



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Sistema de Gestión de Clientes y Pedidos para la Empresa Gas Chillán

Nicolás Alberto Luna Norambuena

Profesora Guía: María Antonieta Soto Chico

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

04 Septiembre de 2020

Chillán, Chile

Agradecimientos

Agradezco a la Vida por tener la oportunidad de realizar en paz la etapa final de mi carrera universitaria en medio de los tiempos desafiantes generados por la pandemia del COVID-19. A pesar de todas las contingencias sanitarias y problemas personales que se presentaron, he podido realizar este proyecto sin mayores preocupaciones o riesgos a los que han estado expuestos millones de personas a nivel mundial, por lo que me siento profundamente agradecido de estar en una condición segura junto a mis seres queridos.

A mi familia por su constante apoyo, especialmente a mi madre, Magdalena Norambuena, que se ha esforzado durante toda su vida para garantizar mi bienestar y educación.

También agradezco a mi novia, Diana Sepúlveda, por la motivación que me entrega para cumplir todos mis objetivos y tratar de ser un esfuerzo constante para ser mejor.

A la docente María Antonieta Soto quien guio y orientó todo el proceso de desarrollo de este proyecto mediante reuniones remotas y constantes correos.

A todos los funcionarios y docentes de la Universidad del Bío-Bío que han sabido adaptarse a la nueva normalidad y han puesto todos sus esfuerzos en seguir entregando el apoyo y educación a sus alumnos durante este particular semestre.

Nicolás Luna Norambuena

Resumen

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

El proyecto titulado “Sistema de Gestión de Clientes y Pedidos para la Empresa Gas Chillán” presentado en este informe, tiene como objetivo detallar y documentar el desarrollo de una aplicación web orientada a la automatización de procesos tanto operativos como administrativos de la empresa.

Gas Chillán es una distribuidora de balones de gas en la localidad de Chillán y Chillán Viejo con más de 10 años de trayectoria y cuenta con una gran cantidad de clientes que realizan pedidos por vía telefónica. Actualmente, la empresa posee su propio sistema utilizando documentos manuscritos y planillas Excel para llevar a cabo los diferentes procedimientos del servicio de distribución, no obstante, la empresa ha detectado que este sistema no tiene la capacidad suficiente para potenciar su estrategia de fidelización de clientes, debido a factores como la falta de reportes e indicadores oportunos que apoyen la toma de decisiones en el área administrativa.

Debido a la problemática expresada anteriormente, el sistema desarrollado en este proyecto ofrece una solución moderna, intuitiva y escalable que permite agilizar todas las operaciones involucradas en el proceso de registro de pedidos y clientes llevada a cabo por el área de recepción. Asimismo, genera un flujo de datos automatizado que proporciona información valiosa y sintetizada para el área de administración.

La aplicación web desarrollada se implementó usando tecnologías web modernas enfocadas en el desarrollo de aplicaciones robustas y escalables. Permite su uso mediante distintos dispositivos, ya que se aplican estrictos principios de usabilidad que garantizan una gran experiencia de usuario. Cabe destacar que la construcción de este sistema se separó en dos grandes módulos de desarrollo, el Frontend y el Backend. El primero corresponde a las interfaces de usuario y el consumo de datos siguiendo una arquitectura MVVM y utilizando el framework Angular. El segundo está orientado al desarrollo de la lógica de negocio en el lado del servidor orientada a proveer servicios web RESTful en Node.js utilizando una arquitectura por capas.

Abstract

This project is presented to comply with the requirements of the Universidad del Bío-Bío in the qualification process for the career of Civil Engineering in Computer Science.

The project entitled "Customer and Order Management System for the Gas Chillán Company" presented in this report, aims to detail and document the development of a web application aimed at automating both operational and administrative processes of the company.

Gas Chillán is a gas cylinder distributor in the town of Chillán and Chillán Viejo with more than 10 years of experience and has a large number of customers who place orders by telephone. Currently, the company has its own system using handwritten documents and Excel spreadsheets to carry out the different procedures of the distribution service, however, the company has detected that this system does not have sufficient capacity to enhance its customer loyalty strategy, due to factors such as the lack of reports and timely indicators that support decision-making in the administrative area.

Due to the problems expressed above, the system developed in this project offers a modern, intuitive and scalable solution that streamlines all the operations involved in the order and customer registration process carried out by the reception area. Also, managing to generate an automated data flow that provides valuable and synthesized information for the administration area.

The developed web application was implemented using modern web technologies focused on developing robust and scalable applications. In addition, allowing its use through different devices, since strict usability principles are applied that guarantee a great user experience. It should be noted that the construction of this system was separated into two development approaches Frontend and Backend, where the first is focused on everything related to user interfaces and data consumption following an MVVM architecture using the Angular framework, and the second is oriented to the development of business logic on the server side oriented to provide RESTful web services in Node.js using a layered architecture.

Índice General

1	<u>INTRODUCCIÓN</u>	12
2	<u>DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN</u>	14
2.1	DESCRIPCIÓN DE LA EMPRESA	14
2.2	DESCRIPCIÓN DE LA PROBLEMÁTICA	14
3	<u>DEFINICIÓN PROYECTO</u>	15
3.1	OBJETIVOS DEL PROYECTO	15
3.2	AMBIENTE DE INGENIERÍA DE SOFTWARE	15
3.2.1	METODOLOGÍA DE DESARROLLO	15
3.2.2	TECNOLOGÍAS	15
3.2.3	HERRAMIENTAS	16
3.3	ARQUITECTURA DE SOFTWARE	17
3.3.1	FRONTEND	17
3.3.2	BACKEND	19
3.3.2.1	API RESTful	19
3.3.2.2	Arquitectura por capas	21
4	<u>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</u>	23
4.1	ALCANCES	23
4.2	OBJETIVO DEL SOFTWARE	24
4.3	DESCRIPCIÓN GLOBAL DEL PRODUCTO	24
4.3.1	INTERFAZ DE USUARIO	24
4.3.2	INTERFAZ DE HARDWARE	25
4.3.3	INTERFAZ SOFTWARE	25
4.4	REQUERIMIENTOS ESPECÍFICOS	26
4.4.1	REQUERIMIENTOS FUNCIONALES DEL SISTEMA	26
4.4.2	INTERFACES EXTERNAS DE ENTRADA	28
4.4.3	INTERFACES EXTERNAS DE SALIDA	28
4.4.4	REQUISITOS NO FUNCIONALES	29
5	<u>FACTIBILIDAD</u>	30
5.1	ESTUDIO DE FACTIBILIDAD	30
5.2	FACTIBILIDAD TÉCNICA	30
5.3	FACTIBILIDAD OPERATIVA	31
5.4	FACTIBILIDAD ECONÓMICA	32
5.4.1	COSTO DE OPERACIÓN	32
5.4.2	INVERSIÓN	32
5.4.3	RESUMEN DE COSTOS	33
5.4.4	BENEFICIOS TANGIBLES	33
5.4.5	FLUJO DE CAJA NETO	34
5.4.6	CÁLCULO DEL VAN	34
5.5	CONCLUSIÓN DE LA FACTIBILIDAD	35

6	<u>ANÁLISIS</u>	36
6.1	PROCESO DE NEGOCIO DEL PROBLEMA A RESOLVER	36
6.2	DIAGRAMA DE CASOS DE USO	38
6.2.1	ACTORES	39
6.2.2	ESPECIFICACIÓN DE LOS CASOS DE USO	40
6.2.2.1	Caso de Uso: Iniciar Sesión	40
6.2.2.2	Caso de Uso: Cerrar Sesión	41
6.2.2.3	Caso de Uso: Ver Resumen Estadísticas Semanales	41
6.2.2.4	Caso de Uso: Ver Resumen Estadísticas Mensuales	42
6.2.2.5	Caso de Uso: Registrar Pedido	42
6.2.2.6	Caso de Uso: Listar Pedidos	43
6.2.2.7	Caso de Uso: Ver Detalle Pedido	43
6.2.2.8	Caso de Uso: Actualizar Pedido	44
6.2.2.9	Caso de Uso: Eliminar Pedido	44
6.2.2.10	Caso de Uso: Ver Pedidos en Curso	45
6.2.2.11	Caso de Uso: Confirmar Pedido	46
6.2.2.12	Caso de Uso: Cancelar Pedido	46
6.2.2.13	Caso de Uso: Listar Ventas	47
6.2.2.14	Caso de Uso: Ver Detalle Venta	47
6.2.2.15	Caso de Uso: Editar Venta	48
6.2.2.16	Caso de Uso: Eliminar Venta	49
6.2.2.17	Caso de Uso: Registrar Cliente	49
6.2.2.18	Caso de Uso: Listar Clientes	50
6.2.2.19	Caso de Uso: Ver Detalle Cliente	51
6.2.2.20	Caso de Uso: Actualizar Cliente	51
6.2.2.21	Caso de Uso: Eliminar Cliente	52
6.2.2.22	Caso de Uso: Registrar Usuario	53
6.2.2.23	Caso de Uso: Listar Usuarios	54
6.2.2.24	Caso de Uso: Ver Detalle Usuario	54
6.2.2.25	Caso de Uso: Actualizar Usuario	55
6.2.2.26	Caso de Uso: Eliminar Usuario	56
6.2.2.27	Caso de Uso: Seguimiento de Clientes	57
6.2.2.28	Caso de Uso: Enviar Correo a Cliente	58
6.2.2.29	Caso de Uso: Enviar Correos Masivos a Clientes	59
6.2.2.30	Ver Estadísticas de Ventas	60
6.2.2.31	Generar Informe de Ventas	61
6.3	MODELAMIENTO DE DATOS	62
7	<u>DISEÑO</u>	63
7.1	DISEÑO ORIENTADO AL CONSUMO DE SERVICIOS	63
7.1.1	RECURSOS Y SERVICIOS	63
7.2	REST (FRONTEND)	68
7.3	REST (BACKEND)	70
7.4	DIAGRAMA DE SECUENCIA Y DE CLASES	72

7.4.1	DIAGRAMAS DE CLASES (FRONTEND)	72
7.4.2	DIAGRAMAS DE SECUENCIA (BACKEND).....	75
7.5	DISEÑO DE FÍSICO DE LA BASE DE DATOS	76
7.5.1	MONGODB	76
7.5.2	JUSTIFICACIÓN DEL USO DE MONGODB	77
7.5.3	CONCEPTOS DE MONGODB.....	77
7.5.4	DISEÑO FÍSICO DE LA BASE DE DATOS.....	80
7.6	DISEÑO INTERFAZ Y NAVEGACIÓN	81
7.6.1	MAPAS DE NAVEGACIÓN	81
7.7	DISEÑO INTERFAZ DEL FRONTEND.....	83
7.7.1	INICIO DE SESIÓN DE LA APLICACIÓN (ESCRITORIO)	84
7.7.2	INICIO DE SESIÓN DE LA APLICACIÓN (MÓVIL)	85
7.7.3	MÓDULO DE RECEPCIÓN.....	86
7.7.3.1	Inicio (Escritorio)	86
7.7.3.2	Inicio (Móvil).....	87
7.7.3.3	Pedidos (Escritorio)	88
7.7.3.4	Pedidos (Móvil).....	89
7.7.3.5	Registrar Pedido (Escritorio)	90
7.7.3.6	Registrar Pedido (Móvil).....	92
7.7.3.7	Ver Pedidos En Curso (Escritorio)	94
7.7.3.8	Ver Pedidos En Curso (Móvil).....	95
7.7.4	MÓDULO DE ADMINISTRACIÓN.....	96
7.7.4.1	Inicio (Escritorio)	96
7.7.4.2	Inicio (Móvil).....	97
7.7.4.3	Seguimiento de Cliente (Escritorio).....	98
7.7.4.4	Seguimiento de Cliente (Móvil)	99
8	<u>PRUEBAS</u>	<u>100</u>
8.1	ESPECIFICACIÓN DE PRUEBAS	101
8.1.1	PRUEBAS FUNCIONALES	101
8.1.2	PRUEBAS DE SEGURIDAD	102
8.1.3	PRUEBAS DE USABILIDAD	103
8.2	DETALLE DE LAS PRUEBAS	104
8.2.1	DETALLE DE PRUEBA FUNCIONAL (MÓDULO RECEPCIÓN).....	104
8.2.2	DETALLE PRUEBA FUNCIONAL (MÓDULO ADMINISTRADOR)	106
8.2.3	DETALLE DE PRUEBA DE SEGURIDAD.....	107
8.2.4	DETALLE DE PRUEBAS DE USABILIDAD.....	109
8.2.4.1	Detalle de Pruebas de Usabilidad (Recepción)	111
8.2.4.2	Detalle de Pruebas de Usabilidad (Administración)	112
8.3	CONCLUSIONES DE PRUEBAS EFECTUADAS	113
9	<u>CONCLUSIONES</u>	<u>114</u>
10	<u>TRABAJO FUTURO</u>	<u>115</u>
11	<u>BIBLIOGRAFÍA</u>	<u>116</u>

12 ANEXOS	118
ANEXO 1: PRUEBAS FUNCIONALES	118
1.1 DETALLE DE PRUEBA FUNCIONAL (MÓDULO ADMINISTRACIÓN): REGISTRAR USUARIO	118
1.2 DETALLE DE PRUEBA FUNCIONAL (MÓDULO RECEPCIÓN): CREAR CLIENTE	120
1.4 DETALLE DE PRUEBA FUNCIONAL (MÓDULO RECEPCIÓN): CONFIRMAR PEDIDO	122
1.5 DETALLE DE PRUEBA FUNCIONAL (MÓDULO RECEPCIÓN): CREAR DIRECCIÓN.....	125
ANEXO 2: ENCUESTA EVALUACIÓN PRUEBAS DE USABILIDAD.....	127

Índice Tablas

<i>Tabla 1: Requisitos Funcionales</i>	27
<i>Tabla 2: Interfaces externas de entrada</i>	28
<i>Tabla 3: Interfaces externas de salida</i>	28
<i>Tabla 4: Costo de Operación</i>	33
<i>Tabla 5: Inversión</i>	33
<i>Tabla 6: Flujo de Caja Neto</i>	34
<i>Tabla 7: Actores del Sistema</i>	40
<i>Tabla 8: Caso de Uso Iniciar Sesión</i>	40
<i>Tabla 9: Caso de Uso Cerrar Sesión</i>	41
<i>Tabla 10: Caso de Uso Resumen de Estadísticas Semanales</i>	41
<i>Tabla 11: Caso de Uso Resumen de Estadísticas Mensuales</i>	42
<i>Tabla 12: Caso de Uso Registrar Pedido</i>	42
<i>Tabla 13 Caso de Uso Listar Pedidos</i>	43
<i>Tabla 14: Caso de Uso Ver Detalle de Pedido</i>	43
<i>Tabla 15: Caso de Uso Actualizar Pedido</i>	44
<i>Tabla 16: Caso de Uso Eliminar Pedido</i>	45
<i>Tabla 17: Caso de Uso Ver Pedidos en Curso</i>	45
<i>Tabla 18: Caso de Uso Confirmar Pedido</i>	46
<i>Tabla 19: Caso de Uso Cancelar Pedido</i>	46
<i>Tabla 20: Caso de Uso Listar Ventas</i>	47
<i>Tabla 21: Caso de Uso Ver Detalle Venta</i>	47
<i>Tabla 22: Caso de Uso Editar Venta</i>	48
<i>Tabla 23 Caso de Uso Eliminar Pedido</i>	49
<i>Tabla 24: Caso de Uso Registrar Cliente</i>	50
<i>Tabla 25: Caso de Uso Listar Clientes</i>	50
<i>Tabla 26: Caso de Uso Ver Detalle Cliente</i>	51
<i>Tabla 27: Caso de Uso Actualizar Cliente</i>	52
<i>Tabla 28: Caso de Uso Eliminar Cliente</i>	52
<i>Tabla 29: Caso de Uso Registrar Usuario</i>	53
<i>Tabla 30: Listar Usuarios</i>	54
<i>Tabla 31: Caso de Uso Ver Detalle Usuario</i>	54
<i>Tabla 32: Caso de Uso Actualizar Usuario</i>	55
<i>Tabla 33: Caso de Uso Eliminar Usuario</i>	56
<i>Tabla 34: Seguimiento de Clientes</i>	58
<i>Tabla 35: Enviar Correo a Cliente</i>	58
<i>Tabla 36: Enviar Correos Masivos a Clientes</i>	60
<i>Tabla 37: Ver Estadísticas de Ventas</i>	60
<i>Tabla 38: Caso de Uso Generar Informe de Ventas</i>	61
<i>Tabla 39: Recursos y Servicios</i>	65
<i>Tabla 40: Recursos y Servicios adicionales</i>	67
<i>Tabla 41: Ejemplo de petición POST</i>	69
<i>Tabla 42: Ejemplo de petición GET</i>	69
<i>Tabla 43: Especificación de Pruebas Funcionales</i>	101

Tabla 44: Especificación de Pruebas de Seguridad	102
Tabla 45: Especificación de Pruebas de Usabilidad	103
Tabla 46: Prueba de CU_05	105
Tabla 47: Pruebas de Seguimiento de Clientes.....	106
Tabla 48: Prueba de CU_01	108
Tabla 49: Resultado Encuesta de Usabilidad.....	110
Tabla 50: Resultados Pruebas de Usabilidad (Recepción)	111
Tabla 51: Resultados Pruebas de Usabilidad (Administración)	112
Tabla 52: Prueba de CU_22	119
Tabla 53: Prueba de CU_17	121
Tabla 54: Prueba de CU_11	124
Tabla 55: Prueba de Crear Dirección.....	126

Índice Figuras

<i>Figura 1: Arquitectura Frontend</i>	18
<i>Figura 2: Arquitectura API REST</i>	20
<i>Figura 3: Arquitectura Aplicación Express</i>	22
<i>Figura 4: Modelo de Proceso de Negocio (BPMN)</i>	37
<i>Figura 5: Diagrama de Casos de Uso</i>	38
<i>Figura 6: Modelo Entidad Relación (MER)</i>	62
<i>Figura 7: Secuencia de Servicios para Crear un Pedido</i>	66
<i>Figura 8: Diagrama de Clases del componente que implementa CU_05</i>	73
<i>Figura 9: Diagrama de Clases del componente que implementa CU_18</i>	74
<i>Figura 10: Diagrama de secuencia del servicio asociado a CU_05</i>	75
<i>Figura 11: Diagrama de secuencia del servicio asociado a CU_18</i>	76
<i>Figura 12: Comparación de Base de Dato Relacional y No Relacional [37]</i>	78
<i>Figura 13: Embedded Data Mongoddb [38]</i>	79
<i>Figura 14: References MongoDB [39]</i>	79
<i>Figura 15: Diseño Físico de la Base de Datos</i>	80
<i>Figura 16: Mapa de Navegación de Recepción</i>	81
<i>Figura 17: Mapa de Navegación de Administración</i>	82
<i>Figura 18: Interfaz de Inicio de Sesión</i>	84
<i>Figura 19: Interfaz de Inicio de Sesión - Móvil</i>	85
<i>Figura 20: Inicio Recepción</i>	86
<i>Figura 21: Inicio Recepción - Móvil</i>	87
<i>Figura 22: Pedidos Recepción</i>	88
<i>Figura 23: Pedidos Recepción - Móvil</i>	89
<i>Figura 24: Interfaz 1 Registrar Pedido</i>	90
<i>Figura 25: Interfaz 2 Registrar Pedido</i>	91
<i>Figura 26: Interfaz 1 Registrar Pedido - Móvil</i>	92
<i>Figura 27: Interfaz 2 Registrar Pedido - Móvil</i>	93
<i>Figura 28: Ver Pedidos en Curso</i>	94
<i>Figura 29: Pedidos en Curso - Móvil</i>	95
<i>Figura 30: Inicio Administración</i>	96
<i>Figura 31: Inicio Administración - Móvil</i>	97
<i>Figura 32: Seguimiento de Cliente</i>	98
<i>Figura 33: Seguimiento de Clientes - Móvil</i>	99

1 INTRODUCCIÓN

En la actualidad, el avance de las tecnologías de información crece exponencialmente y su uso en entornos empresariales cada vez es de mayor importancia. Nos encontramos en una época de constantes cambios, lo que exige a las empresas ser más competitivas, por lo tanto, necesitan agilizar y automatizar sus procesos para acceder a la información de manera más oportuna, de modo que les permita una correcta toma de decisiones y así potenciar sus estrategias que garanticen la calidad del servicio o producto que ofrecen y al mismo tiempo, su permanencia en el mercado.

No obstante, en la región de Ñuble, existen empresas que no implementan tecnologías de información por diferentes factores, en consecuencia, tienen fuertes problemas de productividad debido a que los sistemas que implementan no tienen la capacidad de gestionar demandas de su servicio o producto, ya que aún utilizan documentos manuscritos u otras tecnologías que no sacan el mayor provecho a su modelo de negocio.

Dado el contexto expresado anteriormente, la Empresa Distribuidora “Gas Chillán” ha tomado la decisión de modernizar su actual sistema de gestión de clientes y pedidos, dado que se encuentran en la ejecución de una estrategia de fidelización de clientes y dicho sistema no tiene la capacidad de ofrecer un rápido registro y acceso a la información, ni generar estadísticas que les permita llevar a cabo la estrategia de forma eficiente.

Este proyecto busca desarrollar una aplicación web que permita agilizar la mayoría de los procesos involucrados en la gestión de clientes y pedidos de Gas Chillán, implementando las principales funcionalidades requeridas, tales como registro ágil de pedidos, seguimiento de clientes, entre otras funcionalidades que se alinean con las necesidades de la empresa, permitiéndole llevar a cabo su estrategia de fidelización que garantiza la calidad de su servicio.

En este informe se encuentra descrito todo el proceso llevado a cabo para el desarrollo de la aplicación propuesta, cuenta exactamente con nueve capítulos que tienen como principal propósito dar a conocer al lector distintos aspectos relevantes de la solución implementada para el problema presentado.

En el segundo capítulo se expone información de la empresa y el contexto del problema a resolver. Posteriormente en el capítulo 3 se define el proyecto, específicamente los objetivos (general y específicos), el ambiente de ingeniería de software en donde principalmente se definen la arquitectura que sigue la aplicación en conjunto con las herramientas de desarrollo de software que se utilizan en este proyecto. En el cuarto capítulo se define el alcance y los requisitos del software a desarrollar. En el quinto capítulo, se analiza la factibilidad del proyecto, que busca expresar la viabilidad del proyecto desde una perspectiva operativa, técnica y económica. Luego, en el sexto capítulo se analiza el problema y el sistema a desarrollar mediante modelos que permiten una abstracción de dicho problema que permita comprenderlo de manera más exacta y detallada para facilitar la posterior etapa de diseño. El séptimo capítulo corresponde al diseño del software en donde se representan los datos del sistema mediante un modelo de entidad relación el cual es la base para el modelo físico de la base de datos que se presenta, además se define el sistema como un consumidor de servicios, se especifican los principales servicios REST, se representa cómo funciona el Frontend y Backend cuando se utilizan estos servicios mediante diagramas de secuencia y de clases, además se incluye las navegaciones y las interfaces de usuarios que implementan los módulos del Frontend. En el

octavo capítulo se detallan las pruebas realizadas en el sistema que buscan expresar y verificar la calidad del producto propuesto. Finalmente, en el noveno capítulo se presentan las conclusiones de este proyecto y las proyecciones de los trabajos futuros.

2 DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

2.1 Descripción de la empresa

Gas Chillán es una empresa dedicada al rubro de la distribución de gas con más de 10 años de trayectoria. Cuenta con un equipo capacitado para ofrecer una excelente atención a sus clientes, garantizando un servicio rápido y de calidad en las comunas de Chillán y Chillán Viejo. Actualmente, cuenta con una oficina central ubicada en Avenida Collín #420 en donde se realizan las principales actividades de la organización (Recepción y Bodega).

2.2 Descripción de la problemática

Gas Chillán actualmente posee una gran cantidad de clientes que realizan pedidos por vía telefónica. Para esto, cuenta con su propio sistema de información, en donde se llevan a cabo una variedad de procedimientos tanto operativos como administrativos que permiten realizar y gestionar su servicio de distribución de gas.

Uno de los principales problemas que mantiene esta empresa surge en el área de recepción, dado que, para llevar un registro de los pedidos de gas, se utilizan documentos manuscritos y planillas Excel, ralentizando la atención al cliente e impactando de forma negativa al proceso de toma de decisiones en el área administrativa.

Por otro lado, Gas Chillán está llevando a cabo una estrategia de fidelización de clientes con el fin de potenciar su presencia en el mercado y le permita garantizar su compromiso organizacional de ofrecer un servicio rápido y de calidad. No obstante, su actual sistema no tiene la capacidad suficiente para generar reportes e indicadores que apoyen y faciliten un flujo de datos automatizado al proceso de dicha estrategia empresarial.

Debido a lo expresado anteriormente, surge la necesidad de automatizar y agilizar los procesos de mayor importancia para la empresa, específicamente el registro de pedidos y clientes, ya que son procesos esenciales para desempeñar sus operaciones diarias de manera óptima y eficiente. A su vez, generan datos que se transforman en información valiosa para las decisiones administrativas que buscan lograr un alto nivel competitivo, por lo que deben estar accesibles de manera oportuna e intuitiva.

3 DEFINICIÓN PROYECTO

3.1 Objetivos del proyecto

General:

Desarrollar un sistema de gestión de clientes y pedidos para la Empresa Gas Chillán

Específicos:

- 1) Automatizar el proceso de registro de clientes y pedidos, de modo que permita al área de recepción agilizar la atención al cliente.
- 2) Proporcionar información cuantificada de venta de pedidos, de modo que permita al área de administración una acertada toma de decisiones.
- 3) Proveer acceso oportuno e intuitivo a la información asociada a clientes, de modo que facilite la estrategia de fidelización de clientes.

3.2 Ambiente de Ingeniería de Software

3.2.1 Metodología de Desarrollo

Para este proyecto se utilizó la Metodología de Desarrollo “Iterativa Incremental”, el enfoque de esta metodología es planificar el proyecto en diversos bloques temporales denominados iteraciones. Una iteración es una versión funcional del producto o proyecto a desarrollar, la cual no contempla la totalidad, pero sí una parte y es contemplada como una entrega al cliente. De este modo cada iteración (entrega) va sumando e integrando más funcionalidades hasta proporcionar un producto final [1].

Esta metodología es idónea para este proyecto, ya que cada iteración es una entrega funcional que podrá usar la empresa de gas, de esta manera, se obtiene retroalimentación continua para mejorar la interacción con el sistema y la experiencia de usuario.

3.2.2 Tecnologías

Las tecnologías utilizadas durante el proceso de desarrollo del proyecto están basadas en el desarrollo Full Stack de JavaScript, Stack MEAN (MongoDB, Express, Angular y Node.js), descritas a continuación.

Angular: Framework enfocado en el desarrollo Frontend del sistema a implementar, consiste en dar una interfaz gráfica al usuario capaz de proveer interacción óptima y una favorable experiencia de usuario [2].

Express: Es un framework para aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para el desarrollo de aplicaciones Backend [3].

Node.js: Es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma, que permite crear toda clase de herramientas de lado servidor (Backend) [4].

MongoDB: Base de datos no relacional basado en documentos que son almacenados en formato BSON (representación binaria de JSON). Posee un potente lenguaje de consulta que va acorde con la lógica del proyecto [5]. La justificación del uso de esta base de datos se encuentra en la sección 7.5.2

Por otro lado, cabe destacar que no se han utilizado librerías o frameworks para el diseño de las interfaces de usuario de esta aplicación. Se han diseñado con HTML 5 y Hojas de estilo en cascada (CSS) personalizadas, siguiendo la arquitectura de estilos proporcionada por Angular. A continuación, se describen estas tecnologías:

CSS: De sus siglas en inglés, Cascading Style Sheets, es lo que se denomina lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS separa el contenido de la representación visual del sitio [6]

HTML 5: De sus siglas en inglés “HyperText Markup Language”, es el lenguaje de marcado estándar para el desarrollo del contenido de las páginas y aplicaciones web, por lo que HTML 5, es el nombre que se usa para referirse a la quinta revisión [7]

3.2.3 Herramientas

Postman: Herramienta que permite el desarrollo de APIS para el Backend, ofreciendo funcionalidades como realizar peticiones, documentación automática, etc [8]. Para este proyecto, la herramienta es una gran utilidad para la prueba de la API REST durante el desarrollo de cada uno de los servicios.

Visual Studio Code: Editor de código fuente que permite trabajar con diversos lenguajes de programación [9] . En este proyecto, la herramienta permite escribir todo el código fuente del Frontend y Backend, ofreciendo una gran variedad características para el lenguaje Javascript, además de plugins que permiten agilizar todo el proceso de desarrollo

3.2.4 Hosting

Firebase Hosting: Es una plataforma desarrollada por Google para el desarrollo de aplicaciones web y ofrece un servicio de computación en la Nube, permitiendo alojar aplicaciones web estáticas y Single Page Applications (SPA) [10]. En este proyecto, el Frontend se aloja en esta plataforma, debido a que ofrece facilidades para crear un dominio personalizado, flexibilidad para el número de visitantes y agilizar el proceso de despliegue.

