Universidad Nacional de Ingeniería

Facultad de Electrotecnia y Computación

TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO DE INGENIERO EN COMPUTACIÓN

Sistema web de gestión de inventario y registro de actividades en autolavado “Baño de Espuma”

Elaborado por:

Br. Manuel Abraham Gutiérrez Rojas Carnet: 2008-23295

Br. Jeanneth del Socorro Mota López Carnet: 2008-23337

Br. Darwing Javier Medina Lacayo Carnet: 2010-32819

Tutor:

Msc. Gloria Talía Flores Quintana

Abril, 2023

Dedicatoria

Este trabajo monográfico va dedicado primeramente a Dios, por brindarnos la fuerza y la sabiduría necesarias para superarlos obstáculos y llegar hasta aquí. A pesar de los momentos difíciles y los días agotadores, siempre estuvo a nuestro lado guiándonos en cada paso del camino.

A nuestros queridos padres, a quienes les debemos este logro, pues sin su incondicional apoyo, amor y sacrificio, este trabajo monográfico no habría sido posible. Esta conquista es un tributo a su esfuerzo y dedicación por brindarnos una educación y un futuro mejor.

Y, finalmente, a todos nuestros seres amados, quienes, de manera directa o indirecta, nos han acompañado en esta travesía, brindándonos palabras de aliento y respaldo económico en los momentos más necesarios. A todos ellos, nuestro más profundo agradecimiento y amor.

Agradecimientos

Agradecemos infinitamente a Dios y a nuestros padres por su apoyo y guía en cada etapa de nuestra vida y en cada desafío enfrentado.

A la MSc. Talía Flores, por su acompañamiento y valiosa orientación durante el desarrollo de este trabajo monográfico, su dedicación y compromiso han sido fundamentales para alcanzar nuestros objetivos.

A la Universidad Nacional de Ingeniería, por brindarnos la oportunidad de crecer académica y personalmente, y formarnos como profesionales en el ámbito de la ingeniería en computación.

Extendemos nuestra gratitud a cada profesor que ha sido parte de nuestra formación, compartiendo sus conocimientos, experiencias y pasión por la enseñanza en el transcurso de nuestra carrera. Sus enseñanzas serán siempre atesoradas y aplicadas en nuestra vida profesional.

Gracias a todos ustedes por contribuir en la construcción de nuestro futuro y por ser parte fundamental de nuestro éxito.

Resumen

La presente monografía tiene como objetivo desarrollar un sistema web para la gestión de inventarios y registro de actividades para un autolavado. Para lograrlo, se hace uso de tecnologías web como HTML, CSS, JavaScript, JQuery, Python, Django y MySQL, así como la metodología ágil Scrum.

En el marco teórico se aborda el uso de las mencionadas tecnologías y se profundiza en el uso de Django, un framework de desarrollo web de alta productividad que utiliza el patrón de diseño MVC. Asimismo, se aborda la metodología ágil Scrum, la cual es una metodología de desarrollo de software iterativa e incremental que se centra en la entrega temprana y continua de software de alta calidad.

En el primer capítulo se realiza un estudio de factibilidad en el cual se analizan los requerimientos del sistema, tanto funcionales como no funcionales, las restricciones del sistema, los riesgos asociados al proyecto, así como la factibilidad técnica, operativa, económica, de cronograma y legal. Con base en este análisis se determina la viabilidad del proyecto.

En el segundo capítulo se aborda el análisis y diseño del sistema, utilizando puntos de historia, Product Vision Board, Roles, Historias de Usuario, Product Backlog y Estimación del Product Backlog. Todo esto con el objetivo de obtener una clara visión de lo que se desea construir y cómo se llevará a cabo el proceso de desarrollo.

Finalmente, se presentan las conclusiones y recomendaciones, así como la bibliografía utilizada para el desarrollo de este trabajo. Con esta monografía se busca contribuir al mejoramiento de los procesos de gestión de inventario de “baño de espuma”, a través del uso de tecnologías web y metodologías ágiles de desarrollo. Y demostrar que la tecnología puede acoplarse a múltiples disciplinas, cuando es necesaria para optimizar procesos.

Índice

I. Introducción.....1
II. Antecedentes.....2
III. Objetivos.....4
General.....4
Específicos.....4
IV. Justificación.....5
V. Marco teórico.....7
Conceptos generales de informática para la elaboración de sistemas.....7
i. Sistemas Web.....7
ii. Lenguaje de Marcado HTML.....8
iii. Hojas de Estilo en Cascada CSS.....9
iv. Biblioteca de Manipulación de Objetos JQuery.....9
v. Lenguaje de Programación Python.....10
vi. Framework Web Django.....10
vii. Bases de datos.....11
viii. Estudio de factibilidad.....12
ix. Gestor de Base de Datos MySQL.....13
x. Metodología Ágil Scrum.....13
VI. Estudio de Factibilidad.....15
i. Introducción.....15
ii. Situación actual.....16
iii. Requerimientos del sistema.....17
Requerimientos funcionales.....17
Requerimientos no funcionales.....18
iv. Restricciones del sistema.....19
v. Análisis de riesgo.....20
vi. Factibilidad técnica.....22
Estrategia del hardware.....22
Estrategia del Software.....23
Estrategia de las Comunicaciones.....25
Estrategia de RRHH.....25
vii. Factibilidad operativa.....26
viii. Factibilidad económica.....27
ix. Factibilidad de cronograma.....28

x. Factibilidad legal....32
xi. Conclusiones del capítulo....32
VII. Análisis y Diseño....33
i. Introducción....33
ii. Puntos de historia....34
iii. Product Vision Board....34
Roles en el proceso de desarrollo con Scrum:....35
iv. Historias de Usuario....35
v. Product Backlog....37
vi. Estimación del producto backlog....38
vii. Diagramas de casos de uso....41
viii. Diagramas de estado....44
ix. Diagramas de secuencias....46
x. Diseño de Base de Datos....49
xi. Diseño de Interfaz Gráfica....51
Diseño Adaptable....53
xii. Estrategia de implementación....54
xiii. Estrategia de implantación....57
xiv. Conclusiones del capítulo....58
VIII. Conclusiones....59
IX. Recomendaciones....60
X. Bibliografía....62
XI. Glosario....65
XII. Anexos....68
Anexo 1 : Registro de lavados Software DataShop, pantalla del sistema actual....68
Anexo 2 : Registro de actividades Software Orderry....69
Anexo 3: Resumen Entrevista con Karla Garay, dueña de Autolavado “Baño de Espuma”....70
Anexo 4: Factura de Ejemplo Baño de Espuma....74
Anexo 5: Diccionario de Datos, Sistema web de gestión de inventario y registro de actividades en autolavado “Baño de espuma”....75
1. Modelos Abstractos....75
2. Sucursales....76
3. Crédito....77
4. Clientes....78
5. Empleados....79
6. Pagos....80
7. Servicios....83
8. Tienda....85
9. Suministros....86
10. Vehículos....88

11. Retroalimentación..... 89
Anexo 6: Manual Técnico..... 91
Introducción..... 91
Carpetas..... 91
Entorno de desarrollo..... 92
Clases importantes del sistema..... 95
Views..... 101
Ruta de archivos estáticos (CSS, JavaScript, multimedia)..... 106
Anexo 7: Manual de Usuario..... 110
Acceso al sistema..... 110
Gestionar Sucursales..... 111
Gestionar Clientes..... 113
Gestión de Empleados..... 116
Gestión de Pagos..... 120
Gestión de Promociones..... 124
Gestión Servicios..... 128
Gestión Lavados..... 130
Gestión de Suministros..... 132
Gestión Vehículos..... 136

I. Introducción

En la actualidad, las actividades ejecutadas a lo interno de las organizaciones están cada vez más comunicadas y la tecnología informática desempeña un papel central, la competitividad y los grandes volúmenes de información existentes requieren de sistemas que sean capaces de un manejo y control cada vez más eficientes de dicha información, por lo que las diversas entidades actuales tantos privadas como gubernamentales se ven en la necesidad de implementarlos.

Es así, como la tecnología se ha convertido en un elemento indispensable para el desarrollo de la sociedad. Los sistemas de gestión son una herramienta que permiten mejorar la eficiencia y productividad de los negocios. En este sentido, el sector de los autolavados no es la excepción. Muchos de ellos aún utilizan sistemas de gestión manuales o poco eficientes, lo que afecta su rendimiento y calidad del servicio. Por lo tanto, se hace necesario implementar un sistema de gestión de inventarios y pagos en autolavados que permita optimizar los procesos y mejorar la experiencia de los clientes.

El presente trabajo tiene como objetivo el desarrollo de un sistema web de gestión de inventarios y registro de actividades para autolavados, utilizando Python como lenguaje de programación para el backend, MySQL como gestor de base de datos y HTML, CSS, JavaScript y jQuery para el diseño y funcionalidad de la interfaz de usuario. El sistema tiene como finalidad principal llevar un control efectivo del inventario, permitiendo a los encargados de los autolavados conocer en tiempo real la cantidad de productos y suministros disponibles, además de registrar el proceso de lavado de los vehículos (entradas, salidas, costo y servicios).

La implementación de este sistema permitirá mejorar la gestión de los autolavados, lo que se traducirá en una mayor eficiencia, calidad y rapidez en la atención de los clientes, generando así una mayor satisfacción y fidelización de los mismos. Además, el sistema será altamente escalable y podrá ser utilizado en múltiples sucursales, lo que permitirá una mejor administración y control de los procesos a nivel de empresa.

II. Antecedentes

Autolavado “Baño de Espuma”, es una empresa encargada de brindar los servicios de lavado de vehículos en general (lavado exterior e interior, lavado de Chasís, limpieza profunda, pasteado, encerado y pulido), así como la venta de accesorios, así como servicio de cafetería para comodidad de sus clientes; cuenta con tres años de experiencia, cuenta con dos sucursales, ubicadas en la ciudad de Managua, una en Carretera Sur y la otra en el barrio “El Redentor”.

Actualmente “Baño de Espuma” hace uso de un software de escritorio llamado DataShop (**Ver Anexo 1: Registro de lavados Software DataShop/ pantalla del sistema actual**), el cual les permite gestionar las siguientes tareas:

- Ventas: cotizaciones, cobros, control de inventario, ventas de cafetería, entre otros.
- Promociones: membresías, rifas, lavado de cortesía, cupones y otras ofertas que se realizan de forma no periódica.
- Productos y servicios: combo de productos, multi-inventario, análisis de costos, etc.
- Trabajadores, todo lo relacionado a su contratación y pagos por sus servicios, tales como: marcada(asistencia), elaboración de nómina, bonos, comisiones, vacaciones, contratos.
- Clientes: histórico de visitas, catálogo de vehículos, clientes más frecuentes.
- Seguridad: roles, respaldos automáticos, antivirus.
- Notificaciones: vales, cambio de precio, impresión de tickets.
- Reportes: utilidades, comisiones, rendimiento, venta diaria, mensual y anual.
- Marketing: Mensajes personalizados vía redes sociales, correos electrónicos publicitarios.

La empresa "Baño de Espuma" enfrenta problemas en la gestión de inventarios, pagos y registro de actividades debido a la complejidad de su software actual. Este software de escritorio requiere intervención manual frecuente, afectando la eficiencia y seguridad en los cálculos. Además, presenta dificultades para manejar múltiples sucursales y su costo no está justificado por las necesidades del negocio. El soporte técnico es tedioso y el mantenimiento no es inmediato ni incluido en la suscripción. Por ello, "Baño de Espuma" busca una alternativa más accesible y multiplataforma que se adapte a sus necesidades y ofrezca mejor soporte técnico.

Existen aplicaciones que tienen una funcionalidad similar al proyecto que se pretende crear, a continuación, se citan algunas, las mismas han sido implementadas en otros países:

Orderry[1]: Sistema web de propósito general para la gestión de inventarios y registro de actividades enfocado a distintos tipos de negocio como el autolavado, tiendas de conveniencia, contratistas, talleres, servicios, etc.(**ver Anexo 2: *Registro de actividades Software Orderry***)

Dentro de las principales funciones están las siguientes:

- Control de inventarios (tales como químicos y suministros)
    - Establecer saldos mínimos de consumibles para garantizar el funcionamiento del negocio
    - Ordenar nuevos lotes de insumos
    - Registro de las publicaciones en almacén, transferencias y cancelaciones.
- Plantillas de documentos
    - Hojas de inspección
    - Órdenes de servicio
    - Estimaciones y facturas de autos
- Flujo de Efectivo
    - Contabilidad para ver ingresos / gastos y planificar el presupuesto de forma eficiente.
    - Estadísticas de finanzas
- Registro de Actividad
    - Conocer qué usuario realizó una creación, modificación o eliminación de registros

[1] https://orderry.com/es/car-detailing-software/

III. Objetivos

General

Desarrollar un sistema web para la gestión del registro del inventario y control de actividades en autolavado "Baño de Espuma" con el fin de llevar un mejor control de los productos y automatizar las actividades relacionadas con la atención al cliente.

Específicos

- Analizar los requerimientos del sistema web para la gestión de inventario y registro en el autolavado “Baño de Espuma”.
- Diseñar el modelo del sistema web usando una herramienta de diseño Lenguaje de Modelado Unificado UML.
- Desarrollar el sistema web empleando la metodología ágil Scrum.
- Codificar el sistema web de acuerdo al diseño obtenido utilizando las tecnologías HTML, Python y Javascript.

IV. Justificación

El sistema actual de registro y gestión de inventario en "Baño de Espuma" carece de muchos requerimientos que en la actualidad demanda la empresa, es un Software enlatado[2] y además de escritorio por lo tanto accesible solo desde un ordenador, lo que dificulta el manejo en línea para ambas sucursales. Esto limita la capacidad de ingresar y gestionar las actividades desde cualquier lugar del negocio y también impide la gestión de más de una sucursal con un solo sistema. Como resultado, los datos no están centralizados y puede haber confusión en la gestión de precios, promociones, registro de actividades, etc. Debido a lo anterior es necesario implementar un nuevo sistema que permita una gestión centralizada y accesible desde cualquier lugar.

Un sistema web específico para el autolavado realizará estas tareas con mayor rapidez y precisión. Además, como un requerimiento nuevo para el sistema, está contemplado un módulo que permitiría hacer pedidos a los proveedores, tomando en cuenta el inventario; el sistema podría generar informes automáticamente, lo que permitiría una mejor visibilidad sobre el rendimiento del negocio.

Actualmente, el autolavado cuenta con un sistema costoso que no se está utilizando al máximo. Por esta razón, se propone un sistema a la medida alineándose con las tecnologías actuales que permita controlar el inventario y registrar las transacciones, generando reportes y mejorando el manejo de la existencia de los productos. Este sistema se considera adecuado para la necesidad del autolavado y se podría implementar como caso de estudio para el tema monográfico.

La gestión de inventarios y registro permitirá tener un mejor control de ingresos y egresos del negocio, así como agilizar información a través de la generación de diversos informes y/o reportes. La automatización de estos procesos mediante un sistema web puede ayudar a reducir errores humanos, mejorar la trazabilidad de los inventarios y llevar un registro de clientes. Además, un sistema web de gestión de inventarios y pagos puede ayudar a mejorar la eficiencia

[2] Es un sistema que fue desarrollado con anterioridad por una empresa proveedora, la cual vende “licencias de uso” del mismo.

operativa del negocio, ya que permite una mejor planificación y control de los recursos, y una mejor toma de decisiones.

En el contexto actual, el uso de la tecnología en el manejo de negocios es cada vez más común, y un sistema web de este tipo permitiría a "Baño de Espuma" estar al día con las tendencias tecnológicas y mejorar su competitividad en el mercado. El proyecto tiene una gran importancia educativa ya que es una oportunidad para aplicar los conocimientos adquiridos en el ámbito académico a un problema real y contribuir a mejorar los procesos en “Baño de Espuma”.

V. Marco teórico

A continuación, se presenta una síntesis de los conceptos y teoría utilizada para la elaboración del sistema propuesto.

Conceptos generales de informática para la elaboración de sistemas

i. Sistemas Web

Inicialmente la web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse. El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas que permitieran que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (common gateway interface) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos.[3]

Internet y la Web han influido enormemente tanto en el mundo de la informática como en la sociedad en general. Si nos centramos en la Web, en poco menos de 10 años ha transformado los sistemas informáticos: ha roto las barreras físicas (debido a la distancia), económicas y lógicas (debido al empleo de distintos sistemas operativos, protocolos, etc.) y ha abierto todo un abanico de nuevas posibilidades. Una de las áreas que más expansión está teniendo en la Web en los últimos años son las aplicaciones web. [4]

En la ingeniería de software se denomina aplicación web a un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador o explorador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones[5]. En otras palabras, es un programa que se codifica

[3] CARLES MATEU, Desarrollo de aplicaciones web, Fundació per a la Universitat Oberta de Catalunya, 2004. ISBN: 84-9788-118-4, p 20.
[4] LUJÁN MORA, Sergio. Programación de aplicaciones web: historia, principios básicos y clientes web. Alicante: Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3, p III.
[5] LUJÁN MORA, Sergio. Programación de aplicaciones web: historia, principios básicos y clientes web. Alicante: Editorial Club Universitario, 2002. ISBN 978-84-8454-206-3, p 48.

en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen muchas aplicaciones web como ejemplos, tales como los correos web: Hotmail, Yahoo, Gmail; wikis: Wikipedia; blogs: Wordpress, Blogspot; tiendas en línea: Amazon, Ebay; entre otros, son ejemplos bastante conocidos de aplicaciones web.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

Utilizando las características de los sistemas web, para el autolavado llamado Baño de espuma, va permitir el ahorro de tiempo, optimizar los procesos operativos y tener acceso a información relevante en tiempo real y de forma segura.

ii. Lenguaje de Marcado HTML

HTML es un lenguaje de marcación que sirve para definir el contenido de las páginas web. Se compone en base a etiquetas, también llamadas marcas o tags, con las cuales se consiguen expresar las partes de un documento, cabecera, cuerpo, encabezados, párrafos, etc. En definitiva, el contenido de una página web.

Este será el "esqueleto" de la aplicación, los componentes serán maquetados bajo este lenguaje de marcación, con la particularidad que quien lo generará será el framework Django a través del sistema de plantillas Jinja.

iii. Hojas de Estilo en Cascada CSS

CSS es el lenguaje para describir la presentación de páginas web, incluidos los colores, el diseño y las fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. La separación de HTML de CSS hace que sea más fácil mantener los sitios, compartir hojas de estilo entre páginas y adaptar las páginas a diferentes entornos. Esto se conoce como la separación de la estructura o contenido de la presentación. (W3C,2021)

Será el encargado de agregar los estilos a la capa de presentación, desde los colores definidos en un tema, hasta los layouts de las páginas (columnas, filas, tarjetas, etc.). Existen muchas herramientas que facilitan la creación de código CSS, se eligió Bootstrap 5, ya que brinda mucha flexibilidad en la creación de componentes, permite definir una paleta de colores e incluso, no es requerido descargar código y ponerlo en la aplicación, ya que se puede referenciar desde su servidor. También Bootstrap se complementa adecuadamente con librerías de iconos como fontawesome, la cual se usa para dar estilo a las opciones y botones de la aplicación.

iv. Biblioteca de Manipulación de Objetos JQuery

JQuery es una biblioteca de JavaScript que permite manipular y añadir interactividad a elementos HTML de forma sencilla. En este proyecto, se utilizará JQuery para crear una interfaz de usuario atractiva y dinámica, y para manejar eventos y realizar operaciones de animación. Además, se pueden utilizar plugins de JQuery para agregar funcionalidades adicionales. En resumen, JQuery es una herramienta esencial para el desarrollo de aplicaciones web modernas y eficientes.

El uso de JQuery en este sistema web de gestión de autolavados brinda muchas ventajas en comparación a otras tecnologías. JQuery es una biblioteca de JavaScript que simplifica la forma en que se escribe código para interactuar con el DOM (Document Object Model). Con JQuery, se pueden crear animaciones, hacer peticiones AJAX y manipular elementos HTML de manera más eficiente y fácil de escribir. Esto resulta en una mejor experiencia de usuario y una mayor

velocidad en la carga de páginas. Además, JQuery es compatible con la mayoría de los navegadores web, lo que garantiza el funcionamiento y accesibilidad del sistema en diferentes dispositivos.

v. Lenguaje de Programación Python

Python es un lenguaje de programación de alto nivel, de código abierto, interpretado y con un enfoque en la legibilidad y la eficiencia de su sintaxis. Fue desarrollado en el año 1989 por Guido van Rossum y desde entonces se ha convertido en uno de los lenguajes más utilizados en el mundo para el desarrollo de aplicaciones, desarrollo web, scripts y herramientas de inteligencia artificial. La comunidad de desarrolladores de Python es muy activa y existe una gran cantidad de librerías y paquetes disponibles para resolver una amplia variedad de tareas. Python es un lenguaje versátil y fácil de aprender, por lo que es una excelente opción para el desarrollo de software en diferentes ámbitos.

vi. Framework Web Django

Django es un framework de desarrollo de aplicaciones web de código abierto basado en Python. Fue desarrollado originalmente para manejar varios sitios web a la vez y es conocido por ser un framework rápido, seguro y escalable. Además, cuenta con una amplia comunidad y una amplia gama de herramientas y documentación en línea disponibles. En la aplicación, se utiliza Django como el framework de desarrollo para crear una aplicación web dinámica y robusta que se integre con el motor de base de datos MySQL. Con Django, se pueden crear aplicaciones web complejas y escalables con una cantidad mínima de código, lo que significa que los desarrolladores pueden enfocarse en resolver problemas empresariales en lugar de gastar tiempo en la implementación de características básicas. Django proporciona una amplia gama de herramientas para facilitar la integración de diferentes funcionalidades en la aplicación, como autenticación de usuarios, manejo de formularios, gestión de bases de datos y mucho más.

vii. Bases de datos

Para garantizar la fiabilidad de los datos en el sistema web, estos se deben de tratar de forma importante, es decir garantizando que no existan errores en su manejo, para esto se requiere utilizar un gestor de base de datos.

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.[6]

La base de datos de un sistema de información que integra las entidades y relaciones del sistema en un conjunto de instancias. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos.[7]

En otras palabras, una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas.

Una base de datos tiene las siguientes propiedades implícitas[8]:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado *minimundo* o universo de discurso. Las modificaciones del *minimundo* se reflejan en la base de datos.

- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base

[6] RAFAEL CAMPS PARE, Bases de datos, Fundació per a la Universitat Oberta de Catalunya, 2005. ISBN: 84-9788-269-5, p 5.
[7] RAFAEL CAMPS PARE, Bases de datos, Fundació per a la Universitat Oberta de Catalunya, 2005. ISBN: 84-9788-269-5, p 8.
[8] Ramez Elmasri & Shamkant B. Navathe. (2007). Sistemas de bases de datos - Conceptos fundamentales. Addison-Wesley Iberoamericana p.2.

de datos.

- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

La rápida adopción de la web a los Sistemas de Información hace que los Sistemas Gestores de Bases de Datos incorporen recursos para ser servidores de páginas web, como por ejemplo la inclusión de SQL en guiones HTML, SQL incorporado en Java, etc.[9]

viii. Estudio de factibilidad

Se realizará un estudio de factibilidad para determinar los requerimientos del sistema, la infraestructura tecnológica, la capacidad técnica que implica la implantación de la herramienta informática, los beneficios y el grado de aceptación que se genera en la organización.

El propósito de un estudio de factibilidad es determinar si una oportunidad de negocio es posible, práctica y viable. Cuando se enfrenta a una oportunidad de negocio, muchas personas optimistas tienden a centrarse en los aspectos positivos. Un estudio de factibilidad les permite tener una visión realista de los aspectos positivos y negativos de la oportunidad.

El proceso de definición de un nuevo negocio es fundamental. Un estudio de factibilidad es una herramienta importante para la toma de las decisiones correctas. Una decisión equivocada en este punto a menudo conduce a la quiebra de organizaciones. También los estudios de factibilidad se pueden hacer antes de adquirir un negocio ya existente y antes de la ampliación de una empresa existente.

Con un estudio de factibilidad se permitirá planear con tiempo los recursos requeridos para el desarrollo de un sistema, establecer las bases para efectuar una verdadera evaluación y control del desarrollo, el de informar y aclarar al personal administrativo, usuarios y técnicos en

[9] RAFAEL CAMPS PARE, Bases de datos, Fundació per a la Universitat Oberta de Catalunya, 2005. ISBN: 84-9788-269-5, p 8.

sistemas, respecto a las expectativas reales del sistema. También puede evitar un posible declive en el proyecto, causando que este se descarte por completo y ayuda a encontrar la solución a las necesidades de la facultad, con base en una comprensión de ellas mismas, ajustándose dicha solución a los recursos destinados para el proyecto.

ix. Gestor de Base de Datos MySQL

MySQL es el sistema de administración de bases de datos SQL de código abierto más popular, desarrollado, distribuido y respaldado por Oracle Corporation. Se tomó la decisión de utilizar un software libre y gratuito, que se adapta a las necesidades específicas del usuario. Desde un punto de vista técnico, MySQL fue seleccionado porque es una base de datos relacional, la cual se ajusta a nuestras necesidades ya que no requiere el manejo de grandes cantidades de registros ni será sometida a una frecuente consulta. Además, el mantenimiento de esta base de datos resulta sencillo gracias a los conocimientos de los desarrolladores en bases de datos relacionales, ya sea para la creación de nuevos campos o para una futura migración a un ambiente distinto.

x. Metodología Ágil Scrum

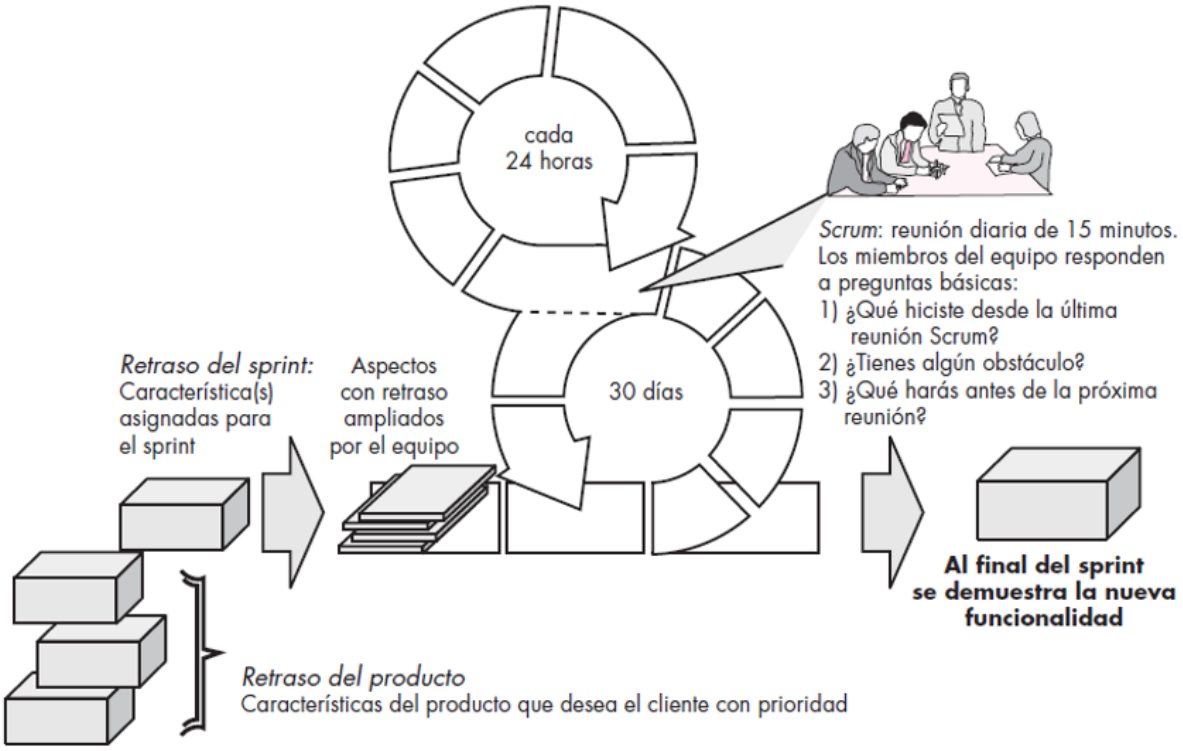
Scrum (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby), es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. (Pressman,2010)

Scrum incorpora un conjunto de patrones del proceso que ponen el énfasis en las prioridades del proyecto, las unidades de trabajo agrupadas, la comunicación y la retroalimentación frecuente con el cliente. (Pressman,2010)

Scrum incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón de proceso llamado sprint.

Figura 1

Flujo de procesos metodología Scrum



VI. Estudio de Factibilidad

i. Introducción

Un proyecto que conlleva el desarrollo de un sistema de información, para la automatización de procesos, es una gran inversión de recursos, tales como: tiempo, dinero y personas; por lo anterior, es necesario estudiar su viabilidad a lo largo de todo del mismo. La viabilidad de un proyecto de este tipo es la medida del beneficio obtenido en una organización a través del desarrollo de un sistema de información.

Fernández Alarcón (2006), ha clasificado los factores que afectan a la viabilidad de un proyecto en las siguientes 6 categorías:

I. Económica
II. Operacional
III. Técnica
IV. De fechas
V. Legal y contractual
VI. Política

En este capítulo se aborda el estudio de factibilidad realizado para el desarrollo del ***"Sistema Web de Gestión de Inventario y Registro de Actividades en Autolavado Baño de Espuma"***, enfocándonos en las siguientes áreas: económica, operacional, técnica, factibilidad de fechas y legal y contractual, omitiendo solamente la política, dado que no aplica para este tipo de proyecto.

ii. Situación actual

"Baño de Espuma" es una empresa dedicada al lavado de vehículos y venta de accesorios, y cuenta con un servicio de cafetería para sus clientes. Al principio, los dueños tenían dudas sobre el éxito del negocio y decidieron contratar a programadores extranjeros para implementar un software enlatado llamado DataShop, el cual cuenta con muchos módulos que no son utilizados por la empresa de autolavado.

Con el crecimiento de la empresa y la creación de nuevas sucursales, surgieron dificultades con el sistema actual utilizado en "Baño de Espuma", ya que carece de funcionalidades necesarias para el negocio y no permite la comunicación entre las diferentes sucursales debido a la limitación de licencia a solo dos computadoras. Esta falta de comunicación impide la compartición del catálogo de clientes y la posibilidad de ofrecer beneficios como descuentos o premios a la fidelidad. Actualmente, se llevan controles separados y se utilizan tablas en Excel o llamadas entre sucursales para fusionar la información, lo que genera molestias en los clientes recurrentes.

iii. Requerimientos del sistema

A continuación, se presentan los requerimientos funcionales y no funcionales que se identificaron. El sistema tiene que cumplir con algunos requerimientos solicitados por los interesados del desarrollo de la aplicación y cada área de esta posee comportamientos específicos, por esto los requerimientos funcionales del sistema necesitan estar clasificados por áreas o funciones.

Requerimientos funcionales

Estos hacen referencia a la descripción de actividades y servicios que un sistema debe proveer, normalmente están vinculados con las entradas, salidas, los procesos y los datos a almacenar en el sistema.

El requisito principal para el sistema es que tenga un diseño responsive o adaptativo, lo que garantiza que la interfaz se ajuste automáticamente a la pantalla y a las características del dispositivo en el que se esté utilizando. Esto es fundamental, ya que los usuarios accederán a la aplicación desde una amplia variedad de dispositivos, como teléfonos inteligentes, tabletas, portátiles y PCs. Al contar con un diseño adaptativo, la experiencia del usuario será óptima en cada dispositivo.

Es importante destacar que el sistema debe ser compatible con diferentes sistemas operativos y navegadores web para cumplir con los requisitos funcionales necesarios. Esto permitirá que la aplicación pueda ser accedida y utilizada desde cualquier dispositivo con conexión a Internet, lo que aumentará su accesibilidad y versatilidad.

Requerimientos funcionales

- **Área de Logística**
    - Almacenar la información de todos los servicios ofertados.
    - Almacenar la información de todo el personal de las sucursales
    - Almacenar la información de los clientes.
    - Almacenar la información de los productos ofertados.
    - Almacenar la información de los vehículos recibidos.
- **Área de venta**
    - Registrar los componentes y/o accesorios de vehículo.
    - Registrar las salidas de accesorios.
- **Área cliente**
    - Registrar las visitas y dar seguimiento a los clientes.
    - Reportar descuentos por fidelidad, concursos, otros.
- **Gerencia general**
    - Visualizar reportes de venta de accesorios.
    - Visualizar reportes de servicios realizados por día, semana y mensual.

Requerimientos no funcionales

Estos describen otras prestaciones, características y limitaciones que debe tener el sistema para alcanzar el éxito, estos tipos de requerimientos engloban características como rendimiento, facilidad de uso, presupuestos, tiempos de entrega, documentación, seguridad y auditorías internas.

Requerimientos no funcionales.

