



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
INGENIERÍA CIVIL EN INFORMÁTICA

SISTEMA DE RESERVA Y GESTIÓN PARA LAVADO DE VEHÍCULOS DESARROLADO CON EL STACK MERN

Por Nicolás Candia Estrada

Memoria para optar al título de Ingeniero Civil en Informática

Profesor Guía : Dr. Miguel Romero Vásquez

Profesor Informante : Dr. Alfonso Rodríguez Ríos

Octubre de 2020

Chillán – Chile

Para Angélica, Esnilda y Helvesia,
mi madre, mi abuela y mi tía
abuela.

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática.

El proyecto titulado “Sistema de reserva y gestión para lavado de vehículos desarrollado con el stack MERN”, busca facilitar el proceso de búsqueda, selección y solicitud de un servicio de limpieza para algún vehículo particular, mediante la implementación de un sistema de reserva. Además, también se busca simplificar el ingreso y acceso a la información de los trabajadores, clientes registrados y registro de servicios realizados, a través de un sistema de gestión para la administración y los trabajadores. Todo esto enmarcado en el desarrollo de un sistema web que posee tres perfiles de usuario: Administrador, trabajador y cliente.

En cuanto a metodología de desarrollo, el sistema fue desarrollado utilizando la metodología cascada, en conjunto con *kanban* y el uso de control de versiones Git.

En términos de desarrollo, se utilizó las tecnologías que componen el *stack* MERN (*MongoDb*, *ExpressJS*, *ReactJS*, *NodeJS*), construyendo una API REST y consumiéndola desde el lado del cliente, todo esto siguiendo la arquitectura Modelo-Vista-Controlador.

Como resultado final se obtiene un sistema web que resuelve necesidades para cada uno de los actores involucrados, y que es totalmente utilizable en cada uno de sus respectivos perfiles, cumpliendo así con el objetivo del proyecto.

Abstract

This project is presented to accomplish the requirements demanded by the Universidad del Bío-Bío in the process of the qualification for the degree of Civil Engineering in Computer Science.

The project entitled " Sistema de reserva y gestión para lavado de vehículos desarrollado con el stack MERN ", seeks to facilitate the process of searching, selecting and requesting a cleaning service for a private vehicle, through the implementation of a reservation system. In addition, it also seeks to simplify the entry and access to information of workers, registered clients and registration of services performed, through a management system for the administration and workers. All this framed in the development of a web system that has three user profiles: Administrator, worker and client.

Regarding development methodology, the system was developed using the waterfall methodology, in conjunction with kanban and the use of Git version control.

In terms of development, it were used the technologies that make up the MERN stack (MongoDb, ExpressJS, ReactJS, NodeJS), building a REST API and consuming it from the client side, all this following the Model-View-Controller architecture.

As a final result, a web system that solves needs for each of the actors involved is obtained, and that is fully usable in each of their respective profiles, thus fulfilling the objective of the project.

Índice General

<u>1</u>	<u>INTRODUCCIÓN</u>	<u>12</u>
<u>2</u>	<u>DEFINICION DE LA EMPRESA O INSTITUCIÓN</u>	<u>15</u>
2.1	DESCRIPCIÓN DE LA EMPRESA	15
2.2	ENTORNO:	15
2.3	DESCRIPCIÓN DEL ÁREA DE ESTUDIO	16
2.4	DESCRIPCIÓN DE LA PROBLEMÁTICA	17
<u>3</u>	<u>DEFINICIÓN DEL PROYECTO</u>	<u>19</u>
3.1	OBJETIVOS DEL PROYECTO	19
3.1.1	OBJETIVO GENERAL	19
3.1.2	OBJETIVOS ESPECÍFICOS	19
3.2	AMBIENTE DE INGENIERÍA DE SOFTWARE	20
3.2.1	METODOLOGÍA A UTILIZAR	20
3.2.2	HERRAMIENTAS A UTILIZAR	21
3.3	DEFINICIONES, SIGLAS Y ABREVIACIONES	22
<u>4</u>	<u>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</u>	<u>24</u>
4.1	ALCANCES	24
4.2	OBJETIVO DEL SOFTWARE	25
4.3	DESCRIPCIÓN GLOBAL DEL PRODUCTO	26
4.3.1	INTERFAZ DE USUARIO	26
4.4	REQUERIMIENTOS ESPECÍFICOS	28
4.4.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA	28
4.4.2	INTERFACES EXTERNAS DE ENTRADA	31
4.4.3	INTERFACES EXTERNAS DE SALIDA	32

<u>5</u>	<u>FACTIBILIDAD.....</u>	<u>33</u>
5.1	FACTIBILIDAD TÉCNICA.....	33
5.2	FACTIBILIDAD OPERATIVA.....	35
5.3	FACTIBILIDAD ECONÓMICA.....	36
5.4	INVERSIÓN.....	36
5.5	COSTO DE DESARROLLO.....	37
5.6	COSTOS FIJOS.....	37
5.7	COSTOS DE OPERACIÓN.....	38
5.8	INGRESOS.....	38
5.9	FLUJO DE CAJA.....	39
5.10	CONCLUSIÓN DE LA FACTIBILIDAD.....	41
<u>6</u>	<u>ANÁLISIS.....</u>	<u>42</u>
6.1	MODELO DE PROCESO DE NEGOCIO.....	42
6.2	DESCRIPCIÓN DEL MODELO DE PROCESO DE NEGOCIO.....	43
6.3	DIAGRAMA DE CASOS DE USO.....	44
6.3.1	ACTORES.....	45
6.3.2	ESPECIFICACIÓN DE LOS CASOS DE USO.....	46
6.4	ESPECIFICACIÓN DE LAS ENTIDADES.....	61
6.4.1	DESCRIPCIÓN DE ENTIDADES.....	61
6.4.2	DESCRIPCIÓN DE RELACIONES.....	61
6.5	MODELAMIENTO DE DATOS.....	62
<u>7</u>	<u>DISEÑO.....</u>	<u>63</u>
7.1	DISEÑO DE FÍSICO DE LA BASE DE DATOS.....	63
7.2	DISEÑO DE ARQUITECTURA.....	65
7.2.1	MVC.....	65

7.2.2	REST.....	66
7.2.3	REDUX.....	69
7.3	DISEÑO INTERFAZ Y NAVEGACIÓN.....	71
7.3.1	NAVEGACIÓN	71
7.3.2	INTERFAZ	73
8	<u>CONCLUSIONES.....</u>	84
9	<u>BIBLIOGRAFÍA</u>	86
	<u>ANEXOS</u>	89
	ANEXO 1: ENLACES A SITIOS WEB DE LAVADOS DE VEHÍCULOS.....	89
	ANEXO 2: RESULTADO DE LOS INCREMENTOS DEL PROYECTO.....	90

Índice Tablas

TABLA 1: REQUISITOS FUNCIONALES DE ADMINISTRACIÓN.....	28
TABLA 2: REQUISITOS FUNCIONALES DE TRABAJADOR.....	29
TABLA 3: REQUISITOS FUNCIONALES DEL CLIENTE	30
TABLA 4: INTERFACES DE ENTRADA.....	31
TABLA 5: INTERFACES DE SALIDA.....	32
TABLA 6: SOFTWARE PARA EL DESARROLLO	34
TABLA 7: REQUISITOS MÍNIMOS DE HARDWARE PARA EL DESARROLLO.....	34
TABLA 8: REQUISITOS DE HARDWARE PARA EL CLIENTE.....	35
TABLA 9: INVERSIÓN.....	36
TABLA 10: COSTO DE DESARROLLO	37
TABLA 11: COSTOS FIJOS	37
TABLA 12: COSTO DE MANTENCIÓN	38
TABLA 13: COSTO DE SOPORTE.....	38
TABLA 14: INGRESOS	39
TABLA 15: DATOS AGRUPADOS.....	39
TABLA 16: FLUJO DE CAJA Y VAN.....	40
TABLA 17: CASO DE USO - ADMINISTRAR SESIÓN.....	46
TABLA 18: CASO DE USO - REGISTRAR SERVICIO DE LIMPIEZA.....	47
TABLA 19: CASO DE USO - VER RESERVAS DEL DÍA.....	48
TABLA 20: CASO DE USO - ASIGNARSE RESERVA.....	49
TABLA 21: CASO DE USO - GESTIONAR SERVICIOS ACTIVOS.....	50
TABLA 22: CASO DE USO - GESTIONAR ENTREGAS PENDIENTES	51
TABLA 23: CASO DE USO - GESTIONAR CLIENTE REGISTRADO.....	52
TABLA 24: CASO DE USO - GESTIONAR EMPLEADOS	53

TABLA 25: CASO DE USO - CONSULTAR ESTADÍSTICAS.....	54
TABLA 26: CASO DE USO - VER REGISTRO DE SERVICIOS	55
TABLA 27: CASO DE USO - VER CALENDARIO DE RESERVAS	56
TABLA 28: CASO DE USO - RESERVAR HORA	57
TABLA 29: CASO USO - GESTIONAR RESERVAS	58
TABLA 30: CASO DE USO - VER DISPONIBILIDAD DE HORAS	59
TABLA 31: CASO DE USO – REGISTRARSE.....	60
TABLA 32: EJEMPLO DE PETICIÓN POST.....	67
TABLA 33: EJEMPLO DE PETICIÓN GET	68

Índice Figuras

FIGURA 1: LOGOTIPO DE LA EMPRESA.....	15
FIGURA 2: BPMN	42
FIGURA 3: DIAGRAMA DE CASOS DE USO.....	44
FIGURA 4: MODELO ENTIDAD RELACIÓN.....	62
FIGURA 5: MODELO FÍSICO DE LA BASE DE DATOS.....	64
FIGURA 6: DIAGRAMA DEL MODELO-VISTA-CONTROLADOR.....	65
FIGURA 7: DIAGRAMA REDUX.....	70
FIGURA 8: MAPA DE NAVEGACIÓN DEL CLIENTE.....	71
FIGURA 9: MAPA DE NAVEGACIÓN DEL TRABAJADOR.....	71
FIGURA 10: MAPA DE NAVEGACIÓN DEL ADMINISTRADOR.....	72
FIGURA 11: INTERFAZ DEL CLIENTE - INICIO	74
FIGURA 12: FORMULARIO DE INICIO DE SESIÓN DEL CLIENTE	75
FIGURA 13: FORMULARIO DE REGISTRO DEL CLIENTE.....	75
FIGURA 14: INTERFAZ DEL CLIENTE - AGENDA.....	76
FIGURA 15: INTERFAZ DEL CLIENTE - QUIÉNES SOMOS	76
FIGURA 16: INTERFAZ DEL TRABAJADOR - INICIO	78
FIGURA 17: INTERFAZ DEL TRABAJADOR - VER RESERVAS.....	78
FIGURA 18: INTERFAZ DEL TRABAJADOR - SERVICIOS ACTIVOS.....	78
FIGURA 19: INTERFAZ DEL TRABAJADOR - ENTREGAS PENDIENTES.....	78
FIGURA 20: INTERFAZ DEL TRABAJADOR - REGISTRAR SERVICIO	79
FIGURA 21: INTERFAZ DEL ADMINISTRADOR - RESUMEN	81
FIGURA 22: INTERFAZ DEL ADMINISTRADOR - ESTADÍSTICAS	81
FIGURA 23: INTERFAZ DEL ADMINISTRADOR – TRABAJADORES.....	82
FIGURA 24: INTERFAZ DEL ADMINISTRADOR – REGISTRO SERVICIOS.....	82

FIGURA 25: INTERFAZ DEL ADMINISTRADOR – CLIENTES.....	83
FIGURA 26: TABLERO KANBAN	91
FIGURA 27: INCREMENTO 1.....	92
FIGURA 28: INCREMENTO 2	92
FIGURA 29: INCREMENTO 3	93
FIGURA 30: INCREMENTO 4.....	93

1 INTRODUCCIÓN

La búsqueda y solicitud de algún servicio suele entrañar alguna dificultad para aquella persona que lo requiere, si este proceso se debe realizar de manera presencial dicha dificultad aumenta, y si a esto se le agrega la posibilidad de que una vez concluido el proceso de búsqueda y selección, el servicio deseado no pueda ser llevado a cabo en el momento idóneo, esto se traduciría en una tremenda pérdida de tiempo para el cliente, o, desde el punto de vista del negocio, en la posible pérdida de uno.

Por otra parte, es conocida la dificultad que conlleva el realizar, y más aún consultar, el registro de la información de un negocio si este proceso se realiza a la manera tradicional, con papel y lápiz. Frente a eso se encuentra la opción de llevar a cabo dichos procesos en hojas de cálculo, pero esto acarrearía otra serie de dificultades, tales como: depender del acceso a la máquina donde estas hojas se encuentren almacenadas; dificultad del control de acceso a la información; contar con una hoja de cálculo muy bien diseñada; probablemente requerir que alguna persona se dedique exclusivamente a registrar toda la información deseada (dada la dificultad de hacer esto mientras se lleva a cabo el servicio) o que muchas personas registren la información una vez finalizada la labor requerida (pudiendo esto incluso extenderse a su tiempo libre); por mencionar algunas.

El desarrollo de este proyecto tiene por objetivo dar solución a esta problemática, de manera concreta, aplicada al negocio de lavado de vehículos y, consecuentemente, adaptándola a las necesidades de este rubro.

La solución aquí ofrecida, consiste en el desarrollo de un sistema Web que cuenta con tres perfiles diferentes: Administrador, Trabajador y Cliente. Donde cada uno apunta a solventar necesidades específicas relativas a cada perfil. Estas necesidades son:

- Administrador: Gestión de la información de los empleados, de los clientes registrados, del registro de servicios realizados y consulta de estadísticas.
- Trabajador: Facilitar el registro de los servicios realizados y gestionar las reservas agendadas.

- Cliente: Ofrecer un sistema de reservas y facilitar la búsqueda y consulta de servicios ofrecidos.

Una de las razones que motiva el desarrollo de este sistema es producto de un contratiempo sufrido por un familiar que, sin entrar en mayores detalles, podría haberse solucionado de manera rápida y sencilla de haber contado con un registro con información de los servicios realizados, en lugar de solo la boleta con el monto.

Una segunda razón que motiva al desarrollo de este sistema, dice relación con la falta de sistemas de gestión, en el rubro, que involucren al cliente como un actor más, cuestión que supondría una ventaja competitiva frente a la gran mayoría de la competencia.

A las dos razones anteriores se suma un incentivo adicional, más bien circunstancial, pero no por ello menos importante. Este se relaciona con la actual contingencia sanitaria mundial¹, lo que, por una parte, conlleva a una mayor preocupación por la limpieza e higiene en general, y por otra, a una mayor demanda de productos y servicios a través de Internet.

En lo que respecta al desarrollo del presente informe, este cuenta con siete capítulos que describen la totalidad del proceso de desarrollo del sistema Web, dichos capítulos que componen el informe son:

- Definición de la empresa o institución: Se presenta y caracteriza la empresa a la que se orientará el desarrollo de la aplicación.
- Definición del proyecto: Se definen tanto los objetivos del proyecto como del ambiente de ingeniería de software.
- Especificación de requerimientos de software: Se especifican los alcances de la aplicación, los objetivos del software, se ofrece una descripción global de la misma y se enuncian los requisitos específicos de la aplicación.
- Factibilidad: Se ofrece el análisis de factibilidad del proyecto, a nivel técnico, operativo y económico.
- Análisis: Se presenta una variedad de modelos que describen el funcionamiento del negocio, de la aplicación, los actores involucrado y el modelamiento de los datos.

¹ Pandemia mundial provocada por el virus SARS-CoV-2, causante de la enfermedad por coronavirus o COVID-19.

- **Diseño:** Se muestra el diseño físico de la base de datos, el diseño de arquitectura utilizado en el desarrollo y el diseño de la interfaz.
- **Conclusiones:** Se ofrece una conclusión relativa tanto a los objetivos del proyecto como al proceso mismo de desarrollo.

2 DEFINICION DE LA EMPRESA O INSTITUCIÓN

2.1 Descripción de la empresa

El presente proyecto apunta a un tipo de negocio en general, el negocio de lavado de vehículos. En la mayoría de los casos, este tipo de organización posee un funcionamiento similar, el cliente lleva su vehículo, se ejecuta el servicio de limpieza y el cliente retira su vehículo. Teniendo en cuenta el funcionamiento general de este tipo de empresas es que se ofrecerá una solución que optimice sus procesos desde las tecnologías de la información.