Heroku: Es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación y permite el alojamiento de aplicaciones web, tanto el cliente (Frontend) como el servidor (Backend) [11]. En este proyecto, el Backend se aloja en esta plataforma, debido a que ofrece herramientas ágiles para realizar modificaciones durante el desarrollo incremental de la aplicación alojada.

Mongo Atlas: Es la base de datos como servicio que permite implementar, utilizar y escalar una base de datos de MongoDB en la nube [12].

3.3 Arquitectura de Software

La arquitectura de software es un conjunto de patrones y lineamientos que proporcionan un marco de referencia necesario para definir y guiar la forma de trabajo de construcción de un software, permitiendo que todo el grupo de desarrollo pueda compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema debido a que se establece la estructura, funcionamiento e interacción entre las partes del software [13].

La construcción de este sistema se dividió en dos enfoques de desarrollo con el propósito de desacoplar la lógica de negocio y las interfaces de usuario; Frontend y Backend.

Para el cliente (Frontend) se utilizó el Framework Angular que implementa una arquitectura de software MVVM (Model View ViewModel). Por otro lado, se desarrolló una API (Backend) utilizando Node.js, que sigue una arquitectura REST para **proveer servicios web RESTful**, la cual se encarga de proveer todos los datos necesarios al cliente mediante una aplicación construida con el Framework Express, que implementa una arquitectura por capas.

A continuación, se detallan cada una de los enfoques y arquitecturas mencionadas anteriormente mediante definiciones y gráficas que representan el nivel más alto del diseño arquitectónico del sistema desarrollado.

3.3.1 Frontend

El Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el lado del cliente/navegador. HTML, CSS y JavaScript son los lenguajes principales del Frontend, de los que se desprenden una cantidad de frameworks y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios [14].

En este proyecto, se desarrolló el cliente del sistema utilizando Angular, el cual es un framework OpenSource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application o Aplicaciones de una sola página en español) utilizando el lenguaje TypeScript, el cual es un superconjunto del lenguaje JavaScript que añade tipado y objetos basados en clases [15]. Este framework sigue la arquitectura MVVM.

El patrón de arquitectura MVVM, de sus siglas en inglés “Model View ViewModel”, es un refinamiento del diseño MVC [16] mediante el cual se divide una aplicación en tres capas.

- **Model:** es la capa que representa toda la lógica de dominio de los datos que utiliza la aplicación.
- **Views:** es la capa que representa la parte visual de la aplicación, todo lo que tiene que ver con la gráfica de presentación de los datos.

- **ViewModel:** es la capa encargada de interactuar tanto con el modelo como con la vista e implementa toda la lógica de presentación de las interfaces de usuario en base al flujo de datos de la aplicación.

La principal diferencia entre MVC Y MVVM es que en MVVM, el “controller” cambia a “ViewModel” y en esta existe un “binder” que sincroniza la información entre vista y modelo, por lo tanto, lo que sucede en el modelo se cambia en la vista y viceversa.

La arquitectura mencionada anteriormente se implementa con el propósito de que la aplicación desarrollada en este proyecto sea altamente escalable y mantenible gracias a la estructura de archivos que provee MVVM.

A continuación, en la Figura 1 se presenta la interacción de la arquitectura implementada. La figura fue diseñada y confeccionada basándose en la información obtenida de la documentación oficial de Angular [17]

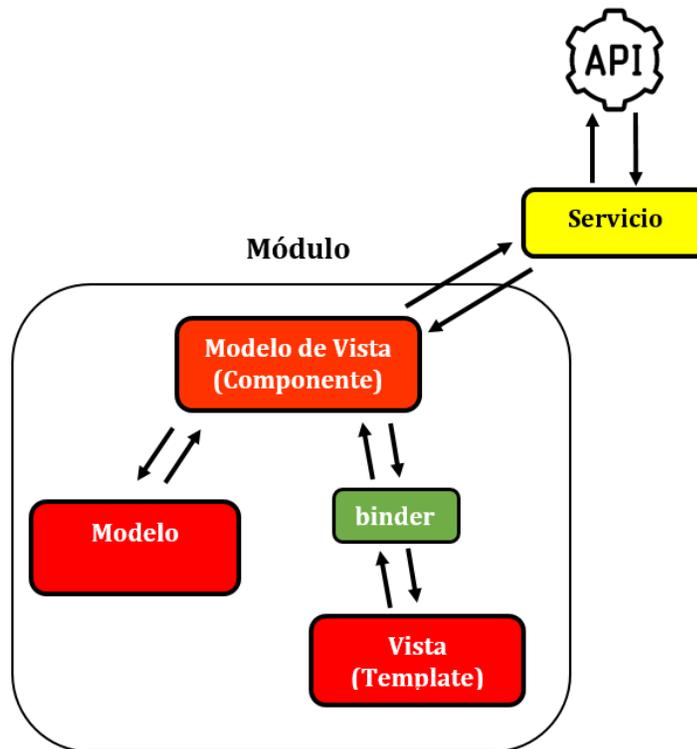


Figura 1: Arquitectura Frontend.

3.3.2 Backend

Backend es la capa de acceso a datos de un software o cualquier dispositivo que no es directamente accesible por los usuarios, además contiene la lógica que maneja dichos datos implementando una aplicación del lado del servidor.

En este proyecto se desarrolló una API utilizando el framework Express que sigue la arquitectura REST **y que se ejecuta en el entorno de Node.js**, el cual es un entorno en tiempo de ejecución multiplataforma que permite desarrollar aplicaciones del lado del servidor con el lenguaje de programación JavaScript. El framework Express básicamente proporciona una capa de abstracción para el desarrollo de aplicaciones que se ejecutan en Node.js e implementa toda la lógica de negocio y que ofrece como resultado la API para este proyecto. Esta aplicación básicamente se encarga de procesar todas las peticiones y respuestas mediante capas que interactúan entre sí.

3.3.2.1 API RESTful

Una API (Application Programming Interface) es un conjunto de funcionalidades que una pieza de software ofrece para que sea utilizada por otras piezas de software. REST (Representational State Transfer) es una arquitectura de software para conectar varios sistemas basados en el protocolo HTTP con el fin de obtener y generar datos u operaciones, devolviéndolos en formatos muy específicos, como XML y JSON (Particularmente JSON en este proyecto).

Por otro lado, API RESTful es un conjunto de funcionalidades que una aplicación web basada en la arquitectura REST ofrece a otras aplicaciones. Una API RESTful necesita seguir los siguientes principios:

- URL basada en recursos
- Respuesta en formato JSON o XML
- Usar HTTP Methods para las acciones
- Separar la API en recursos lógicos.
- Deben ser “stateless” (sin estado)

A continuación, en la Figura 2 se presenta la interacción de esta arquitectura. La figura fue diseñada y confeccionada basándose en los aprendizajes obtenidos de un curso de la plataforma Udemey [18]

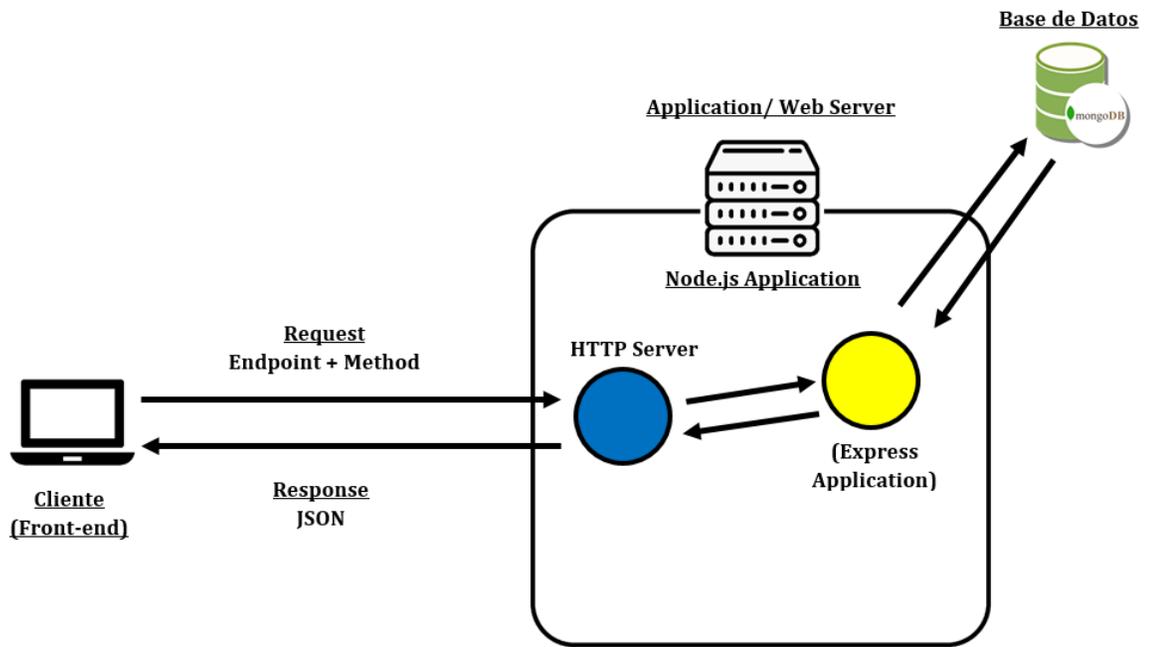


Figura 2: Arquitectura API REST.

3.3.2.2 Arquitectura por capas

La aplicación del lado del servidor que se desarrolló utilizando el framework Express y **que se ejecuta en el entorno Node.js**, sigue una arquitectura de tres capas, ya que se utiliza el principio “Principle of Separation of Concerns” [19] para separar la lógica empresarial de las rutas de la API que ofrece la aplicación al Frontend y así procesar todas las peticiones del cliente (Frontend) para retornar una respuesta con los datos solicitados . Las tres capas se describen brevemente a continuación:

- **Capa de Controlador:** se encarga de realizar toda la lógica de implementación de la aplicación, manipulando el flujo de trabajo e información por medio de la gestión de solicitudes y respuestas. Básicamente se preocupa de aspectos más técnicos de la aplicación [20].
- **Capa de Servicio:** se encarga de realizar todas las operaciones directamente relacionadas con la lógica de negocio de la aplicación [21].
- **Capa Acceso a Datos:** es la última capa de la arquitectura, en donde la dos anteriores dependen de esta para realizar todas las operaciones de almacenamiento de los datos de la aplicación en un medio no volátil (Base de Datos) [22].

A continuación, en la figura 3 se presenta la interacción de esta arquitectura, en donde “Node.js” es el entorno en donde se ejecuta “Express Application” (Framework descrito en 3.2.2). La figura fue diseñada y confeccionada basándose en los aprendizajes de un foro de ingenieros de software que expone temáticas de arquitectura de capas para un proyecto Node.js [23]

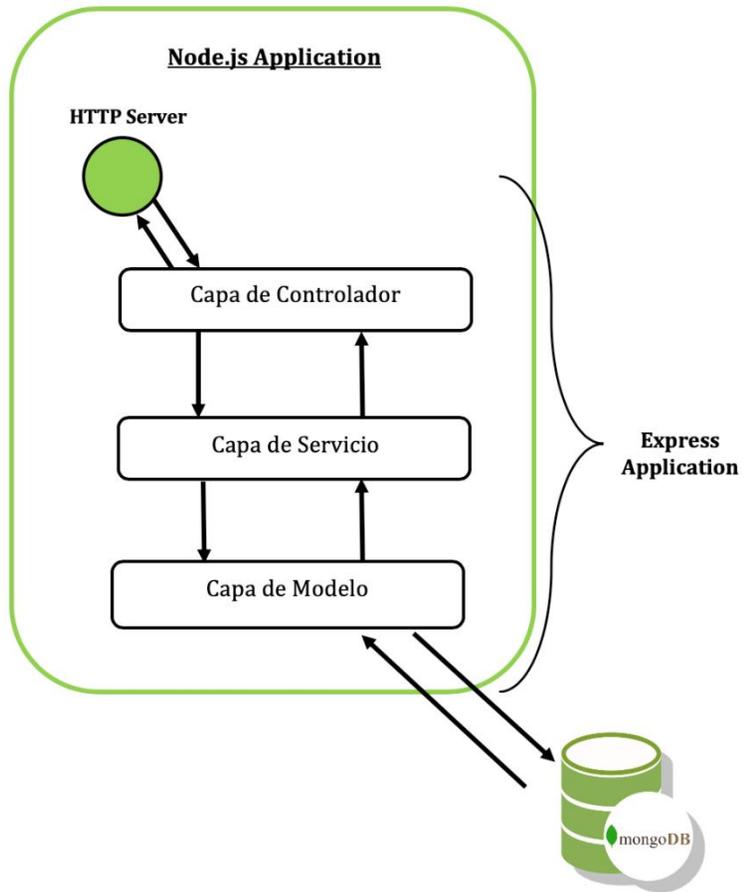


Figura 3: Arquitectura Aplicación Express.

4 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

4.1 Alcances

La aplicación web tiene como principal objetivo gestionar todos los pedidos que realizan los clientes a la empresa, en donde el usuario que posee el rol de recepcionista puede registrar dichos pedidos con el cliente que lo solicitó, y confirmarlo una vez que haya sido entregado físicamente. Además, puede realizar todas las operaciones básicas sobre los pedidos y clientes registrados como visualizarlos, mantenerlos actualizados y también eliminarlos. Por otro lado, el usuario que posee el rol de administrador podrá realizar todas las operaciones básicas sobre los clientes y usuarios del sistema, además de acceder a estadísticas de venta y una funcionalidad de seguimiento de clientes que permite obtener una predicción de compra de todos los clientes registrados y que han realizado pedidos.

La aplicación cuenta con dos perfiles distintos, los cuales se detallan a continuación en orden alfabético.

- **Administrador:** Es el administrador del sistema, quien obtiene información para la toma de decisiones. Puede ingresar nuevos usuarios y obtener estadísticas de clientes y ventas registradas en un periodo determinado.
- **Recepcionista:** Es el usuario que gestiona los pedidos que realizan los clientes, puede realizar todas las operaciones básicas sobre ambos.

Respecto a las limitaciones, cabe destacar que el cliente no se considera como un perfil debido a que no interactúa con el sistema, solo se gestiona su información para obtener estadísticas que ayuden a la toma de decisiones de la empresa.

El sistema no contempla una gestión física del reparto del pedido, solo se gestiona la información asociada a los pedidos para almacenarlos en un medio persistente, con el fin de agilizar el acceso a dicha información.

4.2 Objetivo del software

Haciendo referencia a la problemática descrita en la sección 2.2, se desarrolla una aplicación web que tiene como objetivo permitir la gestión de clientes y pedidos para las áreas de recepción y administración de la empresa.

4.3 Descripción Global del Producto

4.3.1 Interfaz de usuario

La interfaz de usuario debe contemplar criterios de usabilidad deseables que permitan al usuario interactuar con el sistema de manera intuitiva, aportando eficiencia y comodidad en el uso de las funcionalidades. En este proyecto se contemplan 4 criterios basados en estándares de usabilidad y en los principios de Jakob Nielsen [24].

- **Visibilidad del estado del sistema:** la aplicación debe mostrar en todo momento al usuario qué es lo que está sucediendo y en qué punto de navegación se encuentra.
- **Prevención de errores:** mostrar mensajes de confirmación antes de realizar una acción importante que puedan conducir al usuario a un error.
- **Estética y diseño minimalista:** se debe simplificar el contenido y eliminar lo irrelevante para permitir que el usuario solo se enfoque en lo importante.
- **Flexibilidad y eficiencia de uso:** se debe personalizar acciones frecuentes.

Además, cabe destacar que el sistema tiene un enfoque de uso “Desktop First o Escritorio Primero”, es decir, la aplicación se utilizará la mayor parte del tiempo mediante dispositivos laptop o computador de escritorio. De igual manera, pero con menos frecuencia, se utilizará la aplicación por medio de dispositivo móviles como teléfonos inteligentes o tablets, por lo tanto, la interfaz debe ser Responsive, es decir, se debe adaptar automáticamente al tamaño de pantalla de estos dispositivos.

Es importante mencionar que los usuarios que van a interactuar con el sistema, son personas con experiencia básica en computación. Por lo tanto, los mensajes deben cumplir estrictamente con las características de usabilidad mencionadas.

Por otro lado, la aplicación implementa una barra lateral de navegación (sidebar), la cual despliega distintas opciones dependiendo del rol del usuario que inició sesión. Además, cuenta con una barra superior que actúa como barra de acciones y permite realizar configuraciones de la cuenta del usuario, específicamente la opción de cerrar sesión.

4.3.2 Interfaz de Hardware

Dependiendo de donde se accede a la aplicación web, el sistema interactúa con los siguientes dispositivos:

- Desde una laptop o computador de escritorio:
 - Mouse
 - Pantalla
 - Teclado

- Desde un dispositivo móvil:
 - Pantalla táctil

4.3.3 Interfaz Software

La aplicación web requiere de conexión a Internet mediante Ethernet (cable), Wifi (inalámbrica) o redes de datos móviles (3G- 4G). Además, requiere un navegador web actualizado como Google Chrome, Mozilla Firefox, Microsoft Edge o Safari. Se recomienda un navegador actualizado ya que estos garantizan una grata experiencia de usuario que permite interactuar con la aplicación de manera más ágil y eficiente.

4.4 Requerimientos Específicos

4.4.1 Requerimientos Funcionales del sistema

La Tabla 1 muestra los requerimientos funcionales del sistema a desarrollar

Id	Nombre	Descripción
R_01	Autenticación	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado
R_02	Resumen de Estadísticas Semanal	El sistema muestra al usuario de recepción un conjunto de estadísticas cuantitativas resumidas sobre clientes, pedidos y ventas que ha realizado durante la semana.
R_03	Resumen de Estadísticas Mensual	El sistema muestra al usuario de administración un conjunto de estadísticas cuantitativas resumidas sobre clientes, pedidos y ventas que se han realizado durante el mes.
R_04	Registro de Pedidos	El sistema permite que recepción pueda ingresar los pedidos recibidos.
R_05	Listar Pedidos	El sistema permite que recepción pueda ver todos los pedidos registrados en base a filtros de búsqueda
R_06	Ver Detalle de Pedido	El sistema permite que recepción pueda ver el detalle de un pedido registrado en base a filtros de búsqueda
R_07	Actualizar Pedido	El sistema permite que recepción pueda actualizar un pedido registrado
R_08	Eliminar Pedido	El sistema permite que recepción pueda eliminar un pedido registrado
R_09	Ver Pedidos en Curso	El sistema permite ver los pedidos realizados durante el día y que están en curso (no confirmados) de manera intuitiva.
R_10	Confirmar / Cancelar Pedido	El sistema permite confirmar o cancelar un pedido en curso
R_11	Generar Venta	El sistema guardará como venta todos los pedidos que hayan sido confirmados.
R_12	Listar Ventas	El sistema permite que recepción pueda ver todas las ventas registradas en base a filtros de búsqueda
R_13	Ver Detalle de Venta	El sistema permite que recepción pueda ver el detalle de una venta registrada
R_14	Actualizar Venta	El sistema permite que recepción pueda actualizar una venta registrada
R_15	Eliminar Venta	El sistema permite que recepción pueda eliminar una venta registrada.
R_16	Registrar Cliente	El sistema permite crear un cliente de manera explícita con todos los datos requeridos.
R_17	Listar Clientes	El sistema permite que recepción pueda ver todos los clientes registrados en base a filtros de búsqueda.
R_18	Ver Cliente	El sistema permite que recepción pueda ver un cliente registrado en base a filtros de búsqueda.
R_19	Actualizar Cliente	El sistema permite que recepción pueda actualizar un cliente registrado
R_20	Eliminar Cliente	El sistema permite que recepción pueda eliminar un cliente registrado
R_21	Registrar Usuario	El sistema permite que administración ingrese nuevos usuarios al sistema con sus correspondientes credenciales únicas (Rut y Contraseña)
R_22	Listar Usuarios	El sistema permite que administración pueda ver todos los usuarios registrados en base a filtros de búsqueda.

R_23	Ver Usuario	El sistema permite que administración pueda ver un usuario registrado en base a filtros de búsqueda.
R_24	Actualizar Usuario	El sistema permite que administración pueda actualizar un usuario registrado.
R_25	Eliminar Usuario	El sistema permite que administración pueda eliminar un usuario registrado.
R_26	Seguimiento de Clientes	El sistema permite que administración pueda tener un seguimiento de los clientes para la predicción de compra de estos.
R_27	Enviar Correo a Cliente	El sistema permite que administración pueda enviar un correo a un cliente en específico cuando este exceda su fecha de estimación de compra.
R_28	Enviar Correos Masivos a Clientes	El sistema permite que administración pueda enviar correos de manera masiva a los clientes que excedan su fecha de estimación de compra.
R_29	Estadísticas de Ventas	El sistema genera estadísticas e informes de ventas y la posibilidad de descargarlos como un documento en formato PDF.

Tabla 1: Requisitos Funcionales.

4.4.2 Interfaces externas de entrada

En la Tabla 2 se presentan las interfaces externas de entrada.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem
DE_01	Datos de Autenticación	RUT, CONTRASEÑA
DE_02	Datos de Pedido	TELEFONO_CLIENTE, DIRECCION, USUARIOS, CILINDROS, TIPO_PAGO, SUBTOTAL, DESCUENTO, TOTAL, OBSERVACION
DE_03	Datos de Cliente	TELEFONO, NOMBRE, APELLIDO, DIRECCION, TIPO_CLIENTE, EMAIL
DE_04	Datos de Usuario	RUT, CONTRASEÑA, NOMBRES, APELLIDOS, ROL, IMAGEN, FECHA_INGRESO, EMAIL

Tabla 2: Interfaces externas de entrada

4.4.3 Interfaces externas de Salida

En la Tabla 3 se presentan las interfaces externas de entrada.

Identificador	Nombre del ítem.	Detalle de Datos contenidos en ítem	Medio Salida
IS_01	Estadísticas Semanales	TOTAL PEDIDOS, TOTAL VENTAS, META DE PEDIDOS, GRÁFICA DE VENTAS	Pantalla
IS_02	Estadísticas Mensuales	TOTAL PEDIDOS, TOTAL VENTAS, GRÁFICA DE VENTAS	Pantalla
IS_03	Seguimiento de Clientes	DATOS CLIENTES, PERIODICIDAD DE COMPRA, ESTADO DE PREDICCIÓN	Pantalla
IS_04	Informe de Ventas	DATOS, TOTAL VENTAS, GRAFICA DE VENTAS, TOTALES	Pantalla Archivo PDF

Tabla 3: Interfaces externas de salida.

4.4.4 Requisitos No Funcionales

Los requisitos No Funcionales son aquellos que imponen restricciones de diseño o implementación y definen propiedades o cualidades que el producto debe tener. A continuación, se presenta una lista de los Requisitos No Funcionales del sistema:

- El sistema incluirá un proceso de autorización de usuarios, en el cual deben identificarse usando credenciales como Rut y Contraseña. La autenticación debe persistir durante máximo 7 días, luego de eso el sistema debe exigir la re-autenticación del usuario con las credenciales correspondientes.
- El sistema validará toda la información ingresada en los formularios en el cliente y en el servidor.
- El sistema mostrará al usuario mensajes de alerta cuando cometa errores al interactuar con las diferentes funcionalidades, indicándolo de manera clara la causa y solución.
- El sistema mostrará de manera intuitiva la información que requiera el usuario (recepción y administración) mediante tablas interactivas.
- El sistema otorgará paginación y ordenamiento de datos en las distintas tablas presentes en el sistema (Pedidos y Clientes).
- El sistema permitirá generar informes mediante opciones que indiquen dicha acción al usuario.
- El sistema mostrará un mensaje de confirmación antes de realizar una acción importante o irreversible.
- El sistema permite ser utilizado principalmente mediante dispositivos de escritorio y móviles, adaptándose a los distintos tamaños de pantalla.

5 FACTIBILIDAD

5.1 Estudio de Factibilidad

El estudio de factibilidad es una forma de medir los beneficios del desarrollo de un sistema de información. En este proyecto en particular, se desarrolla un sistema que gestiona los clientes y pedidos para la Empresa Distribuidora Gas Chillán. Dado lo anterior, la factibilidad permite determinar los recursos disponibles por parte del desarrollador (estudiante), las habilidades, costos e inversiones que implican el proceso de desarrollo del sistema.

5.2 Factibilidad técnica

La factibilidad técnica considera la tecnología hardware y software que se requiere para el desarrollo y operación de la solución propuesta. A continuación, se indican las tecnologías y equipos que son necesarios:

- Computador con los siguientes requisitos mínimos:
 - Memoria RAM 4 GB
 - Intel® Core™ i5-760 o superior
 - Almacenamiento con 10 GB de espacio disponible
 - Sistema Operativo Windows 10 Versión Home o superior
- Conexión a internet de banda ancha
- Licencia de Office
- Web Hosting (Servidor)
- Software de Desarrollo:
 - Visual Studio Code (User)
 - Google Chrome
 - MongoDB
 - Mongo Compass
 - yEd Graph

En conclusión, se cuenta con todos los recursos técnicos requeridos para llevar a cabo el desarrollo de la solución propuesta.

5.3 Factibilidad operativa

Para esta factibilidad se considera principalmente la habilidad del desarrollador (Ingeniero Civil Informático) para efectos de que el sistema a desarrollar cumpla con las expectativas esperadas. Debe poseer la suficiente destreza en las tecnologías de desarrollo web y la capacidad para comprender los requisitos del sistema. Compaginando estas habilidades, se garantiza un desarrollo óptimo y eficiente. A continuación, se muestran las tecnologías que deben ser dominadas por el desarrollador:

Full Stack MEAN:

- MongoDB
- Express
- Angular 9+
- Node.js 10+

También, un aspecto importante a considerar, son los conocimientos técnicos para la puesta en marcha del sistema, que son señalados (de manera general) a continuación:

- Llevar la aplicación a producción, que básicamente es subir la API REST (Backend) y el cliente (Frontend) a un servidor de pago anual (Web Hosting).
- Pruebas de la aplicación (Unitarias e Integración)

Además, la capacidad de los usuarios que van a interactuar con el sistema debe ser de nivel básico en el manejo de computadoras. Deben poseer específicamente los siguientes conocimientos básicos:

- Uso de Computadoras Nivel Usuario
- Uso de Chrome Nivel Usuario (Escritorio y Móvil)

En conclusión, se cuenta con todos los conocimientos y habilidades por parte del desarrollador para llevar a cabo el proyecto. Además, gran parte de los usuarios que van a operar e interactuar con el sistema, poseen los conocimientos básicos en el uso de computadoras, ya que, en la distribuidora, los usuarios trabajan directamente con el computador o dispositivo móvil.

5.4 Factibilidad económica.

El estudio de la factibilidad económica permite analizar los costos y beneficios del proyecto con el fin de verificar que se cuenta con los recursos económicos necesarios para el desarrollo del proyecto, además de determinar si el proyecto es rentable.

5.4.1 Costo de Operación

El desarrollo y posterior despliegue de la aplicación requieren de un servidor para alojar la aplicación (Frontend y Backend), se estima que el valor anual de un servicio de este tipo es de \$20.000 aproximadamente en la nube de Heroku¹. Además, para alojar la base de datos de MongoDB se usará el servicio proveído por Mongo Atlas², con una anualidad de \$10.000, considerando que el tipo de dato que se almacenará es texto en su mayoría.

5.4.2 Inversión

Se necesita un Ingeniero Civil en Informática con especialización en Desarrollo de Software y considerando que el desarrollo de un sistema con estas características ronda los 3 meses, se requiere una inversión de \$2.100.000³ por concepto de sueldo del ingeniero.

Los costos que implica la implementación del sistema están directamente relacionados con los requerimientos técnicos necesarios para el desarrollo del proyecto. En lo que refiere al hardware, se dispone de todos los equipos señalados anteriormente, lo cual no se necesita invertir en este aspecto. En cuanto a software, la gran parte de ellos son gratuitos y libres, la excepción la constituyen los dos siguientes productos:

- Licencia de Windows 10 Home: \$170.500⁴
- Licencia Microsoft Office: \$50.000 anual⁵

¹ Precio obtenido de: www.heroku.com/pricing

² Precio obtenido de: www.mongodb.com/pricing

³ Cantidad Obtenida de: www.mifuturo.cl

⁴ Precio obtenido de www.microsoft.com/es-cl/store/b/windows

⁵ Precio obtenido de www.microsoft.com/es-cl/microsoft-365

5.4.3 Resumen de Costos

A continuación, en las Tablas 4 y 5 se expresa el resumen de los costos e inversión del proyecto.

