimiento del mercado automotor, lo cual implica una mayor demanda en soluciones como esta.

- Gran número de playas de estacionamiento que necesitan automatizar sus procesos de negocio.

4.9.2.2 Amenazas

- Las competencias existentes en el mercado, es decir, empresas o personas que ofrezcan soluciones alternativas que tengan las mismas funcionalidades.
- La resistencia al cambio por parte de los usuarios finales, existen muchas playas de estacionamiento que no cambiarán la forma en que realizan las cosas.
- Pandemias mundiales que puedan frenar el trabajo de las playas de estacionamiento.

4.9.3 Estrategias a implementar

Una vez hecho el análisis interno y externo al proyecto a implementar, se definen a continuación las siguientes estrategias:

4.9.3.1 Estrategias FO

Estas estrategias están destinadas a maximizar las fortalezas y en consecuencia aprovechar las oportunidades existentes:

- La posibilidad de que la solución pueda implementarse desde cualquier dispositivo con acceso a Internet permitirá que las nuevas personas que ingresen al mercado automotor puedan utilizarla sin problemas.
- Esta solución diferenciadora permitirá aumentar el nivel de clientes que estén dispuestos a utilizar la aplicación.

4.9.3.2 Estrategias FA

Estas estrategias están destinadas a maximizar las fortalezas para hacerle frente a las amenazas que puedan aparecer:

- Ofrecer una solución diferenciadora permitirá destacarse respecto de la competencia.
- La posibilidad de poder utilizarse desde cualquier dispositivo permitirá disminuir el nivel de resistencia al cambio, ya que hoy en día, casi todas las personas utilizan sus teléfonos para realizar sus tareas.

4.9.3.3 Estrategias DO

Estas estrategias están destinadas a minimizar las debilidades y en consecuencia aprovechar las oportunidades existentes:

- Realizar la aplicación web de tal forma que no se noten muchas diferencias respecto a una aplicación móvil, de tal forma que todos los usuarios estén dispuestos a utilizarla.

4.9.3.4 Estrategias DA

Estas estrategias están destinadas a minimizar las debilidades para hacerle frente a las amenazas que puedan aparecer:

- Realizar la aplicación web de tal forma que no se noten muchas diferencias respecto a una aplicación móvil, para poder así diferenciarla de las posibles soluciones de los competidores.

4.10 Análisis de Riesgos

Según la guía de PMBok el riesgo de un proyecto es un evento o condición incierta que, de producirse, tiene un efecto positivo o negativo en uno o más de los objetivos del proyecto, tales como el alcance, el cronograma, el costo y la calidad. Un riesgo puede tener una o más causas y, de materializarse, uno o más impactos. Una causa puede ser un requisito especificado o potencial, un supuesto, una restricción o una condición que crea la posibilidad de consecuencias tanto negativas como positivas. Los riesgos del proyecto tienen su origen en la incertidumbre que está presente en todos los proyectos. Las organizaciones perciben el riesgo como el efecto de la incertidumbre sobre los objetivos del proyecto y de la organización. Las organizaciones y los interesados están dispuestos a aceptar diferentes niveles de riesgo, en función de su actitud frente al riesgo, en pocas palabras, depende de tres factores: apetito de riesgo, tolerancia al riesgo y umbral de riesgo.

Para representar todas aquellas situaciones que pueden poner en riesgo el desarrollo y la implementación de la solución, se realizará una matriz de probabilidad e impacto para delimitar la probabilidad de ocurrencia de 1 a 5 y si el impacto en el negocio es muy bajo, bajo, medio o alto o muy alto. De esta manera, se podrá identificar los riesgos que tienen que tenerse en cuenta y poder así definir todos los planes de contingencia para mitigarlos o evitarlos:

		Impacto				
		Muy Bajo (1)	Bajo (2)	Medio (3)	Alto (4)	Muy Alto (5)
Probabilidad	Muy Baja (0-0.2]	Aceptar	Aceptar	Aceptar	Aceptar	Transferir/Mitigar
	Baja (0.2-0.4]	Aceptar	Aceptar	Aceptar	Transferir/Mitigar	Evitar
	Media (0.4-0.6]	Aceptar	Aceptar	Transferir/Mitigar	Transferir/Mitigar	Evitar
	Alta (0.6-0.8]	Aceptar	Aceptar	Transferir/Mitigar	Evitar	Evitar
	Muy Alta (0.8-1)	Aceptar	Transferir/Mitigar	Transferir/Mitigar	Evitar	Evitar

FIGURA 37: Matriz de Probabilidad - Impacto.