Para llevar a cabo este proyecto se creó una empresa ficticia para la que se diseñará y desarrollará un sistema de reserva y gestión. La empresa ficticia en cuestión es Auto Limpio Chillán². Con fines de diseño de cara al desarrollo de la aplicación, se creó un logo para la empresa ficticia (Figura 1).



Figura 1: Logotipo de la empresa

2.2 Entorno:

- Cuota de mercado: Referente a la cuota de mercado, se debe analizar el tamaño del parque automotriz de la ciudad de Chillán. Respecto a esto, cabe señalar que

² No está demás explicitar que todos los datos relativos a esta empresa son ficticios (nombres, dirección, cargos, etc.). De igual manera, cualquier diagrama o logo alusivo a esta, ha sido diseñado y creado exclusivamente para este proyecto.

“durante el año pasado se inscribieron 54.803 vehículos motorizados (...) Lo cierto es que para el año en curso las unidades que registrará la urbe superarán las 55.000, lo que corresponde a un alza constante y progresiva que se viene observando en los últimos 14 años.” (Brizuela, 2020). Además de esto, Brizuela (2020) señala que mientras el año 2005, según datos históricos, en la ciudad circulaban alrededor de 19.191 vehículos (autos, jeep, camionetas, microbuses y camiones) para el 2015 la cifra llegó a 42.000 aproximadamente, sin embargo, el alza se registra sobre todo en lo que son los vehículos particulares pues los microbuses y colectivos registran un congelamiento permanente. Teniendo en cuenta esta información es posible ver que existe un amplio mercado al que ofrecer servicios relacionados con el mundo automotriz.

2.3 Descripción del área de estudio

Como ya se ha señalado, las empresas dedicadas exclusivamente al rubro del lavado de vehículo suelen funcionar todas de manera similar, por esta razón, es que el área de estudio será la empresa completa en sí misma.

La intención general es llevar a cabo y orientar buena parte de sus operaciones hacia una plataforma digital (gestión), esto implicaría, por una parte, ofrecer a la administración la capacidad de gestionar a sus clientes, trabajadores y los servicios ofrecidos por la empresa (además de llevar un registro de los mismos), y por otra, ofrecer a los trabajadores una solución sencilla que agilice el proceso de registro de los servicios realizados, con toda la información relevante que ello conlleva. Todo esto de la mano con el desarrollo de un sitio web que permita al cliente no solo saber más de la empresa, sus servicios y sus precios, sino que también lo involucre como un actor más dentro del sistema, ofreciéndole la posibilidad de realizar reservas para llevar a cabo el servicio de limpieza seleccionado. Además del servicio de reservas, se contará con un sistema de notificación al cliente, de tal manera que apenas su vehículo esté listo para ser retirado, éste reciba una notificación vía email.

Las funciones relacionadas con el desarrollo del proyecto son:

- **Administración:** Encargado de las labores de gestión del negocio, a saber, registro de trabajadores y clientes, decidir los servicios que se ofrecen al público, mantener un registro de todos los servicios realizados y tomar decisiones respecto al rumbo del

mismo negocio (contratar más gente, modernizar equipamiento, abrir nuevas sucursales, aumentar o disminuir el horario de funcionamiento, etc.).

- **Trabajador:** Es la persona encargada de llevar a cabo un servicio solicitado por algún cliente, a su vez, es también el encargado de registrar de alguna manera dicho servicio.

En relación a los objetivos del área de estudio cabe destacar:

- Facilitar el registro y acceso a la información de trabajadores, clientes y servicios prestados por la empresa.
- Facilitar a los trabajadores el registro e incorporación de sus actividades.
- Buscar la fidelización de los clientes.

2.4 Descripción de la problemática

Un breve análisis, tanto online como en terreno³, muestra la falta de sistemas de gestión que involucren al cliente como un actor más. Esto se aprecia de manera explícita en la misma forma de operar que tienen muchos servicios o empresas de lavado de autos, esta es: el cliente llega con el vehículo, lo deja (si es que hay cupos disponibles), y luego va a retirarlo en un tiempo estimado que el lavador de turno le indica o cuando el cliente lo cree conveniente. Esto es así para la mayoría de los casos, sin embargo, también es posible encontrar otras variantes, como la posibilidad de reservar horas de manera presencial. El problema de esta variante, es que el cliente debe mínimo ir tres veces de manera presencial para completar todo el proceso de lavado (ir a reservar hora, ir a dejar el vehículo, ir a retirar el vehículo). Otra variante es, desde el punto de vista web, donde si bien existen muchos lavados de auto que cuentan con un sitio, la gran mayoría no cuenta con reserva para el cliente (de todos los sitios revisados, solo uno ofrecía esa posibilidad y esta tenía restricciones⁴), en resumidas cuentas, en la gran mayoría de los casos el sitio web servía como vitrina, lo que en sí mismo no está mal, pero ciertamente es posible aprovecharlo mejor.

³ En terreno se apreció el funcionamiento de una empresa de lavado de vehículos cercana a la intersección de avenida Ecuador con Fermín Vivaceta, cuya forma de operar se basa simplemente en el orden de llegada. También se visitó otra empresa que requería de reserva para realizar el trabajo, la forma de reservar era presencial. En cuanto al análisis online, se visitaron varios sitios web de todo Chile, cuyas referencias serán indicadas en el anexo A.

⁴ <https://lavadolaser.cl/>

Teniendo esta información en cuenta, es posible afirmar que la gran mayoría de lavados de autos poseen un sistema informático mejorable, o simplemente poseen un sistema de gestión manual (papel y lápiz), con todas las dificultades que ello supone⁵.

Habiendo explicitado todo esto, se presenta el problema a resolver: ¿Cómo manejar un lavado de autos involucrando al cliente como un actor del sistema, y aprovechando los recursos tecnológicos?

⁵ El gestionar un negocio de manera totalmente manual, supone muchos sacrificios de tiempo para analizar y registrar los datos, lo que supone dificultades para tomar acciones producto de la información extraída de dichos datos, y a la larga, se traduce en pérdidas monetarias.

3 DEFINICIÓN DEL PROYECTO

3.1 Objetivos del proyecto

3.1.1 Objetivo general

- Desarrollar una aplicación Web que simplifique el ingreso y consulta de datos e información para tres perfiles en concreto: administración, trabajadores y clientes. De tal manera que facilite y satisfaga las necesidades de gestión de la administración, de registro de actividades de los trabajadores, y de búsqueda y solicitud de servicios de los clientes. En pocas palabras, practicar un proceso de transformación digital en la empresa.

3.1.2 Objetivos específicos

- Dar a conocer la empresa y sus servicios a través de la web.
- Integrar al cliente dentro del sistema.
- Optimizar el uso de las instalaciones mediante el sistema de reserva y mejorando los tiempos de entrega de los vehículos con un sistema de aviso.
- Ofrecer una forma de llevar un registro mucho más detallado.
- Que la forma de ingresar un registro detallado sea a la vez simple y rápida.
- Simplificar la gestión de los trabajadores y de los clientes registrados para la administración.
- Ofrecer estadísticas relevantes del negocio a la administración.

3.2 Ambiente de Ingeniería de Software

3.2.1 Metodología a utilizar

Dada tanto la naturaleza del proyecto, como la forma en que se pretendió llevar a cabo el desarrollo del mismo, la metodología que mejor se ajustó a las circunstancias fue la del modelo en cascada, también llamado ciclo de vida clásico.

A pesar de las críticas que se suelen hacer a esta metodología de desarrollo, lo cierto es que para este desarrollo en particular funcionó bastante bien. Esto, por una parte, debido al hecho de que el desarrollo fue aplicado a una empresa ficticia, con los requerimientos claramente definidos desde el comienzo, y por otra, debido a las circunstancias en que se realizó el desarrollo del mismo: trabajo unipersonal, en confinamiento debido a la contingencia sanitaria mundial⁶ y teniendo que intercalar el desarrollo con periodos de estudio de las tecnologías aquí utilizadas para poder pasar a desarrollar así otra parte del sistema.

El modelo en cascada cuenta con cinco fases ("El modelo en cascada: desarrollo secuencial de software", 2019):

1. **Análisis:** planificación, análisis y especificación de los requisitos.
2. **Diseño:** diseño y especificación del sistema.
3. **Implementación:** programación y pruebas unitarias.
4. **Verificación:** integración de sistemas, pruebas de sistema y de integración.
5. **Mantenimiento:** entrega, mantenimiento y mejora.

Si bien estas fases no fueron llevadas a cabo al pie de la letra en su totalidad, si se realizó lo medular de cada una, cuestión que se ve plasmada en el presente informe.

Lo anterior se complementó con *Kanban*, que es un método visual de organización del trabajo mediante el uso de un tablero, con el uso de *yEd* para la elaboración de diagramas, y con el uso de control de versiones Git, en concreto, *GitLab*.

Todo esto permitió el desarrollo del proyecto de una manera progresiva y realista en el contexto universitario.

⁶ Esto imposibilitó el acudir a algún cliente real.

3.2.2 Herramientas a utilizar

Las herramientas a utilizar en el proyecto serán:

- **Bootstrap:** De acuerdo al sitio W3Schools ("Bootstrap 4 Get Started", 2020), bootstrap es un framework de front-end gratuito para un rápido fácil desarrollo web. Este framework incluye plantillas basadas en HTML y CSS, así como también plugins opcionales hechos con JavaScript.
- **Express:** Como se señala en inglés en su sitio web ("Express Web site - Home", 2020), Express es un framework web minimalista y flexible para Node.js, que provee un robusto conjunto de características para aplicaciones web o móviles, y que permite crear API's rápido y fácil.
- **GitLab:** Sistema de control de versiones y trabajo colaborativo en línea basado en Git.
- **MongoDB:** "MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube." ("MongoDB Web site - Home", 2020) Es un sistema de bases de datos no exclusivamente relacionales (no SQL) de código abierto.
- **Mongoose:** Mongoose es una herramienta de mapeo objeto-documento, ODM por sus siglas en inglés, diseñada para trabajar en ambientes asíncronos. Este ODM se utiliza para trabajar con MongoDB.
- **NodeJS:** Node.js se define como un "entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables." ("Acerca de Node.js", 2020)
- **Postman:** Postman es principalmente una herramienta para el testing de API, de acuerdo a su sitio web ("Postman Web site - Home", 2020), Postman es una plataforma de colaboración para el desarrollo de API. Las funciones de Postman simplifican cada paso de la creación de una API y agilizan la colaboración para que pueda crear mejores API, más rápido.
- **React:** "React es una biblioteca Javascript para crear interfaces de usuario (...) diseñada desde su inicio para ser adoptado gradualmente" ("React Web site - Empezando", 2020)

- **Redux:** Según su documentación ("Getting Started with Redux", 2020), Redux es un contenedor predecible de estado para aplicaciones JavaScript, que ayuda a escribir aplicaciones que se comporten de manera consistente.
- **Trello:** Tablero web con estilo Kanban para la organización de tareas.
- **Visual Studio Code:** Editor de código, de código abierto que posee una gran variedad de extensiones que facilitan la labor de programación de acuerdo a cada lenguaje.
- **yEd:** Software gratuito para realizar una gran variedad de diferentes diagramas y modelos.

3.3 Definiciones, Siglas y Abreviaciones

- **API:** Interfaz de programación de aplicaciones, o application program interface por sus siglas en inglés, "es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones." ("Qué son las API y para qué sirven ", 2020)
- **Aplicación:** Hace referencia a un programa diseñado para llevar a cabo alguna tarea o tareas específicas.
- **Biblioteca o librería:** De acuerdo a Wikipedia (colaboradores de Wikipedia, 2020), es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.
- **Footer:** Es aquella parte la interfaz gráfica que se suele ubicar al pie de un elemento, dispuesto de manera horizontal, en forma de fila y que suele ofrecer información y algunos elementos de navegación.
- **Framework:** Entorno de trabajo que cuenta con una estructura conceptual y tecnológica definida.
- **MERN:** Sigla utilizada para describir un conjunto de tecnologías en concreto, estas son: MongoDB, Express, ReactJs y Node.js

- **Navbar:** Podría traducirse literalmente como “barra nav”, y hace alusión a aquella parte de la interfaz gráfica representada por una barra con opciones de navegación. Generalmente se ubica en la parte superior de la interfaz y suele estar presente en todas las páginas de un sitio.
- **NoSQL:** Hace alusión a los sistemas de gestión de bases de datos que no requieren de estructuras fijas para almacenar los datos, y que no son exclusivamente relacionales, *not only SQL*, por la traducción de su sigla.
- **Sidebar:** Podría traducirse literalmente como “barra lateral”, y hace alusión a aquella parte de la interfaz gráfica representada por una barra con opciones de navegación dispuesta como columna.
- **Stack:** Es un conjunto de componentes o servicios utilizados para construir y/o ejecutar una aplicación.
- **REST:** Transferencia de estado representacional, o *representational state transfer* por sus siglas en inglés. Es un tipo de arquitectura de software utilizado para describir un tipo de interfaz entre sistemas que utilicen el protocolo HTTP.

4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

4.1 Alcances

El sistema desarrollado en este proyecto de título puede ser dividido en tres áreas diferentes: Área de administración, área de trabajadores y área de cliente. Cada una con diferentes alcances de acuerdo a los objetivos propuestos.

Área de administración:

- El sistema permitirá a la administración gestionar trabajadores y clientes (agregar, ver, editar y eliminar).
- El sistema permitirá al administrador visualizar las reservas pasadas, presentes y futuras.
- El sistema permitirá al administrador ver el registro de los servicios realizados.
- El sistema ofrecerá al administrador estadísticas relativas a la demanda de cada servicio y la relación entre reservas y servicios realizados sin una reserva previa.

Área de trabajador:

- El sistema permitirá al trabajador ver las reservas del día y asignarse alguna cuando llegue el cliente.
- El sistema permitirá al trabajador ver los servicios que está llevando a cabo.
- El sistema permitirá al trabajador registrar un servicio, aunque este no tenga una reserva previa.
- El sistema permitirá al trabajador ver los servicios finalizados y pendientes de entregar.
- El sistema permitirá al trabajador registrar de manera sencilla cuando un vehículo es entregado una vez que el servicio se ha finalizado

Área de cliente:

- El sistema ofrecerá al cliente (tanto registrado como no registrado) un sitio web en el que consultar información de los servicios ofrecidos, los detalles de la empresa y consultar la disponibilidad de reservas
- El sistema ofrecerá la posibilidad al cliente de registrarse
- El sistema permitirá, al cliente registrado, realizar una reserva para llevar a cabo un servicio.
- El sistema notificará vía email cuando el vehículo esté listo para ser retirado, tanto al cliente registrado como al cliente que desee dar su correo al momento de solicitar un servicio.

Como limitación, cabe destacar que el sistema no contará con un sistema de caja, por lo tanto, no gestionará los cobros de ninguna manera (ni presencial ni online). El sistema tampoco se encargará de llevar un control de inventario de los insumos y maquinarias de la empresa. En lo que respecta a la infraestructura, el sistema tampoco tendrá en consideración el estado u otros pormenores de ésta, la única salvedad a esto, es la cantidad de plazas disponible para reserva online, dato necesario para poder gestionar las reservas de manera adecuada.

4.2 Objetivo del software

Objetivo general:

Mejorar la forma de registro, consulta y modificación de los datos relativos al negocio y los principales actores involucrados en este, además de integrar al cliente como un actor más dentro del sistema, con miras de lograr la fidelización de éste último.

Objetivos específicos:

- Facilitar a la administración el registro, edición, visualización y eliminación de trabajadores y clientes.
- Ofrecer estadísticas relativas a la demanda de servicios y de las reservas versus los servicios realizados sin reserva, de manera gráfica.
- Facilitar a los trabajadores el registro de sus actividades.

- Permitir a los trabajadores visualizar y asignarse de manera sencilla las reservas del día
- Ofrecer a todos los clientes información de la empresa y sus servicios, así como también de la disponibilidad de reservas
- Ofrecer a todos los clientes la posibilidad de registrarse en el sistema
- Ofrecer a los clientes no registrados la opción de recibir una notificación vía email cuando su vehículo esté listo para ser retirado
- Ofrecer a los clientes registrados la posibilidad de realizar reservas a través del sitio web

4.3 Descripción Global del Producto

4.3.1 Interfaz de usuario

Si bien la interfaz de usuario posee una serie características en común, como ya se ha mencionado, la aplicación cuenta con tres grandes áreas, la de la administración, la del trabajador y la del cliente, cada una con una interfaz propia.

Características generales de la interfaz de usuario:

- El logotipo de la empresa⁷ está presente en la interfaz de cada uno de los tres tipos de usuario (administrador, trabajador y cliente).
- El navbar o sidebar de cada tipo de usuario es de color oscuro para lograr un mejor contraste con el logotipo.