Costo de Operación	
Costo de Web Hosting (Firebase)	\$ 30.000
Costo Almacenamiento BDD (MongoAtlas)	\$10.000
Mantenición	\$50.000
Total Costo de Operación	\$90.000

Tabla 4: Costo de Operación

Inversión	
Licencia de Windows	\$ 170.500
Licencia Microsoft Office	\$50.000
Costo de desarrollador	\$ 2.100.000
Total Inversión	\$ 2.320.500

Tabla 5: Inversión.

5.4.4 Beneficios Tangibles

Para la medición realizada de los beneficios tangibles que se presentan a continuación, es importante destacar que las cifras mencionadas **fueron obtenidas en una reunión presencial con la jefatura de Gas Chillán.**

- **Automatización de procesos:** En el actual sistema de Gas Chillán, se debe analizar documentos manuscritos y planillas Excel para llevar a cabo sus procesos de negocio, por lo tanto, necesitan del tiempo extra de un empleado para analizar este tipo de artefactos. Esta tarea se realiza con bastante frecuencia, generalmente entre los/as recepcionistas, debido a que es un trabajo extenso y agobiante, considerado la gran cantidad de documentos existentes. En base a la información proporcionada por la jefatura de la empresa, se estima un ahorro de dinero de \$50.000 al mes, correspondiente al salario extra que recibe el empleado encargado de esta labor, **con un total de \$600.000 al año.**
- **Agilización de atención al cliente:** La jefatura de Gas Chillán, estima que aproximadamente se dejan de atender 2 clientes por semana debido a los retrasos que ocurren en recepción, por ende, los clientes deciden no realizar sus pedidos, o en el peor de los casos, realizándolo a la competencia (a otras distribuidoras). **Se considera un promedio de \$20.000 por cliente, con un total de \$40.000 por semana y \$2.200.000 anuales**

- Beneficio del Seguimiento de Clientes:** Al apoyar y facilitar la estrategia de fidelización de clientes con la funcionalidad de Seguimiento que el nuevo sistema implementará, las fugas de clientes detectadas (clientes que no compran por un período considerable de tiempo) son solucionadas inmediatamente y produce que los clientes vuelvan a comprar a la distribuidora. La jefatura de Gas Chillán, en conjunto con los administradores, especulan que aproximadamente 2 clientes dejan de comprar en la empresa durante el periodo de un mes. **Con una media de \$20.000 por cliente, el beneficio anual sería de \$480.000**

5.4.5 Flujo de Caja Neto

Para determinar la factibilidad económica, se utiliza el indicador Valor Actual Neto (VAN) que permitirá obtener valor de decisión frente al costo de esta.

Para el presente proyecto, el análisis considera los siguientes 2 factores:

- Se estiman 5 años de vida útil del proyecto
- Se considera una tasa de evaluación del 10%

A continuación, en la Tabla 6 se muestra el flujo de caja neto

	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
(+) Beneficios	0	\$3.280.000	\$3.280.000	\$3.280.000	\$3.280.000	\$3.280.000
(-) Costos						
Web Hosting (Servidor)	0	\$ 40.000	\$ 40.000	\$ 40.000	\$ 40.000	\$ 40.000
Mantenición	0	\$ 50.000	\$ 50.000	\$ 50.000	\$ 50.000	\$ 50.000
Licencias Software	\$ 220.500					
Desarrollador	\$2.100.000					
TOTAL	\$2.320.500	\$3.190.000	\$3.190.000	\$3.190.000	\$3.190.000	\$3.190.000

Tabla 6: Flujo de Caja Neto.

5.4.6 Cálculo del VAN

$$\text{VAN (10\%)} = - (2.320.000) + \frac{3.190.000}{(1+0.1)^1} + \frac{3.190.000}{(1+0.1)^2} + \frac{3.190.000}{(1+0.1)^3} + \frac{3.190.000}{(1+0.1)^4} + \frac{3.190.000}{(1+0.1)^5}$$

$$\text{VAN (10\%)} = 2.900.000 + 2.636.364 + 2.396.694 + 2.178.813 + 1.980.739 - 2.320.000$$

$$\text{VAN (10\%)} = \$ 9.772.610$$

Con una tasa de evaluación del 10%, el resultado del Valor Actual Neto es \$9.772.610, por lo tanto, la inversión en el primer año (Año 0) de \$2.320.000, produce una gran rentabilidad, por lo que el proyecto se considera totalmente factible económicamente.

5.5 Conclusión de la factibilidad

Al haber realizado el análisis de factibilidad del proyecto, se determina que el proyecto representa una buena oportunidad de inversión, por lo que se considera factible su desarrollo e implementación. Desde un punto de vista técnico, se cuenta con todos los recursos necesarios, potenciando a la vez el aspecto de factibilidad operativa, ya que el desarrollador cuenta con los conocimientos necesarios para el adecuado desarrollo del sistema.

Finalmente, desde el punto de vista económico, se observa un gran beneficio para la empresa debido a que soluciona los principales problemas que mantiene, estos son: la fuga de clientes, atención de cliente no realizada y la contratación de personal extra para el análisis de documentos. Esto permitirá que la empresa mantenga sus clientes satisfechos, cumpliendo con su compromiso organizacional de atención rápida y de calidad.

6 ANÁLISIS

Los diagramas de caso de uso y modelo entidad relación presentados aquí fueron confeccionados con la herramienta “yEd” [25], debido a sus características de usabilidad y agilidad para diseñar. Además, el modelo de proceso de negocio fue confeccionado con la herramienta Bizagi [26].

6.1 Proceso de Negocio del problema a resolver

Para comprender la actual situación de la empresa Gas Chillán, se elabora un modelo usando la notación BPMN, el cual representa de manera gráfica cómo se realizan los procesos de negocio dentro de la empresa, permitiendo tener una mejor abstracción del problema a resolver.

En la Figura 4 se muestra el modelo gráfico con notación BPMN, y se representan 2 participantes; la empresa distribuidora (compuesto por Administración, Reparto/Chofer y Recepción) y el Cliente.

Este proceso comienza cuando el cliente solicita un pedido al área de recepción de la empresa, el cliente se contacta la mayor parte del tiempo por vía telefónica, otras vías de comunicación son por aplicaciones de mensajería instantánea como WhatsApp o Facebook Messenger.

Posteriormente, recepción registra el pedido y asigna su reparto a un repartidor/chofer quien debe entregar el pedido al cliente, confirmándolo cuando esta paga el pedido. El chofer termina su turno y entrega rendiciones a administración. Administración recibe dichas rendiciones y debe cuadrar inventario a medida que cada repartidor las realiza.

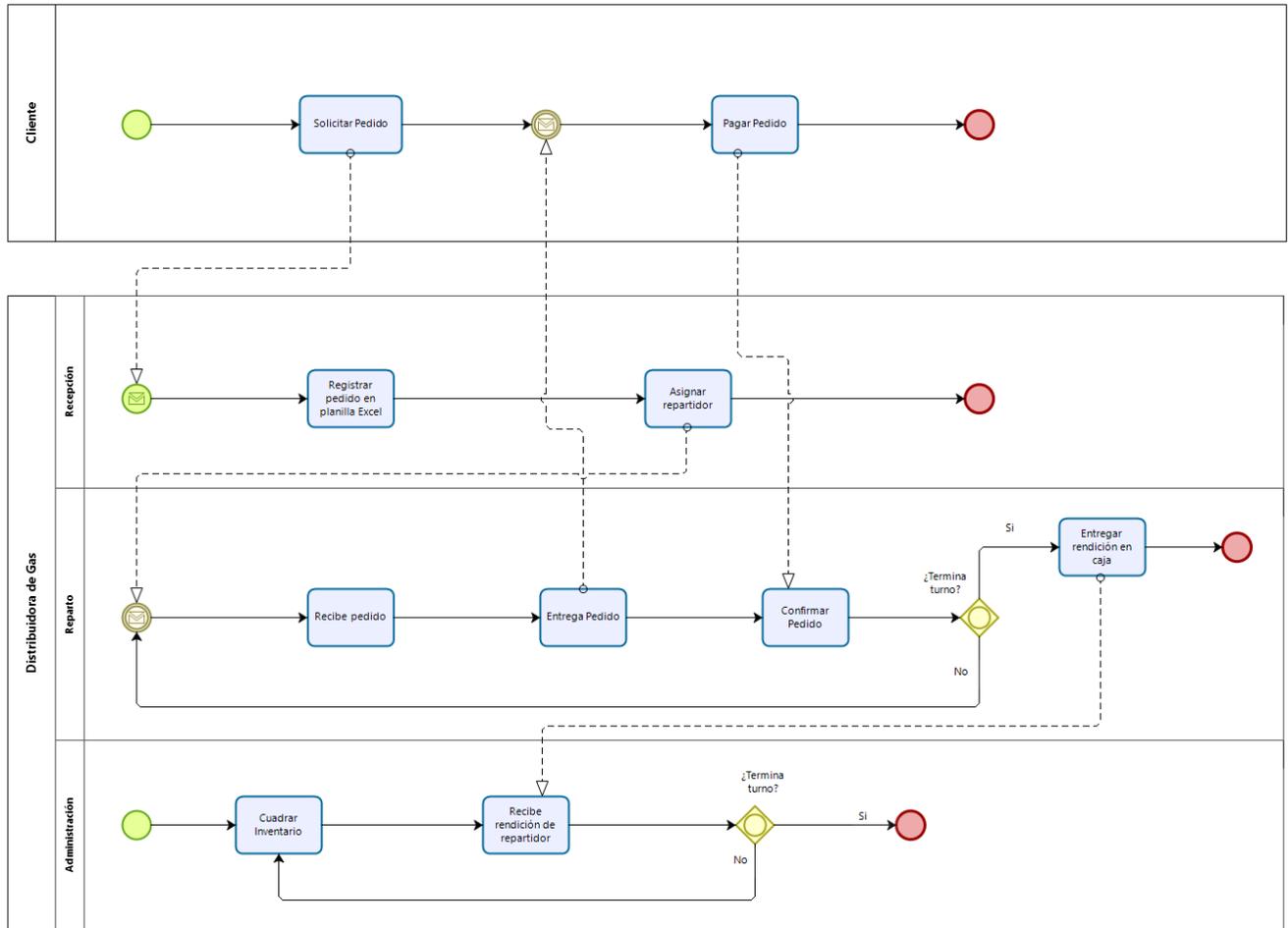


Figura 4: Modelo de Proceso de Negocio (BPMN).

6.2 Diagrama de casos de uso

A continuación, se presenta el diagrama de caso de uso del sistema para la empresa Gas Chillán, el cual tiene como finalidad representar las principales funcionalidades con las que van a interactuar los usuarios (actores).

En la Figura 5 se muestra el diagrama de casos de uso.

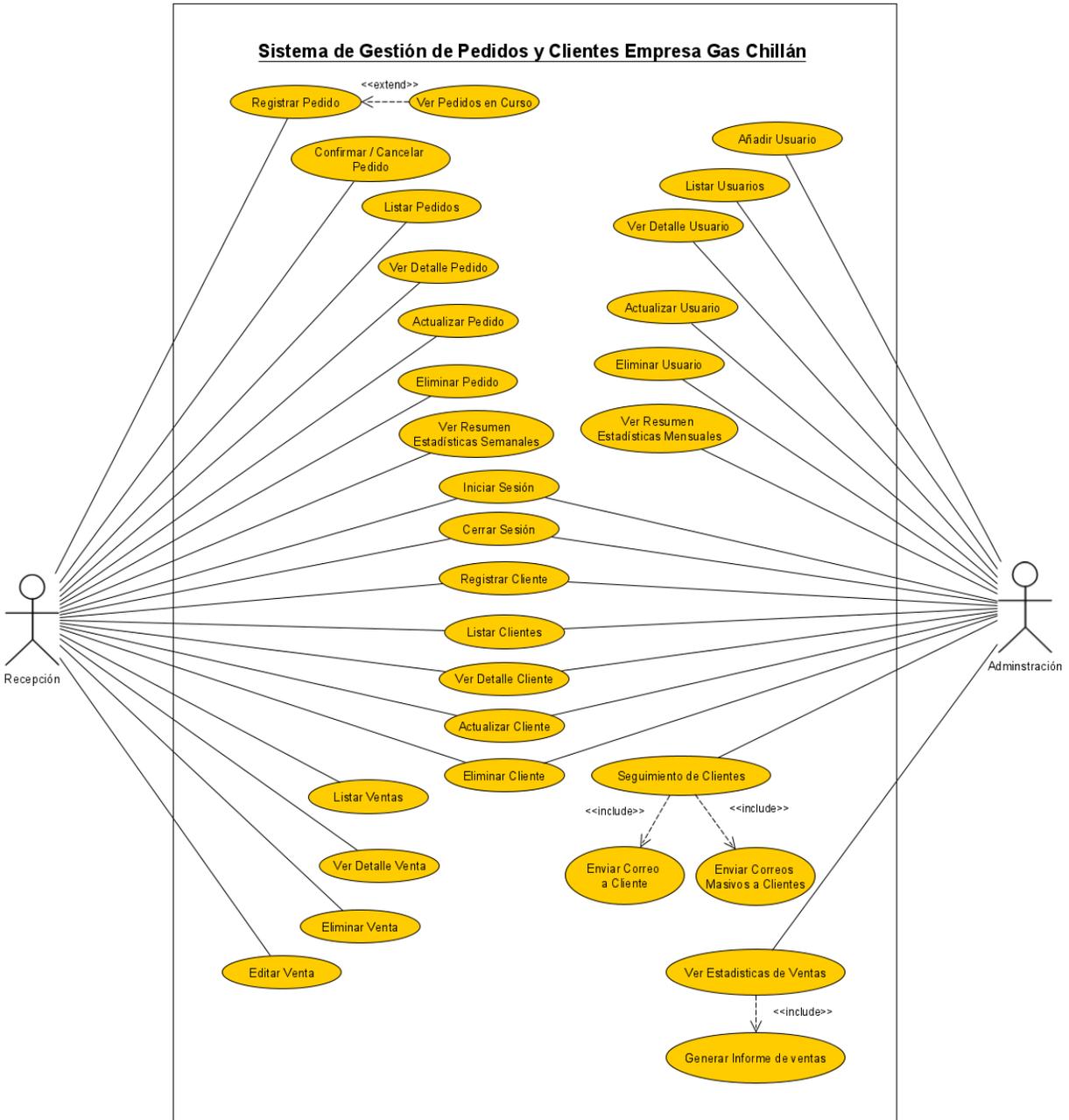


Figura 5: Diagrama de Casos de Uso.

6.2.1 Actores

Los Actores especifican el rol de una entidad organizacional dentro del sistema y son los que interactúan con las diferentes funcionalidades (casos de uso) disponibles. A continuación, se detallan los dos actores involucrados en este sistema, administración y recepción.

Es importante destacar que los privilegios “Limitados” hacen referencia a que no existe un actor que represente un súper usuario. Esto debido a que cada actor se enfoca solamente en las funcionalidades propias de su área.

A continuación, en la Tabla 7 se detallan los actores del sistema.

Actor	Funciones	Conocimientos Técnicos	Privilegios	Funcionalidades Permitidas
Administración	<ul style="list-style-type: none"> Gestionar tareas organizacionales Toma de decisiones de la empresa 	Computación Nivel Usuario	Limitado	<ul style="list-style-type: none"> Iniciar Sesión Cerrar Sesión Ver Resumen de Estadísticas Mensuales. Seguimiento de Clientes Enviar Correo a Cliente Enviar Correos Masivos a Clientes Ver Estadísticas de Ventas Generar Informe de Ventas Añadir Usuario Listar Usuarios Ver Detalle Usuario Actualizar Usuario Eliminar Usuario Registrar Cliente Listar Cliente Ver Detalle Cliente Actualizar Cliente Eliminar Cliente
Recepción	<ul style="list-style-type: none"> Atender pedidos de gas. Visualizar las ventas 	Computación Nivel Usuario	Limitado	<ul style="list-style-type: none"> Iniciar Sesión Cerrar Sesión Registrar Pedido Ver Pedidos en Curso Confirmar / Cancelar Pedido Listar Pedidos Ver Detalle Pedido Actualizar Pedido Actualizar Pedido Eliminar Pedido Ver Resumen de Estadísticas Semanales Listar Ventas Ver Detalle Venta Eliminar Venta

				<ul style="list-style-type: none"> • Editar Venta • Registrar Cliente • Listar Cliente • Ver Detalle Cliente • Actualizar Cliente • Eliminar Cliente
--	--	--	--	--

Tabla 7: Actores del Sistema.

6.2.2 Especificación de los Casos de Uso

Para cada especificación de caso de uso que se detalla a continuación se consideran los principales atributos de estos, tales como su correspondiente identificación (ID), descripción, actores del sistema que interactúan con el caso de uso, precondiciones, flujo principal, postcondiciones, flujos alternativos (si corresponde) y destacar la fila “Requisito” que indica el ID del requisito funcional (4.4.1) asociado al caso de uso.

Además, se señalan las interfaces de entrada (DE) y de salida externas (IS) descritas en la sección 4.4.2 y 4.4.3 en los casos de uso que correspondan.

6.2.2.1 Caso de Uso: Iniciar Sesión

En la Tabla 8 se especifica el caso de uso “Iniciar Sesión” en donde los usuarios realizan el proceso de autenticación para acceder al sistema.

Caso de Uso	Iniciar Sesión	
ID	CU_01	
Requisito ID	R_01	
Descripción	Permite el acceso del usuario al sistema mediante credenciales únicas y da inicio al proceso de autenticación.	
Actores del Sistema	Recepción, Administración	
Precondiciones	Que el usuario no esté autenticado aún.	
Flujos Principales	Actor	Sistema
		1. El sistema despliega la interfaz de inicio de sesión, cargando las entradas para ingresar las credenciales correspondientes.
	2. El usuario ingresa sus credenciales. [DE_2]	
		3. El sistema realiza el proceso de autenticación y verifica los datos ingresados e identifica al usuario.
Flujo Alternativo	3a. Los datos ingresados no corresponden a ningún usuario válido registrado en el sistema, por lo tanto, vuelve a iniciar sesión (paso 1)	
Postcondiciones	El usuario está autenticado con los permisos y privilegios correspondientes al rol del usuario.	

Tabla 8: Caso de Uso Iniciar Sesión

6.2.2.2 Caso de Uso: Cerrar Sesión

En la Tabla 9 se especifica el caso de uso “Cerrar Sesión” en donde los usuarios se desconectan del sistema.

Caso de Uso		Cerrar Sesión	
ID	CU_02		
Requisito ID	R_01		
Descripción	Permite al usuario desconectarse del sistema		
Actores del Sistema	Recepción, Administración.		
Precondiciones	Que el usuario esté autenticado en el sistema.		
Flujos Principales	Actor	Sistema	
		1. El sistema mantiene durante toda navegación la opción para cerrar la sesión.	
	2. El usuario escoge salir del sistema mediante esta opción		
		3. El sistema redirecciona a la interfaz de inicio de sesión para proteger las vistas que requieren autenticación.	
Postcondiciones	El usuario ha salido del sistema		

Tabla 9: Caso de Uso Cerrar Sesión

6.2.2.3 Caso de Uso: Ver Resumen Estadísticas Semanales

En la Tabla 10 se especifica el caso de uso “Ver Resumen Estadísticas Semanales” en donde el usuario puede visualizar un conjunto de útiles estadísticas resumidas.

Caso de Uso		Ver Resumen Estadísticas Semanales	
ID	CU_03		
Requisito ID	R_02		
Descripción	Permite al usuario visualizar estadísticas básicas de pedidos, ventas y clientes al momento de iniciar sesión (Home Page / Página de Inicio)		
Actores del Sistema	Recepción		
Precondiciones	Usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. Selecciona la sección “Inicio” del sistema		
		2. El sistema muestra mediante una interfaz intuitiva un conjunto de estadísticas que son relevantes para el usuario [IS_1]	
Postcondiciones	Ninguna		

Tabla 10: Caso de Uso Resumen de Estadísticas Semanales

6.2.2.4 Caso de Uso: Ver Resumen Estadísticas Mensuales

En la Tabla 11 se especifica el caso de uso “Ver Resumen Estadísticas Mensuales” en donde el usuario puede visualizar un conjunto de útiles estadísticas resumidas.

Caso de Uso		Ver Resumen Estadísticas Mensuales	
ID	CU_04		
Requisito ID	R_03		
Descripción	Permite al usuario visualizar estadísticas básicas de pedidos, ventas y clientes al momento de iniciar sesión (Home Page / Página de Inicio)		
Actores del Sistema	Administración		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. El usuario selecciona la sección “Inicio” del sistema	2. El sistema muestra mediante una interfaz intuitiva un conjunto de estadísticas que son relevantes para el usuario [IS_2]	
Postcondiciones	Ninguna		

Tabla 11: Caso de Uso Resumen de Estadísticas Mensuales

6.2.2.5 Caso de Uso: Registrar Pedido

En la Tabla 12 se especifica el caso de uso “Registrar Pedido” en donde los usuarios pueden crear un nuevo pedido en el sistema.

Caso de Uso		Registrar Pedido	
ID	CU_05		
Requisito ID	R_04		
Descripción	Permite al usuario registrar pedidos en el sistema		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. El usuario elige la opción para registrar un nuevo pedido.	2. El sistema despliega la interfaz con las entradas para los datos requeridos.	
	3. El usuario ingresa las entradas correspondientes [DE_2]	4. El sistema verifica cada dato de la entrada	
	5. El usuario envía el formulario para su almacenamiento	6. El sistema guarda el nuevo pedido.	
	Flujo alternativo	4.a El sistema determina que los datos son incorrectos y no permite registrar el pedido. Vuelve al paso 2.	
	Postcondiciones	Pedido registrado en el sistema y base datos actualizada.	

Tabla 12: Caso de Uso Registrar Pedido

6.2.2.6 Caso de Uso: Listar Pedidos

En la Tabla 13 se especifica el caso de uso “Listar Pedidos” en donde los usuarios pueden ver una lista de todos los pedidos realizados y registrados.

Caso de Uso		Listar Pedidos	
ID	CU_06		
Requisito ID	R_05		
Descripción	Permite al usuario visualizar los pedidos registrados en el sistema.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. El usuario selecciona sección de “Pedidos”	2. El sistema despliega la interfaz que muestra todos los pedidos registrados con paginación y permite las acciones correspondientes a cada pedido mostrado, incluyendo filtros de búsqueda	
Postcondiciones	Ninguna		

Tabla 13 Caso de Uso Listar Pedidos

6.2.2.7 Caso de Uso: Ver Detalle Pedido

En la Tabla 14 se especifica el caso de uso “Ver Detalle Pedido” en donde los usuarios pueden ver todos los datos relacionados a un pedido registrado en específico.

Caso de Uso		Ver Detalle de Pedido	
ID	CU_07		
Requisito ID	R_06		
Descripción	Permite al usuario visualizar todos los datos de un pedido registrado en el sistema.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado. Que se hayan listado los pedidos previamente o que se haya buscado el pedido por algún criterio		
Flujos Principales	Actor	Sistema	
	1. El usuario selecciona un pedido de la lista desplegada	2. El sistema despliega la interfaz con todos los datos del pedido seleccionado y muestra las acciones que se pueden realizar (Editar, Eliminar) para cada uno de ellos.	
Postcondiciones	Ninguna		

Tabla 14: Caso de Uso Ver Detalle de Pedido

6.2.2.8 Caso de Uso: Actualizar Pedido

En la Tabla 15 se especifica el caso de uso “Actualizar Pedido” en donde los usuarios pueden realizar cambios a los datos de los pedidos registrados.

Caso de Uso		Actualizar Pedido	
ID	CU_08		
Requisito ID	R_07		
Descripción	Permite al usuario realizar cambios en los datos de un pedido registrado.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado. Haber seleccionado un pedido previamente		
Flujos Principales	Actor	Sistema	
	1. El usuario modifica los campos que desee.		
		2. El sistema verifica la validez de los datos	
	3. El usuario envía el formulario con los nuevos datos		
	4. El sistema guarda los nuevos datos y muestra un mensaje que informe al usuario si los cambios se realizaron o no.		
Flujo Alternativo	2.2 El sistema determina que los datos son incorrectos y no permite actualizar el pedido. Vuelve al paso 1		
Postcondiciones	Pedido actualizado en el sistema		

Tabla 15: Caso de Uso Actualizar Pedido

6.2.2.9 Caso de Uso: Eliminar Pedido

En la Tabla 16 se especifica el caso de uso “Eliminar Pedido” en donde los usuarios pueden remover un pedido del sistema.

Caso de Uso		Eliminar Pedido	
ID	CU_09		
Requisito ID	R_08		
Descripción	El usuario podrá eliminar un pedido y toda su información relacionada.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado Que el usuario haya seleccionado el pedido o que haga clic en la acción de “eliminar” en la lista de pedidos.		
Flujos Principales	Actor	Sistema	
	1. El usuario realiza la acción de eliminar.		
		2. El sistema emite y espera la confirmación del usuario.	

	3. El usuario confirma la eliminación del pedido	
		4. El sistema muestra un mensaje que informe al usuario si la eliminación se realizó o no.
Postcondiciones	Pedido removido del sistema.	

Tabla 16: Caso de Uso Eliminar Pedido

6.2.2.10 Caso de Uso: Ver Pedidos en Curso

En la Tabla 17 se especifica el caso de uso “Ver Pedidos en Curso” en donde los usuarios pueden ver los pedidos que no han sido confirmados aún.

Caso de Uso	Ver Pedidos en Curso	
ID	CU_10	
Requisito ID	R_09	
Descripción	Permite al usuario ver los pedidos que han sido registrados y decidir la confirmación o cancelación de estos (CU_7 y CU_8)	
Actores del Sistema	Recepción	
Precondiciones	El usuario debe estar autenticado	
Flujos Principales	Actor	Sistema
		1. El sistema muestra durante toda la sesión del usuario una pequeña interfaz donde se visualizan los pedidos que se han hecho en el día y no han sido confirmados.
	2. El usuario ve el pedido directamente en la pequeña interfaz. 2b. El usuario ve todos los pedidos en curso seleccionando la opción “Ver Todo”, desplegando una interfaz que muestre todos los detalles de los pedidos ordenados por fecha y hora de manera ascendente 3. Continúa en CU_07 y CU_08	
Postcondiciones	Ninguna	

Tabla 17: Caso de Uso Ver Pedidos en Curso

6.2.2.11 Caso de Uso: Confirmar Pedido

En la Tabla 18 se especifica el caso de uso “Confirmar Pedido” en donde los usuarios confirman un pedido para generar una nueva venta en el sistema.

Caso de Uso		Confirmar Pedido	
ID	CU_11		
Requisito ID	R_10		
Descripción	Permite al usuario confirmar un pedido en el sistema		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	El actor viene del flujo principal del CU_6		
	1. El usuario selecciona el botón que “Confirmar” del pedido.		
		2. El sistema muestra un mensaje que el pedido ha sido confirmado de manera exitosa.	
Postcondiciones	Pedido actualizado en el sistema. Venta generada y registrada en el sistema.		

Tabla 18: Caso de Uso Confirmar Pedido

6.2.2.12 Caso de Uso: Cancelar Pedido

En la Tabla 19 se especifica el caso de uso “Cancelar Pedido” en donde los usuarios cancelan un pedido para eliminarlo del sistema.

Caso de Uso		Cancelar Pedido	
ID	CU_12		
Requisito ID	R_10		
Descripción	Permite al usuario cancelar un pedido en el sistema		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	El actor viene del flujo principal del CU_6		
	1. El usuario selecciona el botón “Cancelar” del pedido.		
		2. El sistema requiere pregunta al usuario la confirmación. 2b. El sistema muestra un mensaje que el pedido ha sido confirmado de manera exitosa.	
Postcondiciones	Pedido eliminado del sistema		

Tabla 19: Caso de Uso Cancelar Pedido

6.2.2.13 Caso de Uso: Listar Ventas

En la Tabla 20 se especifica el caso de uso “Listar Ventas” en donde los usuarios pueden ver una lista de todas las ventas registradas.

Caso de Uso		Listar Ventas
ID	CU_13	
Requisito ID	R_12	
Descripción	Permite al usuario visualizar las ventas registradas en el sistema.	
Actores del Sistema	Recepción	
Precondiciones	El usuario debe estar autenticado	
Flujos Principales	Actor	Sistema
	1. El usuario selecciona sección de “Ventas”	2. El sistema despliega la interfaz que muestra todos las ventas registradas con paginación y permite las acciones correspondientes sobre venta mostrada, incluyendo filtros de búsqueda.
Postcondiciones	Ninguna	

Tabla 20: Caso de Uso Listar Ventas

6.2.2.14 Caso de Uso: Ver Detalle Venta

En la Tabla 21 se especifica el caso de uso “Ver Detalle Venta” en donde los usuarios pueden ver todos los datos relacionados a una venta registrada en específico.