- Facilidad de uso por medio de una interfaz gráfica intuitiva para los usuarios.
- Compatibilidad con las versiones de exploradores más populares.
- Respaldo de la base de datos diario.
- Creación de todos los reportes en PDF.
- Validación de los campos.

iv. Restricciones del sistema

Identifican los límites del software originados por aquellos problemas o situaciones que el sistema NO está en capacidad de resolver, por las limitaciones del ámbito del proyecto o de los recursos disponibles.

En el Sistema Web de Gestión de Inventario y Registro de Actividades en Autolavado "BAÑO DE ESPUMA" las restricciones del sistema son las que se detallan a continuación:

- El sistema ha sido diseñado para funcionamiento vía web.
- El sistema será de uso exclusivo para la empresa denominada Autolavado "BAÑO DE ESPUMA" y sus respectivas sucursales.
- El sistema no estará en capacidad de comunicarse con cualquier otro sistema que esté desarrollado en la empresa, limitando su uso a las funciones para las que fue creado.

v. Análisis de riesgo

Constituye la previsión de los riesgos que podrían afectar el desarrollo satisfactorio del sistema, estos podrían ser de carácter técnico, presupuestarios, de personal, de tiempo, etc. así como plantear las posibles soluciones.

Tabla 1
Análisis de riesgo

Riesgo	Descripción	Consecuencia	Solución
El sistema será entregado tarde o nunca será entregado.	El sistema sufrirá retrasos o se abandonará el desarrollo del mismo.	La organización perderá interés en el sistema y por ende abandonará su desarrollo.	Se ha elaborado un calendario de trabajo que contempla los tiempos de entrega.
El sistema será difícil de usar.	El sistema no contará con una interfaz gráfica intuitiva, o los procedimientos serán complicados de realizar.	El personal se resistirá a usarlo, y por ende el sistema caerá en desuso.	Se tomará en cuenta la retroalimentación por parte de los usuarios en cada uno de los avances presentados del sistema, a fin de realizar ajustes necesarios a la interfaz gráfica o los procedimientos.
El sistema tendrá caídas recurrentes.	El sistema debido a la sobrecarga de trabajo por los usuarios o alguna deficiencia en la red de conectividad o sistema eléctrico, causará que el mismo deje de funcionar.	El sistema caerá en desuso por la ineficacia técnica.	Se harán pruebas de rendimiento para garantizar que los recursos de hardware destinado soportan perfectamente la plataforma desarrollada, así mismo se verificará que todas las condiciones de conectividad, sean las adecuadas para

Riesgo	Descripción	Consecuencia	Solución
			garantizar el funcionamiento.
Los costos de realización de este proyecto superan el presupuesto de la organización	Los altos costos representan un problema para permitir que se continúe con el desarrollo del mismo.	El sistema será abandonado por los altos costos.	La organización no tiene que invertir en el desarrollo del sistema, ya que, al ser un proyecto de monografía, el personal de desarrollo ha asumido los costos implicados.
El sistema será difícil de mantener técnicamente	Las tecnologías utilizadas para el desarrollo del sistema, son poco conocidas y no existe personal que pueda brindar servicio de actualización o mejoras.	El sistema caerá en abandono o estará desfasado tecnológicamente.	Se publicará el código fuente en una plataforma abierta, a fin de que cualquier programador con experiencias en las tecnologías utilizadas, pueda seguir con el desarrollo del mismo.
El sistema será vulnerable a robo de información o hackeo.	El sistema podría ser víctima de alguna actividad de hackeo que comprometa la integridad de la información, como la disponibilidad del mismo.	El sistema presentará inconsistencia en los datos almacenados, o los mismos podrían desaparecer por alguna actividad de hackeo.	El sistema contará con medidas de seguridad que garanticen la integridad de los datos, tanto a nivel de sistema operativo, como en la misma aplicación.

Fuente: Elaboración propia.

vi. Factibilidad técnica

Esta es una medida del éxito de la puesta en práctica de la solución técnica específica y de la disponibilidad de los recursos y los conocimientos técnicos necesarios.

Presenta la tecnología que se requiere para alcanzar la funcionalidad y el rendimiento del sistema, contemplando tanto la disponibilidad de los recursos como la necesidad de nuevos recursos de hardware y software para el desarrollo y funcionamiento del mismo. Dentro de la factibilidad técnica, nos hemos basado en cuatro ámbitos, los cuales se detallan a continuación:

- Elección del hardware
- Elección del software
- Elección del sistema de comunicaciones
- Elección de los recursos humanos

Estrategia del hardware

Se propone utilizar una computadora personal para el desarrollo del sistema y manejo de la información para la aplicación, que consiste en una laptop con las siguientes características:

Tabla 2
Recursos de Hardware Desarrollo

Marca	Apple
Versión	MacBook Pro (15-inch, 2019)
Procesador	2.3 GHz 8-Core Intel Core i9
Memoria	16 GB 2400 MHz DDR4

Fuente: Levantamiento de datos.

Para la etapa de producción, se utilizará el servidor virtual en la nube de Google, el cual consta de las siguientes características de hardware.

Tabla 3
Recursos de Hardware Producción

Procesador	Intel Broadwell
Instancia	e2-small
Disco Duro	10 GB
Memoria	2 GB
CPU Virtuales	2

Fuente: Levantamiento de datos.

Estrategia del Software

Con respecto al software, durante la etapa de desarrollo se trabajará con sistema de administración de contenedores virtuales Docker. Además, se realizó un levantamiento de los recursos de software necesarios para el desarrollo del sistema.

Tabla 4
Estrategia de Software Desarrollo

Sistema Operativo	macOS Monterey 12.5.1
Servidor HTTP	Nginx 1.18.0
Gestor de Base de Datos	MySQL 8.0.32
Lenguaje de Programación	Python 3.10.6
Herramientas de Diseño	Figma
Herramientas Auxiliares	Docker Desktop v4.14.1 PyCharm 2021.3.3 (Professional Edition)

Fuente: Levantamiento de datos.

En la fase de producción, se utilizará un entorno de máquina virtual instalada y configurada en el servidor principal, con las características siguientes:

Tabla 5
Estrategia de Software Producción

Sistema Operativo	Ubuntu 22.04.2 LTS
Servidor HTTP	Nginx 1.18.0
Gestor de Base de Datos	MySQL 8.0.32
Lenguaje de Programación	Python 3.10.6
Framework	Django 4.1.7

Fuente: Levantamiento de datos.

Para el acceso al sistema, solamente se requiere una laptop con acceso a internet y un navegador web moderno compatible con los estándares actuales de HTML, a continuación, se detallan los navegadores y las versiones sugeridas para garantizar la compatibilidad con el sistema desarrollado.

Tabla 5
Navegadores compatibles

Navegador Web	Versión
Google Chrome	108.0
Mozilla Firefox	100.3
Microsoft Edge	104.0

Fuente: Levantamiento de datos.

Estrategia de las Comunicaciones

Estrategia de RRHH

El equipo de trabajo solamente está integrado por un equipo de desarrolladores externos a la institución, sin embargo, para el desarrollo del sistema se ha interactuado con personal de la empresa, para consultas, pedir retroalimentación en cada uno de los sprint de desarrollo. Para obtener un producto de calidad se requiere un ambiente de trabajo armonioso, respetuoso, y con disposición a colaborar entre ambas partes.

Las cualidades necesarias durante las etapas de análisis, diseño y programación se detallan a continuación: pensamiento crítico, creatividad, proactividad, capacidad para resolver problemas, adaptabilidad a situaciones desfavorables y reponerse de fallos potenciales del grupo de trabajo.

Durante la etapa de análisis y diseño, uno de los desarrolladores se estará desempeñando como analista, es necesaria tener una buena capacidad de análisis y entendimiento de situaciones, sólidos conocimientos en la metodología de análisis y diseño orientado a objetos y buen manejo de las herramientas CASE para la elaboración de los diagramas UML.

Durante la construcción del sistema ciertas habilidades son esenciales para poder trabajar en este proyecto. Dado su propósito multiplataforma es esencial el entendimiento para la elaboración y estructuración de una aplicación web, siendo requeridas habilidades de desarrollo avanzado en lenguajes como HTML, Javascript y Python, teniendo como únicas herramientas para esto, editores de texto como Visual Studio Code y PyCharm y los navegadores web compatibles con la aplicación a desarrollar.

También son requeridos buenos conocimientos del lenguaje SQL y del gestor MySQL Workbench dado que muchos de los procesos serán elaborados desde el gestor para simplificar la programación. Es necesario un buen desempeño en el framework Django, para desarrollar los módulos de esta aplicación.

El proyecto pretende diseñar un prototipo funcional, el cual está sujeto a las solicitudes de mejoras y cambios al momento de su implementación, para cumplir con esto será necesario que las autoridades o interesados y miembros del grupo designen a un encargado para continuar con sus mejoras y mantenimiento posterior a la entrega del documento monográfico y prototipo de la aplicación.

vii. Factibilidad operativa

Para el desarrollo de esta aplicación, se ha garantizado que la interfaz gráfica sea intuitiva y de fácil uso, permitiendo que los nuevos usuarios se acostumbren rápidamente al uso de cada una de las funciones.

En las entrevistas con personal de la empresa, se manifestó la necesidad de que algunas funciones del sistema pudieran ser accedidas desde un dispositivo móvil, como un celular, debido a que la aplicación es Web, algunas funciones se podrán consultar por esta vía, pero otras no, debido a la complejidad de operar los datos en pantalla, por lo que recomendamos que estas operaciones se realicen desde un dispositivo con pantalla más grande, como una laptop.

Durante las primeras pruebas del sistema con los usuarios de la empresa, estos se mostraron satisfechos con los avances y la retroalimentación fue muy positiva, lo que manifiesta que no existe resistencia al cambio, lo que hace que el sistema sea factible operacionalmente.

En el proceso de capacitación con el personal de la empresa se presentarán los procedimientos para las operaciones del sistema, así mismo se hará hincapié en los niveles de responsabilidad de los usuarios del sistema, debido a la importancia de los datos que se manejan.
En este caso se crearán 4 niveles de usuarios:

- Usuario dueño/super administrador.
- Usuario administrador.
- Usuario cajero.
- Usuario lavador.

Esto con el fin de que las operaciones de eliminar y crear nuevos accesorios, solamente esté limitada a los primeros tres tipos de usuario y el resto del personal (lavadores), pueda tener acceso al mismo sistema con un tipo de usuario de consulta, sin temer que este elimine alguna información sensible en el sistema.

viii. Factibilidad económica

El costo monetario que implica este proyecto monográfico se plantea a continuación. Se evalúan los gastos tanto en el diseño como en el desarrollo del sistema, los equipos de hardware y el requerimiento de software necesarios para la debida implementación. Para el desarrollo de este sistema, las herramientas de diseño, análisis y producción.

El equipo de programación tampoco cobrará ningún costo por el desarrollo del sistema, ya que se trata de un trabajo monográfico para optar al título de Ingeniero en Computación, sin embargo, la empresa ha solicitado cuantificar el costo del desarrollo de la herramienta, para registrarlo como una donación por parte del equipo.

Tabla 6
Costo del desarrollo del sistema

Etapa de desarrollo	Duración (meses)	Personal	Costo por etapa	Costo total
Análisis	2	1	$1,500.00	$3,000.00
Diseño	1	1	$1,400.00	$1,400.00
Codificación	3	1	$1,400.00	$4,200.00
Pruebas	1	1	$1,250.00	$1,250.00
Total	**7**	**4**	**$5,550.00**	**$9,850.00**

Fuente: Levantamiento propio

ix. Factibilidad de cronograma

Se ha estimado el tiempo de desarrollo del Sistema Web, por los autores del trabajo monográfico, y está ajustada a las necesidades de "BAÑO DE ESPUMA", por tanto, sobrelleva una serie de etapas que alargan el proceso de desarrollo de la misma. Para la estimación de la cantidad de trabajo a realizar para el desarrollo se efectuará un análisis de puntos de función.

Los puntos de función se derivan con una relación empírica según las medidas contables del dominio de información de software y las evaluaciones de la complejidad del software.

Tabla 7
Puntos de función sin ajustar - PFSA

No	Tipo	Cantidad	Complejidad Media	Sub - Total
1	Entrada Externa - EI	10	4	40
2	Salida Externa - EO	9	5	45
3	Consulta Externa - EQ			0
4	Archivo Lógico Interno - ILF	10	10	100
5	Archivo de Interfaz Externo- EIF			0
Total PFSA				**185**

Fuente: Levantamiento propio

Tabla 8
Factor de Ajuste Complejidad - FAC

No	Factor de Ajuste Complejidad - FAC	Puntaje
1	Comunicación de datos	4
2	Procesamiento distribuido	4
3	Rendimiento	1

No	Factor de Ajuste Complejidad - FAC	Puntaje
4	Copias de seguridad y recuperación fiables	1
5	¿Se ejecutará el sistema en un entorno operativo y existente y fuertemente utilizado	3
6	¿Requiere el sistema de entradas de datos interactivas en línea?	5
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	2
8	¿Se actualizan los archivos maestros de forma interactiva en línea?	3
9	Procesamiento complejo de entradas, salidas o consultas	1
10	Procesamiento interno complejo	1
11	Reusabilidad del código	0
12	Facilidad de implementación	1
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	2
14	¿Facilidad de cambio para poder ser fácilmente utilizada por el usuario?	4
Total Factor de Ajuste		**32**

Fuente: Levantamiento propio

Puntos de Función Ajustados

PFA = PFSA * [0.65+(0.01*FAC)]

PFA = 185 * [0.65 + (0.01 * 32)]

PFA = 185 * [0.65 + 0.32]

PFA = 185 * [0.97]

PFA = 179.45

Horas Hombre = PFA * Horas PF

HH = 179.45 * 8 (Lenguaje de programación de 4ta generación)

Horas Hombre = 1435.6

Estimación de duración del proyecto

Desarrolladores - D	3
Horas laborables – HR (sábados y domingos)	64
Días de trabajo por mes – DT (sábados y domingos)	8

Horas por desarrollador

Horas = HH/Desarrollador

Horas = 1435.6 / 3

Horas = 478.53

Días de trabajo

DT = HH / HL

478.53/64 = 7.47

7.47 días

Meses de trabajo

7.47 /8

1 mes

Tabla 9
Cronograma de actividades

Actividades a Desarrollar	Duración Días	Observaciones/Productos
Recopilación de Historias de usuarios	**1**	Elaborado Documento de Requerimientos
Planificación del Product Backlog	**15**	Elaborado el análisis y diseño del Sistema
Ejecución de Sprint de 5 días	**54**	Desarrollo del Sistema Web
Sprint N	**5**	Actividades del Sprint
Sprint Planning	**0.25**	Actividades que se realizan en cada Sprint
Especificaciones de Casos de Uso	**0.25**	
Sprint Backlog	**4**	
Sprint Review	**0.25**	
Sprint Retrospective	**0.25**	
Documentación Final	**16**	Documento Final
Pre Defensa	**1**	
Defensa	**1**	

Fuente: Elaboración propia

x. Factibilidad legal

La siguiente parte del estudio ha sido empleado para determinar si los requisitos violan o atentan contra alguna ley o reglamento. En este caso se cuenta con la aprobación de los clientes interesados, se ha presentado el software a utilizar y no hay ningún problema con las leyes del país.

xi. Conclusiones del capítulo

Luego de realizar el presente estudio de factibilidad y analizar cada uno de los factores que influyen en este, se obtuvo como resultado que el desarrollo del SISTEMA WEB DE GESTIÓN DE INVENTARIO Y REGISTRO DE ACTIVIDADES EN AUTOLAVADO "BAÑO DE ESPUMA", se considera viable desde el punto de vista estratégico, debido a que dará solución a las necesidades planteadas por dicha empresa, considerando los principales aspectos que debían tratarse y que fueron identificados en conjunto con los usuarios a través de los requerimientos funcionales.

VII. Análisis y Diseño

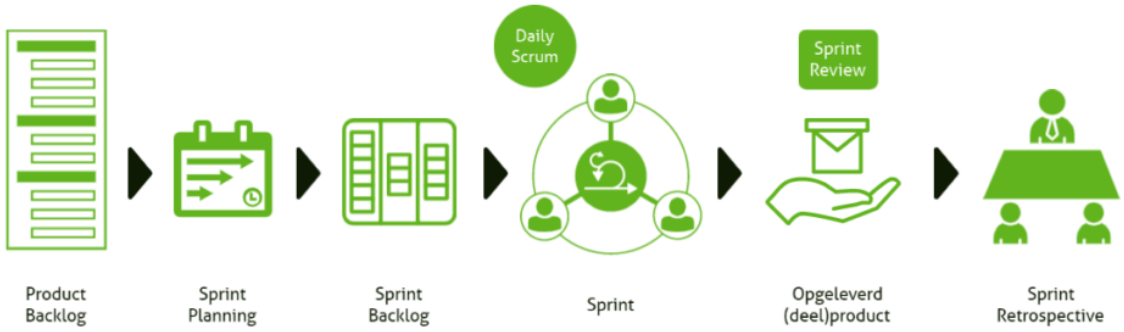
i. Introducción

El análisis y diseño de sistemas tiene como objetivo planificar el desarrollo de los sistemas de información mediante la comprensión y la especificación en detalle de lo que debe hacer un sistema y cómo deben implementarse los diferentes componentes del mismo para trabajar conjuntamente [10].

A continuación, en este capítulo se muestran el análisis y diseño utilizado para el desarrollo de la aplicación. Haciendo uso de Scrum

Figura 2

Flujo de trabajo de la metodología Scrum



Nota: tomado de: leansixsigmagroep.nl/lean-agile-en-six-sigma/scrum

[10] https://teorema-rd.com/producto/curso-analisis-y-diseno-de-sistemas#:~:text=El%20an%C3%A1lisis%20y%20dise%C3%B1o%20de,del%20mismo%20para%20trabajar%20conjuntamente.

ii. Puntos de historia

Tabla 10
Clasificación de puntos de historia

Tamaño	Puntos	Tiempo (Días)
XtraSmall (XS)	1	½
Small (S)	2	1
Medium (M)	3	2
Large (L)	5	3
XtraLarge (L)	8	5

Fuente: Elaboración Propia.

iii. Product Vision Board

Tabla 11
Product Vision Board para el sistema

Baño de Espuma	Sistema web de gestión de inventario y registro de actividades en autolavado "Baño de espuma"		
Usuarios	**Necesidades**	**Producto**	**Valor**
● Dueño ● Administrador ● Cajero ● Lavador	Control de inventarios Plantillas de documentos Flujo de Efectivo Registro de Actividad	Administración por sucursal Registro de Vehículos Registro de Servicios Promociones Directorio de Clientes Inventarios de insumos Inventario de Cafetín Registro de Personal Modalidad de Pago Cuentas por Cobrar Reporte de rendimiento Quejas y sugerencias	Eficiencia operativa del negocio Mejorar la trazabilidad de los inventarios

Fuente: Elaboración Propia.

Roles en el proceso de desarrollo con Scrum:

Product Owner
Scrum Master
Development Team
Cliente: Baño de Espuma

iv. Historias de Usuario

Tabla 12
Lista de historias de usuario

HU1: Ingresar al sistema	
Como:	Dueño, Administrador, Cajero, Lavador
Quiero:	Ingresar al sistema
Para:	Tener acceso a toda la funcionalidad según mi rol

HU2: Gestionar usuarios del sistema	
Como:	Dueño
Quiero:	Listar, crear, modificar y eliminar los usuarios del sistema
Para:	Que utilicen el sistema según el rol proporcionado

HU3: Administrar Perfil de Usuario	
Como:	Dueño, Administrador, Cajero, Lavador
Quiero:	Ver y actualizar los datos personales
Para:	Personalizar la cuenta de usuario

HU4: Cambiar contraseña	
Como:	Dueño, Administrador, Cajero, Lavador
Quiero:	Cambiar la contraseña asignada
Para:	Mantener segura la sesión y acciones en el sistema

HU5: Gestionar Sucursales	
Como:	Dueño
Quiero:	Listar, buscar, crear, modificar y eliminar sucursales
Para:	Permite escalar el negocio agregando sucursales

HU6: Gestionar Empleados	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar, eliminar y asignar usuarios de empleados
Para:	Administrar información sobre el personal cada sucursal

HU7: Gestionar Clientes	
Como:	Dueño, Administrador, Lavador
Quiero:	Listar, buscar, crear, modificar y eliminar clientes
Para:	Registrar los datos de cliente para un control adecuado de estos

HU8: Gestionar Vehículos	
Como:	Dueño, Administrador, Lavador
Quiero:	Listar, buscar, crear, modificar y eliminar vehículos de clientes
Para:	Llevar registro de los vehículos de clientes y los servicios ofrecidos

HU9: Gestionar Suministros	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar, eliminar, control de existencias los suministros de la sucursal
Para:	Registrar el inventario de los insumos necesarios para brindar los servicios de lavado

HU10: Gestionar Inventario	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar, eliminar, control de existencias de la tienda
Para:	Registrar el inventario de los artículos de la tienda

HU11: Gestionar Servicios	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar y eliminar servicios
Para:	Llevar un catalogo de los servicios que ofrece el lavado

HU12: Gestionar Promociones	
Como:	Dueño
Quiero:	Listar, buscar, crear, modificar y eliminar promociones
Para:	Registrar promociones como: descuentos, ofertas y paquetes de servicio

HU13: Registrar Lavados	
Como:	Lavador
Quiero:	Registrar servicio de lavado de autos
Para:	Registrar el servidor de lavado de autos

HU14: Gestionar Lavados	
Como:	Lavador
Quiero:	Actualizar, Eliminar registro de lavado de autos
Para:	Actualizar estado o cancelar del servicio de lavado y generar solicitud de pago

HU15: Gestionar Créditos	
Como:	Dueño, Administrador
Quiero:	Buscar, listar, crear, modificar y eliminar créditos
Para:	Registro de las cuentas por cobrar

HU16: Abonar Crédito	
Como:	Cajero
Quiero:	Abonar a un crédito no completado
Para:	Permite registrar abono de créditos

HU17: Registrar Venta	
Como:	Lavador
Quiero:	Registrar una venta de tienda
Para:	Llevar registro de ventas del autolavado

HU18: Gestionar Tienda	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar y eliminar ventas de tienda
Para:	Registrar las ventas de la tienda del autolavado

HU19: Realizar Pago	
Como:	Cajero
Quiero:	Crear y cobrar pagos de servicios y ventas
Para:	Registrar pago de servicios y ventas

HU20: Gestionar Pagos	
Como:	Dueño, Administrador
Quiero:	Listar, buscar, crear, modificar y eliminar pagos
Para:	Registrar pagos de servicios y ventas

Fuente: Levantado de requerimientos.

v. Product Backlog

Tabla 13

Lista de tareas con prioridad

Historia de usuario	Descripción	Prioridad
HU1	Ingresar al sistema	1
HU2	Gestionar usuarios del sistema	1
HU3	Administrar Perfil de Usuario	1
HU4	Cambiar contraseña	1
HU5	Gestionar Sucursales	2
HU6	Gestionar Empleados	2
HU7	Gestionar Clientes	2
HU8	Gestionar Vehículos	2
HU9	Gestionar Suministros	3
HU10	Gestionar Inventario	3
HU11	Gestionar Servicios	3
HU12	Gestionar Promociones	3
HU13	Registrar Lavados	4
HU14	Gestionar Lavados	4
HU15	Gestionar Créditos	5
HU16	Abonar Crédito	5
HU17	Registrar Venta	6
HU18	Gestionar Tienda	6
HU19	Realizar Pago	7
HU20	Gestionar Pagos	7

vi. Estimación del producto backlog

Tabla 14
Estimación de tareas basado en puntos de historia

Sprint	HU	Descripción	Tamaño	Puntos	Tiempo
Primer Sprint	HU1	Ingresar al sistema	M	3	2
	HU2	Gestionar usuarios del sistema	M	3	2
	HU3	Administrar Perfil de Usuario	XS	1	0.5
	HU4	Cambiar contraseña	XS	1	0.5
Segundo Sprint	HU5	Gestionar Sucursales	M	3	2
	HU6	Gestionar Empleados	L	5	3
Tercer Sprint	HU7	Gestionar Clientes	L	5	3
	HU8	Gestionar Vehículos	M	3	2
Cuarto Sprint	HU9	Gestionar Suministros	XL	8	5
Quinto Sprint	HU10	Gestionar Inventario	XL	8	5
Sexto Sprint	HU11	Gestionar Servicios	L	5	3
	HU12	Gestionar Promociones	M	3	2
Séptimo Sprint	HU13	Registrar Lavados	M	3	2
	HU14	Gestionar Lavados	L	5	3
Octavo Sprint	HU15	Gestionar Créditos	L	5	3
	HU16	Abonar Crédito	M	3	2
Noveno Sprint	HU17	Registrar Venta	M	3	2
	HU18	Gestionar Venta	L	5	3
Decimo Sprint	HU19	Realizar Pago	XL	8	5
Undécimo Sprint	HU20	Gestionar Pagos	XL	8	5
Puntos de historia / Tiempo estimado (Time Boxing)				88	55

Fuente: Elaboración propia.

Tabla 15
Listado de sprints de desarrollo

Sprint 1

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU1	Ingresar al sistema	3	2	02/01/2023	03/01/2023
HU2	Gestionar usuarios del sistema	3	2	04/01/2023	05/01/2023
HU3	Administrar Perfil de Usuario	1	0.5	06/01/2023	06/01/2023
HU4	Cambiar contraseña	1	0.5	06/01/2023	06/01/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 2

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU5	Gestionar Sucursales	3	2	09/01/2023	10/01/2023
HU6	Gestionar Empleados	5	3	11/01/2023	13/01/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 3

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU7	Gestionar Clientes	5	3	16/01/2023	18/01/2023
HU8	Gestionar Vehículos	3	2	19/01/2023	20/01/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 4

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU9	Gestionar Suministros	8	5	23/01/2023	27/01/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 5

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU10	Gestionar Inventario	8	5	30/01/2023	03/02/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 6

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU11	Gestionar Servicios	5	3	06/02/2023	08/02/2023
HU12	Gestionar Promociones	3	2	09/02/2023	10/02/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 7

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU13	Registrar Lavados	3	2	13/02/2023	14/02/2023
HU14	Gestionar Lavados	5	3	15/02/2023	17/02/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 8

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU15	Gestionar Créditos	5	3	20/02/2023	22/02/2023
HU16	Abonar Crédito	3	2	23/02/2023	24/02/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 9

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU17	Registrar Venta	3	2	27/02/2023	28/02/2023
HU18	Gestionar Venta	5	3	01/03/2023	03/03/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 10

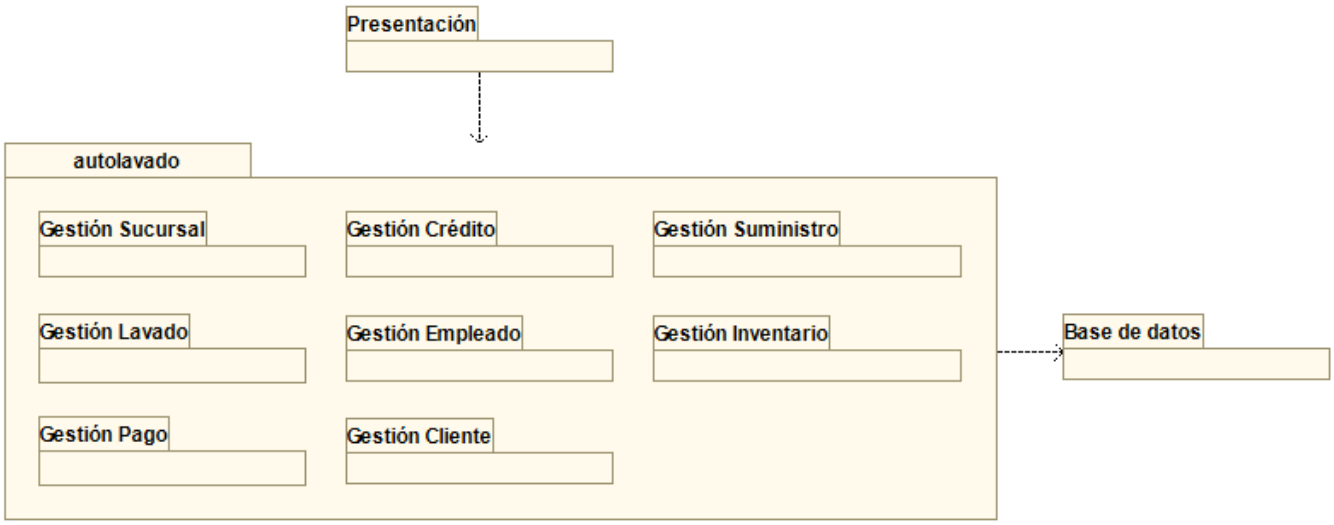
HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU19	Realizar Pago	8	5	06/03/2023	10/03/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Sprint 11

HU	Descripción	Puntos	Tiempo	Inicio	Finalización
HU20	Gestionar Pagos	8	5	13/03/2023	17/03/2023
Puntos de historia / Tiempo estimado (Time Boxing)		8	5		

Figura 3

Diagrama de Paquete



Fuente: Elaboración Propia.

En el diagrama anterior, se muestran cada una de las áreas contempladas en el sistema de autolavado.