Características de la interfaz de administración:

- La interfaz está pensada para ser visualizada principalmente desde un ordenador de escritorio.

⁷ Ver Figura 1.

- Sin perjuicio de lo anterior, la interfaz cuenta con un sidebar de navegación colapsable para facilitar la visualización de la información en pantallas más pequeñas.
- La interfaz cuenta con un navbar siempre presente, en la parte superior, con el nombre del administrador y un botón para cerrar la sesión.
- La información sobre clientes, trabajadores y registros de servicio es presentada en tablas.
- Las acciones que se puedan realizar sobre la información presentada por el sistema estarán representadas por botones con iconografía alusiva a dicha acción.

Características de la interfaz de trabajador:

- La interfaz de está pensada para ser visualizada desde dispositivos móviles.
- La interfaz cuenta con un navbar siempre presente, en la parte superior, con el nombre del trabajador y un botón para cerrar la sesión.
- Los botones del menú principal son grandes e indican de manera clara su función.
- Las acciones que se puedan realizar sobre la información presentada por el sistema estarán representadas por botones con iconografía alusiva a dicha acción.

Características de la interfaz del cliente:

- La interfaz está pensada para ser visualizada tanto desde un ordenador de escritorio, como desde un dispositivo móvil.
- La interfaz cuenta con un navbar siempre presente, en la parte superior, con el menú de navegación y un botón que permita al cliente iniciar sesión o registrarse.
- La interfaz ofrece al cliente un formulario de contacto, información básica sobre la empresa, información sobre los servicios ofrecidos y un calendario que muestra de manera gráfica las reservas agendadas de la semana.
- La interfaz contiene un botón siempre presente, en el costado inferior derecho, que ofrece al cliente la posibilidad de realizar una reserva.

4.4 Requerimientos Específicos

4.4.1 Requerimientos Funcionales del sistema

A continuación, se presentarán los requisitos funcionales asociados al sistema. Dichos requisitos serán agrupados en tablas correspondiente a cada uno de los perfiles del sistema, esto para facilitar su lectura, y para evitar la repetición excesiva a la hora de especificar a quién corresponde cada funcionalidad.

Requisitos funcionales de administración:

Id	Nombre	Descripción
RF_01	Inicio de sesión administración	El sistema debe permitir iniciar sesión a un administrador.
RF_02	Agregar trabajadores	El sistema debe permitir agregar nuevos trabajadores.
RF_03	Ver trabajadores	El sistema debe permitir ver una lista de todos los trabajadores activos.
RF_04	Ver clientes	El sistema debe permitir ver una lista de todos los clientes.
RF_05	Ver servicios registrados	El sistema debe permitir ver una lista de todos servicios que se han registrado.
RF_06	Ver calendario de reservas (administración)	El sistema debe permitir ver un calendario que muestre las reservas de la semana. El administrador podrá ver las reservas de la semana en curso, las pasadas y las futuras.
RF_07	Generar gráfico demanda de servicios	El sistema debe generar un gráfico de barras que muestre la demanda de cada uno de los servicios ofrecidos.
RF_08	Generar gráfico reserva/presencial	El sistema debe generar un gráfico de barras que muestre la cantidad de reservas (cuyo servicio ya se haya realizado) junto a la cantidad de servicios realizados sin reserva previa.
RF_09	Editar trabajadores	El sistema debe permitir editar los datos de los trabajadores.
RF_10	Editar de clientes	El sistema debe permitir editar los datos de los clientes.
RF_11	Desvincular trabajador	El sistema debe permitir marcar a un trabajador como desvinculado (no activo).
RF_12	Eliminar cliente	El sistema debe permitir eliminar a un cliente.
RF_13	Cerrar sesión	El sistema debe permitir cerrar su sesión.

Tabla 1: Requisitos funcionales de administración

Requisitos funcionales de trabajador:

Id	Nombre	Descripción
RF_14	Iniciar sesión (trabajador)	El sistema debe permitir iniciar sesión a un trabajador.
RF_15	Ver reservas del día	El sistema debe permitir ver las reservas del día.
RF_16	Asignar reserva	El sistema debe permitir al trabajador asignarse alguna reserva del día.
RF_17	Registrar servicio	El sistema debe permitir al trabajador registrar la ejecución de un nuevo servicio (que no cuente con reserva).
RF_18	Ver servicios activos	El sistema debe permitir ver los servicios activos del trabajador.
RF_19	Ver entregas pendientes	El sistema debe permitir al trabajador ver todas las entregas pendientes, de todos los trabajadores.
RF_20	Registrar entrega	El sistema debe permitir al trabajador registrar la entrega de un vehículo cuyo servicio haya finalizado.
RF_21	Cerrar sesión	El sistema debe permitir al trabajador cerrar su sesión.

Tabla 2: Requisitos funcionales de trabajador

Requisitos funcionales del cliente:

Id	Nombre	Descripción
RF_22	Iniciar sesión (cliente)	El sistema debe permitir iniciar sesión al cliente registrado.
RF_23	Registrarse	El sistema debe permitir, al cliente anónimo, registrarse en el mismo.
RF_24	Ver calendario de reservas (cliente)	El sistema debe mostrar un calendario con la reservas de la semana en curso, y futuras, al cliente anónimo y registrado
RF_25	Agendar reserva	El sistema debe permitir agendar una reserva, al cliente registrado.
RF_26	Notificación vía email	El sistema debe notificar vía email que el servicio ha finalizado. Esta notificación se hace de manera automática al cliente que haya realizado una reserva, y es opcional para el cliente que concurra directamente a solicitar un servicio.
RF_27	Cerrar sesión	El sistema debe permitir al cliente registrado cerrar su sesión.

Tabla 3: Requisitos funcionales del cliente

4.4.2 Interfaces externas de entrada

A continuación, se describirá cada interfaz de entrada, junto a todos los grupos de datos que serán ingresados al sistema independiente del medio de ingreso. En este caso, se presentarán todas las interfaces de entrada en una misma tabla.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DE_01	Datos del administrador	NOMBRE, RUT, TELÉFONO, EMAIL, PASSWORD
DE_02	Datos del trabajador	NOMBRE, RUT, TELÉFONO, EMAIL, PASSWORD, DIRECCIÓN
DE_03	Datos del cliente	NOMBRE, RUT, TELÉFONO, EMAIL, PASSWORD
DE_04	Datos de servicio	NOMBRE, DESCRIPCIÓN, TIPO DE VEHÍCULO, COSTO, DURACIÓN EN MINUTOS
DE_05	Datos de registro de servicio	SERVICIO, TRABAJADOR QUE EJECUTA, FECHA, HORA DE INICIO, HORA DE FIN, HORA DE RECEPCIÓN, HORA DE ENTREGA, TRABAJADOR QUE ENTREGA, CLIENTE, PATENTE, ESTADO, RESERVA
DE_06	Datos de reserva	CLIENTE, SERVICIO, PATENTE, FECHA Y HORA, ESTADO
DE_07	Datos de agenda	HORARIO DE MAÑANA, HORARIO DE TARDE, PLAZAS DISPONIBLES, RESERVAS
DE_08	Datos de horario	NOMBRE, HORARIO DE MAÑANA, HORARIO DE TARDE, PLAZAS DISPONIBLES

Tabla 4: Interfaces de entrada

4.4.3 Interfaces externas de Salida

Por último, se especificarán las interfaces de salida del sistema, junto al medio de salida. Al igual que en el caso anterior, estas serán presentadas en una sola tabla.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IS_01	Información de trabajadores	NOMBRE, RUT, TELÉFONO, EMAIL, PASSWORD, DIRECCIÓN	Pantalla
IS_02	Información de clientes	NOMBRE, RUT, TELÉFONO, EMAIL	Pantalla
IS_03	Registro de servicios realizados	SERVICIO, TRABAJADOR QUE EJECUTA, FECHA, HORA DE INICIO, HORA DE FIN, HORA DE RECEPCIÓN, HORA DE ENTREGA, TRABAJADOR QUE ENTREGA, CLIENTE, PATENTE, ESTADO, RESERVA	Pantalla
IS_04	Reservas del día	CLIENTE, SERVICIO, PATENTE, FECHA Y HORA, ESTADO	Pantalla
IS_05	Reservas de la semana	FECHA, HORA DE INICIO, HORA DE FIN	Pantalla
IS_06	Gráfico demanda de servicio	Nº DE SERVICIOS REALIZADOS MEDIANTE RESERVA, Nº DE SERVICIOS REALIZADOS SIN RESERVA	Pantalla
IS_07	Gráfico reserva v/s presencial	Nº DE SERVICIOS REALIZADOS POR CADA UNO DE LOS SERVICIOS OFRECIDOS	Pantalla

Tabla 5: interfaces de salida

5 FACTIBILIDAD

5.1 Factibilidad técnica.

En este apartado se analizará la viabilidad del proyecto respecto al software y hardware necesario para llévalo a cabo. Dichos requisitos serán analizados tanto desde la perspectiva del desarrollador, como del cliente.

En cuanto a desarrollo, se llevó a cabo con el *stack* MERN (*MongoDB, Express, ReactJs, NodeJs*), donde todo el conjunto es de código abierto y, por lo tanto, libre para ser usado por cualquier desarrollador.

Debido a que el *stack* completo utiliza *JavaScript* como lenguaje de programación, los requisitos de hardware son muy bajos, y se remiten básicamente a los requisitos del navegador, *IDE* de desarrollo o editor de código utilizado. En este caso, esto fue llevado a cabo utilizando los navegadores *Mozilla Firefox* y *Google Chrome*; y *Visual Studio Code* como editor de código. Adicional a esto, se debe sumar el uso de la *suite de ofimática Office 2016*, que se utilizó para generar todo tipo de documentación necesaria. Habiendo especificado esto, se tomarán como los requisitos de hardware, los requisitos mínimos del software que tenga los requisitos más altos dentro del conjunto recién mencionado, dicho software es *Office 2016*. En la siguiente tabla se especifican los requisitos de software y los requisitos de hardware recomendado según el sitio oficial de Microsoft (Microsoft, 2020).

Software	Licencia
MongoDb	Gratuita
ExpressJs	Gratuita
ReactJs	Gratuita
NodeJs	Gratuita
Mozilla Firefox	Gratuita
Google Chrome	Gratuita
Visual Studio Code	Gratuita
Office 2016	Comercial

Tabla 6: Software para el desarrollo

Componente	Requisito
Procesador	CPU x86 a 1GHz con instrucciones SSE2
Memoria RAM	2 GB de RAM
Disco duro	3 GB de HDD
Tarjeta gráfica	compatible con DirectX 10
Sistema operativo	Windows 7 SP 1 o superior
Navegador	Microsoft Edge, Mozilla Firefox, Chrome 40, Internet Explorer 9 o superiores
Otros softwares	.NET 3.5 como mínimo

Tabla 7: Requisitos mínimos de hardware para el desarrollo

Desde el punto de vista del cliente, los requisitos de software y hardware, se limitan al navegador que utilice. Para esto, se utilizará como referencia al navegador más popular según la revista PCWorld (Mora, 2020), este es Google Chrome.

Componente	Requisito
Procesador	1.0 GHz
Memoria RAM	256 MB de RAM
Disco duro	600 MB de espacio libre
Tarjeta gráfica	Tarjeta compatible con DirectX 8.1 o superior y con 64 MB
Sistema operativo	Windows XP o superior
Sonido	Compatible con DirectX

Tabla 8: Requisitos de hardware para el cliente

Por lo tanto, se concluye que el proyecto es factible técnicamente, tanto desde la perspectiva del desarrollo, como desde la perspectiva de cliente.

5.2 Factibilidad operativa.

Debido a que el estado actual del proyecto no permite realizar pruebas, y no existen negociaciones con ninguna empresa en particular (lo que impide determinar algún nivel de interés), la viabilidad operativa se realizará desde un escenario hipotético.

Considerando que el sistema ofrece una buena usabilidad, tomando como guía las heurísticas de Nielsen (Nielsen, 1994), adaptación a los distintos dispositivos en que se utilice (computadores personales, teléfonos, tabletas) y que en la actualidad la mayoría de las personas está familiarizada, al menos a un nivel básico, con el uso de ordenadores de escritorio o de bolsillo (*smartphones* y *tablets*), el sistema debería ser utilizado completa y correctamente, tanto por la administración, como por los trabajadores y los clientes. Esta afirmación se justificaría por la correcta aplicación de las heurísticas de Nielsen, las que proveen guías generales para lograr una buena usabilidad, tales como un diseño minimalista, consistencia y estándares, prevención de errores, entre otras.

Dicho todo esto, parece razonable asumir que la factibilidad operativa sería satisfecha por el presente proyecto.

5.3 Factibilidad económica.

La factibilidad económica busca analizar la relación costo/beneficio de un proyecto, en este caso, la elaboración y posterior arriendo⁸ de un sistema de gestión para un negocio de lavado de autos. Para llevar a cabo este análisis, se consideró un horizonte de cinco años, la inversión en equipo, el costo de desarrollo del sistema, los costos fijos y operacionales, el capital de trabajo y una estimación de clientes por año. Con esta información se elaboró un flujo de caja, y se calculó el criterio de inversión VAN. A continuación, se presentarán las tablas involucradas, junto con las notas o aclaraciones pertinentes.

5.4 Inversión

Como costos de inversión, se consideró una licencia OEM de Office 2016, para elaborar toda la documentación necesaria utilizando software original y pagado; También se consideró la asesoría legal ("Redacción de contratos y documentos legales", 2020), pensando en la elaboración de un eventual contrato. Esto queda plasmado en la Tabla 9.

Licencia OEM Office 2016	\$ 27,360
Asesoría Legal	\$ 75,000
Total	\$ 102,360

Tabla 9: Inversión

⁸ Si bien el proyecto fue desarrollado para una empresa ficticia, la forma en que se desarrolló fue pensada para ser una aplicación que se adapte a este tipo de empresa, eso por esto que será posible ofrecer este sistema a más de un cliente.

5.5 Costo de desarrollo

La Tabla 10 muestra el costo de desarrollar el proyecto completo, para una persona. Para el desarrollo de esta aplicación, se consideró una jornada laboral de 40 horas a la semana, repartidas en cinco días de trabajo, durante tres meses.

Valor de hora/hombre de desarrollo	\$ 5,000	CLP
Tiempo necesario para el desarrollo (1 persona)	3	Meses (4 semanas c/u)
Semanas de trabajo	12	
Días de trabajo a la semana	5	
Horas diarias de trabajo	8	
Horas necesarias	480	
Costo del desarrollo	\$ 2,400,000	

Tabla 10: Costo de desarrollo

5.6 Costos fijos

Como costo fijo, Tabla 11, se consideró el pago de hosting, a modo de referencia, el plan de negocios del proveedor Hostgator. El plan de Negocios de Hostgator, ofrece 50GB de almacenamiento SSD, certificado SSL gratis, cPanel, sitios web y correos ilimitados, por lo que es una excelente opción para un emprendimiento como este.

Plan Negocios Anual (Hostgator)	\$ 50,328
---------------------------------	-----------

Tabla 11: Costos fijos

5.7 Costos de operación

Como parte de los costos de operación se consideró a la mantención y soporte. Dentro de dicho periodo de tiempo, también se pretende ir realizando mejoras o agregando funcionalidades. Para la mantención, Tabla 12, se supuso que se le dedicaría cuatro horas a la semana por dos semanas al mes. En el caso del soporte, Tabla 13, dos horas a la semana, en caso de que sea necesario.

Valor de hora/hombre de Mantención	\$ 5,000	CLP
Tiempo necesario para la mantención (1 persona)	4	horas a la semana
Semanas de trabajo (2 Semanas por mes)	24	
Costo de mantención anual	\$ 480,000	

Tabla 12: Costo de mantención

Valor de hora/hombre de Soporte	\$ 4,000	CLP
Tiempo dedicado al soporte (1 persona)	2	horas a la semana
Semanas de trabajo (4 Semanas por mes)	48	
Costo de soporte anual	\$ 384,000	

Tabla 13: Costo de soporte

5.8 Ingresos

En cuanto a los ingresos, debido a la naturaleza de la aplicación, esta será posible de vender e implementar en varias empresas, con o sin sitio web. Dicho esto, cabe señalar que en la ciudad de Chillán tienen presencia las empresas: EcoWashCar, Copec (Servicio LavaMax), El castillo Limpio, Pacific Car Wash, Autolavar, Autolavado SPA, Alasan Limitada, Servitodo, Lubricentro AutoLimpio, Auto Color, Auroflop y Water LYC. De estas, solo las tres primeras cuentan con un sitio web (únicamente informativo, no ofrece posibilidad de realizar reservas). A todas estas empresas también se podría añadir a los numerosos estacionamientos que hay en la ciudad, que también ofrecen servicio de lavado, por otra parte, muy probablemente también quedaron empresas fuera de este listado.