Caso de Uso		Ver Detalle Venta
ID	CU_14	
Requisito ID	R_13	
Descripción	Permite al usuario visualizar todos los datos de una venta registrada en el sistema.	
Actores del Sistema	Recepción	
Precondiciones	El usuario debe estar autenticado. Que se hayan listado las ventas previamente o que se haya buscado la venta por algún criterio	
Flujos Principales	Actor	Sistema
	1. El usuario selecciona una venta de la lista desplegada	2. El sistema despliega la interfaz con todos los datos de la venta seleccionada y muestra las acciones que se pueden realizar (Editar, Eliminar) para cada uno de ellos.
Postcondiciones	Ninguna	

Tabla 21: Caso de Uso Ver Detalle Venta

6.2.2.15 Caso de Uso: Editar Venta

En la Tabla 22 se especifica el caso de uso “Editar Venta” en donde los usuarios pueden realizar cambios a los datos de las ventas registradas.

Caso de Uso		Editar Venta	
ID	CU_15		
Requisito ID	R_14		
Descripción	Permite al usuario realizar cambios en los datos de una venta registrada.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado. Haber seleccionado una venta previamente		
Flujos Principales	Actor	Sistema	
	1. El usuario modifica los campos que desee.		
		2. El sistema verifica la validez de los datos.	
	3. El usuario envía el formulario con los nuevos datos		
	4. El sistema guarda los nuevos datos y mostrará un mensaje que informe al usuario si los cambios se realizaron o no.		
Flujo Alternativo	2.a El sistema determina que los datos son incorrectos y no permite actualizar la venta. Vuelve al Paso 1.		
Postcondiciones	La venta ha sido editada		

Tabla 22: Caso de Uso Editar Venta

6.2.2.16 Caso de Uso: Eliminar Venta

En la Tabla 23 se especifica el caso de uso “Eliminar Venta” en donde los usuarios pueden remover una venta del sistema.

Caso de Uso		Eliminar Pedido	
ID	CU_16		
Requisito ID	R_15		
Descripción	El usuario podrá eliminar una venta y toda su información relacionada.		
Actores del Sistema	Recepción		
Precondiciones	El usuario debe estar autenticado Que el usuario haya seleccionado el pedido o que haga click en la acción de “eliminar” en la lista de ventas.		
Flujos Principales	Actor	Sistema	
	1. El usuario realiza la acción de eliminar.		
		2. El sistema emite y espera la confirmación del usuario.	
	3. El usuario confirma la eliminación de la venta		
		4. El sistema mostrará un mensaje que informe al usuario si la eliminación se realizó o no.	
Postcondiciones	La venta es removida del sistema.		

Tabla 23 Caso de Uso Eliminar Pedido

6.2.2.17 Caso de Uso: Registrar Cliente

En la Tabla 24 se especifica el caso de uso “Registrar Cliente” en donde los usuarios pueden crear un nuevo cliente de manera explícita en el sistema. La forma implícita de este caso de uso es cuando se registra un pedido, en donde se piden los datos del cliente y una vez registrado el pedido, también se registra el cliente.

Caso de Uso		Registrar Cliente	
ID	CU_17		
Requisito ID	R_16		
Descripción	Permite al usuario registrar un cliente en el sistema		
Actores del Sistema	Recepción, Administración		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. El usuario elige la opción para registrar un nuevo pedido.		
		2. El sistema despliega la interfaz con las entradas para los datos requeridos.	
	3. El usuario ingresa las entradas correspondientes [DE_3]		

		4. El sistema verifica cada dato de las entradas
	5. El usuario envía el formulario para el almacenamiento	
		6. El sistema guarda el nuevo cliente.
Flujo alternativo	4.a. El sistema determina que los datos son incorrectos y no permite registrar el cliente. Vuelve al paso 3	
Postcondiciones	Cliente registrado en el sistema	

Tabla 24: Caso de Uso Registrar Cliente

6.2.2.18 Caso de Uso: Listar Clientes

En la Tabla 25 se especifica el caso de uso “Listar Clientes” en donde los usuarios pueden ver una lista de todos los clientes registrados.

Caso de Uso		Listar Clientes	
ID	CU_18		
Requisito ID	R_17		
Descripción	Permite al usuario visualizar los clientes registrados en el sistema.		
Actores del Sistema	Recepción, Administración		
Precondiciones	El usuario debe estar autenticado		
Flujos Principales	Actor	Sistema	
	1. El usuario selecciona sección de “Clientes”	2. El sistema despliega la interfaz que muestra todos los clientes registrados con paginación y permite las acciones correspondientes sobre cada cliente mostrado, incluyendo filtros de búsqueda	
Postcondiciones	Ninguna		

Tabla 25: Caso de Uso Listar Clientes

6.2.2.19 Caso de Uso: Ver Detalle Cliente

En la Tabla 26 se especifica el caso de uso “Ver Detalle Cliente” en donde los usuarios pueden ver todos los datos relacionados a un cliente registrado en específico.

Caso de Uso		Ver Detalle de Cliente
ID	CU_19	
Requisito ID	R_18	
Descripción	Permite al usuario visualizar todos los datos de un cliente registrado en el sistema.	
Actores del Sistema	Recepción, Administración	
Precondiciones	El usuario debe estar autenticado. Que se hayan listado los clientes previamente o que se haya buscado el cliente por algún criterio	
Flujos Principales	Actor	Sistema
	1. El usuario selecciona un cliente de la lista desplegada	2. El sistema despliega la interfaz con todos los datos del cliente seleccionado y muestra las acciones que se pueden realizar (Editar, Eliminar) para cada uno de ellos.
Postcondiciones	Ninguna	

Tabla 26: Caso de Uso Ver Detalle Cliente

6.2.2.20 Caso de Uso: Actualizar Cliente

En la Tabla 27 se especifica el caso de uso “Actualizar Cliente” en donde los usuarios pueden realizar cambios a los datos de los clientes registrados.

Caso de Uso		Actualizar Cliente
ID	CU_20	
Requisito ID	R_19	
Descripción	Permite al usuario realizar cambios en los datos de un cliente registrado.	
Actores del Sistema	Recepción, Administración	
Precondiciones	El usuario debe estar autenticado. Haber seleccionado un cliente previamente	
Flujos Principales	Actor	Sistema
	1. El usuario modifica los campos que desee.	2. El sistema verifica la validez de los datos
	3. El usuario envía el formulario con los nuevos datos	4. El sistema guarda los nuevos datos y muestra un mensaje que informe al usuario si los cambios se realizaron o no.
Flujo Alternativo	2.a. El sistema determina que los datos son incorrectos y no permite actualizar al cliente. Vuelve al paso 1.	

Postcondiciones	Cliente modificado en el sistema
------------------------	----------------------------------

Tabla 27: Caso de Uso Actualizar Cliente

6.2.2.21 Caso de Uso: Eliminar Cliente

En la Tabla 28 se especifica el caso de uso “Eliminar Cliente” en donde los usuarios pueden remover un cliente del sistema.

Caso de Uso		Eliminar Cliente
ID	CU_21	
Requisito ID	R_20	
Descripción	El usuario podrá eliminar un cliente y toda su información relacionada.	
Actores del Sistema	Recepción, Administración	
Precondiciones	El usuario debe estar autenticado Que el usuario haya seleccionado el cliente o que haga click en la acción de “eliminar” en la lista de clientes.	
Flujos Principales	Actor	Sistema
	1. El usuario realiza la acción de eliminar.	
		2. El sistema emite y espera la confirmación del usuario.
	3. El usuario confirma la eliminación del cliente	
		4. El sistema muestra un mensaje que informe al usuario si la eliminación se realizó o no.
Postcondiciones	El cliente es removido del sistema.	

Tabla 28: Caso de Uso Eliminar Cliente

6.2.2.22 Caso de Uso: Registrar Usuario

En la Tabla 29 se especifica el caso de uso “Registrar Usuario” en donde los usuarios pueden crear un nuevo usuario en el sistema.

Caso de Uso		Registrar Usuario	
ID	CU_22		
Requisito ID	R_21		
Descripción	Permite al usuario registrar un usuario en el sistema.		
Actores del Sistema	Administración		
Precondiciones	El usuario debe estar autenticado.		
Flujos Principales	Actor	Sistema	
	1. El usuario elige la opción para registrar un nuevo usuario.		
		2. El sistema despliega la interfaz con las entradas para los datos requeridos.	
	3. El usuario ingresa las entradas correspondientes [DE_4]		
		4. El sistema verifica cada dato de las entradas	
	5. El usuario envía el formulario para el almacenamiento		
	6. El sistema guarda el nuevo usuario.		
Flujo alternativo	4.a. El sistema determina que los datos son incorrectos y no permite registrar al Usuario. Vuelve al Paso 3.		
Postcondiciones	Usuario registrado en el sistema		

Tabla 29: Caso de Uso Registrar Usuario

6.2.2.23 Caso de Uso: Listar Usuarios

En la Tabla 30 se especifica el caso de uso “Listar Usuarios” el administrador puede ver una lista de todos los usuarios registrados.

Caso de Uso		Listar Usuarios
ID	CU_23	
Requisito ID	R_22	
Descripción	Permite al usuario visualizar los usuarios registrados en el sistema.	
Actores del Sistema	Administración	
Precondiciones	El usuario debe estar autenticado	
Flujos Principales	Actor	Sistema
	1. El usuario selecciona sección de “Usuarios”.	2. El sistema despliega la interfaz que muestra todos los Usuarios registrados con paginación y permite las acciones correspondientes a cada Usuario mostrado, incluyendo filtros de búsqueda.
Postcondiciones	Ninguna.	

Tabla 30: Listar Usuarios

6.2.2.24 Caso de Uso: Ver Detalle Usuario

En la Tabla 31 se especifica el caso de uso “Ver Detalle Usuario” en donde los usuarios pueden ver todos los datos relacionados a un usuario registrado en específico.

Caso de Uso		Ver Detalle de Usuario
ID	CU_24	
Requisito ID	R_23	
Descripción	Permite al usuario visualizar todos los datos de un usuario registrado en el sistema.	
Actores del Sistema	Administración	
Precondiciones	El usuario debe estar autenticado. Que se hayan listado los usuarios previamente o que se haya buscado el Usuario por algún criterio	
Flujos Principales	Actor	Sistema
	1. El usuario selecciona un Usuario de la lista desplegada	2. El sistema despliega la interfaz con todos los datos del Usuario seleccionado y muestra las acciones que se pueden realizar (Editar, Eliminar) para cada uno de ellos.
Postcondiciones	Ninguna	

Tabla 31: Caso de Uso Ver Detalle Usuario

6.2.2.25 Caso de Uso: Actualizar Usuario

En la Tabla 32 se especifica el caso de uso “Actualizar Usuario” en donde los usuarios pueden realizar cambios a los datos de los Usuarios registrados.

Caso de Uso		Actualizar Usuario	
ID	CU_25		
Requisito ID	R_24		
Descripción	Permite al usuario realizar cambios en los datos de un usuario registrado.		
Actores del Sistema	Administración		
Precondiciones	El usuario debe estar autenticado. Haber seleccionado un Usuario previamente		
Flujos Principales	Actor	Sistema	
	1. El usuario modifica los campos que desee.		
		2. El sistema verifica la validez de los datos	
	3. El usuario envía el formulario con los nuevos datos		
	4. El sistema guarda los nuevos datos y muestra un mensaje que informe al usuario si los cambios se realizaron o no.		
Flujo Alternativo	2.a. El sistema determina que los datos son incorrectos y no permite actualizar al Usuario. Vuelve al Paso 1.		
Postcondiciones	Usuario modificado en el sistema		

Tabla 32: Caso de Uso Actualizar Usuario

6.2.2.26 Caso de Uso: Eliminar Usuario

En la Tabla 33 se especifica el caso de uso “Eliminar Usuario” en donde los usuarios pueden remover un Usuario del sistema.

Caso de Uso		Eliminar Usuario	
ID	CU_26		
Requisito ID	R_25		
Descripción	El usuario podrá eliminar virtualmente un Usuario (deshabilitar)		
Actores del Sistema	Administración		
Precondiciones	El usuario debe estar autenticado Que el usuario haya seleccionado al Usuario o que haga click en la acción de “eliminar” en la lista de Usuarios.		
Flujos Principales	Actor	Sistema	
	1. El usuario realiza la acción de eliminar.		
		2. El sistema emite y espera la confirmación del usuario.	
	3. El usuario confirma la eliminación del Usuario.		
		4. El sistema muestra un mensaje que informe al usuario si la eliminación se realizó o no.	
Postcondiciones	El usuario es deshabilitado del sistema.		

Tabla 33: Caso de Uso Eliminar Usuario

6.2.2.27 Caso de Uso: Seguimiento de Clientes.

En la Tabla 34 se especifica el caso de uso “Seguimiento de Clientes” en donde el usuario puede visualizar a todos los clientes registrados en el sistema mediante filtros o criterios de búsqueda y se muestre información relevante respecto a las compras.

Caso de Uso		Seguimiento de Clientes	
ID	CU_27		
Requisito ID	R_26		
Descripción	<p>Permite visualizar información de cada uno de los clientes registrados en base a filtros y criterios de búsqueda con el objeto de brindar indicadores de predicción de compra. Para cada cliente se muestran tres columnas adicionales:</p> <ul style="list-style-type: none"> - Última Compra: indica la última compra que realizó el cliente. - Periodicidad de compra: indica cada cuánto tiempo el cliente realiza un pedido de gas. - Estado de Predicción: varía en dos valores, estos son, “Próximo a comprar”, “Compra atrasada”. - Acción: permite enviar un correo a los clientes que presente retrasos en su periodo habitual de compra. 		
Actores del Sistema	Administración.		
Precondiciones	El usuario debe estar autenticado.		
Flujo Principal	Actor	Sistema	
	1. El usuario selecciona la sección de Seguimiento de Clientes.	<p>2. El sistema realiza el cálculo de predicción en base a todas las compras realizadas por cada uno de los clientes registrados.</p> <p>3. El sistema muestra un indicador de carga para informar al usuario que se está realizando el cálculo.</p> <p>4. El sistema despliega la interfaz de todos los clientes con sus datos y la información de periodicidad de compra calculada, estado de predicción calculado y las funciones para enviar correos a clientes descritas en CU_28 y CU_29.</p>	

Flujo Alternativo	<p>4a. Si el cálculo no se realizó correctamente debido a factores técnicos, el sistema muestra un mensaje descriptivo de dicho error con el botón “Reintentar”.</p> <p>4.b El usuario intenta nuevamente realizar el cálculo con el botón “Reintentar”. (Hecho esto, se repite desde el Paso 2)</p>
--------------------------	--

Tabla 34: Seguimiento de Clientes

6.2.2.28 Caso de Uso: Enviar Correo a Cliente

En la Tabla 35 se especifica el caso de uso “Enviar Correo a Cliente” en donde el usuario puede enviar un correo automático cuando un cliente tiene una predicción de compra en estado “atrasada”.

Caso de Uso	Enviar Correo a Cliente	
ID	CU_28	
Requisito ID	R_27	
Descripción	El usuario puede enviar un correo automático al cliente que tenga una predicción de compra en estado “atrasada”	
Actores del Sistema	Administración.	
Precondiciones	El usuario debe estar autenticado. El cliente seleccionado debe tener correo electrónico asociado.	
Flujo Principal	Actor	Sistema
	Este caso de uso está incluido en el flujo principal del CU_27.	
	1. El usuario selecciona el botón “Enviar Correo” en un cliente desplegado en la lista de clientes.	
		2. El sistema emite un mensaje de confirmación para enviar el correo.
	3. El usuario confirma la acción	
		4. El sistema envía un correo al cliente seleccionado y actualiza el dato del cliente que indica que el correo fue enviado, además de cambiar el botón “Enviar Correo” por un texto “Correo enviado”
		5. El sistema muestra un mensaje si la acción se realizó o no.
Flujo Alternativo	Si el envío no se realizó correctamente debido a factores técnicos, el sistema muestra un mensaje descriptivo de dicho error y el usuario debe intentar nuevamente realizar el envío con el botón “Enviar Correo” (Paso 1)	
Postcondiciones	Cliente actualizado en el sistema con su atributo “correo enviado”	

Tabla 35: Enviar Correo a Cliente

6.2.2.29 Caso de Uso: Enviar Correos Masivos a Clientes.

En la Tabla 36 se especifica el caso de uso “Enviar Correos Masivos a Clientes” en donde el usuario puede enviar a cada cliente que esté con un estado de “compra atrasada” de manera masiva.

Caso de Uso		Enviar Correos Masivos a Clientes	
ID	CU_29		
Requisito ID	R_28		
Descripción	Permite al usuario enviar correos masivos a los clientes que no estén comprando en su periodo habitual.		
Actores del Sistema	Administración.		
Precondiciones	El usuario debe estar autenticado.		
Flujos Principales	Actor	Sistema	
	Este caso de uso está incluido en el flujo principal del CU_27.		
	1. El usuario selecciona el botón “Enviar Correo Masivo”.		
		2. El sistema emite un mensaje de confirmación al usuario, solicitando que ingrese sus credenciales nuevamente.	
	3. El usuario ingresa sus credenciales [DE_1] y confirma la acción		
	4. El sistema envía un correo a todos los clientes y actualiza el dato de cada cliente que indica que el correo fue enviado, además de cambiar el botón “Enviar Correo” por un texto “Correo enviado”		
	5. El sistema muestra un mensaje si la acción se realizó o no.		
Postcondiciones	Clientes actualizados con el dato de “correo enviado” en la base de datos del sistema.		
Flujo Alternativo	5a. Si los datos ingresados no corresponden a ningún usuario valido registrado en el sistema o con permisos insuficientes, vuelve a intentar ingresarlas (Paso 2).		
	5b. Si el envío no se realizó correctamente debido a factores técnicos, el sistema muestra un mensaje descriptivo de dicho error		

	y el usuario debe intentar nuevamente realizar el envío con el botón “Enviar Correo Masivo” (Paso 1)
--	--

Tabla 36: Enviar Correos Masivos a Clientes

6.2.2.30 Ver Estadísticas de Ventas.

En la Tabla 37 se especifica el caso de uso “Ver Estadísticas de Ventas” en donde el usuario puede visualizar gráficas y datos relevantes de las ventas realizadas durante un periodo de tiempo determinado.

Caso de Uso		Ver Estadísticas de Ventas	
ID	CU_30		
Requisito ID	R_29		
Descripción	Permite visualizar información relevante y sintetizada de las ventas realizadas durante un periodo de tiempo que el usuario desee (por día, semana, mes, semestre o año) mediante gráficas y tablas interactivas.		
Actores del Sistema	Administración.		
Precondiciones	Seleccionar sección de Estadísticas de Ventas		
Flujos Principales	Actor	Sistema	
	1. El usuario selecciona la sección “Estadísticas de Ventas”		
		2. El sistema realiza el cálculo de todas las estadísticas (descritas en IS_04)	
		3. El sistema despliega la interfaz de estadísticas con los datos calculados. Además despliega un botón que realiza la funcionalidad (descrita en CU_31) de descargar dichas estadísticas en un formato PDF,	
	4. El cliente selecciona otro periodo de tiempo (día, semana, semestre o año)		
	5. El sistema realiza el cálculo nuevamente desplegando la interfaz correspondiente.		
Postcondiciones	Ninguna		

Tabla 37: Ver Estadísticas de Ventas

6.2.2.31 Generar Informe de Ventas.

En la Tabla 38 se especifica el caso de uso “Generar Informe de Ventas” en donde el usuario puede generar un informe en formato PDF en base a las estadísticas de un determinado periodo de tiempo.

Caso de Uso		Generar Informe de Ventas	
ID	C_31		
Requisito ID	R_29		
Descripción	Permite descargar un documento PDF con información de estadísticas desplegada en CU_30		
Actores del Sistema	Administración.		
Precondiciones	El usuario debe estar autenticado.		
Flujos Principales	Actor	Sistema	
	Este caso de uso está incluido en el flujo principal del CU_30.		
	1. El cliente selecciona “Descargar PDF” para generar las estadísticas desplegadas en CU_30		2. El sistema realiza la conversión y genera el documento PDF,
	3. El usuario descarga el documento PDF		
Postcondiciones	PDF de estadísticas de ventas generado y descargado.		

Tabla 38: Caso de Uso Generar Informe de Ventas

6.3 Modelamiento de datos

Un modelo entidad relación (MER) representa de **manera conceptual** las principales entidades del sistema, con sus atributos y las correspondientes relaciones entre ellas.

Se destaca la entidad Pedido, la cual hace referencia a varias tablas mediante claves foráneas ya que es el núcleo del sistema, todo comienza con un pedido.

La entidad Usuario representa a los actores del sistema, los cuales se identifican mediante el atributo "rol". El Usuario con rol "recepción" gestiona uno o muchos pedidos, en donde los datos del Usuario asociados al Pedido se utilizan para fines estadísticos.

El Cliente, que tiene una o muchas Direcciones, realiza uno o muchos Pedidos, y un Pedido tiene asociada solo una venta. El pedido tiene un Detalle de Pedido, el cual tiene como propósito detallar los cilindros que solicitó el cliente. En la figura 6 se presenta el modelo entidad relación del sistema desarrollado en este proyecto.

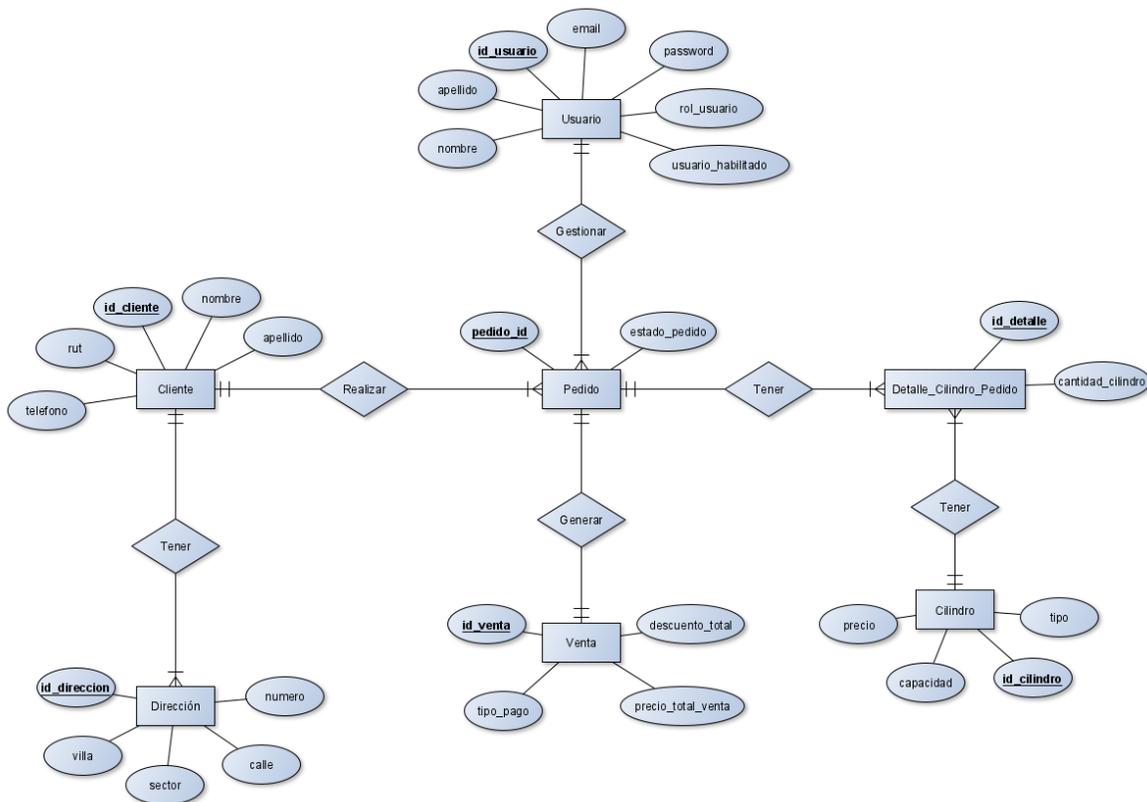


Figura 6: Modelo Entidad Relación (MER).

7 DISEÑO

7.1 Diseño orientado al consumo de servicios

El sistema de gestión de clientes y pedidos desarrollado sigue un diseño orientado al consumo de servicios. Básicamente, los servicios reflejan las actividades del negocio en el mundo real, las cuales forman parte de los procesos de negocio de la empresa.

En este proyecto, los servicios son **expuestos por la API (Backend) y consumidos por el cliente (Frontend)**, este último realiza todos los procesos que interactúan con el usuario, y con ayuda de los servicios, se garantizan tiempos de respuesta más rápidos y obtención de información más segura gracias al uso de estándares para la interoperabilidad del sistema.

Los diagramas presentados en esta sección fueron diseñados y confeccionados con la herramienta Draw.io. [27]

7.1.1 Recursos y Servicios

Un recurso es considerado como una entidad que representa un concepto del negocio y que puede ser expuesto y accedido por diferentes aplicaciones y/o sistemas. Para cumplir con los casos de uso detallados en la sección 6.2, se ha definido cuatro recursos y treinta y un servicios que se desarrollaron para la implementación del sistema, los cuales permiten alinear las funcionalidades del software con las necesidades del modelo de negocio de Gas Chillán, y así potenciar sus estrategias de negocio.

A continuación, en la Tabla 39 se detallan los recursos y servicios que el cliente (Frontend) consume de la API RESTful (Backend).

ID	Recurso	Servicio	CU Asociado
Q_Usuario_Auth	Usuario	Consulta que, a partir del RUT y la contraseña de un usuario, retorna el estado de la respuesta, ID, rol, nombres, apellidos y email del usuario. Además, la consulta retorna un token JWT por medio de la respuesta y cookies para la autenticación del usuario.	CU_01
Q_Usuario_Logout	Usuario	Consulta que, a partir del token del usuario, retorna el estado de respuesta para cerrar sesión del usuario.	CU_02
C_Estadisticas_S	Pedido, Ventas, Usuario	Consulta que retorna un conjunto de estadísticas semanales en base a todos los datos de los Recursos de Pedido, Ventas y Usuarios	CU_03
C_Estadisticas_M	Pedido, Ventas, Usuario	Consulta que retorna un conjunto de estadísticas mensuales en base a todos los datos de los Recursos de Pedido, Ventas y Usuarios	CU_04

C_Pedido	Pedido	Creación que registra un nuevo pedido con nombres y apellidos del recepcionista y chofer asociados, numero de pedido, estado, cliente asociado, dirección, detalle de cilindros, observación y fecha.	CU_05
Q_Pedidos	Pedido	Consulta que retorna el id, nombres y apellidos del recepcionista y chofer asociados, numero de pedido, estado, cliente asociado, dirección, detalle de cilindros, observación y fecha de creación de todos los pedidos.	CU_06 CU_10
Q_Pedido	Pedido	Consulta que, a partir del ID de un pedido retorna el id, nombres y apellidos del recepcionista y chofer asociados, numero de pedido, estado, cliente asociado, dirección, detalle de cilindros, observación y su fecha de creación.	CU_07 CU_10
U_Pedido	Pedido	Actualización que, a partir del ID de un pedido permite actualizar los nombres y apellidos del recepcionista y chofer asociados, numero de pedido, estado, cliente asociado, dirección, detalle de cilindros, observación o su fecha de creación.	CU_08 CU_11 CU_12
D_Pedido	Pedido	Eliminación que, a partir del ID de un pedido permite eliminar virtualmente un pedido.	CU_09
Q_Ventas	Venta	Consulta que retorna el id, pedido asociado, tipo de pago, subtotal, descuento total, total y fecha de creación de todas las ventas	CU_13
Q_Venta	Venta	Consulta que, a partir del ID de una venta retorna el id, pedido asociado, tipo de pago, subtotal, descuento total, total y fecha de creación de una venta.	CU_14
U_Venta	Venta	Actualización que, a partir del ID de una venta permite actualizar el pedido asociado, tipo de pago, subtotal, descuento total, total o fecha de creación de una venta.	CU_15
D_Venta	Venta	Eliminación que, a partir del ID de una venta, elimina virtualmente una venta.	CU_16
C_Cliente	Cliente	Creación que permite registrar un nuevo cliente con tipo de cliente, nombre, apellido, direcciones asociadas, teléfono y el email del cliente	CU_17
Q_Clientes	Cliente	Consulta que, a partir del ID de un cliente retorna el id, tipo de cliente, nombre, apellido, direcciones asociadas, teléfono, periodicidad de	CU_18

		compra, correo enviado, y el email del cliente	
Q_Cliente	Cliente	Consulta que, a partir del dato de un cliente retorna el id, tipo de cliente, nombre, apellido, direcciones asociadas, teléfono, periodicidad de compra, correo enviado, y el email del cliente	CU_19
D_Cliente	Cliente	Actualización que, a partir del ID de un cliente permite actualizar el, tipo de cliente, nombre, apellido, direcciones asociadas, teléfono, periodicidad de compra, correo enviado, o el email del cliente.	CU_20
D_Cliente	Cliente	Eliminación que, a partir del ID de un cliente permite eliminar virtualmente un cliente.	CU_21
C_Usuario	Usuario	Creación que permite registrar un usuario con nombres, apellidos, rol, rut, password, y email.	CU_22
Q_Usuarios	Usuario	Consulta que retorna el id, nombres, apellidos, rol, rut, password, email y fecha de ingreso de todos los usuarios	CU_23
Q_Usuario	Usuario	Consulta que, a partir del ID de un usuario, retorna el id, nombres, apellidos, rol, rut, password, email y fecha de ingreso de un usuario	CU_24
U_Usuario	Usuario	Actualización que, a partir del ID de un usuario, permite actualizar los nombres, apellidos, rol, rut, password, email y fecha de ingreso de un usuario	CU_25
D_Usuario	Usuario	Eliminación que, a partir del ID de un usuario permite eliminar virtualmente un usuario.	C_26
Q_Seguimiento	Cliente, Pedido	Consulta que permite obtener la predicción de compra y estado de predicción de todos los clientes en base a todos sus pedidos asociados.	CU_27
Q_Estadísticas_Venta	Venta	Consulta que retorna estadísticas de ventas de un periodo de tiempo determinado.	CU_30

Q: Query, **C:** Create, **U:** Update, **D:** Delete

Tabla 39: Recursos y Servicios.

