

UNIVERSIDAD PÚBLICA DE EL ALTO  
CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE PRODUCTOS Y PAGOS  
PARA MEJORAR LA PROMOCIÓN DE LÍDERES Y ASESORAS”

CASO: ZERMAT LA PAZ

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

Mención: GESTIÓN Y PRODUCCIÓN

Postulante: Univ. Edna Andrea Eulate Eugenio

Tutor Metodológico: M. Sc. Lic. Ing. Enrique Flores Baltazar

Tutor Revisor: Lic. Fredy Alanoca Coareti

Tutor Especialista: M. Sc. Lic. Ing. Ramiro Kantuta Limachi

EL ALTO – BOLIVIA

2023



Editar con WPS Office

## DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Edna Andrea Eulate Eugenio estudiante con C.I. 4976159 LP mediante la presente declaro de manera pública que la propuesta del PROYECTO DE GRADO titulada "SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE ENTREGA DE PRODUCTOS Y RESPECTIVO PAGO PARA MEJORAR LA PROMOCIÓN DE LÍDERES Y ASESORAS" es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el PROYECTO DE GRADO haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre del 2023.

Edna Andrea Eulate Eugenio  
C.I. 4976159 LP  
e-mail: ednaandreaee80@gmail.com



Editar con WPS Office

## DEDICATORIA

*Este proyecto de grado se la dedico principalmente a Dios, quien supo guiarme por el buen camino, darme fuerza para seguir adelante, enseñándome a encarar las adversidades sin perder nunca la dignidad, ni desfallecer en el intento.*

*A mis padres Andrés y Ana, mis héroes, por haberme forjado como la persona que soy, por su apoyo incondicional durante todos estos años, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi capacidad.*

*A mis hermanos y sobrinos, por sus palabras y su compañía, quienes han creído en mí siempre, dándome ejemplo de superación, humildad y sacrificio.*

*A mi esposo, por su inmenso amor, quien ha sido mi mayor motivación para nunca rendirme en los estudios.*

*A mis hijitos hermosos, quiero dejarles la legacia de hacer obras constructivas y quiero que tengan siempre en su corazón la fortaleza para luchar por sus convicciones.*

*A todos mis amigos y compañeros por su ayuda desinteresada durante el transcurso de nuestro tiempo en la carrera.*



## AGRADECIMIENTOS

*Agradezco a Dios por darme siempre fuerzas para continuar en lo adverso, por guiarme en el camino de lo prudente y darme sabiduría para mejorar día a día mi quehacer profesional.*

*A mi Universidad, por haberme aceptado ser parte de ella y abierto las puertas de su seno científico para poder estudiar mi carrera.*

*Al M. Sc. Lic. Ing. Enrique Flores Baltazar, por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento.*

*Al Lic. Fredy Alanoca Coareti, por sus conocimientos, su orientación, quien ha inculcado en mí un sentido de seriedad, responsabilidad y rigor académico.*

*Al M. Sc. Lic. Ing. Ramiro Kantuta Limachi, por su dedicación y paciencia en todo este proceso del proyecto de grado.*



## RESUMEN

El presente trabajo contiene el proceso de desarrollo del Sistema de Información Web para la gestión de entrega de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras. El proyecto tiene como finalidad implementar un sistema de gestión administrativa permitiendo conocer la estabilidad de la empresa, en cuanto a ventas, costos y llevar un registro adecuado de la información de los productos y clientes, que contribuyen enormemente a la toma de decisiones.

Primeramente, en el marco preliminar se presenta la introducción con las descripciones de la empresa y referencias de trabajos similares que fueron desarrollados anteriormente, posteriormente se diagnostica la situación problemática, para luego extraer los problemas y objetivos propuestos para la solución de estos, así mismo se presenta la justificación, como los límites y alcances, metodología, herramientas de desarrollo y los aportes del proyecto.

Dentro del marco teórico, se describe la metodología UWE, lenguajes de modelado UML, herramientas, pruebas de software, métricas de calidad ISO/IEC 25010 y estimación de costos de software.

En el marco aplicativo, se aplica y desarrolla las fases de la metodología propuesta y los modelados de UML y las respectivas pruebas de software propuestas en el marco teórico.

Posteriormente, se aplica las métricas de calidad, las políticas de seguridad y estimaciones de costos del software.

Finalmente se presenta conclusiones y recomendaciones, las conclusiones se reflejan a los objetivos alcanzados en el presente proyecto y las recomendaciones son las ideas que se aportan para el mantenimiento del sistema y las futuras mejoras que se puede llevar a cabo



## ABSTRACT

*This work contains the development process of the Web Information System for the management of product delivery and respective payment to improve the promotion of Leaders and Advisors. The purpose of the project is to implement an administrative management system allowing to know the stability of the company, in terms of sales, costs and to keep an adequate record of product and customer information, which contribute greatly to decision making.*

*Firstly, in the preliminary framework, the introduction is presented with the descriptions of the company and references of similar works that were previously developed, then the problematic situation is diagnosed, and then the problems and objectives proposed for their solution are extracted, likewise. presents the justification, such as the limits and scope, methodology, development tools and the contributions of the project.*

*Within the theoretical framework, the UWE methodology, UML modeling languages, tools, software testing, ISO/IEC 25010 quality metrics and software cost estimation are described.*

*In the application framework, the phases of the proposed methodology and the UML modeling and the respective software tests proposed in the theoretical framework are applied and developed.*

*Subsequently, quality metrics, security policies and software cost estimates are applied.*

*Finally, conclusions and recommendations are presented, the conclusions are reflected in the objectives achieved in this project and the recommendations are the ideas that are provided for the maintenance of the system and the future improvements that can be carried out.*



## INDICE

1.1.	INTRODUCCION.....	1
1.2.	ANTECEDENTES.....	2
1.2.1.	Antecedentes Institucionales.....	2
1.2.1.1.	Visi ó n.....3	
1.2.1.2.	Misi ó n.....3	
1.2.1.3.	Organigrama.... 3	
1.2.2.	Antecedentes afines al proyecto de grado.....	4
1.2.2.1.	Antecedentes Internacionales.....	4
1.2.2.2.	Antecedentes nacionales.....	5
1.3.	PLANTEAMIENTO DEL PROBLEMA.....	8
1.3.1.	Problema principal.....	8
1.3.2.	Problemas secundarios.....	8
1.3.3.	Formulaci ó n del problema.....	8
1.4.	OBJETIVOS 8	
1.4.1.	Objetivo General.....	8
1.4.2.	Objetivos Espec í ficos.....	9
1.5.	JUSTIFICACION.....	9
1.5.1.	Justificaci ó n T é cnica.....	9
1.5.2.	Justificaci ó n Econ ó mica.....	9
1.5.3.	Justificaci ó n Social.....	10
1.6.	METODOLOGIA.....	10
1.6.1.	Metodolog í a de desarrollo.....	10
1.6.2.	UML (Lenguaje Modelado Unificado).....	11
1.6.3.	M é tricas de Calidad.....	11
1.6.4.	COCOMO II.... 11	
1.6.5.	Seguridad..... 12	
1.6.6.	Pruebas de Software.....	12
1.6.6.1.	Caja negra.... 12	
1.6.6.2.	Caja blanca.... 13	
1.6.6.3.	Pruebas de estr é s.....	13
1.7.	HERRAMIENTAS.....	13



1.7.1.	Servidor Web Apache.....	13
1.7.2.	HTML5.....	14
1.7.3.	CSS.....	14
1.7.4.	PHP.....	14
1.7.5.	MariaDB.....	14
1.7.6.	Framework Laravel.....	15
1.7.7.	Composer.....	15
1.7.8.	JQuery.....	15
1.8.	LIMITES Y ALCANCES.....	16
1.8.1.	Limites.....	16
1.8.2.	Alcances.....	16
1.9.	APORTES	16
1.9.1.	Aporte Económico.....	16
1.9.2.	Aporte Social.....	17
2.1.	INTRODUCCIÓN.....	18
2.2.	Marco conceptual.....	18
2.2.1.	Sistema.....	18
2.2.2.	Sistema Web.....	18
2.2.3.	Preventa.....	19
2.2.4.	Catálogo.....	19
2.2.5.	Control de Inventarios.....	19
2.2.6.	Inventario	20
2.2.7.	Catálogo en Línea.....	20
2.2.8.	Inventario físico de almacén.....	21
2.2.9.	Catálogo en Línea.....	21
2.2.10.	Inventario físico de almacén.....	21
2.3.	Metodología UWE.....	21
2.3.1.	Principales características.....	22
2.3.2.	Fases de desarrollo.....	22
2.3.2.1.	FASE I: Análisis de requisitos.....	23
2.3.2.2.	FASE II: Diseño conceptual.....	23





2.3.2.3.	FASE III: Dise ñ o navegacional.....	24
2.3.2.3.1.	Modelo de espacio de navegaci ó n.....	25
2.3.2.3.2.	Modelo de estructura de navegaci ó n.....	25
2.3.2.4.	FASE IV: Dise ñ o de presentaci ó n.....	26
2.4.	LENGUAJE DE MODELADO UNIFICADO (UML).....	27
2.4.1.	ESPECIFICACIONES DE CASOS DE USO.....	28
2.4.2.	DIAGRAMAS.....	30
2.4.2.1.	DIAGRAMA DE CLASES.....	30
2.4.2.2.	DIAGRAMA DE ESTADOS.....	32
2.4.2.3.	DIAGRAMA DE SECUENCIA.....	33
2.4.2.4.	DIAGRAMA DE ACTIVIDADES.....	34
2.4.2.5.	DIAGRAMA DE DESPLIEGUE.....	34
2.4.2.6.	DIAGRAMA DE COMPONENTES.....	35
2.5.	HERRAMIENTAS.....	37
2.5.5.1	C ó mo funciona Modelo-Vista-Controlador (MVC).....	41
2.6.	INGENIER Í A DE SOFTWARE.....	43
2.6.1.	LA INGENIER Í A DE REQUERIMIENTOS Y SUS PRINCIPALES ACTIVIDADES.....	44
2.6.2.	CARACTER Í STICAS DE LOS REQUERIMIENTOS.....	45
2.7.	M É TRICAS DE CALIDAD.....	48
2.7.1.1.	Adecuaci ó n funcional.....	49
2.7.1.2.	Eficiencia de desempe ñ o.....	50
2.7.1.3.	Compatibilidad	50
2.7.1.4.	Usabilidad.....	50
2.7.1.5.	Fiabilidad.....	51
2.7.1.6.	Seguridad.....	52
2.7.1.7.	Mantenibilidad	52
2.7.1.8.	Portabilidad.....	53
3.	MARCO APLICATIVO.....	58
3.1.	Introducci ó n.....	58
3.2.	Arquitectura y requerimiento de software.....	58
3.2.1.	Arquitectura.....	58
3.3.	Funciones.....	58



3.4.	Fase 1 lista de requisitos.....	59
3.4.1.	Requisitos de Hardware y Software de desarrollo.....	60
3.5.	Fase 3 Análisis y diseño de la base de datos.....	61
3.5.1.	Descripciones de los actores.....	61
3.5.2.	Diagrama de caso de uso general.....	62
3.5.3.	Diseño de base de datos e implementación.....	63
3.6.	Fase 4 Programación - Base de datos.....	66
3.7.	Fase 5 Implementación y pruebas - base de datos.....	69
3.7.1.	Resultados.....	70
3.8.	Fase 3 Análisis y diseño - Módulo usuario.....	71
3.8.1.	Caso de Uso.....	72
3.8.2.	Diagrama de clases.....	73
3.8.3.	Diagrama de actividades o de flujo.....	74
3.8.4.	Diagrama de secuencia.....	74
3.9.	Fase 4 Programación - Módulo usuario.....	75
3.10.	Fase 5 Implementación y pruebas - Módulo usuario.....	77
3.10.1.	Prueba de Caja Blanca.....	77
3.10.2.	Prueba de Caja Negra.....	78
3.10.3.	Resultados.....	79
3.11.	Fase 3 Análisis y diseño - Módulo de Productos.....	79
3.11.1.	Caso de Uso.....	80
3.11.2.	Diagrama de clases.....	81
3.11.3.	Diagrama de actividades o de flujo.....	82
3.11.4.	Diagrama de secuencia.....	82
3.12.	Fase 4 Programación - Módulo de Productos.....	83
3.13.	Fase 5 Implementación y pruebas - Módulo de productos.....	84
3.13.1.	Prueba de Caja Blanca.....	84
3.13.2.	Prueba de Caja Negra.....	95
3.13.3.	Resultados.....	95
3.14.	Fase 3 Análisis y diseño - Módulo de clientes.....	96
3.14.1.	Caso de Uso.....	96
3.14.2.	Diagrama de clases.....	97