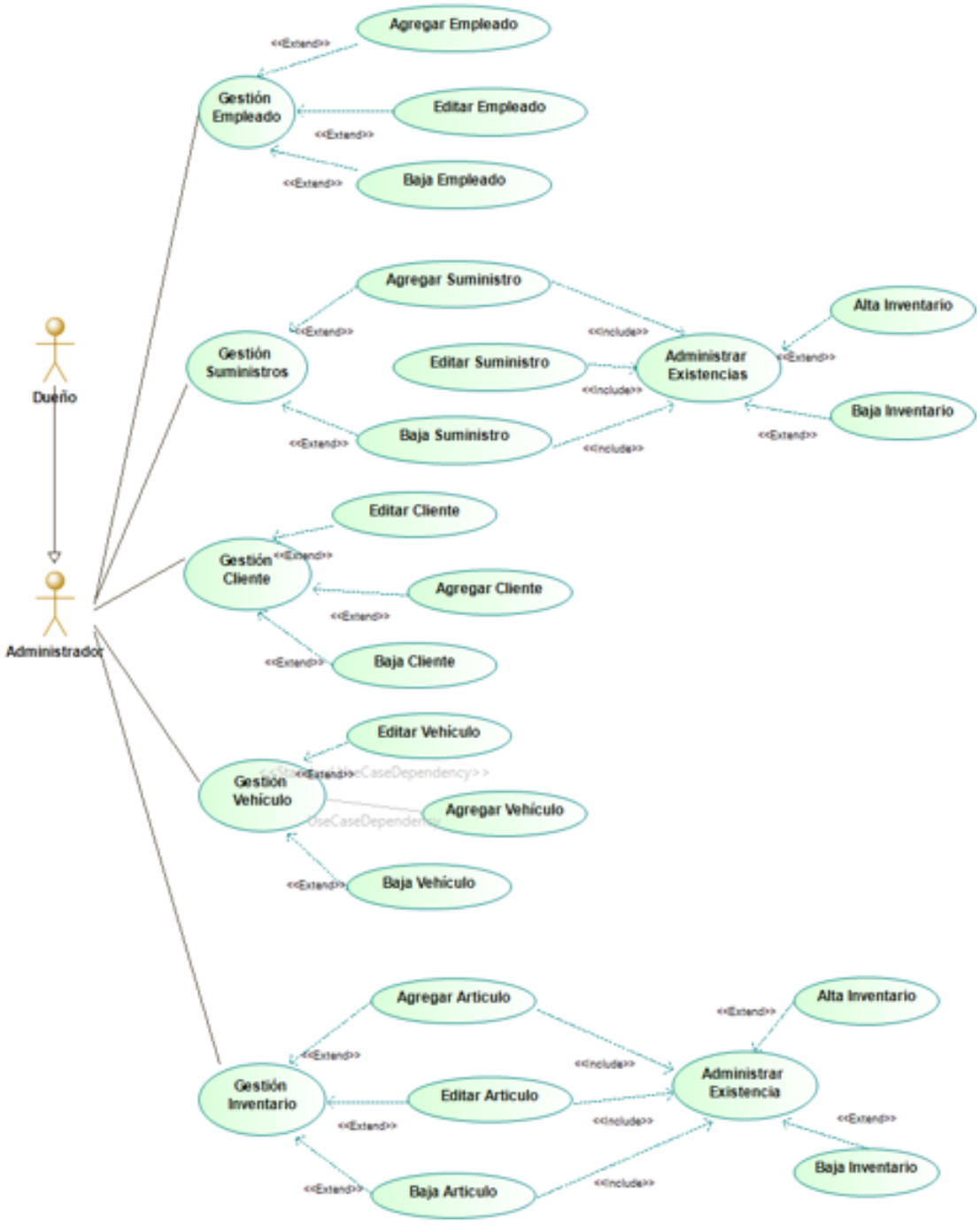
vii. Diagramas de casos de uso

Un caso de uso provee a los desarrolladores un panorama sobre lo que desean los usuarios. Un caso de uso es visto como una secuencia de transacciones en un sistema. Un caso de uso siempre describe tres cosas: un actor que inicia un evento, el evento que desencadena un caso de uso y el caso de uso que realiza las acciones desencadenadas por el evento. [11]

[11] Kenneth E. Kendall, Julie E. Kendall, 2011, Análisis y diseño de Sistemas, Octava edición, México, Pearson education, Capítulo 10 Análisis y diseño orientado a objetos mediante el uso de UML, Modelado de casos de uso, p 287

Figura 4

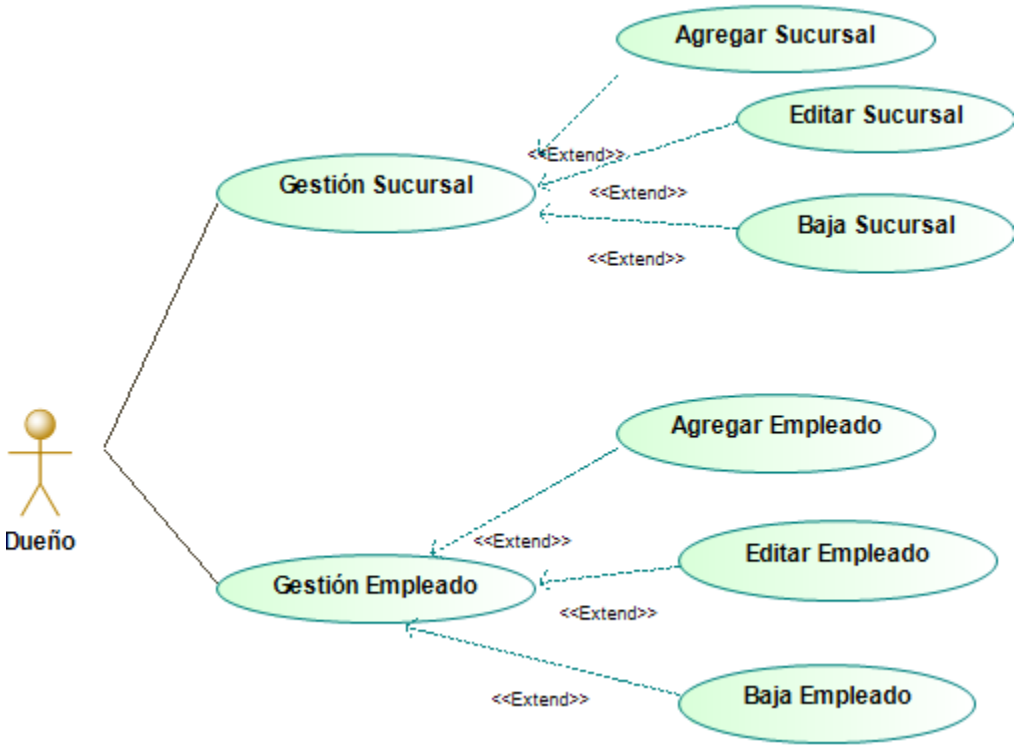
Diagrama de Caso de Uso Rol Administrador



Fuente: Elaboración Propia

Figura 5

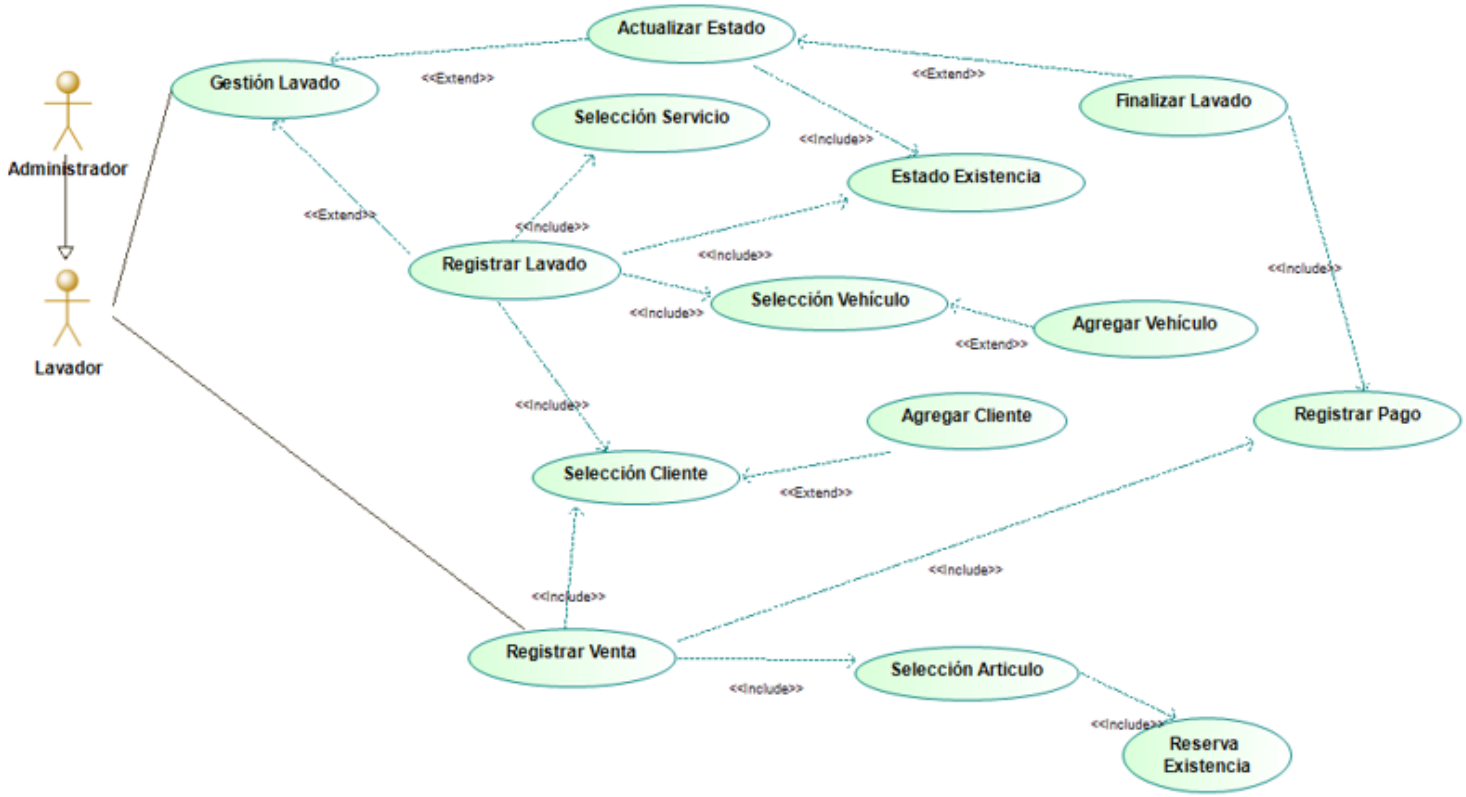
Diagrama de Caso de Uso Rol Dueño



Fuente: Elaboración Propia.

Figura 6

Diagrama de Caso de Uso Rol Lavador



Fuente: Elaboración Propia.

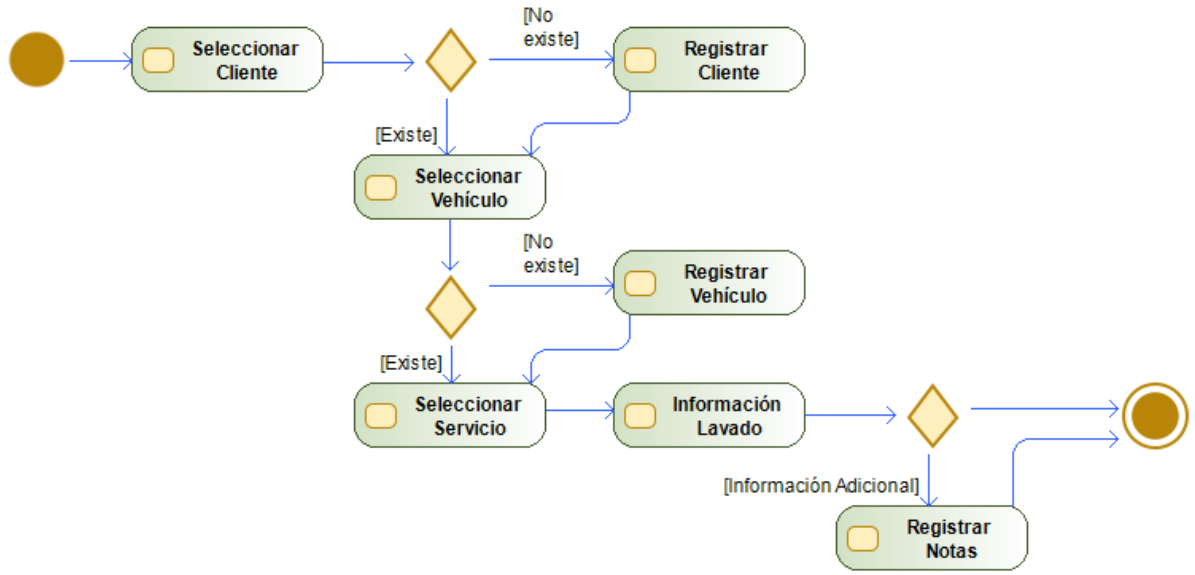
viii. Diagramas de estado

El diagrama de estado, es otra herramienta para determinar los métodos de las clases. Se utiliza para determinar los distintos estados que puede tener un objeto. [12]

[12] Kenneth E. Kendall, Julie E. Kendall, 2011, Análisis y Diseño de Sistemas, Octava edición, México, Pearson Education, Capítulo 10 Análisis y Diseño Orientado a Objetos mediante el uso de UML, Diagramas de comunicación, P 309

Figura 7

Diagrama de Actividad de Registro de Lavado

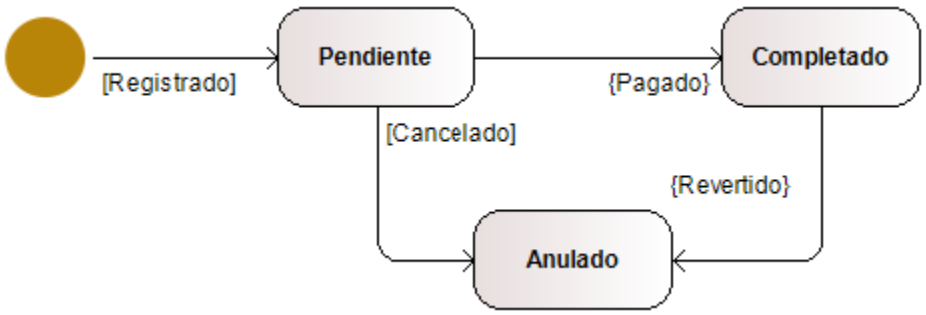


Nota: Elaboración Propia.

Se presenta el diagrama relacionado al registro correspondiente al lavado de vehículos, en el cual, se registra el cliente en el caso de no existir, en caso contrario se procede a buscar en el sistema. De igual manera, se buscan o registran los clientes, se indica el servicio ofertado y se registra el mismo.

Figura 8

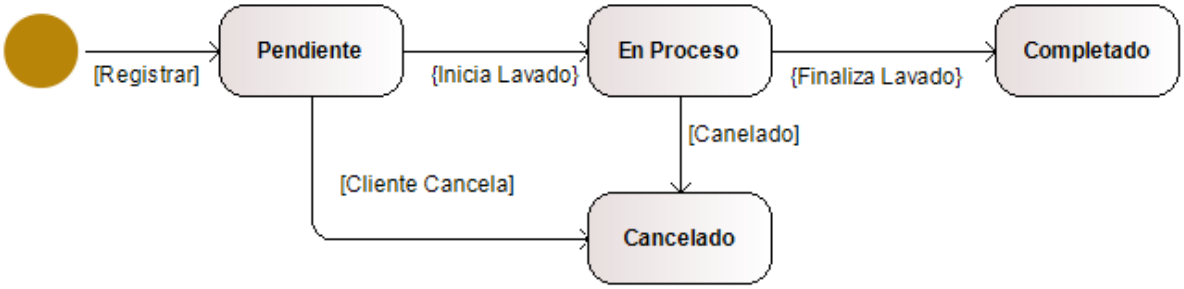
Diagrama de Actividad de Pago



Nota: Elaboración Propia.

Figura 9

Diagrama de Actividad de lavado



Nota: Elaboración Propia.

ix. Diagramas de secuencias

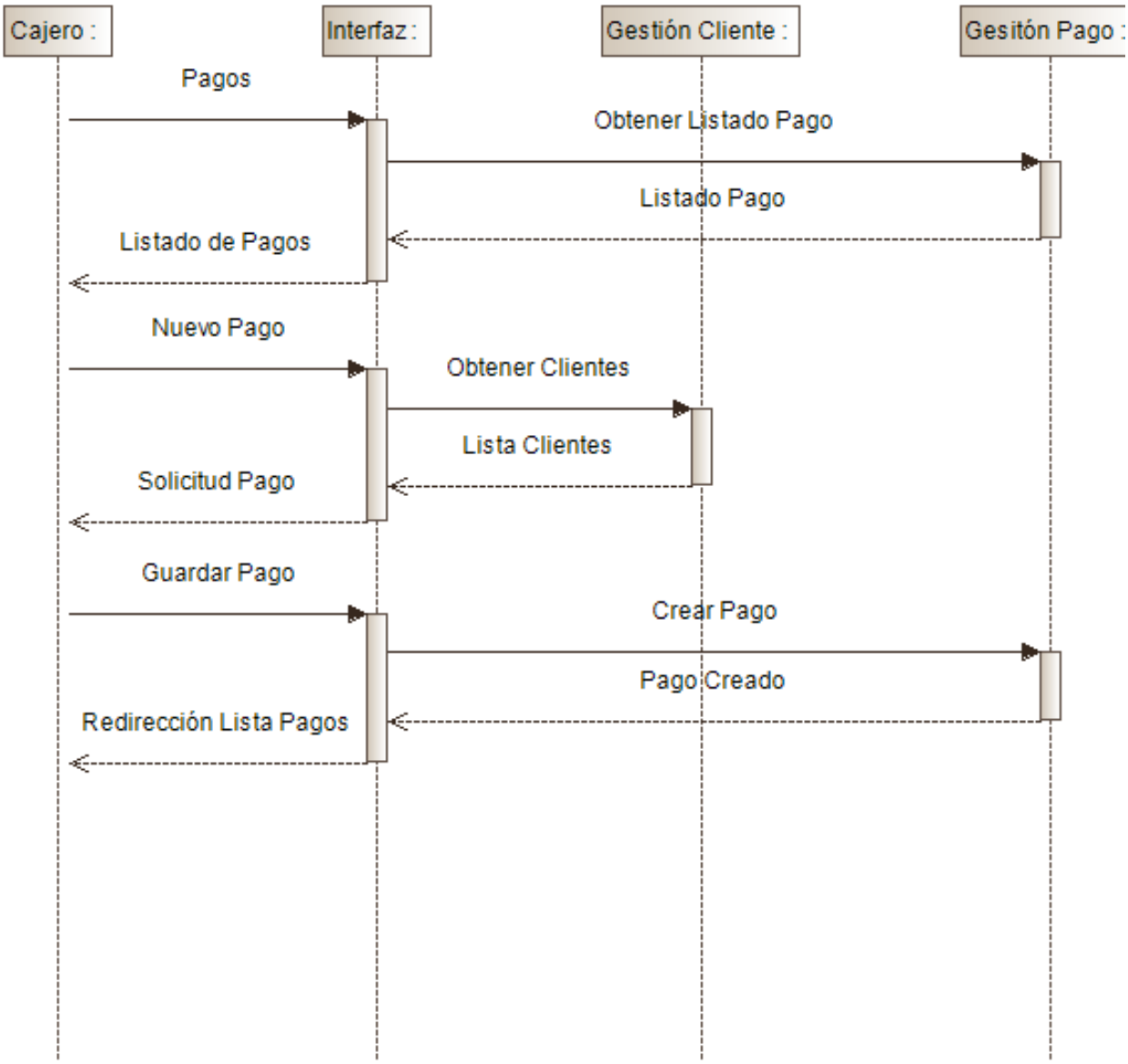
Los diagramas de secuencia son una solución de modelado dinámico popular en UML porque se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.[13]

A continuación, se muestran algunos de los diagramas de secuencia que se han realizado para el sistema.

[13] https://www.lucidchart.com/pages/es/diagrama-de-secuencia

Figura 10

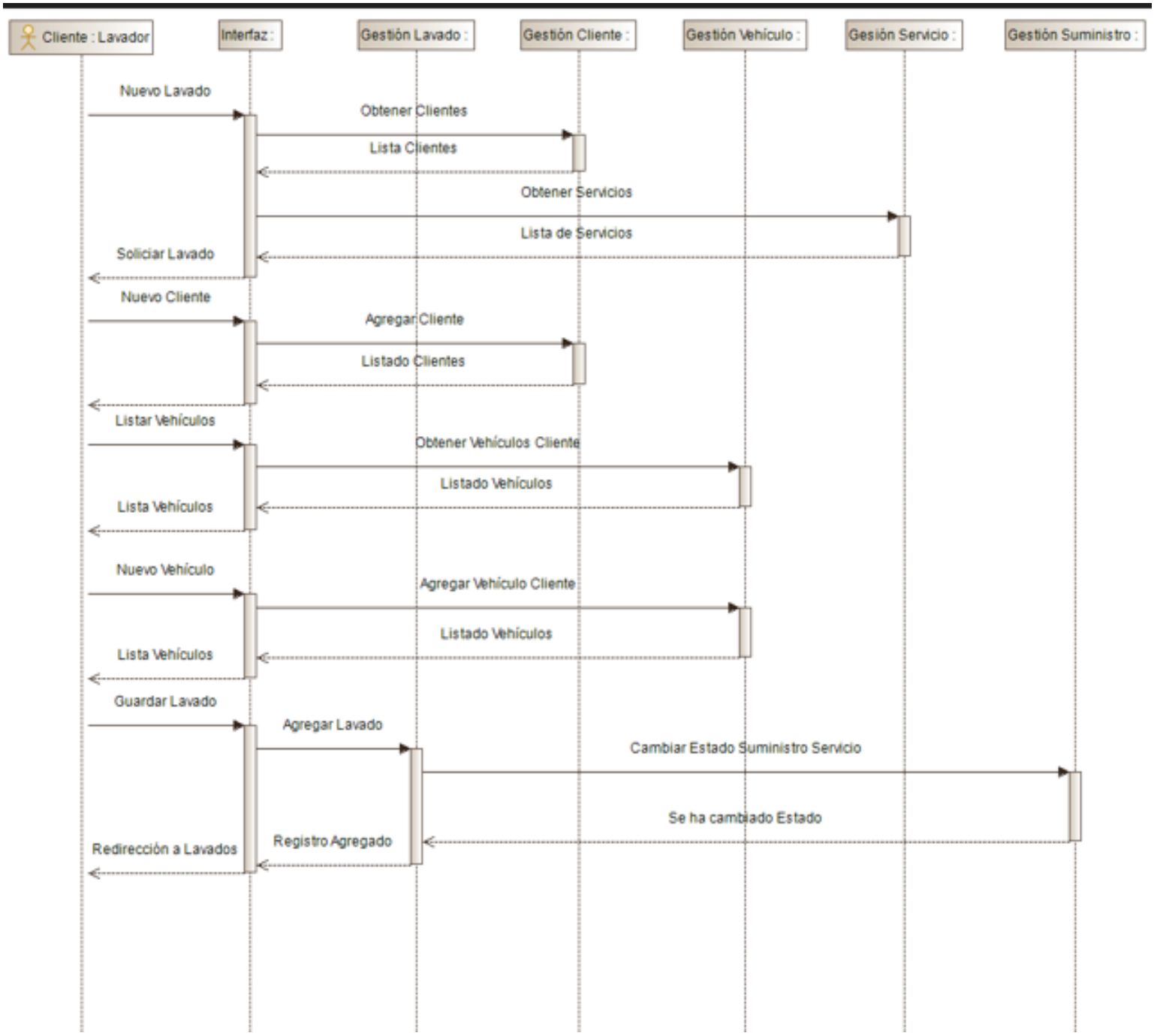
Diagrama de Secuencia de lavado



Nota: Elaboración Propia.

Figura 11

Diagrama de Secuencia de Registro de lavado



Nota: Elaboración Propia.

x. Diseño de Base de Datos

Luego de identificar lo que el sistema debe hacer (requisitos funcionales) y cómo debe comportarse (requisitos no funcionales), se comenzó a trabajar en la estructura de datos que el sistema utilizará, siguiendo una serie de pasos propuestos por una metodología de diseño de bases de datos relacionales.

Primero, se abordó la etapa de diseño conceptual, en la que se identificaron las entidades y relaciones que representan la realidad del sistema, basándose en lo que se había definido previamente.

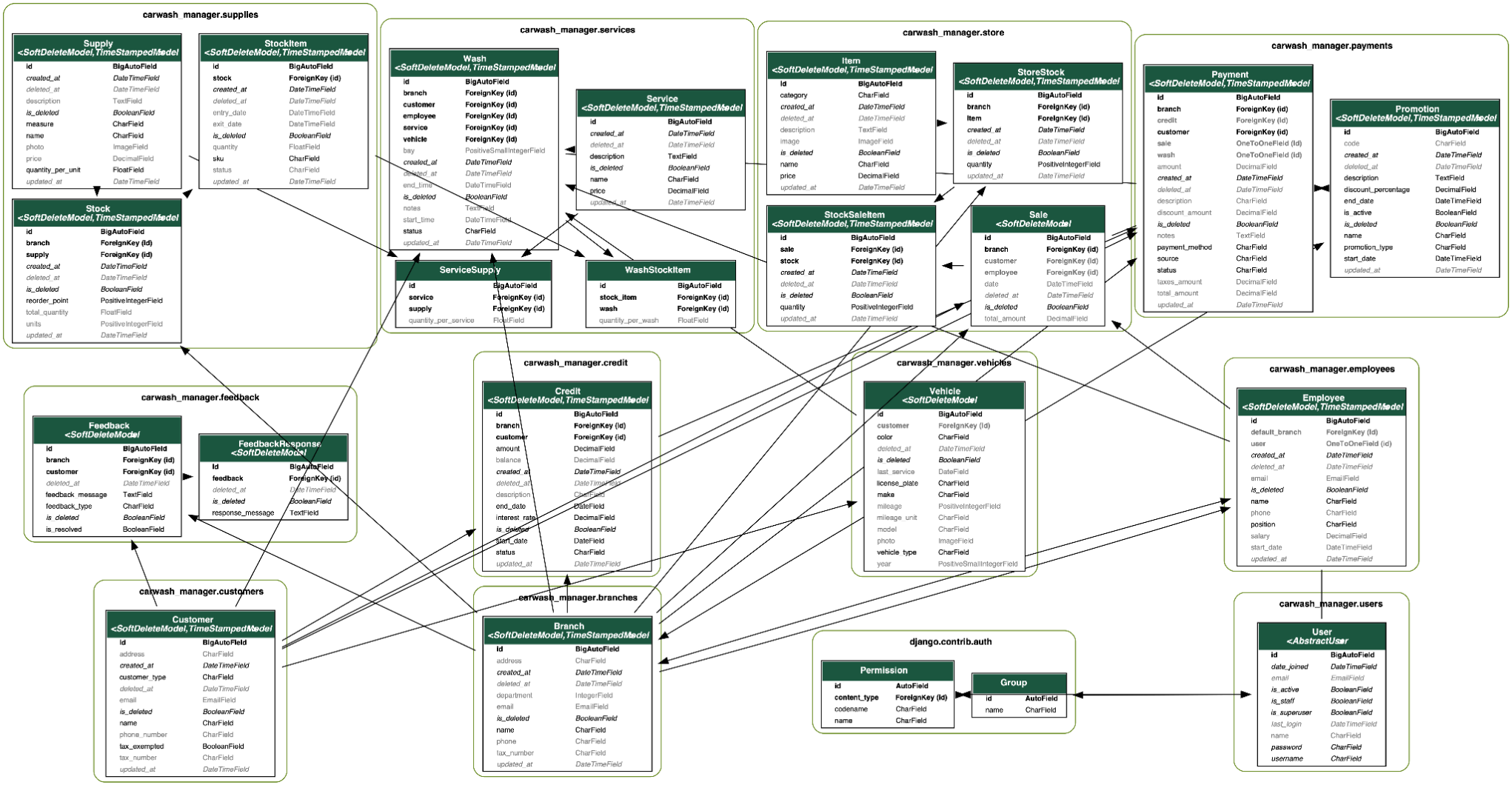
Después, se trabajó en el diseño lógico, donde se convirtieron las entidades identificadas en objetos más adecuados para los gestores de bases de datos. Para esto, se utilizaron técnicas de normalización para asegurar que la información sea coherente y evitar redundancias.

Como resultado de estas etapas de diseño, se creó un modelo relacional que se incorporó en el gestor de base de datos MySQL a través del ORM del Framework Web Django. La elección de este programa se basó en un análisis técnico previo (**Consultar Sección vi: Factibilidad Técnica**).

A continuación, se presenta la estructura de datos relacional obtenida a través del proceso de diseño:

Figura 12

Modelo Entidad Relación (ER)



xi. Diseño de Interfaz Gráfica

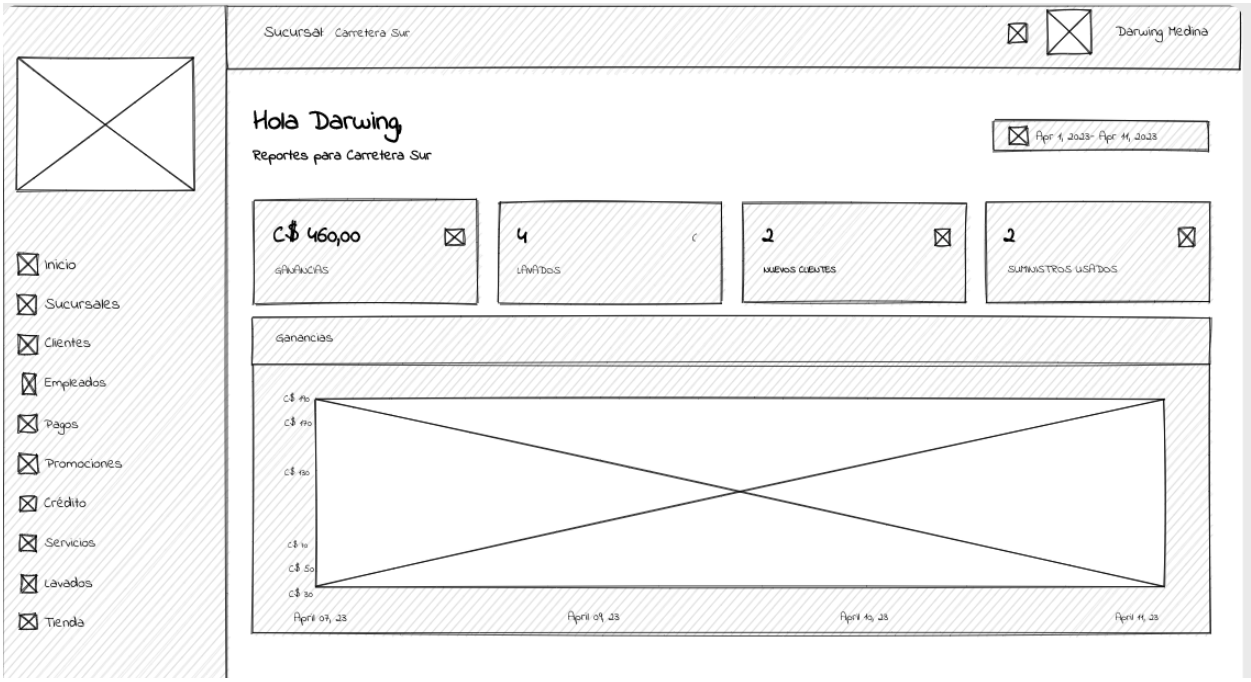
El diseño meticuloso de la interfaz de usuario (UI) es esencial en el desarrollo de software, ya que debe adaptarse a las habilidades, experiencia y expectativas de quienes la utilizarán. Las interfaces gráficas de usuario (GUI) desempeñan un papel crucial en el éxito de un producto. Un diseño de interfaz de usuario bien ejecutado puede evitar errores comunes y mejorar la experiencia del usuario.

En el caso del sistema de gestión de inventario y registro de actividades en el autolavado "Baño de Espuma", se aplicaron varios principios de diseño en la creación de la interfaz de usuario:

- **Familiaridad del usuario**: la interfaz debe emplear términos, conceptos e imágenes conocidos por el usuario, relacionados con su ámbito laboral.

Figura 13

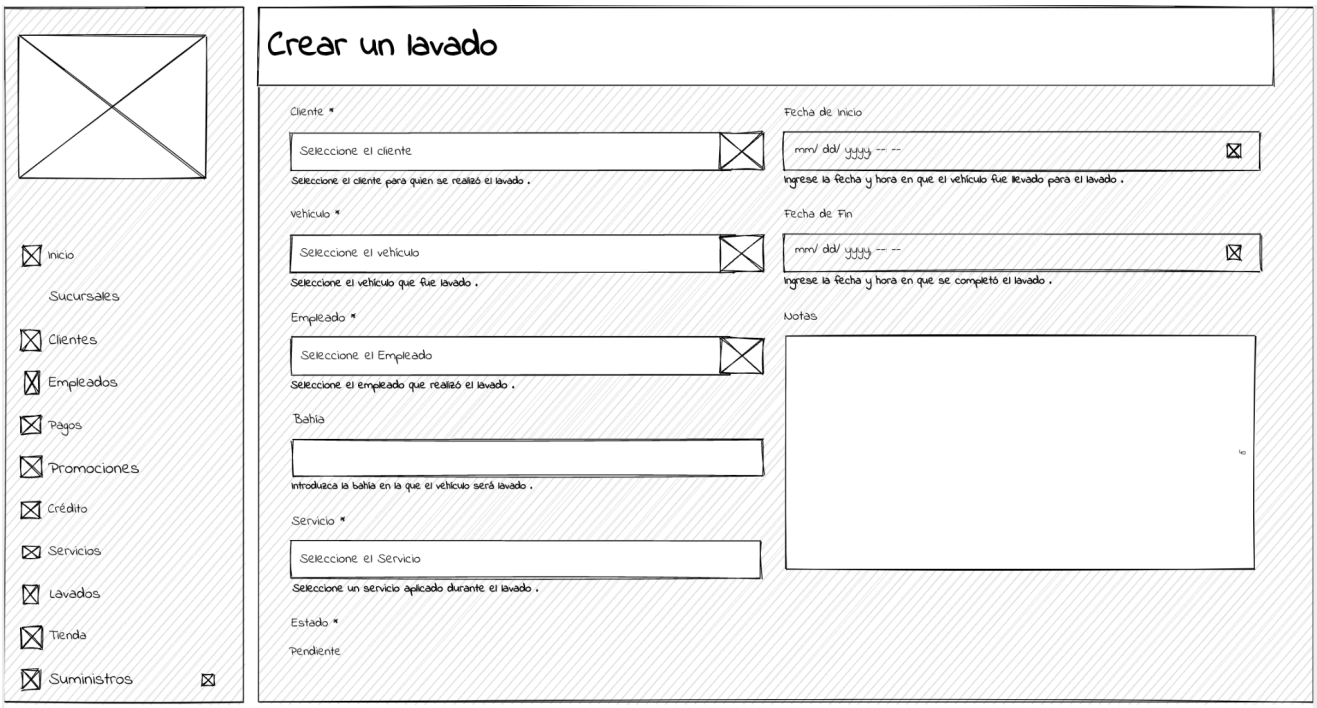
Wireframe de diseño del dashboard



- **Uniformidad y consistencia**: la interfaz debe mantener un formato homogéneo en comandos y menús para reducir el tiempo de aprendizaje.
- **Mínima sorpresa**: se deben evitar situaciones inesperadas que puedan generar irritabilidad o pérdida de interés en el usuario.
- **Asistencia al usuario**: la interfaz debe proporcionar ayuda clara e integrada al sistema, sin saturar con información.

Figura 14

Wireframe de diseño del formulario de lavados



- **Perspectiva del usuario**: se debe evaluar la interfaz desde el punto de vista del usuario y las tareas que realizará.
- **Realimentación:** la interfaz debe proporcionar respuestas inmediatas a las acciones del usuario.
- **Anticipación:** la aplicación debe prever las necesidades del usuario y ofrecer información relevante.
- **Percepción de color y tamaño:** se deben considerar usuarios con problemas de visualización del color y distribuir objetos en la pantalla para evitar la sobrecarga visual.

Figura 15

Paleta de colores usada en el sistema

- Legibilidad: el tipo, tamaño y color de letra deben ser legibles y contrastar con el fondo.
- Eficiencia: la interfaz debe ser eficiente y rápida, con mensajes de ayuda, menús y etiquetas sencillos y claros.

Diseño Adaptable

El diseño adaptable o responsive se refiere a la creación de interfaces de usuario que se ajustan automáticamente al tamaño y orientación de la pantalla del dispositivo en el que se visualizan, ofreciendo una experiencia de usuario óptima en diferentes dispositivos, como teléfonos inteligentes, tabletas y computadoras[14]. Este enfoque de diseño es fundamental en la era actual, donde los usuarios acceden a aplicaciones y sitios web desde una amplia variedad de dispositivos con diferentes características de pantalla.