De acuerdo con lo anterior, gran parte de los casos de uso quedan cubiertos por los recursos y servicios definidos, la excepción la constituyen los casos de uso CU_28, CU_29 y CU_31 que son funcionalidades que se realizan en el lado Frontend con el consumo de los datos proveídos por los servicios Q_Cliente, Q_Clientes y Q_Estadísticas_Venta, respectivamente.

Gran parte de los servicios son independientes, ya que no necesitan datos de otro servicio para funcionar, no obstante, el servicio de registrar pedido (C_Pedido) es la excepción, debido a que se necesita una secuencia de peticiones para obtener los datos que requiere un pedido.

Con este fin, **se incluyen diez servicios y dos recursos adicionales** para complementar la funcionalidad de registro de pedidos (los cuales son detallados en la Tabla 40), de modo que permita agilizar el proceso de obtención de la información al momento de gestionar los pedidos. **Por lo tanto, se implementó un total de cuarenta y seis servicios y seis recursos.**

En la figura 7 se muestra un esquema de la secuencia que sigue el servicio para registrar un pedido, en donde los servicios adicionales son Q_Direccion, C_Direccion y Q_Cilindros, los cuales se encuentran descritos y detallados en la Tabla 40. Se inicia con el servicio Q_User_Auth, luego se elige consultar por un cliente ya registrado (Q_Cliente) o crear un nuevo cliente (C_Cliente), posteriormente se elige entre consultar una dirección ya registrada (Q_Dirección) o crear una nueva dirección (C_Dirección), luego se consultan todos los cilindros (Q_Cilindros) y finalmente se realiza el servicio para registrar un pedido en el sistema (C_Pedido).

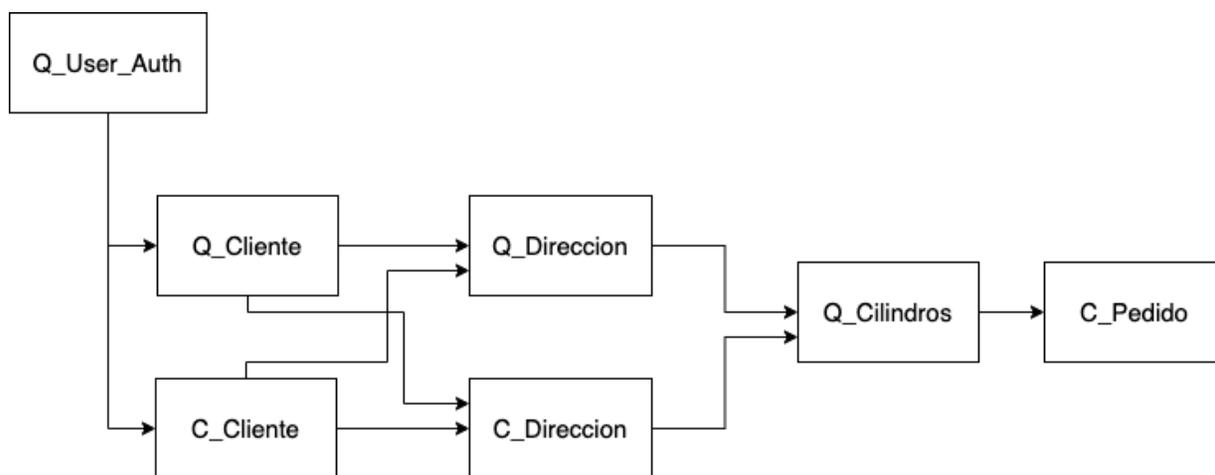


Figura 7: Secuencia de Servicios para Crear un Pedido.

A continuación, en la Tabla 40 se detallan los servicios adicionales consumidos por el Frontend, los cuales permiten que la funcionalidad de registrar pedido sea más eficiente en **términos de almacenamiento de información**, puesto que se evita la **redundancia de datos de direcciones y cilindros** asociadas a los registros de los pedidos del sistema.

ID	Recurso	Servicio
C_Direccion	Dirección	Creación que permite registrar una dirección con un numero, departamento, calle, villa y sector.
Q_Direccion	Dirección	Consulta que retorna el id, numero, departamento calle, villa y sector de todas las direcciones.
Q_Direccion	Dirección	Consulta que, a partir del ID de una dirección retorna la calle, villa, departamento y sector de la dirección.
U_Direccion	Dirección	Actualización que, a partir del ID de una dirección permite actualizar la calle, numero, departamento, villa o sector de la dirección.
D_Direccion	Dirección	Eliminación que, a partir del ID de una dirección permite remover virtualmente una dirección.
C_Cilindro	Cilindro	Creación que permite registrar un cilindro con un tipo, capacidad y precio
Q_Cilindros	Cilindro	Consulta que retorna el id, tipo, capacidad y precio de todos los cilindros
Q_Cilindro	Cilindro	Consulta que, a partir del ID de un cilindro retorna el id, tipo, capacidad y precio del cilindro.
U_Cilindro	Cilindro	Actualización que, a partir del ID de un cilindro retorna el id, tipo, capacidad y precio del cilindro.
D_Cilindro	Cilindro	Eliminación que, a partir del ID de un cilindro elimina virtualmente un cilindro.

Q: Query, **C:** Create, **U:** Update, **D:** Delete

Tabla 40: Recursos y Servicios adicionales.

7.2 REST (Frontend)

Como se menciona en la sección 7.1, el Frontend es quien consume los servicios proporcionados por la API RESTful, por lo tanto, se realizarán peticiones que, según la arquitectura REST descrita en la Sección 3.3.2.1, se deben realizar con la librería HttpClient que viene integrada con Angular, la cual permite convertir la API HTTP en una interfaz TypeScript con los métodos REST habituales y obtener las respuestas en formato JSON.

En la implementación, existen los Servicios de Angular, que son Clases TypeScript cuyo principal propósito es contener la lógica de negocio en base a las peticiones de servicios a una API, esto con el fin de que los componentes de la aplicación consuman los datos obtenidos mediante inyección de dependencias.

A continuación, en la Tabla 41 y 42 se muestran dos de los principales servicios consumidos por el Frontend; C_Pedido y Q_Client, que corresponden a métodos POST y GET respectivamente. Ambas tablas muestran datos asociados al servicio especificado, como su identificador, URL de acceso a los recursos en el servidor, el método REST asociado para enviar los datos, los parámetros que se deben enviar y un ejemplo del cuerpo o body de la petición. Se incluye además un ejemplo acotado de cómo HttpClient realiza la petición mediante la interfaz TypeScript que provee. Además, cabe destacar que el JWT Token no se envía en el cuerpo de la petición, debido a que se almacena en las cookies del navegador, por lo tanto, por cada petición que se realice, el token se enviará automáticamente al servidor.

ID Servicio	C_Pedido
URL	https://gas-chillan.herokuapp.com/api/v1
Método	POST
Recurso	/pedidos
Cuerpo de Petición	<pre>{ "cliente": "5f1e31c1cf9f78953623e026", "empleados": { "repcionista": "5eefe4d37f96944ae0f72a4c", "chofer": "5eefe43b11255b4260f29a2d" }, "direccion": "5eebddcb7c61f624b44dc9ab", "detalleCilindros": [{ "cantidad": 1, "cilindro": "5f0677962ce35f33d6e0e7ca" }, { "cantidad": 1, "cilindro": "5eec29c0b6e62640b8ee24ed" }], "observacion": "Pedido con varios cilindros" }</pre>

Llamada de ejemplo	<pre> obtenerPedidos(pagina='1', resultados='40', ordenar='cliente', ...filtros) { let url = `\${ this.urlApi }/pedidos`; let params = new HttpParams() .set('pagina', pagina) .set('resultados', resultados) .set('ordenar', ordenar) if (filtros.length > 0) { params = params.set(filtros[0].tipo, filtros[0].valor); } return this.http.get(url, {params}); } </pre>
---------------------------	--

Tabla 41: Ejemplo de petición POST

ID Servicio	Q_Cliente
URL	https://gas-chillan.herokuapp.com/api/v1
Método	GET
Recurso	/clientes
Cuerpo de Petición	No aplica
Llamada de ejemplo	<pre> obtenerClientes() { let url = `\${ this.urlApi }/clientes`; return this.http.get(url); } </pre>

Tabla 42: Ejemplo de petición GET.

7.3 REST (Backend)

De igual manera, como se menciona en la sección 7.1, la aplicación del lado del servidor (Backend) es quien expone los servicios mediante una API REST para que sean solicitados por el Frontend mediante peticiones.

Por lo tanto, en la implementación, dichas peticiones deben ser gestionadas por un servidor HTTP, que es implementado en la **capa de controlador** de la aplicación Express, dicho servidor provee métodos para acceder al **objeto de petición** y permite que esta **capa**, en conjunto con las demás capas, pueda realizar distintas operaciones en base a la información de ese objeto y posteriormente retornar un **objeto respuesta** al Frontend (características mencionadas en la sección 3.3.2.2). El contenido del **objeto petición** es lo que especifica el **servicio y el recurso** solicitado por el Frontend (el objeto de petición contiene los datos presentados en las tablas 41 y 42).

A continuación, en las Tablas 43 y 44 se muestran algunos de los datos correspondientes a los dos principales **objetos respuesta** que el Backend retorna al Frontend, y que son propios de los servicios Crear Pedido (C_Pedido) y Obtener Clientes (Q_Clientes) cuando se han realizado con éxito. Se presentan datos como el código y cuerpo de la respuesta, además, un ejemplo acotado de cómo la capa controlador de la aplicación gestiona las solicitudes entrantes para acceder a los recursos de los servicios.

ID Servicio	C_Pedido
Código	200
Cuerpo de Respuesta	<pre>{ "ok": true, "data": { "docs": { "numero_pedido": 999, "estado": "EN_CURSO", "_id": "5f205d98be07d0001715fe75", "cliente": "5f1e31c1cf9f78953623e026", "empleados": { "repcionista": "5eefe4d37f96944ae0f72a4c", "chofer": "5eefe43b11255b4260f29a2d" }, "direccion": "5eebddcb7c61f624b44dc9ab", "detalleCilindros": [{ "cantidad": 1, "cilindro": "5f0677962ce35f33d6e0e7ca" }, { "cantidad": 1, "cilindro": "5eec29c0b6e62640b8ee24ed" }] } } }</pre>

	<pre> "observacion": "Pedido con varios cilindros", "fecha_creacion": "2020-07-28T17:17:12.820Z", "_v": 0 } } } </pre>
Llamada de ejemplo	<pre> app.post('/', [crearFecha, catchAsync(obtenerNumeroPedido)], crud.crear(pedidoService)); </pre>

Tabla 43: Ejemplo de objeto de respuesta 1.

ID Servicio	Q_Cliente
Código de Respuesta	200
Cuerpo de la respuesta	<pre> { "ok": true, "data": { "docs": [{ "tipo_cliente": "NORMAL", "periodicidad_compra": "No registra fecha", "correoEnviado": false, "habilitado": true, "_id": "5f1ba2b00b939f141554f85d", "direcciones": ["5f1ba2c40b939f141554f85e"], "nombre": "Nicolas", "apellido": "Luna", "telefono": "61552064", "email": "nlunaxx@gmail.com" }] } } </pre>
Llamada de ejemplo	<pre> app.get('/', crud.obtenerTodos(clienteService)); </pre>

Tabla 44: Ejemplo de objeto de respuesta 2.

7.4 Diagrama de Secuencia y de Clases

Para representar la estructura del sistema, se presentan **dos diagramas de secuencia** que representan las capas que implementan CU_05 Y CU_18 desde el lado del servidor (Backend). Del mismo modo, se presentan **dos diagramas de clase** que corresponden a los componentes que implementan CU_05 Y CU_18 en la aplicación Frontend.

El objetivo de estos diagramas es explicar **cómo interactúan ambas partes del sistema** al momento de utilizar los servicios, particularmente se eligen dos de los más utilizados en el sistema; crear pedido (C_Pedido) y obtener clientes (Q_Clientes), puesto que permiten obtener una representación general de alto nivel sobre cómo funcionan todos los servicios que, de igual manera, siguen la misma estructura. Esto debido a que no es posible representar el sistema completo a través de solo cuatro diagramas, a causa de que en las aplicaciones Frontend y Backend existen treinta y dos componentes (en el Frontend) y veinticinco objetos (en el Backend) que implementan los servicios descritos anteriormente.

7.4.1 Diagramas de Clases (Frontend)

Un diagrama de clase tiene como objetivo representar claramente la estructura de un sistema y/o aplicación con sus correspondientes clases, atributos, métodos y relaciones entre dichas clases [28].

De este modo, los diagramas de clase presentados aquí tienen como principal propósito ilustrar la interacciones entre los componentes de la aplicación del lado del cliente (Angular), representando principalmente la inyección de dependencias que realizan los componentes de Angular para acceder a la información provista por los servicios e implementar las vistas en base a esa información (características mencionadas en la sección 7.2).

Cabe destacar que en los componentes RegistrarPedidoComponent.ts y ClientesComponent.ts se han omitido los atributos que están asociados a interfaces de usuario (Layout y Templates), debido a que no son relevantes para lo que se busca representar aquí, por lo tanto, se expresan los caracteres de puntos suspensivos para dar entender que se implementan más atributos

En la figura 8, se presenta el diagrama de clases correspondiente al consumo de datos en la aplicación Angular del servicio asociado con el caso de uso CU_05 (Crear Pedido).

En este diagrama se puede distinguir la **inyección de los servicios en el componente RegistrarPedidosComponent**, el cual consume los datos provistos por los servicios de Cliente, Dirección y Cilindro, que se encargan de la comunicación con los servicios de la API mediante métodos de la librería HttpClient.

Los servicios utilizados por este componente corresponden a las clases: ClienteService.ts, DireccionService.ts y CilindroService.ts, en donde cada uno de ellos crea una instancia de un objeto para modelar la información que utiliza.

Finalmente, para registrar un pedido, el componente **RegistrarPedidosComponent**, utiliza el servicio PedidoService.ts, el cual crea una instancia de Pedido.ts para modelar toda la información del pedido y enviar la solicitud al servicio del Backend, el cual se encarga de registrar un pedido en la base de datos.

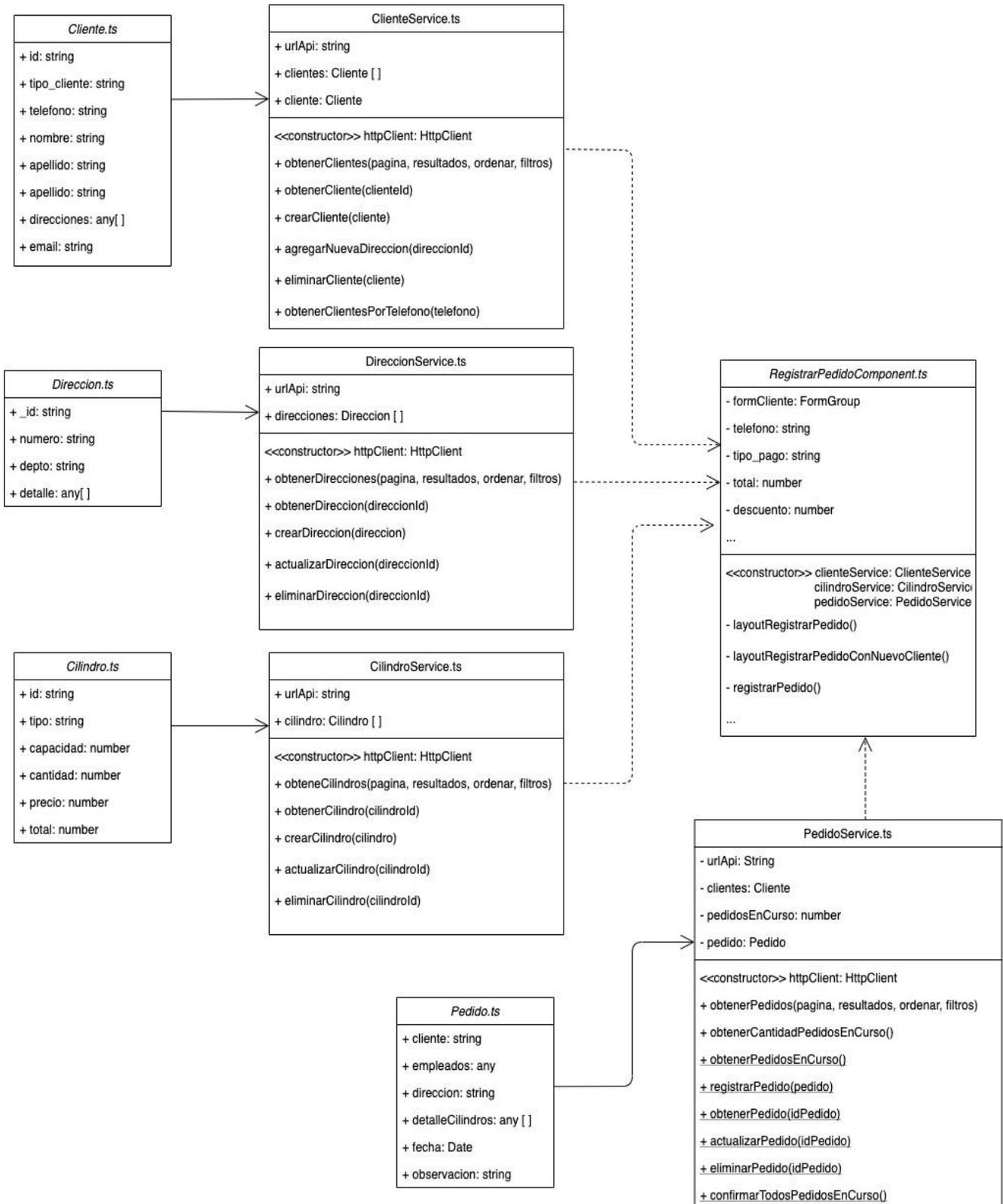


Figura 8: Diagrama de Clases del componente que implementa CU_05.

Por otro lado, en la Figura 9 se presenta el diagrama de clases correspondiente al consumo de los datos en la aplicación Angular del servicio asociado al CU_18 (Obtener Clientes).

En este diagrama, se puede distinguir la **inyección de los servicios en el componente ClientesComponent**, el cual consume los datos provistos por el servicio de Cliente, que se encarga de la comunicación con el servicio de la API mediante métodos de la librería HttpClient.

Este componente permite visualizar todos los clientes registrados en el sistema. Los servicios utilizados por este componente es la clase ClienteService.ts, la cual crea una instancia de un objeto Cliente.ts para modelar la información obtenida de la petición realizada al Backend.

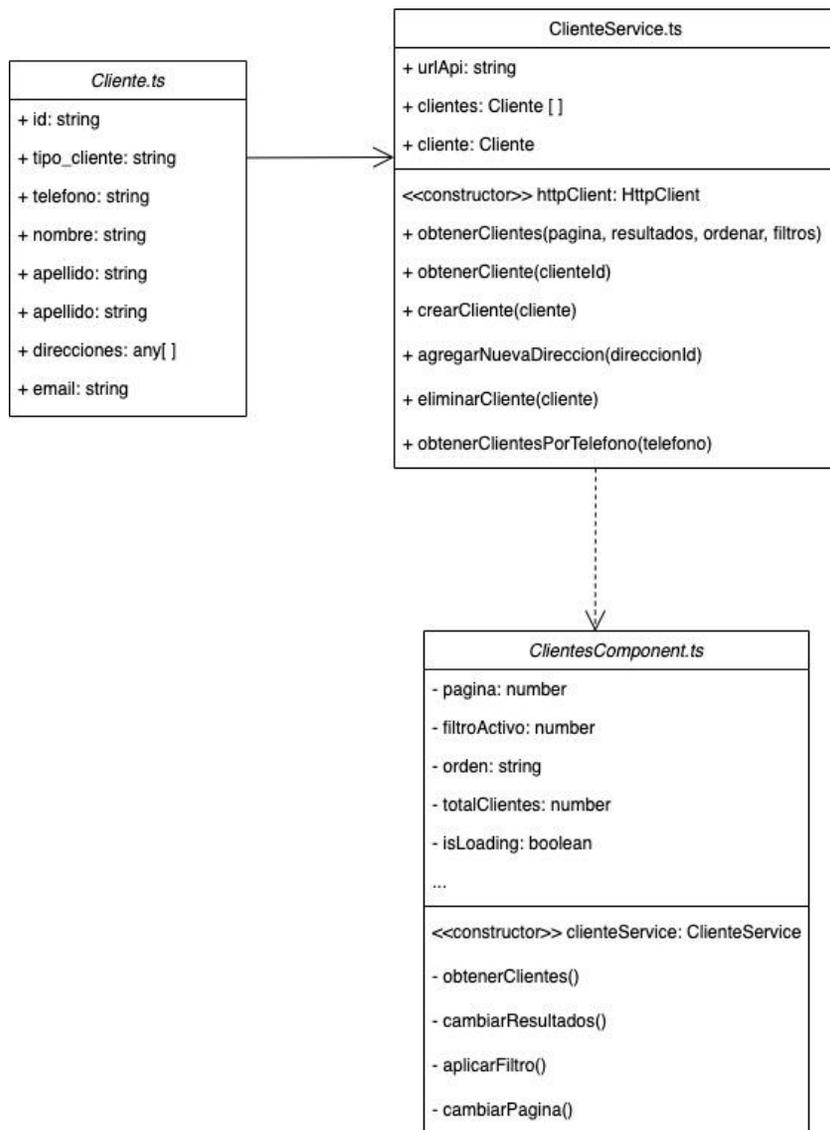


Figura 9: Diagrama de Clases del componente que implementa CU_18.

7.4.2 Diagramas de Secuencia (Backend)

Los diagramas de secuencia presentados aquí tienen como principal propósito ilustrar la interacción de las tres capas (en la aplicación Express) que implementan los servicios mencionados anteriormente. Cada capa representa el **módulo del recurso** que realiza una funcionalidad específica, como por ejemplo procesar la rutas, implementar lógica de negocio o interactuar con la base de datos.

Como se menciona en la sección 3.3.2.2, existen tres capas que permiten realizar todo el proceso del servicio al que apunta la petición del cliente (Frontend) y retornar una respuesta en concordancia.

En la Figura 10 se presenta el diagrama de secuencia correspondiente al servicio asociado a CU_05 (Registrar Pedido), en donde el Frontend es quien comienza el flujo de secuencia con una petición al Backend. En primer lugar, entran en acción los módulos de App y PedidosRoutes que corresponden a la **capa controlador**, que es donde se implementa el servidor http para gestionar todas las solicitudes entrantes y se analiza la URL para acceder al objeto petición (req) y saber que se trata de una petición con método POST para la creación de un pedido (1 y 2).

Posteriormente, PedidosRoutes crea una instancia de PedidoService para pasarla como argumento a la función **crear** (3) del módulo CrudFactory. Luego, este extrae el Modelo instanciado en PedidoService, que hace referencia al modelo de Pedido y lo pasa como argumento a la función **crear** (4) del módulo CRUDFactoryService que corresponde a la **capa de servicio**. Como siguiente paso, la capa de servicio se comunica con el módulo Pedido mediante el método **create** (5) del Modelo recibido como parámetro, el cual hace referencia a dicho módulo y que corresponde a la **capa de modelo**. La función **create** recibe como argumento el **objeto pedido** extraído del cuerpo del **objeto petición** (req.body).

Luego, esta capa **retorna un objeto** que contiene los datos del **pedido creado** (6), en donde el módulo CRUDFactoryService recibe este objeto y lo retorna a CrudFactory (7) que también hará el retorno al módulo PedidosRoutes (8), el cual gestiona el envío del pedido creado. Finalmente, App recibe este objeto (9) con todas las configuraciones correspondientes y envía la respuesta (res) al cliente (10).

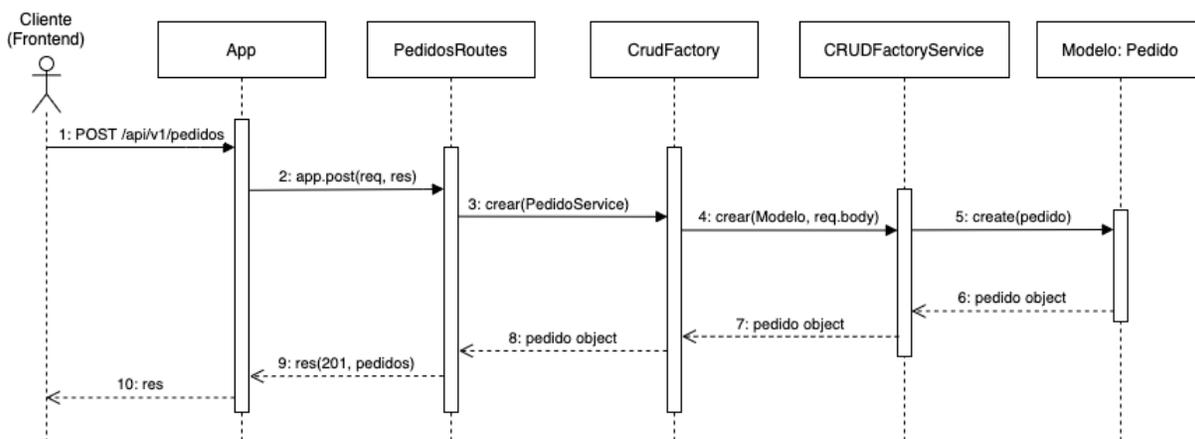


Figura 10: Diagrama de secuencia del servicio asociado a CU_05.

Por otro lado, la Figura 11 corresponde al servicio asociado a CU_18 (Obtener Clientes), en donde se realizan las mismas interacciones explicadas en la descripción de la Figura 10, con la diferencia de que la petición realizada por el Frontend es un **método GET** y en el módulo CRUDFactoryService se establece comunicación con el módulo de Cliente mediante el Modelo que fue recibido como parámetro y utilizando la función **find** (4), la cual permite obtener todos los clientes registrados en la base de datos.

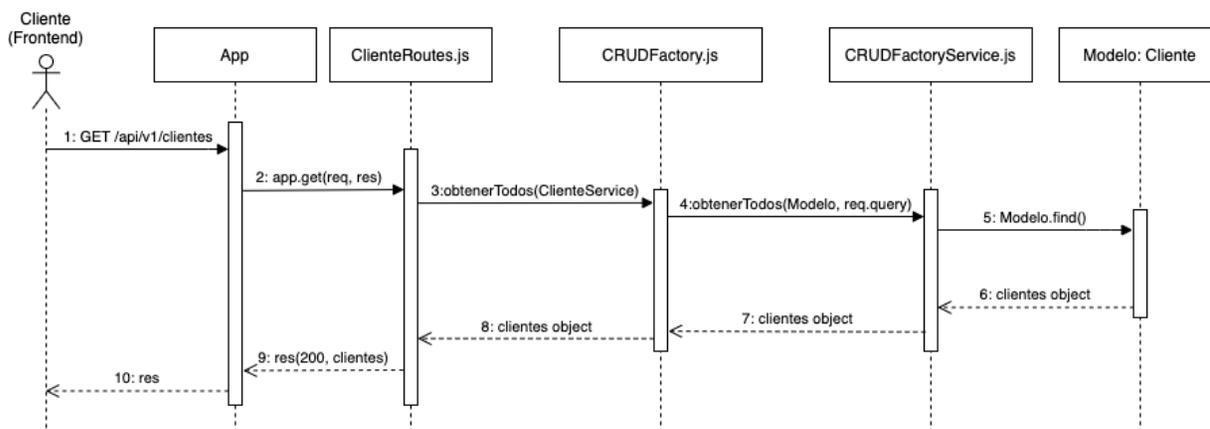


Figura 11: Diagrama de secuencia del servicio asociado a CU_18.