3.14.3.	Diagrama de actividades o de flujo.....	98
3.14.4.	Diagrama de secuencia.....	99
3.15.	Fase 4 Programaci3n - M3dulo de clientes.....	100
3.16.	Fase 5 Implementaci3n y pruebas - M3dulo de Clientes.....	101
3.17.	Prueba de Caja Blanca.....	102
3.17.1.	Prueba de Caja Negra.....	103
3.17.2.	Resultados.....	104
3.18.	Fase 3 An3lisis y dise1o - M3dulo de compras.....	104
3.18.1.	Caso de Uso..	104
3.18.2.	Diagrama de clases.....	105
3.18.3.	Diagrama de actividades o de flujo.....	106
3.18.4.	Diagrama de secuencia.....	107
3.19.	Fase 4 Programaci3n - M3dulo de compras.....	107
3.20.	Fase 5 Implementaci3n y pruebas - M3dulo de compras.....	109
3.20.1.	Prueba de Caja Blanca.....	109
3.20.2.	Prueba de Caja Negra.....	110
3.20.3.	Resultados.....	111
3.21.	Fase 3 An3lisis y dise1o - M3dulo de ventas.....	111
3.21.1.	Caso de Uso..	112
3.21.2.	Diagrama de clases.....	113
3.21.3.	Diagrama de actividades o de flujo.....	114
3.21.4.	Diagrama de secuencia.....	114
3.22.	Fase 4 Programaci3n - M3dulo de ventas.....	115
3.23.	Fase 5 Implementaci3n y pruebas - M3dulo de Ventas.....	117
3.23.1.	Prueba de Caja Blanca.....	117
3.23.2.	Prueba de Caja Negra.....	118
3.23.3.	Resultados.....	119
4.	M3TRICAS DE CALIDAD, SEGURIDAD Y ESTIMACI3N DE COSTOS.....	121
4.1.	Norma ISO/IEC 25010.....	121
4.1.1.	Adecuaci3n funcional.....	121
4.1.2.	Usabilidad.....	124
4.1.3.	Fiabilidad.....	125



4.1.4.	Mantenibilidad	127
4.1.5.	Portabilidad	128
4.1.6.	Calidad General	129
4.2.	Seguridad	130
4.2.1.	Autenticación	130
4.2.2.	Asignación de roles	131
4.2.3.	Encriptación	132
4.2.4.	Seguridad en la base de datos	133
4.3.	Estimación de costos del software	134
4.3.1.	Cocoma II	134
5.	CONCLUSIONES Y RECOMENDACIONES	138
5.1.	Conclusiones	138
5.2.	Recomendaciones	139
	BIBLIOGRAFIA	140



## ÍNDICE DE TABLAS

Tabla 1	Roles del equipo de trabajo.....	59
Tabla 2	Lista de Requisitos del proyecto.....	60
Tabla 3	Requisitos de Software y Hardware m í nimo para el Desarrollo.....	61
Tabla 4	Descripci ó n de los Actores del Sistema.....	62
Tabla 5	Modelo relacional de la Base de Datos.....	64
Tabla 6	Diccionario de datos de la Base de Datos.....	66
Tabla 7	M ó dulo de usuario, seguridad y las acciones.....	71
Tabla 8	Inicio de sesi ó n ingresa al m ó dulo de usuario.....	73
Tabla 9	C á lculo de complejidad.....	78
Tabla 10	Caminos independientes.....	78
Tabla 11	Prueba de Caja negra del m ó dulo usuario, seguridad y acciones.....	79
Tabla 12	M ó dulo Productos.....	79
Tabla 13	Descripci ó n de caso de uso del m ó dulo productos.....	80
Tabla 14	Inicio de sesi ó n, ingresar al m ó dulo de productos.....	81
Tabla 15	C á lculo de complejidad.....	85
Tabla 16	Caminos independientes.....	95
Tabla 17	Resultados de la implementaci ó n del m ó dulo de productos.....	95
Tabla 18	M ó dulo clientes.....	96
Tabla 19	Descripci ó n de caso de uso del m ó dulo cliente.....	96
Tabla 20	C á lculo de complejidad.....	102
Tabla 21	Caminos independientes.....	103
Tabla 22	Resultados de la implementaci ó n del m ó dulo cliente.....	103
Tabla 23	M ó dulo de Compras.....	104
Tabla 24	Descripci ó n de caso de uso del M ó dulo de compras.....	104
Tabla 25	Inicio de sesi ó n, ingresar al m ó dulo de compras.....	105
Tabla 26	Diagrama de clases del m ó dulo de compras.....	106
Tabla 27	C á lculo de complejidad.....	110
Tabla 28	Caminos independientes.....	110
Tabla 29	Resultados de la implementaci ó n del m ó dulo de compras.....	111
Tabla 30	M ó dulo de ventas.....	111



Tabla 31	Descripción de caso de uso del Módulo de ventas.....	112
Tabla 32	Inicio de sesión, ingresar al módulo de ventas.....	113
Tabla 33	Cálculo de complejidad.....	118
Tabla 34	Caminos independientes.....	118
Tabla 35	Resultados de la implementación del módulo de ventas.....	119
Tabla 36	Parámetros de medida y su cantidad.....	122
Tabla 37	Parámetros de medida y su cantidad.....	122
Tabla 38	Valores de ajuste de complejidad.....	123
Tabla 39	Encuesta sobre Usabilidad del Sistema.....	125
Tabla 40	Valores de fiabilidad de cada módulo.....	126
Tabla 41	Información requerida por el IMS.....	128
Tabla 42	Información de Calidad Global.....	130
Tabla 43	Líneas de código total.....	134
Tabla 44	Resultados de COCOMO II.....	137



## ÍNDICE DE FIGURAS

Figura 1 Organigrama Zermat La Paz.....	4
Figura 2 Diagrama de Casos de Uso.....	23
Figura 3 Diagramas de Clases.....	24
Figura 4 Diagrama de Navegación de Gestión de Inventarios.....	26
Figura 5 Diagrama de Presentación Página Principal.....	27
Figura 6 Símbolos Básicos de los Diagramas.....	28
Figura 7 Organigrama de la norma ISO 25010.....	49
Figura 8 Diagrama de Caso de Uso General.....	63
Figura 9 Modelo relacional de la Base de Datos.....	65
Figura 10 Prueba e implementación de Base de datos en el servidor.....	70
Figura 11 Caso de uso del Módulo usuario en la seguridad.....	72
Figura 12 Diagrama de clases Modelo Usuario y acceso de seguridad.....	73
Figura 13 Diagrama de actividades y módulo de usuario y seguridad.....	74
Figura 14 Diagrama de secuencia del módulo de usuario y seguridad, registro.....	75
Figura 15 Resultados módulo usuario y seguridad, acciones de registros.....	75
Figura 16 Acciones de registros.....	76
Figura 17 Grafo de flujo módulo de usuario y seguridad.....	77
Figura 18 Caso de uso del módulo productos.....	80
Figura 19 Diagrama de clases del Módulo de productos.....	81
Figura 20 Diagrama de actividades o flujo modulo productos.....	82
Figura 21 Diagrama de secuencia módulo de productos.....	83
Figura 22 Entornos gráficos del módulo de productos/Registros.....	83
Figura 23 Grafo de flujo módulo de productos.....	85
Figura 24 Caso de uso del Módulo cliente.....	97
Figura 25 Inicio de sesión, ingresar al módulo cliente.....	97
Figura 26 Diagrama de clases del módulo de cliente.....	98
Figura 27 Diagrama de actividades o flujo módulo cliente.....	99
Figura 28 Diagrama de secuencia módulo de registro cliente.....	100
Figura 29 Entornos gráficos del Módulo de clientes/registros.....	101
Figura 30 Grafo de flujo módulo cliente.....	102



Figura 31 Caso de uso del M ó dulo de compras.....	105
Figura 32 Diagrama de actividades o flujo m ó dulo de compras.....	106
Figura 33 Diagrama de secuencia m ó dulo de compra.....	107
Figura 34 Entornos gr á ficos del M ó dulo de compras.....	108
Figura 35 Grafo de flujo m ó dulo de compras.....	109
Figura 36 Descripci ó n de caso de uso del M ó dulo de ventas.....	112
Figura 37 Diagrama de clases del M ó dulo de ventas.....	113
Figura 38 Diagrama de actividades o flujo m ó dulo de ventas.....	114
Figura 39 Diagrama de secuencia m ó dulo de registro ventas.....	115
Figura 40 Entornos gr á ficos del M ó dulo de ventas.....	116
Figura 41 Grafo de flujo m ó dulo de ventas.....	117
Figura 42 C ó digo de validaci ó n de autenticaci ó n.....	131
Figura 43 Roles de asignaci ó n y Acciones.....	132
Figura 44 Encriptaci ó n Laravel.....	133
Figura 45 Inserci ó n de datos para COCOMO I.....	135
Figura 46 Resultado de estaci ó n de costo.....	136





# CAPÍTULO I



# I. CAP Í TULO I

## I.1. INTRODUCCION

Zermat en la ciudad de La Paz, se dedica a la comercializaci ó n de perfumes, cosm é ticos y productos de cuidado personal seg ú n cat á logo a la ciudadan í a en general por lo que rescatamos el art í culo de IBCE (2017), donde indica que las ventas por cat á logo generan 150.000 empleos, donde el 80% de esos 150.00 fuentes de empleo lo ocupan mujeres, siendo los rubros de mayor incidencia perfumer í a y cosm é tica. Para el 66% de estas 150.000, la venta de productos por cat á logo significa un ingreso complementario para su sustento diario.

De acuerdo al mismo informe el segmento de perfumer í a y cosm é tica, es el de mayor incidencia en los hogares bolivianos, con un porcentaje del 49% de participaci ó n de mercado del sector de venta directa. En segundo lugar, se encuentra la venta de indumentaria (ropa) y en tercero, el sector de nutrici ó n, con un 15% y 12% respectivamente.

Conscientes de esta necesidad la empresa ZERMAT INTERNACIONAL BOLIVIA se asienta en nuestro pa í s y esta empresa es l í der en la venta de productos de salud y belleza donde invita a la poblaci ó n boliviana a obtener atractivas ganancias, beneficios y reconocimientos, siguiendo 3 simples pasos: COMPRA, INVITA y GANA.

De acuerdo a lo indicado anteriormente los problemas surgen en la falta de control en la entrega de productos y su respectivo pago, por lo que se retrasan los informes de entrega de productos y cobros de las l í deres y asesoras. Este sistema tiene como objetivo principal generar informaci ó n confiable y oportuna, facilitando la promoci ó n de l í deres y asesoras.

Siguiendo una metodolog í a Use-Case-Driven Web Engineering (UWE) y las herramientas de desarrollo como Apache, PHP, HMTL5, CSS, Ajax, Bootstrap, jQuery y MySQL este sistema estar á orientado a la web

## I.2. ANTECEDENTES

Para llevar a cabo el desarrollo del Sistema de Informaci ó n Web para la gesti ó n de entrega de



productos y respectivo pago para mejorar la promoción de Líderes y Asesoras, se realizaron diversas investigaciones que incluyeron la revisión de bibliografías, artículos y libros relacionados con el desarrollo de software. Además, se llevaron a cabo visitas y entrevistas al personal de Zermat La Paz. Estas actividades de recopilación de información y datos permitieron obtener información relevante y actualizada sobre las mejores prácticas en la gestión de ventas y los requerimientos específicos de los usuarios. Con base en este análisis, se propone implementar el sistema con el objetivo de solucionar los problemas actuales, mejorar la eficiencia y brindar beneficios significativos a la empresa y sus usuarios.

### **1.2.1. Antecedentes Institucionales**

Zermat es una empresa 100% mexicana dedicada a la fabricación y comercialización de perfumes, cosméticos y productos de cuidado personal, elaborados con fórmulas y tecnología de vanguardia e ingredientes de la más alta calidad. En 1987 Zermat, cuyo nombre está inspirado en un bello pueblo suizo ubicado en las faldas de la montaña Matterhorn, fue fundada en 1987 en la Ciudad de México, gracias a los sueños, entusiasmo, perseverancia y trabajo del visionario Ing. Adolfo Enríquez Ríos y la Psicóloga Claudia Carrillo Enciso. La alta calidad de sus productos, el cuidado en el diseño de sus presentaciones, la extensión de sus líneas, su precio competitivo y su excelente modelo de negocio la han consolidado como una de las más importantes empresas en el país, además de ser líder también en Centroamérica y en el mercado hispano de los Estados Unidos.

En pocos años el exitoso modelo de negocio en el que está basada permitió expandir a todo México las fragancias Zermat.

Nuestra amplia gama de productos y nuestros altos estándares de calidad nos ha convertido en la marca que mejor entiende las necesidades del mercado latino.

Las fragancias Zermat están elaboradas con esencias francesas de primer nivel y nuestras líneas de cuidado Zermat Skin y Zermat Body Evolution contienen activos desarrollados por los laboratorios SEDERMA, Francia.

Esto ha permitido que Zermat se convierta en la primera empresa de perfumería y cosmética



de Latinoamérica en ser distinguida con el “Premio Internacional a la Calidad” recibido en 1999 y en 2009, en París, Francia.