A continuación, se realizará la identificación de cada uno de los problemas a los cuales está sometido el presente proyecto, con su correspondiente probabilidad e impacto:

N°	Descripción	Probabilidad	Impacto	Acción
1	Vulneración de claves privadas del proyecto	0.3	5	Evitar
2	Malas estimaciones de costos	0.6	3	Mitigar
3	Usuarios indecisos con los requerimientos	0.7	2	Aceptar
4	Insuficiente participación de usuarios	0.6	2	Aceptar
5	Mala estimación de tiempo y alcance	0.6	3	Mitigar
6	Diseño inadecuado	0.4	3	Aceptar
7	Confiar demasiado en herramientas no exploradas	0.8	4	Evitar
8	Falla en servidores en la nube	0.1	5	Mitigar
9	Dificultad de uso de sistema por parte de los usuarios finales	0.8	3	Transferir/Mitigar
10	Robo o pérdida de datos privados de tarjetas de crédito o débito	0.1	5	Transferir/Mitigar
11	Acceso no autorizado al sistema por parte de usuarios malintencionados	0.4	5	Evitar

TABLA 43: Listado de riesgos del proyecto.

Una vez evaluados los riesgos, se pasará a explicar los planes de contingencia para realizar todas las acciones correspondientes para evitar dichos riesgos o disminuir sus consecuencias en caso de que se hagan realidad, de tal forma que se garantice que el proyecto pueda seguir en el rumbo correcto.

Cabe aclarar que los planes de contingencia están planeados para todos aquellos riesgos en los cuáles no se acepte su ocurrencia, y esto depende exclusivamente de la tolerancia al riesgo que puedan llegar a tener los usuarios finales en caso de que ocurran. Para todos aquellos riesgos los cuales se necesite transferirlos, mitigarlos o evitarlos se definirán estas medidas preventivas:

1. Vulneración de claves privadas del proyecto

- Utilizar variables de entorno y excluirlas del repositorio de Github. Estas variables de entorno residen en el sistema operativo de la aplicación, de tal

forma que serán utilizables tanto en un entorno de desarrollo como de producción.

- Proteger dispositivos con usuario y contraseña.

2. Malas estimaciones de costos

- Utilizar información actualizada de honorarios al corriente año.
- Hacer los cambios de divisas correctamente y al día, ya que las licencias del software a utilizar están en dólares.

3. Mala estimación de tiempo y alcance

- Definir las funcionalidades a realizar en función de la capacidad del equipo de desarrollo.
- Negociar el alcance del proyecto con el cliente.

4. Confiar demasiado en herramientas no exploradas

- Tratar de relacionarlas con herramientas conocidas y estudiadas.
- Revisar constantemente la documentación online.

5. Falla en servidores de la nube

- Aplicar políticas de backup y replicación, que sirvan como máquina de tiempo en caso de que aparezcan problemas.
- Migrar las bases de datos hacia otro servidor.
- Proveer un feedback al usuario haciéndole saber de la situación.

6. Dificultad de uso del sistema por parte de los usuarios finales

- Que el Analista Funcional realice capacitaciones a los usuarios finales o que de soporte en los momentos indicados.
- Proveer documentación que explique el funcionamiento del sistema.
- Utilizar interfaces sencillas de entender y que sean intuitivas para el usuario.
- Seguir patrones de diseño estándares para mejorar la interacción del usuario con el sistema.

7. Robo o pérdida de datos de tarjetas de crédito o débito:

- Utilizar variables de entorno.
- Cifrar los datos de las tarjetas, utilizando hasheo de claves²⁷, es decir, aplicarles algoritmos criptográficos para cambiar este texto de entrada en un texto de salida alfanumérica de longitud fija.
- Cerrar sesión de usuario por inactividad.

8. Acceso no autorizado al sistema por parte de usuarios malintencionados

- Autenticar usuario con usuario y contraseña.
- Bloquear al usuario ante tres intentos fallidos de autenticación.
- Monitorear inicios de sesión, usuarios, etc.

²⁷ <https://www.genbeta.com/desarrollo/que-son-y-para-que-sirven-los-hash-funciones-de-resumen-y-firmas-digitales>

Capítulo V: Resultados

5.1 Resultados obtenidos con uso de sistema

Para realizar la verificación del proyecto desarrollado, se realizó la implementación del sistema en diez playas de estacionamiento ubicadas alrededor de Salta, Capital, previo consentimiento de los encargados de estas. Para poder comparar estos resultados con los generados previos a la implementación del mismo. La duración de la implementación fue de 1 mes calendario, contando desde el mes de Julio hasta agosto de 2021.

Cabe destacar que las métricas utilizadas para analizar estos resultados provienen de un análisis y relevamiento de datos exhaustivo, ya que proveen información muy importante de la situación de cada una de las playas de estacionamiento en general. Estas métricas con sus respectivos resultados se representarán de forma gráfica para facilitar su análisis:

Playas de estacionamiento	Ventas Julio 2021	Ventas Agosto 2021	Porcentaje de ventas(%)
1	\$ 35.000	\$ 41.200	12
2	\$ 47.000	\$ 50.200	10
3	\$ 29.000	\$ 37.500	-15
4	\$ 42.500	\$ 40.000	-25
5	\$ 38.700	\$ 45.000	17
6	\$ 29.400	\$ 38.000	10
7	\$ 45.300	\$ 45.000	-1
8	\$ 50.520	\$ 52.000	14,1
9	\$ 42.150	\$ 43.000	8,5
10	\$ 30.900	\$ 35.000	11
TOTAL	\$ 400.470	\$ 420.900	17,5