En nuestro sistema, se implementó un diseño adaptable utilizando Bootstrap 5, un marco de diseño ampliamente utilizado y popular que facilita la creación de interfaces de usuario responsive y accesibles[15]. Bootstrap proporciona una serie de componentes predefinidos y una estructura de cuadrícula flexible que permite a los desarrolladores adaptar fácilmente la interfaz a diferentes tamaños de pantalla y resoluciones. Al emplear las clases y componentes de Bootstrap

[14] Marcotte, Ethan. (2010). Responsive Web Design. A List Apart. Recuperado de: https://alistapart.com/article/responsive-web-design/
[15] Bootstrap. (2021). Bootstrap 5. Recuperado de: https://getbootstrap.com/docs/5.0/getting-started/introduction/

5, nuestro sistema garantiza una experiencia de usuario coherente y funcional en todos los dispositivos.

Figura 16

Diseño responsive de recibo de pago en el sistema

xii. Estrategia de implementación

Para llevar a cabo pruebas y validaciones destinadas al cliente, se adoptó una estrategia de implementación fundamentada en las tecnologías empleadas durante el desarrollo del sistema. **(Consultar Sección vi: Factibilidad Técnica)**.

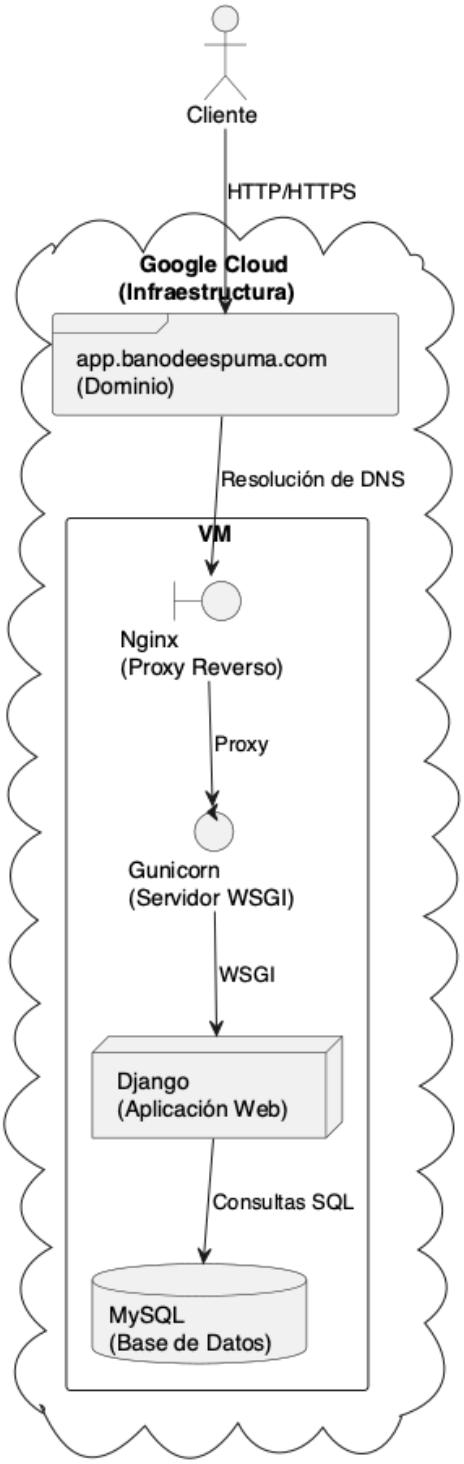
Esta metodología de implementación condujo al éxito en la creación de una aplicación robusta y confiable, ubicada en un entorno de producción en la nube. Además, se utilizó un dominio para facilitar el acceso a la aplicación.

Los pasos abordados incluyeron:

1. Se seleccionó Google Cloud como plataforma en la nube para alojar la aplicación, aprovechando sus servicios y recursos para garantizar una infraestructura escalable y confiable.
2. Se creó y configuró una máquina virtual en Google Cloud, que permite alojar y ejecutar la aplicación de manera eficiente.
3. Se configuró el entorno de desarrollo en la máquina virtual, instalando y ajustando las dependencias necesarias para Python, Django, Gunicorn, Nginx y MySQL.
4. Se estableció la base de datos MySQL, diseñando y optimizando el esquema de la base de datos para garantizar un rendimiento adecuado y la integridad de los datos.
5. Se implementó Gunicorn como servidor de aplicaciones WSGI, lo que facilitó la comunicación entre la aplicación Django y el servidor web Nginx.
6. Se configuró Nginx como servidor web y proxy inverso, permitiendo una distribución eficiente de la carga y una mayor seguridad en las comunicaciones entre el cliente y la aplicación.
7. Se asoció el dominio de Google app.banodeespuma.com con la máquina virtual en Google Cloud, asegurando que los usuarios pudieran acceder a la aplicación a través de una URL amigable y fácil de recordar.
8. Se realizaron pruebas exhaustivas y se monitoreo la aplicación en la nube, lo que permitió identificar y solucionar problemas rápidamente, garantizando un rendimiento óptimo y una experiencia de usuario satisfactoria.
9. Se configuró una estrategía de respaldos de base de datos, en base a las opciones disponibles en Google Cloud para instancias de bases de datos MySQL

Figura 17

Diagrama de implementación



xiii. Estrategia de implantación

Cabe destacar que, aunque el software ha sido desarrollado y se encuentra en funcionamiento en un entorno de pruebas, no se llevará a cabo la implantación al cliente de inmediato, ya que este no está en disposición de adoptarlo en este momento. Además, siendo un proyecto monográfico, la implantación está limitada a los requerimientos funcionales establecidos previamente. No obstante, la estrategia de implantación propuesta se describe a continuación y queda a disposición del cliente cuando este considere que el software cumple con todos los requisitos funcionales y esté preparado para la implantación.

1. **Capacitación del personal**: Proporcionar capacitación a los empleados del cliente sobre cómo utilizar el software correctamente, familiarizándolos con las funciones y características disponibles.
2. **Integración de datos**: Importar los datos reales del cliente al sistema, asegurando la integridad y precisión de la información. Esto incluye la migración de datos de sistemas antiguos, si es necesario.
3. **Prueba en un entorno real**: Realizar pruebas del software en el entorno real del cliente, utilizando procedimientos reales y monitoreando su rendimiento para identificar y solucionar problemas antes de la implementación completa.
4. **Implementación gradual**: Comenzar con una implementación gradual del software en las operaciones del cliente, permitiendo que los empleados se adapten a los cambios y asegurando una transición sin problemas.
5. **Soporte y actualizaciones automáticas**: Las actualizaciones automáticas del software están configuradas en Google Cloud, garantizando que el sistema se mantenga actualizado y seguro. Se transferirán todos los accesos al proyecto en la nube al cliente, quien deberá asumir los costos asociados.
6. **Liberación del repositorio de código**: Se liberará el repositorio del código con una licencia de software libre, permitiendo al cliente adaptar y mejorar el software según sus necesidades futuras.
7. **Manuales de Usuario y desarrollador:** Se entregarán los manuales de usuario y desarrollador al cliente tanto en versión digital como física, lo que facilitará a nuevos

desarrolladores y analistas capacitarse para realizar mantenimientos y mejoras al software.

8. **Contrato de mantenimiento**: Se discutirá internamente la posibilidad de establecer un contrato de mantenimiento con el cliente, con el fin de ofrecer soporte técnico continuo y garantizar el correcto funcionamiento del software a largo plazo.

xiv. Conclusiones del capítulo

Luego de realizar el presente diseño y analizar cada uno de los factores que influyen en este, se obtuvo como resultado que el desarrollo del SISTEMA WEB DE GESTIÓN DE INVENTARIO Y REGISTRO DE ACTIVIDADES EN AUTOLAVADO "BAÑO DE ESPUMA", es de fácil manejo, sencillo pero robusto, accesible a los usuarios que harán uso del mismo, sobre todo cumple con cada uno de los requerimientos realizados por el cliente.

VIII. Conclusiones

Con la finalización del SISTEMA WEB DE GESTIÓN DE INVENTARIO Y REGISTRO DE ACTIVIDADES EN AUTOLAVADO "BAÑO DE ESPUMA", el cual ha sido desarrollado como trabajo monográfico, se cumple con una de las necesidades que la empresa tenía desde hace un buen tiempo, ajustándose a lo solicitado por los clientes y tomando en consideración cada una de las áreas involucradas, en el levantamiento de requerimientos, a partir de los cuales se desarrollan los diagramas correspondientes a la metodología de análisis y diseño orientada a objetos.

Con una herramienta que gestione la información de forma centralizada, se garantiza que los datos ingresados en los procesos de control de inventario, registro de clientes, servicios ofrecidos y sobre todo poder ser utilizado en el mismo momento en ambas sucursales, viniendo a solventar una de las principales dificultades que fueron presentadas por el cliente.

El sistema fue diseñado para agilizar y mejorar los procesos relacionados al control de inventario, así como del lavado de autos, para Baño de Espuma, sin embargo, el sistema se puede adaptar a las necesidades de otro autolavado con características similares a este.

IX. Recomendaciones

- El sistema desarrollado se encuentra implementado en un entorno de pruebas, basado en los requerimientos del cliente, se recomienda agregar nuevos módulos para mejorar la funcionalidad e inclusión de información de otras áreas de la empresa, como por ejemplo la parte de control de personal e incluso nómina.

- Se recomienda utilizar el navegador Google Chrome, dado que este fue el navegador web que se utilizó en toda la fase de desarrollo y prueba, otros navegadores webs modernos, basados en el motor webkit funcionan sin ningún problema.

- Es recomendable el uso de tabletas con pantalla de 7 pulgadas, con una resolución mínima de 1024x600 pixeles, para poder visualizar la información de manera cómoda.

- Durante el desarrollo del sistema, se ingresó información con el propósito de llevar a cabo pruebas de su funcionamiento. Sin embargo, esta información no es suficiente para el uso real del sistema por parte de Baño de Espuma. Es necesario ingresar todo el inventario disponible en la empresa, así como registrar a todos los clientes existentes. De esta manera, se garantiza que el sistema esté completo y listo para su uso efectivo en las operaciones diarias de la empresa.

- Programar un día de capacitación integral para todo el personal: Es esencial que todos los empleados estén familiarizados y capacitados en el manejo del nuevo sistema. Por lo tanto, se aconseja designar un día para que los colaboradores puedan recibir una formación exhaustiva que cubra todas las funciones y características del sistema.

- Dividir la capacitación en sesiones teóricas y prácticas: Para garantizar una comprensión completa del sistema, es útil dividir la capacitación en dos partes. En la primera, los empleados recibirán información teórica sobre el funcionamiento y las características del

sistema. En la segunda parte, se les permitirá a los empleados interactuar con el sistema y poner en práctica lo aprendido en un entorno controlado.

- Designar un día por sucursal para pruebas de implementación: Una vez que el personal haya sido capacitado, es fundamental asegurarse de que el sistema funcione correctamente en cada una de las sucursales. Por lo tanto, se recomienda asignar un día específico para realizar pruebas de implementación en cada sucursal, lo que permitirá detectar y solucionar cualquier problema o inconveniente antes de su puesta en marcha oficial.

X. Bibliografía

Andreu, R., & Ricart, J. (1996). *Estrategias y Sistemas de Información*. Madrid: McGraw-Hill.

Arias, R. (2019). *Aprende Python en un fin de semana*. México D.F.: Ra-Ma Editorial.

Baca Urbina, G. (2014). *Adminitración Informática I*. México DF: Grupo Editorial Patria.

Beynon-Davies, P. (2003). *Database Systems (Tercera ed.)*. Palgrave.

Booch, G. (1999). *Guía de usuario del Lenguaje Unificado de Modelado (Segunda ed.)*. Pearson Education.

Brewer, J. (2015). *HTML and CSS: diseño y creación de páginas web*. Anaya Multimedia.

Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP y MySQL: Tecnologías para el desarrollo de aplicaciones web*. Madrid: España.

Conde, J. (2014). *Desarrollo web con PHP y MySQL*. México D.F.: Ra-Ma Editorial.

DataShop. (20 de 11 de 2023). *DataShop Software.* Obtenido de https://www.datashop.com.mx/

Duckett, J. (2011). *HTML & CSS: Diseño y Construcción de Sitios Web.* Indianápolis, Indiana, Estados Unidos: John Wiley & Sons, Inc.

Eguíluz Pérez, J. (2008). *Introducción a JavaScript*. Madrid: Creative Commons.

Flanagan, D. (2011). *JavaScript: the definitive guide*. O'Reilly Media.

García, R. M. (2014). *Diseño de interfaces web*. Editorial UOC.

Malhotra, N. K. (2010). *Investigación de mercados: Un enfoque aplicado*. México D.F.: Pearson.

G. Bautista, M. Á. (2019). *Aprende jQuery en un fin de semana*. México D.F.: Ra-Ma Editorial.

Gallego Sánchez, A. J. (2013). *Bootstrap 3: Introducción al diseño Responsive.*

Gutiérrez, D. (2019). *Introducción a la programación con Python*. México D.F.: Universidad Nacional Autónoma de México.

JQuery. (06 de 02 de 2023). *What is JQuery.* Obtenido de https://jquery.com/

Kendall, K. E., & Kendall, J. E. (2011). *Análisis y diseño de sistemas*. Pearson Educación.

Larman, C. (1999). *UML y Patrones: Introducción al análisis y diseño Orientado a Objetos*. Naucalpan de Juárez, Estado de México, México: Prentice Hall.

López Rosciano, R. A., & Pech Montejo, J. A. (2015). *Desarrollo de herramientas de gestión de proyectos RUP usando metodología SCRUM + XP: Pruebas*. Madrid: Universidad Politécnica de Madrid.

Lozano, L. (viernes de septiembre de 2013). *Estándares de calidad del software.* Obtenido de http://estandarescalidadsoftware.blogspot.com/

Montgomery, D. C. (2013). *Diseño y análisis de experimentos (8a ed.)*. México D.F.: Limusa.

Mozilla Web Docs. (06 de 02 de 2023) *What is Django.* Obtenido de https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction#what_is_django

MySQL. (20 de 11 de 2020). *Manual de referencia de MySQL 8.0.* Obtenido de https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Pilone, D., & Pitman, N. (2005). *UML 2.0 in a Nutshell: A desktop quick reference.* Sebastopol, California, Estados Unidos: O'Reilly.

Pressman, R. (2002). *Ingeniería del Software: un enfoque práctico (Quinta ed.).* McGraw Hill.

Pressman, R. S. (2010). *Ingeniería del software un enfoque práctico*. McGraw-Hill Education.

Python. (06 de 02 de 2023) *What is Python. Executive Summary.* Obtenido de https://www.python.org/doc/essays/blurb/

Saldaña, g. (s.f.). introduccion a GIT.

Scrum. (06 de 02 de 2023) *What is Scrum.* Obtenido de https://www.scrum.org/resources/what-is-scrum

Sommerville, I. (2005). *Ingeniería del Software.* Madrid: Pearson.

Sommerville, I. (2007). *Ingeniería del Software.* Madrid: Pearson Educación.

Straub, S. C. (2014). *Pro Git.*

Thibaud, C. (2006). *MySQL 5.* Barcelona: Ediciones ENI.

Torre, G. S., & Codner, D. G. (2013). *Fundamentos del comercio electrónico (Segunda ed.)*. Argentina: Universidad Virtual de Quilmes.

W3C. (11 de 02 de 2021). *HTML y CSS - W3C*. Obtenido de https://www.w3.org/standards/webdesign/htmlcss#whatc

XI. Glosario

AJAX: Técnica de desarrollo web que permite actualizar partes de una página sin recargar completamente, mejorando la interacción del usuario.

Bootstrap 5: Versión 5 de Bootstrap, un popular framework de diseño para construir sitios web responsivos y adaptativos.

CGI: Common Gateway Interface, un protocolo que permite a los servidores web ejecutar programas externos para generar contenido dinámico.

DNS: Domain Name System, un sistema que traduce nombres de dominio a direcciones IP y viceversa.

Docker: Plataforma de contenedores que permite empaquetar y ejecutar aplicaciones de forma aislada y eficiente.

DOM: Document Object Model, una representación estructurada de un documento HTML o XML que permite su manipulación con lenguajes de programación.

Fontawesome: Biblioteca de iconos escalables y personalizables para aplicaciones web.

Framework: Estructura o conjunto de herramientas y bibliotecas que facilitan el desarrollo de aplicaciones en un lenguaje de programación específico.

Gunicorn: Servidor de aplicaciones WSGI para Python que permite ejecutar aplicaciones web basadas en este lenguaje.

GUI: Graphical User Interface, interfaz gráfica que permite a los usuarios interactuar con dispositivos y software mediante elementos visuales.

Hardware: Componentes físicos de un sistema informático, como CPU, memoria y dispositivos de entrada y salida.

HTTP: Hypertext Transfer Protocol, protocolo de comunicación utilizado en la World Wide Web para transferir información entre clientes y servidores.

JavaScript: Lenguaje de programación utilizado principalmente para agregar interactividad y dinamismo a las páginas web.

Jinja: Motor de plantillas para Python, utilizado en el desarrollo web para generar contenido HTML dinámico.

MVC: Model-View-Controller, un patrón de diseño de software que separa la lógica de negocio, la presentación y la interacción del usuario.

Máquina virtual (VM): Emulación de un sistema informático que permite ejecutar programas en un entorno aislado y controlado.

Nginx: Servidor web y proxy inverso de alto rendimiento utilizado para servir contenido estático y balancear la carga.

ORM: Object-Relational Mapping, técnica que facilita la interacción entre una base de datos y un lenguaje de programación orientado a objetos.

PDF: Portable Document Format, formato de archivo utilizado para representar documentos de manera independiente del software, hardware y sistema operativo.

Proxy: Intermediario entre un cliente y un servidor que permite filtrar y modificar las solicitudes y respuestas entre ellos.

Scrum: Marco de trabajo ágil para la gestión de proyectos de software que se basa en iteraciones cortas llamadas sprints.

Software: Programas y aplicaciones que permiten a los usuarios interactuar con el hardware y realizar tareas específicas.

Software Enlatado: Software comercial preempaquetado que está diseñado para satisfacer las necesidades de un amplio grupo de usuarios.

Sistema Web: Aplicación informática que se ejecuta en un servidor y es accesible a través de un navegador web.

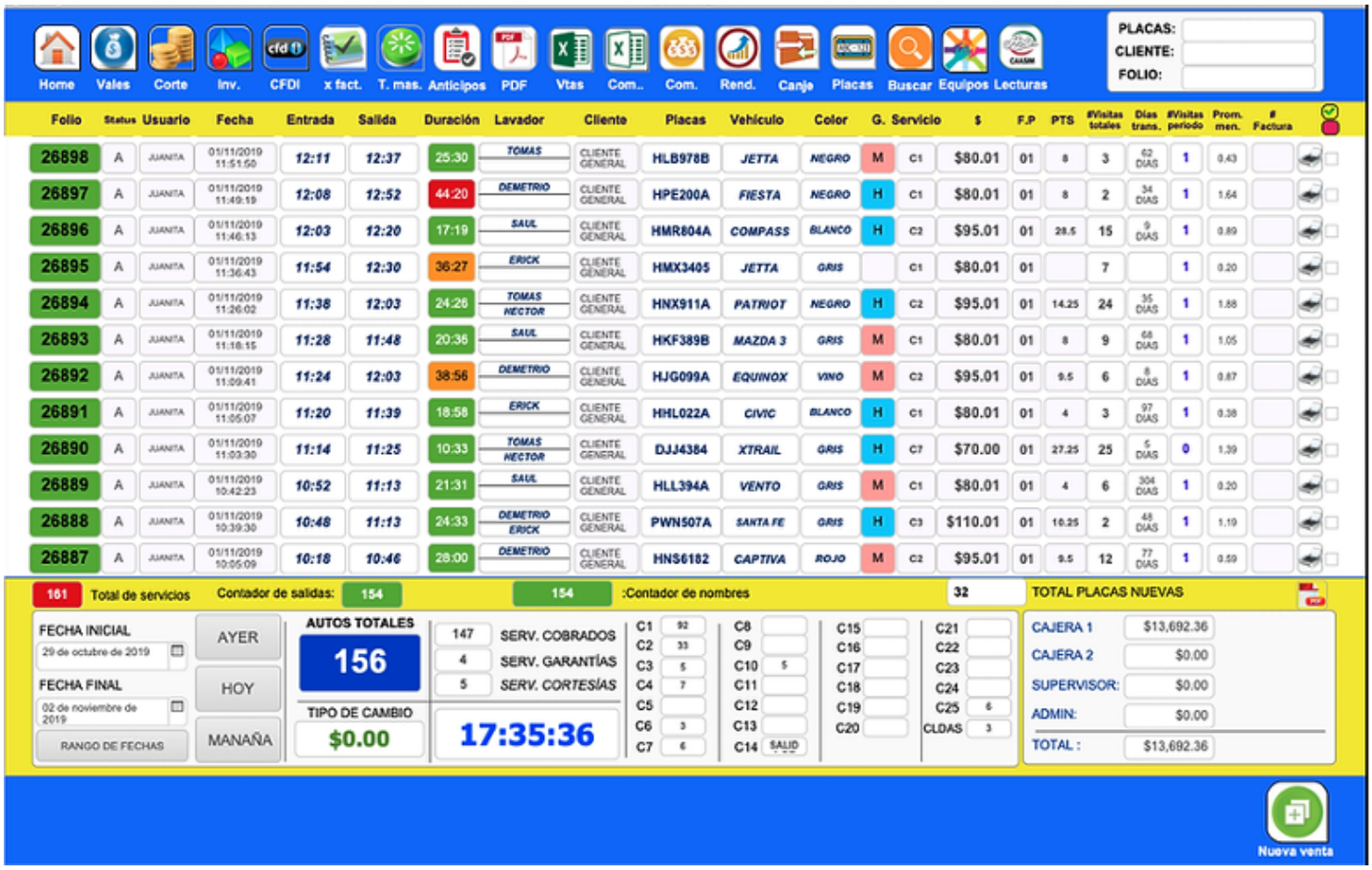
Sprint: Iteración corta y fija en el proceso de desarrollo de software dentro del marco de trabajo Scrum.

SQL: Structured Query Language, lenguaje de programación utilizado para gestionar y manipular bases de datos relacionales.

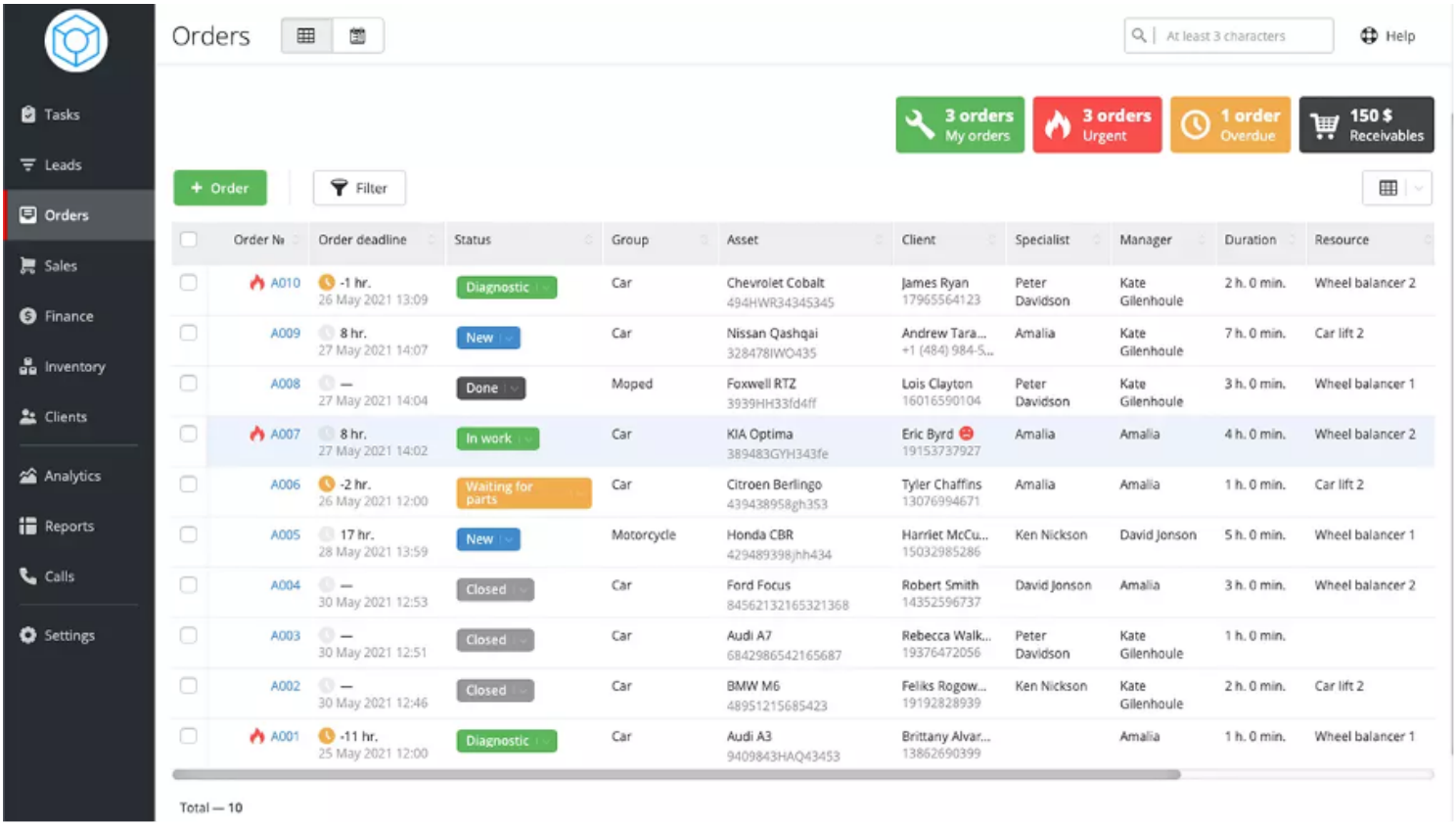
UML: Unified Modeling Language, lenguaje

XII. Anexos

Anexo 1 : Registro de lavados Software DataShop, pantalla del sistema actual.



Anexo 2 : Registro de actividades Software Orderry



Anexo 3: Resumen Entrevista con Karla Garay, dueña de Autolavado “Baño de Espuma”

¿En el sistema actual que estás utilizando tenés de alguna forma como hacer citas? ¿Eso sería una funcionalidad que te serviría?

En la entrevista realizada con el dueño del local, se pudo confirmar que en el sistema actual no existe una función para hacer citas. El dueño del local explicó que no ve necesaria esta función debido a que para hacer citas se necesitaría tener un servicio especial con una tarifa especial y dedicar una bahía para que esté disponible solo para la persona que hizo la cita. Además, mencionó que es difícil que la gente quiera pagar más por la exclusividad de que le laven su auto en cuanto llegue y que prefieren utilizar las bahías disponibles sin cita previa.

¿Coméntanos qué funcionalidades usas ahorita de tu sistema?

En el sistema actual, se pueden ingresar los vehículos y registrar sus datos como marca, modelo, tipo de servicio y precio asociado. Además, se puede llevar un registro de los clientes, incluyendo su nombre, teléfono, dirección e identificación, lo que permite asociar los vehículos con los clientes para fines de reportes. También se pueden controlar los insumos utilizados para los lavados y las ventas de productos de la cafetería, como gaseosas y chiverías. El sistema también lleva un control del personal, incluyendo su fecha de inicio y el histórico de vehículos que ha lavado cada uno. Además, se pueden implementar promociones y descuentos, como la hora feliz o visitas gratuitas por cada cierto número de visitas. El sistema genera reportes de todo tipo, incluyendo el cálculo de los pagos correspondientes a los lavadores en caso de ser necesario. También tiene opciones de rifa y sorteos.

¿Todas estas funcionalidades son las funcionalidades que están usando?

Sí, todas las funcionalidades mencionadas son las que se están utilizando actualmente, excepto la venta de productos de cafetería y la parte de los costos de los insumos. El dueño del local indica que aún no utiliza la venta de productos de cafetería con el sistema porque necesitaría comprar

una maquinita para el lector de código de barras y hacerlo más rápido. También menciona que no está utilizando la parte de los costos de los insumos.

¿Cómo se manejan los pagos con tarjeta?

El sistema tiene una opción para registrar la forma de pago, ya sea en efectivo, con tarjeta o a crédito. Si el cliente paga con tarjeta, el sistema envía la información a otro módulo de cuentas por cobrar para llevar un registro de los montos adeudados por cada cliente. El pago con tarjeta se realiza a través del POS del banco.

¿Funcionaría que los clientes pudieran dejar sus comentarios, sugerencias?

Sí, sería una buena idea implementar una función en el sistema para que los clientes puedan dejar sus comentarios y sugerencias asociados a su número de ticket. Esto facilitaría la recopilación de feedback de los clientes y permitiría a la empresa conocer las opiniones de los clientes de manera más rápida y eficiente. Además, el sistema podría mostrar estadísticas sobre la frecuencia de comentarios y sugerencias, lo que permitiría a la empresa dar prioridad a los problemas o sugerencias más relevantes.

¿Con qué problemas problemas te has encontrado en el sistema actual que sentís que debería de mejorarse?

El principal problema que se ha identificado es la falta de una opción para controlar la bahía, es decir, el espacio físico en el que se atiende el vehículo. Esto dificulta el seguimiento y la resolución de ciertos problemas que puedan surgir durante el proceso de atención al cliente.

Una opción útil sería tener la capacidad de agregar servicios adicionales a un ticket ya pagado, sin tener que eliminar el registro anterior y volver a ingresarlo todo. Sería conveniente tener algún tipo de autorización o permiso para editar el servicio, quizás por parte de un supervisor o administrador, para evitar que el cajero lo edite sin permiso. De esta manera, se podría mantener

el conteo por carro y no por servicio, incluso si se agrega un servicio adicional a un ticket ya pagado.

¿Ya mencionaste algo sobre la hora que comienza a lavarse y la hora que termina. ¿Tenés algún reporte de performance de los trabajadores respecto a eso?

Es posible que el sistema no tenga un reporte específico para medir el desempeño de los trabajadores en términos de los tiempos que toman para lavar los vehículos. Pero podrías revisar los reportes de ingresos generados por los trabajadores para tener una idea general de su desempeño.

Por ejemplo, establecer que una camioneta la lavan en 40 minutos como máximo, entonces si una persona dura una hora ya tenés ahí un asterisco del por qué se demoraron tanto.

Sí, establecer tiempos máximos para cada tipo de vehículo y monitorear el tiempo que tarda cada trabajador en lavar un vehículo puede ser útil para identificar posibles problemas y áreas de mejora en el proceso de lavado. Además, poder asociar los tiempos con los servicios adicionales que se ofrecen al cliente puede ayudar a entender mejor por qué se tardó más de lo esperado en lavar un vehículo en particular.

Respecto a los roles ¿Quiénes son los que trabajan con el sistema?

El sistema tiene varios roles disponibles, como el de administrador, cajero, supervisor, entre otros, pero en la práctica, solo se utilizan el rol de administrador y el de cajero. También hay diferentes niveles de permisos dentro de los roles, como cajero uno, cajero dos y cajero administrador, pero en la práctica, solo hay un cajero de planta y otro que cubre cuando el primero está ausente. Cuando se necesita hacer algo que está fuera de las capacidades del cajero, el administrador accede al sistema de forma remota y realiza las acciones necesarias. En resumen, hay tres roles disponibles en el sistema, pero en la práctica, solo se utilizan dos de ellos.

Ya que se menciona el término "remoto", ¿Es requerido que vos, como dueña del negocio puedas acceder desde tu casa?

El sistema sí tiene la opción de permitir que ingresen de forma remota desde tu casa, pero es posible que solo puedas hacerlo si estás en la misma red que la computadora en la que se encuentra el sistema. Sin embargo, esto dependería de la configuración específica del sistema y de la red en la que se encuentra. Sería necesario verificar con los encargados del sistema si es posible acceder de forma remota desde otra ubicación.

¿Qué modalidad te funciona mejor, remota o desde la oficina del autolavado?