#### *1.2.1.1. Visión*

Ser una Empresa de Clase Mundial con presencia Global en el mercado de Venta Directa, que ofrezca la mejor propuesta en productos de Moda, Salud y Belleza, incorporando la esencia latina en sus desarrollos y asegurando la mejor relación Precio-Calidad

#### *1.2.1.2. Misión*

Brindar a personas con espíritu emprendedor un modelo de Negocio ético y confiable a través de la venta de Productos de Moda, Salud y Belleza para toda la Familia, con un sistema de ganancias por Multinivel que garantiza el desarrollo personal y profesional mejorando su calidad de vida.

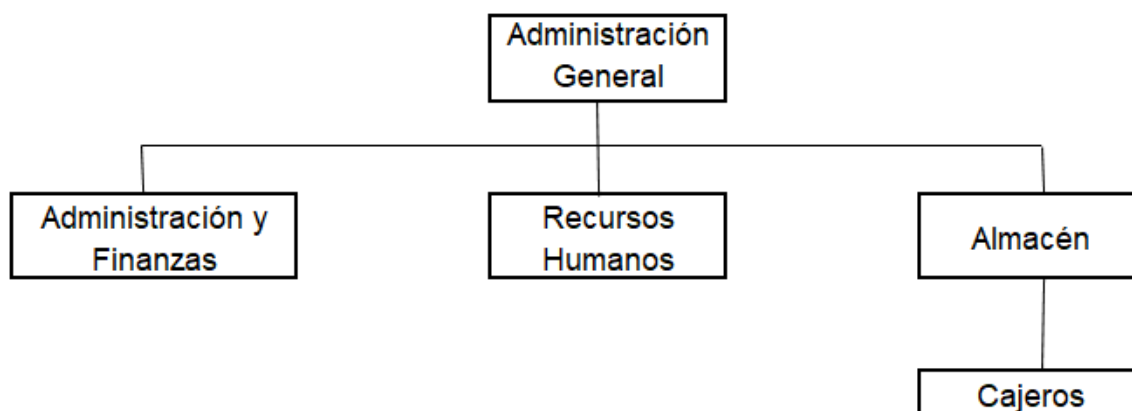
#### *1.2.1.3. Organigrama*

Zermat La Paz tiene la siguiente estructura organizacional:



Figura 1

Organigrama Zermat La Paz



Fuente: Elaboración Propia

## 1.2.2. Antecedentes afines al proyecto de grado

### 1.2.2.1. Antecedentes Internacionales

Según MONTOYA (2020), "DESARROLLO DE UNA TIENDA VIRTUAL PARA LA COMERCIALIZACIÓN ELECTRÓNICA DE PRODUCTOS DE LA MICROEMPRESA FAMILIAR ANGEL 'S CAKE", el presente trabajo obtiene información sobre la preparación y ejecución de un proyecto para la incorporación de una tienda virtual asignada a la microempresa ANGEL 'S CAKES, esta se realiza debido a un estudio minucioso realizado sobre los procesos para encontrar las falencias y las debidas soluciones a estas. La Universidad Católica de Pereira y la ciudad de Pereira fueron el escenario donde se recolectó información valiosa para el estudio, en estos lugares se llevó a cabo la investigación adecuada para descubrir cuáles eran los parámetros a seguir y bajo qué medidas, todo en pro de la ejecución del objetivo general. Se analizó sobre la importancia de mantener actualizada una empresa con los cambios no solo del consumidor, sino del entorno en el que se planea sacarla, por ello la parte tecnológica en esta época es



de vital importancia para la comercialización de productos o servicios.

Citando a TUNJA (2018), con el tema “DESARROLLO DE UNA TIENDA VIRTUAL PARA LA VENTA DE REPUESTOS AUTOMOTRICES EN LA EMPRESA MEGA REPUESTOS” , el proyecto trata del desarrollo de una tienda online para la Empresa Mega Repuestos. Esta plataforma proporcionará a la empresa, un servicio que facilita, agiliza y simplifica sus procesos de ventas de repuestos automotrices, así como la posibilidad de que sus clientes puedan realizar sus compras desde cualquier punto con una conexión a Internet y en cualquier momento, teniendo como única exigencia que los usuarios se encuentren registrados en la base de datos. El sistema es de fácil manejo con una interface amigable para el usuario, de manera que sus transacciones sean realizadas sin complicaciones.

### 1.2.2. Antecedentes nacionales

De acuerdo con FLORES, DOLZ Y CONTRERAS (2016), “SISTEMAS DE INFORMACIÓN PARA EL CONTROL Y EL SEGUIMIENTO DE VENTAS ON-LINE DE PRODUCTOS FARMACÉUTICOS DISTRIBUIDORA PHARMICA” . La tesis de grado propone ofrecer un servicio de información sobre los pedidos, ventas y el seguimiento de los mismos que sea confiable, pertinente y además que la información acerca del comportamiento de la empresa PHARMICA esté almacenada en una base de datos, segura en un servidor y protegida por un acceso restringido para su posterior utilización. Para la implementación del sistema se utilizaron los lenguajes de Programación: Php, Html, JavaScript, Css, los frameworks: CodeIgniter, JQuery, una base de datos creada en Mysql y el servidor apache los cuales se adecuan a las necesidades y requerimientos de la Institución. La metodología utilizada para el desarrollo del sistema es Proceso Unificado de Racional (RUP) para el análisis y diseño en sus diferentes fases, el modelado del sistema se realizó con el Lenguaje de Modelado Unificado (UML) y la herramienta de modelado Enterprise Architech.

Como se señala CALLE (2020), “GESTIÓN DE CONTROL DE MATERIA PRIMA, VENTA DE PRODUCTOS E INVENTARIOS USANDO SUPPLY CHAIN MANAGEMENT, CASO: INGENIERÍA Y TECNOLOGÍA BOLIVIANA INTEB SRL A. la tesis está enfocada en la gestión de inventario es una parte vital de cualquier negocio ya sea una tienda tradicional de accesorios de computadora o un sitio web de compras en línea. La gestión de inventario le proporciona información crítica



sobre el estado de sus productos, en caso que tenga poco stock y su inventario necesita reponerse. La gestión de inventario también le da una idea de que productos se venden bien y que productos no, con esta información le permite tomar mejores decisiones comerciales sobre como comercializar sus productos y abrir campo para nuevos clientes que están interesados en los productos y servicios que ofrece la empresa. La empresa INGENIERÍA Y TECNOLOGÍA BOLIVIANA INTEB SRL., con sede en la ciudad de El Alto, ofrece diversos servicios como diseño, construcción, mantenimiento de equipos y maquinaria industrial a diferentes instituciones a nivel nacional, de tal forma que toda la información generada por la empresa se genera en distintos lugares del territorio nacional. A medida que esta empresa fue creciendo, la información también fue incrementándose en sus distintos departamentos y áreas. Para el desarrollo del Sistema Web para el control de materia prima, venta de productos e inventarios basado en la arquitectura scm caso: Ingeniería y Tecnología Boliviana INTEB SRL. se utilizara herramientas de diseño y programación como ser para el Back-End se utilizara el lenguaje de programación C# (C-Sharp) con .Net Core 2.2 como framework, para la Base de Datos se utilizará Sql Server y para el Front-End se utilizará el Framework VUE.JS, para el modelado se utilizará los diagramas UML aplicado en una variación en la ingeniería de software abocada al desarrollo web denominado UWE. Para la seguridad se hará uso de la ISO/IEC 27002 anteriormente denominada ISO 17799:2005, describe los objetos de control y controles recomendables que va orientado a la seguridad de la información en las empresas u organizaciones, de modo que las probabilidades de ser afectados por robo, daño o pérdida de información se minimicen al máximo.

### 1.2.2.3. Antecedentes Locales

De acuerdo con CHOQUE (2020), en el “SISTEMA DE INFORMACIÓN PARA EL CONTROL DE INVENTARIO”, el presente proyecto tiene como objetivo desarrollar e implementar sistema de información para el control de inventario de ventas para la empresa “Victory Motors Bolivia”, que le permita administrar la información de manera eficiente, agilizando y mejorando los procesos que se realizan en la empresa. Para el análisis y diseño de la aplicación web se utilizó la metodología UWE UML (UML-Based Web Engineering) para la construcción y el diseño. UWE es el proceso de desarrollo para aplicaciones web, basada en las técnicas de UML. Para evaluar la calidad del software se utilizó la norma ISO 9126, para



la seguridad a la empresa se utilizó la norma 27000 y para la seguridad de la información se recurrió a la norma 27002 como finalmente para la estimación de costo de producto se usó COCOMO basado en el peso o líneas de código. La arquitectura MVC (Modelo, Vista, Controlador) ayuda a desarrollo a mantener ordenado los aspectos visuales de a lógica de negocio, PHP es el lenguaje de programación elegido para crear la aplicación del lado del servidor, Laravel 5.7 es un framework que utiliza la arquitectura MVC y MySQL como gestor de Base de Datos.

Desde el punto de vista CALLE (2020), “SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL DE COMPRAS VENTA DE INVENTARIOS” , el presente proyecto es un sistema de información web apoyado en las técnicas y procesos de Control de Compras, Ventas e Inventario con lo cual se busca mejorar el funcionamiento de la “EMPRESA PROVEEDORA T&J” , en los procesos de control de compras, ventas e inventario. El sistema brinda una serie de funcionalidades que fueron plasmadas en historias de usuarios, estas funcionalidades buscan satisfacer los requerimientos de la empresa y los funcionarios internos. El proyecto fue desarrollado utilizando la metodología de desarrollo UWE, con el lenguaje de programación Java Web y JavaScript, con la ayuda de los framework Bootstrap, todo junto con el sistema de gestor de base de datos MariaDB, además, como lenguaje de programación se utilizó las tecnologías de: Php, Ajax, jQuery los cuales incluyen herramientas de desarrollo Cliente Servidor. La evaluación de la calidad del software se detalla cualitativamente la calidad mediante la aplicación de métricas, tomando como factor de calidad el estándar ISO/IEC 9126. Para obtener los costos del proyecto se aplicó el modelo COCOMO II avanzado. Se implementó un sistema de información web que permita tener un control productivo a través de las compras, ventas e inventarios, cumpliendo con los objetivos propuestos.

### 1.3. PLANTEAMIENTO DEL PROBLEMA

#### 1.3.1. Problema principal

Los procesos de registro de líderes y asesoras se lo realizan manualmente en un archivo Excel, la entrega de productos se lo hace de la misma manera y el cobro de los productos entregados





es similar por lo que no tenemos información consolidada para la promoción de las asesoras a un cargo superior y también cumplir con las metas del plan multinivel.

### 1.3.2. Problemas secundarios

- Registros manuales de líderes y asesoras.
- Carencia de información en línea de la entrega de productos de acuerdo al plan multinivel de la empresa y tampoco el registro de los pagos.
- No se cuenta con una base de datos integral de todo el manejo de la empresa y el mismo se lo realiza en Excel.
- Al no contar con un sistema de información los datos de los clientes, productos y servicios se encuentran sin una base de datos de respaldo lo cual lleva a la pérdida de información que es muy importante la empresa.

### 1.3.3. Formulación del problema

¿De qué manera el Sistema de Información Web para la gestión de entrega de productos y su respectivo pago para mejorar la promoción de líderes y asesoras en la Empresa ZERMAT La Paz contribuirá a mejorar la eficiencia y control en los procesos de inventario y también que agilicen los procesos de registro de productos y líderes y asesoras, distribución y seguimiento mediante el Sistema de Información propuesto?

## 1.4. OBJETIVOS

### 1.4.1. Objetivo General

Implementar un Sistema de Información Web para la gestión de entrega de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras de la empresa ZERMAT La Paz.

### 1.4.2. Objetivos Específicos

- Realizar el análisis de requerimientos de la institución.
- Seleccionar el software base para el desarrollo de la solución.



- Diseñar la base de datos para tener control del manejo de las entregas y cobros de pedidos.
- Construir el módulo de ventas para las líderes y asesoras.
- Desarrollar el módulo de control y seguimiento de entrega de productos, de la misma para la gestión de los pagos.
- Realizar el módulo de información gerencial para promover líderes y asesoras de acuerdo a las ventas realizadas.

## 1.5. JUSTIFICACION

### 1.5.1. Justificación Técnica

La empresa ZERMAT La Paz cuenta con equipos de hardware y software, así como acceso a Internet, la implementación de un sistema web permitirá automatizar procesos, centralizar la información de las entregas de productos a las líderes y asesoras, también tener controlado la gestión de los cobros por los pedidos entregados a los clientes.

### 1.5.2. Justificación Económica

El Implementar un Sistema Web Gestor de Líderes y Asesoras en el Proceso de Entrega de Productos mediante la administración de procesos de negocio para la empresa ZERMAT La Paz. Al automatizar los procesos de control y seguimiento de ventas, se reducirá el tiempo dedicado a tareas administrativas repetitivas, lo que permitirá liberar recursos humanos que podrán ser asignados a actividades de mayor valor agregado. Esto mejorará la productividad y eficiencia de la empresa.

Además, aprovechando la infraestructura tecnológica existente, como equipos de computación y conexión a Internet, el desarrollo e implementación del sistema web no requerirá una inversión significativa en nuevos recursos, lo que lo convierte en una solución rentable y también mencionar que el uso del software no significará ningún costo para la empresa.