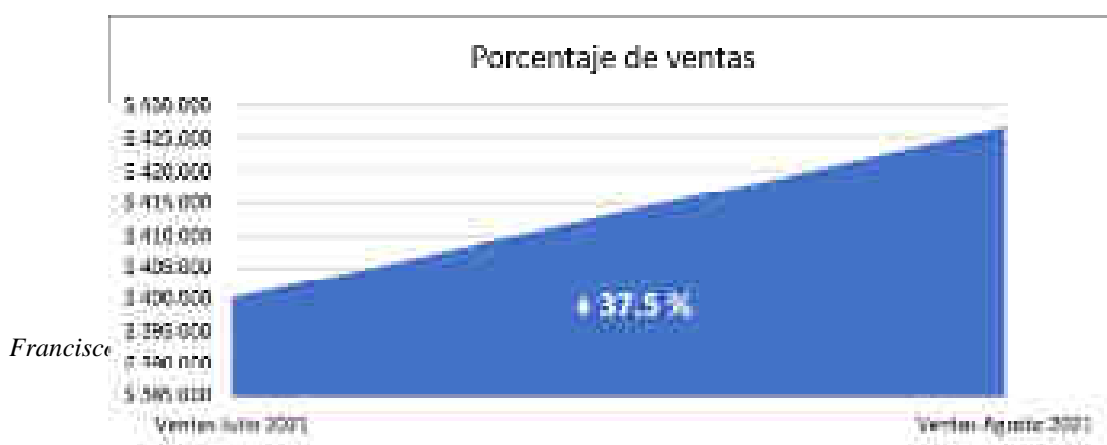
- *Aumento de porcentaje de ventas mensual*

TABLA 44: Datos sobre margen de ventas de playas de estacionamiento relevadas.

FIGURA 38: Gráfico de aumento de ventas de playas de estacionamiento relevadas.

Se puede ver que hubo un aumento considerable del porcentaje total de ventas en tan solo un mes.

- *Disminución de gastos en papel*



Playas de estacionamiento	Gastos Julio 2021	Gastos Agosto 2021	Porcentaje de gastos(%)
1	\$ 8.000	\$ 2.500	-55
2	\$ 8.500	\$ 3.400	-31
3	\$ 7.900	\$ 5.000	-29
4	\$ 7.000	\$ 1.800	-53
5	\$ 4.300	\$ 4.000	-3
6	\$ 5.800	\$ 1.000	-48
7	\$ 5.800	\$ 3.000	-28
8	\$ 9.200	\$ 2.000	-72
9	\$ 5.000	\$ 800	-47
10	\$ 6.700	\$ 200	-67
TOTAL	\$ 68.100	\$ 23.700	-12

TABLA 45: Datos sobre margen de gastos de playas de estacionamiento relevadas.

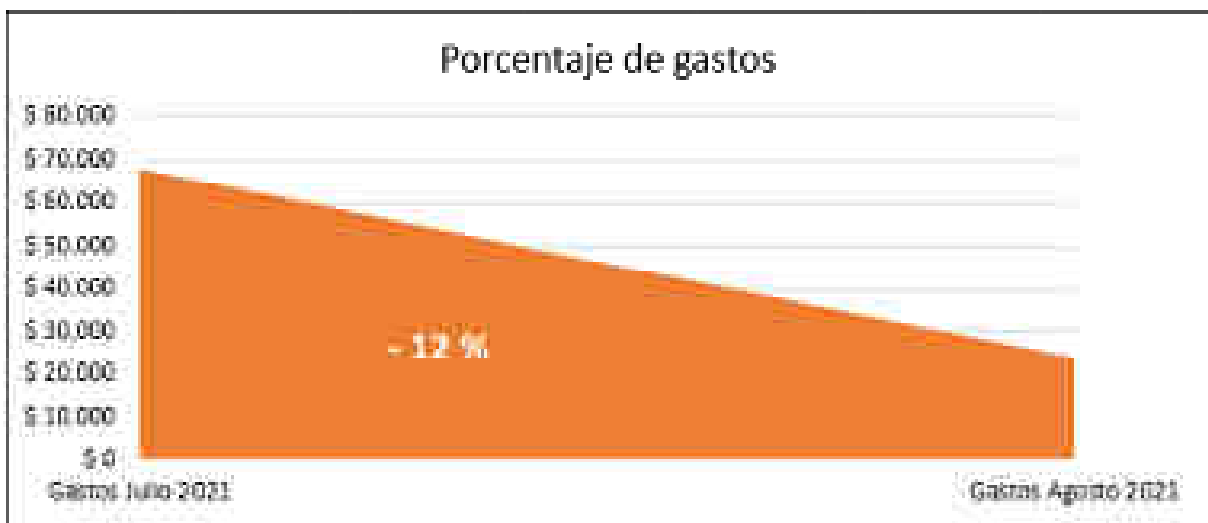


FIGURA 39: Gráfico de disminución de gastos de playas de estacionamiento relevadas.

Se puede ver que hubo una disminución considerable en el porcentaje de gastos de papel en cada una de las playas de estacionamiento en tan solo un mes. Esto era de esperarse, ya que la solución está pensada para reducir a 0 el gasto de papel relacionado con la generación de tickets y talonarios.