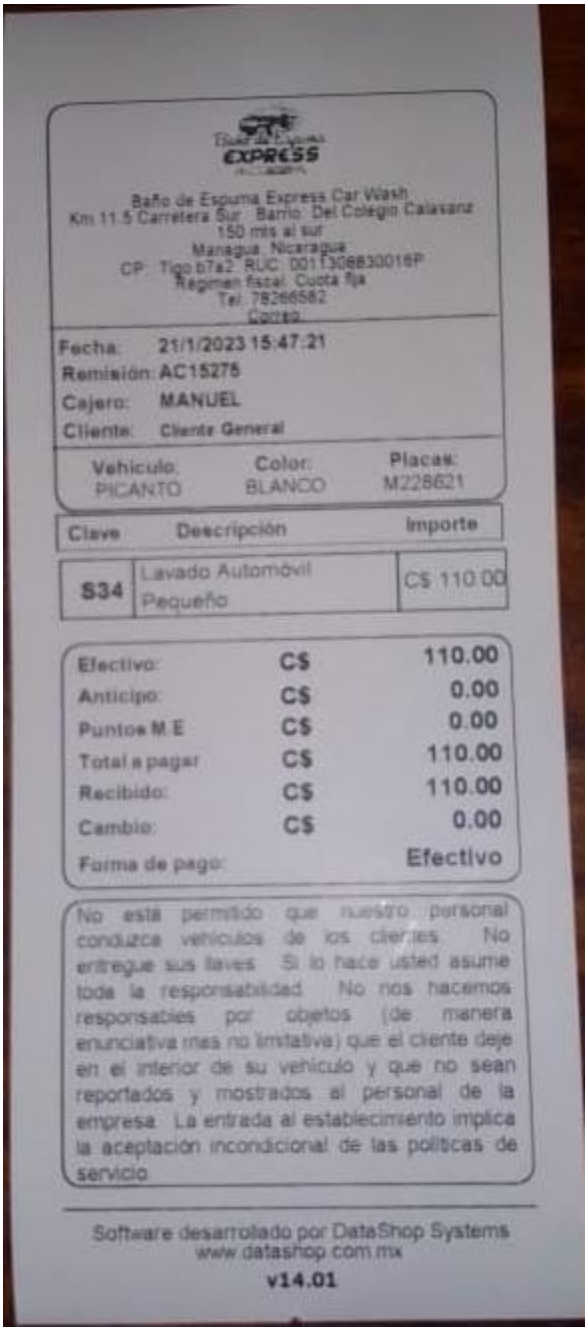
La modalidad de acceso remoto parece ser la más conveniente para poder manejar la información de ambos locales de manera centralizada. Además, sería beneficioso tener un sistema que permita la integración de las bases de datos de ambos locales para tener una visión completa de la información de los clientes.

¿Es posible acceder al sistema desde dispositivos móviles como teléfonos o tabletas para ingresar datos y hacer transacciones de forma más conveniente?

Esto sería útil para los cajeros que necesitan estar afuera en la Bahía para revisar los vehículos y las placas mientras ingresan los datos en el sistema.

Claro, lo que hacemos a nivel técnico es desarrollar el sitio para que se vea bien en todos los dispositivos. Precisamente esa es una de las ventajas de tener una aplicación web. **¿Tienen alguna otra pregunta?** Creo que eso sería todo. Muchas gracias y buenas noches

Anexo 4: Factura de Ejemplo Baño de Espuma



EXPRESS

Baño de Espuma Express Car Wash
Km 11.5 Carretera Sur Barrio: Del Colegio Calasanz
150 mts al sur
Managua, Nicaragua
CP: Tigo b7a2 RUC: 0011308830016P
Regimen fiscal: Cuota fija
Tel: 78266582
Correo:

Fecha: 21/1/2023 15:47:21
Remisión: AC15275
Cajero: MANUEL
Cliente: Cliente General

Vehiculo:	Color:	Placas:
PICANTO	BLANCO	M228621

Clave	Descripción	Importe
S34	Lavado Automóvil Pequeño	C$ 110.00

Efectivo:	C$	110.00
Anticipo:	C$	0.00
Puntos M E	C$	0.00
Total a pagar	C$	110.00
Recibido:	C$	110.00
Cambio:	C$	0.00
Forma de pago:		Efectivo

No está permitido que nuestro personal conduzca vehículos de los clientes. No entregue sus llaves. Si lo hace usted asume toda la responsabilidad. No nos hacemos responsables por objetos (de manera enunciativa mas no limitativa) que el cliente deje en el interior de su vehículo y que no sean reportados y mostrados al personal de la empresa. La entrada al establecimiento implica la aceptación incondicional de las politicas de servicio

Software desarrollado por DataShop Systems
www.datashop.com.mx
v14.01

Anexo 5: Diccionario de Datos, Sistema web de gestión de inventario y registro de actividades en autolavado “Baño de espuma”

1. Modelos Abstractos

SoftDeleteModel			
Modelo de borrado suave			
Campo	**Tipo**	**Descripción**	**Nullable**
is_deleted	BooleanField	Indica si el objeto ha sido eliminado lógicamente	No
deleted_at	DateTimeField	La fecha y hora en que se eliminó lógicamente el objeto	Sí

TimeStampedModel			
Modelo con marca de tiempo			
Campo	**Tipo**	**Descripción**	**Nullable**
created_at	DateTimeField	La fecha y hora de creación del objeto	No
updated_at	DateTimeField	La fecha y hora de la última modificación del objeto	Sí
created_by	ForeignKey	El usuario que creó el objeto	Sí
updated_by	ForeignKey	El usuario que modificó por última vez el objeto	Sí

2. Sucursales

Branch (SoftDeleteModel, TimeStampedModel)			
Catálogo de Sucursales			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	El nombre de la sucursal	No
address	CharField	La dirección de la sucursal	Sí
department	IntegerField	El departamento donde se encuentra la sucursal, se proporciona una lista de opciones a través de la variable `DEPARTMENTS`	No
phone	CharField	El número de teléfono de la sucursal	Sí
email	EmailField	La dirección de correo electrónico de la sucursal	Sí
tax_number	CharField	El número de identificación fiscal de la sucursal	Sí
created	DateTimeField	La fecha y hora de creación de la sucursal	No
modified	DateTimeField	La fecha y hora de la última modificación de la sucursal	No

is_deleted	BooleanField	Indica si la sucursal ha sido eliminada lógicamente	No
deleted_at	DateTimeField	La fecha y hora en que se eliminó lógicamente la sucursal	Sí

3. Crédito

Credit (SoftDeleteModel, TimeStampedModel)			
Catálogo de créditos			
Campo	**Tipo**	**Descripción**	**Nullable**
customer	ForeignKey	El cliente asociado con el crédito	No
description	CharField	Una descripción opcional del crédito	Sí
status	CharField	El estado actual del crédito se proporciona una lista de opciones a través del campo `CreditStatus.choices`	No
amount	DecimalField	La cantidad total del crédito	No
interest_rate	DecimalField	La tasa de interés del crédito como un porcentaje	Sí
start_date	DateField	La fecha en que se otorgó el crédito	No

end_date	DateField	La fecha en que el crédito debe estar completamente pagado	No
balance	DecimalField	El saldo pendiente del crédito	Sí
branch	ForeignKey	La sucursal donde se realizó el pago si corresponde	Sí

4. Clientes

Customer (SoftDeleteModel, TimeStampedModel)			
Catálogo de Clientes			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	El nombre del cliente	No
customer_type	CharField	El tipo de cliente se proporciona una lista de opciones a través del campo `CustomerTypes.choices`	No
phone_number	CharField	El número de teléfono del cliente	Sí
email	EmailField	La dirección de correo electrónico del cliente	Sí
address	CharField	La dirección del cliente	Sí
tax_number	CharField	El número de impuestos del cliente	Sí

tax_exempted	BooleanField	Indica si el cliente está exento de impuestos	No

5. Empleados

Employee (SoftDeleteModel, TimeStampedModel)			
Catálogo de Empleados			
Campo	**Tipo**	**Descripción**	**Nullable**
user	OneToOneField	El usuario asociado al empleado	No
name	CharField	El nombre del empleado	No
email	EmailField	La dirección de correo electrónico del empleado. Si no hay dirección de correo electrónico, dejar en blanco.	Sí
phone	CharField	El número de teléfono del empleado. Si no hay número de teléfono, dejar en blanco.	Sí
position	CharField	La posición del empleado, se proporciona una lista de opciones a través del campo `Positions.choices`	No

salary	DecimalField	El salario del empleado. Si el salario no ha sido asignado, dejar en blanco.	Sí
start_date	DateTimeField	La fecha en que el empleado comenzó su trabajo. Si la fecha de inicio no se ha establecido, dejar en blanco.	Sí
default_branch	ForeignKey	La sucursal predeterminada del empleado. Si la sucursal predeterminada no se ha establecido, dejar en blanco.	Sí
branches	ManyToManyField	Las sucursales asociadas al empleado. Si el empleado no está asociado con ninguna sucursal, dejar en blanco.	No

6. Pagos

Promotion (SoftDeleteModel, TimeStampedModel)			
Catálogo de Promociones			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	El nombre de la promoción	No
branches	ManyToManyField	Las sucursales donde se aplica esta	No

		promoción	
description	TextField	Una descripción de la promoción	No
code	CharField	El código de la promoción se puede, dejar en blanco	Sí
discount_percentage	DecimalField	El descuento como porcentaje	No
start_date	DateTimeField	La fecha de inicio de la promoción	No
end_date	DateTimeField	La fecha de finalización de la promoción	No
is_active	BooleanField	Indica si la promoción está activa o no	No
promotion_type	CharField	El tipo de promoción. Se proporciona una lista de opciones a través del campo `PromotionTypes.choices`	No

Payment (SoftDeleteModel, TimeStampedModel)			
Catálogo de Pagos			
Campo	**Tipo**	**Descripción**	**Nullable**
description	CharField	Una descripción del pago, se puede dejar en blanco	Sí
status	CharField	El estado del pago. Se proporciona una lista de opciones a través del campo `PaymentStatus.choices`	No

source	CharField	El origen del pago. Se proporciona una lista de opciones a través del campo `PaymentSource.choices`	No
customer	ForeignKey	El cliente que hizo el pago	No
branch	ForeignKey	La sucursal donde se hizo el pago	No
amount	DecimalField	La cantidad del pago	No
discount_amount	DecimalField	La cantidad del descuento del pago, se puede dejar en blanco	Sí
taxes_amount	DecimalField	La cantidad de los impuestos, se puede dejar en blanco	Sí
total_amount	DecimalField	El total del pago después de los impuestos y descuentos, se puede dejar en blanco	Sí
payment_method	CharField	El método de pago. Se proporciona una lista de opciones a través del campo `PaymentMethod.choices`	No
notes	TextField	Notas adicionales sobre el pago, se puede dejar en blanco	Sí
promotions	ManyToManyField	Las promociones aplicadas al pago, se puede dejar en blanco	Sí

wash	OneToOneField	La lavada asociada con el pago, se puede dejar en blanco	Sí
credit	ForeignKey	El crédito asociado con el pago, se puede dejar en blanco	Sí

7. Servicios

Service (SoftDeleteModel, TimeStampedModel)			
Catálogo de Servicios			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	Nombre del servicio	No
description	TextField	Descripción del servicio	No
price	DecimalField	Precio del servicio	No

ServiceSupply (SoftDeleteModel, TimeStampedModel)			
Tabla intermedia de Servicios con Suplementos			
Campo	**Tipo**	**Descripción**	**Nullable**
service	ForeignKey	Servicio	No
supply	ForeignKey	Suministro	No
quantity_per_service	FloatField	Cantidad por servicio	Sí

Wash (SoftDeleteModel, TimeStampedModel)			
Catálogo de Lavados			
Campo	**Tipo**	**Descripción**	**Nullable**
branch	ForeignKey	Sucursal donde se realizó el lavado	No

customer	ForeignKey	Cliente para quien se realizó el lavado	No
employee	ForeignKey	Empleado que realizó el lavado	No
vehicle	ForeignKey	Vehículo que se lavó	No
start_time	DateTimeField	Fecha y hora en que se llevó el vehículo para el lavado	Sí
bay	PositiveSmallIntegerField	Número de bahía donde se lavó el vehículo	Sí
end_time	DateTimeField	Fecha y hora en que se completó el lavado	Sí
service	ForeignKey	Servicio aplicado durante el lavado	No
notes	TextField	Notas sobre el lavado	Sí
status	CharField	Estado del lavado	No

WashStockItem (SoftDeleteModel, TimeStampedModel)			
Tabla intermedia entre Lavados y Artículos de Inventario			
Campo	**Tipo**	**Descripción**	**Nullable**
wash	ForeignKey	Lavado	No
stock_item	ForeignKey	Artículo del inventario utilizado durante el lavado	No
quantity	FloatField	Cantidad de artículo del inventario utilizado durante el lavado	No

8. Tienda

Item (SoftDeleteModel, TimeStampedModel)			
Catálogo de Productos de la tienda			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	Nombre del item	No
description	TextField	Descripción del item	No
price	DecimalField	Precio del item	No
image	ImageField	Imagen del item	Sí
category	CharField	Categoría del item	No

StoreStock (SoftDeleteModel, TimeStampedModel)			
Catálogo de Inventario de la tienda			
Campo	**Tipo**	**Descripción**	**Nullable**
item_id	IntegerField	ID del item en el stock	No
branch_id	IntegerField	ID de la sucursal donde se encuentra el stock	No
quantity	PositiveIntegerField	Cantidad de items en la sucursal	No

Sale (SoftDeleteModel, TimeStampedModel)			
Catálogo de ventas de la tienda			
Campo	**Tipo**	**Descripción**	**Nullable**
branch_id	IntegerField	ID de la sucursal donde se realizó la venta	No
date	DateTimeField	Fecha y hora en que se realizó la venta	No

payment_id	IntegerField	ID del pago realizado para la venta	Sí

SaleItem (SoftDeleteModel, TimeStampedModel)			
Tabla Intermedia entre Artículo y Venta			
Campo	**Tipo**	**Descripción**	**Nullable**
item_id	IntegerField	ID del item vendido	No
sale_id	IntegerField	ID de la venta en la que se vendió el item	No
quantity	PositiveIntegerField	Cantidad de items vendidos	No

9. Suministros

Supply (SoftDeleteModel, TimeStampedModel)			
Catálogo de Suministros			
Campo	**Tipo**	**Descripción**	**Nullable**
name	CharField	Nombre del suministro	No
photo	ImageField	Foto del suministro	Sí
description	TextField	Descripción del suministro	Sí
price	DecimalField	Precio unitario del suministro	Sí
measure	CharField	Unidad de medida del suministro	No
quantity_per_unit	FloatField	Cantidad por unidad del suministro	No

Stock (SoftDeleteModel, TimeStampedModel)			
Catálogo de Inventario de las suministros			
Campo	**Tipo**	**Descripción**	**Nullable**
supply	ForeignKey	Suministro en stock en esta sucursal	No
branch	ForeignKey	Sucursal donde está en stock el suministro	No
units	PositiveIntegerField	Unidades de suministro en stock en esta sucursal	Sí
total_quantity	FloatField	Suma de todas las cantidades por unidad en la medida del suministro	Sí
reorder_point	PositiveIntegerField	Punto de reorden de este suministro	No

StockItem (SoftDeleteModel, TimeStampedModel)			
Tabla Intermedia entre Inventario y Suministros			
Campo	**Tipo**	**Descripción**	**Nullable**
sku	CharField	Código identificador único del artículo	No
stock	ForeignKey	Stock asociado con este artículo	No
entry_date	DateTimeField	Fecha y hora de entrada del artículo en el stock	Sí
exit_date	DateTimeField	Fecha y hora de salida del artículo del stock	Sí
quantity	FloatField	Cantidad del artículo en el stock	Sí

status	CharField	Estado del artículo en el stock	Sí

10. Vehículos

Vehicle (SoftDeleteModel, TimeStampedModel)			
Catálogo de Vehículos			
Campo	**Tipo**	**Descripción**	**Nullable**
id	Integer	Identificador único del vehículo	No
vehicle_type	CharField	Tipo de vehículo	No
photo	ImageField	Imagen del vehículo	Sí
make	CharField	Marca del vehículo	No
color	CharField	Color del vehículo	No
model	CharField	Modelo del vehículo	Sí
year	PositiveSmallIntegerField	Año del vehículo	Sí
license_plate	CharField	Placa del vehículo	No
customer_id	ForeignKey	Identificador único del cliente que posee el vehículo	No

11. Retroalimentación

Feedback (SoftDeleteModel, TimeStampedModel)			
Catálogo de Retroalimentación			
Campo	**Tipo**	**Descripción**	**Nullable**
branch	ForeignKey	ID de la sucursal donde se recibió el feedback	No
customer	ForeignKey	ID del cliente que envió el feedback	No
feedback_type	CharField	Tipo de feedback	No
feedback_message	TextField	Mensaje de feedback	No
is_resolved	BooleanField	Indica si el feedback ha sido resuelto	No

FeedbackResponse (SoftDeleteModel, TimeStampedModel)			
Catálogo de Respuestas de Retroalimentación			
Campo	**Tipo**	**Descripción**	**Nullable**
feedback	ForeignKey	ID del feedback al que se está respondiendo	No
response_message	TextField	Mensaje de respuesta	No

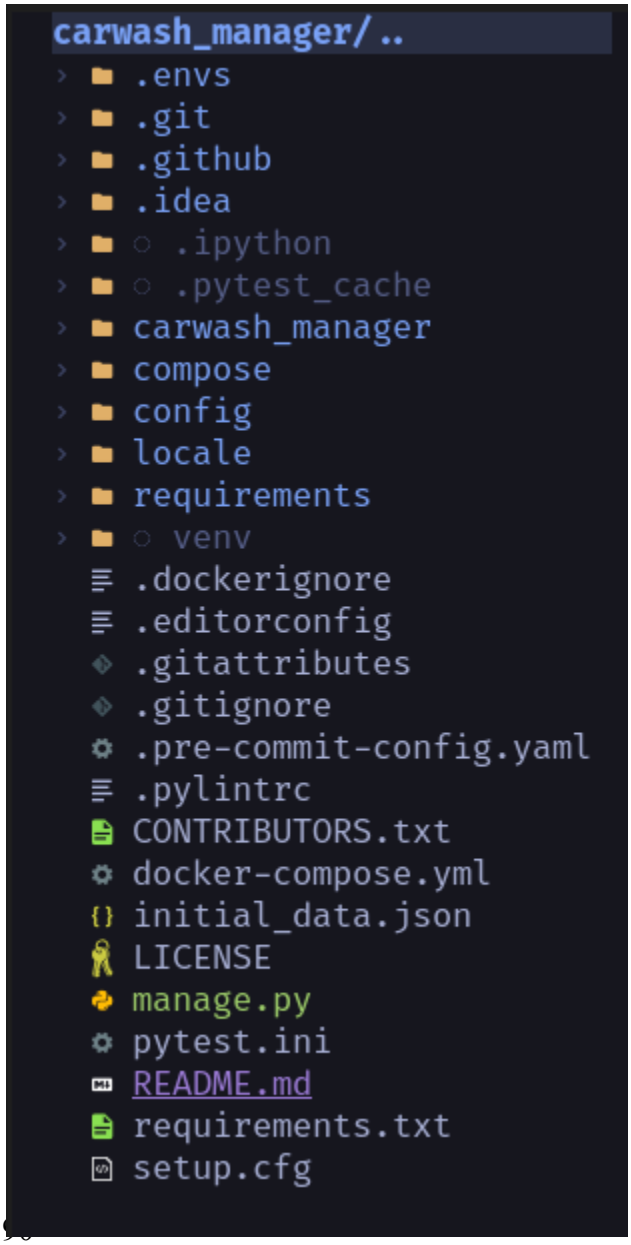
Anexo 6: Manual Técnico

Introducción

El objetivo del siguiente manual es ser documentación de referencia para los equipos de desarrollo que futuramente trabajen en el sistema web, para mayor facilidad de refactorización, adición de funcionalidades, ejecutarlo en servidores locales y aplicar pruebas de código.

Figura 18

Estructura de directorios



Carpetas

.envs: contiene las variables de entorno requeridas para ejecutar el sistema.

carwash_manager: tiene todos los archivos necesarios para ejecutar el sistema web en el backend y frontend.

compose: contiene los archivos para las imágenes personalizadas de Docker para Django y MySQL

config: archivos de configuraciones de Django

locale: archivos para fines de traducción multi-idioma

requirements: archivos para configuración de entornos virtuales de intérpretes de Python ya sea en software local o en contenedor virtual

Entorno de desarrollo

Es necesario contar con estas herramientas con sus respectivas dependencias:

- Sistema operativo Mac OS, Linux o Windows
- Docker
- Docker compose
- PyCharm, o Visual Studio Code
- Git
- Python 3.10

Figura 19

Obtener el código fuente de un repositorio en Github

```
.../Developing/python/carwash_manager
python  git clone git@github.com:darwing1210/carwash_manager.git
Cloning into 'carwash_manager'...
remote: Enumerating objects: 1745, done.
remote: Counting objects: 100% (446/446), done.
remote: Compressing objects: 100% (238/238), done.
remote: Total 1745 (delta 301), reused 310 (delta 198), pack-reused 1299
Receiving objects: 100% (1745/1745), 1.42 MiB | 485.00 KiB/s, done.
Resolving deltas: 100% (1120/1120), done.
python  cd carwash_manager/
carwash_manager [main] ls
carwash_manager  config           docker-compose.yml  LICENSE  manage.py   README.md     requirements.txt
compose          CONTRIBUTORS.txt initial_data.json   locale   pytest.ini  requirements  setup.cfg
carwash_manager [main]
```

Commandos: (suponiendo que se tiene conocimiento sobre el uso de terminal y de bash scripting)

- ***git clone*** *<url o ssh del repositorio>*: para descargar en la computadora en el directorio donde se desea el sistema web para correrlo en local.
- ***cd*** *<carpeta del repositorio>*: para moverse al directorio del repositorio
- ***ls***: para verificar la integridad del contenido

Figura 20

Abrir el proyecto en PyCharm

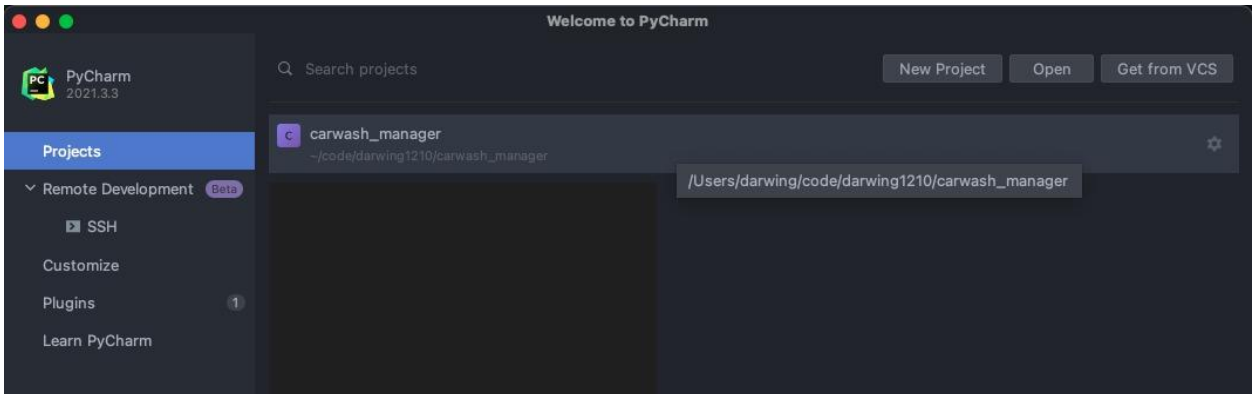


Figura 21

Arrancar el proyecto con Docker y Docker Compose

- Verificar si se tiene Docker instalado
- Verificar si el orquestador Docker Compose está instalado
- Revisar el estado actual de Docker si hay contenedores corriendo

Figura 22

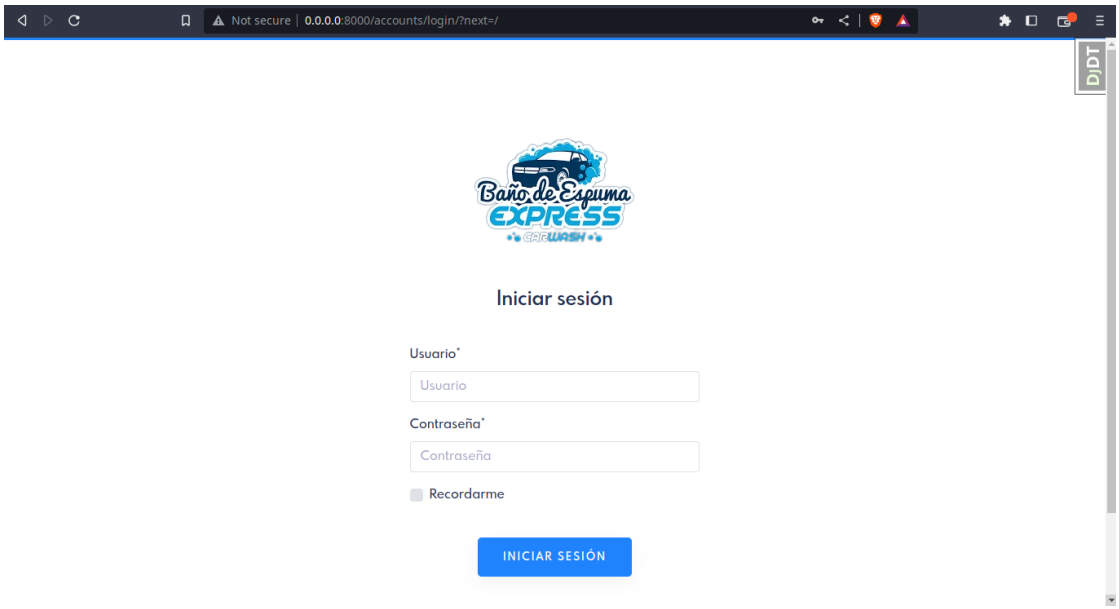
Etapa de arranque

```
sudo su
File  Edit  View  Search  Terminal  Help
[root@fedora]/home/ale/Clients/Darwing/carwash/carwash_manager# docker-compose up
Creating network "carwash_manager_default" with the default driver
Creating carwash_manager_local_mysql ... done
Creating carwash_manager_local_django ... done
Attaching to carwash_manager_local_mysql, carwash_manager_local_django
carwash_manager_local_mysql | 2023-03-25 00:08:06+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
carwash_manager_local_mysql | 2023-03-25 00:08:06+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
carwash_manager_local_mysql | 2023-03-25 00:08:06+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.32-1.el8 started.
carwash_manager_local_mysql | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
carwash_manager_local_django | Waiting for MySQL to become available...
carwash_manager_local_mysql | 2023-03-25T00:08:07.126046Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated
 and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
carwash_manager_local_mysql | 2023-03-25T00:08:07.127222Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.32) starting as
process 1
carwash_manager_local_mysql | 2023-03-25T00:08:07.134884Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
carwash_manager_local_mysql | 2023-03-25T00:08:07.273208Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
carwash_manager_local_mysql | 2023-03-25T00:08:07.485175Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
carwash_manager_local_mysql | 2023-03-25T00:08:07.485257Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS.
 Encrypted connections are now supported for this channel.
carwash_manager_local_mysql | 2023-03-25T00:08:07.487062Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Locat
ion '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
carwash_manager_local_mysql | 2023-03-25T00:08:07.517753Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address:
 '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
carwash_manager_local_mysql | 2023-03-25T00:08:07.517845Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Vers
ion: '8.0.32'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
carwash_manager_local_django | Operations to perform:
carwash_manager_local_django |   Apply all migrations: account, admin, auth, branches, contenttypes, credit, customers, employees, feed
back, payments, services, sessions, socialaccount, store, supplies, users, vehicles
carwash_manager_local_django | Running migrations:
carwash_manager_local_django |   Applying payments.0005_alter_payment_discount_amount_and_more... OK
carwash_manager_local_django | INFO:     Will watch for changes in these directories: ['/app']
carwash_manager_local_django | INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
carwash_manager_local_django | INFO:     Started reloader process [1] using WatchFiles
```

- Ejecutar ***docker-compose up*** para levantar los contenedores personalizados con las imágenes de Django y MySQL (recordar estar en el directorio del archivo docker-compose.yml).

Figura 23

Verificar ejecución



Ingresar a http://localhost:8000 en el navegador, debe aparecer la siguiente pantalla con el login.

Clases importantes del sistema

Figura 24

Modelo Branches

```
Darwing Medina
class Branch(SoftDeleteModel, TimeStampedModel):
    DEPARTMENTS = [...]

    name = models.CharField(
        verbose_name=_("Name"),
        max_length=100,
        help_text=_("Enter the name of the branch."),
    )
    address = models.CharField(
        verbose_name=_("Address"),
        max_length=200,
        help_text=_("Enter the address of the branch."),
        blank=True,
    )
    department = models.IntegerField(
        verbose_name=_("Department"),
        choices=DEPARTMENTS,
        help_text=_("Select the department where the branch is located."),
        blank=True,
        null=True,
    )
```

Nota: Ruta: /carwash_manager/branches/models.py

Esta definición de clase cuenta con todo lo necesario para los datos de cada sucursal del lavacar.

Figura 25

Modelo Customers

```
Darwing Medina
class Customer(SoftDeleteModel, TimeStampedModel):
    name = models.CharField(
        max_length=100,
        help_text=_("Enter the name of the customer."),
        verbose_name=_("Name"),
    )
    customer_type = models.CharField(
        max_length=20,
        choices=CustomerTypes.choices,
        default=CustomerTypes.PERSON,
        verbose_name=_("Customer type"),
    )

    phone_number = models.CharField(
        max_length=20,
        null=True,
        blank=True,
        help_text=_("Enter the phone number of the customer."),
        verbose_name=_("Phone number"),
    )
```

Nota: Ruta: /carwash_manager/customers/models.py

Estas definiciones tienen todo lo necesario para administrar los clientes clasificándolos por tipos y guardando la información necesaria de cada cliente.

Figura 26

Modelo Employees

```
Darwing Medina
class Employee(SoftDeleteModel, TimeStampedModel):
    user = models.OneToOneField(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name="employee",
        blank=True,
        verbose_name=_("User"),
    )
    name = models.CharField(_("Name"), max_length=100)
    email = models.EmailField(blank=True, verbose_name=_("Email"))
    phone = models.CharField(_("Phone"), max_length=15, blank=True)
    position = models.CharField(
        _("Position"),
        max_length=50,
        choices=Positions.choices,
    )
```

Nota: Ruta: /carwash_manager/employees/models.py

En estas definiciones administramos la información de los empleados y sus posiciones o cargos en el lavacar en las diferentes sucursales en las que trabajan.

Figura 27

Modelo Services

```
Darwing Medina
class WashStockItem(models.Model):
    wash = models.ForeignKey(Wash, on_delete=models.CASCADE, verbose_name=_("Wash"))
    stock_item = models.ForeignKey(
        StockItem, on_delete=models.CASCADE, verbose_name=_("Stock item")
    )
    quantity_per_wash = models.FloatField(
        blank=True, verbose_name=_("Quantity per wash")
    )
```

```
Darwing Medina
class ServiceSupply(models.Model):
    service = models.ForeignKey(
        Service, on_delete=models.CASCADE, verbose_name=_("Service")
    )
    supply = models.ForeignKey(
        Supply, on_delete=models.CASCADE, verbose_name=_("Supply")
    )
    quantity_per_service = models.FloatField(
        blank=True, verbose_name=_("Quantity per service")
    )

    Darwing Medina
    def save(self, *args, **kwargs):
        if not self.quantity_per_service:
            self.quantity_per_service = self.supply.quantity_per_unit
        super().save(*args, **kwargs)

    Darwing Medina
    def __str__(self) -> str:
        return f"{self.service.name} - {self.supply.name}"
```

```
Darwing Medina
class Wash(SoftDeleteModel, TimeStampedModel):
    branch = models.ForeignKey(
        Branch,
        on_delete=models.CASCADE,
        help_text=_("Select the branch where the wash was performed."),
        verbose_name=_("Branch"),
    )
    customer = models.ForeignKey(
        Customer,
        on_delete=models.CASCADE,
        help_text=_("Select the customer for whom the wash was performed."),
        verbose_name=_("Customer"),
    )
    employee = models.ForeignKey(
        Employee,
        on_delete=models.CASCADE,
        help_text=_("Select the employee who performed the wash."),
        verbose_name=_("Employee"),
    )
```

```
Darwing Medina
class WashStatusChoices(models.TextChoices):
    PENDING = "pending", _("Pending")
    IN_PROGRESS = "in_progress", _("In Progress")
    COMPLETED = "completed", _("Completed")
    CANCELED = "canceled", _("Canceled")
```

Estos modelos se encargan de administrar los servicios que se pueden aplicar en un lavado, tomando en cuenta el dónde, cuándo, quién lo hizo, a quién se atendió, qué vehículo se lavó, y el costo del servicio.

Figura 28

Modelo Payments

```
Darwing Medina
class Promotion(SoftDeleteModel, TimeStampedModel):
    name = models.CharField(
        max_length=100,
        help_text=_("Enter the name of the promotion."),
        verbose_name=_("Name"),
    )
    branches = models.ManyToManyField(
        Branch,
        help_text=_("Select the branches where this promotion applies."),
        verbose_name=_("Branches"),
    )
    description = models.TextField(
        help_text=_("Enter a description of the promotion."),
        verbose_name=_("Description"),
    )
    code = models.CharField(
        max_length=20,
        help_text=_("Enter the promotion code."),
        blank=True,
        null=True,
        verbose_name=_("Code"),
    )
```

Nota: Estas definiciones administran los pagos de los servicios.