### 1.5.3. Justificación Social



La implementación de un Sistema de Información Web para la gestión de entrega de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras para la empresa ZERMAT La Paz, al mejorar los procesos de control y seguimiento de ventas permitirá que las Líderes y asesoras tengan la información en línea. En conjunto, estos aspectos contribuirán a una imagen sólida y favorable de la Empresa ZERMAT La Paz hacia los usuarios del sistema.

## 1.6. METODOLOGIA

### 1.6.1. Metodología de desarrollo

La metodología UWE, es un método utilizado en la ingeniería de software para desarrollo de aplicativos webs, y al ser este una extensión directa de UML puede utilizar diagramas propios de UML. Para la implementación se deben de visualizar los siguientes modelos.

Modelo de aplicación web según la metodología UWE:

- Modelo de Casos de Uso: Se modela requisitos funcionales de la aplicación web para ver como interactúa cada uno de ellos.
- Modelo Conceptual: Materializa en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- Modelo de Navegación: Especifica el entorno en la cual se realizará el aspecto de navegación de la aplicación web.
- Modelo de Presentación: Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

### 1.6.2. UML (Lenguaje Modelado Unificado)

Es una herramienta que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma fácil de comprender, para comunicar a otras personas. UML trabaja con los siguientes diagramas:

- Diagramas de casos de uso



- Diagrama de secuencias
- Diagrama de estados
- Diagrama de actividades
- Diagrama de colaboración
- Diagrama de clases

### 1.6.3. Métricas de Calidad

Según ISO 25000 (2022), en el desarrollo del Sistema de Información Web para la gestión de productos y pagos para mejorar la promoción de Líderes y Asesoras, se utilizarán métricas de calidad basadas en la norma ISO/IEC 25010:2011. Estas métricas nos permitirán evaluar y asegurar la calidad del software en términos de características específicas, como funcionalidad, usabilidad, eficiencia, fiabilidad y seguridad. Algunas de las métricas que se aplicarán incluyen el cumplimiento de requisitos funcionales, la facilidad de uso de la interfaz, la eficiencia en el uso de recursos, la disponibilidad del sistema, la protección de la información y el cumplimiento de normativas de seguridad. Estas métricas nos ayudarán a medir y garantizar la calidad del sistema, proporcionando una experiencia óptima a los usuarios.

### 1.6.4. COCOMO II

De acuerdo con RUIZ GONZALES (1999) el Modelo Constructivo de Costos (COCOMO) en su versión intermedia nos sirve para estimar los costos del software. El COCOMO es un modelo matemático basado en datos empíricos y consta de tres submodelos: básico, intermedio y detallado. En este caso, nos centraremos en el COCOMO Intermedio, que ofrece mayor detalle y precisión a medida que avanzamos en el desarrollo del software. Al utilizar este modelo, consideraremos factores como el tamaño estimado del software, la complejidad del producto, la experiencia del equipo de desarrollo y las características de la plataforma. Esto nos permitirá obtener estimaciones más precisas sobre el esfuerzo, la duración y los costos asociados al proyecto. Con el COCOMO Intermedio, podremos planificar y gestionar de manera



eficiente el desarrollo del Sistema de Información Web para la gestión de productos y pagos para mejorar la promoción de Líderes y Asesoras.

### **1.6.5. Seguridad**

Desde el punto de vista NB/ISO/IEC (2023) a la hora de implantar un Sistema de Gestión de la Seguridad de la Información (SGSI) según la norma ISO 27001, debemos considerar como eje central de este sistema la Evaluación de Riesgos. Este capítulo de la Norma, permitirá a la dirección de la empresa tener la visión necesaria para definir el alcance y ámbito de aplicación de la norma, así como las políticas y medidas a implantar, integrando este sistema en la metodología de mejora continua, común para todas las normas ISO.

### **1.6.6. Pruebas de Software**

#### **1.6.6.1. Caja negra**

Según PRESSMAN (2002), las pruebas de caja negra son para evaluar la funcionalidad del Sistema de Información Web para la gestión de productos y pagos para mejorar la promoción de Líderes y Asesoras de la empresa ZERMAT La Paz, sin considerar los detalles internos de su implementación. Estas pruebas garantizarán la calidad del sistema y su cumplimiento de los requisitos establecidos. Se utilizarán técnicas como pruebas de equivalencia, casos de uso, límites e interfaz. El objetivo es minimizar errores y ofrecer una experiencia óptima a los usuarios finales (p. 294).

#### **1.6.6.2. Caja blanca**

De acuerdo a PRESSMAN (2002) Las pruebas de caja blanca se enfocan en evaluar los detalles internos de la implementación del Sistema de Información Web para la gestión de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras de la empresa ZERMAT La Paz. Se analiza el código fuente y la lógica de programación para identificar posibles errores y mejorar la calidad del sistema. Estas pruebas incluyen técnicas como pruebas de cobertura, evaluación de bucles y condicionales, y análisis estático del código. Su objetivo es



asegurar que el sistema funcione correctamente a nivel interno y cumpla con las mejores prácticas de programación. Al realizar estas pruebas de manera rigurosa, se mejora la confiabilidad y calidad del sistema, reduciendo los riesgos de errores y fallos en su funcionamiento (p. 286).

### 1.6.3. Pruebas de estrés

Citando a KINSTA (2023), las pruebas de estrés son un tipo de pruebas de software que ayudan a determinar el rendimiento de un software en términos de velocidad, tiempo de respuesta, escalabilidad, uso de recursos y estabilidad bajo una carga de trabajo determinada. El objetivo de las pruebas de rendimiento es identificar todos los cuellos de botella de rendimiento de una aplicación.

Se realizarán pruebas de estrés en el Sistema de Información Web para la gestión de productos y pago para mejorar la promoción de Líderes y Asesoras para evaluar su rendimiento bajo condiciones extremas de carga.

## 1.7. HERRAMIENTAS

En el desarrollo del Sistema de Información Web para la gestión de entrega de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras se utilizarán diversas herramientas y tecnologías. A continuación, se mencionan algunas de las principales:

### 1.7.1. Servidor Web Apache

Como se ñala APACHE.ORG (1995), el servidor HTTP Apache es un software de servidor web multiplataforma gratuito y de código abierto, publicado bajo los términos de la licencia Apache 2.0. Es desarrollado y mantenido por una comunidad de desarrolladores bajo los auspicios de la Apache Software Foundation.

### 1.7.2. HTML5

Como afirma OpenWebinars (2023) del HTML5 lo primero que tenemos que saber que es la última versión de la tecnología de HTML, cuyas siglas corresponden a "HyperText Mark Language". HTML5, Es un estándar que sirve como referencia del software que conecta con la



elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos y entre otros.

### 1.7.3. CSS

Desde el punto de vista de HOSTINGER (2023), CSS son las siglas en inglés de Cascading Style Sheets, que significa “hoja de estilo en cascada”, es un lenguaje que se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS fue desarrollado por W3C “World Wide Web Consortium” en 1996 por una razón muy sencilla la de manejar el diseño y presentación de las páginas web.

### 1.7.4. PHP

Teniendo en cuenta a PHP GROUP (2023), PHP es un lenguaje de secuencias de comandos de propósito general orientado al desarrollo web. Fue creado originalmente por el programador danés-canadiense Rasmus Lerdorf en 1993 y lanzado en 1995. La implementación de referencia de PHP ahora es producida por PHP Group. PHP era originalmente una abreviatura de Página de inicio personal, pero ahora significa el inicialismo recursivo PHP: Preprocesador de hipertexto.

### 1.7.5. MariaDB

Según MariaDB Foundation (2023), MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius -fundador de MySQL-, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB-. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

### 1.7.6. Framework Laravel

Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones



escritas en este lenguaje de programación. Este framework o más bien podríamos llamarlo compañero de ahora en adelante, nos ayuda en muchas cosas al desarrollar una aplicación, por medio de sus sistema de paquetes y de ser un framework del tipo MVC (Modelo-Vista-Controlador) da como resultado que podamos “despreocuparnos” (por así decirlo) en ciertos aspectos del desarrollo, cómo instanciar clases y métodos para usarlos en muchas partes de nuestra aplicación sin la necesidad de escribirlo y repetirlo muchas veces con lo que eso conlleva a la hora de modificar algo en el código.

### 1.7.7. Composer

Es un gestor de dependencias para php. Composer nos **permitirá instalar librerías o paquetes de terceros**, de una forma muy ordenada y mantenible dentro de nuestro proyecto web.

### 1.7.8. JQuery

Como afirma JQUERY (2023), es un software libre y de código abierto (posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2). Cuenta con un diseño que facilita la navegación por un documento y seleccionar elementos DOM proporcionando a los desarrolladores de aplicaciones web complementos que agilizan el desarrollo de proyectos. Esto permite a los desarrolladores centrarse en lo importante y crear abstracciones para interacción y animación de bajo nivel, efectos avanzados y widgets temáticos de alto nivel sin invertir tiempo en desarrollar complejos algoritmos y métodos que los controlen desde cero y generando menos código que las aplicaciones hechas con JS puro.

## 1.8. LIMITES Y ALCANCES

### 1.8.1. Limites

En la implementación del Sistema de Información Web para la gestión de productos y pagos para mejorar la promoción de Líderes y Asesoras, se deben tener en cuenta los siguientes límites:

- No estará disponible el sistema para las asesoras, ya que el mismo es para las líderes para arriba de acuerdo a estructura organizacional.

### 1.8.2. Alcances





El Sistema de Información Web para la gestión de productos y pagos para mejorar la promoción de Líderes y Asesoras tendrá los siguientes alcances:

- Módulo de Administrador.
- Módulo de Control y Seguimiento de entrega de productos.
- Módulo de Líderes y Asesoras.
- Módulo de Cálculo de Bonos de Productividad.
- Módulo de información gerencial y toma de decisiones

## 1.9. APORTES

### 1.9.1. Aporte Económico

El aporte en cuestión económica será la minimización de costos en la elaboración del sistema tanto en la aplicación web y la base de datos con el objetivo de poder implementarlo de manera más consecuente.

En el aporte económico se están utilizando tecnologías de código abierto por lo cual la empresa no está pagando una licencia

### 1.9.2. Aporte Social

El presente sistema será de gran utilidad para las líderes ya que podrán sugerir la promoción de las asesoras como líderes de acuerdo a la información gerencial que proporcionara el sistema, ya que al ser amigable será de un manejo sencillo y podrá presentar una información eficaz y oportuna.





# CAPÍTULO II



## 2.1. INTRODUCCIÓN

El objetivo de este capítulo es dar a conocer la metodología de desarrollo de software que emplearemos, los conceptos y las definiciones, los mismos que fortalecerán las bases teóricas del presente proyecto de grado. De esta forma se garantiza un buen desarrollo del sistema para alcanzar los objetivos planteados.

## 2.2. Marco conceptual

En este punto se presenta algunos términos mencionados en el presente proyecto.

### 2.2.1. Sistema

Según Seen (1992), es “Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común” (p. 19).

Por otra parte, para Castellanos (2011), lo considera “Un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo” (p. 5).

Por lo tanto, sistema es un conjunto de actividades administrativas y procesos interrelacionados, cuyo objeto es generar información para la toma de decisiones.

### 2.2.2. Sistema Web

Según Oberta (2017), “crear aplicaciones web que ayuden a generar más tráfico, a compartir los contenidos, a aumentar las ventas, a mejorar la relación del cliente con la marca, a mejorar la reputación”

Por otra parte, López (2015), denomina “aplicación Web a aquellas herramientas a las que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador” .



En síntesis, los sistemas web son software que está almacenado en un servidor web, que son utilizadas por medio de un navegador y una conexión a internet.

### 2.2.3. Preventa

Las preventas es un sistema en el cual ponemos a disposición de los clientes productos físicos que saldrán al mercado en un futuro cercano, con la finalidad de que puedan asegurar su mercancía.

La preventa tiene como objetivo vender un producto o algún bien antes de que esté construido para garantizar la financiación de los materiales del producto.

### 2.2.4. Catálogo

De acuerdo con Martínez de Sousa J. (1993) lo define como el “conjunto de descripciones, de acuerdo con unas normas, de los asientos bibliográficos de los libros y documentos de una biblioteca, con la indicación del lugar que ocupa en ella” (p.241).

Por su parte Buonocore (1973) es “la nómina ordenada de las obras existentes de una biblioteca pública, privada o librería, con la indicación, mediante una clave o símbolo (signatura topográfica), del lugar donde puede ser hallado el libro.

Un catálogo es la lista ordenada o clasificada que se hará sobre cualquier tipo de objetos (monedas, bienes a la venta, documentos, entre otros) o en su defecto personas y también un catálogo será aquel conjunto de publicaciones u objetos que se encuentran clasificados normalmente para la venta.