Además de lo anterior, se debe sumar la posibilidad de extender la venta del sistema más allá del nivel comunal, lo que aumentaría considerablemente la cuota de mercado.

Dicho esto, y estudiando la factibilidad económica con un horizonte de cinco años, cobrando 70 mil pesos mensuales, se espera tener dos clientes el primer año, y luego ir sumando uno más a cada año. En la Tabla 14 se puede ver el ingreso esperado producto de esto.

Período (anual)	1	2	3	4	5
Cientes previstos	2	3	4	5	6
Ingreso Anual	\$ 1,680,000	\$ 2,520,000	\$ 3,360,000	\$ 4,200,000	\$ 5,040,000

Tabla 14: Ingresos

5.9 Flujo de caja

La Tabla 15 presenta los datos agrupados en una sola, y es a partir de esta, en conjunto con la de ingresos (tabla 14) que se elabora el contenido de la Tabla 16, correspondiente al flujo de caja y VAN. Cabe destacar que, para este proyecto, y por fines de practicidad, se supuso una tasa de una tasa de descuento de 10% (Tabla 15), Que es el porcentaje genérico utilizado habitualmente.

Como se observa en la Tabla 16, el proyecto tiene un balance positivo en el flujo de caja, y un valor actual neto (VAN) de \$2.073.458,67 Millones de pesos. Todo esto indica que parece ser un proyecto rentable.

Horizonte del proyecto	5	años
Precio de arriendo mensual	\$ 70,000	CLP
Inversión	\$ 102,360	
Costo de desarrollo (considerado solo 1er año)	\$ 2,400,000	CLP
Costos fijos	\$ 50,328	Anual
Tasa de descuento	10%	

Tabla 15: Datos agrupados

FLUJO DE CAJA						
Período	0	1	2	3	4	5
(+) Ingreso por venta		\$ 1,680,000	\$ 2,520,000	\$ 3,360,000	\$ 4,200,000	\$ 5,040,000
(-) CV costos operacionales		-\$ 1,248,000	-\$ 1,632,000	-\$ 2,016,000	-\$ 2,400,000	-\$ 2,784,000
(-) Costos Fijos		-\$ 50,328	-\$ 50,328	-\$ 50,328	-\$ 50,328	-\$ 50,328
(-) Costos de desarrollo	-\$ 2,400,000					
(-) Inversión	-\$ 102,360					
Flujo neto	-\$ 2,502,360	\$ 381,672	\$ 837,672	\$ 1,293,672	\$ 1,749,672	\$ 2,205,672
VAN	2,073,458.67					

Tabla 16: Flujo de caja y VAN

5.10 Conclusión de la factibilidad

El estudio de la factibilidad técnica muestra las herramientas necesarias en contraste con las disponibles para el desarrollo del proyecto. La factibilidad operativa, si bien hipotética dadas las circunstancias⁹, es respaldada por la teoría. La factibilidad económica, producto del tipo de aplicación aquí desarrollada, indica que es viable económicamente. Por lo tanto, se concluye que el proyecto es factible a la luz de estos tres criterios.

⁹ Pandemia mundial provocada por el virus SARS-CoV-2.

6 ANÁLISIS

6.1 Modelo de proceso de negocio

Para describir el proceso de negocio involucrado se hará uso de la notación de modelo procesos de negocio, o BPMN por sus siglas en inglés:

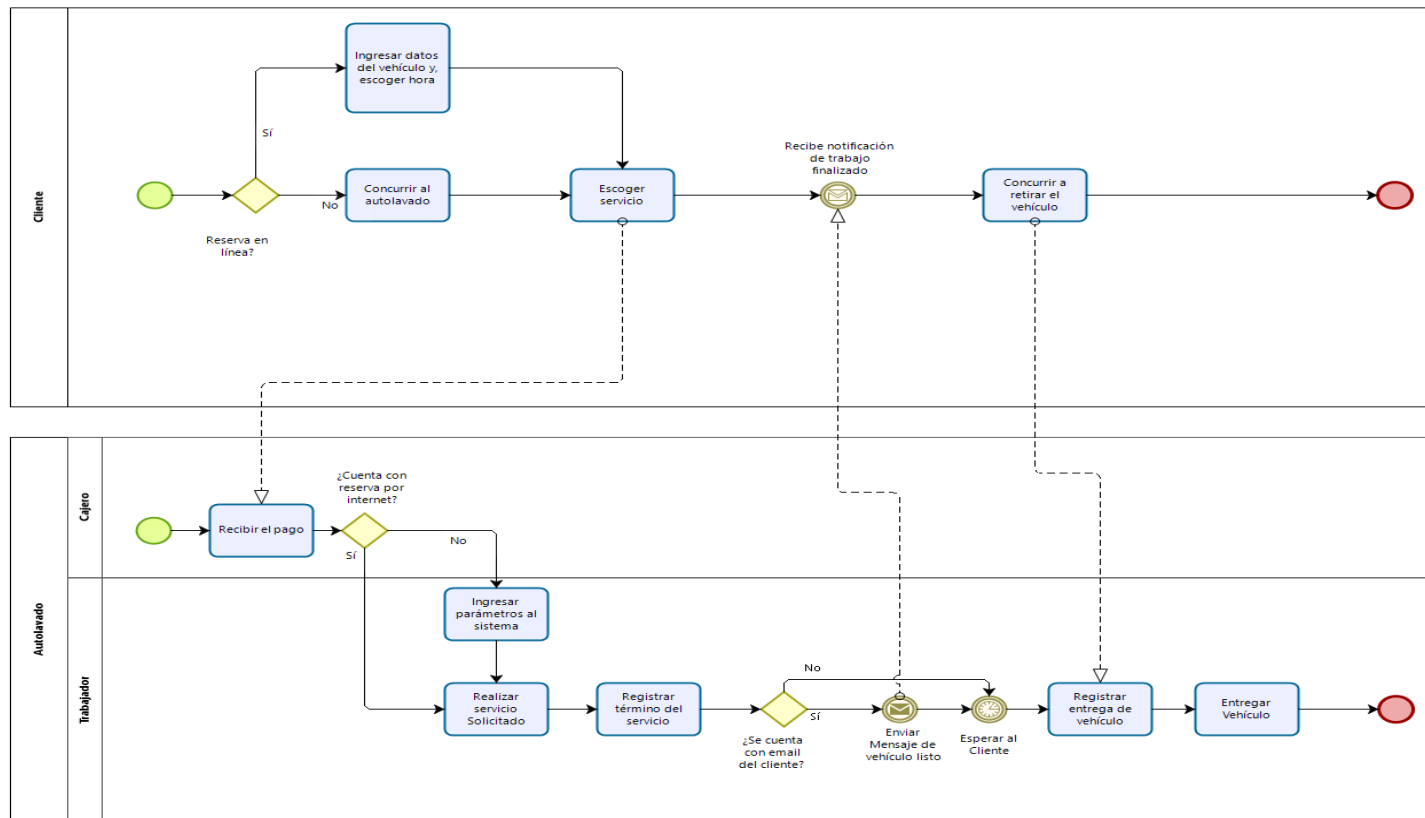


Figura 2: BPMN

6.2 Descripción del modelo de proceso de negocio

El modelo de proceso de negocios contempla dos casos: El cliente solicita un servicio mediante reserva, y aquel que lo hace sin una de por medio. A su vez, éste último caso con contiene una pequeña subdivisión, determinada por el hecho de que el cliente entregue al trabajador su dirección de correo electrónico para recibir una notificación una vez que el servicio de limpieza se haya completado.

En el primero de los casos, el cliente selecciona un servicio en el sitio web, ingresa los datos requeridos del vehículo, escoge una hora y agenda su reserva. Hecho esto, el cliente concurre a la empresa el día y hora de la reserva, paga¹⁰, un trabajador recibe su vehículo y registra el ingreso, éste ejecuta el servicio solicitado. El trabajador registra una vez que se ha finalizado el servicio, así, el sistema notifica esto de manera automática al cliente, a través de correo electrónico. El cliente concurre a buscar su vehículo, algún trabajador registra la salida del vehículo y se lo entrega al cliente.

En el segundo caso, un cliente concurre directamente a la empresa de lavado, sin una reserva previa, escoge el servicio que desea solicitar, un trabajador registra el servicio solicitado, los datos del vehículo y ofrece al cliente la posibilidad de entregar una dirección de correo electrónico para recibir una notificación cuando el servicio se haya completado. El servicio de limpieza es ejecutado, una vez finalizado, el trabajador registra el término del servicio, en caso de que el cliente haya entregado alguna dirección de correo electrónico, éste recibe una notificación de que el servicio se ha completado, y que puede ir a retirar su vehículo. En el caso de que el cliente no haya entregado un correo electrónico, el trabajador le indica al cliente una hora estimada para que vaya a retirar su vehículo. En ambos casos, se espera a que el cliente concurra a retirar su vehículo, una vez que éste llega y paga, un trabajador registra la salida del vehículo y hace entrega de éste al cliente.

¹⁰ Cabe destacar que, si bien el pago se realiza una vez que el servicio ya se ha realizado, no existe ningún inconveniente con ubicarlo antes de que dicho servicio se realice, es decir, si el cliente deseara hacer el cobro al comienzo, el funcionamiento del sistema no se vería afectado.

6.3 Diagrama de casos de uso

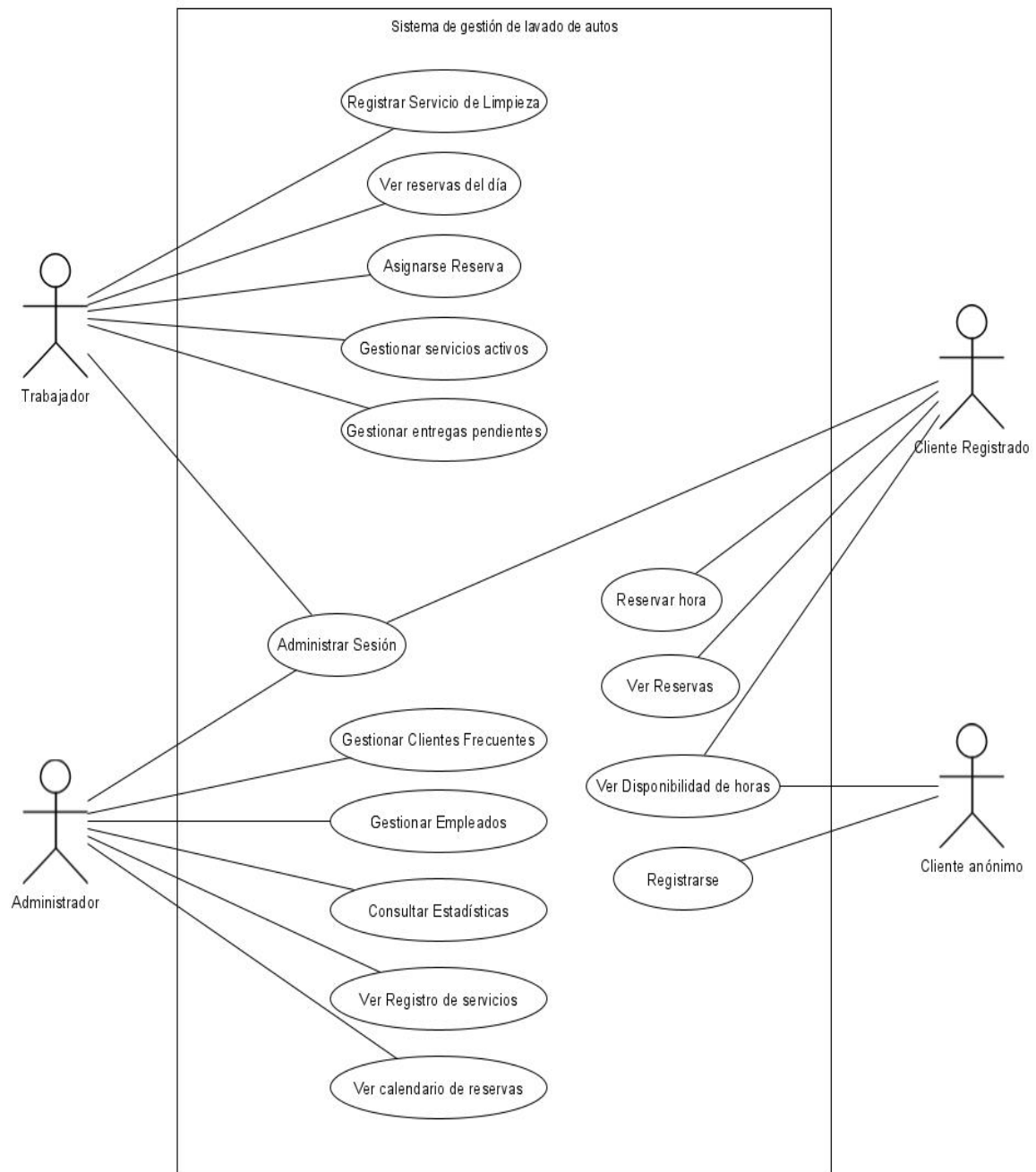


Figura 3: Diagrama de casos de uso

6.3.1 Actores

Administrador: Este actor es el encargado de toda la gestión del negocio. Él es el encargado de agregar, ver, editar y desvincular trabajadores; de ver y editar la información de los clientes; de consultar las estadísticas (de cara a tomar decisiones respecto al rumbo del negocio).

Trabajador: El trabajador es el encargado de ver y asignarse las reservas del día, así como también de registrar los servicios realizados que no cuenten con una reserva previa. Además de esto, también es el encargado de entregar los vehículos cuyo servicio de limpieza haya finalizado, y registrar esto.

Cliente registrado: El cliente registrado es un actor cuya información está almacenada dentro del sistema, él puede ver la disponibilidad de reservas de la semana en curso, puede realizar una reserva y también ver sus reservas agendadas.

Cliente anónimo: Al igual que el cliente registrado, el cliente anónimo puede ver la disponibilidad de reservas de la semana en curso, sin embargo, éste no puede realizar reservas. A pesar de esto, el cliente anónimo tiene la posibilidad de registrarse en el sistema.

6.3.2 Especificación de los Casos de Uso

Caso de Uso	Administrar Sesión
ID	1
Descripción	Este caso de uso se refiere a todo el proceso de administración de sesión, es decir, al inicio y cierre de sesión, del administrador, empleado o cliente.
Actores Principales	Administrador, empleado y cliente
Actores Secundarios	-
Precondiciones	-
Flujo de Eventos	
Actor	Sistema
<p>1. El administrador, empleado o cliente ingresan a su URL correspondiente, e introducen sus credenciales.</p> <p>3. El administrador, empleado o cliente seleccionan la opción de cerrar sesión.</p>	<p>2. El sistema valida dichas credenciales y permite el ingreso al perfil.</p> <p>4. El sistema cierra la sesión correspondiente y redirige a la ventana de acceso.</p>
Flujo alternativo	
	2(a) El sistema rechaza las credenciales ingresadas, muestra un aviso en pantalla y mantiene al usuario en su pantalla de login.

Tabla 17: Caso de uso - Administrar Sesión

Caso de Uso	Registrar Servicio de Limpieza
ID	2
Descripción	El trabajador registra todos los datos del vehículo y su dueño, así como también el servicio de limpieza seleccionado
Actores Principales	Trabajador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de eventos	
Actor	Sistema
1. El trabajador ingresa los datos del vehículo, del cliente y del servicio seleccionado.	<p>2. El sistema valida que estén los datos necesarios, y almacena los datos en la base de datos, junto con la fecha y hora.</p> <p>3. El sistema muestra un mensaje de registro ingresado correctamente.</p>

Tabla 18: Caso de uso - Registrar Servicio de Limpieza

Caso de Uso	Ver Reservas del día
ID	3
Descripción	El sistema muestra al trabajador una tabla con todas las reservas agendadas para el día en curso.
Actores Principales	Trabajador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
1. El trabajador ingresa a la página de "Ver reservas"	2. El sistema genera y muestra una tabla con las reservas del día.

Tabla 19: Caso de uso - Ver Reservas del día

Caso de Uso	Asignarse Reserva
ID	4
Descripción	El trabajador escoge una reserva de la tabla de reservas del día, y se la asigna a sí mismo (lo que lo asocia con dicho servicio)
Actores Principales	Trabajador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
1. Dentro de la página de “Ver reservas”, el trabajador escoge una reserva y se la asigna.	2. El sistema crea un registro con los detalles del servicio a llevar a cabo asociado al trabajador. 3. El sistema muestra un mensaje de asociación exitosa.