5.2 Dificultades no previstas encontradas y sus soluciones

A lo largo del desarrollo del proyecto el autor del mismo se encontró con ciertas dificultades que no estaban previstas al iniciarlo, esto retrasó en mayor o menor medida el progreso hasta poder encontrar las soluciones que se mencionan:

5.2.1 Vulneración de credenciales del proyecto



FIGURA 40: Correo electrónico enviado por google por alerta de vulneración de credenciales del proyecto.

Este problema surgió porque las credenciales del proyecto estaban subidas en el repositorio de Github. La solución fue utilizar variables de entorno²⁸ y esconderlas dentro del repositorio, de tal forma que no sean accesibles por nadie.

²⁸ <https://www.genbeta.com/desarrollo/variables-entorno-que-sirven-como-podemos-editarlas-windows-linux>

5.2.2 Reglas no seguras en la base de datos



FIGURA 41: Correo electrónico enviado por google por alerta de reglas de seguridad no seguras.

Este problema surgió porque las reglas de seguridad de firebase que se establecieron para el proyecto permitían la lectura y escritura de la base de datos, lo cual podría provocar gastos innecesarios en el proyecto. Para solucionarlo se estableció una regla de seguridad para que solamente los usuarios autenticados puedan acceder a la misma. Como se dijo anteriormente en el apartado de seguridad de sistema en el capítulo IV, estas reglas de seguridad tienen el siguiente formato con instrucciones match y allow:

```
rules_version = '2';
service cloud_firestore {
  match /databases/{database}/documents {
    match /{document-**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

FIGURA 42: Regla de seguridad implementada.

5.3.3 Acceso no autorizado de una dirección IP al proyecto



FIGURA 43: Correo electrónico enviado por google por alerta de acceso no autorizado al sistema.

Este problema surgió por el mismo motivo anterior, pero en este caso, sí hubo una vulneración del proyecto ya que una dirección IP desconocida accedió al mismo y quiso utilizarlo para minar criptomonedas, por lo cual, la solución fue la misma que el caso anterior.

Capítulo VI: Conclusiones y futuras líneas de trabajo

A lo largo de este proyecto se pudo demostrar que es posible desarrollar una solución de negocio utilizando tecnologías no relacionales y sin servidor, teniendo un rendimiento mucho más óptimo y costos mucho menos elevados comparado con tecnologías tradicionales de desarrollo como bases de datos relacionales y servidores físicos.

Hay que tener en cuenta que, según las estadísticas obtenidas con la implementación del sistema es posible pensar que esta solución pueda crecer y aplicarse en todas las playas de estacionamiento del país, ya que no hay muchas otras presentes en el mercado.

Además, el desarrollo de este proyecto me ha servido para pensar de una forma totalmente diferente, rompiendo el paradigma tradicional del modelo relacional de datos, lo cual sirve como experiencia al momento de afrontar nuevos proyectos en el futuro.

En futuras líneas de investigación es recomendable el desarrollo de los siguientes trabajos:

- Capacidad de reconocer patentes de los vehículos que ingresan.
- Migrar todo el sistema a una aplicación móvil para una mayor comodidad del usuario.
- Reconocimiento de código QR de la reserva y vinculación con barreras de entrada.

Capítulo VII: Bibliografía.

Acosta, D. Ult. modificación: 30 de 05 de 2019. ¿Qué es pci-dss? <https://www.pcihispano.com/que-es-pci-dss/>. Página vigente al 15 de 08 de 2021.

Alaimo, M. 2013. Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos. Ed. Kleer.

Documentación de Express. (s.f). Recuperado desde: <https://expressjs.com/es/>. Página vigente al 15 de 08 de 2021.

Documentación de Firebase. (s.f). Recuperado desde: <https://firebase.google.com/?hl=es>. Página vigente al 15 de 08 de 2021.

Documentación de Git. (s.f). Recuperado desde: <https://git-scm.com/>. Página vigente al 15 de 08 de 2021.

Documentación de Github. (s.f). Recuperado desde: <https://github.com/>. Página vigente al 15 de 08 de 2021.

Documentación de Heroku. (s.f). Recuperado desde: <https://devcenter.heroku.com/>. Página vigente al 15 de 08 de 2021.