### 2.2.5. Control de Inventarios

Según Espinoza O. (2011) El control de inventarios es una herramienta fundamental en la administración moderna, ya que esta permite a las empresas y organizaciones conocer las cantidades existente de productos disponibles para la venta, en un lugar y tiempo determinado, así como las condiciones de almacenamiento aplicables en las industrias.

Por otra parte, Aguilar (2009) señala que la importancia del control de inventarios reside en



el objetivo primordial de toda empresa: “obtener utilidades, que reside en gran parte de ventas” , ya que este es el motor de la empresa, sin embargo, si la función del inventario no opera con efectividad, la empresa no tendrá material suficiente para poder trabajar, el cliente se inconforma y la oportunidad de tener utilidades se disuelve; entonces, sin inventario simplemente no hay ventas (p.5)

Consiste en el ejercicio del control de las existencias; tanto reales como en proceso de producción y su comparación con las necesidades presentes y futuras, de los niveles de existencias y las adquisiciones precisas para atender la demanda de producción.

### 2.2.6. Inventario

Muller (2005) sostiene que los inventarios de una compañía están constituidos por sus materias primas, productos en proceso, los suministros que utiliza en sus operaciones y los productos terminados (p.1).

Moreno (2009) es el conjunto de bienes tangibles, en existencia, propios y disponibles para venta, consumo o producción de otros bienes, o bien, relación ordenada y valorada de bienes tangibles, propios, en existencia y disponibilidad inmediata, que en curso normal de operaciones, y en plazo razonable, están destinados a la venta ya sea su estado original de adquisición, después de transformados o modificados (p.97).

Se puede definir como el registro documental de todos los bienes tangibles que se tienen para la venta para su posterior comercialización (materias primas, productos en proceso y productos terminados).

### 2.2.7. Catálogo en Línea

Según Michel Gorman (1982), el catálogo en línea es un sistema de control bibliográfico que permite leer y recuperar los datos almacenados en la computadora por medio de numerosos puntos de acceso. Los datos recuperados son desplegados en la pantalla de la terminal y pueden ser impresos si así se requiere.

Por otra parte, G.S. Lawrence J.R. Matthews y C.E Miller (1986) sostienen que los catálogos en



Línea pueden ser descritos a partir de tres componentes básicos: un equipo de cómputo, una base de datos y una red de telecomunicaciones.

El catálogo en línea es considerado un sistema de almacenamiento y recuperación de información que se analiza en distintos niveles.

#### 2.2.8. Inventario físico de almacén:

Es una forma de verificación física que consiste en constatar la existencia o presencia real de los suministros, insumos, repuestos y otros bienes o materiales almacenados, apreciar su estado de conservación y condiciones de seguridad.

#### 2.2.9. Catálogo en Línea

Según Michel Gorman (1982), el catálogo en línea es un sistema de control bibliográfico que permite leer y recuperar los datos almacenados en la computadora por medio de numerosos puntos de acceso. Los datos recuperados son desplegados en la pantalla de la terminal y pueden ser impresos si así se requiere.

Por otra parte, G.S. Lawrence J.R. Matthews y C.E Miller (1986) sostienen que los catálogos en línea pueden ser descritos a partir de tres componentes básicos: un equipo de cómputo, una base de datos y una red de telecomunicaciones.

El catálogo en línea es considerado un sistema de almacenamiento y recuperación de información que se analiza en distintos niveles.

#### 2.2.10. Inventario físico de almacén:

Es una forma de verificación física que consiste en constatar la existencia o presencia real de los suministros, insumos, repuestos y otros bienes o materiales almacenados, apreciar su estado de conservación y condiciones de seguridad.

### 2.3. Metodología UWE

Según Rossainz & Ocampo (2001) la ingeniería web basada UML, en un proceso del desarrollo para aplicaciones web enfocado sobre el diseño sistemático, la personalización y la



generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación web

UWE (UML Web Engineering, en español ingeniería web basada en UML) es una metodología que permite especificar de mejor manera una aplicación web, para el proceso de creación de aplicaciones detalla esta, con una gran cantidad de definiciones, en el proceso de diseño lista que debe utilizarse. Procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajos y puntos de control.

UWE se especializa en la especificación de aplicaciones que se adaptan, y por eso hace énfasis especial en las características de personalización, y la definición de los modelos de usuario o en un patrón de características de navegación de los modelos de usuario o en un patrón de características de navegación basado en preferencias, tareas o conocimiento. Otros aspectos de interés de la metodología a UWE es la orientación a objetos, usuarios y la definición de un modelo de referencia que da soporte a la metodología y formaliza los modelos por el grado de restricciones y definiciones que proporciona.

### 2.3.1. Principales características

La metodología a UWE se basa principalmente en las siguientes características:

- Notación estándar: el uso de la metodología a UML para la elaboración de todos los modelos.
- Métodos definidos: pasos definidos para la construcción de cada modelo.
- Especificación de restricciones: se recomienda la elaboración de las recomendaciones de manera escrita, para que la exactitud en cada modelo aumente.

### 2.3.2. Fases de desarrollo

Con respecto al proceso de creación de la aplicación, uwe se vale mediante el uso de metodologías estándares como UML principalmente y también del lenguaje de especificación de restricciones asociado OCL (Object Constraint Lenguaje, en español Lenguaje de Restricciones para Objetos). Para recolectar los requerimientos necesarios para el





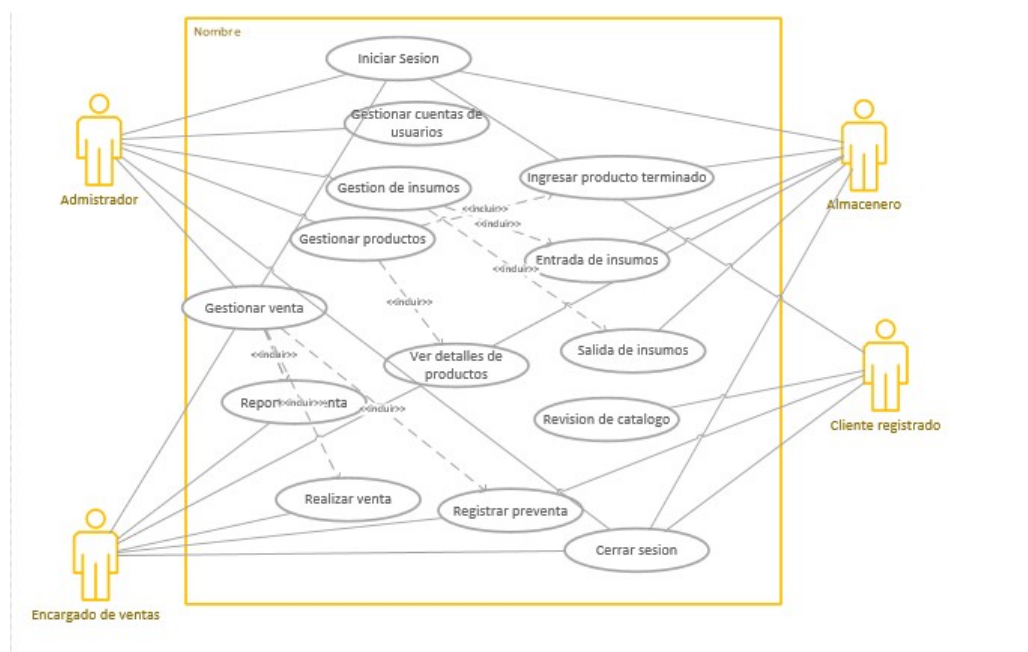
desarrollo de las aplicaciones web, esta metodología propone una ampliación utilizada en el proceso de creación, la cual se divide en las siguientes cuatro actividades:

### 2.3.2.1. FASE I: Análisis de requisitos

Fase en la que se plasman los requerimientos funcionales de la aplicación web, mediante modelos de casos de uso.

Figura 2

Diagrama de Casos de Uso



Fuente: Elaboración propia

### 2.3.2.2. FASE II: Diseño conceptual

Se define mediante un modelo de dominio, considerando los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle.

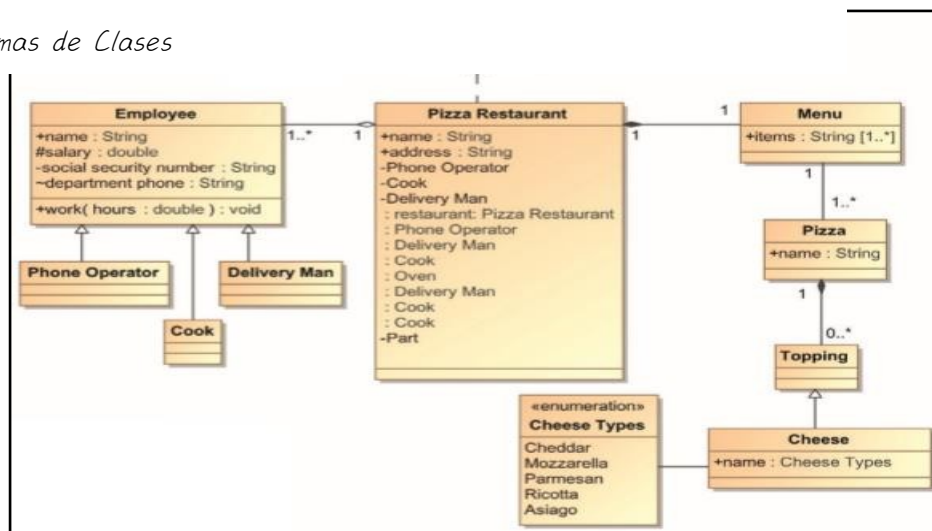
En esta etapa se representa el dominio del problema con un diagrama de clases de UML, que permiten determinar, métodos y atributos. El propósito de este diagrama es construir un modelo del dominio que intenta no considerar el paseo de la navegación, la presentación y los aspectos de interacción. Aspectos que se analizarán en los pasos



respectivos de navegación y presentación de la planificación.

Figura 3

Diagramas de Clases



Fuente: Elaboración propia

### 2.3.2.3. FASE III: Diseño navegacional

Basado en el diagrama de la fase conceptual, donde se especifican los objetivos que serán visitados dentro de la aplicación web y la relación entre los mismos.

Su objetivo es especificar las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación.

Este modelo se destaca en el marco de UWE como el más importante, ya que representa elementos estáticos, a la vez que se pueden incorporar al lineamiento semántico de referencia para las funcionalidades dinámicas de una aplicación web.

#### 2.3.2.3.1. Modelo de espacio de navegación

Su objetivo es especificar que objetos pueden ser visitados a través de la aplicación.



### 2.3.2.3.2. Modelo de estructura de navegación

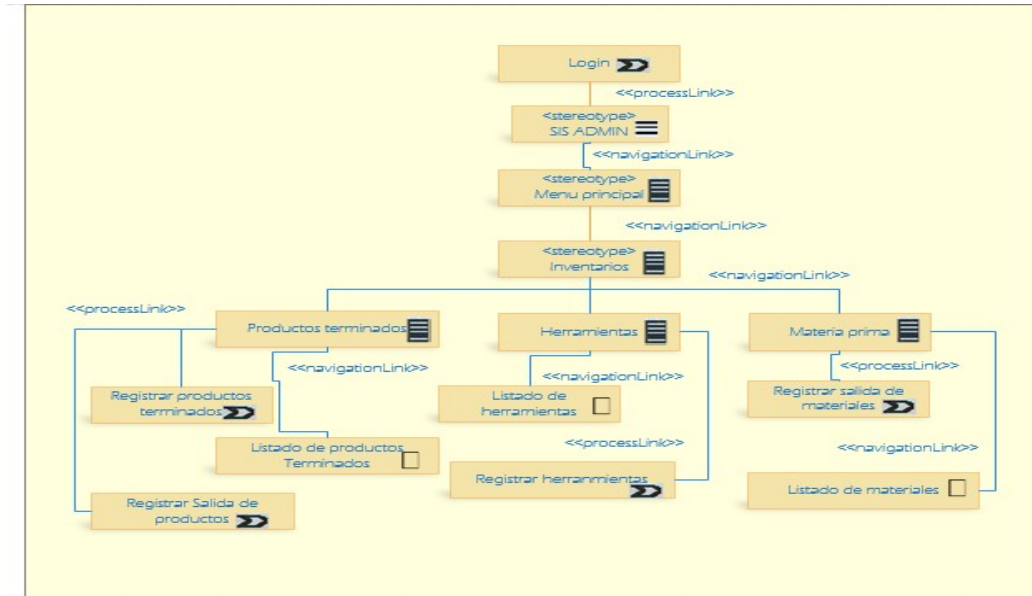
Amplía el modelo con un conjunto de estructuras de acceso necesarias para la navegación como:

- Los índices son una colección de objetos permitiendo una navegación directa.
- Un formulario facilita al usuario ingresar información para completar las condiciones de selección de objetos pertenecientes a las colecciones de índices y visitas guiadas.
- Un menú es un elemento parte de la navegación con un número específico de conexiones a otros objetos.
- Las visitas guiadas compuestas por un grupo de referencias, permitiendo una navegación secuencial a través de la aplicación.