Tabla 20: Caso de uso - Asignarse Reserva

Caso de Uso	Gestionar Servicios activos
ID	5
Descripción	El sistema permite ver los detalles de las reservas activas asociadas al trabajador, así como también registrar el término de dicha actividad. Además, también permite a un trabajador asociar a otro trabajador (en el caso de que el servicio requiera del trabajo de más de una persona)
Actores Principales	Trabajador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El trabajador ingresa a la página "Servicios activos"</p> <p>3. El trabajador selecciona la opción de registrar el término de servicio.</p>	<p>2. El sistema genera y muestra una tabla con todos los servicios activos asociados al trabajador.</p> <p>4. El sistema registra la hora del término de servicio, actualiza el estado del registro de servicio y envía un email de notificación al cliente, en caso de haber alguno asociado.</p> <p>5. El sistema muestra un mensaje de acción exitosa.</p>
Flujo Alternativo	
3(a). El trabajador asocia a otro trabajador a uno de sus servicios activos.	4(a). El sistema asocia a un trabajador adicional al registro del servicio y muestra un mensaje de asociación exitosa.

Tabla 21: Caso de uso - Gestionar Servicios activos

Caso de Uso	Gestionar entregas pendientes
ID	6
Descripción	El sistema permite ver a todos los trabajadores, los detalles de todos los servicios finalizados, así como también registrar la entrega del vehículo.
Actores Principales	Trabajador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El trabajador ingresa a la página “Entregas pendientes”</p> <p>3. El trabajador selecciona la opción de registrar entrega de vehículo.</p>	<p>2. El sistema genera y muestra una tabla con todos los servicios que hayan sido registrados como terminados.</p> <p>4. El sistema registra la hora de entrega, el nombre del trabajador que entregó y actualiza el estado del registro de servicio.</p> <p>5. El sistema muestra un mensaje de acción exitosa.</p>

Tabla 22: Caso de uso - Gestionar entregas pendientes

Caso de Uso	Gestionar Cliente registrado
ID	7
Descripción	El sistema permite ver detalles, editar o eliminar a un cliente registrado
Actores Principales	Administrador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El administrador ingresa a la página de gestión de clientes frecuentes.</p> <p>3. El administrador selecciona la opción que desee realizar (crear, editar o eliminar).</p> <p>5. El administrador llena o edita el formulario con los datos solicitados.</p>	<p>2. El sistema muestra la lista de clientes frecuentes, y su información general.</p> <p>4. El sistema muestra un formulario para agregar o editar a un cliente registrado. En caso de eliminación, muestra un mensaje de confirmación.</p> <p>6. El sistema valida que el formulario esté llenado correctamente.</p> <p>7. El sistema ingresa o edita al cliente registrado (muestra un mensaje de confirmación en el caso de la edición).</p> <p>8. El sistema muestra un mensaje de éxito.</p>

Tabla 23: Caso de uso - Gestionar Cliente registrado

Caso de Uso	Gestionar Empleados
ID	8
Descripción	El sistema permite añadir, editar o eliminar a un empleado
Actores Principales	Administrador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El administrador ingresa a la página de gestión de empleados.</p> <p>3. El administrador selecciona la opción que desee realizar (crear, editar o eliminar).</p> <p>5. El administrador llena o edita el formulario con los datos solicitados.</p>	<p>2. El sistema muestra la lista de empleados, y su información general.</p> <p>4. El sistema muestra un formulario para agregar o editar a un empleado. En caso de eliminación, muestra un mensaje de confirmación.</p> <p>6. El sistema valida que el formulario esté llenado correctamente.</p> <p>7. El sistema ingresa o edita al empleado (muestra un mensaje de confirmación en el caso de la edición).</p> <p>8. El sistema muestra un mensaje de éxito.</p>

Tabla 24: Caso de uso - Gestionar Empleados

Caso de Uso	Consultar Estadísticas
ID	9
Descripción	El sistema muestra al administrador un informe con estadísticas relevantes, en un intervalo establecido de tiempo
Actores Principales	Administrador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El administrador ingresa a la página de estadísticas.</p> <p>3. El administrador selecciona el reporte deseado.</p> <p>5. El administrador ingresas las fechas.</p>	<p>2. El sistema muestra las opciones de reportes a solicitar.</p> <p>4. El sistema solicita una fecha de inicio y una de fin, que determine el intervalo de tiempo.</p> <p>6. El sistema genera el reporte solicitado, para el intervalo de tiempo establecido.</p>

Tabla 25: Caso de uso - Consultar Estadísticas

Caso de Uso	Ver registro de servicios
ID	10
Descripción	El sistema permite ver la totalidad de los registros de servicio.
Actores Principales	Administrador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
1. El administrador ingresa a la página de registro de servicios.	2. El sistema muestra la lista de todos los registros de servicios y sus detalles.

Tabla 26: Caso de uso - Ver registro de servicios

Caso de Uso	Ver calendario de reservas
ID	11
Descripción	El sistema permite añadir, editar o eliminar a un empleado
Actores Principales	Administrador
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
1. El administrador ingresa a la página de resumen.	2. El sistema muestra un calendario semanal con las reservas agendadas.

Tabla 27: Caso de uso - Ver calendario de reservas

Caso de Uso	Reservar hora
ID	12
Descripción	El sistema permite a un cliente registrado realizar una reserva de hora para el lavado de auto.
Actores Principales	Cliente registrado
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión
Flujo de Eventos	
Actor	Sistema
<p>1. El cliente selecciona la opción de reserva.</p> <p>3. El cliente selecciona los parámetros de la reserva, la fecha/hora deseada y solicita su reserva.</p>	<p>2. El sistema muestra un formulario para escoger los detalles del servicio y la fecha/hora de la reserva.</p> <p>4. El sistema valida que la fecha/hora no se encuentre ocupada y realiza la reserva.</p> <p>7. El sistema cambia el estado de la hora a “no disponible” y muestra un mensaje de reserva exitosa.</p>

Tabla 28: Caso de uso - Reservar hora

Caso de Uso	Ver Reservas
ID	13
Descripción	El sistema permite a un cliente registrado ver sus reservas ya agendadas.
Actores Principales	Cliente registrado
Actores Secundarios	-
Precondiciones	Haber hecho inicio de sesión, Tener una hora reservada
Flujo de Eventos	
Actor	Sistema
<p>1. El cliente selecciona la opción de ver reservas.</p> <p>3. El cliente escoge la reserva que desea y selecciona la opción de ver detalles.</p>	<p>2. El sistema muestra la lista con sus reservaciones.</p> <p>4. El sistema muestra un mensaje con los detalles de la reserva.</p>

Tabla 29: Caso uso - Gestionar Reservas

Caso de Uso	Ver disponibilidad de horas
ID	14
Descripción	Permite al cliente ver las horas disponibles y reservadas
Actores Principales	Cliente registrado y cliente anónimo
Actores Secundarios	-
Precondiciones	-
Flujo de Eventos	
Actor	Sistema
1. El cliente ingresa al sistema.	2. El sistema muestra un calendario de la semana en curso con las reservas agendadas.

Tabla 30: Caso de uso - Ver disponibilidad de horas

Caso de Uso	Registrarse
ID	15
Descripción	Permite al cliente anónimo registrarse dentro del sistema
Actores Principales	Cliente anónimo
Actores Secundarios	-
Precondiciones	-
Flujo de Eventos	
Actor	Sistema
<p>1. El cliente ingresa al sistema y selecciona la opción para registrarse.</p> <p>3. El cliente ingresa los datos requeridos.</p>	<p>2. El sistema muestra un formulario con los datos requeridos.</p> <p>4. El sistema valida los datos, crea un registro y muestra un mensaje de éxito.</p>

Tabla 31: Caso de uso – Registrarse

6.4 Especificación de las entidades

A continuación, se presenta una breve descripción de las entidades y relaciones del modelo entidad relación antes presentado.

6.4.1 Descripción de entidades

- **Agenda:** Entidad donde se registra cada una de las reservas del día.
- **RegistroServicio:** Es la entidad encargada de registrar todos los datos asociados a la realización de un servicio de limpieza, similar a una boleta.
- **Reserva:** Corresponde a la entidad que almacena los datos necesarios de una reserva.
- **Servicio:** Entidad que representa algún servicio en particular ofrecido por la empresa.
- **Trabajador:** Representa a aquel trabajador que ejecuta el servicio de limpieza, y es el encargado de recibir y registrar el servicio realizado.
- **Usuario:** Es el cliente que solicita una reserva.

6.4.2 Descripción de relaciones

- **Almacena:** Relación que vincula a “Reserva” con “Agenda”. La entidad agenda es la encargada de almacenar cada una de las reservas existentes, además, también sirve como instrumento de consulta para saber si existe disponibilidad de reservas.
- **Asocia:** Relación que vincula a “Trabajador” con “RegistroServicio”. Cada “RegistroServicio” debe tener asociado a al menos un trabajador, quien es quien realizó un determinado servicio.
- **Realiza:** Relación que vincula a “Usuario” con “Reserva”. Señala el acto de solicitar una reserva por parte de algún cliente, vinculando a dicha reserva con él.
- **Registra:** Relación que vincula tanto “Reserva” con “RegistroServicio”, como a “Servicio” con “Reserva”. Esta relación señala la asociación referencial entre las entidades señaladas, concretamente, una reserva contiene el registro de servicio, y “RegistroServicio” contiene los datos de reserva.

6.5 Modelamiento de datos

En este último apartado de la sección de análisis, en la Figura 4, se presenta el modelo entidad relación de los datos del sistema.

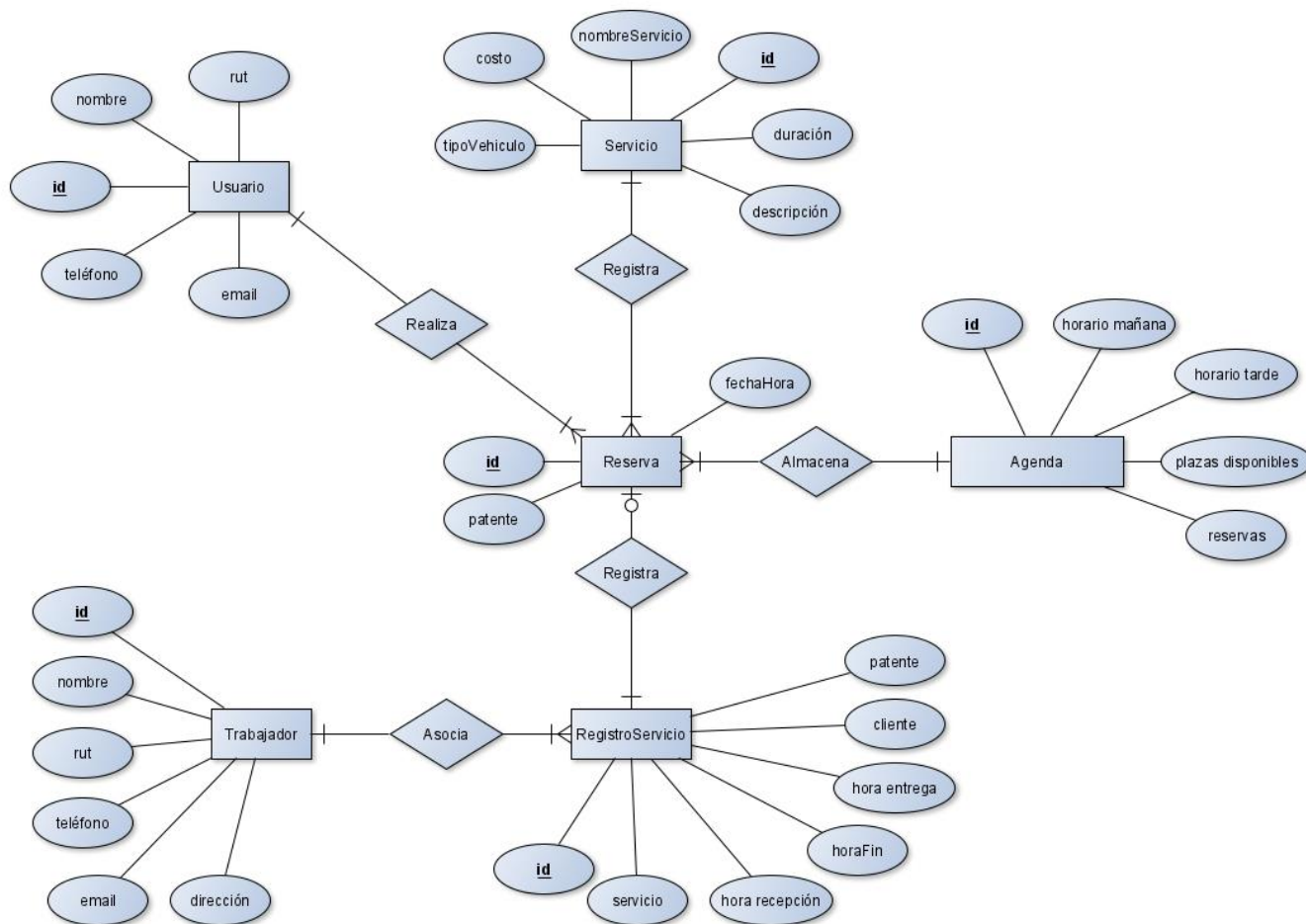


Figura 4: Modelo entidad relación

7 DISEÑO

7.1 Diseño de Físico de la Base de datos

El diseño físico de la base de dato es una parte fundamental en el desarrollo de una aplicación, ya que, si bien el modelo entidad relación ofrece un buen modelo conceptual acerca de la interacción de los mismos, el diseño físico muestra la manera específica y concreta de cómo estos se almacenan e interactúan. Como sistema de base de datos se utilizó *MongoDB*, sistema de base de datos NoSQL¹¹.

En cuanto a diseño, como señala Akshay Pore (Pore, 2019) este se debe realizar guiado por la forma en que se consultarán los datos, más que teniendo en mente la optimización del espacio de almacenamiento que estos utilizarán, de esta forma, el diseño de las vistas está estrechamente relacionado con el diseño de la base de datos. Así, según se señala en la documentación oficial ("Data Modeling Introduction", 2020), el desafío clave en el modelado de datos es equilibrar las necesidades de la aplicación, las características de rendimiento del motor de la base de datos y los patrones de recuperación de datos.

Si se quiere, a modo de ayuda comparativa, se puede entender a las colecciones como el análogo de las tablas de una base de datos SQL, y a los documentos como el análogo de las filas.

Dicho esto, en la Figura 5 se muestra el diseño físico de la base de datos, aquí se muestra el conjunto de todas las colecciones con sus respectivos documentos.

¹¹ No está de sobra volver a señalar que NoSQL significa *not Only SQL*, lo que indica que no es exclusivamente relacional, es decir, se cuenta con la posibilidad de establecer relaciones, solo que esto es opcional.