Ruta: /carwash_manager/services/models.py

Mixins

Figura 29

Update Users y Update Branches

```
Darwing Medina
class UpdateUserMixin:
    Darwing Medina
    def form_valid(self, form):
        if self.object is None or self.object.pk is None:
            # if the object is new, set the created_by field
            form.instance.created_by = self.request.user
        # always set the updated_by field
        form.instance.updated_by = self.request.user
        return super().form_valid(form)


Darwing Medina
class UpdateBranchMixin:
    Darwing Medina
    def form_valid(self, form):
        form.instance.branch = self.request.active_branch
        return super().form_valid(form)
```

Estos mixins son para agilizar y optimizar seguridad en los formularios para actualizar el cambio de branch (sucursal) y para insertar automáticamente el branch de turno en los forms de los modelos donde el branch es requerido por relación foreign key.

Views

Figura 30

Vista para listado de lavados

```
Darwing Medina
class WashListView(PermissionRequiredMixin, BranchFilteredListView):
    model = Wash
    template_name = "services/wash_list.html"
    context_object_name = "washes"
    permission_required = "services.view_wash"

    Darwing Medina
    def get_queryset(self):
        qs = super().get_queryset()
        employee = self.request.user._employee
        if employee and employee.position == Positions.WASHER:
            qs = qs.filter(employee=employee)
        return qs
```

Figura 31

Vista para crear lavado

```
Darwing Medina
class WashCreateView(
    PermissionRequiredMixin, UpdateUserMixin, UpdateBranchMixin, CreateView
):
    model = Wash
    form_class = WashForm
    template_name = "services/wash_create.html"
    permission_required = "services.add_wash"

    Darwing Medina
    def get_context_data(self, **kwargs: Any) -> dict[str, Any]:
        context = super().get_context_data(**kwargs)
        context["vehicles"] = VehicleForm()
        context["employees"] = EmployeeForm()
        context["customers"] = CustomerForm()
        context["services"] = ServiceForm()
        return context

    Darwing Medina
    def get_form_kwargs(self):...
```

Figura 32

Vista para actualizar o editar un lavado

```
Darwing Medina
class WashUpdateView(
    UserPassesTestMixin, PermissionRequiredMixin, UpdateUserMixin, UpdateView
):
    model = Wash
    form_class = WashForm
    template_name = "services/wash_create.html"
    success_url = reverse_lazy("services:wash_list")
    permission_required = "services.change_wash"

    Darwing Medina
    def test_func(self):...

    Darwing Medina
    def get_form_kwargs(self):...

    Darwing Medina
    def get_context_data(self, **kwargs: Any) -> dict[str, Any]:...

    Darwing Medina
    def form_valid(self, form):...
```

Figura 33

Vista para ejecutar el pago de un lavado

```
Darwing Medina
class WashPayView(PermissionRequiredMixin, View):
    permission_required = ("payments.change_payment", "payments.add_payment")

    Darwing Medina
    def get(self, request, *args, **kwargs):
        obj_pk = self.kwargs.get("pk")
        obj = get_object_or_404(Wash, pk=obj_pk)

        payment = obj.generate_payment()
        return redirect("payments:payments_update", pk=payment.pk)
```

Figura 34

Vista para eliminar un lavado

```
Darwing Medina
class WashDeleteView(PermissionRequiredMixin, UpdateUserMixin, DeleteView):
    model = Wash
    template_name = "services/wash_delete.html"
    success_url = reverse_lazy("wash_list")
    permission_required = "services.delete_wash"
```

Figura 35

Vista para la lista de promociones

```
Darwing Medina
class PromotionListView(PermissionRequiredMixin, ActiveBranchMixin, ListView):
    model = Promotion
    template_name = "payments/promotions_list.html"
    context_object_name = "promotions"
    permission_required = "payments.view_promotion"

    Darwing Medina
    def get_queryset(self):
        branch = self.get_active_branch()
        qs = super().get_queryset()
        return qs.filter(branches__in=[branch])
```

Figura 36

Vista para detalles de cada promoción listada

```
Darwing Medina
class PromotionDetailView(PermissionRequiredMixin, DetailView):
    model = Promotion
    template_name = "payments/promotions_detail.html"
    context_object_name = "promotion"
    permission_required = "payments.view_promotion"
```

Figura 37

Vista para crear una nueva promoción

```
Darwing Medina
class PromotionCreateView(
    PermissionRequiredMixin, UpdateBranchMixin, UpdateUserMixin, CreateView
):
    model = Promotion
    template_name = "payments/promotions_create.html"
    form_class = PromotionForm
    success_url = reverse_lazy("payments:promotions_list")
    permission_required = "payments.add_promotion"
```

Figura 38

Vista para actualizar una promoción

```
Darwing Medina
class PromotionUpdateView(PermissionRequiredMixin, UpdateUserMixin, UpdateView):
    model = Promotion
    template_name = "payments/promotions_create.html"
    form_class = PromotionForm
    success_url = reverse_lazy("payments:promotions_list")
    permission_required = "payments.change_promotion"
```

Figura 39

Vista para eliminar una promoción

```
Darwing Medina
class PromotionDeleteView(PermissionRequiredMixin, UpdateUserMixin, DeleteView):
    model = Promotion
    template_name = "payments/promotions_delete.html"
    success_url = reverse_lazy("payments:promotions_list")
    permission_required = "payments.delete_promotion"
```

Figura 40

Vista para listar los pagos

```
Darwing Medina
class PaymentListView(PermissionRequiredMixin, BranchFilteredListView):
    model = Payment
    template_name = "payments/payments_list.html"
    context_object_name = "payments"
    permission_required = "payments.view_payment"
```

Figura 41

Vista para crear pagos

```
Darwing Medina
class PaymentCreateView(
    PermissionRequiredMixin, UpdateBranchMixin, UpdateUserMixin, CreateView
):
    model = Payment
    template_name = "payments/payments_create.html"
    form_class = PaymentForm
    permission_required = "payments.add_payment"

    Darwing Medina
    def form_valid(self, form):
        response = super().form_valid(form)
        self.object.calculate_discount_amount()
        self.object.calculate_taxes_amount()
        self.object.calculate_total_amount()
        return response
```

Figura 42

Vista para actualizar pagos

```
Darwing Medina
class PaymentUpdateView(PermissionRequiredMixin, UpdateUserMixin, UpdateView):
    model = Payment
    template_name = "payments/payments_create.html"
    form_class = PaymentForm
    permission_required = "payments.change_payment"

    Darwing Medina
    def form_valid(self, form):
        response = super().form_valid(form)
        self.object.calculate_discount_amount()
        self.object.calculate_taxes_amount()
        self.object.calculate_total_amount()
        return response
```

Figura 43

Vista para ver detalle por pago

```
Darwing Medina
class PaymentDetailView(PermissionRequiredMixin, DetailView):
    model = Payment
    template_name = "payments/payments_detail.html"
    context_object_name = "payment"
    permission_required = "payments.view_payment"
```

Figura 44

Vista para eliminar un pago

```
Darwing Medina
class PaymentDeleteView(PermissionRequiredMixin, UpdateUserMixin, DeleteView):
    model = Payment
    template_name = "payments/payments_delete.html"
    success_url = reverse_lazy("payments:list")
    permission_required = "payments.delete_payment"
```

Ruta de archivos estáticos (CSS, JavaScript, multimedia)

Figura 45

Ruta de archivos estáticos para la UI

Figura 46

JavaScript

Figura 47

CSS

Figura 48

Fuentes

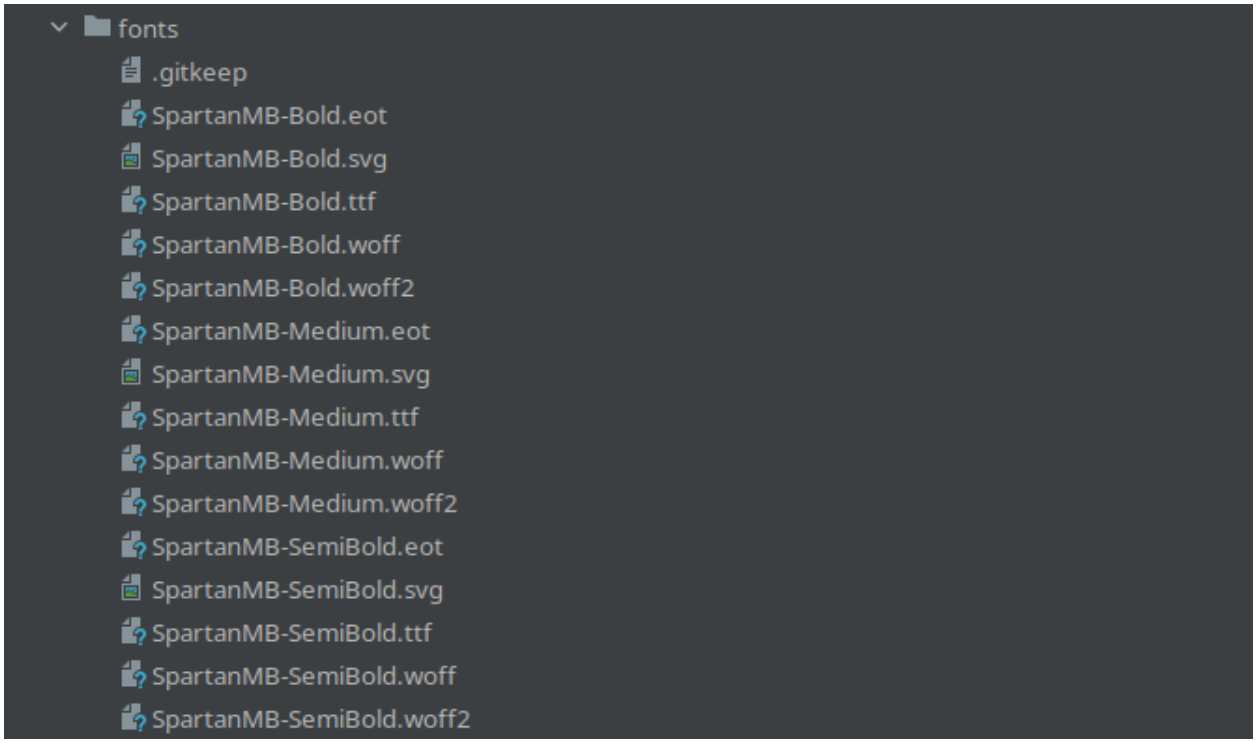


Figura 49

Imágenes

Anexo 7: Manual de Usuario

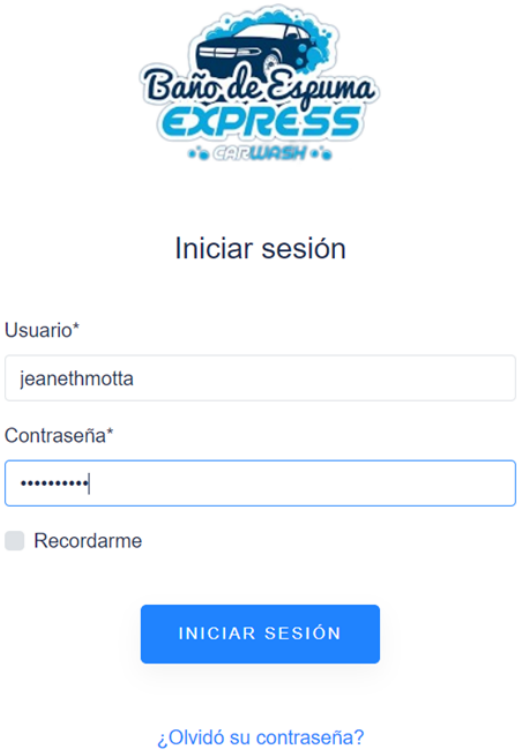
A través de este documento, se busca brindar asistencia a los usuarios que harán uso del sistema web de gestión de inventario y registro de actividades en autolavado “Baño de espuma”.

Acceso al sistema

El ingreso al Sistema web de gestión de inventario y registro de actividades en autolavado “Baño de Espuma” se realizará desde la siguiente dirección : **https://app.banodeespuma.com/** en la cual, una vez se auntentifique como usuario autorizado (Propietario,Gerente,Cajero,Lavador) se, podrá ingresar al sistema.

Figura 50

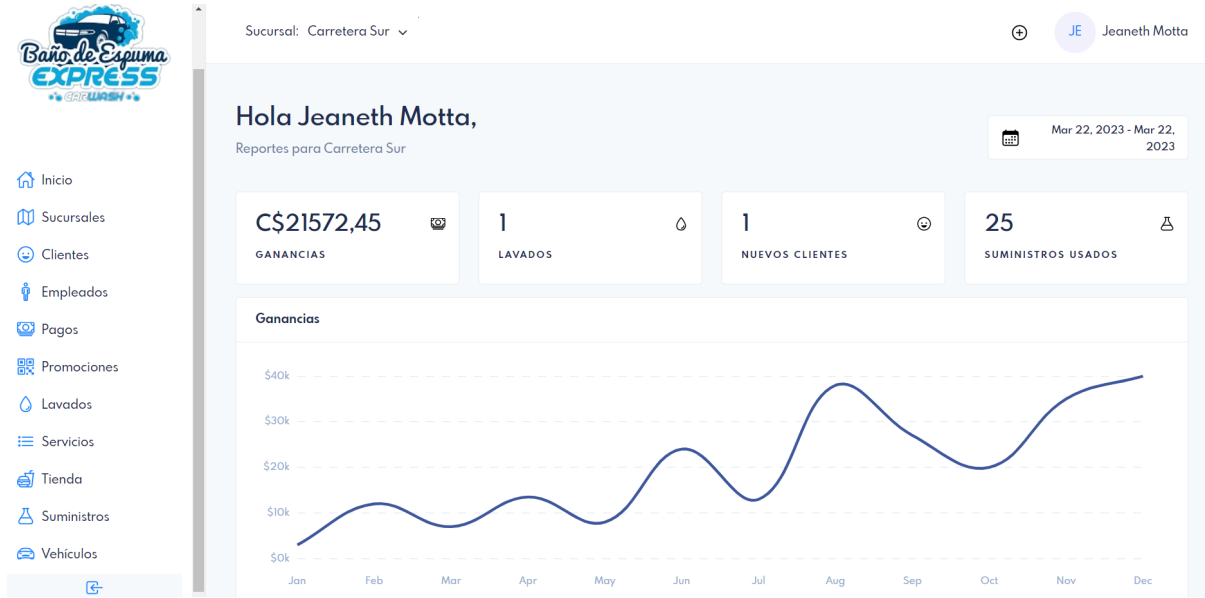
Pantalla de inicio de sesión



Posterior a la autenticación exitosa, el sistema automáticamente redirige al usuario a la pantalla inicial.

Figura 51

Pantalla de inicio del sistema



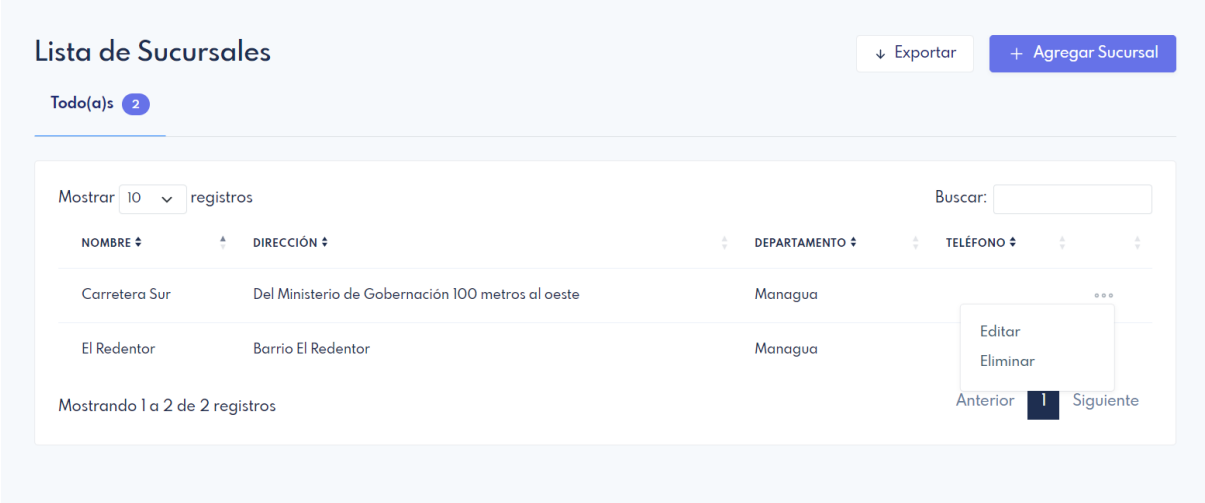
Desde la pantalla inicial se puede acceder a las diferentes opciones que tiene el sistema tales como: sucursales, promociones, pagos. Estas se mostrarán en dependencia del rol asignado al usuario que ingrese al sistema.

Gestionar Sucursales

En este formulario se llevará a cabo el registro de las sucursales con los que cuenta la empresa.

Figura 52

Listado de sucursales



En el formulario principal de las sucursales podrá encontrar las siguientes opciones:

***Botón Agregar Sucursal*:** Se desplegará un formulario para registrar una nueva sucursal.

***Editar**:* Se despliega un formulario en donde permite actualizar la información de una sucursal.

Figura 53

Pantalla para agregar una sucursal

Figura 54

Pantalla para actualizar una sucursal

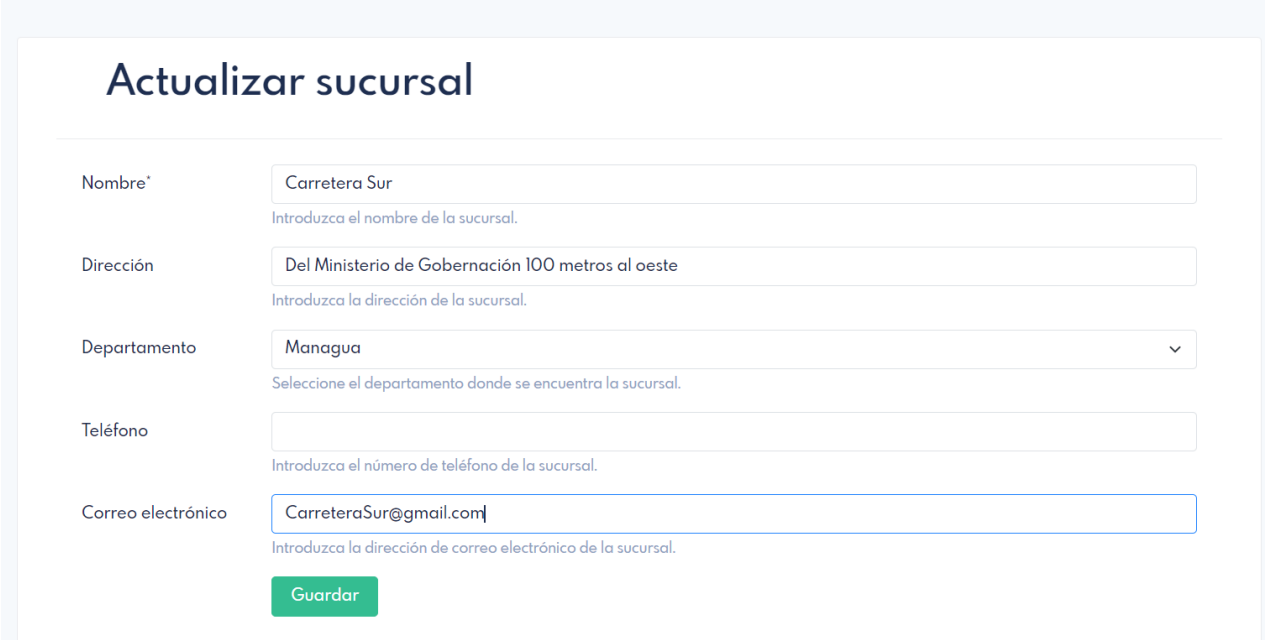
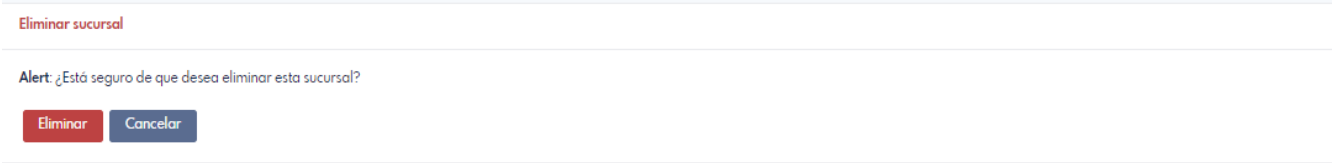


Figura 55

Eliminar Sucursal : mensaje de confirmación

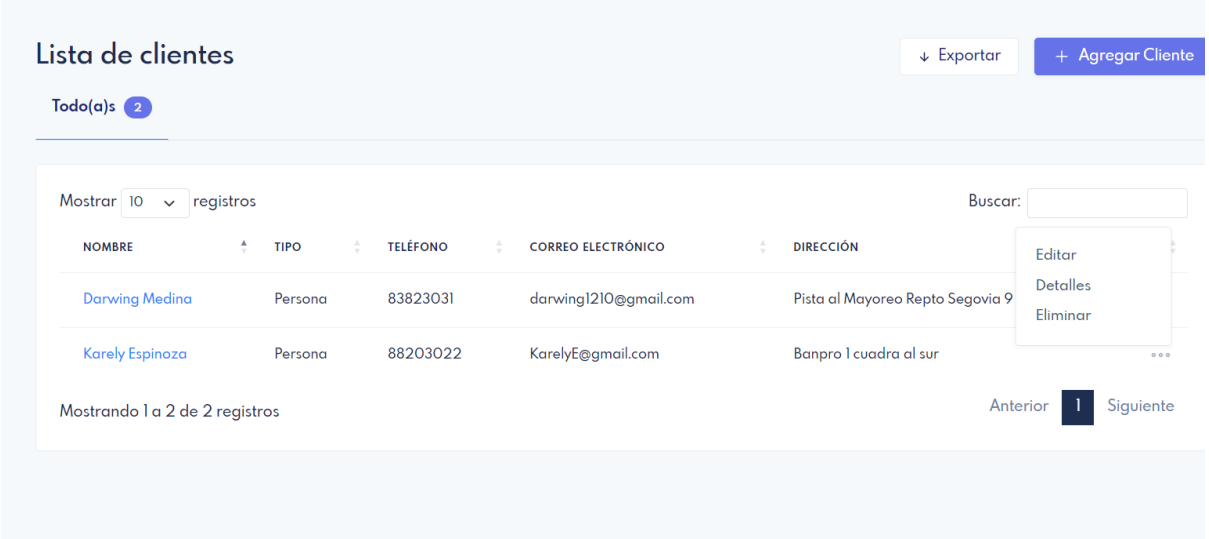


Gestionar Clientes

En este formulario se muestra información sobre los clientes.

Figura 56

Listado de Clientes

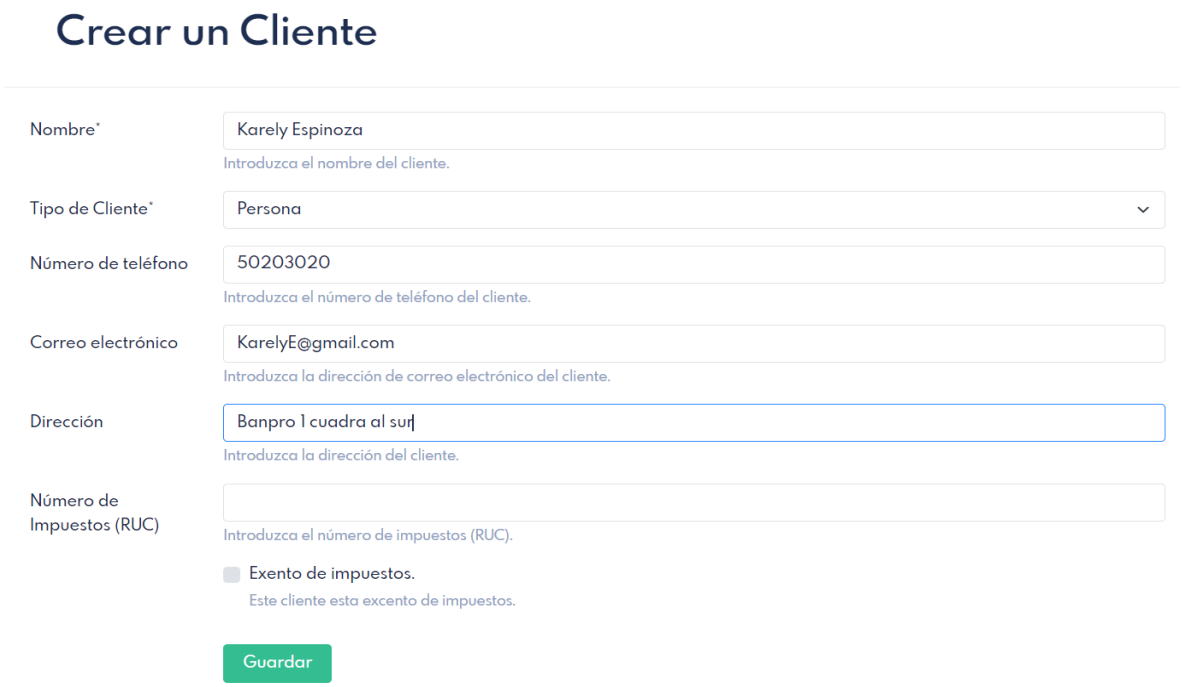


En el formulario principal de los clientes podrá encontrar las siguientes opciones:

***Botón Agregar Cliente*:** Se desplegará el siguiente formulario que permite registrar la información de los nuevos clientes.

Figura 57

Pantalla para agregar un cliente



***Opción Editar**:* Se despliega un formulario en donde permite actualizar la información de un cliente.

Figura 58

Pantalla para actualizar los datos del cliente

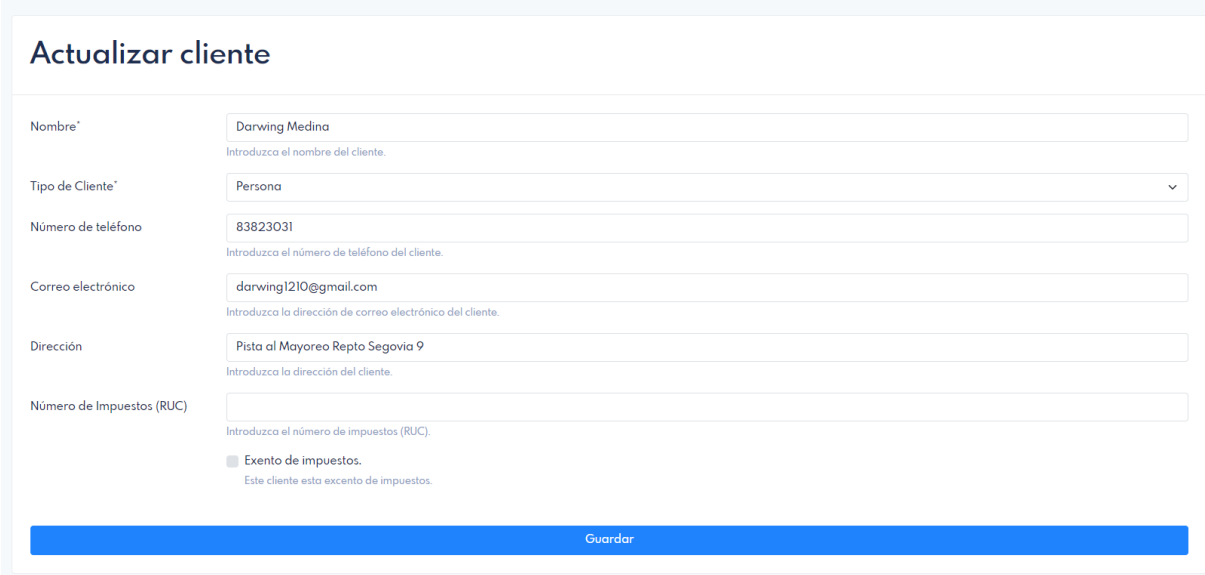


Figura 59

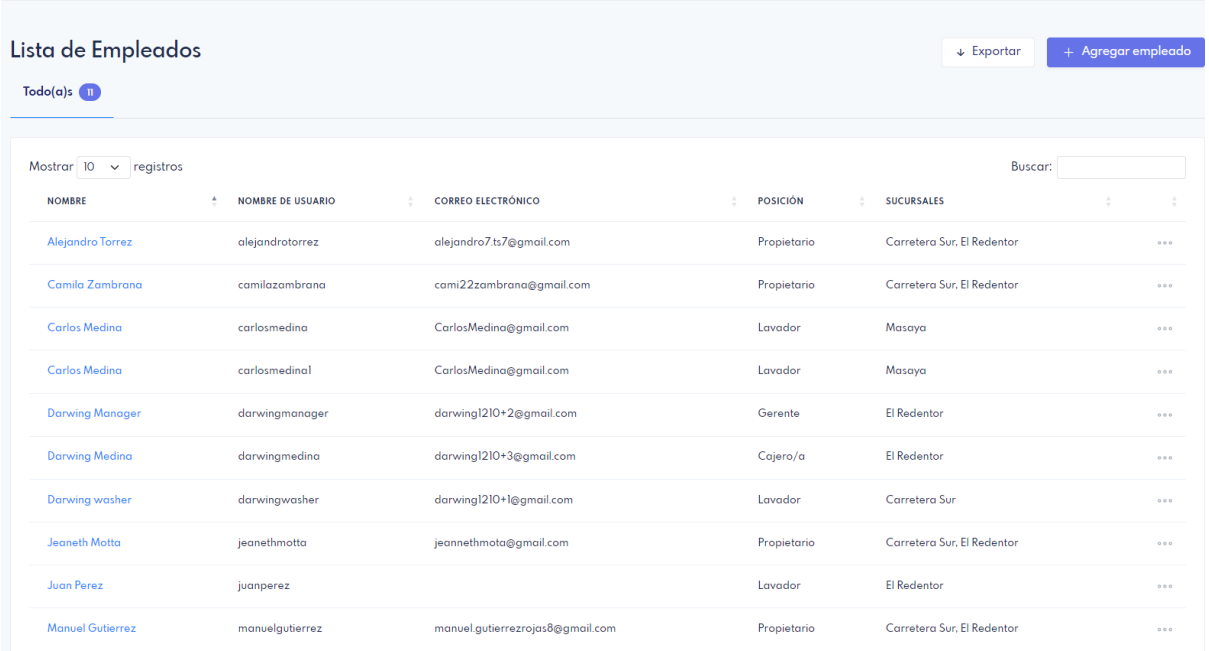
Eliminar Cliente: mensaje de confirmación

Gestión de Empleados

En este formulario se muestra información sobre los empleados del autolavado.

Figura 60

Lista de empleados



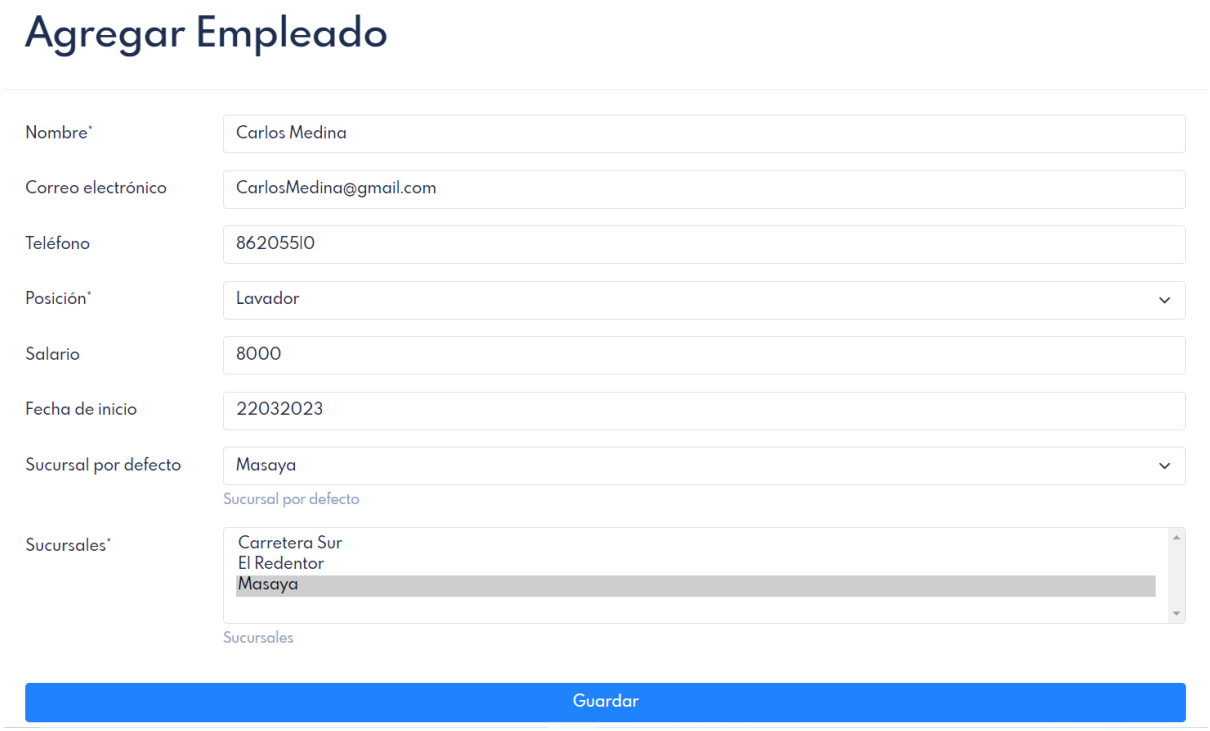
NOMBRE	NOMBRE DE USUARIO	CORREO ELECTRÓNICO	POSICIÓN	SUCURSALES
Alejandro Torrez	alejandrotorrez	alejandro7.ts7@gmail.com	Propietario	Carretera Sur, El Redentor
Camila Zambrana	camilazambrana	cami22zambrana@gmail.com	Propietario	Carretera Sur, El Redentor
Carlos Medina	carlosmedina	CarlosMedina@gmail.com	Lavador	Masaya
Carlos Medina	carlosmedina1	CarlosMedina@gmail.com	Lavador	Masaya
Darwing Manager	darwingmanager	darwing1210+2@gmail.com	Gerente	El Redentor
Darwing Medina	darwingmedina	darwing1210+3@gmail.com	Cajero/a	El Redentor
Darwing washer	darwingwasher	darwing1210+1@gmail.com	Lavador	Carretera Sur
Jeaneth Motta	jeanethmotta	jeannethmota@gmail.com	Propietario	Carretera Sur, El Redentor
Juan Perez	juanperez		Lavador	El Redentor
Manuel Gutierrez	manuelgutierrez	manuel.gutierrezrojas8@gmail.com	Propietario	Carretera Sur, El Redentor

En el formulario de los empleados podrá encontrar las siguientes opciones:

***Botón Agregar Empleado*:** Se desplegará el siguiente formulario que permite registrar la información de los nuevos empleados.