Figura 4

Diagrama de Navegación de Gestión de Inventarios



Fuente: Elaboración propia

#### 2.3.2.4. FASE IV: Diseño de presentación

Permite la especificación lógica de la aplicación web. Basada sobre este modelo lógico, una representación física puede ser construida. Representa las interfaces del usuario por medio de vistas estándares de interacción UML. Dentro de este modelo se distinguen dos diferentes vistas:

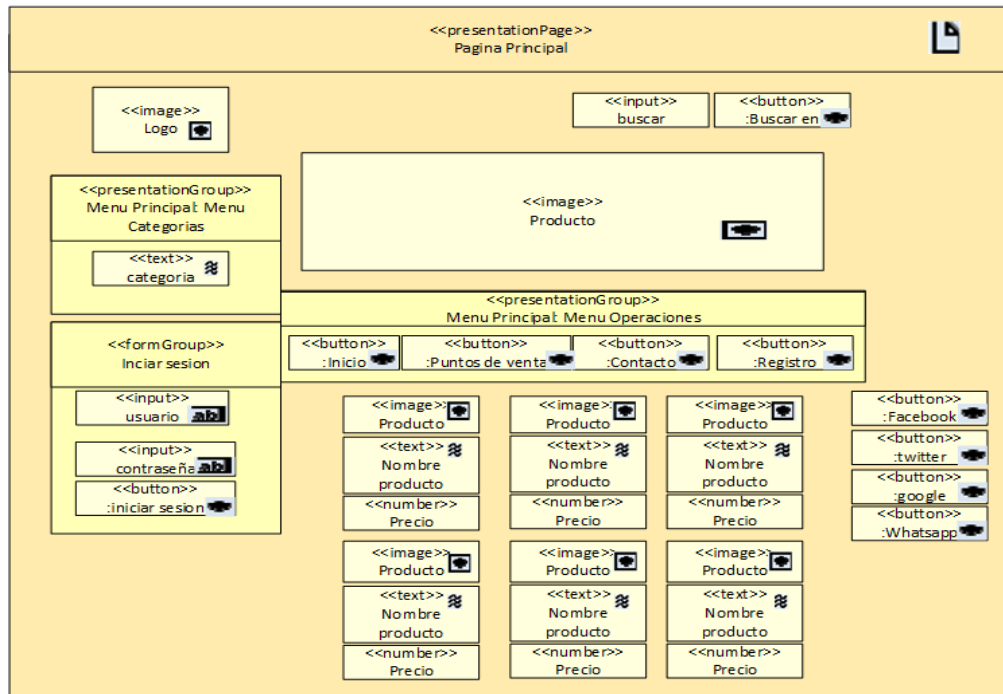
- Estructura de vista: muestra la estructura del espacio de presentación.
- Interfaz de usuario (UI por sus siglas en inglés de user interface):

Vista que presenta detalles acerca de los elementos de la interfaz de usuario dentro de las páginas.



Figura 5

Diagrama de Presentación Página Principal



Fuente: Elaboración propia

#### 2.4. LENGUAJE DE MODELADO UNIFICADO (UML)

Es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (ObjectModelingTechnique) o Booch sí definen procesos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el



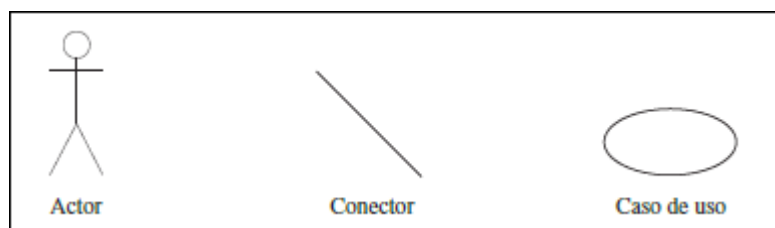
proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo. Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos.

#### 2.4.1. ESPECIFICACIONES DE CASOS DE USO

El estándar de lenguaje de modelado unificado de OMG define una notación gráfica para realizar diagramas de casos de uso, pero no el formato para describir casos de uso. Mucha gente sufre la equivocación pensando que un mal de uso es una notación gráfica (o es su descripción). Mientras la notación gráfica y las descripciones son importantes, ellos forman parte de la documentación de un caso de uso, un propósito para el que el actor puede usar el sistema el cual es representado en la siguiente gráfica:

Figura 6

*Símbolos Básicos de los Diagramas*



Fuente: Kimmel (2008, p. 40)

- La descripción escrita del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Esta descripción se enfoca en el valor suministrado por el sistema a entidades externas tales como usuarios humanos u otros sistemas.
- La posición o contexto del caso de uso entre otros casos de uso. Dado que es un mecanismo de organización, un conjunto de casos de uso coherente, consistente promueve una imagen fácil de comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo.

La interacción entre actores no se ve en el diagrama de casos de uso. Si esta interacción es esencial para una descripción coherente del comportamiento deseado, quizás los límites



del sistema o del caso de uso deban de ser reexaminados. Alternativamente, la interacción entre actores puede ser parte de suposiciones usadas en el caso de uso. Sin embargo, los actores son una especie de rol, un usuario humano u otra entidad externa pueden jugar varios papeles o roles.

### Relaciones de casos de uso

Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML el cual describe notación gráfica para esas relaciones.

#### Inclusión (include)

Es una forma de interacción o creación, un caso de uso dado puede “incluir” otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido. Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual, desde el caso de uso que lo incluye hasta el caso de uso incluido, con la etiqueta <<include>>. Este uso se asemeja a una expansión de un macro, donde el comportamiento del caso incluido es colocado dentro del comportamiento del caso de uso base. No hay parámetros o valores de retorno.

#### Extensión (Extend)

Es otra forma de interacción, un caso de uso dado, (la extensión) puede extender a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación, es una flecha de punta abierta con línea discontinua, desde el caso de uso extensión al caso de uso extendido, con la etiqueta <<extend>>. Esto puede ser útil para lidiar con casos especiales, o para acomodar nuevos requisitos durante el mantenimiento del sistema y sus extensiones. La extensión se utiliza en casos de uso, un caso de uso a otro caso siempre debe tener extensión o inclusión.

#### Generalización

En la tercera forma de relaciones entre casos de uso, existe una relación



generalización/especialización. Un caso de uso dado puede estar en forma especializada de un caso de uso existente. La notación es una línea sólida terminada en un triángulo dibujado desde el caso de uso especializado al caso de uso general. Esto se asemeja al concepto orientado a objetos de subclases, el a practica puede ser factor izar comportamientos comunes, restricciones al caso de uso general, describirlos una vez, enfrentarse a los detalles excepcionales en los casos de uso especializados.

## 2.4.2. DIAGRAMAS

Los diagramas representan los conceptos visualmente, por lo cual el siguiente propósito es organizar los conceptos de alto nivel (clasificación estructural, comportamiento dinámico y gestión del modelo) hablando de UML, esta se divide en pequeños conjuntos de vistas y diagramas. Muestra cómo implementar diferentes conceptos para describir un sistema, y como encajar las vistas de diseño de unas con otras.

### 2.4.2.1. DIAGRAMA DE CLASES

Un diagrama de clase muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones entre ellos.

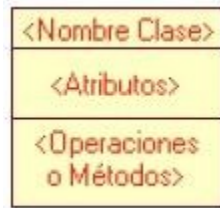
Los diagramas de clase se usan en el diseño del modelo estático para ver un sistema. Para las demás partes, este modelado involucra el vocabulario del sistema, el modelado de colaboraciones, o modelado de esquemas. Los diagramas de clase son también la base para un par de diagramas relacionados: Diagramas de Componente y Diagramas de Instalación. Los diagramas de clase son importantes no solo para la visualización, especificación y documentación del modelo estructural, pero también para la construcción de sistemas ejecutables. Ingeniería hacia adelante e ingeniería inversa.

La construcción de software tiene muchas características similares, excepto, que la calidad de software, uno tiene la habilidad de definir la construcción de bloques básicos para ir detallando.

Un diagrama de clases muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones. Gráficamente un diagrama de clase es una colección de vértices y arcos.







### PROPIEDADES COMUNES:

Un diagrama de clase es justo un tipo especial de diagrama y comparte propiedades comunes al igual que todos los otros diagramas -un nombre y un contenido gráfico son una proyección dentro de un modelo. Un diagrama de clases comúnmente tiene lo siguiente:

### OBJETO

Se define a un objeto como un concepto, abstracción o cosa con límites bien definidos y con significado a efectos del problema que se tenga entre manos. Los objetos tienen dos propósitos: promover la comprensión del mundo real y proporcionar una base práctica para la implementación por computadora. La descomposición de un problema en objetos depende del juicio y la naturaleza del problema.

### CLASE

Una clase de objetos describe un grupo de objetos con propiedades (atributos) similares, con relaciones comunes con otros y con una semántica común. Todo proceso tiene un poseedor, una prioridad y una lista de recursos requeridos. Los objetos y sus clases suelen aparecer como sustantivos en las descripciones de problemas.

### ATRIBUTOS

Un atributo es un valor de un dato que está almacenado en los objetos de una clase cada atributo tiene un valor para cada instancia del objeto. Las instancias de un cierto objeto pueden tener el mismo valor o valores distintivos para un atributo dado. El nombre del atributo es único dentro de la clase.

### 2.4.2.2. DIAGRAMA DE ESTADOS



Los diagramas de estado son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede entrar un objeto particular y la manera en que cambia el estado del objeto, como resultado de los eventos que llegan a él. En la mayor parte de las técnicas Orientadas a Objetos, los diagramas de estado se dibujan para una sola clase, mostrando el comportamiento de un solo objeto durante todo su ciclo de vida.

### **Dependencias**

Un objeto permanece en un estado por un tiempo finito, hasta que se cumpla la condición de cambio. Se construyen a partir del Diagrama de Estructura Estática (Diagramas de clases), identificando cuáles objetos cambian de estado, cual es el estado inicial y el final, definiendo a qué eventos puede responder el objeto, y qué transacciones ejecutará. Este diagrama muestra la secuencia de los estados de un objeto durante su ciclo de vida, en respuesta a un estímulo recibido. Los estados de los objetos están dados por el valor de sus atributos (estados) lo cual cambia sus comportamientos (mó todos).

Los estados hacen referencia a una condición durante la vida de un objeto o a una interacción durante la cual se satisface alguna condición (ejecutar alguna acción, esperar algún evento, etc.), por ejemplo, una validación de una captura.

### **Cuando utilizar los diagramas de estado**

Los diagramas de estados son buenos para describir el comportamiento de un objeto a través de varios casos de uso. No son tan buenos para describir un comportamiento que involucra cierto número de objetos que colaboran entre ellos. Así pues, es útil combinar los diagramas de interacción son buenos para la descripción del comportamiento de varios objetos en un mismo caso de uso. Por su parte, los diagramas de actividades son buenos para mostrar la secuencia general de las acciones de varios objetos y casos de uso.

Hay quienes consideran que los diagramas de estado son naturales, pero muchos no los consideran así. Preste atención al modo en que los emplean quienes trabajan con ellos, podría ocurrir que su equipo no considere útiles los diagramas de estado, debido a su modo de trabajar. Esto no sería un gran problema; como siempre, deben combinarse las técnicas



que sean de utilidad.

### 2.4.2.3. DIAGRAMA DE SECUENCIA

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

#### Tipos de mensajes

Existen dos tipos de mensajes: sincrónicos y asincrónicos. Los mensajes sincrónicos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada. Este tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asincrónicos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta. También se representa la respuesta a un mensaje con una flecha discontinua. Pueden ser usados en dos formas:

**De instancia:** describe un escenario específico (un escenario es una instancia de la ejecución de un caso de uso).

**Genérico:** describe la interacción para un caso de uso. Utiliza ramificaciones ("Branches"), condiciones y bucles.



#### 2.4.2.4. DIAGRAMA DE ACTIVIDADES

En un diagrama de actividades se muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Estas acciones las pueden llevar a cabo personas, componentes de software o equipos.

Puede usar un diagrama de actividades para describir procesos de diversos tipos, como los ejemplos siguientes:

- Un proceso de negocio o un flujo de trabajo entre los usuarios y el sistema.
- Los pasos realizados en un caso de uso.
- Un protocolo de software, es decir, las secuencias de interacciones permitidas entre los componentes.
- Un algoritmo de software

#### 2.4.2.5. DIAGRAMA DE DESPLIEGUE

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo:

- Dispositivos
- Procesadores
- Memoria

Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos. Las instancias de los nodos pueden contener instancias de ejecución, como instancias de componentes y objetos. El modelo puede mostrar dependencias



entre las instancias y sus interfaces, y también modelar la migración de entidades entre nodos u otros contenedores. Esta vista tiene una forma de descriptor y otra de instancia. La forma de instancia muestra la localización de las instancias de los componentes específicos en instancias específicas del nodo como parte de una configuración del sistema. La forma de descriptor muestra qué tipo de componentes pueden subsistir en qué tipos de nodos y qué tipo de nodos se pueden conectar, de forma similar a un diagrama de clases, esta forma es menos común que la primera.

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).