Documentación de Material-UI. (s.f). Recuperado desde: <https://material-ui.com/>. Página vigente al 15 de 08 de 2021.

Documentación de Mercadopago. (s.f). Recuperado desde: <https://www.mercadopago.com.ar/developers/es/guides>. Página vigente al 15 de 08 de 2021.

Documentación de Netlify. (s.f). Recuperado desde: <https://docs.netlify.com/>. Página vigente al 15 de 08 de 2021.

Documentación de NodeJs. (s.f). Recuperado desde: <https://nodejs.org/es/docs/>. Página vigente al 15 de 08 de 2021.

Documentación de React. (s.f). Recuperado desde: <https://es.reactjs.org/docs/getting-started.html>. Página vigente al 15 de 08 de 2021.

Clara, B. 2001. Manual de Derecho Informático. Ed. Nova Tesis Editorial Jurídica.

Consejo Profesional de Agrimensores, Ingenieros y Profesiones Afines. (s.f). Honorarios de Informática de COPAIPA. <http://www.copaipa.org.ar/informatica/>. Página vigente al 15 de 08 de 2021.

Coppola, M. Ult. modificación: 05 de 08 de 2021. Diseño responsive: qué es, cómo funciona y cómo implementarlo. <https://blog.hubspot.es/marketing/disenio-responsive>. Página vigente al 15 de 08 de 2021.

Costa, J. Ult. modificación: 03 de 02 de 2021. ¿Qué es SAAS? El mejor aliado para la transformación digital. <https://www.iebschool.com/blog/que-es-saas-definicion-ventajas-digital-business/>. Página vigente al 15 de 08 de 2021.

De La Torre Valdés, J. Ult. modificación: 01 de 12 de 2020. React - La Guía Completa: Hooks Context Redux MERN +15 Apps <https://www.udemy.com/course/react-de-principiante-a-experto-creando-mas-de-10-aplicaciones/>. Página vigente al 15 de 08 de 2021.

Gustavo B. (s.f). ¿Qué es SSL/TLS y HTTPS? ¿Qué es el certificado SSL? <https://www.hostinger.com.ar/tutoriales/ssl-tls-https>. Página vigente al 15 de 08 de 2021.

Herrera, F. 2021. Ult. modificación: 01 de 03 de 2021. Node: De cero a experto (Edición 2021) <https://www.udemy.com/course/node-de-cero-a-experto/>. Página vigente al 15 de 08 de 2021.

ISACA. Ult. modificación: 17 de 03 de 2021. Geolocalización: Riesgo, problemas y estrategia <https://sites.google.com/a/student.ie.edu/sistgeolocalizacion/>. Página vigente al 15 de 08 de 2021.

Lafuente, A. Ult. modificación: 01 de 08 de 2018. Bases de datos relacionales vs. no relacionales: ¿qué es mejor? <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>. Página vigente al 15 de 08 de 2021.

MDN contributors. Ult. modificación: 17 de 06 de 2021. History API <https://developer.mozilla.org/en-US/docs/Web/API/Window/history>. Página vigente al 15 de 08 de 2021.

MDN contributors. Ult. modificación: 05 de 08 de 2021. Location API <https://developer.mozilla.org/en-US/docs/Web/API/Location>. Página vigente al 15 de 08 de 2021.

MDN contributors. Ult. modificación: 24 de 04 de 2021. SPA <https://developer.mozilla.org/en-US/docs/Glossary/SPA>. Página vigente al 15 de 08 de 2021.

MDN contributors. Ult. modificación: 27 de 05 de 2021. Web Storage API https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API. Página vigente al 15 de 08 de 2021.

Oracle. (s.f). Bases de datos en la nube. <https://www.oracle.com/ar/database/what-is-a-cloud-database/>. Página vigente al 15 de 08 de 2021.

Presta, M. (s.f). Las 5 principales bases de datos en tiempo real <https://blog.back4app.com/es/las-5-principales-bases-de-datos-en-tiempo-real/>. Página vigente al 15 de 08 de 2021.

Red Hat. (s.f) ¿Qué es una API de REST? <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. Página vigente al 15 de 08 de 2021.

Sandra, M. Ult. modificación: 16 de 05 de 2018. ¿Qué es y cómo funciona la nube? <https://mydatascope.com/blog/es/que-es-y-como-funciona-la-nube/>. Página vigente al 15 de 08 de 2021.

Sommerville, I. 2005. Ingeniería de Software 9ma edición. Ed. Addison Wesley.

Stark, G. (s.f). Desarrollo de software a medida. <https://www.evaluandosoftware.com/software-a-medida>. Página vigente al 15 de 08 de 2021.

Capítulo VIII: Anexos

7.1 Normativas

7.1.1 Facturación en guarderías de vehículos y playas de estacionamiento



FACTURACION EN GUARDERIAS DE VEHICULOS Y PLAYAS DE ESTACIONAMIENTO

LEY 7.464
SALTA, 30 de Agosto de 2007
Boletín Oficial, 09 de Septiembre de 2007
Vigente, de alcance general
ID SAN: LPA0007464

Sumario