Figura 61

Pantalla para agregar un empleado



Opción Editar: Se despliega un formulario en donde permite actualizar la información de un empleado.

Figura 62

Pantalla para Editar los datos de un empleado

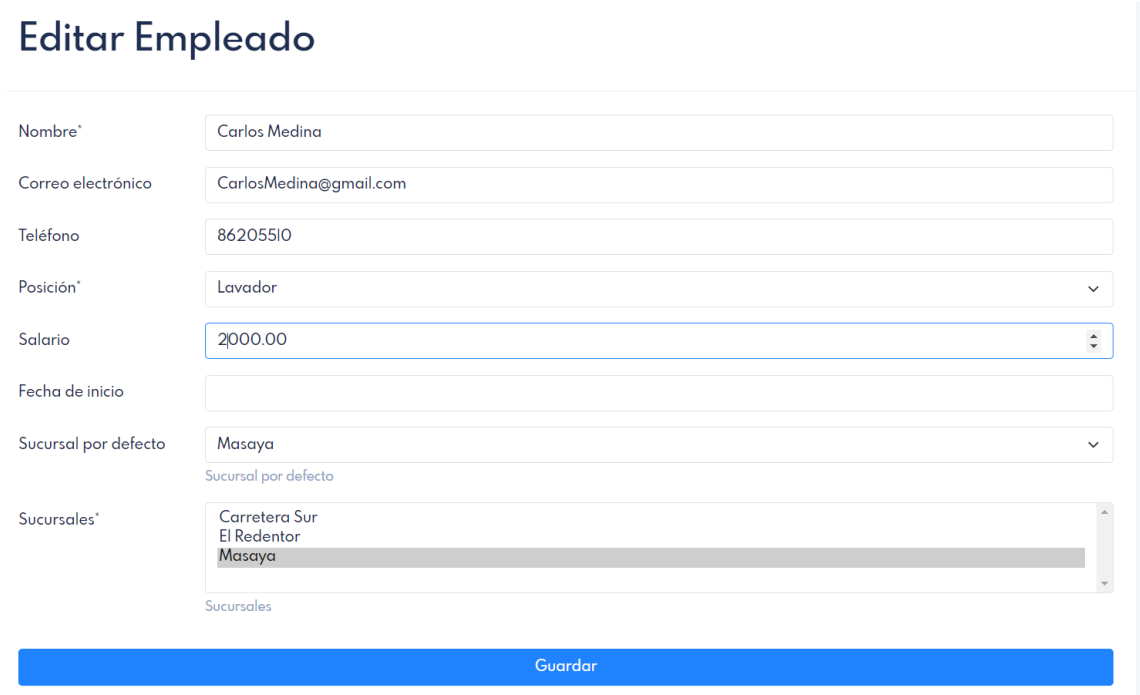
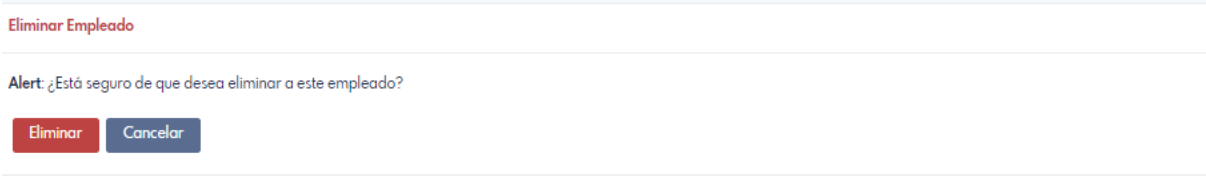


Figura 63

Eliminar Empleado: mensaje de confirmación

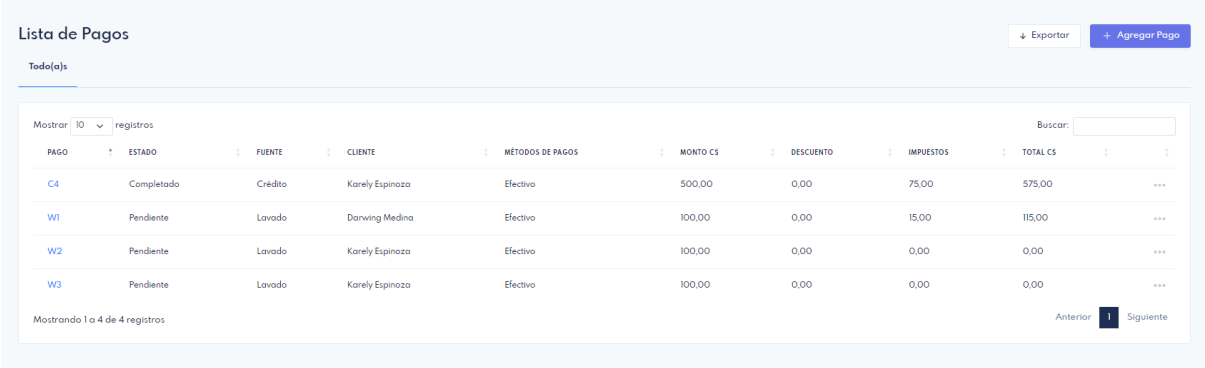


Gestión de Pagos

En este formulario se muestra información sobre los pagos que se han realizado en el autolavado.

Figura 64

Pantalla que muestra la lista de los pagos

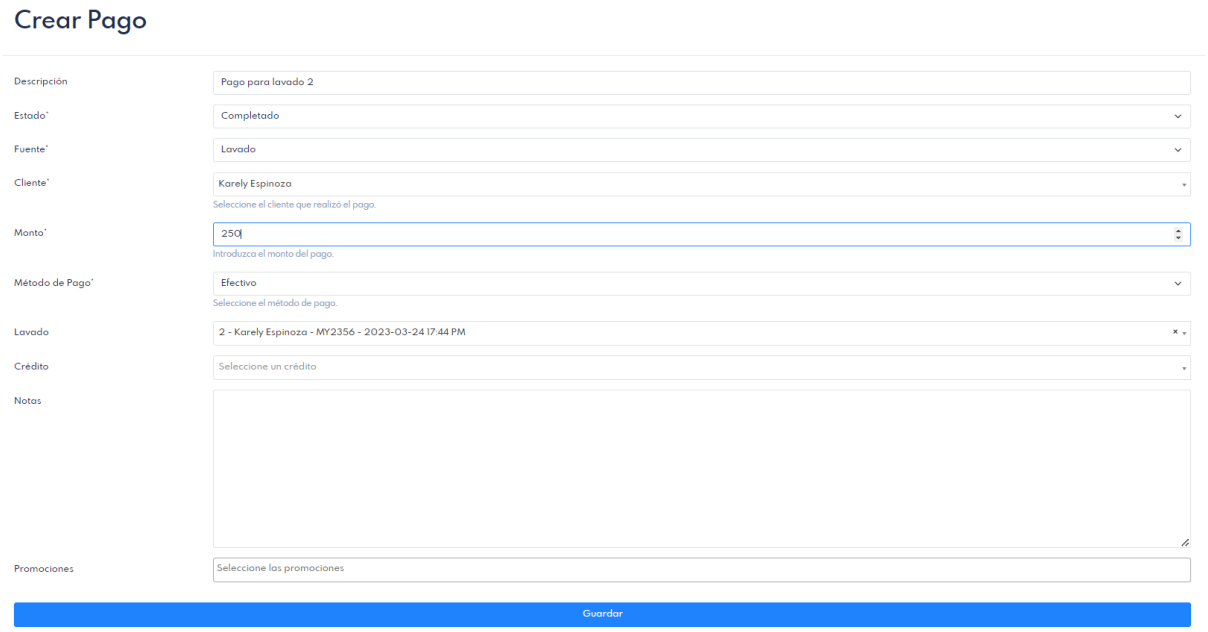


En el formulario de pagos podrá encontrar las siguientes opciones:

***Botón Agregar Pago*:** Se desplegará el siguiente formulario que permite registrar la información de los pagos.

Figura 65

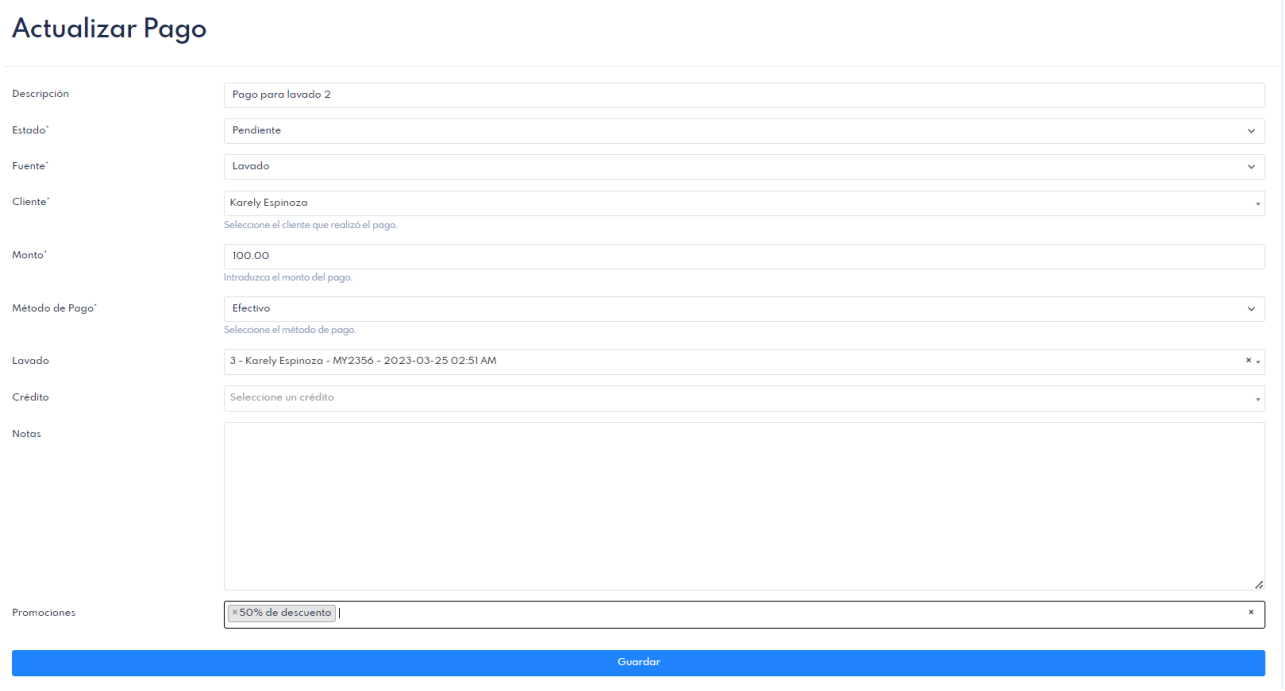
Pantalla para agregar un pago



***Opción Editar**:* Se despliega un formulario en donde permite actualizar la información de un pago.

Figura 66

Pantalla para actualizar un pago



Opción Detalles *:* muestra información general de los pagos, en este formulario muestra las opciones:

Editar: Actualizar la información de un pago.

Eliminar: esta opción da de baja a un pago.

Figura 67

Pantalla para muestra el detalle de un pago

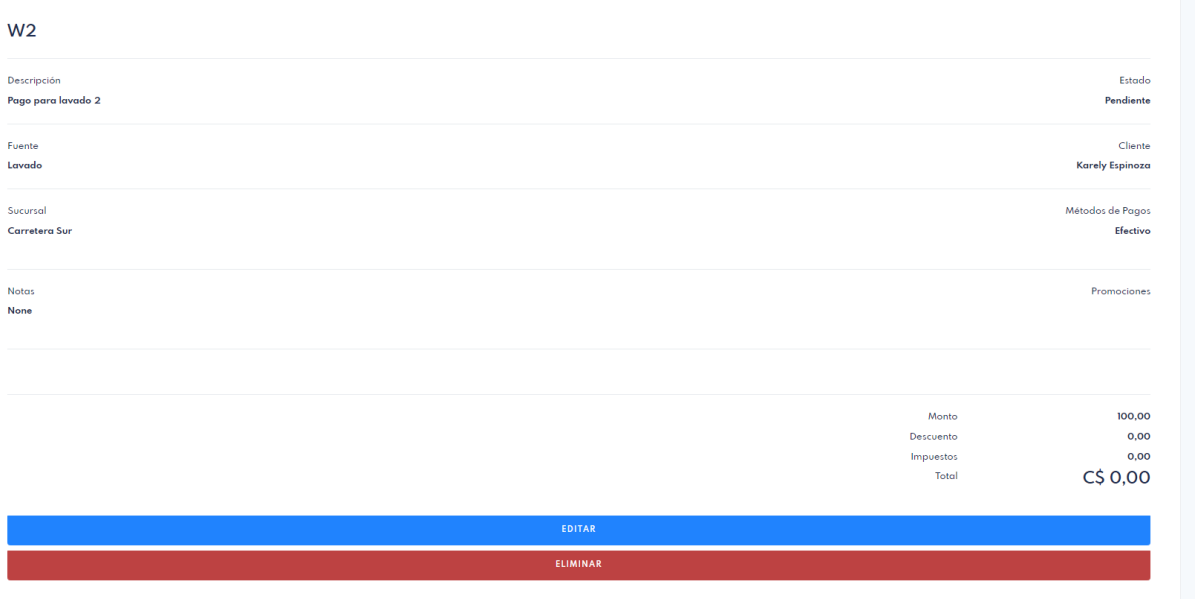
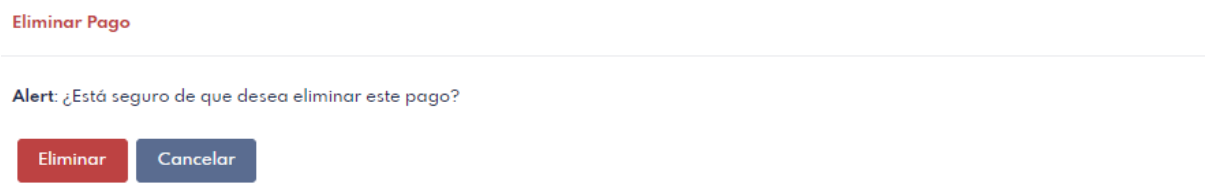


Figura 68

Eliminar Pago: mensaje de confirmación

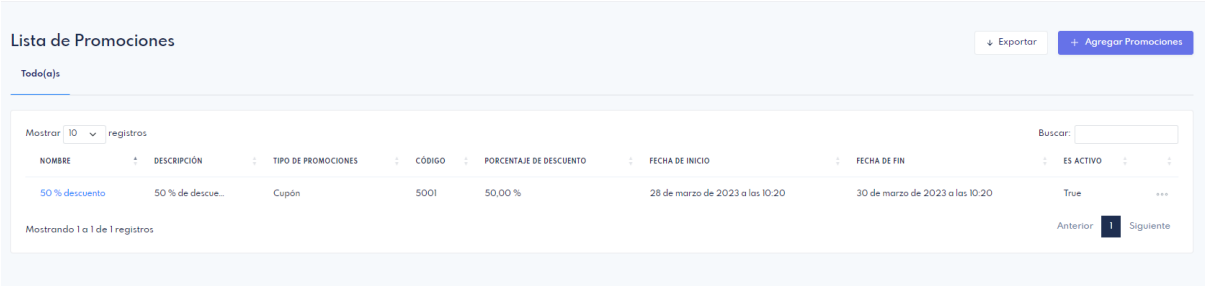


Gestión de Promociones

En este formulario se muestra información sobre las distintas promociones que se aplican en el autolavado.

Figura 69

Pantalla de listado de promociones

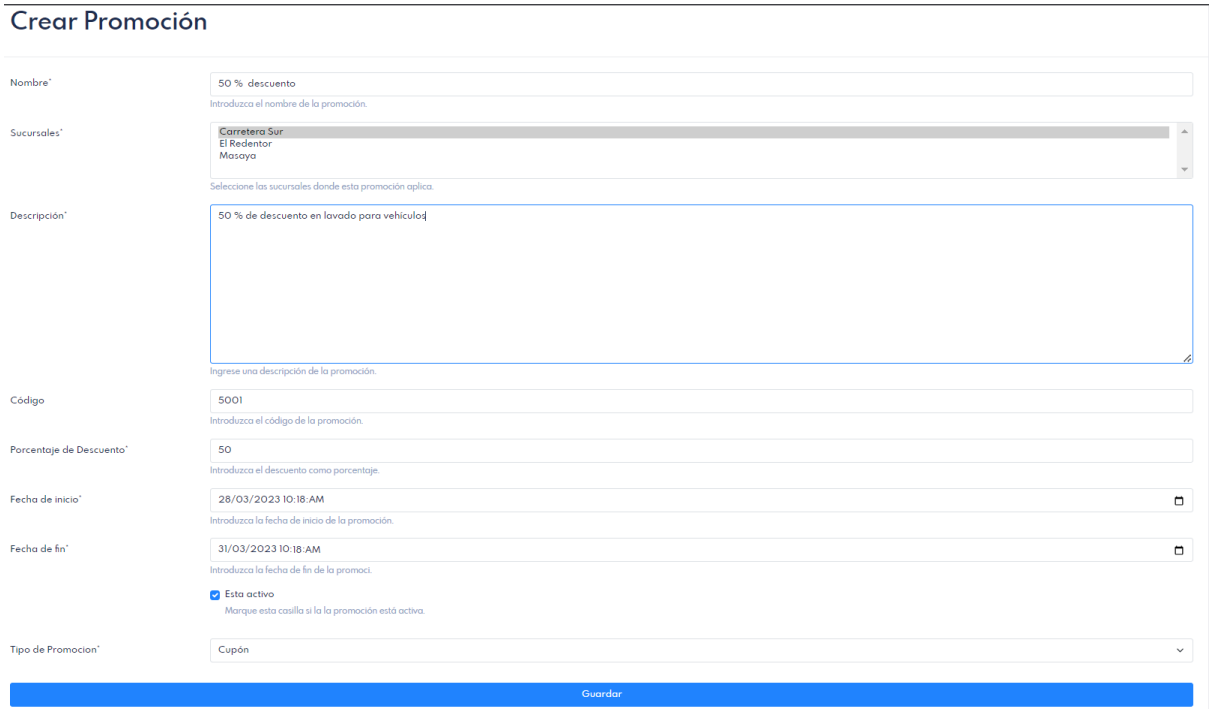


En el formulario de Promociones podrá encontrar las siguientes opciones:

***Botón Agregar Promociones*:** Se desplegará el siguiente formulario que permite registrar la información de las promociones que se llevarán a cabo en el autolavado.

Figura 70

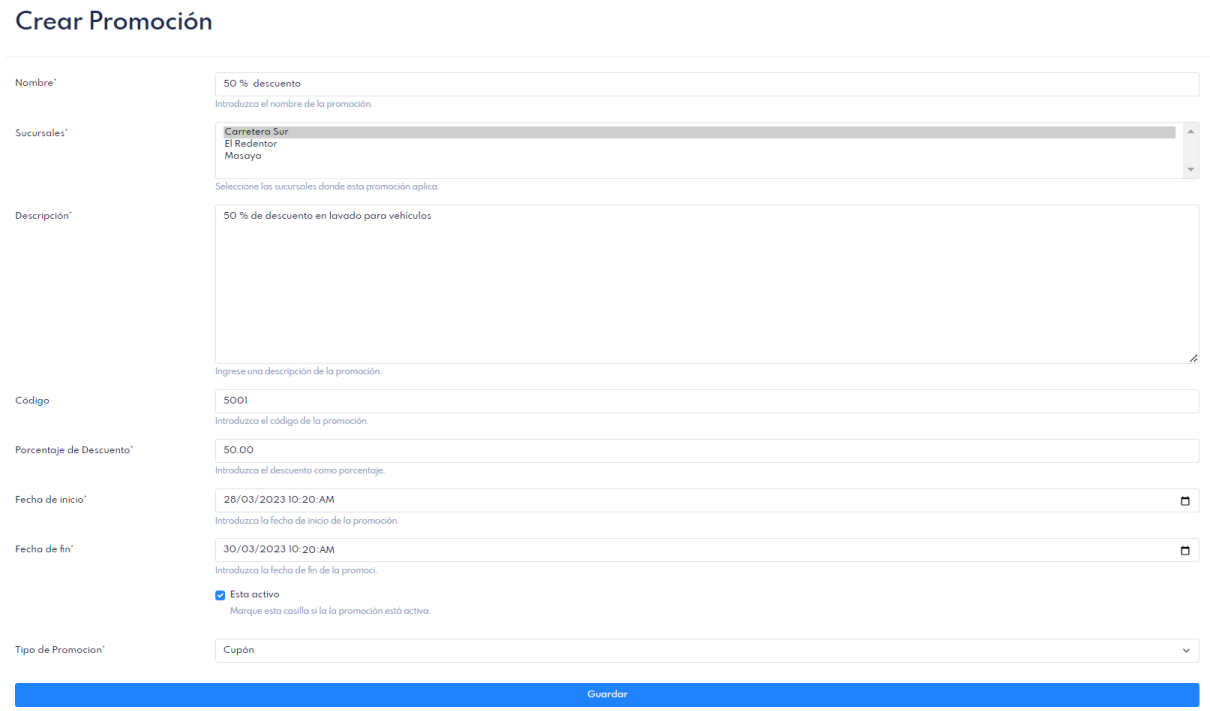
Pantalla para agregar una promoción



Opción Editar: Se despliega un formulario en donde permite actualizar la información de las promociones.

Figura 71

Pantalla para editar una promoción



Opción Detalles *:* muestra información general de las promociones, en este formulario muestra las opciones:

Editar: Actualizar la información de una promoción.

Eliminar: esta opción da de baja a una promoción.

Figura 72

Pantalla que muestra la información general de una promoción

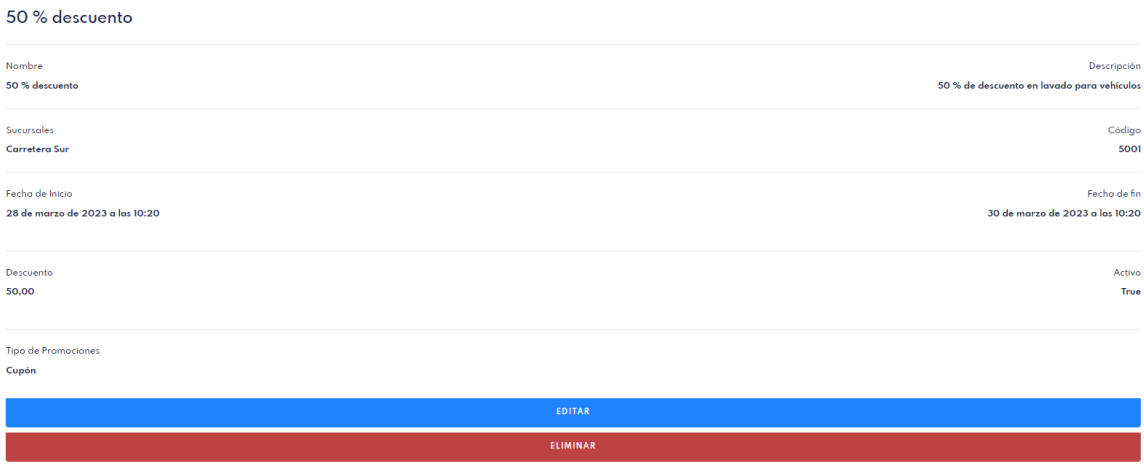
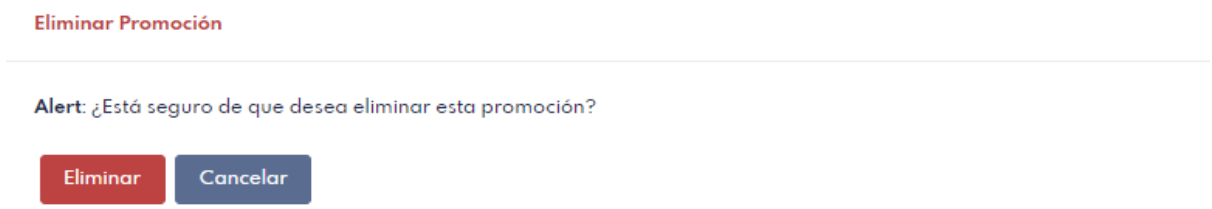


Figura 73

Eliminar Promoción: mensaje de confirmación

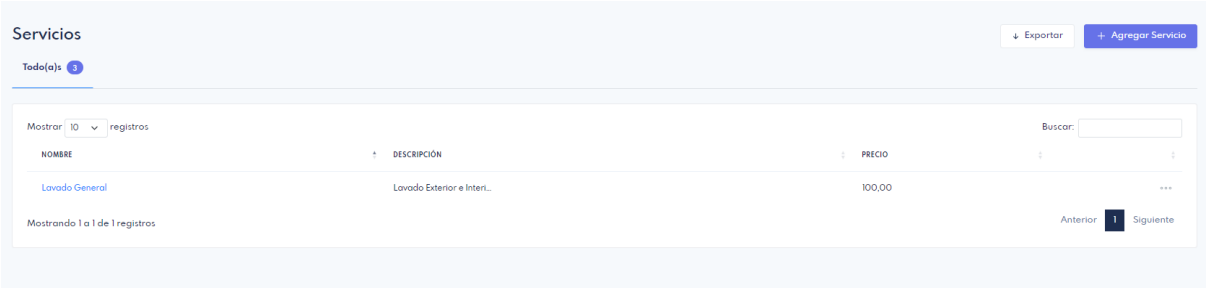


Gestión Servicios

En este formulario se muestra información sobre los distintos servicios que se realizan en un autolavado.

Figura 74

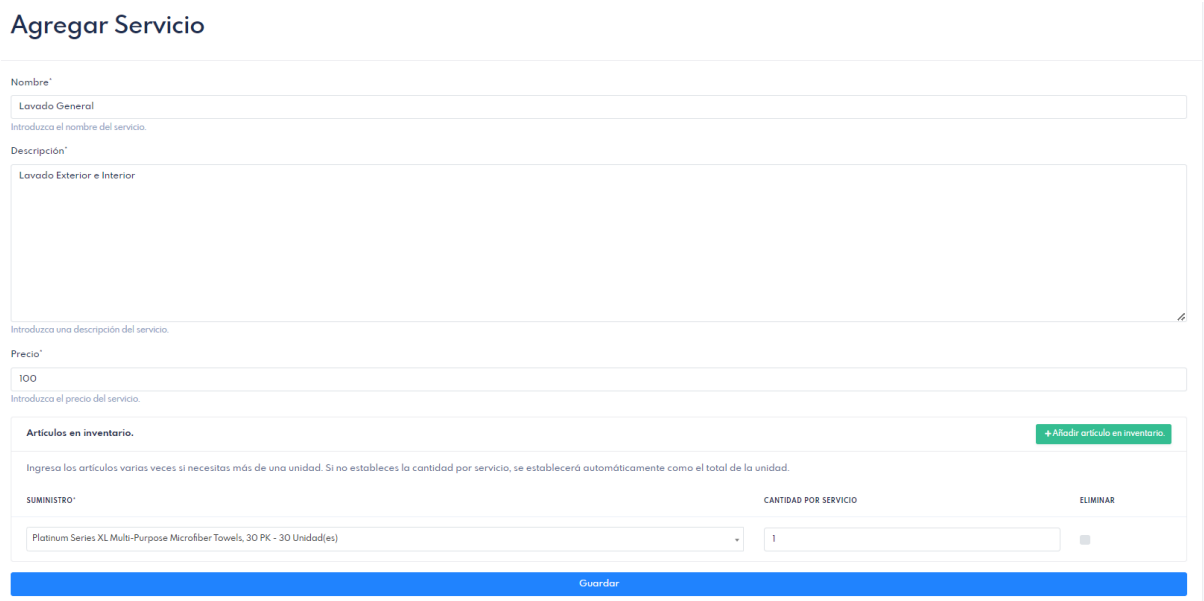
Pantalla de listado de servicios



***Botón Agregar Servicio*:** Se desplegará el siguiente formulario que permite registrar la información del servicio que se llevará a cabo en el autolavado.

Figura 75

Pantalla para agregar un servicio



***Opción Editar**:* Se despliega un formulario en donde permite actualizar la información de los servicios que se realizan en el autolavado.

Figura 76

Pantalla para editar un servicio

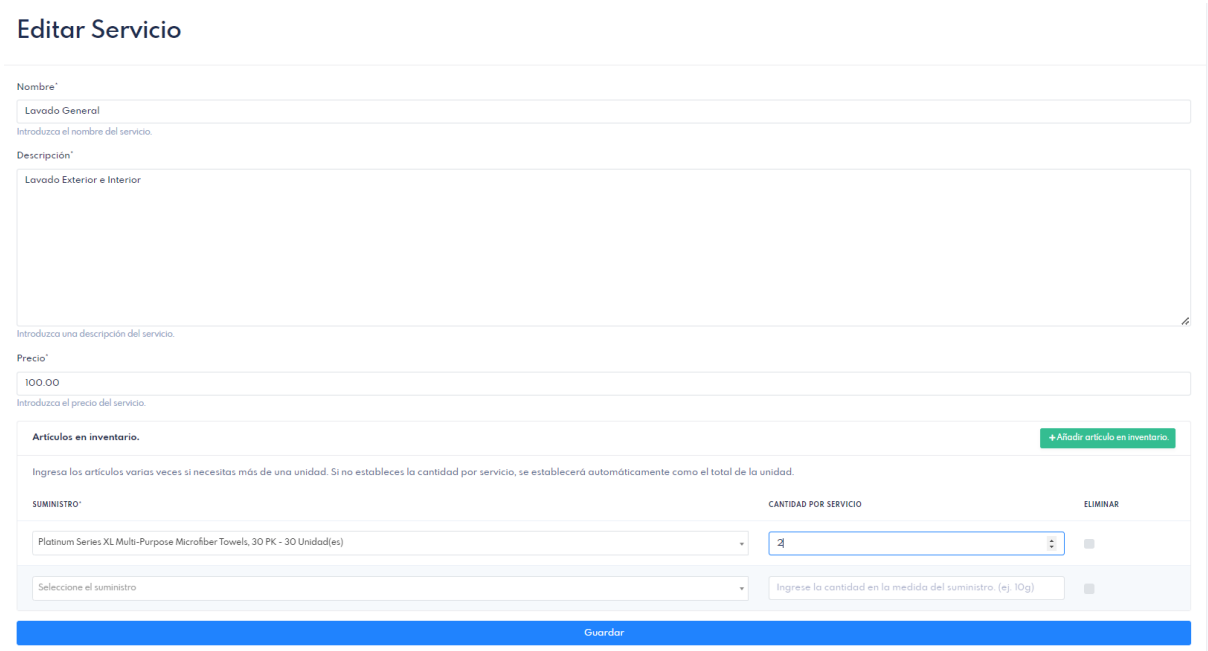


Figura 77

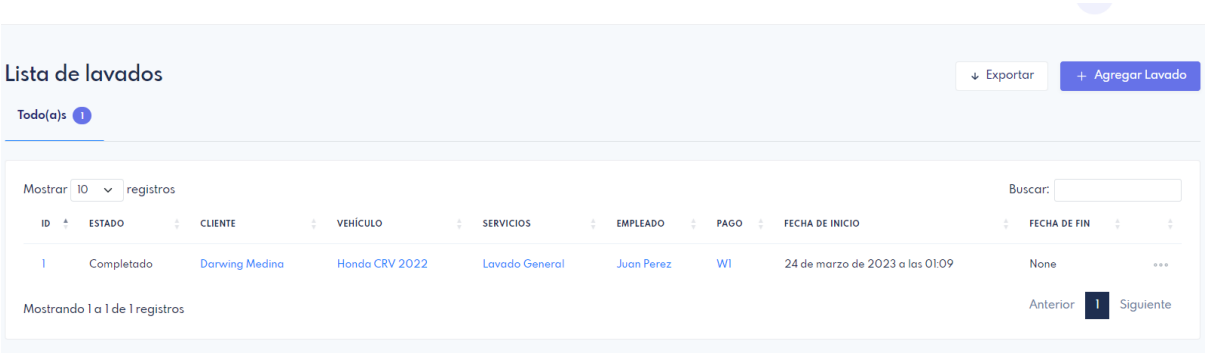
Eliminar Servicio: mensaje de confirmación

Gestión Lavados

En este formulario se muestra información sobre los lavados.