7.5 Diseño de Físico de la Base de datos

El modelo entidad relación proporciona una base para diseñar el modelo físico de la base de datos, el cual representa una solución para la construcción de esta. Sin embargo, una de las principales características de este tipo de modelos es que está orientado a dar una abstracción para la implementación de bases de datos relacionales, las cuales ocupan un lenguaje de consultas SQL. No obstante, en este proyecto se ocupa MongoDB, que es una base de datos no relacional (NoSQL) orientada a documentos y de código abierto que ofrece una gran escalabilidad, flexibilidad y un modelo de consultas e indexación avanzado. Por lo tanto, en el diseño físico de la base de datos de este proyecto se deben tener en cuenta otros conceptos, que son definidos a continuación.

7.5.1 MongoDB

MongoDB ofrece la posibilidad de almacenar datos con un esquema flexible y dinámico permitiendo usar “datos no estructurados”. Esta es una ventaja sobre las bases de datos relacionales SQL, en donde se debe definir y declarar la estructura de los datos antes de insertarlos en la base de datos, la modificación de esta estructura se complica una vez que el sistema crece en complejidad. MongoDB almacena datos en documentos, y las colecciones de dichos documentos no imponen una estructura, incluso si tienen una estructura similar.

Por lo tanto, el desafío de modelar los datos con MongoDB (y su diseño físico) es equilibrar su flexibilidad con las necesidades que se presentan en el desarrollo Backend para acceder a datos mediante patrones predecibles, teniendo un modelo para que los desarrolladores se guíen de una manera intuitiva mediante un diagrama y puedan implementar la base de datos.

7.5.2 Justificación del uso de MongoDB

En este proyecto se utiliza una base de datos NoSQL porque tiende a ser una mejor opción para aplicaciones web modernas que tienen conjuntos de datos más complejos y en constante cambio, requiriendo un modelo de datos flexible que no necesita ser definido de forma rígida.

La mayoría de los desarrolladores u organizaciones que prefieren las bases de datos NoSQL, se sienten atraídos por las características ágiles que les permiten entrar al mercado más rápido y hacer actualizaciones de manera más eficaz [29]. A diferencia de las bases de datos relacionales tradicionales basadas en SQL, las bases de datos NoSQL pueden almacenar y procesar datos en tiempo real con mucho mejor rendimiento, pudiendo integrar características como chat con clientes y choferes en tiempo real, características que, en el alcance de este proyecto no están consideradas por factores de tiempo, por lo tanto, serán desarrolladas e integradas en un futuro [30].

Por otro lado, si bien las bases de datos SQL tienen algunos casos de uso específicos, las bases de datos NoSQL tienen muchas características que las bases de datos SQL no son capaces de manejar sin enormes costos y sacrificios críticos de velocidad o agilidad para un proyecto de este tipo [31].

7.5.3 Conceptos de MongoDB

Como se menciona anteriormente, MongoDB es una base de datos documental y no una base de datos relacional. Es una de las bases de datos que han surgido del movimiento NoSQL y que intenta incrementar la escalabilidad de un sistema. La idea básica es sustituir el **concepto de fila por el concepto de documento**. A continuación, se introducen los principales conceptos teóricos de esta base de datos, debido a que, como es una tecnología emergente, se debe entrar en contexto para lograr una mayor comprensión del diseño físico de la base de datos presentado en la siguiente sección.

- **Documento (Document):** es la unidad básica de datos en MongoDB, normalmente equivale al concepto de fila en una base de datos no relacional. Se almacenan como “Documentos BSON”, que básicamente es JSON en una forma binaria y contiene los campos (fields) del documento, similar a las columnas.
- **Colección (Collection):** las colecciones son un contenedor que almacena los documentos, similar a las tablas en base de datos no relaciones.

A continuación, en la Figura 12 se presenta la jerarquía de estos conceptos y la comparación con una base de datos relacional (RDBMS).

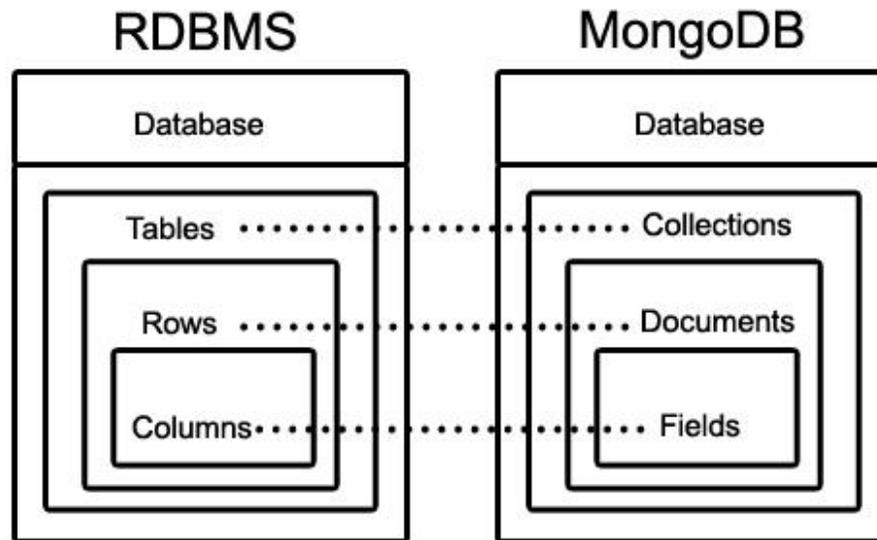


Figura 12: Comparación de Base de Dato Relacional y No Relacional [38].

Otro de los puntos clave del diseño de una base de datos MongoDB gira en torno a la **estructura que tienen los documentos** y cómo se **representan las relaciones entre los datos**. Debido a esto, una de las principales características de esta base de datos, es que los datos (que puede ser una colección entera) que están relacionados con una colección pueden ser incrustados en un solo documento. Por lo tanto, surgen dos conceptos muy importantes a la hora de diseñar; **Embedded Data y References**, que dan paso a la desnormalización y normalización (respectivamente) de las colecciones relacionadas. A continuación, se explican brevemente estos conceptos:

- **Embedded Data:** se incrustan datos relacionados en un solo documento. En la Figura 13 se presenta un ejemplo de este tipo de estructura.



Figura 13: Embedded Data MongoDB [39].

- References:** Las referencias almacenan las relaciones entre los datos al incluir enlaces o referencias de un documento de otra colección. Las aplicaciones pueden resolver estas referencias para acceder a los datos relacionados. En términos generales, estos son modelos de datos normalizados. En la Figura 14 se presenta un ejemplo de este tipo de estructura.

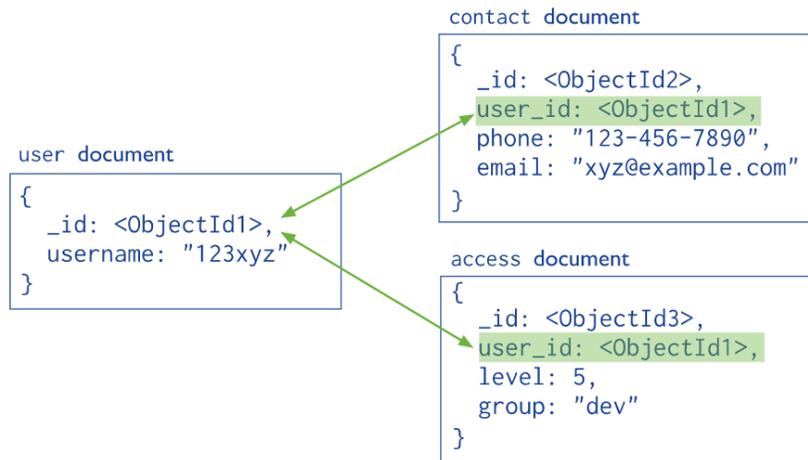


Figura 14: References MongoDB [40].

7.5.4 Diseño físico de la Base de Datos.

Para la construcción del diseño físico de la base de datos se utilizó una herramienta de modelado para bases de datos MongoDB que permite representar el diseño físico y luego conectar con la base de datos, permitiendo crear las colecciones con sus documentos en ella.

Para este proyecto se ocupó la herramienta **Moon Modeler** [32], que permitió definir visualmente el esquema de la bases de datos.

A continuación, en la Figura 15 se encuentra el diagrama del modelo físico de la base de datos MongoDB del sistema, si bien exhibe diferencias con el modelo entidad relación (MER) representado en la figura 6, ello es debido al proceso de desnormalización y normalización de MongoDB.

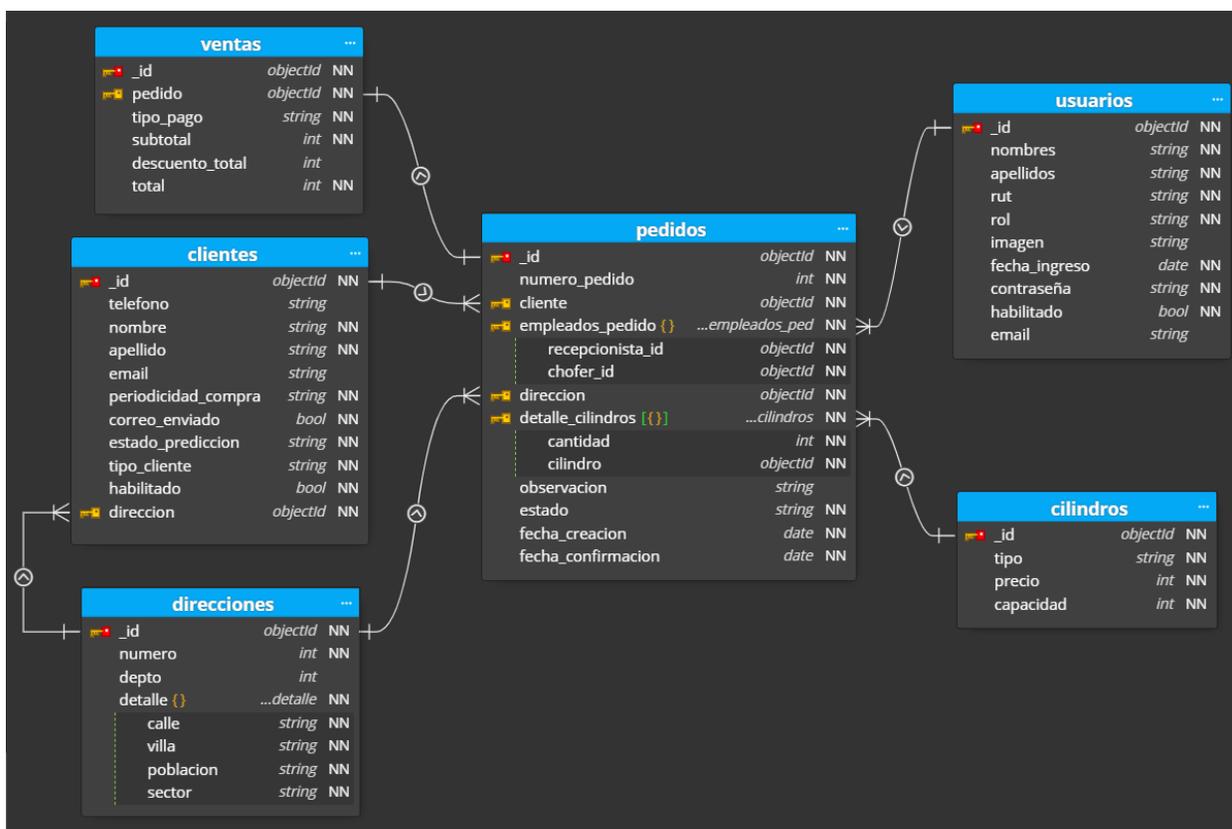


Figura 15: Diseño Físico de la Base de Datos.

7.6 Diseño interfaz y navegación

Para el desarrollo de la navegación de interfaz de usuario se utilizó la aplicación LucidChart [33], la cual permite diseñar una gran variedad de modelos, en este caso se diseñó un modelo conceptual de las navegaciones de la aplicación.

7.6.1 Mapas de navegación

Los mapas de navegación son esquemas de organización de la información de una estructura web, la cual indica las bases principales de la aplicación, se expresa en términos de jerarquía y secuencia. A continuación, se presentan estos mapas de navegación según el rol del usuario del sistema.

A) Recepcionista:

En la Figura 16 se especifica el mapa de navegación del usuario que tiene rol recepcionista

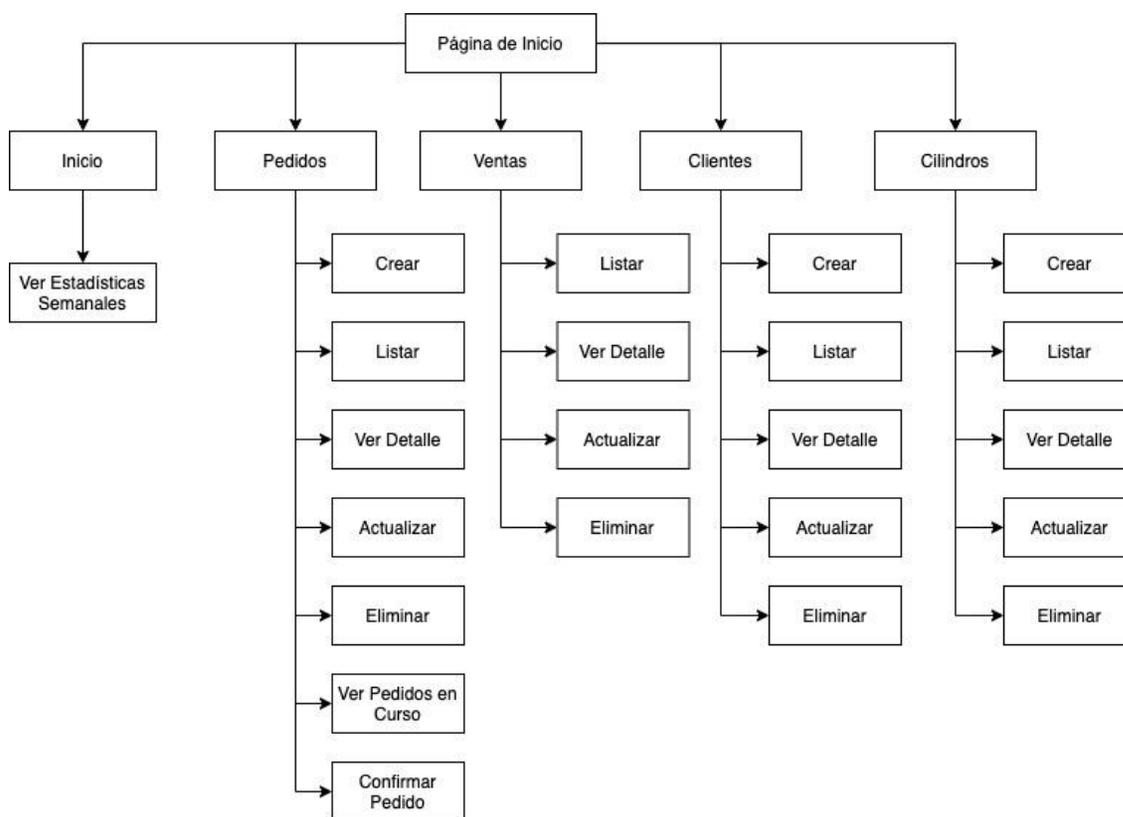


Figura 16: Mapa de Navegación de Recepción.

B) Administrador:

En la figura 17 se especifica el mapa de navegación del usuario que tiene rol administrador.

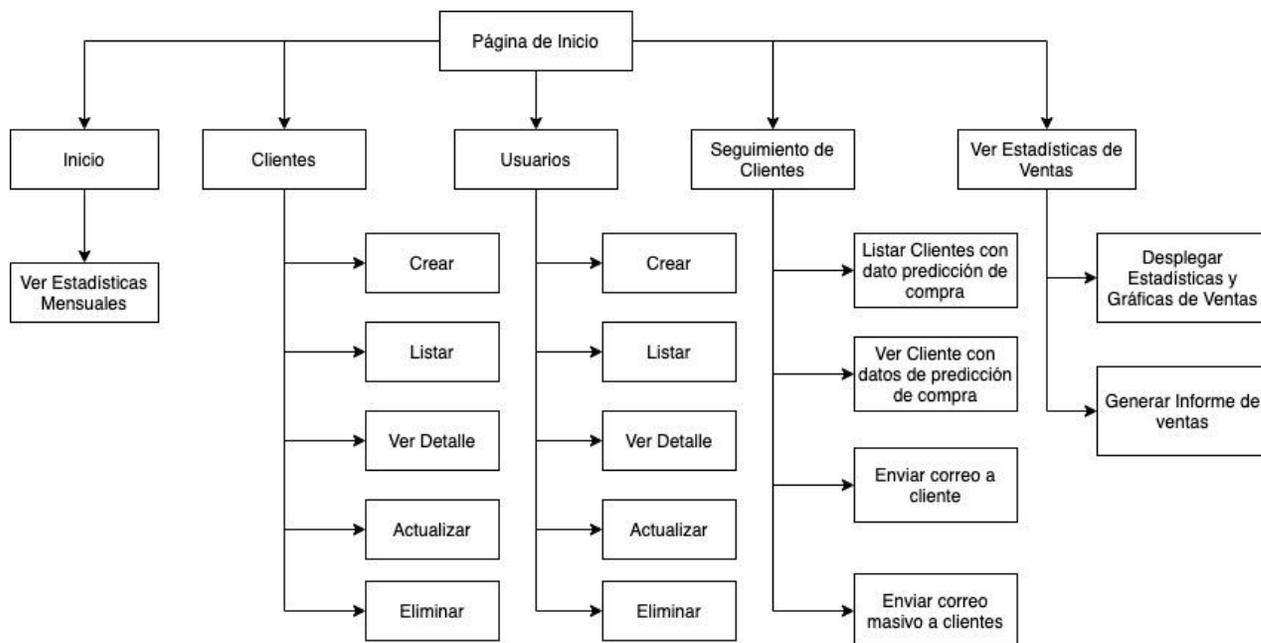


Figura 17: Mapa de Navegación de Administración.

7.7 Diseño interfaz del Frontend

En esta sección se presentan los diseños de las principales interfaces de la aplicación en sus dos versiones de pantalla; escritorio y móvil (Responsive), en donde la versión móvil posee las mismas funcionalidades que las de escritorio con la diferencia de que el contenido se adapta a pantallas con dimensiones más pequeñas (Tablets y Smartphones, generalmente debajo de los 1200px de ancho de pantalla)

Se adjuntan capturas de pantalla basadas en las vistas que fueron diseñadas e implementadas durante el desarrollo, por lo que el diseño final de la aplicación puede variar con las interfaces presentadas aquí. Además, se incluye una breve descripción para comprender su funcionalidad e interacción con el usuario.

Una característica importante del sistema es que las interfaces a desplegar en el Frontend dependerán del **actor** que inicie sesión en el sistema. Es decir, si el usuario que se ha autenticado es un/a recepcionista se desplegarán las interfaces de las funcionalidades permitidas, específicamente las funcionalidades mencionadas en la sección (6.2.1). De igual manera sucede con el actor de administración que se autentique.

Por lo tanto, **la aplicación se divide en dos módulos, administrador y recepcionista**, dado esto, en esta sección se presentarán los diseños de las interfaces para ambos módulos. Sin embargo, es importante mencionar que la vista de inicio de sesión es igual para ambos módulos.

Además, cabe destacar que se han omitido las interfaces de Clientes, Cilindros, Direcciones y Ventas correspondientes al módulo de recepcionista, debido a que la vista presentada en 7.7.3.3 de “Pedidos” es similar a todas las vistas señaladas.

7.7.1 Inicio de Sesión de la aplicación (Escritorio)

La Figura 18 corresponde a la vista de escritorio de inicio de sesión, desplegada cuando el actor no está autenticado en el sistema. Permite ingresar el Rut y Contraseña del actor que desea ingresar. Además, existe la opción de recordar el Rut con el propósito de agilizar el próximo inicio de sesión que el actor realice.

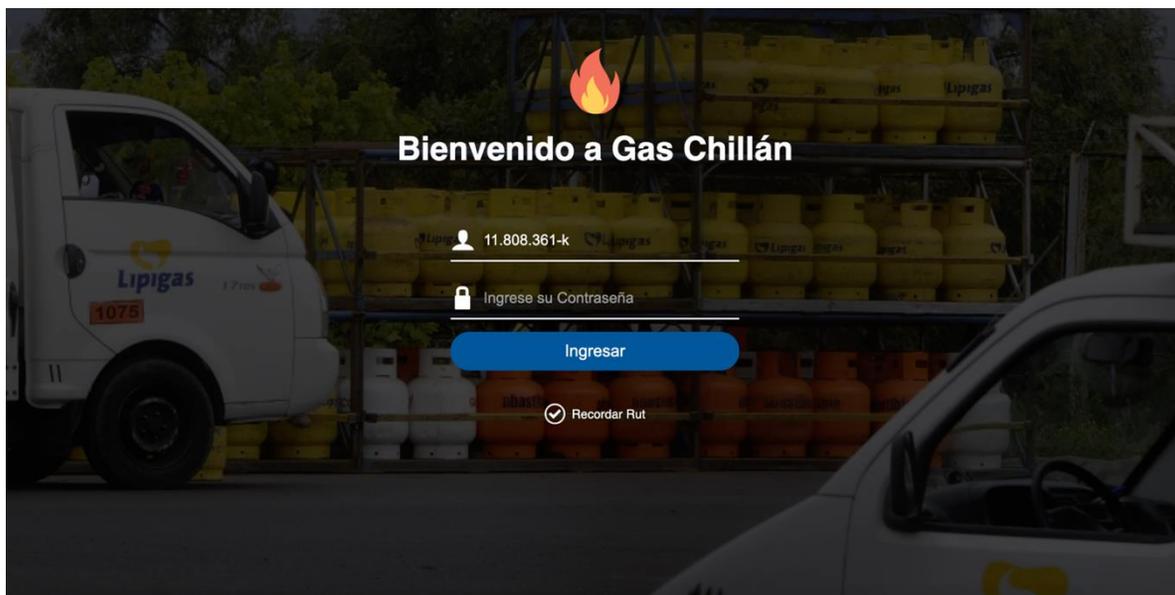


Figura 18: Interfaz de Inicio de Sesión.

7.7.2 Inicio de Sesión de la aplicación (Móvil)

La Figura 19 corresponde a la vista móvil de inicio de sesión.



Figura 19: Interfaz de Inicio de Sesión – Móvil.

7.7.3 Módulo de Recepción

El módulo de recepción corresponde a todas las funcionalidades relacionadas con la gestión de pedidos y clientes. Específicamente puede ejercer las operaciones básicas sobre los recursos a los que tiene permisos el actor recepcionista y que son mencionados en la sección 7.1.1.

7.7.3.1 Inicio (Escritorio)

La Figura 20 corresponde a la versión de escritorio del Inicio que se despliega cuando el actor recepcionista se ha autenticado en el sistema.



Figura 20: Inicio Recepción.

7.7.3.2 Inicio (Móvil)

La Figura 21 es la versión móvil del Inicio que se despliega cuando el actor recepcionista se ha autenticado en el sistema.



Figura 21: Inicio Recepción – Móvil.

7.7.3.3 Pedidos (Escritorio)

La Figura 22 corresponde a la vista que despliega todos los pedidos registrados en el sistema en base a filtros que permiten al usuario buscar un pedido por diferentes criterios. Además, la opción de ordenar por atributos establecidos, mostrar una cantidad determinada de resultados y la paginación, la cual permite estructurar el contenido de todos los pedidos de forma agrupada y minimalista.

CLIENTE	RECEPCIONISTA	CILINDROS	ESTADO	ACCIONES
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	
Camila	Carola Del Carmen	NO_CAT 11 kgs, CAT 15 kgs ...	CONFIRMADO	

Figura 22: Pedidos Recepción.

7.7.3.4 Pedidos (Móvil)

La Figura 23 es la vista móvil que despliega todos los pedidos registrados. Posee una particular diferencia con la de escritorio, que es la visualización de la tabla, el usuario deberá deslizarla hacia la derecha para poder visualizar todos los datos asociados a los pedidos.

NÚMERO	CLIENTE	RECEPCIÓN
1	Diana Sepulveda	Magdalena
2	Diana Sepulveda	Magdalena
3	Diana Sepulveda	Magdalena
4	Juan Perez	Magdalena
4	Fernando Herrera	Magdalena
5	Juan Perez	Magdalena
5	Fernando Herrera	Magdalena
6	Filiberto Del Valle	Magdalena

Figura 23: Pedidos Recepción – Móvil.

7.7.3.5 Registrar Pedido (Escritorio)

Corresponde a la vista para registrar un pedido en el sistema, esta vista se divide en dos interfaces, en donde la primera permite a recepción buscar el cliente que solicita el pedido mediante su número telefónico, o en su defecto, crear un pedido con nuevo cliente.

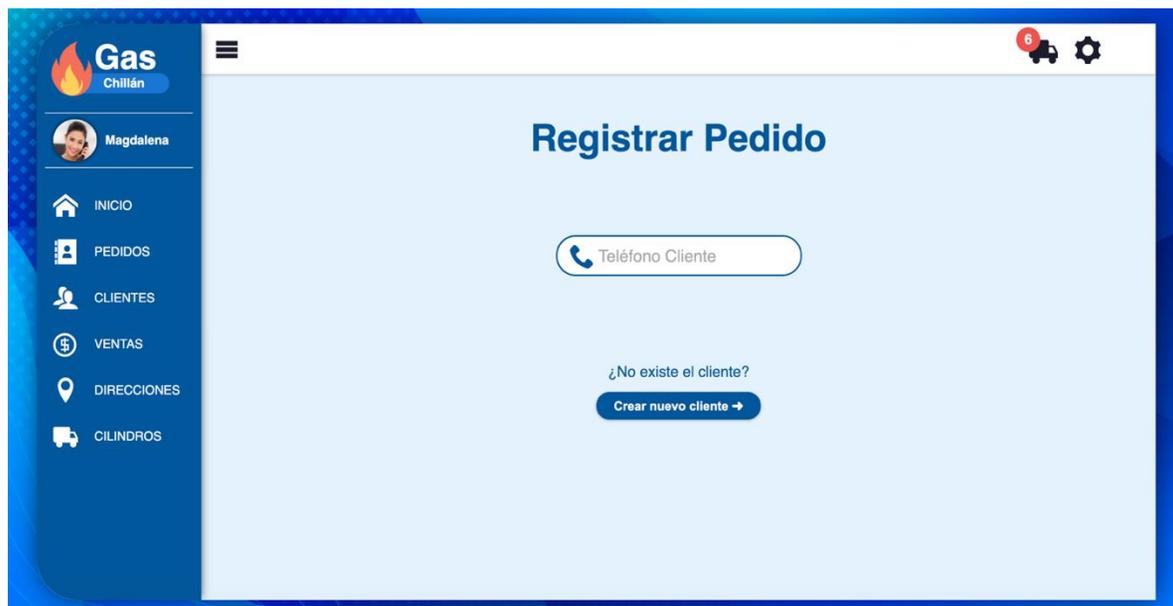


Figura 24: Interfaz 1 Registrar Pedido.

Una vez encontrado el cliente con su número telefónico o elegida la opción de crear un pedido con un nuevo cliente, se muestra la interfaz para crear un pedido. Aquí, el actor recepcionista puede ingresar todos los datos que necesita un pedido y la funcionalidad de registrarlo en el sistema.

La Figura 25 corresponde a esta segunda vista de registrar pedido.

Gas Chillán

Magdalena

- INICIO
- PEDIDOS
- CLIENTES
- VENTAS
- DIRECCIONES
- CILINDROS

Datos Cliente

Tipo Cliente: NORMAL

Nombre: Nicolas

Apellido: Luna

Teléfono: 982631970

Email: No tiene correo asociado

Actualizar Cliente

Dirección: Arica Zañartu Ultraestacion 279

Chofer: Seleccione un chofer

Ingrese los cilindros del pedido

Seleccione un cilindro: 1

Tipo	Kilos	Cantidad	Precio	Total	Acción
Agregue los cilindros del pedido					

Seleccione Tipo de Pago

+ Añadir Observación

Subtotal	\$	0
Descuento	\$	0
Total	\$	0

Registrar Cancelar

Figura 25: Interfaz 2 Registrar Pedido.

7.7.3.6 Registrar Pedido (Móvil)

La Figura 26 y 27 corresponden a la versión móvil de la vista para registrar un pedido.