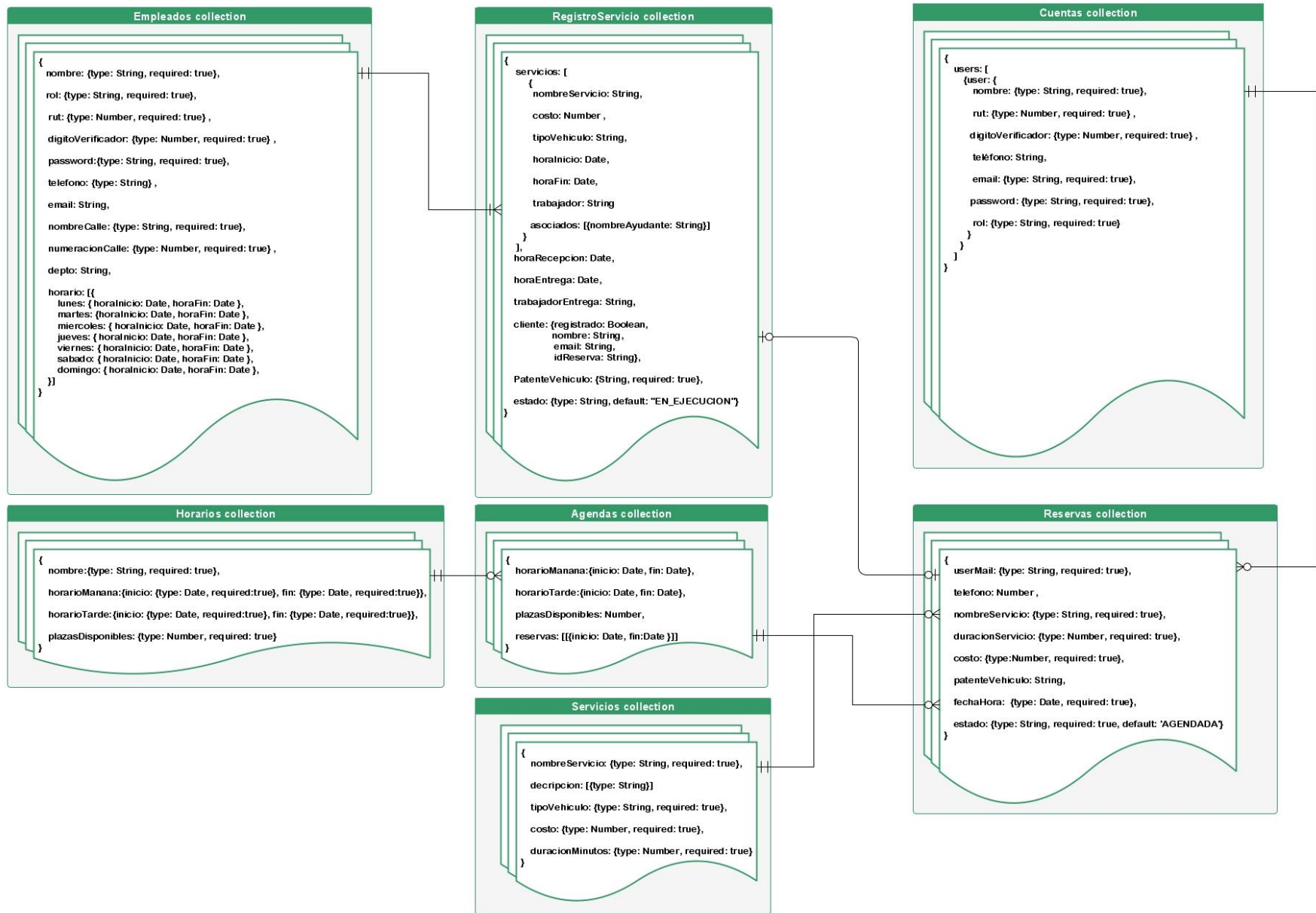


Figura 5: Modelo físico de la base de datos

7.2 Diseño de arquitectura

En este apartado se describen los aspectos más importantes y significativos del diseño de arquitectura utilizados en el desarrollo del proyecto.

7.2.1 MVC

Como patrón de arquitectura de software se utilizó Modelo-Vista-Controlador (MVC) que, según el sitio de la universidad de Alicante (Alicante, 2020) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

A continuación, en la Figura 6 se presenta un diagrama del funcionamiento de MVC. En el caso de la presente aplicación, ReactJs se encarga de renderizar las vistas, y NodeJs en conjunto con Express y Mongoose son los encargados de gestionar los procesos de los controladores y modelos. Un ejemplo de esto sería: El administrador ingresa al apartado de clientes, React realiza una petición GET al endpoint de la API Rest del servidor, el servidor recibe dicha petición y es gestionada por el controlador, quién a su vez, a través del modelo (mediante Mongoose) solicita la información a la base de datos. La base de datos responde al modelo, que le responde al controlador, que le responde a la vista, la que finalmente responde visualmente al usuario.

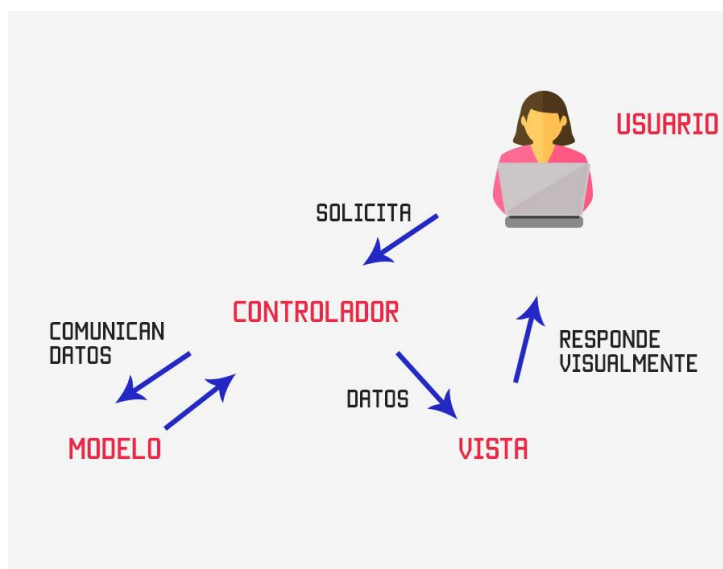


Figura 6: Diagrama del Modelo-Vista-Controlador

7.2.2 REST

REST, o *representational state transfer* (transferencia de estado representacional) por sus siglas en inglés, en palabras sencillas, “es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.” (BBVAOpen4U, 2016).

En este caso, REST está implementado mediante el framework Express, concretamente, en forma de una API REST, que es la encargada de gestionar las llamadas HTTP hechas desde el lado del cliente. De esta manera es cómo se gestiona la comunicación entre la vista, que interactúa con el usuario, y el controlador, que interactúa con el modelo.

A continuación, en las Tablas 8 y 9, se muestran las peticiones agregar empleado y listar empleados, que corresponden a un método POST y a un método GET respectivamente. En dichas tablas se especifica el *Endpoint* de la petición (en un servidor local), el método, los parámetros y un ejemplo de respuesta¹²

¹² Cabe destacar el uso de paréntesis con puntos suspensivos, en el caso de la respuesta del segundo ejemplo, es para indicar que los datos siguen, solo que se omiten para no saturar la tabla con demasiado texto.

Acción	Agregar Empleado
Endpoint	http://localhost:3000/api/empleado/agregar-empleado
Método	POST
Parámetros	<pre>{ "nombre": "trabajador 1", "rut": "17516749", "digitoVerificador": "8", "password": "123Password", "telefono": "+56995905135", "email": "correo@mail.com", "nombreCalle": "Av. Chile", "numeracionCalle": "123", "depto": "D", }</pre>
Respuesta	<pre>{ "ok": true, "empleado": { "_id": "5f4f35bd1f664642fc1b5f5f", "nombre": "trabajador 1", "rut": 17516749, "digitoVerificador": 8, "password": "123Password", "telefono": "+56995905135", "email": "asd@asd.com", "nombreCalle": "Av. Chile", "numeracionCalle": 123, "depto": "D", "__v": 0 } }</pre>

Tabla 32: Ejemplo de petición POST

Acción	Listar Empleados
Endpoint	http://localhost:3000/api/empleado/lista-empleados
Método	GET
Parámetros	---
Respuesta	<pre>{ "ok": true, "empleados": [{ "_id": "5f069d0b2b608a38c40cf583", "nombre": "trabajador 1", "rut": 17516749, "digitoVerificador": 8, "password": "123Password", "telefono": "+56995905135", "email": "correo@mail.com", "nombreCalle": "Av. Chile", "numeracionCalle": 123, "depto": "D", "__v": 0 }, { "_id": "5f153ebed2bacf0fe8a82984", "nombre": "trabajador 2", (...) }, (...)] }</pre>

Tabla 33: Ejemplo de petición GET

7.2.3 Redux

En lo que respecta al lado del cliente, a la vista, se implementa patrón y librería Redux para el manejo de estados de la misma.

La implementación de este patrón de diseño facilita el control del estado de la información en la aplicación, cuestión de mucha utilidad en aplicaciones de mediano o gran tamaño. Fernando Herrera (Herrera, 2018) ofrece una descripción bastante clara de los conceptos fundamentales de Redux:

1. Toda la data de la aplicación se encuentra en una estructura previamente DEFINIDA
2. Toda la información se encontrará almacenada en un único lugar llamado STORE
3. El store JAMÁS se modifica de forma directa
4. Las interacciones de usuario y/o código, disparan acciones que describen qué sucedió
5. El valor actual de la información de la aplicación se llama estado - STATE
6. Un nuevo estado es creado, en base a la combinación del viejo estado y una acción por una función llamada REDUCER

Para controlar el manejo de estado, y poder llevar a cabo estos conceptos fundamentales, Redux utiliza los conceptos de *Action*, *Reducer*, *State* y *Store*, que implementados se describen como:

- *Action*: Es un evento que describe lo que se va a hacer
- *Reducer*: Es una función que toma un estado anterior y una acción, y produce un nuevo estado.
- *State*: Es como se encuentra la información actualmente.
- *Store*: Es el encargado de gestionar todo. Contiene el estado de la aplicación

A continuación, en la Figura 7 (Oak Tree Software, 2020), se puede apreciar un diagrama que representa el funcionamiento de Redux, y que ilustra todo lo que se ha descrito.

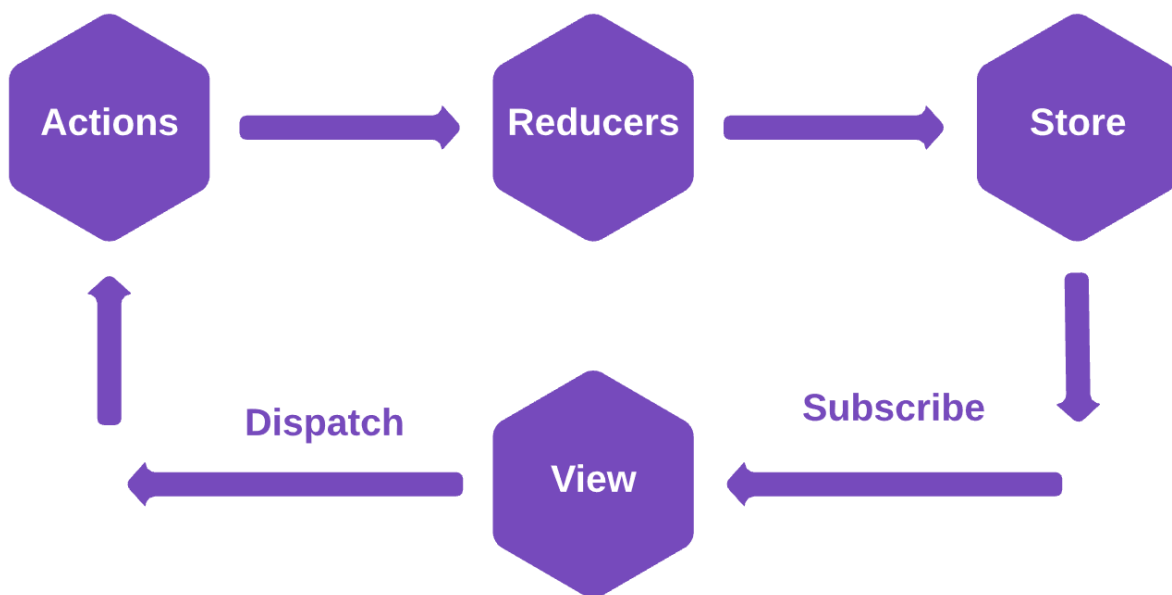


Figura 7: Diagrama Redux

7.3 Diseño interfaz y navegación

7.3.1 Navegación

En las Figuras 8, 9 y 10 se muestran los mapas de navegación del cliente, del trabajador y del administrador, respectivamente.

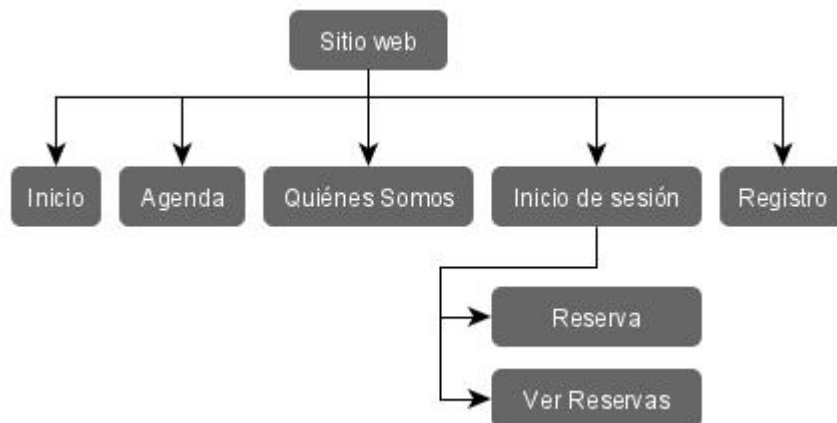


Figura 8: Mapa de navegación del cliente



Figura 9: Mapa de navegación del trabajador

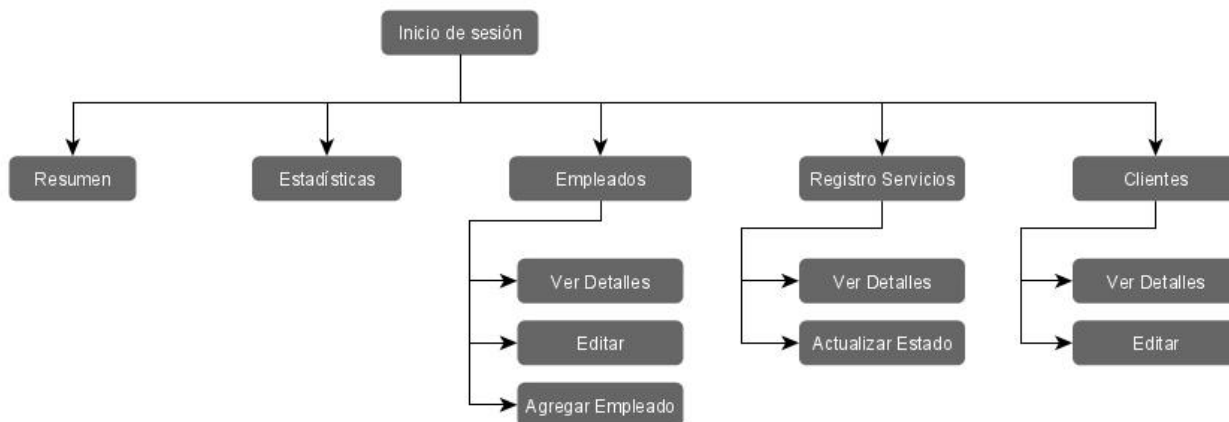


Figura 10: Mapa de navegación del administrador

7.3.2 Interfaz

A continuación, se presentará de manera agrupada y junto a una breve descripción, el diseño de las interfaces de cada perfil.

7.3.2.1 Cliente

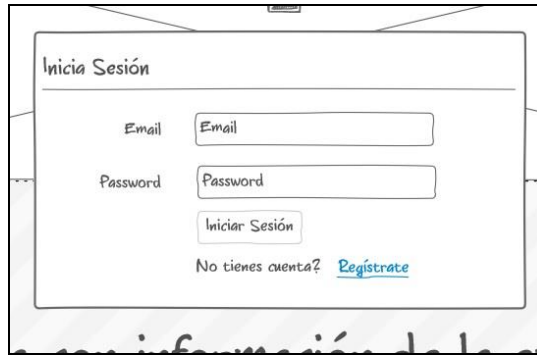
Como ya se ha anticipado en el desarrollo de este proyecto, la barra de navegación del cliente incluye el logotipo de la empresa, un botón de registro o inicio de sesión, que despliega un modal con un formulario de ingreso y, alternativamente, de registro (Ver Figuras 12 y 13), y las opciones de navegación “Inicio”, “Agenda” y “Quiénes Somos”. Además de la barra de navegación, la interfaz del cliente también cuenta con un footer que contiene información básica y de utilidad acerca de la empresa (ubicación, contacto, horario de atención y horario de reserva), además del logotipo y enlaces a redes sociales. Por último, también cabe destacar la presencia de un botón flotante en la esquina inferior derecha, con la palabra “Reserva”. Tal como su nombre lo indica, este botón se utiliza para generar una nueva reserva y, de manera alternativa para el caso del cliente anónimo, señalarle lo fácil que es solicitar una reserva e incentivarlo a registrarse. Estos tres elementos son comunes a todas las páginas de la interfaz del cliente (anónimo y registrado).