Derecho comercial, estacionamiento, playas de estacionamiento, información al consumidor, Transporte. El Senado y la Cámara de Diputados de la Provincia de Salta, sancionan con fuerza de Ley y:

Artículo 1º.- En todas las guarderías de vehículos y playas de estacionamiento existentes en el ámbito de la Provincia otorgarán crédito en lugar visible con todos los datos técnicos de la ley.

Art. 2º.- Establécese como requisito indispensable para la habilitación y funcionamiento de guarderías de vehículos y playas de estacionamiento en el ámbito de la Provincia, el uso del "Ticket Tarjetas".

Art. 3º.- Pasada la primera hora de guardería, el tiempo adicional se computará en fracciones de quince (15) minutos en base a las tarjetas expedidas por la respectiva guardería y/o playa de estacionamiento.

Art. 4º.- Por cada fracción de tiempo adicional no podrá facturarse un monto mayor al veinticinco por ciento (25%) de la tarifa percibir por una hora de estadia.

Art. 4º bis.- Los propietarios o explotadores de guarderías y playas de estacionamiento de vehículos, deberán controlar un seguro infocral que otorgue cobertura suficiente respecto de los vehículos hasta su custodia.

La aplicación de esta norma regirá aun cuando el servicio de guardería y/o playa tenga el carácter de adicional a una explotación comercial principal.

Art. 5º.- Para el cumplimiento de lo estipulado en el artículo 1º se establece un plazo máximo de noventa días, a partir de la promulgación de la presente Ley.

Art. 6º.- A los efectos de la presente Ley, será Autoridad de Aplicación la Secretaría de Defensa del Consumidor de la Provincia o el Organismo que en el futuro la reemplace.

Art. 7º.- La presente Ley deberá ser reglamentada dentro del término de sesenta (60) días de su promulgación.

Art. 8º.- Comuníquese al Poder Ejecutivo.

Firmantes

GODOY-PORCELO-CORREGIDOR-CATALANO

FIGURA 44: Normativa sobre facturación en playas de estacionamiento. Fuente: <http://www.saij.gob.ar/>

7.1.2 Obligación para playas de estacionamiento de emitir tickets o comprobantes de ingreso de vehículos

Sistema Argentino
de Administración Judicial

Obligación para playas estacionamiento de emitir tickets o comprobantes de ingreso de vehículos.

LEY 7482
MENDOZA, 8 de Febrero de 2006
Boletín Oficial, 17 de Marzo de 2006
Vigente, de alcance general
ID SAJF LPM10007482

Sumario

playas de estacionamiento, estacionamiento, ticket, Transporte, Derecho administrativo

EL SENADO Y CÁMARA DE DIPUTADOS DE LA PROVINCIA DE MENDOZA, SANCIONAN CON FUERZA DE LEY:

Artículo 1º - Establécese la obligación de emitir ticket o comprobante de ingreso por parte del administrador de las playas de estacionamiento, a todo vehículo que haga uso de las respectivas instalaciones.

Artículo 2º - Quedan incluidas en lo dispuesto en el Artículo 1º:

- a) Las establecimientos que presten servicios de estacionamiento de vehículos automotores, ya sea por hora o por la modalidad de estadia.
- b) Las playas de estacionamiento gratuito, pertenecientes a supermercados o hipercentros de compras.

Artículo 3º - El responsable del estacionamiento deberá, en el ticket o comprobante, insertar en forma mecánica, magnética, manual, o mediante el uso de computadoras y/u otros sistemas similares, el ingreso y egreso del vehículo, con el dato del dominio y del titular o poseedor del mismo.

Artículo 4º - En caso de pérdida del ticket o talón de estacionamiento por parte del usuario del servicio, el propietario del estacionamiento está obligado a consultar sus registros para determinar de manera fehaciente la titularidad del mismo, como así también el tiempo transcurrido desde el comienzo del uso del servicio, a los efectos de hacer efectivo el cobro por tal concepto, no pudiendo obligar al usuario a abonar una suma mayor.

Artículo 5º - El tenedor del comprobante o ticket deberá, al momento de retirar el vehículo, acreditar su identidad a través de cualquier medio fehaciente, a los efectos de confrontar los datos con los que cuenta el titular del estacionamiento.

En caso de existir dudas o imposibilidad de acreditar identidad o titularidad del vehículo, el responsable del estacionamiento no deberá entregarlo, dando el correspondiente aviso a la fuerza policial.