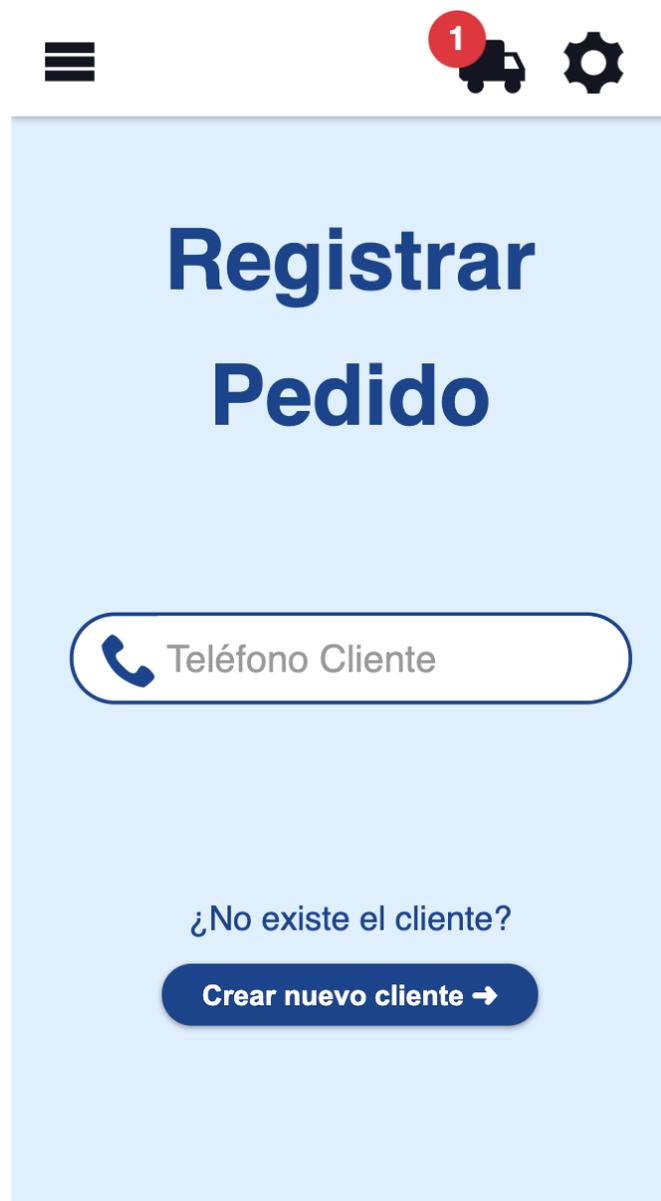


Figura 26: Interfaz 1 Registrar Pedido – Móvil.

Dirección

+ Simon Bolivar Zañartu Ultra Estacion 618 ▾

Chofer

Seleccione un chofer ▾

Ingrese los cilindros del pedido

Seleccione un cilindro ▾ 1 ✓

Tipo	Kilos	Cantidad	Precio	Total
Agregue los cilindros del pedido				
Seleccione Tipo de Pago ▾				
+ Añadir Observación				
Subtotal		\$	0	
Descuento		\$	0	
Total		\$	0	

Registrar
Cancelar

Figura 27: Interfaz 2 Registrar Pedido – Móvil.

7.7.3.7 Ver Pedidos En Curso (Escritorio)

La Figura 28 corresponde a la vista que permite visualizar todos los pedidos que han sido registrados durante una jornada. El recepcionista puede confirmar o cancelar cada uno de ellos de forma individual, o puede **confirmarlos o cancelarlos todos** para acelerar el proceso de confirmación de pedidos.

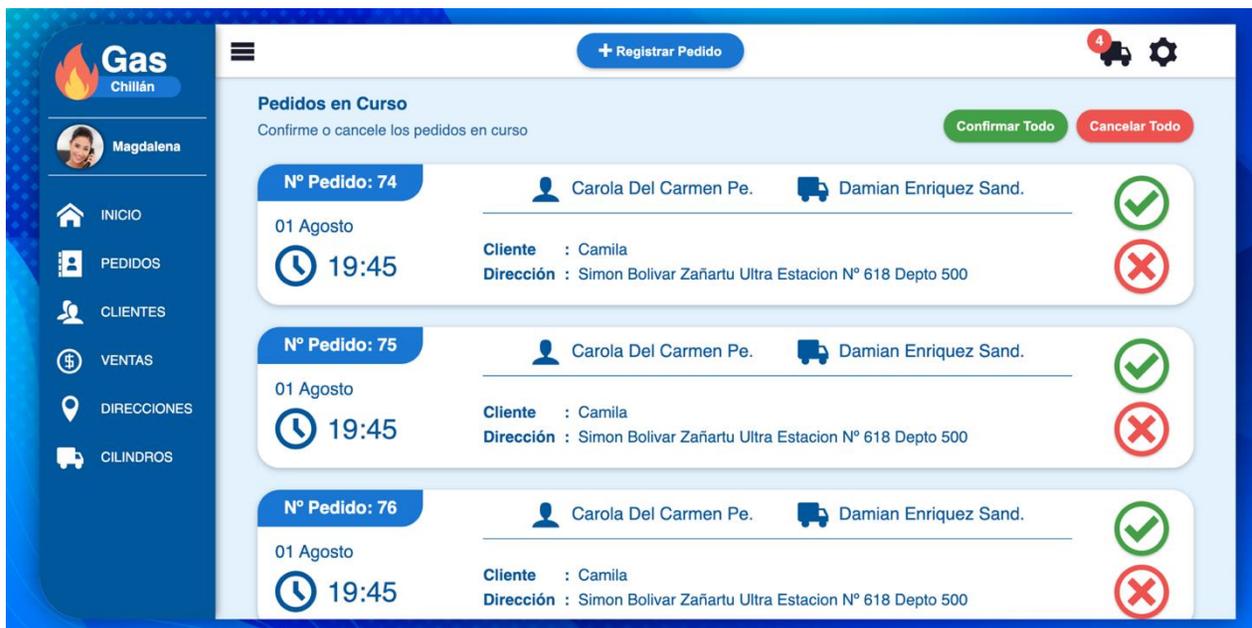


Figura 28: Ver Pedidos en Curso.

7.7.3.8 Ver Pedidos En Curso (Móvil)

La Figura 29 es la versión móvil de la vista que despliega todos los pedidos en curso.

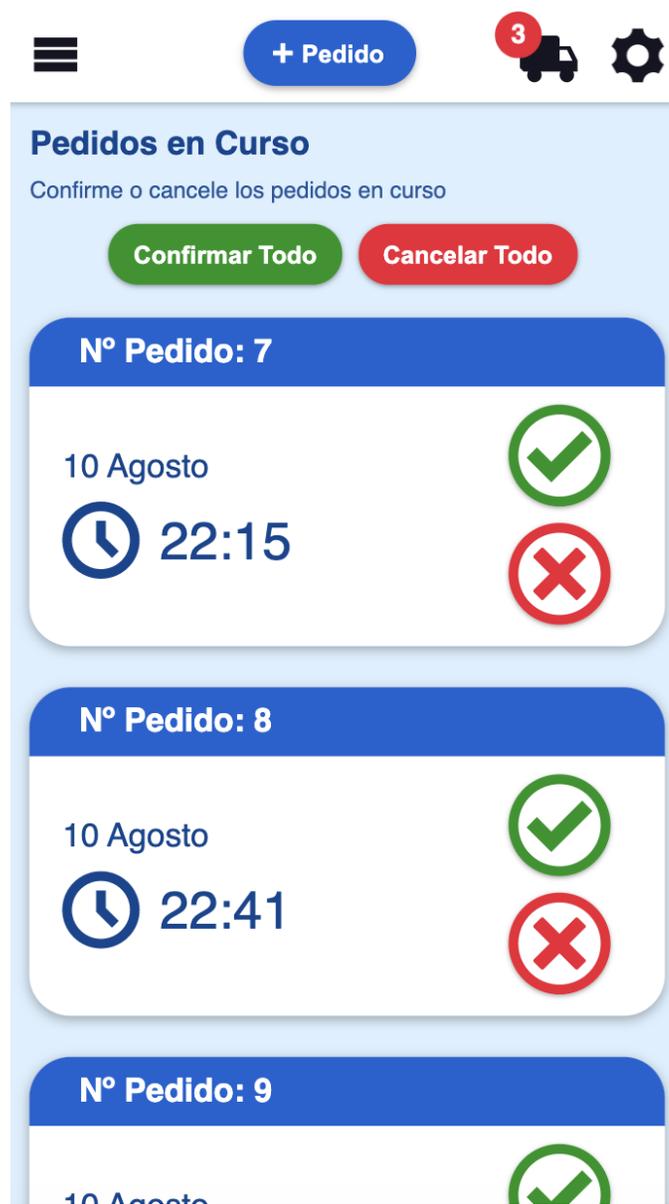


Figura 29: Pedidos en Curso – Móvil.

7.7.4 Módulo de Administración

El módulo de administración se enfoca principalmente en la gestión de clientes y ventas registradas en el sistema. Puede realizar una operación avanzada sobre los clientes como el seguimiento y envío de correos.

7.7.4.1 Inicio (Escritorio)

La Figura 30 corresponde a la versión de escritorio del Inicio que se despliega cuando el actor administrador se ha autenticado en el sistema.



Figura 30: Inicio Administración.

7.7.4.2 Inicio (Móvil)

La Figura 31 es la versión móvil del Inicio que se despliega cuando el actor administrador se ha autenticado en el sistema.

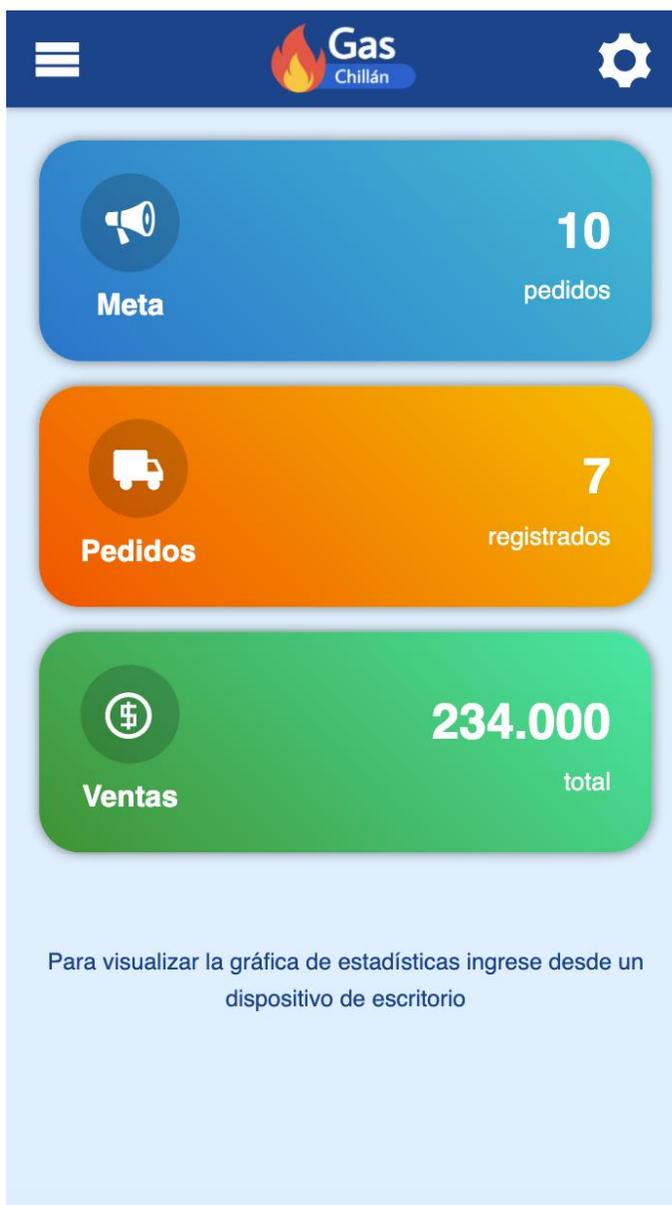


Figura 31: Inicio Administración – Móvil.

7.7.4.3 Seguimiento de Cliente (Escritorio)

La Figura 32 corresponde a la vista en versión de escritorio del seguimiento de clientes del sistema. En donde el administrador puede visualizar información asociada a las compras de los clientes, principalmente su periodicidad y predicción de compra.

Seguimiento de Clientes
Seguimiento y predicción de compra

+ Enviar Correo Masivo Ordenar por: Predicción

CLIENTE	TELÉFONO	ÚLTIMA COMPRA	PERIODICIDAD	PREDICCIÓN	ACCIÓN
Fernando Herrera	989045678	11 Agosto	Sin compras	Sin compras	
Diana Sepulveda	987612390	02 Julio	Cada 18 días	COMPRA_ATRASADA	Enviar Correo
Nicolás Luna	961552064	Sin compras	Sin compras	Sin compras	
Juan Perez	98745678	11 Julio	Cada 61 días	CLIENTE_ACTIVO	
Filberto Del Valle	961552064	13 Agosto	Cada 32 días	CLIENTE_ACTIVO	
Victor Palma	954678932	11 Agosto	Solo 1 compra	Solo 1 compra	

Figura 32: Seguimiento de Cliente.

7.7.4.4 Seguimiento de Cliente (Móvil)

La Figura 33 es la versión móvil de la vista del seguimiento de clientes del sistema. El actor administrador debe desplazar la tabla para visualizar todos los datos asociados a la periodicidad y predicción de compra de los clientes.

CLIENTE	TELÉFONO	ÚLTIMA COMPRA
Fernando Herrera	989045678	11 Ago
Diana Sepulveda	987612390	02 Jul
Nicolás Luna	961552064	Sin cc
Juan Perez	98745678	11 Jul
Filiberto Del Valle	961552064	13 Ago
Victor Palma	954678932	11 Ago

Figura 33: Seguimiento de Clientes – Móvil.

8 PRUEBAS

Las pruebas de software son un conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento de este [34].

Básicamente, su principal función es detectar errores, pero también sirven para medir la calidad del software, integración adecuada de componentes y tecnologías, así como su confiabilidad con el fin de garantizar el cumplimiento de las funcionalidades para generar el valor esperado por el cliente. Dichas funcionalidades están expresadas y especificadas con el Diagrama Casos de Uso en la sección 6.2.

Siguiendo la metodología de desarrollo **Iterativo e Incremental** mencionada en la sección 3.2.1, el desarrollo de la aplicación de este proyecto se dividió en **tres incrementos**. En cada uno de ellos se realizaron pruebas de integración entre las distintas tecnologías usadas. En el último incremento, se realizaron pruebas para evaluar los requisitos que fueron implementados. Cabe destacar que algunos requisitos no fueron implementados por razones que se detallan en la Sección 10.

Es importante mencionar que las pruebas de integración realizadas no están incluidas en este documento, debido a que están totalmente orientadas a factores técnicos. A modo de resumen, Angular implementa sus propios tipos de pruebas de integración que permiten verificar que los componentes funcionen correctamente, mientras que en el entorno de Node.js se ocupan librerías incorporadas para comprobar el flujo de peticiones y respuestas de la API desarrollada.

La aplicación se somete a tres tipos de pruebas que tienen como principal propósito asegurar el cumplimiento de los requerimientos del software (Sección 4.4.1). Dichas pruebas son:

- **Pruebas Funcionales:** son pruebas de **tipo caja negra** basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software [35].

Las pruebas de tipo caja negra, se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero derivar conjuntos de condiciones de entrada que ejercitaran por completo todos los requisitos funcionales de un programa [36].

- **Pruebas de Seguridad:** Son pruebas enfocadas en la seguridad de la aplicación, la cual es un atributo no funcional que influye directamente en la calidad del producto.
- **Pruebas de Usabilidad:** Son pruebas enfocadas en la interacción entre la interfaz del sistema y el usuario, buscan evaluar qué tan intuitivo y amigable es la interfaz del software para los distintos usuarios del mismo. [37]

Por otro lado, es importante mencionar que las pruebas fueron aplicadas a nivel de sistema, es decir, se realizan peticiones desde el Frontend para obtener y consumir los servicios ofrecidos por el Backend. Por lo tanto, en las siguientes secciones se detallan cada una de las pruebas funcionales, de seguridad y usabilidad que fueron aplicadas.

8.1 Especificación de Pruebas

A continuación, se especifican los tipos de pruebas mencionados anteriormente, en donde se da a conocer la característica del sistema a probar, el nivel de prueba, su principal objetivo, enfoque, la técnica de definición de casos de prueba utilizada, las actividades de prueba que fueron aplicadas y los correspondientes criterios de cumplimiento.

8.1.1 Pruebas Funcionales

En la Tabla 43 se especifican las pruebas funcionales del sistema.

Pruebas Funcionales	
Característica a Probar	Funcionalidad
Nivel de Prueba	Sistema (Frontend y Backend)
Objetivo	Verificar y garantizar que el sistema cumpla con los requisitos funcionales especificados.
Enfoque	Caja Negra
Técnica de definición de casos de prueba	Se utilizan datos que sean válidos e inválidos, con el fin de comprobar los siguientes escenarios: <ul style="list-style-type: none"> • Ocurran los resultados esperados cuando se utilizan datos válidos. • Los mensajes de error o notificaciones informan cuando se utilizan datos no válidos.
Actividades de Prueba	<ul style="list-style-type: none"> • Definir los casos de prueba considerando principalmente datos inválidos. • Aplicar los casos de prueba a los elementos especificados. • Registro de resultados. • Corrección de errores.
Criterios de Cumplimiento	<ul style="list-style-type: none"> • El resultado obtenido es el esperado para la funcionalidad probada. • Se han controlado todos los errores.

Tabla 43: Especificación de Pruebas Funcionales.

8.1.2 Pruebas de Seguridad

En la Tabla 44 se especifican las pruebas de seguridad del sistema.

Pruebas de Seguridad	
Característica a Probar	Seguridad
Nivel de Prueba	Sistema (Frontend y Backend)
Objetivo	Verificar y garantizar que el sistema cumpla con los principales requisitos de seguridad, tales como restringir el acceso mediante los roles de los usuarios y seguridad de autenticación.
Enfoque	Caja Negra
Técnica de definición de casos de prueba	Se utilizan datos de usuario que sean válidos e inválidos, con el fin de comprobar los siguientes escenarios: <ul style="list-style-type: none"> • Correcta autenticación del usuario al ingresar datos válidos. • Mensaje de error cuando se utilizan datos inválidos en la autenticación. • Intento de ingreso a una URL cuando un usuario no tiene los permisos en base a su rol en el sistema.
Actividades de Prueba	<ul style="list-style-type: none"> • Autenticación del sistema con distintos usuarios. • Intentar forzar la URL para acceder a los distintos módulos de la aplicación. • Registrar los resultados obtenidos • Corrección de errores.
Criterios de Cumplimiento	<ul style="list-style-type: none"> • Se restringen los accesos en base al rol de cada usuario. • Se permite el acceso solo a usuarios autorizado.

Tabla 44: Especificación de Pruebas de Seguridad.

8.1.3 Pruebas de Usabilidad

En la Tabla 45 se especifican las pruebas de usabilidad del sistema.

Pruebas de Usabilidad	
Característica a Probar	Interfaz y navegación
Nivel de Prueba	Sistema (Frontend y Backend)
Objetivo	Determinar la experiencia de usuario; qué tan intuitiva y usable es la aplicación para los distintos usuarios que interactúan con ella.
Enfoque	Caja Negra
Técnica de definición de casos de prueba	Se solicita a personas que desempeñen las funcionalidades de los actores del sistema (administración y recepcionista). Luego se les realiza una encuesta.
Actividades de Prueba	<ul style="list-style-type: none"> • Se seleccionan voluntarios para que interactúen y utilicen las funcionalidades del sistema • Se realiza una encuesta a dichos voluntarios para determinar qué tan intuitivo es el sistema según su experiencia.
Criterios de Cumplimiento	Los usuarios que interactúan con el sistema no tienen dificultad para utilizar las funcionalidades.

Tabla 45: Especificación de Pruebas de Usabilidad.

8.2 Detalle de las pruebas

La aplicación cuenta con dos módulos; módulo de administración y recepción, que son los **principales actores que interactúan con el sistema**. Por lo tanto, las pruebas que se detallan están asociadas a los **principales casos de uso permitidos para cada actor (módulo)**.

No obstante, en esta sección, solo se presenta **la principal prueba funcional y la principal prueba de usabilidad** para ambos módulos, debido a que están asociadas al caso de uso más utilizado. El resto de las pruebas funcionales se encuentran en el Anexo 1.

8.2.1 Detalle de Prueba Funcional (Módulo Recepción)

En la Tabla 46 se presentan las pruebas realizadas para comprobar la funcionalidad del caso de uso con identificador CU_05, correspondiente a Registrar Pedido (Sección 6.2.2.5). Seleccionado por ser una de las principales funcionalidades en el módulo de recepción.

Prueba: Registrar Pedido (CU_05)				
ID Prueba	PR_01			
Caso de Uso	CU_05			
Módulo / Actor	Recepción			
Objetivo	Verificar el correcto registro de un pedido con un cliente ya registrado previamente.			
Pre - Condición	Usuario autenticado con rol de recepción			
Prueba	Datos de entrada / Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte superior de la navegación (Header o Menú Superior) se selecciona "Registrar Pedido"	La aplicación redirige a la vista para registrar un pedido. Esta vista muestra el campo para buscar un cliente por su número telefónico. Además muestra la opción para registrar un pedido con un nuevo cliente.	La aplicación redirige a la vista correspondiente. Se muestra el campo para buscar un cliente por su número telefónico e incluye la opción para registrar un pedido con un nuevo cliente.	Aprobada
Acción 2	Se ingresa el número o parte del número telefónico de un cliente en el	La aplicación muestra el o los clientes que su número telefónico coincide con el	La aplicación muestra el o los clientes que su número telefónico coincide con el	Aprobada

	campo correspondiente.	número ingresado. Al seleccionar uno, la aplicación muestra el nombre del cliente seleccionado. Luego, la aplicación muestra la opción para registrar un pedido con los datos del cliente seleccionado. Si no encuentra el cliente, muestra un mensaje señalando que el cliente no existe en el sistema.	número ingresado. Al seleccionar uno, la aplicación muestra el nombre del cliente seleccionado y luego se muestra la opción para registrar un pedido con los datos del cliente seleccionado. Si no encuentra el cliente, muestra un mensaje indicando que el cliente no existe en el sistema.	
Datos Válidos	Se ingresan y seleccionan todos los campos solicitados: Nombre de Chofer, Dirección, Cilindros, Tipo de Pago, Observación y Descuento. Una vez ingresados se presiona en el botón "Registrar".	La aplicación registra el pedido en el sistema e informa al usuario mediante un mensaje informando que el pedido se ha registrado con éxito.	La aplicación registra el pedido en el sistema e informa al usuario mediante un mensaje indicando que el pedido se ha registrado con éxito.	Aprobada
Datos No Válidos	Se ingresa algún dato inválido o no se selecciona un dato solicitado: Nombre de Chofer, Dirección, Cilindros, Tipo de Pago, Observación o Descuento. Luego, se presiona en el botón "Registrar".	La aplicación muestra un mensaje alertando que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No se registra el pedido.	La aplicación muestra un mensaje de alerta indicando que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No se registra el pedido.	Aprobada

Tabla 46: Prueba de CU_05.

8.2.2 Detalle Prueba Funcional (Módulo Administrador)

En la Tabla 47 se presentan las pruebas realizadas para comprobar la funcionalidad del caso de uso con identificador CU_26, correspondiente a Seguimiento de Clientes (Sección 6.2.2.27). Seleccionado por ser una de las principales funcionalidades en el módulo de administración.

Prueba: Seguimiento de Clientes (CU_26)				
ID Prueba	PR_02			
Caso de Uso	CU_26			
Módulo / Actor	Administración			
Objetivo	Verificar que el seguimiento de clientes se realice correctamente.			
Pre - Condición	Usuario autenticado con rol de administración.			
Prueba	Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte izquierda de la navegación de la aplicación (Sidebar o Menú Lateral), se selecciona "Seguimiento"	La aplicación redirige a la vista de seguimiento de clientes mostrando un mensaje de "Cargando Seguimiento". Luego, la vista muestra todos los clientes registrados con sus datos e información de seguimiento: Última Compra, Periodicidad de Compra, Estado de Predicción y Acciones.	La aplicación redirige a la vista de seguimiento de clientes mostrando un mensaje de "Cargando Seguimiento". La vista muestra todos los clientes registrados con sus datos e información de seguimiento: Última Compra, Periodicidad de Compra, Estado de Predicción y Acciones.	Aprobada

Tabla 47: Pruebas de Seguimiento de Clientes.

8.2.3 Detalle de Prueba de Seguridad.

En la Tabla 48 se presentan las pruebas de seguridad realizadas para comprobar la seguridad en la autenticación del usuario. Funcionalidad que está asociada al CU_01, correspondiente a Iniciar Sesión (Sección 6.2.2.1). Las pruebas fueron aplicadas a ambos módulos (Recepción y Administración).

Prueba: Iniciar Sesión (CU_01)				
ID Prueba	PR_03			
Caso de Uso	CU_01			
Módulo / Actor	Administración, Recepción			
Objetivo	Verificar que el Inicio de Sesión sea correcto y autentique solo a los usuarios autorizados.			
Pre - Condición	No aplica.			
Prueba	Datos de Entrada/ Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Datos Válidos	El usuario de recepción accede a la página de Iniciar Sesión. Luego, ingresa un Rut (con formato 11111111-1) y Contraseña válidos. Luego, presiona en el botón de "Iniciar Sesión"	La aplicación autentica al usuario con el rol de recepción y se carga el módulo con todas las navegaciones y opciones correspondientes al rol de recepción; Estadísticas Semanales, Pedidos, Ventas, Clientes, Direcciones y Cilindros.	La aplicación autentica al usuario con el rol de recepción y se carga el módulo con todas las navegaciones y opciones correspondientes al rol de recepción; Estadísticas Semanales, Pedidos, Ventas, Clientes, Direcciones y Cilindros.	Aprobada
	El usuario de administración accede a la página de Iniciar Sesión. Luego, ingresa un Rut (con formato 11111111-1) y Contraseña válidos. Luego, presiona en el botón de "Iniciar Sesión"	La aplicación autentica al usuario con el rol de administración y se carga el módulo con todas las navegaciones y opciones correspondientes al rol de administración;	La aplicación autentica al usuario con el rol de administración y se carga el módulo con todas las navegaciones y opciones correspondientes al rol de administración;	Aprobada

		Estadísticas Mensuales, Clientes, Empleados, Seguimiento, Estadísticas y Metas. Además muestra un mensaje que indica que se ha iniciado de sesión	Estadísticas Mensuales, Clientes, Empleados, Seguimiento, Estadísticas y Metas. Además muestra un mensaje que indica que se ha iniciado de sesión	
Datos No Válidos	El usuario accede a la página de Iniciar Sesión y no ingresa Rut ni Contraseña, luego presiona en "Iniciar Sesión"	La aplicación muestra un mensaje de alerta al usuario indicando que debe ingresar un Rut y Contraseña. No autentica al usuario.	La aplicación muestra un mensaje alertando al usuario que debe ingresar un Rut y Contraseña. No autentica al usuario.	Aprobada
	El usuario accede a la página de Iniciar Sesión e ingresa un Rut con el formato incorrecto, luego presiona en "Iniciar Sesión"	La aplicación muestra un mensaje de error al usuario indicando que debe ingresar un Rut con el formato 11111111-1. No autentica al usuario.	La aplicación muestra un mensaje alertando al usuario que debe ingresar un Rut con el formato 11111111-1. No autentica al usuario.	Aprobada
	El usuario accede a la página de Iniciar Sesión e ingresa un Rut o Contraseña incorrecta, luego presiona en "Iniciar Sesión"	La aplicación muestra un mensaje de error al usuario indicando que el <u>Rut o la Contraseña ingresada son incorrectas</u> . No autentica al usuario.	La aplicación muestra un mensaje de error al usuario indicando que el <u>Rut o la Contraseña ingresada son incorrectas</u> . No autentica al usuario.	Aprobada

Tabla 48: Prueba de CU_01.

8.2.4 Detalle de Pruebas de Usabilidad.

Las pruebas de usabilidad de este proyecto se realizaron en las últimas semanas del tercer y último incremento del desarrollo de la aplicación. Estas pruebas buscan asegurar una buena y grata experiencia de usuario en el sistema desarrollado.

En esta sección, se presentan las pruebas de usabilidad realizadas en este proyecto. Se seleccionaron tres personas voluntarias que simularon los roles de la distribuidora de gas; un administrador y dos recepcionistas. En el escenario real, se cuenta con más administradores y recepcionistas, por lo que las pruebas de usabilidad con un mayor número de usuarios quedan como trabajo futuro para este proyecto (Razones expuestas en la Sección 10).

Es importante mencionar que el usuario de administración que se creó para estas pruebas fue registrado mediante la funcionalidad de “Crear Usuario” por el usuario de administración predeterminado de la aplicación. El detalle de la prueba de dicha funcionalidad se encuentra en el Anexo 1 de la sección 12 .

Luego de efectuadas las pruebas con los participantes, se les realizó una encuesta con seis preguntas (disponible en el Anexo 2 de la sección 12), que permite determinar el nivel de Acuerdo o Desacuerdo de los participantes:

1. La estructura de la interfaz de la aplicación es intuitiva para realizar las funcionalidades.
2. Es fácil comprender los datos que se deben ingresar en los formularios.
3. La aplicación muestra mensajes y alertas con mensajes descriptivos sobre las funcionalidades realizadas.
4. Los colores de la interfaz son agradables y representan a la empresa.
5. La navegación es visible y resulta sencillo encontrar las opciones para realizar las funcionalidades.
6. Los botones de la aplicación tienen nombres descriptivos que dan a entender la funcionalidad que realizan.