Figura 78

Pantalla para listado de lavados

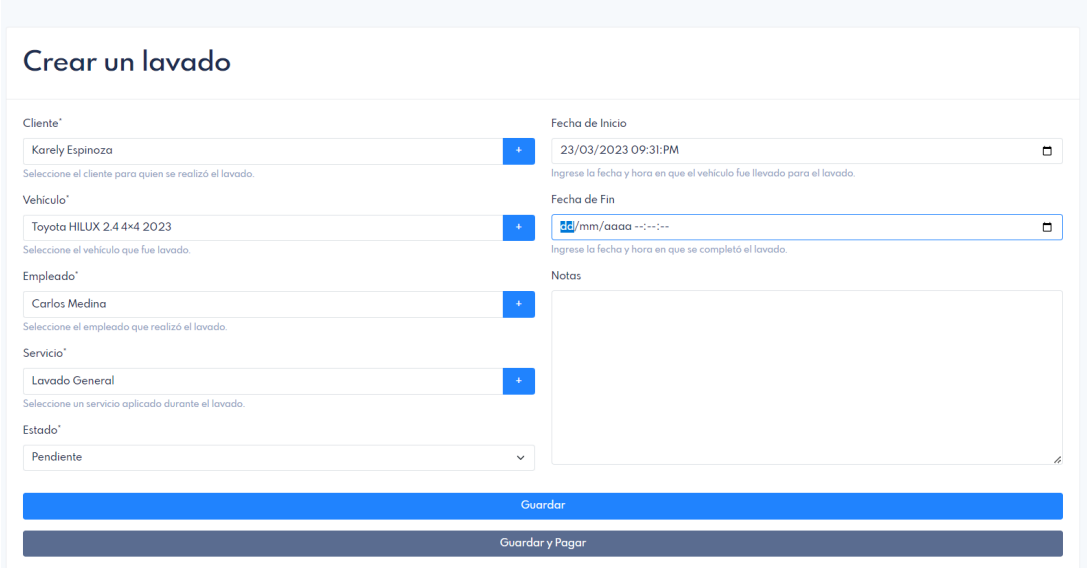


En el formulario de los lavados podrá encontrar las siguientes opciones:

Botón Agregar Lavado **:** Se desplegará el siguiente formulario que permite registrar la información de los lavados.

Figura 79

Pantalla para agregar un lavado



Opción Editar: Se despliega un formulario en donde permite actualizar la información de un lavado.

Figura 80

Pantalla para actualizar un lavado

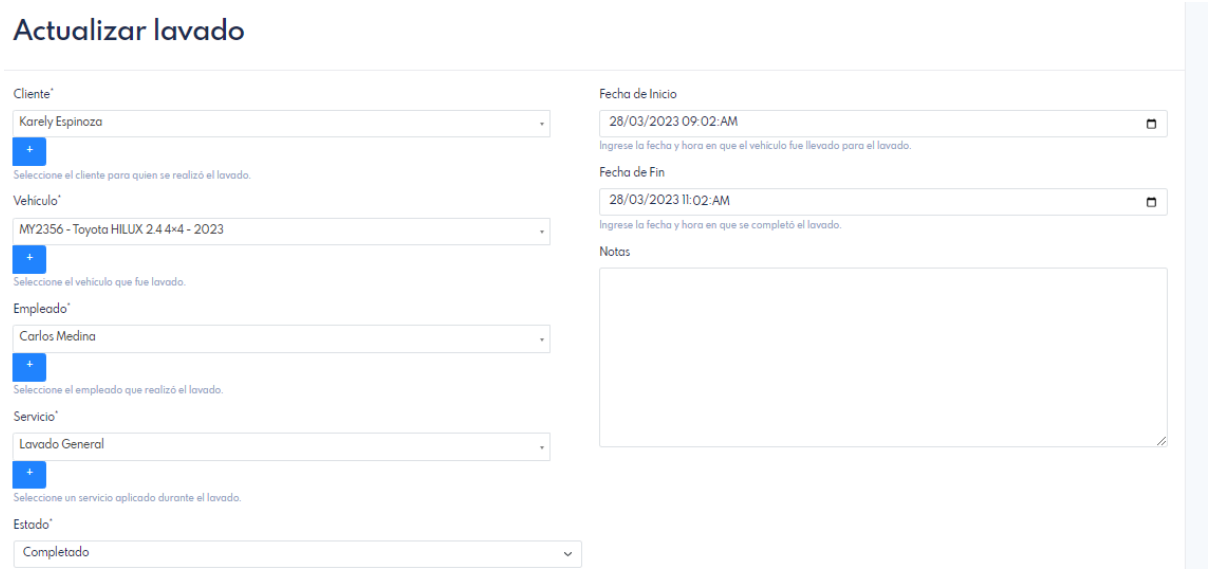
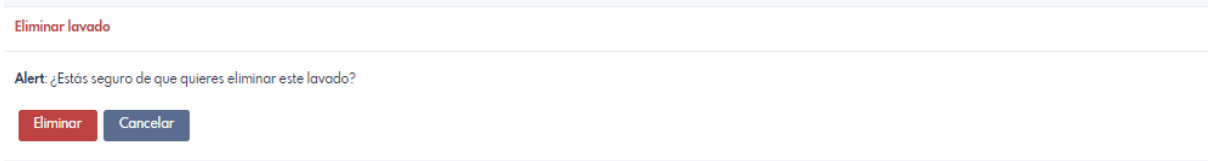


Figura 81

Eliminar lavado: mensaje de confirmación

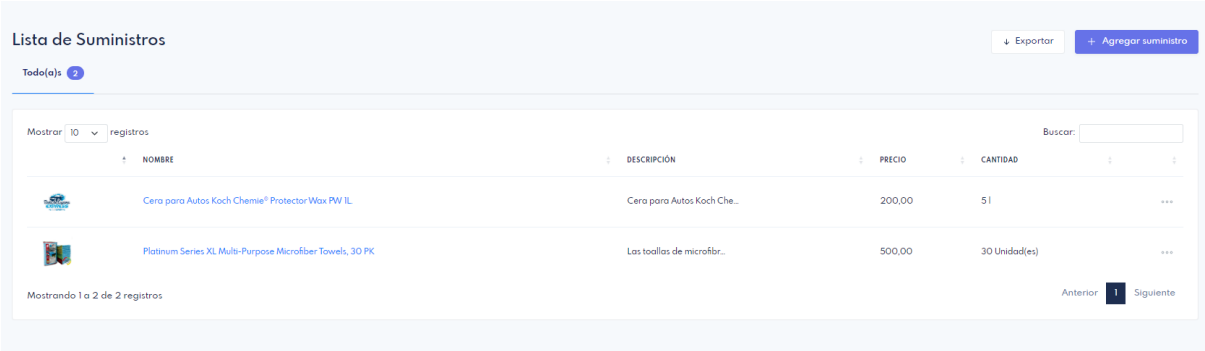


Gestión de Suministros

En este formulario se muestra información sobre los suministros del autolavado.

Figura 82

Pantalla lista de suministros

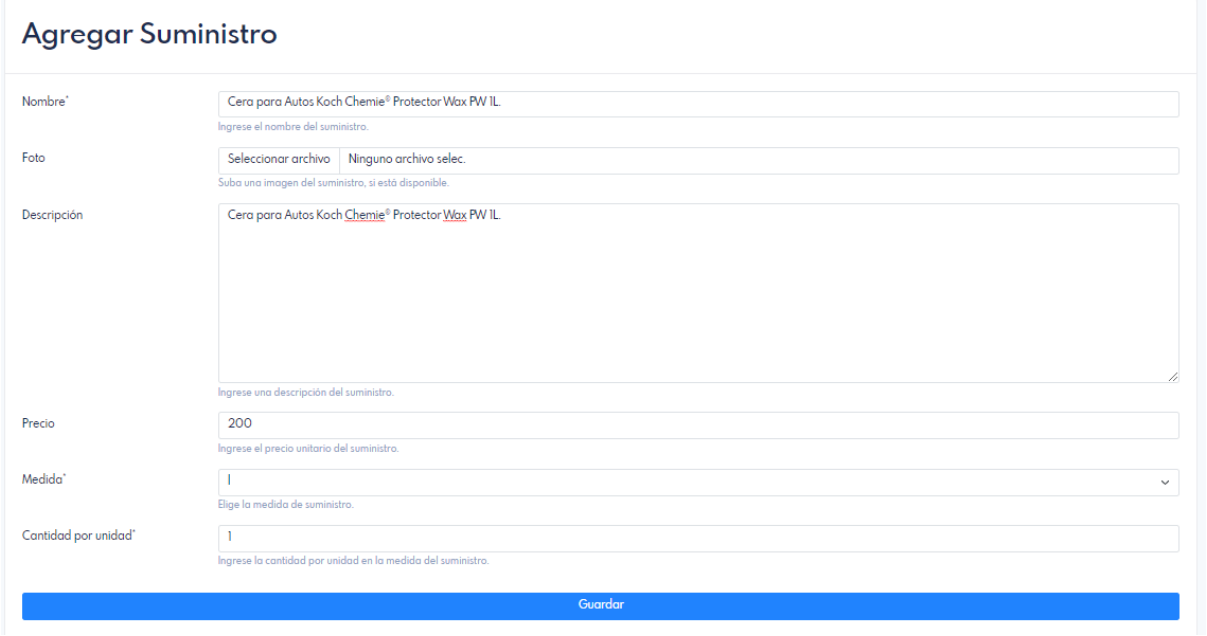


En el formulario de suministro podrá encontrar las siguientes opciones:

***Botón Agregar Suministro*:** Se desplegará el siguiente formulario que permite registrar la información de los suministros.

Figura 83

Pantalla para agregar suministro



Opción Editar: Se despliega un formulario en donde permite actualizar la información de un Suministro.

Figura 84

Pantalla para editar un suministro

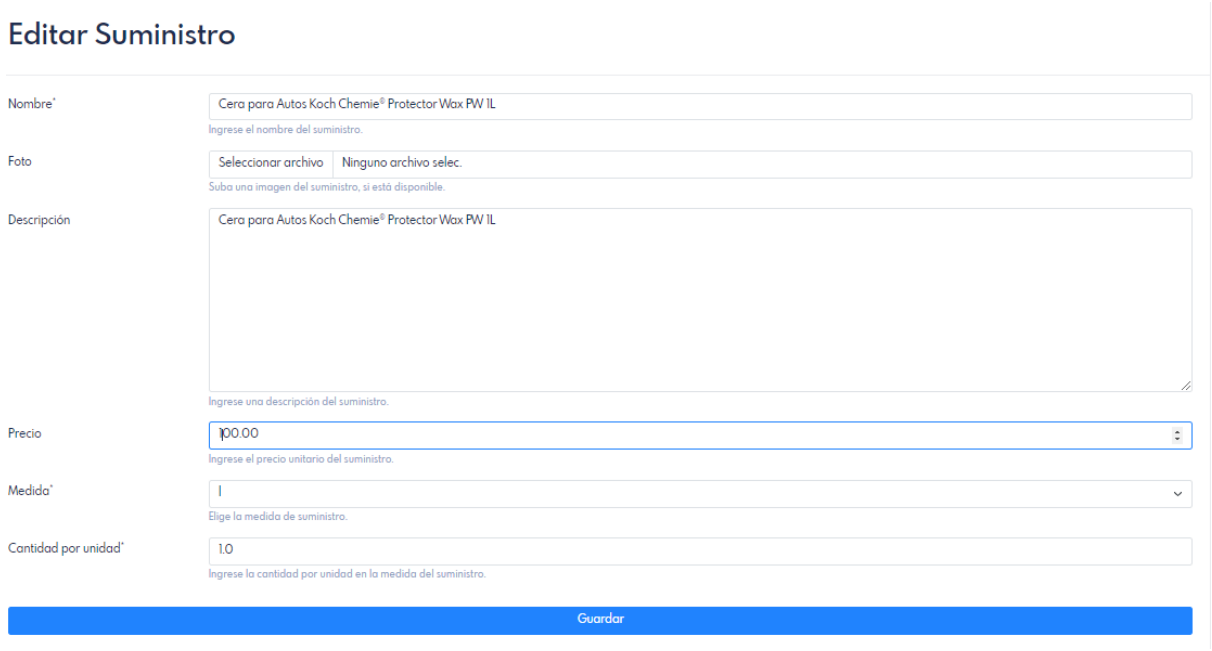
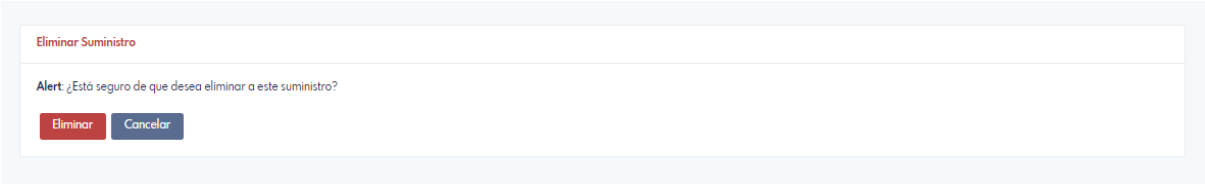


Figura 85

Eliminar Suministro: mensaje de confirmación

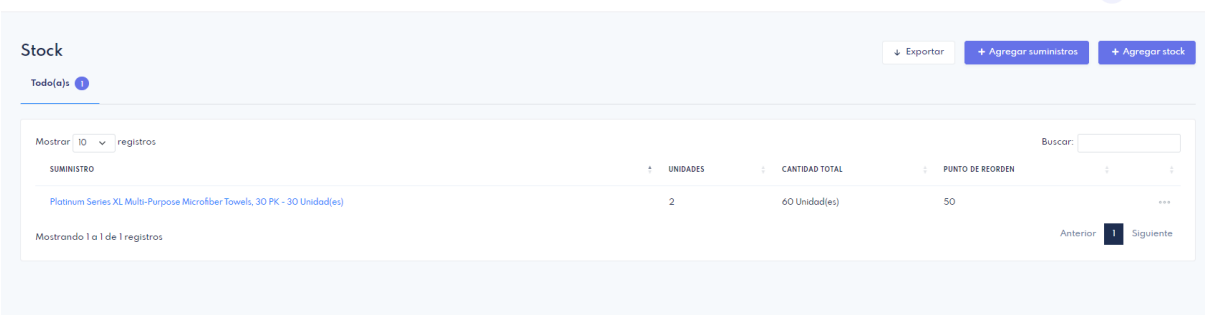


Stock

En este formulario se llevará a cabo el registro del stock de los productos

Figura 86

Pantalla lista stock de productos

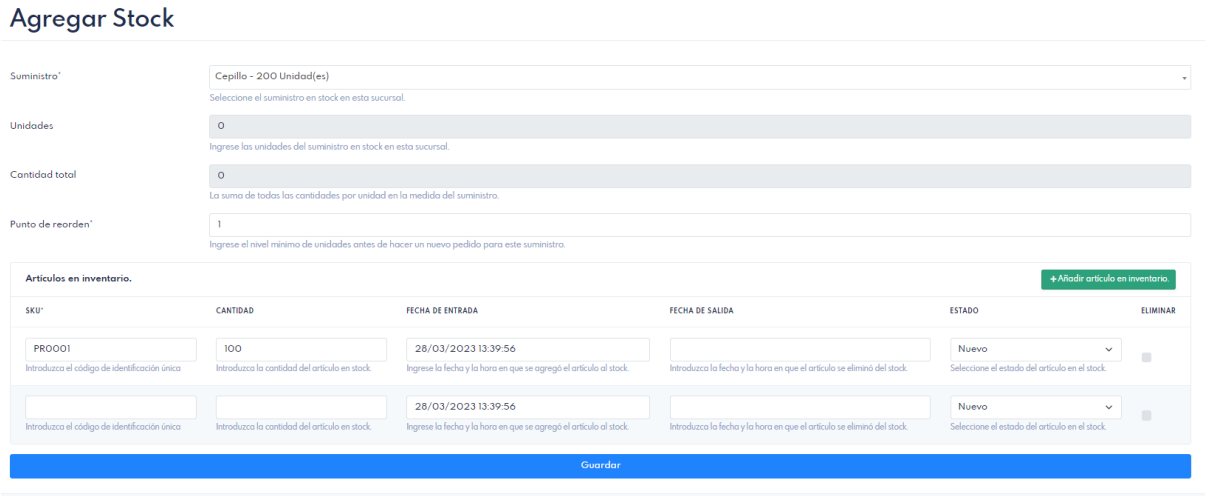


En el formulario de Stock podrá encontrar las siguientes opciones:

***Botón Agregar Stock*:** Se desplegará el siguiente formulario que permite registrar los productos en inventario.

Figura 87

Pantalla para agregar stock



***Opción Editar*:** Se despliega un formulario en donde permite actualizar la información del stock de un producto.

Figura 88

Pantalla para editar el stock de un producto

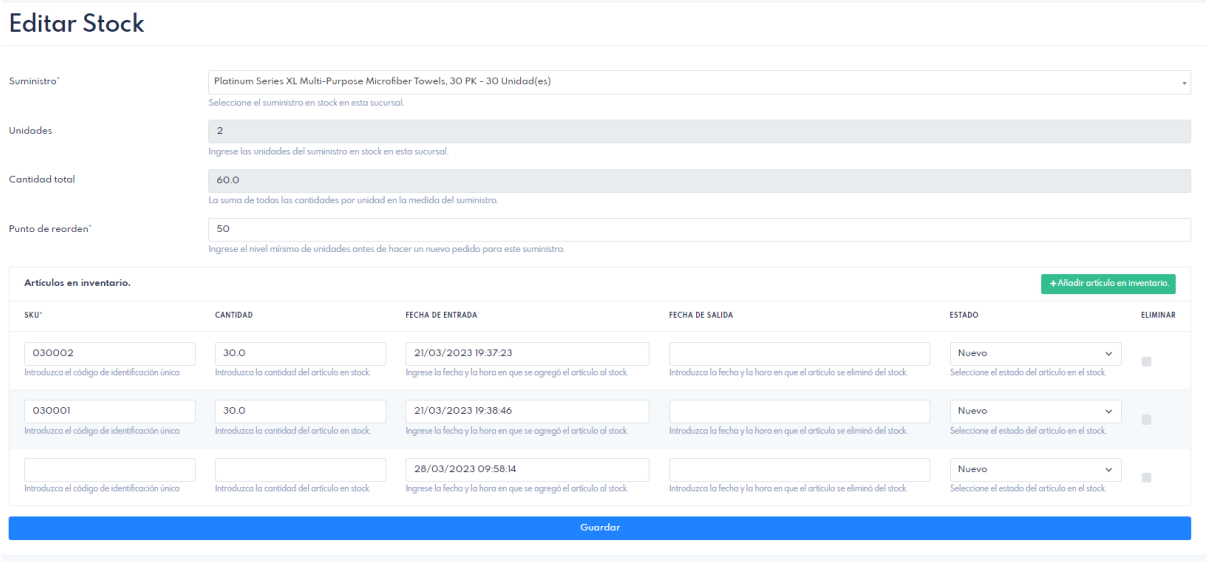
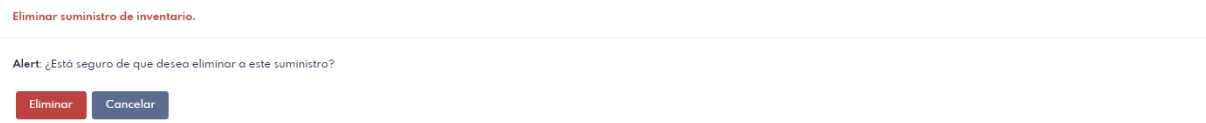


Figura 89

Eliminar suministro de inventario: mensaje de confirmación.

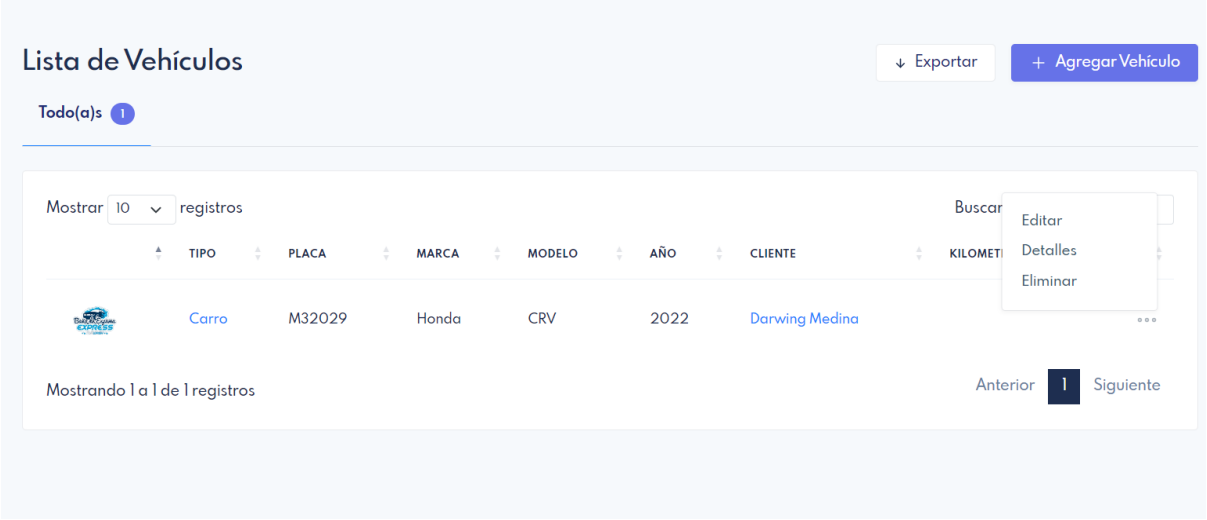


Gestión Vehículos

En este módulo se muestra información sobre los vehículos.

Figura 90

Pantalla lista de vehículos

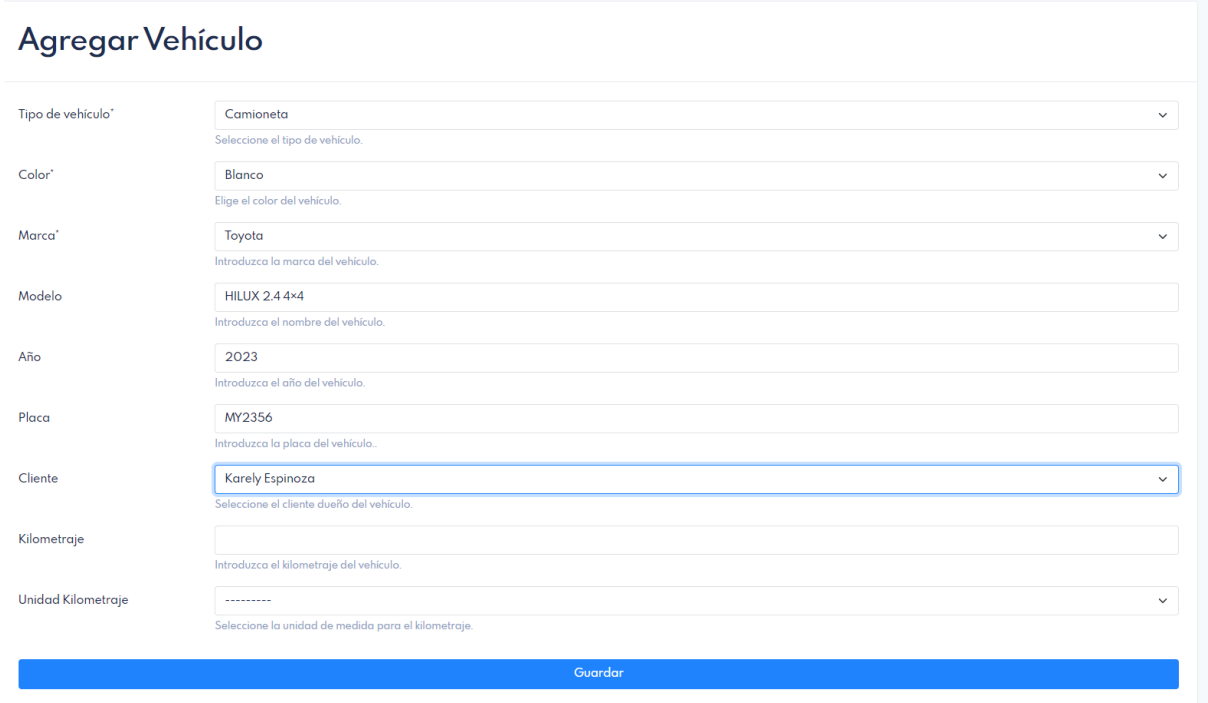


En el formulario de los vehículos podrá encontrar las siguientes opciones:

***Botón Agregar Vehículo*:** Se desplegará el siguiente formulario que permite registrar la información de los Vehículos.

Figura 91

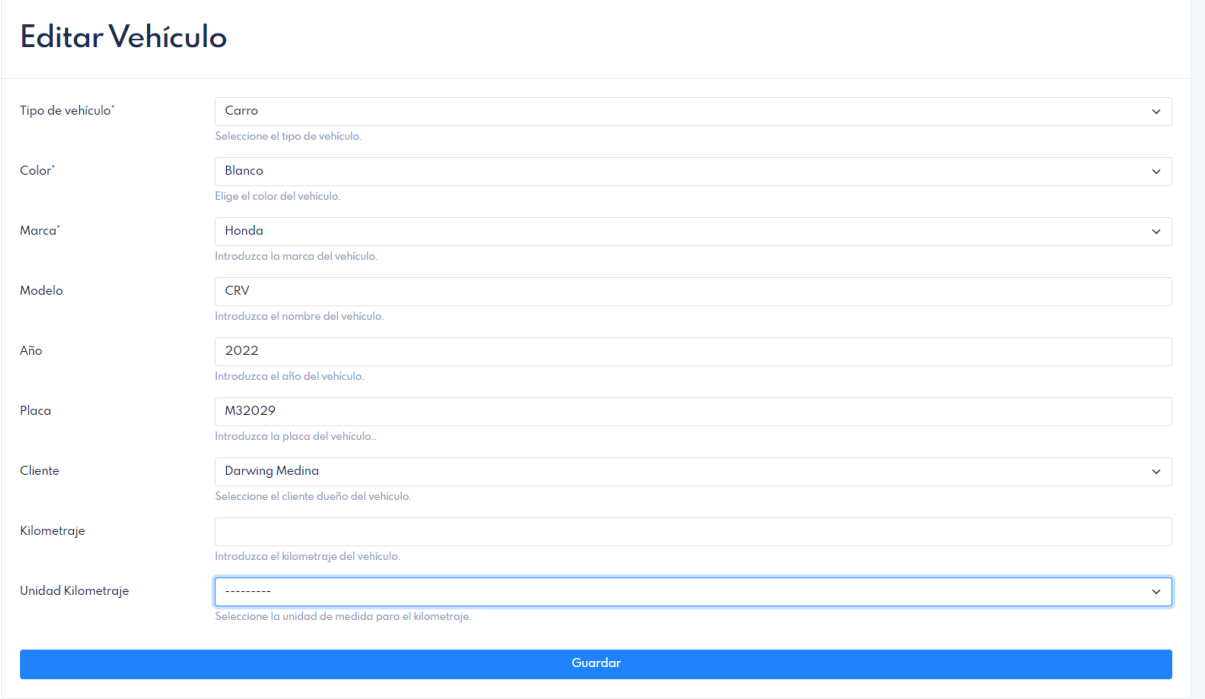
Pantalla para ingresar un vehículo



Opción Editar: Se despliega un formulario en donde permite actualizar la información de un vehículo.

Figura 92

Pantalla para editar la información de un vehículo



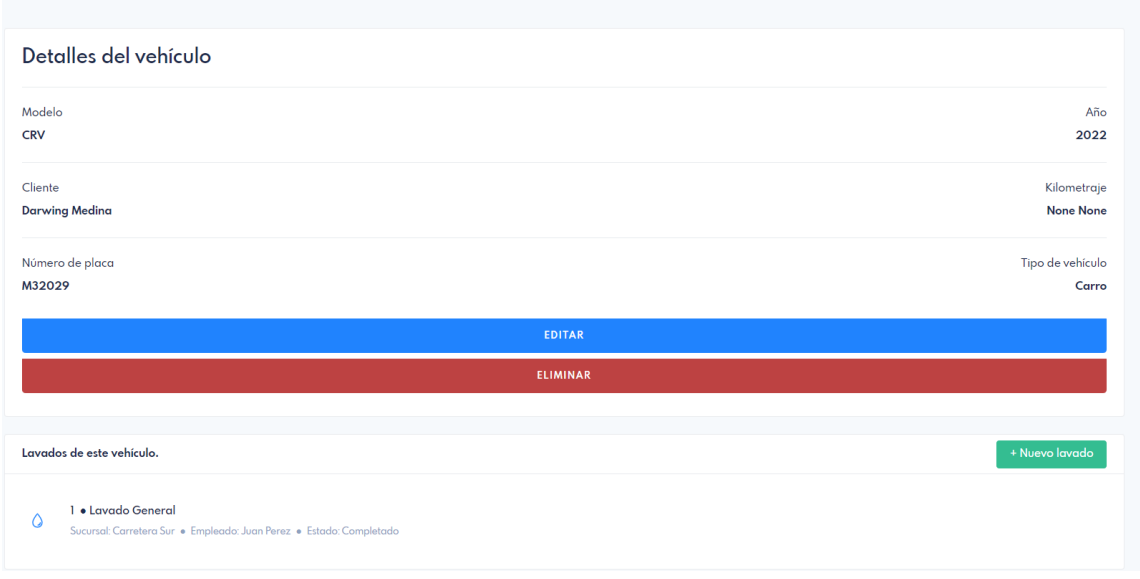
Opción Detalles *:* muestra los lavados que se han realizado a determinado vehículo, este formulario muestra las opciones:

Editar: Actualizar la información de vehículo.

Eliminar: esta opción da de baja a un Vehículo.

Figura 93

Pantalla que muestra la información general de un vehículo



,

Nuevo Lavado: Se desplegará el formulario de lavado, en donde se registra el lavado correspondiente a ese vehículo.

Figura 94

Pantalla para ingresar un nuevo lavado

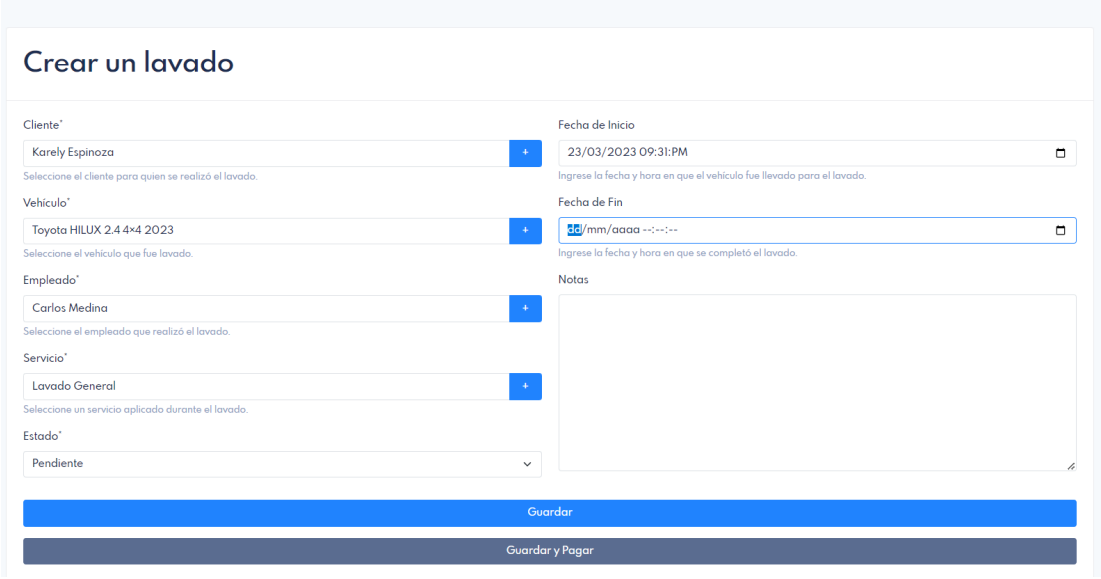


Figura 95

Eliminar Vehículo: mensaje de confirmación