Artículo 6º - La Dirección de Fiscalización y Control será el órgano de aplicación, debiendo la reglamentación determinar las multas a aplicarse en caso de incumplimiento de la presente Ley.

Artículo 7º - Comuníquese al Poder Ejecutivo.

FIGURA 45: Normativa sobre obligación de emisión de tickets. Fuente: <http://www.sajj.gob.ar/>

7.1.3 Playas de Estacionamiento y Garage. Ciclomotores y motocicletas. Se establece la obligación de fijar una tarifa



FIGURA 46: Normativa sobre tarifas en playas de estacionamiento. Fuente: <http://www.saij.gob.ar>

7.1.4 Publicación de leyenda en carteles o tickets en playas de estacionamiento vehicular

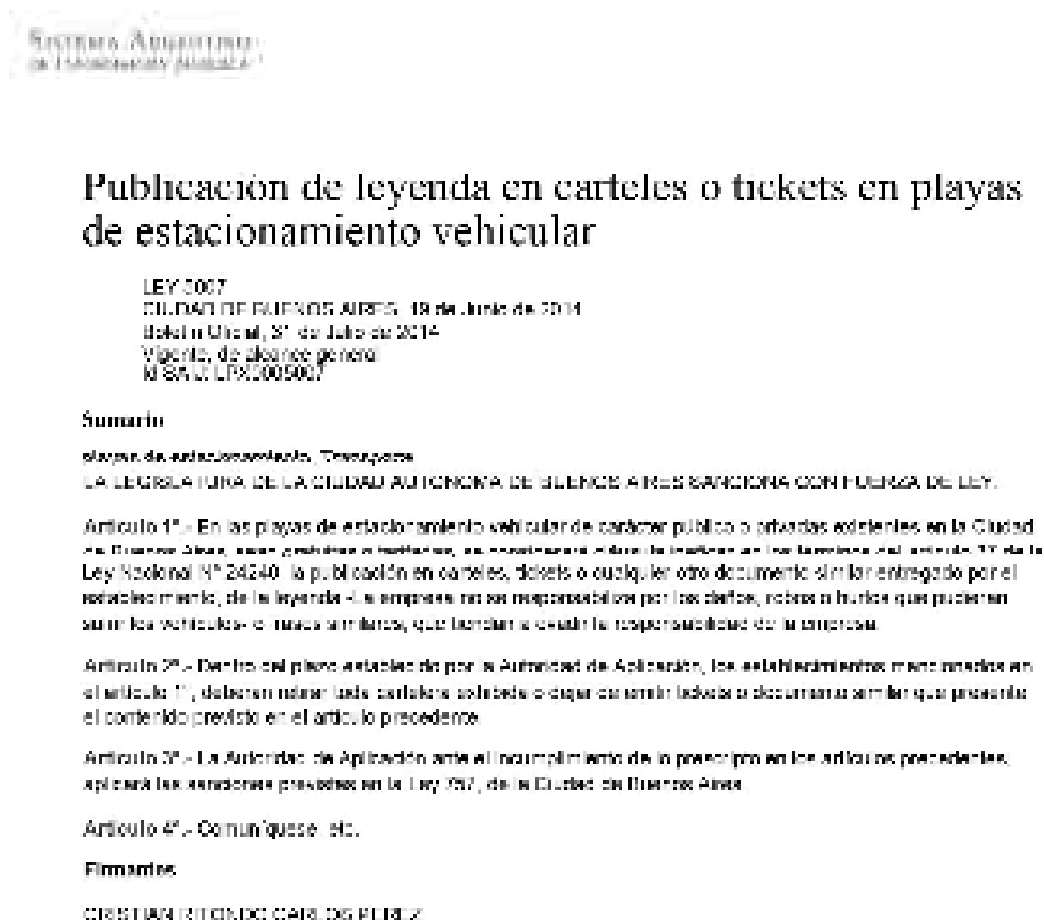


FIGURA 47: Normativa sobre leyendas de tickets de playas de estacionamiento. Fuente: <http://www.saij.gob.ar/>

7.1.5 Registro de playas de estacionamiento



REGISTRO DE PLAYAS DE ESTACIONAMIENTO

LEY 7.772

SAN JUAN, 30 de Noviembre de 2008

Boletín Oficial, 19 de Enero de 2007

Vigente, de alcance general

Inf. Sistema: CONSOLIDADA. VIGENTE EN LEY PROVINCIAL 323-A

Id SAIJ: LPJ0007772

Sumario

registros públicos, playas de estacionamiento, estacionamiento, Transporte, responsabilidad del playero, obligación de seguridad, inscripción en el registro, Derecho administrativo, Derecho civil
LA CAMARA DE DIPUTADOS DE LA PROVINCIA DE SAN JUAN SANCIONA CON FUERZA DE LEY

ARTICULO 1.- La presente ley es de aplicación a las personas físicas y/o jurídicas que a título oneroso o gratuito, por imperio de la legislación vigente o voluntariamente, brinden servicios de estacionamiento de medios de transporte, cualquiera sea la naturaleza de éstos.-

ARTICULO 2.- Los sujetos comprendidos en el Artículo anterior deberán inscribirse, sin perjuicio de las demás obligaciones que establezcan la legislación vigente, en el Registro de Playas de Estacionamiento que se crea a efecto y que estará a cargo del Ministerio de Gobierno, a través de la Policía de San Juan, donde constará como mínimo:

- a) Nombre del propietario del establecimiento, denominación o razón social;
- b) Certificación de habilitación emitida por autoridades competentes;
- c) Nombre del encargado de la Playa de Estacionamiento y del personal que preste servicios en ella;
- d) Sistema del control implementado a efectos del cumplimiento de las obligaciones impuestas por la ley.
- e) Capacidad de estacionamiento de la Playa, indicando separadamente vehículos automotores, motocicletas y bicicletas.-

ARTICULO 3.- Constituye obligación del propietario del estacionamiento informar al Registro creado, toda modificación que se produjera respecto a los datos consignados en el artículo anterior, ello en el término de tres (3) días hábiles de ocurrido.-

ARTICULO 4.- Las Playas de Estacionamiento cuya capacidad exceda de treinta (30) vehículos automotores, deberán obligatoriamente implementar un sistema donde se registre fehacientemente día y hora de ingreso y egreso del vehículo, entregando al usuario el respectivo comprobante y dejando registrado en su sistema, igual información.-

ARTICULO 5.- Constituye obligación y responsabilidad de los propietarios o encargados de la explotación de playas de estacionamiento la asignación de custodios que garanticen la seguridad de los rodados a su cargo que puedan ser objeto de hurto, robo o daño del automotor. El número de personal de custodios deberá estar acorde con la cantidad de rodados ingresados. Este personal estará debidamente identificado y portando de modo visible credencial rubricada por el establecimiento.

REPÚBLICA ARGENTINA
INFORMÁTICA JURÍDICA

La responsabilidad que esta Ley asigna y que se incorpora en las normas del Código de Faltas (Ley N. 8141), no exime de la responsabilidad civil por el resarcimiento ante las mismas situaciones de hurto, robo o daño, resulten obligados los responsables.-

ARTICULO 6.- Constituye obligación del establecimiento verificar la tenencia del comprobante por parte del usuario que egrese de aquel, debiendo, en su caso, tomar las medidas de seguridad correspondientes para verificar el derecho a hacerla, dejando constancia escrita de tal circunstancia.-

ARTICULO 7.- Establécese el siguiente régimen de penalidades por violación a las obligaciones Impuestas en la presente ley:

a) Violación de la obligación Impuesta por el artículo 2:

1. Multa de mil (1000) litros de nafta súper e intimación de clausura en el término de siete (7) días;

2. Clausura del establecimiento;

b) Violación a la obligación Impuesta por el Artículo 3 de la presente Ley:

1. Multa de mil (1000) litros de nafta súper;

c) Violación a la obligación Impuesta por el Artículo 4 de la presente Ley:

1. Multa de mil quinientos (1500) litros de nafta súper e intimación de clausura en el término de siete (7) días;

2. Clausura del establecimiento;

d) Violación a la obligación Impuesta por el Artículo 5 de la presente Ley:

1. Multa de mil quinientos (1500) litros de nafta súper e intimación de clausura en el término de siete (7) días.

2. Clausura del establecimiento;

e) Violación a la obligación Impuesta por el Artículo 6 de la presente Ley:

1. Multa de dos mil (2000) litros de nafta súper.-

ARTICULO 8.- Será Juez competente para intervenir y resolver en las causas formadas por incumplimiento de esta ley, el Juez de Faltas que corresponda, según el lugar de comisión del hecho. De las sentencias que recaigan, podrán deducirse los recursos que establece el Código de Faltas de la Provincia.-

ARTICULO 9.- Es de aplicación en su integridad, salvo en lo que se oponga a la presente, las disposiciones de la Ley N. 8141 (Código de Faltas).-

ARTICULO 10.- Esta ley comenzará a regir a partir de los noventa (90) días de su publicación en el Boletín Oficial de la Provincia.-

ARTICULO 11.- Comuníquese al Poder Ejecutivo.-

Firmante:

LIMA-HERRERO

FIGURA 48: Normativa registros de playas de estacionamiento. Fuente: <http://www.saij.gob.ar/>

7.2 Credenciales de usuarios de prueba

Administrador del sistema:

- Correo electrónico: administrador@gmail.com
- Contraseña: 123456

Encargado de playa de estacionamiento:

- Correo electrónico: franciscoalfredojarma@gmail.com
- Contraseña: 123456

Usuario común:

- Correo electrónico: prueba@gmail.com
- Contraseña: 123456

7.3 Cómo instalar el proyecto localmente

Antes que nada, se tiene que instalar node, se lo puede descargar directamente desde su página web: <https://nodejs.org/es/>. Se puede verificar si se tiene instalado node mediante el siguiente comando en consola, bash de Linux o editor de código: `node`. Nos mostrará información sobre la versión que tenemos instalada:



FIGURA 49: Mensaje de verificación de instalación correcta de node.

Una vez hecha y verificada la instalación, se tiene que clonar el repositorio con el código fuente del proyecto y del servidor de pagos, que se encuentran alojados en github, para esto se utiliza la herramienta de control de versiones git, que se descarga directamente desde su página: <https://git-scm.com/>.

Se recomienda crear dos carpetas distintas para alojar ambos repositorios.

Una vez instalado, se tiene que correr los siguientes comandos desde consola o bash de linux o editor de código:

```
git clone https://github.com/FranJarma/parking-app.git
git clone https://github.com/FranJarma/mercadopago.git
```

Después de realizar la clonación de ambos repositorios, se podrán ver todos los archivos de estos, pero de forma local en el dispositivo de la persona.

Para correr el proyecto principal, se tiene que acceder desde consola, bash o editor de código a la ruta de la carpeta creada para éste y tipiar los siguientes comandos en el siguiente orden:

1. `npm install` (creará la carpeta `Node_Modules`)
2. `npm init`
3. `npm start`

Esto permitirá ejecutar localmente el proyecto que creamos utilizando *create-react-app*²⁹, en el puerto 3000 por defecto.

Por último, para dejar funcionando el servidor de pagos, se debe acceder desde la consola a la ruta de la carpeta creada para este repositorio, hasta donde se encuentra el archivo: `pago.js`. Una vez ahí, se tiene que tipear el siguiente comando: `node pago.js`. Si todo se realiza de forma correcta, tendría que aparecer el siguiente mensaje:



FIGURA 50: Mensaje de verificación de instalación correcta de servidor de pagos.

Y esto permitirá correr el servidor de pagos de manera local.

²⁹ <https://es.reactjs.org/docs/create-a-new-react-app.html>