A continuación, en la Tabla 49 se muestran los resultados obtenidos de la encuesta.

Resultados Encuesta Prueba de Usabilidad					
Pregunta	Número de Personas				
	Muy de Acuerdo	De Acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Muy en desacuerdo
1	3				
2	3				
3	3				
4	3				
5	3				

Tabla 49: Resultado Encuesta de Usabilidad.

8.2.4.1 Detalle de Pruebas de Usabilidad (Recepción)

Para las personas que simularon el **rol de recepcionista** en el sistema, la prueba consistió en los siguientes diez pasos:

1. Iniciar Sesión.
2. Visualizar Estadísticas Semanales
3. Registrar un Cliente
4. Crear Dirección
5. Registrar un Pedido con el cliente registrado previamente
6. Visualizar Pedidos en Curso
7. Confirmar Pedido
8. Registrar un Pedido con nuevo Cliente
9. Visualizar Pedidos en Curso
10. Cancelar Pedido

A continuación, en la Tabla 50 se muestran los resultados obtenidos de la prueba realizada con **dos personas** que simularon dicho rol.

Funcionalidades realizadas	Logrado	No Logrado	Observaciones Generales
Iniciar Sesión	2	0	
Visualizar Estadísticas Semanales	2		
Registrar un Cliente	2	0	
Crear Dirección	2	0	Al registrar una dirección, deberían aparecer sugerencias de similitud en base a direcciones ya registradas, con el fin de autocompletar una calle, una villa o un sector.
Registrar un Pedido con el cliente registrado previamente	2	0	
Visualizar Pedidos en Curso	2	0	
Confirmar Pedido	2	0	En el mensaje de la confirmación de un pedido, debería mostrarse una opción para ver los detalles de la venta asociada.
Registrar un Pedido con un Nuevo Cliente	2	0	
Visualizar Pedidos en Curso	2	0	
Cancelar Pedido	2	0	

Tabla 50: Resultados Pruebas de Usabilidad (Recepción).

8.2.4.2 Detalle de Pruebas de Usabilidad (Administración)

Para la persona que simuló el **rol de administración** en el sistema, la prueba consistió en los siguientes ocho pasos:

1. Iniciar Sesión.
2. Visualizar Estadísticas Mensuales
3. Crear Empleado (Recepcionistas)
4. Visualizar Empleados
5. Visualizar Clientes
6. Visualizar a Seguimiento de Clientes
7. Ordenar Clientes en Seguimiento de Clientes.
8. Crear Meta

A continuación, en la Tabla 51 se muestran los resultados obtenidos de las pruebas realizadas con **una persona** que simuló dicho rol.

Funcionalidades realizadas	Logrado	No Logrado	Observaciones Generales
Iniciar Sesión	1	0	
Visualizar Estadísticas Mensuales	1	0	
Crear Empleados (Recepcionista)	1	0	
Visualizar Empleados	1	0	
Visualizar Clientes	1	0	
Visualizar Seguimiento de Clientes.	1	0	Funcionalidad de buscar un cliente en específico.
Ordenar Clientes en Seguimiento de Clientes.	1	0	
Crear Meta	1	0	

Tabla 51: Resultados Pruebas de Usabilidad (Administración).

8.3 Conclusiones de Pruebas Efectuadas

La ejecución de pruebas durante el desarrollo de la aplicación de este proyecto permitió detectar errores y problemas en la implementación de los casos de uso, logrando desarrollar un software de calidad que ofrezca el valor esperado por el cliente y garantizando características como una correcta funcionalidad, una seguridad robusta y una grata experiencia de usuario.

La implementación de pruebas funcionales permitió detectar fallas específicas, como validación de formularios en las distintas vistas que implementan funcionalidades de creación, como lo es el CU_05 de Registrar Pedido y CU_21 de crear un empleado.

Las pruebas de seguridad es uno de los factores de mayor importancia ya que permiten evaluar la confidencialidad, integridad y disponibilidad del flujo de información de la aplicación. En la prueba de seguridad enfocada en la autenticación del usuario (PR_03), se obtuvieron resultados satisfactorios, ya que se logra garantizar que la aplicación solo autentica a usuarios autorizados, es decir, que hayan sido previamente registrados por un administrador. Además, cabe destacar que la realización de extensas pruebas de seguridad es costosa en tiempo, pero se compensa con los grandes beneficios que estas otorgan, por lo tanto, estas pruebas no deben ser dejadas de lado y esperar a que se produzcan serios problemas de seguridad para recién implementarlas.

Por otro lado, las pruebas de usabilidad realizadas ayudaron a mejorar aspectos orientados a la experiencia de usuario del sistema. Mediante la encuesta realizada, se determinó que el sistema cumple satisfactoriamente con las características de usabilidad deseadas, mencionadas en la sección 4.3.1.

Es de vital importancia detectar fallas y errores en una etapa temprana del desarrollo, esto permite corregirlos rápidamente y disminuir el impacto que puedan tener en un futuro, por lo que el proceso de pruebas no debe ser realizado únicamente en la etapa final del desarrollo, sino que debe ser un proceso continuo integrado en la construcción del software, tal como se realizó en este proyecto con pruebas de integración con las tecnologías ocupadas.

En conclusión, se obtienen resultados positivos en los distintos tipos y casos de pruebas a los que se sometió la aplicación. El resultado satisfactorio de estas pruebas permite una mejora importante para la entrega final del proyecto, por lo tanto, el software desarrollado garantiza que sus funcionalidades sean usadas de forma correcta, segura e intuitiva.

9 CONCLUSIONES

Desde el primer día del desarrollo de este proyecto, se han obtenido aprendizajes y experiencias de manera inexorable, debido a la puesta en práctica de los conocimientos de distintas áreas, que van desde el modelado de datos hasta el diseño de interfaces de usuarios, siguiendo patrones de desarrollo y estándares de ingeniería de software que han permitido construir un Sistema de Gestión de Clientes y Pedidos para la Empresa Gas Chillán.

La metodología iterativa incremental fue una excelente decisión para este tipo de proyecto, debido a que esta metodología aporta un enfoque ágil para aplicaciones que son construidas por solo un desarrollador de software, obteniendo retroalimentación constantemente de los incrementos que van proporcionando un resultado parcial del producto final. No obstante, uno de los factores que presentó mayor dificultad fue definir correctamente los tiempos de desarrollo del proyecto debido al nivel principiante en gestión de proyectos, aún así se logró finalizar el proyecto con la mayoría de los requisitos que fueron definidos en las etapas iniciales.

Respecto a las arquitecturas elegidas para el desarrollo del sistema fueron acertadas, debido a que la arquitectura implementada en el Frontend se enfoca en separar lo máximo posible la lógica de negocio con la lógica de interfaces de usuario, lo cual permite un código mucho más mantenible y escalable a través del tiempo, considerando todos los trabajos futuros que serán desarrollados. Por otro lado, la arquitectura de tres capas del Backend, también proporciona una aplicación altamente escalable, debido a que el código queda organizado en capas de datos, servicios y controladores, facilitando su lectura y comprensión para permitir que la aplicación pueda crecer en funcionalidad y en números de servicios en los trabajos futuros.

Gracias a las pruebas de usabilidad que se realizaron con las personas voluntarias, a las pruebas funcionales y a las de seguridad que fueron aplicadas en el sistema, se obtuvieron resultados positivos que indican que el software desarrollado es altamente funcional, intuitivo y seguro de usar. Por lo tanto, el usuario de Gas Chillán que utilice e interactúe con este sistema, puede confiar plenamente en las funcionalidades que ofrece, ya que garantizan la confidencialidad, integridad y disponibilidad del flujo de información.

Desde un punto de vista personal, el desarrollo de este proyecto ha sido totalmente enriquecedor, debido a que se han puesto a prueba todos los conocimientos obtenidos a lo largo del transcurso de la carrera de Ingeniería Civil en Informática, especialmente en el área de Ingeniería de Software, obteniendo nuevos conocimientos como arquitecturas, metodologías y estándares de desarrollo que permiten ofrecer un producto de calidad en la industria del software.

Finalmente, se concluye que el proyecto cumple con los objetivos propuestos de manera exitosa, en donde se obtiene una aplicación web que implementa la mayoría de los requisitos planteados, permitiendo que la Empresa Gas Chillán pueda modernizar y reinventar su sistema con tecnologías de información y adaptarse a las nuevas demandas en un mercado que está lleno de desafíos y constantes cambios. Además, se logra agilizar los principales procesos tanto administrativos como operativos gracias al valor que ofrece la aplicación desarrollada, la cual también permite garantizar y mantener su compromiso de ofrecer a sus clientes un servicio rápido y de calidad.

10 TRABAJOS FUTUROS

No es de menor importancia destacar que todo el proceso de desarrollo de este proyecto se realizó durante la propagación de la pandemia por COVID-19, que impactó tanto a nivel nacional como mundial, generando cambios radicales en las modalidades de trabajo, teniendo que adaptarse a nuevas normalidades. El principal impacto de la contingencia en este proyecto es que se dificultó la definición de tiempos de desarrollo y cumplimiento exacto de entregas, debido a que fueron surgiendo problemas personales y dificultades generadas por dicha contingencia sanitaria. No obstante, se ha aprendido una gran lección relacionada con la definición de riesgos en la gestión de un proyecto de software.

Afortunadamente, se pudo contar con la comprensión de docentes y también por parte de la Empresa Gas Chillán para dar flexibilidad al desarrollo del proyecto en sí. De todos modos, se ha logrado cumplir con los requisitos más importantes de la aplicación.

Debido a lo expresado anteriormente, se debe señalar que no se ha logrado cumplir con algunos de los requisitos funcionales de la aplicación, principalmente por factores de tiempo, considerando que el proyecto fue realizado de forma individual y además por la carga académica de este particular semestre. Por lo tanto, estos requisitos quedarán considerados como trabajos futuros, los cuales se detallan a continuación, haciendo referencia a los casos de uso asociados (definidos en la sección 0):

1. **CU_08:** Editar Pedido
2. **CU_15:** Editar Venta
3. **CU_20:** Actualizar Cliente
4. **CU_25:** Actualizar Usuario
5. **CU_29:** Enviar Correo Masivo a Clientes
6. **CU_30:** Estadísticas de Venta
7. **CU_31:** Generar Informe de Ventas

Por otro lado, también quedan como trabajos futuros las funcionalidades adicionales que darán una mayor robustez al sistema, estas son:

1. Desarrollar un sitio web en donde los clientes puedan realizar los pedidos de forma online. Este sitio deberá interactuar con el sistema y mostrar a recepción todos los pedidos que se hayan realizado por este medio. Además, la opción de un chat en tiempo real entre un cliente y la recepción de la empresa mediante este sitio web.
2. Desarrollar un módulo para los usuarios que tengan el rol de Chofer, el cual les permitirá confirmar los pedidos una vez que hayan sido entregados físicamente. Esto permitirá agilizar el proceso de “rendiciones” de los choferes, el cual es un proceso de bastante importancia para las decisiones administrativas de la empresa.
3. Desarrollar la funcionalidad para el control de bodega, el cual permitirá registrar y controlar las existencias de los cilindros que posee la empresa.
4. Desarrollar la funcionalidad de venta en oficina, que permitirá registrar todos los pedidos que se realicen de forma presencial.

11 BIBLIOGRAFÍA

- [1] P. Roger, Ingeniería de Software, Un enfoque Práctico, 2010.
- [2] Angular, «Angular,» [En línea]. Available: <https://angular.io/>. [Último acceso: 08 2020].
- [3] Express, «Express,» [En línea]. Available: <https://expressjs.com/>. [Último acceso: 08 2020].
- [4] Node.js, «Node.js,» [En línea]. Available: <https://nodejs.org/>. [Último acceso: 08 2020].
- [5] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/es/what-is-mongodb>. [Último acceso: 08 2020].
- [6] Hostinger, «Hostinger,» [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-css/>. [Último acceso: 08 2020].
- [7] D. Web, «Desarrolloweb6,» [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-html.html>. [Último acceso: 08 2020].
- [8] Postman, «Postman,» [En línea]. Available: <https://www.postman.com>. [Último acceso: 08 2020].
- [9] Microsoft, «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/docs/editor/whyvscode>. [Último acceso: 08 2020].
- [10] Google, «Firebase,» [En línea]. Available: <https://firebase.google.com/docs/hosting>. [Último acceso: 08 2020].
- [11] Heroku, «Heroku,» [En línea]. Available: <https://www.heroku.com/what>. [Último acceso: 08 2020].
- [12] MongoDB, «Mongo Atlas,» [En línea]. Available: <https://docs.atlas.mongodb.com/>. [Último acceso: 08 2020].
- [13] R. Pressman, Ingeniería de Software, Un Enfoque Práctico, 2010.
- [14] Platzi, «Platzi,» [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/#:~:text=Frontend%20es%20la%20parte%20de,corre%20del%20lado%20del%20servidor..> [Último acceso: 06 2020].
- [15] Wikipedia, «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/TypeScript>. [Último acceso: 07 2020].
- [16] O. Webinars, «Open Webinars,» [En línea]. Available: <https://openwebinars.net/blog/la-arquitectura-mvvm-y-sus-componentes/>. [Último acceso: 08 2020].
- [17] Angular, «Angular,» 01 Agosto 2020. [En línea]. Available: <https://angular.io/docs>.
- [18] Udemy, «Udemy,» 01 Agosto 2020. [En línea]. Available: <https://www.udemy.com/course/nodejs-express-mongodb-bootcamp/learn/lecture/15065564?start=300#overview>.
- [19] Wikipedia, «Separation of concerns,» [En línea]. Available: https://en.wikipedia.org/wiki/Separation_of_concerns#:~:text=In%20computer%20s

- ciencia%2C%20separation%20of,code%20of%20a%20computer%20program..
[Último acceso: 08 2020].
- [20] bytearcher, «bytearcher,» [En línea]. Available: <https://bytearcher.com/articles/node-project-structure/>. [Último acceso: 08 2020].
- [21] bytearcher, «bytearcher,» [En línea]. Available: <https://bytearcher.com/articles/node-project-structure/>. [Último acceso: 08 2020].
- [22] dev.to, «dev.to,» [En línea]. Available: <https://dev.to/santypk4/bulletproof-node-js-project-architecture-4epf>.
- [23] S. Quinn, «dev.to,» 01 Agosto 2020. [En línea]. Available: <https://dev.to/santypk4/bulletproof-node-js-project-architecture-4epf#folder>.
- [24] semrush, «semrush,» [En línea]. Available: <https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>. [Último acceso: 08 2020].
- [25] yEd, «Yed,» 01 Agosto 2020. [En línea]. Available: <https://www.yworks.com/products/yed>.
- [26] Bizagi, «Bizagi,» 01 Agosto 2020. [En línea]. Available: <https://www.bizagi.com/>.
- [27] Draw.io, «App Diagrams,» 01 Agosto 2020. [En línea]. Available: <https://app.diagrams.net/>.
- [28] LucidChart, «LucidChart,» [En línea]. Available: lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml. [Último acceso: 08 2020].
- [29] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/es/nosql-explained>. [Último acceso: 07 2020].
- [30] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/scale/nosql-databases-pros-and-cons>. [Último acceso: 07 2020].
- [31] agiratech, «agiratech,» [En línea]. Available: <https://www.agiratech.com/blog/the-key-differences-between-sql-and-nosql-database>. [Último acceso: 07 2020].
- [32] Datsen, «Datsen,» Julio 2020. [En línea]. Available: <https://www.datensen.com/mongodb-design-tool.html>.
- [33] LucidChart, «LucidChart,» Julio 2020. [En línea]. Available: <https://www.lucidchart.com/>.
- [34] pmoinformatica, «pmoinformatica,» 08 2020. [En línea]. Available: <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
- [35] Wikipedia, «Wikipedia,» 25 Noviembre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Pruebas_funcionales.
- [36] R. Pressman, Ingeniería del Software, Un Enfoque Práctico, 2010.
- [37] Hostgator, 08 Agosto 2020. [En línea]. Available: <https://www.hostgator.mx/blog/que-es-un-test-de-usabilidad/>.
- [38] Mongo, «beginnersbook,» Septiembre 2017. [En línea]. Available: <https://beginnersbook.com/2017/09/mapping-relational-databases-to-mongodb/>. [Último acceso: Julio 2020].
- [39] MongoDB, «MongoDB Manual,» 01 Agosto 2020. [En línea]. Available: <https://docs.mongodb.com/manual/core/document/>.

[40] MongoDB, «MongoDB Manual,» Julio 2020. [En línea]. Available: <https://docs.mongodb.com/manual/reference/database-references/>.

12 ANEXOS

Anexo 1: Pruebas Funcionales

Este anexo contiene las pruebas funcionales de los principales casos de uso del sistema de gestión de clientes y pedidos.

1.1 Detalle de Prueba Funcional (Módulo Administración): Registrar Usuario

En la Tabla 52, se presentan las pruebas realizadas para comprobar la funcionalidad del caso de uso implementado CU_22. Cabe destacar que el recurso “Usuario” utilizado en esta funcionalidad se representa como Empleado, con el propósito de que el administrador pueda comprender rápidamente que hace referencia a los empleados de la empresa.

Prueba: Registrar Usuario (CU_22)				
ID Prueba	PR_04			
Caso de Uso	CU_22			
Módulo / Actor	Administración			
Objetivo	Verificar el correcto registro de un usuario/empleado en el sistema			
Pre - Condición	Usuario autenticado con rol de administrador.			
Prueba	Datos de entrada / Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte lateral de la navegación de la aplicación (Sidebar o Menú Lateral), se selecciona “Empleados”	La aplicación redirige a la vista que permite visualizar una lista con todos los empleados. Además, muestra un botón con el texto “Crear Empleado”.	La aplicación redirige a la vista que permite visualizar todos los empleados. Además, muestra un botón con el texto “Crear Empleado”.	Aprobada

Acción 2	Se presiona en el botón de “Crear Empleado”	La aplicación redirige a la vista que permite ingresar todos los datos para crear un nuevo usuario/empleador.	La aplicación redirige a la vista que permite ingresar todos los datos para crear un nuevo usuario/empleador.	Aprobada
Datos Válidos	Se ingresan y seleccionan todos los campos solicitados: Nombres, Apellidos, Rol, Rut, Contraseña y Email Una vez ingresados se presiona en el botón “Registrar Empleado”.	La aplicación registra el empleado en el sistema e informa al usuario mediante un mensaje de que el empleado se ha registrado con éxito.	La aplicación registra el empleado en el sistema e informa al usuario mediante un mensaje de que se ha registrado con éxito.	Aprobada
Datos No Válidos	Se ingresa algún dato inválido y/o no se selecciona un dato solicitado: Nombres, Apellidos, Rol, Rut, Contraseña o Email Luego, se presiona en el botón “Registrar Empleado”.	La aplicación muestra un mensaje de alerta que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No registra al empleado.	La aplicación muestra un mensaje de alerta que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No registra al empleado.	Aprobada

Tabla 52: Prueba de CU_22.

1.2 Detalle de Prueba Funcional (Módulo Recepción): Crear Cliente

En la Tabla 53, se presentan las pruebas realizadas para comprobar la funcionalidad del caso de uso implementado CU_17.

Prueba: Crear Cliente (CU_22)				
ID Prueba	PR_05			
Caso de Uso	CU_17			
Módulo / Actor	Cliente, Administración			
Objetivo	Verificar la correcta creación de un cliente en el sistema			
Pre - Condición	Usuario autenticado con rol de recepción o administrador.			
Prueba	Datos de entrada / Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte lateral de la navegación de la aplicación (Sidebar o Menú Lateral), se selecciona "Clientes"	La aplicación redirige a la vista que permite visualizar una lista con todos los clientes. Además, muestra un botón con el texto "Crear Cliente".	La aplicación redirige a la vista que permite visualizar todos los clientes. Además, muestra un botón con el texto "Crear Cliente".	Aprobada
Acción 2	Se presiona en el botón de "Crear Cliente"	La aplicación redirige a la vista que permite ingresar todos los datos para crear un nuevo cliente.	La aplicación redirige a la vista que permite ingresar todos los datos para crear un nuevo cliente	Aprobada
Datos Válidos	Se ingresan y seleccionan todos los campos solicitados: Nombre, Apellido, Teléfono, Tipo Cliente y Email Una vez ingresados se presiona en el botón "Crear Cliente".	La aplicación registra el cliente en el sistema e informa al usuario mediante un mensaje de que el cliente se ha registrado con éxito.	La aplicación registra el cliente en el sistema e informa al usuario mediante un mensaje de que se ha registrado con éxito.	Aprobada
Datos No Válidos	Se ingresa algún dato inválido y/o no	La aplicación muestra un mensaje	La aplicación muestra un mensaje	Aprobada

	<p>se selecciona un dato solicitado:</p> <p>Nombre, Apellido, Teléfono, Tipo Cliente o Email</p> <p>Luego, se presiona en el botón "Crear Cliente".</p>	<p>de alerta que no se han ingresado los campos solicitados.</p> <p>Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado.</p> <p>No registra el cliente.</p>	<p>de alerta que no se han ingresado los campos solicitados.</p> <p>Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado.</p> <p>No registra el cliente.</p>	
--	---	---	---	--

Tabla 53: Prueba de CU_17.

1.4 Detalle de Prueba Funcional (Módulo Recepción): Confirmar Pedido

En la Tabla 54, se presentan las pruebas realizadas para comprobar la funcionalidad del caso de uso implementado CU_11 (Confirmar Pedido).

Prueba: Crear Dirección				
ID Prueba	PR_06			
Caso de Uso	CU_11			
Módulo / Actor	Recepción			
Objetivo	Verificar la correcta confirmación de un pedido en el sistema.			
Pre - Condición	Usuario autenticado con rol de recepción.			
Prueba	Datos de entrada / Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte superior de la navegación (Header o Menú Superior), se selecciona el ícono con figura de camión, el cual representa los "Pedidos en Curso"	<p>La aplicación redirige a la vista que permite visualizar una lista con todos los pedidos en curso. En cada ítem de esta lista, se muestran dos botones, uno para confirmar (con ícono check) y otro para cancelar un pedido (con un ícono cruz).</p> <p>Además, en la parte superior de esta lista muestra dos botones:</p> <ul style="list-style-type: none"> - Confirmar Todos - Cancelar Todos. <p>Se recomienda revisar la sección 7.7.3.7 para comprender mejor la muestra de estos botones.</p>	<p>La aplicación redirige a la vista que permite visualizar una lista con todos los pedidos en curso. En cada ítem de esta lista, se muestran dos botones, uno para confirmar y otro para cancelar un pedido.</p> <p>Además, en la parte superior de esta lista muestra dos botones:</p> <ul style="list-style-type: none"> - Confirmar Todos - Cancelar Todos. 	Aprobada
Acción 2	Se presiona en el botón para confirmar (ícono	La aplicación actualiza el pedido en el sistema e	La aplicación actualiza el pedido en el sistema e	Aprobada

	check) en el ítem de la lista.	informa al usuario mediante un mensaje de que se ha confirmado el pedido.	informa al usuario mediante un mensaje de que se ha confirmado el pedido.	
Acción 3	Se presiona en el botón para cancelar (ícono de cruz) en el ítem de la lista.	La aplicación realiza una eliminación lógica del pedido en el sistema e informa al usuario mediante un mensaje de que se ha cancelado el pedido.	La aplicación realiza una eliminación lógica del pedido en el sistema e informa al usuario mediante un mensaje de que se ha cancelado el pedido.	Aprobada
Acción 4	Se presiona en el botón "Confirmar Todos" para confirmar todos los pedidos en curso.	La aplicación pregunta al usuario mediante una vista emergente si realmente desea confirmar todos los pedidos.	La aplicación pregunta al usuario mediante una vista emergente si realmente desea confirmar todos los pedidos.	Aprobada
Acción 4.1	Se presiona en el botón "Aceptar" de la ventana emergente de la Acción 4.	La aplicación actualiza todos los pedidos e informa al usuario mediante un mensaje de que se han confirmado todos los pedidos.	La aplicación actualizado todos los pedidos e informa al usuario mediante un mensaje de que se han confirmado todos los pedidos.	Aprobada
Acción 4.1	Se presiona en el botón "Cancelar" de la ventana emergente de la Acción 4.	La aplicación quita la vista emergente. No confirma todos los pedidos.	La aplicación La aplicación quita la vista emergente. No confirma todos los pedidos.	Aprobada
Acción 5	Se presiona en el botón "Cancelar Todos" para cancelar todos los pedidos en curso.	La aplicación pregunta al usuario mediante una vista emergente si realmente desea cancelar todos los pedidos.	La aplicación pregunta al usuario mediante una vista emergente si realmente desea cancelar todos los pedidos.	Aprobada
Acción 5.1	Se presiona en el botón "Aceptar" de la ventana emergente de la Acción 5.	La aplicación realiza una eliminación lógica de todos los pedidos e informa al usuario mediante un mensaje de que se han cancelado todos los pedidos.	La aplicación realiza una eliminación lógica de todos los pedidos e informa al usuario mediante un mensaje de que se han cancelado todos los pedidos.	Aprobada

Acción 5.1	Se presiona en el botón "Cancelar" de la ventana emergente de la Acción 5.	La aplicación quita la vista emergente. No cancela todos los pedidos.	La aplicación La aplicación quita la vista emergente. No cancela todos los pedidos.	Aprobada
-------------------	--	--	--	----------

Tabla 54: Prueba de CU_11.

1.5 Detalle de Prueba Funcional (Módulo Recepción): Crear Dirección

En la Tabla 55, se presentan las pruebas realizadas para comprobar la funcionalidad de Crear Dirección, que no está asociado a ningún caso de uso, pero es utilizado para asociar direcciones a los pedidos y clientes del sistema.

Prueba: Crear Dirección				
ID Prueba	PR_07			
Caso de Uso	No Aplica			
Módulo / Actor	Recepción			
Objetivo	Verificar la correcta creación de una dirección en el sistema			
Pre - Condición	Usuario autenticado con rol de recepción.			
Prueba	Datos de entrada / Acciones realizadas	Resultado Esperado	Resultado Obtenido	Evaluación
Acción 1	En la parte lateral de la navegación de la aplicación (Sidebar o Menú Lateral), se selecciona "Direcciones"	La aplicación redirige a la vista que permite visualizar todas las direcciones. Además, muestra un botón con el texto "Crear Dirección".	La aplicación redirige a la vista que permite visualizar todas las direcciones. Además, muestra un botón con el texto "Crear Dirección".	Aprobada
Acción 2	Se presiona en el botón de "Crear Dirección"	La aplicación muestra un modal (vista flotante) que permite ingresar todos los datos para crear una nueva dirección.	La aplicación muestra un modal (vista flotante) que permite ingresar todos los datos para crear una nueva dirección.	Aprobada
Datos Válidos	Se ingresan todos los campos solicitados: Calle, Villa, Sector y Número. Siendo opcional el campo "Departamento" Una vez ingresados se presiona en el botón "Crear Dirección".	La aplicación registra la dirección en el sistema e informa al usuario mediante un mensaje de que la dirección se ha registrado con éxito.	La aplicación registra la dirección en el sistema e informa al usuario mediante un mensaje de que se ha registrado con éxito.	Aprobada

<p>Datos No Válidos</p>	<p>Se ingresa algún dato inválido: Calle, Villa, Sector, Número o Departamento Luego, se presiona en el botón “Crear Dirección”.</p>	<p>La aplicación muestra un mensaje de alerta que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No registra la dirección.</p>	<p>La aplicación muestra un mensaje de alerta que no se han ingresado los campos solicitados. Además, remarca con un color llamativo el campo erróneo o que no fue ingresado / seleccionado. No registra la dirección.</p>	<p>Aprobada</p>
--------------------------------	--	--	--	-----------------

Tabla 55: Prueba de Crear Dirección.

Anexo 2: Encuesta Evaluación Pruebas de Usabilidad

Seleccione una opción en cada pregunta:

1. La estructura de la interfaz de la aplicación es intuitiva para realizar las funcionalidades.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.

2. Es fácil comprender los datos que se deben ingresar en los formularios.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.

3. La aplicación muestra mensajes y alertas con mensajes descriptivos sobre las funcionalidades realizadas.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.

4. Los colores de la interfaz son agradables y representan a la empresa.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.

5. La navegación es visible y resulta sencillo encontrar las opciones para realizar las funcionalidades.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.

6. Los botones de la aplicación tienen nombres descriptivos que dan a entender la funcionalidad que realizan.
 - a) Muy de acuerdo.
 - b) De acuerdo.
 - c) Ni de acuerdo ni desacuerdo.
 - d) En desacuerdo.
 - e) Muy en desacuerdo.