Respecto al contenido de cada una de las páginas (fuera de los dos elementos recién mencionados), cabe señalar:

- “Inicio”, en la Figura 11, está dividido en tres secciones: Portada, que muestra una imagen alusiva al negocio, que también puede ser un carrusel de imágenes, y permitiría mostrar promociones u ofrecer algún otro tipo de información; Área con información de la empresa, que está pensada para ofrecer al cliente información breve y relevante que se desee desatacar acerca de la empresa; Área de nuestros servicios, que está pensada para mostrar todos los servicios ofrecidos por la empresa, así como también el detalle de cada uno.
- “Agenda”, en la Figura 14, es una página de una única sección, cuya finalidad es mostrar un calendario con la agenda de la semana, que es donde se muestran las reservas agendadas.
- Finalmente, en la Figura 15, “Quiénes Somos”, es una página meramente informativa, que contiene una imagen alusiva al negocio, acompañada de un texto con más detalles del mismo.



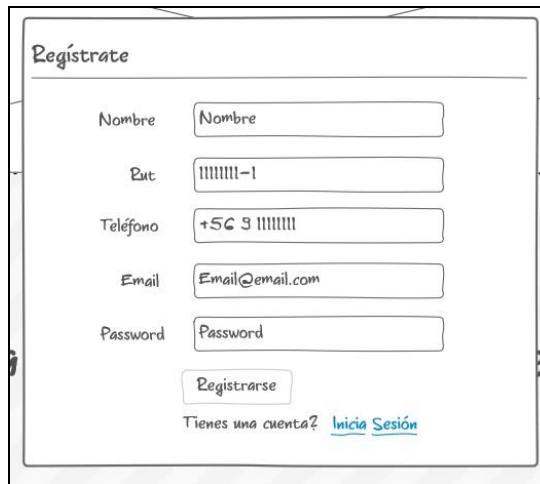
Figura 11: Interfaz del cliente - Inicio



Formulario de inicio de sesión del cliente. El formulario tiene un título "Inicia Sesión" y contiene los siguientes elementos:

- Campo "Email" con el texto "Email" dentro del cuadro de entrada.
- Campo "Password" con el texto "Password" dentro del cuadro de entrada.
- Botón "Iniciar Sesión".
- Enlace "No tienes cuenta? [Regístrate](#)".

Figura 12: Formulario de inicio de sesión del cliente



Formulario de registro del cliente. El formulario tiene un título "Regístrate" y contiene los siguientes elementos:

- Campo "Nombre" con el texto "Nombre" dentro del cuadro de entrada.
- Campo "Eut" con el texto "||||||| -1" dentro del cuadro de entrada.
- Campo "Teléfono" con el texto "+56 9 |||||" dentro del cuadro de entrada.
- Campo "Email" con el texto "Email@email.com" dentro del cuadro de entrada.
- Campo "Password" con el texto "Password" dentro del cuadro de entrada.
- Botón "Registrarse".
- Enlace "Tienes una cuenta? [Inicia Sesión](#)".

Figura 13: Formulario de registro del cliente

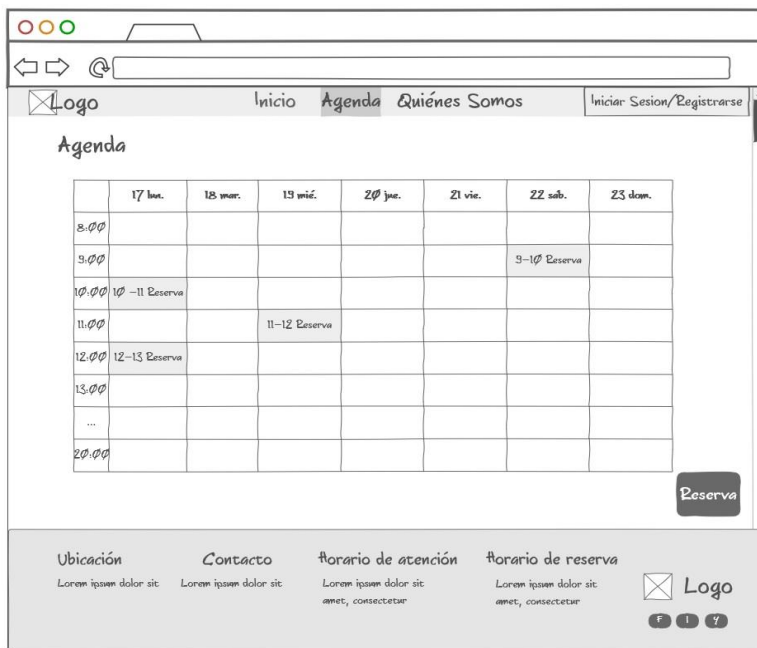


Figura 14: Interfaz del cliente - Agenda



Figura 15: Interfaz del cliente - Quiénes Somos

7.3.2.2 Trabajador

La interfaz del trabajador, por su parte, está pensada para ser vista desde un dispositivo móvil. Además de esto, esta interfaz está pensada para ser lo más simple y fácil de utilizar posible.

La interfaz del trabajador cuenta con un elemento común a todas sus páginas, el navbar, que contiene el logotipo de la empresa y un botón para cerrar la sesión.

Fuera de este elemento común, la interfaz del trabajador cuenta con una página inicial (Figura 16), que contiene cuatro botones que conducen a cada una de las cuatro páginas que contienen las funcionalidades relativas al trabajador, estas son:

- “Ver Reservas”, Figura 17, que muestra una tabla con las reservas del día, que contiene el nombre del servicio a realizar, la hora y las acciones que el trabajador puede llevar a cabo. Estas acciones son: Ver detalles¹³ y asignarse el trabajo. Cada uno representado por un icono alusivo.
- “Servicios Activos”, Figura 18, muestra una tabla con los servicios que el trabajador está llevando a cabo, y las siguientes acciones: Ver detalles, finalizar servicio, agregar trabajador.
- “Entregas Pendientes”, Figura 19, muestra una tabla con los servicios que ya se han finalizados, y cuya entrega de vehículo está pendiente. las siguientes acciones disponibles son: Ver detalles y Entregar vehículo.
- “Registrar Servicios”, Figura 20, muestra un formulario que permite al trabajador registrar un servicio que no cuente con una reserva previa, donde se ingresan los datos relevantes de registro. Además, también ofrece la posibilidad de ingresar un correo electrónico para notificar al cliente cuando el servicio solicitado se haya completado.

¹³ La acción ver detalles está presente en todas las tablas que se muestran en la interfaz del trabajador, esto porque, debido al tamaño de la pantalla del dispositivo en que se visualiza la interfaz, no es posible (ni necesario) mostrar todos los datos en columnas de la tabla de manera fácil y rápida de leer.



Figura 16: Interfaz del trabajador - Inicio



Figura 18: Interfaz del trabajador - Servicios Activos



Figura 17: Interfaz del trabajador - Ver Reservas



Figura 19: Interfaz del trabajador - Entregas Pendientes

The image shows a mobile application interface for registering a service. At the top, there is a header with a close button (X), a 'Logo' button, and a 'Salir' button. Below the header is a title bar with a back arrow and the text 'Registrar Servicio'. The main form contains several sections: 'Tipo de Vehículo' with a dropdown menu 'Seleccione tipo de vehículo...'; 'Servicio' with a dropdown menu 'Escoja Servicio ...'; 'Costo' with a text input field containing '\$\$\$'; 'Duración Aprox.' with a text input field containing 'XX Minutos'; 'Nombre Cliente' with a text input field containing 'Escribe el nombre del Cliente...'; 'Patente' with a text input field containing 'Escribe la patente del vehículo...'; and 'Correo electrónico (opcional)' with a text input field containing 'correo@ejemplo.com'. At the bottom of the form is a button labeled 'Agregar'.

Figura 20: Interfaz del trabajador - Registrar Servicio

7.3.2.3 Administrador

La interfaz del administrador está pensada para ser vista principalmente desde un computador, sin embargo, igualmente ésta cuenta con un sidebar colapsable para facilitar su visualización en dispositivos móviles.

Los elementos comunes que tiene esta interfaz son el navbar, que contiene un mensaje de bienvenida y un botón para cerrar la sesión; y el sidebar, que contiene el logo de la empresa, un mensaje con el área de quién visualiza la interfaz (administración) y un menú vertical de navegación.

Las páginas contenidas por el menú de navegación son:

- “Resumen”, Figura 21, que muestra el horario de atención, el horario de reservas y un calendario semanal navegable que muestra las reservas agendadas.
- “Estadísticas”, Figura 22, muestra dos gráficos con estadísticas relevantes para la administración, y cuyos intervalos de tiempo son seleccionables, estos son: “Demanda por servicio” y “Reservas v/s Presencial”.
- “Trabajadores”, Figura 23, muestra una tabla con todos los trabajadores contratados, y las siguientes acciones para cada fila (representadas por íconos alusivos): Ver detalles¹⁴ y editar¹⁵. Además, esta página también cuenta con un botón para agregar a un nuevo trabajador.
- “Registro servicios”, Figura 24, muestra una tabla con todos los registros de servicios realizados y las siguientes acciones para cada fila: Ver detalles y actualizar estado.
- “Clientes”, Figura 25, muestra una tabla con todos clientes registrados y las siguientes acciones para cada fila: Ver detalles y editar.

¹⁴ De manera similar a como ocurre con la interfaz del trabajador, el botón “ver detalles” aquí también está presente porque no se mostrará toda la información en forma de columnas en la tabla.

¹⁵ Ante la eventual desvinculación de un trabajador, la actualización de dicho estado se hará a través de la edición del mismo.

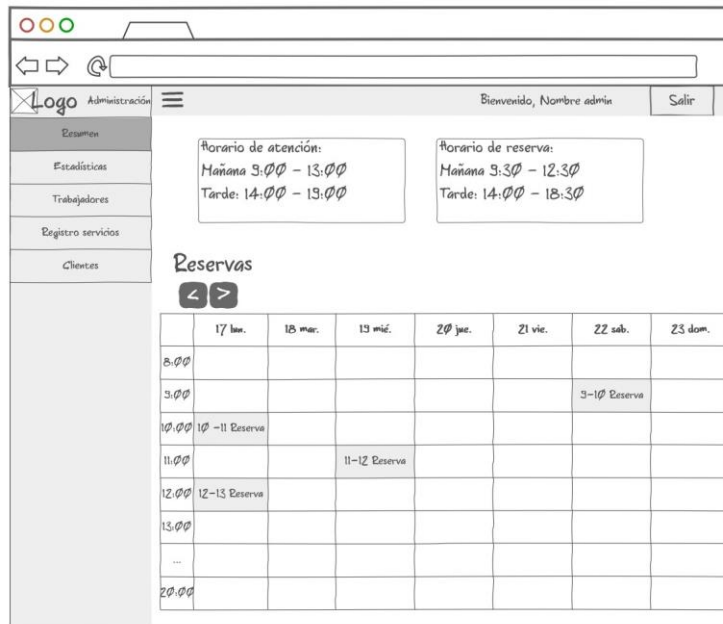


Figura 21: Interfaz del Administrador - Resumen

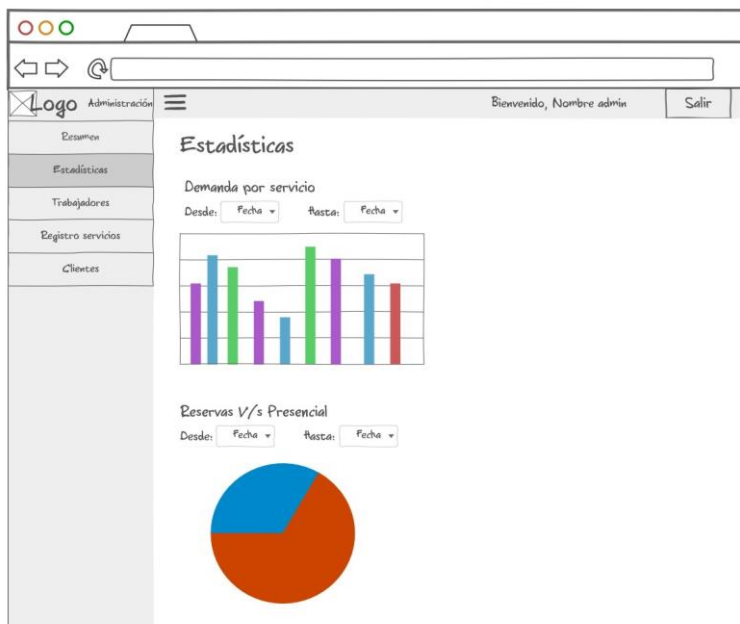


Figura 22: Interfaz del Administrador - Estadísticas

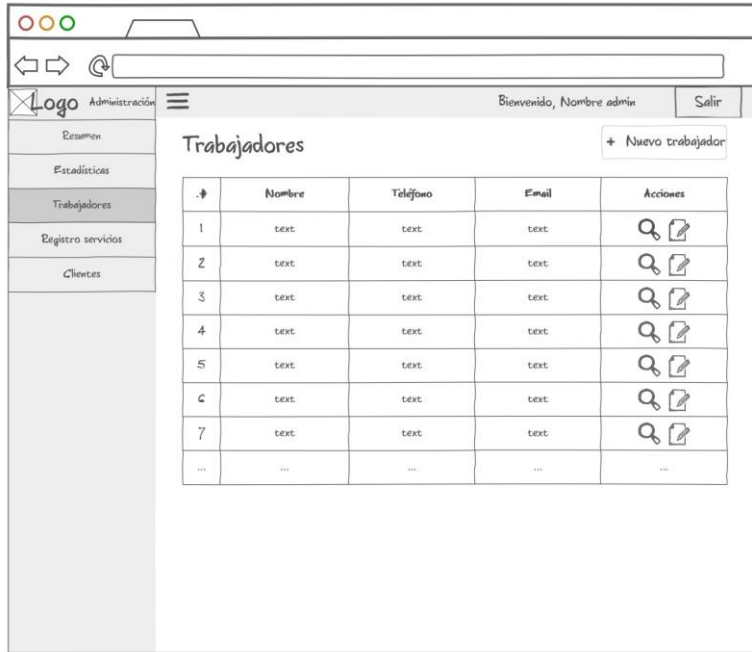


Figura 23: Interfaz del Administrador – Trabajadores

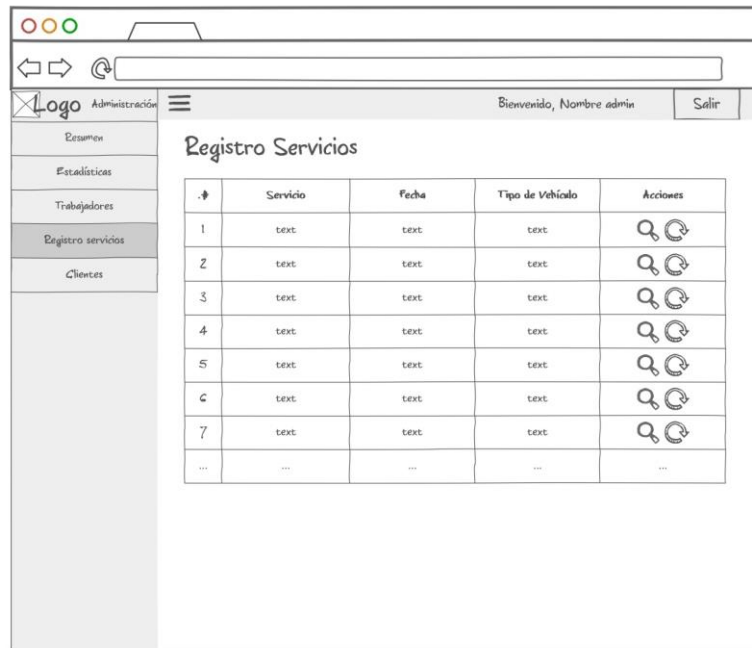


Figura 24: Interfaz del Administrador – Registro Servicios

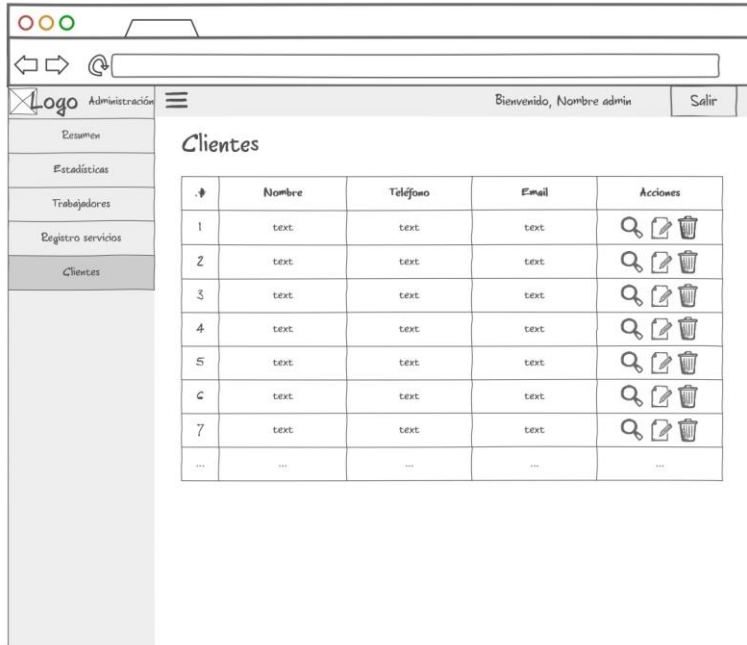


Figura 25: Interfaz del Administrador - Clientes

8 CONCLUSIONES

Ya con el desarrollo del proyecto y el informe finalizado, se presentan aquí las conclusiones de la totalidad del proceso de trabajo, abordado desde distintas perspectivas.

En cuanto a los objetivos del proyecto es posible afirmar que se cumplieron sus objetivos, producto del correcto cumplimiento de los objetivos del sistema. Esto se debe principalmente al buen y exhaustivo proceso de planificación realizado al comienzo del desarrollo del mismo, donde destacan la fase de especificación de requisitos, de análisis y de diseño. De esta manera, se encaminó muy bien al proceso de desarrollo de la aplicación, eliminando la necesidad de realizar cambios sustanciales a la aplicación durante el mencionado proceso de desarrollo.

En lo relativo uso de metodologías, cabe destacar que, a pesar de las fuertes críticas que existen hacia la metodología de desarrollo en cascada, lo cierto es que funcionó bastante bien en este caso. Por supuesto esto de ninguna manera echa por tierra las críticas más fuertes hacia esta metodología de desarrollo, pero enmarcada en un contexto universitario, trabajando de manera individual, llevando a cabo un proceso de familiarización con las tecnologías utilizadas y con el proceso de desarrollo de una aplicación completa, la metodología de desarrollo en cascada funcionó bastante bien. Por supuesto esto también debido al buen proceso de planificación recién expuesto.

Con respecto a las herramientas y lenguajes aquí utilizados, cabe señalar que, si bien con algunas existía cierto grado de familiaridad, debido a su aprendizaje y utilización tanto ramos de la carrera como la práctica profesional, el desarrollar una aplicación de principio a fin ha sido un gran desafío, que, por las tecnologías seleccionadas para llevarlo a cabo, ha tenido extensos períodos de investigación y de estudio, para nada exento de dificultades. Sin embargo, y a pesar de haber escogido tecnologías relativamente nuevas, que en sí mismas son un gran campo de estudio dentro del mundo del desarrollo, y con las cuales no había tanta familiaridad, el hecho de haber completado de buena manera los objetivos propuestos, implicó un gran aprendizaje.

Referente a la planificación inicial del proyecto, se debe decir que, si bien no todo salió perfecto, la distribución del trabajo y estimaciones de tiempo hechas en un inicio, fueron bastante acertadas. En la práctica, algo que contribuyó enormemente a la correcta ejecución de la planificación fue el uso de un tablero *kanban* y la correcta división de tareas.

En cuanto a la aplicación en sí misma, se debe destacar que existe un gran mercado potencial, ya que prácticamente ninguna empresa dedicada al rubro de lavado de vehículos cuenta con un

sistema ni remotamente cercano al que aquí se ha desarrollado. A esto se suma el bajo costo de arriendo, lo que presenta a la aplicación como una solución atractiva para los negocios. Además, el hecho de haber realizado la aplicación con la última tecnología extiende su vida útil en cuanto soporte, y, como característica propia del *stack* utilizado, desataca su capacidad de escalar sin mayor dificultad, y la alta reusabilidad del código.

En el futuro se desea crear una empresa para ofrecer esta aplicación al público, además, también se espera implementar nuevas funcionalidades, tales como: ofrecer la posibilidad de solicitar un servicio de limpieza a domicilio; más y mejores estadísticas; agregar un sistema de pago online; agregar una nueva sección para los clientes registrados a las vistas; implementar una ficha de seguimiento para los vehículos, y permitir que los trabajadores puedan realizar comentarios y complementarlos con fotografías. Además de esto, también se pretende agregar nuevos servicios al catálogo, de tal manera que el sistema de reserva de lavado pase a ser un servicio más dentro de una serie de servicios que estarán disponibles (servicio mecánico, vulcanización, cambio de aceite, etc.).

Desde el punto de vista académico, el proceso de planificación, el proceso de desarrollo de la aplicación y la elaboración del mismo informe, sumado al proceso de investigación, ciertamente han sido una enorme contribución al proceso de aprendizaje y de familiarización con lo que significa llevar a cabo el desarrollo completo de una aplicación, ya que fue una instancia donde se debió aplicar buena parte de los conocimientos obtenidos a lo largo de la carrera, repasando algunos y afianzando otros. Lo que, desde el punto de vista académico, significó una experiencia bastante enriquecedora.

9 BIBLIOGRAFÍA

- "Acerca de Node.js". (2020). *Acerca de Node.js®*. Obtenido de Node.js Web site: <https://nodejs.org/es/about/>
- "Bootstrap 4 Get Started". (2020). *Bootstrap 4 Get Started*. Obtenido de w3schools Web site: https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp
- "Data Modeling Introduction". (2020). *Data Modeling Introduction*. Obtenido de MongoDB documentation Web site: <https://docs.mongodb.com/manual/core/data-modeling-introduction/>
- "El modelo en cascada: desarrollo secuencial de software". (11 de Marzo de 2019). *El modelo en cascada: desarrollo secuencial de software*. Obtenido de Ionos Web site: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- "Express Web site - Home". (2020). *Home*. Obtenido de Express Web site: <https://expressjs.com/>
- "Getting Started with Redux". (2020). *Getting Started with Redux*. Obtenido de Redux Web site: <https://redux.js.org/introduction/getting-started>
- "MongoDB Web site - Home". (2020). *Home*. Obtenido de MongoDB Web site: <https://www.mongodb.com/es>
- "Postman Web site - Home". (2020). *Home*. Obtenido de Postman Web site: <https://www.postman.com/>
- "Qué son las API y para qué sirven ". (2020). *¿Qué es una API? : Qué son las API y para qué sirven*. Obtenido de Red hat Web site: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- "React Web site - Empezando". (2020). *Empezando*. Obtenido de React Web site: <https://es.reactjs.org/docs/getting-started.html>
- "Redacción de contratos y documentos legales". (2020). *Redacción de contratos y documentos legales*. Obtenido de derecho-chile Web site: <http://www.derecho-chile.cl/redaccion-de-contratos/>

- "Redux Essentials, Part 1: Redux Overview and Concepts". (2020). *Redux Essentials, Part 1: Redux Overview and Concepts*. Obtenido de Redux Web site: <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
- BBVAOpen4U. (23 de Marzo de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. Obtenido de BBVA API_Market Web site: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Brizuela, E. (24 de Febrero de 2020). *Municipio proyecta otra alza en vehículos en Chillán*. Obtenido de La Discusión Web site: <http://www.ladiscusion.cl/municipio-proyecta-otra-alza-en-vehiculos-en-chillan/>
- colaboradores de Wikipedia. (16 de Mayo de 2020). *Biblioteca (informática)*. Obtenido de Wikipedia, La enciclopedia libre.: [https://es.wikipedia.org/wiki/Biblioteca_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica))
- Herrera, F. (9 de Julio de 2018). *0.0 - ¿Qué es redux? [Archivo de video]*. Obtenido de <https://www.youtube.com/watch?v=sgFQjRL5niY>
- Microsoft. (2020). *Recursos de Microsoft 365 y Office*. Obtenido de Microsoft Web Site: <https://www.microsoft.com/es-cl/microsoft-365/microsoft-365-and-office-resources?rtc=1#coreui-heading-38bzvj>
- Mora, A. (28 de Enero de 2020). *Los mejores navegadores web de 2020*. Obtenido de pcworld Web site: <https://www.pcworld.es/mejores-productos/internet/mejores-navegadores-web-3672988/>
- Nielsen, J. (24 de Abril de 1994). *10 Usability Heuristics for User Interface Design*. Obtenido de NN Group Web site: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Oak Tree Software. (2 de Enero de 2020). *What is Redux, its importance and integration?* Obtenido de Oak Tree Software Web site: <https://www.oaktreecloud.com/what-is-redux-its-importance-and-integration/>
- Pore, A. (28 de Junio de 2019). *How to Design Schema for Your NoSQL Database?* Obtenido de Dataversity Web site: <https://www.dataversity.net/how-to-design-schema-for-your-nosql-database/#>

- Universidad de Alicante. (2020). *Modelo Vista Controlador*. Obtenido de Universidad de Alicante Web site: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

ANEXOS

ANEXO 1: Enlaces a sitios web de lavados de vehículos

A continuación, se presenta una lista de algunos de los sitios visitados, enmarcados en el proceso de investigación online. Cabe señalar que esta lista abarca servicios en todo el país, no solamente en la región.

- <https://ecowashcar.cl/>
- <https://elcastillolimpio.cl/>
- <https://ww2.copec.cl/lavado>
- <https://www.flashwash.cl/>
- <http://www.ciclon.cl/>
- <https://www.movicenter.cl/precios-lavado-autos-movicenter/>
- <https://www.carssplash.cl/>
- <http://prowash.cl/>
- <https://lavadodeautosbrothers.cl/>
- <https://www.lavadodeautoslosrios.cl/>
- <https://waw.cl/>
- <https://lavadolaser.cl/>
- <https://www.bubblescar.cl/>

ANEXO 2: Resultado de los incrementos del proyecto

Como se mencionó en el cuerpo del informe, para la gestión de tareas se utilizó Trello como herramienta de gestión para llevar a cabo el método *Kanban*. Todo este proceso abarcó un tiempo de tres meses y medio de trabajo.

En total, se trabajó con un total de 8 columnas:

- **Product backlog:** Columna que lista de todas las tareas a realizar durante el proyecto. Para cada iteración se selecciona un conjunto de éstas y se coloca en la columna "To do". Además, también se añadió tres tarjetas informativas para señalar la correspondencia entre un color de etiqueta y su temática.
- **To do:** Lista de tareas a realizar en la iteración en curso
- **Doing:** Tarea que se encuentra en desarrollo
- **Done:** Lista de tareas finalizadas.
- **Incrementos (4 columnas):** Lista de tareas finalizadas correspondientes a un periodo de trabajo.

De manera resumida, el resultado y los temas abarcados en cada incremento fueron:

- **Incremento 1:** Actualización de informe,¹⁶ investigación y trabajo en base de datos (modelo físico y conceptual), con MongoDB.
- **Incremento 2:** Investigación y trabajo en back end, con NodeJs, Express y Mongoose.
- **Incremento 3:** Investigación y trabajo en front end, con React y Redux.
- **Incremento 4:** Generar autenticación e integrar front end con back end.

A continuación, en la Figura 26, se muestra una captura del tablero completo, y en las Figuras 27, 28, 29 y 30, una captura de los incrementos 1,2,3 y 4 respectivamente.

¹⁶ Es necesario señalar que, el proceso de trabajo en el informe no se ve reflejado en el tablero, salvo por algunas tarjetas en la iteración 1.

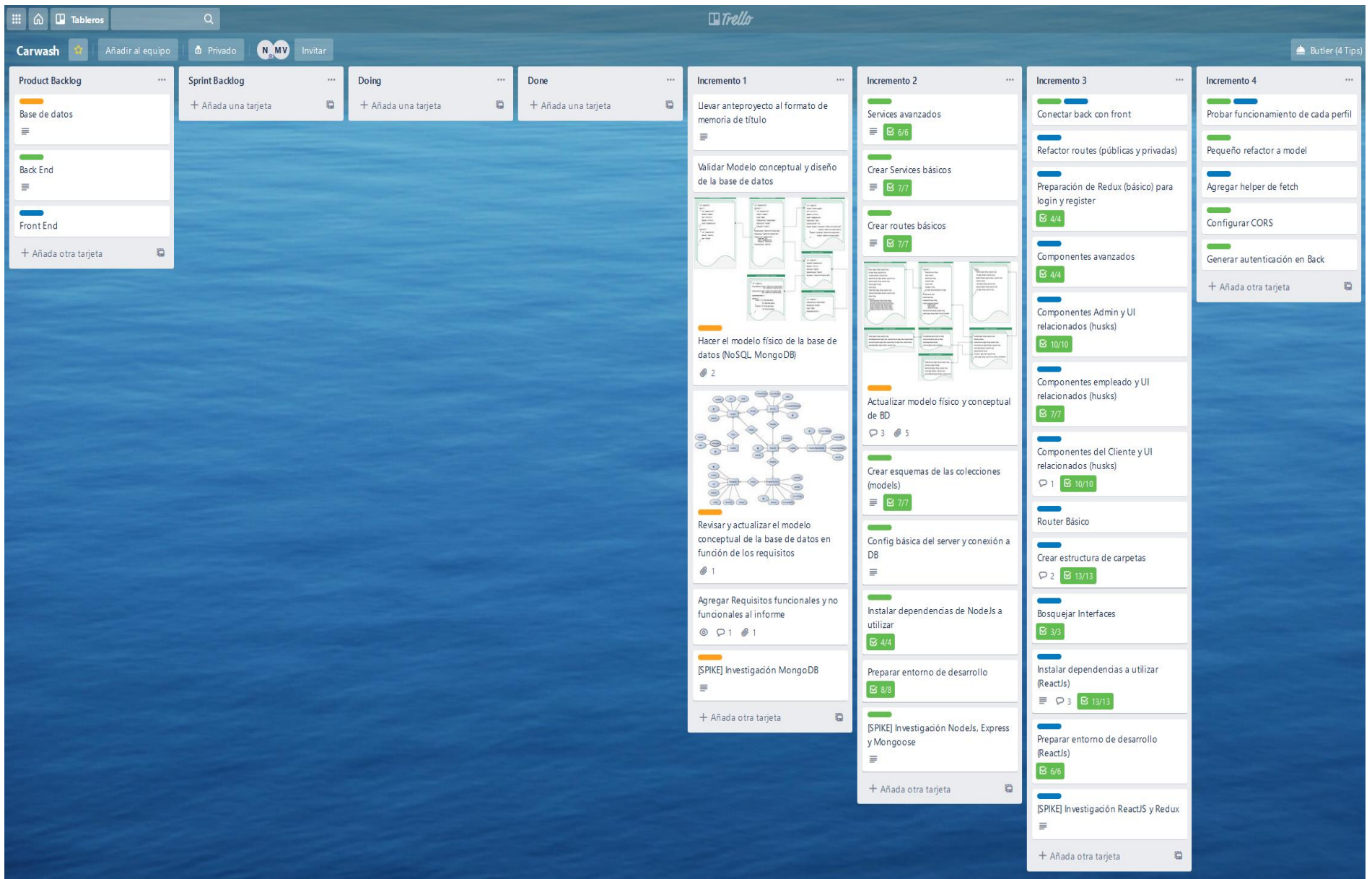


Figura 26: Tablero Kanban

Incremento 1 ***

Llevar anteproyecto al formato de memoria de título

Validar Modelo conceptual y diseño de la base de datos



Hacer el modelo físico de la base de datos (NoSQL, MongoDB)

 2



Revisar y actualizar el modelo conceptual de la base de datos en función de los requisitos

 1

Agregar Requisitos funcionales y no funcionales al informe

  1  1

[SPIKE] Investigación MongoDB

Figura 27: Incremento 1

Incremento 2 ***

Services avanzados

 6/6

Crear Services básicos

 7/7

Crear routes básicos

 7/7



Actualizar modelo físico y conceptual de BD

 3  5

Crear esquemas de las colecciones (models)

 7/7

Config básica del server y conexión a DB

Instalar dependencias de NodeJs a utilizar

 4/4

Preparar entorno de desarrollo

 8/8

[SPIKE] Investigación NodeJs, Express y Mongoose

Figura 28: Incremento 2



Figura 29: Incremento 3

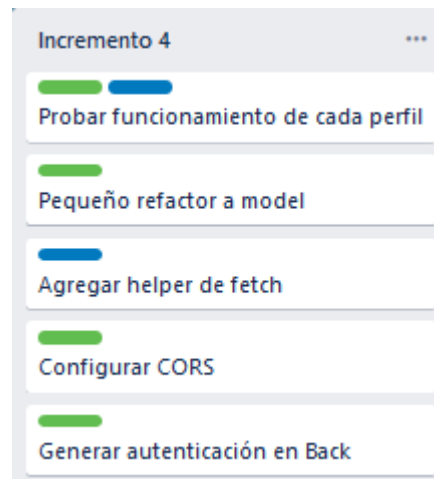


Figura 30: Incremento 4