#### 2.4.2.6. DIAGRAMA DE COMPONENTES

Un diagrama de componentes es un diagrama tipo del Lenguaje Unificado de Modelado y estas se describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas.

Un componente de sólo compilación es aquel que es significativo únicamente en tiempo de compilación. Un componente ejecutable es un programa ejecutable y por eso este diagrama



tiene s ó lo una versi ó n con descriptors, no tiene versi ó n con instancias. Para mostrar las instancias de los componentes se debe usar un diagrama de despliegue.

Un diagrama de componentes muestra clasificadores de componentes, las clases definidas en ellos, y las relaciones entre ellas. Los clasificadores de componentes tambi é n se pueden anidar dentro de otros clasificadores de componentes para mostrar relaciones de definici ó n.

Contiene clasificadores de componentes y de nodo se puede utilizar para mostrar las dependencias del compilador, que se representa como flechas con l í neas discontinuas (dependencias) de un componente cliente a un componente proveedor del que depende. Los tipos de dependencias son espec í ficos del lenguaje y se pueden representar como estereotipos de las dependencias.

El diagrama tambi é n puede usarse para mostrar interfaces y las dependencias de llamada entre componentes, usando flechas con l í neas discontinuas desde los componentes a las interfaces de otros componentes.

**Objetivo.-** Se utilizan para modelar la vista est á tica de un sistema. Muestra la organizaci ó n y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

## 2.5. HERRAMIENTAS

### 2.5.1. MagicDraw

Es una herramienta visual de modelado UML y CASE con soporte de trabajo en equipo. Dise ñ ada para analistas de negocio, analistas de software, analistas de sistemas, programadores, ingenieros de control de calidad y redactores de documentaci ó n, esta herramienta de desarrollo din á mica y vers á til facilita el an á lisis y el dise ñ o de sistemas



orientados a objetos (OO) y bases de datos. Proporciona el mejor mecanismo de ingeniería de código de la industria (con soporte completo de ida y vuelta para los lenguajes de programación J2EE, C#, C++, CORBA IDL, .NET, XML Schema, WSDL), así como el modelado de esquemas de bases de datos, la generación de DDL y las facilidades de ingeniería inversa. MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs) Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros y los escritores de la documentación. (Magicdraw, 2021)

- ▶ **Un práctico editor UML.** Una interfaz gráfica de usuario amigable y personalizable le permite expresar sus ideas en UML de la forma más rápida y directa posible. Toda la notación UML 2 y la semántica son compatibles.
- ▶ **Una poderosa herramienta de ingeniería de código.** Puede construir un modelo UML, generar código fuente a partir de él, escribir más código a mano o en su IDE favorito, luego revertir el código, hacer algunos cambios en el MagicDraw, y fusionar su modelo con el código de nuevo. No se pierde ningún detalle, no importa el camino (hacia adelante o hacia atrás) que usted vaya. Soporte para Java, C#, C++, CORBA IDL, DDL.
- ▶ **Una facilidad de informes del modelo UML.** Basado en los archivos XSL, puede generar informes HTML personalizados para cada elemento del modelo. MagicDraw le permite seleccionar qué partes del modelo incluir y cómo debe ser el informe.
- ▶ **Una herramienta de análisis de modelos OO.** MagicDraw genera diagramas de dependencia de paquetes, recupera y muestra árboles de herencia, o rastrea cualquier otro tipo de relación como un parámetro de método o un valor de retorno, tales como dependencias, asociaciones, realizaciones y uso de clases.
- ▶ **Un generador de modelos OO personalizado.** Conjunto predefinido de patrones de diseño incluyendo GoF, Java, patrón de diseño JUnit. Todos los patrones pueden ser extendidos y nuevos patrones pueden ser creados usando Java o Jython.



- **Herramienta de modelado en equipo.** Más de un desarrollador puede trabajar con el mismo modelo OO. El modelo UML se almacena en el repositorio de Teamwork Server. Cada desarrollador puede bloquear una parte del modelo y trabajar en esa parte individualmente. Más tarde, los cambios pueden confirmarse en el servidor y compartirse con el equipo. El servidor funciona como un sistema de control de versiones de código fuente.

**Herramienta de modelado de bases de datos.** Puede diseñar el esquema de la base de datos en un diagrama de clases UML, y luego generar su código DDL. Si desea analizar la estructura de la base de datos existente, puede realizar ingeniería inversa a través del puente JDBC.

### 2.5.2. Lenguaje de Programación PHP

De acuerdo con PHP GROUP (2023) en lo referente a lenguajes de programación: PHP es un lenguaje de secuencias de comandos de propósito general orientado al desarrollo web. Fue creado originalmente por el programador danés-canadiense Rasmus Lerdorf en 1993 y lanzado en 1995. La implementación de referencia de PHP ahora es producida por PHP Group. PHP era originalmente una abreviatura de Página de inicio personal, pero ahora significa el inicialismo recursivo PHP: Preprocesador de hipertexto.

De acuerdo a Beati (2011), es un lenguaje de programación interpretado, PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, pueden interactuar con los servidores de web más populares ya que existen en versión CGI, módulo para Apache, e ISAPI (p. 67).

Según Bahit (2012), PHP es un lenguaje de programación de código abierto del lado del servidor que se utiliza principalmente para crear páginas web dinámicas. La abreviatura nació originariamente de "Personal Home Page





Tools” , aunque hoy en día se ha convertido en el acrónimo recursivo para “PHP: Hypertext Preprocessor” (p. 46).

Por lo facilidad de aprendizaje y también muy utilizado y desde luego los frameworks que permite, se utilizará en este proyecto.

### 2.5.3. HTML5

Citando a OpenWebinars (2019) cuando hablamos de HTML5 lo primero que tenemos que saber que es la última versión de la tecnología de HTML, cuyas siglas corresponden a “HyperText Mark Language” .

HTML5, Es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos y entre otros.

### 2.5.4. CSS

Según HOSTINGER (2023) CSS son las siglas en ingles de Cascading Style Sheets, que significa “hoja de estilo en cascada” , es un lenguaje que se usa para estilizar elementos escritos en un lenguaje de marcado como HTML.

CSS fue desarrollado por W3C “World Wide Web Consortium” en 1966 por una razón muy sencilla.

### 2.5.5. MariaDB

Según mariadb (2023), MariaDB Server es un sistema de gestión de bases de datos relacionales de código abierto. Es uno de los servidores de bases de datos más populares del mundo, con usuarios notables como Wikipedia , WordPress.com y Google . MariaDB Server se publica bajo la licencia de código abierto GPLv2 y se garantiza entonces que seguirá siendo abierto.



Como afirma MARIADB (2023), MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius -fundador de MySQL-, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB-. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

En conclusión, el motivo que lleva a trabajar con MariaDB por su facilidad de manejo, es software libre y muy portable.

### 2.5.6. Framework Laravel

Laravel es un framework PHP que trabaja con una arquitectura de carpetas avanzadas, de modo que promueve la separación de archivos con un orden correcto y definido, también dispone de una arquitectura de clases también muy adecuado que promueve la separación de código en una arquitectura de modelo-vista- controlador. Laravel pone en énfasis la calidad del código y la escalabilidad, mantenimiento lo cual permite realizar proyectos pequeños a grandes y muy grandes (desarrolloweb.com, 2020).

Contiene un amplio conjunto de características, que sirven para realizar la mayoría de las aplicaciones web:

- Un sistema de rutas, mediante la cual es fácil crear y mantener todo tipo de URLs.
- Un sistema de abstracción de base de datos, con un Mapeo Relacional de Objetos potente pero sencillo de manejar, mediante el podemos tratar los datos de la base de datos como si fueran simples objetos.
- Varias configuraciones para envíos de email, con diversos proveedores.
- Gestión de sesiones.



- Sistema de autenticación, con todo lo necesario.
- La posibilidad de acceder a los datos en realtime y recibir notificaciones cuando estos se alteren en la base de datos.

### 2.5.5.1 Cómo funciona Modelo-Vista-Controlador (MVC)

Estructuralmente el diseño del software está en tres partes:

- Modelo
- Vista
- Controlador

Modelo. – Es donde se procesa y obtienen los datos, la conexión con la base de datos.

Vista. – Presenta los datos en la pantalla, es donde va el código HTML.

Controlador. – Se encarga de controlar los datos de forma rápida que obtiene los datos en un modelo, luego los procesa y se los pasa a la vista.

### 2.5.7. Debian

Según DEBIAN (2023), es una distribución del Sistema operativo GNU/Linux compuesto por más de 40000 paquetes de software pre compilados, mayormente software libre y de código abierto bajo la Licencia Pública General GPL que incluye a Linux como principal núcleo del sistema. Debian también cuenta con una distribución especial como servidor, esta será la herramienta en la cual trabajará nuestro sistema.

### 2.5.8. Bootstrap

De acuerdo a Bootstrap (2023) es un framework CSS utilizado en aplicaciones front-end es decir, en la pantalla de interfaz con el usuario, para desarrollar aplicaciones que se adaptan a cualquier dispositivo.



### 2.5.9. Ajax

Como se ñala DEVELOPER MOZILLA.ORG (2023), Ajax es un conjunto de t é cnicas de desarrollo web que utiliza varias tecnolog ías web en el lado del cliente para crear aplicaciones web asincr ónicas. .Con Ajax, las aplicaciones web pueden enviar y recuperar datos de un servidor de forma asincr ónica (en segundo plano) sin interferir con la visualizaci ón y el comportamiento de la p á gina existente.

### 2.5.10. JQuery

Como afirma JQUERY (2023) es un software libre y de c ódigo abierto (posee un doble licenciamiento bajo la Licencia MIT y la Licencia P ública General de GNU v2). Cuenta con un dise ño que facilita la navegaci ón por un documento y **seleccionar elementos DOM** proporcionando a los desarrolladores de aplicaciones web complementos que agilizan el desarrollo de proyectos. Esto permite a los desarrolladores centrarse en lo importante y crear abstracciones para interacci ón y animaci ón de bajo nivel, efectos avanzados y widgets tem áticos de alto nivel sin invertir tiempo en desarrollar complejos algoritmos y m é todos que los controlen desde cero y generando **menos c ódigo** que las aplicaciones hechas con JS puro.

## 2.6. INGENIER ÍA DE SOFTWARE

De acuerdo a Pradel & Raya (1999), el IEEE define la ingenier ía como "la aplicaci ón de un enfoque sistem ático, disciplinado y cuantificable a las estructuras, má quinas, productos, sistemas o procesos para obtener un resultado esperado" y, más concretamente, la ingenier ía del software como "(1) La aplicaci ón de un enfoque sistem ático, disciplinado y cuantificable en el desarrollo, la operaci ón y el mantenimiento del software; es decir, la aplicaci ón de la ingenier ía al software. (2) El estudio de enfoques como en (1)" (p. 13).

Citando (G ómez, Cervantes, & Gonz áles, 2019)

Como se ñala (PRESSMAN, 2002, p á g. 14), "la ingenier ía del software es el establecimiento y uso de principios robustos de la ingenier ía a fin de obtener econ ómicamente software que



sea fiable y que funciones eficientemente sobre máquinas reales” (p.14).

Citando a Sommerville (2011), “La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software” (p. 3).

Por lo tanto, la ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software).

Detallando aún más, esta disciplina trasciende la actividad de programación, que es el pilar fundamental para la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema.

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura; la construcción es el desarrollo del producto; y la transición es la transferencia del producto terminado a los usuarios.

Una vez que se completa este ciclo, entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos (muchas veces advertidos por los propios usuarios) y se incorporan actualizaciones para hacer frente a los nuevos requisitos. El proceso de mantenimiento incorpora además nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas.

Un campo directamente relacionado con la ingeniería de software es la arquitectura de



sistemas, que consiste en determinar y esquematizar la estructura general del proyecto, diagramando su esqueleto con un grado relativamente alto de especificidad y señalando los distintos componentes que serán necesarios para llevar a cabo el desarrollo, tales como aplicaciones complementarias y bases de datos. Se trata de un punto fundamental del proceso, y es muchas veces la clave del éxito de un producto informático.

## 2.6.1. LA INGENIERÍA DE REQUERIMIENTOS Y SUS PRINCIPALES ACTIVIDADES

Un tema de ingeniería de software tiene diferentes significados. De las muchas definiciones que existen para requerimiento, a continuación, se presenta la definición que aparece en el glosario IEEE.

- 1ra condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- 2da condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal
- Una representación documentada de una condición o capacidad como en la primera y segunda condición.

En la parte de la ingeniería de software los requerimientos se dividen en requerimientos funcionales y requerimientos no funcionales.

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema como, por ejemplo, el rendimiento en tiempo y espacio utilizado, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad, estándares, etc.

## 2.6.2. CARACTERÍSTICAS DE LOS REQUERIMIENTOS

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación, se presentan las más importantes:



**Necesario:** un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.

**Conciso:** un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

**Completo:** un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

**Consistente:** un requerimiento es consistente si no es contradictorio con otro requerimiento.

**No ambiguo:** un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

**Verificable:** un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Dificultades para definir los requerimientos

- Los requerimientos no son obvios y vienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- Los requerimientos están relacionados unos con otros, ya su vez se relacionado con otras partes del proceso.



- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

Se darán algunos conceptos para ingeniería de requerimientos:

Ingeniería de requerimientos vs. Administración de requerimientos, el proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es denominado Ingeniería de requerimientos. La meta de la ingeniería de requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa.

Citando a Boehm (1979), “Ingeniería de requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas en donde se describen las funciones que realizará el sistema” .

De la misma manera Guide (1987), “Ingeniería de requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones” .

Respecto a este tema Leite (1987), “Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos” .

Por lo tanto, Ingeniería de requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; e también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto.





Los principales beneficios que se obtienen de la ingeniería de requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada: cada actividad de la IR consiste en una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados
- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la RE.

## 2.7. MÉTRICAS DE CALIDAD

En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto.

La medición es muy común en el mundo de la ingeniería. Medimos potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos.

Hay varias razones para medir un producto.

- Para identificar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados de uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación.
- Para ayudar a justificar el uso de las nuevas herramientas o de formación adicional.

Las mediciones del mundo físico pueden englobarse en dos categorías: medidas directas y medidas indirectas.



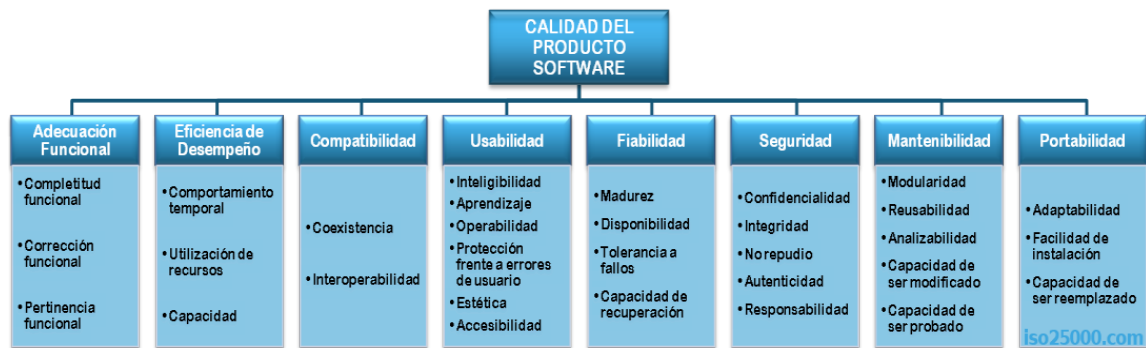
- Medidas directas: En el proceso de se encuentran el costo, y el esfuerzo aplicado. Las líneas de códigos producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado periodo de tiempo.
- Medidas indirectas: Se encuentra la funcionalidad, calidad, complejidad, fiabilidad, facilidad de mantenimiento, entre otras.

### 2.7.1. ISO/IEC 25010

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura 7.

Figura 7

Organigrama de la norma ISO 25010



Fuente: (iso25000.com, 2019)

#### 2.7.1.1. Adecuación funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica también se subdivide en las siguientes subcaracterísticas:

- **Complejidad funcional.** Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- **Corrección funcional.** Capacidad del producto o sistema para proveer resultados



correctos con el nivel de precisión requerido.

- **Pertinencia funcional.** Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

#### 2.7.1.2. Eficiencia de desempeño

Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica también se subdivide en las siguientes subcaracterísticas:

- **Comportamiento temporal.** Los tiempos de respuesta y procesamiento y las ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.
- **Utilización de recursos.** Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
- **Capacidad.** Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

#### 2.7.1.3. Compatibilidad

Dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica también se subdivide en las siguientes subcaracterísticas:

- **Coexistencia.** Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- **Interoperabilidad.** Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

#### 2.7.1.4. Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo



para el usuario, cuando se usa bajo determinadas condiciones. Esta característica también se subdivide en las siguientes subcaracterísticas:

- **Capacidad para reconocer su adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad para ser usado.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de hacer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

#### 2.7.1.5. Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica también se subdivide en las siguientes subcaracterísticas:

- **Madurez.** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- **Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.



- **Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

#### 2.7.1.6. Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Confidencialidad.** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- **Integridad.** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- **No repudio.** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- **Responsabilidad.** Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- **Autenticidad.** Capacidad de demostrar la identidad de un sujeto o un recurso.

#### 2.7.1.7. Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Modularidad.** Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- **Reusabilidad.** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.



- **Analizabilidad.** Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- **Capacidad para ser modificado.** Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- **Capacidad para ser probado.** Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

#### 2.7.1.8. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado. (iso25000.com, 2019)

#### 2.7.2. Prueba de caja blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante esta prueba de la caja blanca se puede obtener casos de prueba que:



- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o más todo.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

**Pruebas de ruta básica.** - Es una técnica que permite derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida para definir un conjunto de rutas básicas de ejecución.

**Notación de gráfico o grafo de flujo.** - En el gráfico de flujo de control lógico se muestra en cada círculo que es llamado nodo de gráfico de flujo, representan una o más enunciados de procedimientos, las flechas que los unen son llamadas aristas o enlaces, representan el flujo de control. Una arista debe terminar en un nodo, las áreas acotadas por aristas y nodos se llaman regiones.

Se presentan los siguientes pasos para derivar la ruta básica:

- Dibujar el gráfico de flujo a partir del diseño o código.
- Determinar la complejidad ciclométrica del gráfico de flujo resultante.
- Determinar un conjunto básico de rutas linealmente independiente.
- Preparar casos de prueba para forzar la ejecución de cada ruta en el conjunto básico.

Con esta prueba se obtiene como resultados la disminución en un gran porcentaje el número de errores existentes en el software y por ende una mayor calidad y confiabilidad en la codificación.

### 2.7.3. Prueba de caja negra



Las pruebas de caja negra, también denominadas pruebas de comportamientos, se centran en los requisitos funcionales del software. Analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Las pruebas de caja negra intentan encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de estructuras de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de inicio y fin.

Al ejecutar las pruebas de Caja Negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra.

#### 2.7.4. Análisis de costos COCOMO II

El modelo original COCOMO (Constructive Cost Model) fue publicado por primera vez en 1981 por Barry Boehm y reflejaba las prácticas en desarrollo de software de ese entonces. En las décadas siguientes las técnicas de desarrollo de software cambiaron drásticamente.

Estos cambios incluyen el gasto de tanto esfuerzo en diseñar y gestionar el proceso de desarrollo software como en la creación del producto software, un giro total desde los mainframes que trabajan con procesos batch nocturnos hacia los sistemas en tiempo real y un énfasis creciente en la reutilización de software ya existente y en la construcción de nuevos sistemas que utilizan componentes software a medida. (Boehm, 1981).

Los años y el avance tecnológico hicieron que la aplicación del modelo COCOMO original empezara a resultar problemática, como solución se determinó reinventar el modelo para





aplicarlo nuevamente y así después de muchos años de esfuerzo combinado surge COCOMO II, un modelo de estimación de costo que refleja los cambios en la práctica de desarrollo de software profesional.

Este nuevo y mejorado COCOMO resultará de gran ayuda para los estimadores profesionales de coste software. El principal cálculo en el modelo COCOMO es el uso de la ecuación del esfuerzo para estimar el número de personas o de meses necesarios para desarrollar el proyecto.

COCOMO define tres modos de desarrollo o tipos de proyectos:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Por otro lado, existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC.
- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

La función básica que utilizan los tres modelos es:

$$E = a (K \times l)^b \times m(X)$$



Donde:

- $a$  y  $b$  son constantes con valores definidos en cada submodelo.
- $Kl$  es la cantidad de líneas de código, en miles.
- $m(X)$ , es un multiplicador que depende de 15 atributos. El resultado se da en unidades salario/mes y horas-hombre.





# CAPITULO III



### 3. MARCO APLICATIVO

#### 3.1. Introducción

En este capítulo se desarrolla el “Sistema de Información Web para la gestión de entrega de productos y su respectivo pago para mejorar la promoción de líderes y asesoras en la Empresa ZERMAT La Paz” siguiendo la metodología de desarrollo Scrumban, para obtener la representación conceptual y física del sistema en el proceso de la elaboración.

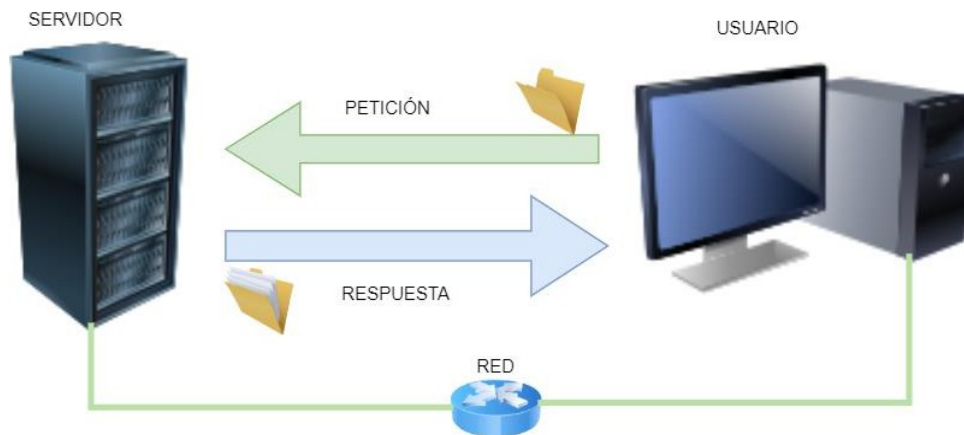
#### 3.2. Arquitectura y requerimiento de software

##### 3.2.1. Arquitectura

La arquitectura del software funciona de la siguiente forma en la interacción entre el servidor y el usuario encargado para poder realizar los debidos trabajos en el sistema.

Figura 1

Arquitectura del Software



Fuente: Elaboración propia

#### 3.3. Funciones

Se muestra en la siguiente tabla las funciones planteadas para el desarrollo del presente proyecto.



**Tabla 1**

Roles del equipo de trabajo

Funciones	Trabajos	Responsable
Equipo de desarrollo	Análisis y diseño	Edna Andrea Eulate
	Programación	Edna Andrea Eulate
	Implementación y pruebas	Edna Andrea Eulate
Cliente	Gerente General Zermat	Roxana de la Barra

Fuente: Elaboración propia

### 3.4. Fase I lista de requisitos

Para poder seleccionar las tareas los primeros pasos que se debe realizar es definir las listas de requisitos del sistema a priorizar, para esto fue necesario realizar reuniones con el cliente, que en este caso la función del cliente la cumplió la Gerente General Roxana de la Barra.

Para para obtención de la lista de requisitos se realizó diversas entrevistas personales con la Gerente general, preguntando el proceso y las etapas que se deben seguir para llevar los diferentes procesos de registro de los productos, como es la captación de clientes, ventas, compras, respaldo de documentos, cuantos tipos de usuarios existen la cual poder administrar, es decir expresarlos como usuarios y roles.

Viendo todos los detalles que el cliente proporciono se hizo un análisis de toda la información recolectada se elabora una tabla con los requisitos que son descrita en la siguiente tabla:



Tabla 2

Requisitos del proyecto

REQUISITOS					
N <sup>o</sup>	Nombre	Modulo	Prioridad	Tiempo de desarrollo	Pruebas
1	Diseño de la base de datos	Diseño	Demasiada Alta	10 Días	Implementar en el servidor
2	Autenticación de Usuario	Modulo Usuario	Muy Alta	8 Días	Probar la autenticación de usuario con los datos correctos Login al menú principal
3	Acciones de Usuario		Muy Alta	5 Días	Verificar las diferentes URLs de acceso y el CRUD de usuarios.
4	Roles de Usuarios		Muy Alta	5 Días	Probar los diferentes roles de asignación a cada usuario
5	Crear registro de tablas paramétricas	Módulo Paramétricas	Alta	10 Días	Verificar el funcionamiento y acciones que se realizan en CRUD
6	Crear registros de compras	Módulo compras	Alta	8 Días	Verificar el funcionamiento y acciones que se realizan en CRUD
7	Acciones a realizar en compras		Alta	8 Días	Verificar el funcionamiento y acciones que se realizan en CRUD
8	Crear registros de ventas	Módulo ventas	Alta	6 Días	Verificar el funcionamiento y acciones que se realizan en CRUD
9	Acciones a realizar en ventas		Alta	6 Días	Verificar el funcionamiento y acciones que se realizan en CRUD
10	Crear la generación de facturas o recibos	Generar Facturas o Recibos	Alta	6 Días	Probar la correcta generación de las facturas de ventas
11	Verificar ventas para promoción de asesoras	Reportes	Alta	6 Días	Verificar las ventas de acuerdo a plan multinivel

Fuente: Elaboración propia

### 3.4.1. Requisitos de Hardware y Software de desarrollo

Para el desarrollo del proyecto se necesita tanto hardware como software, en la cual se irá realizando la implementación y debidas pruebas también diferentes herramientas que nos ayudarán en el desarrollo.



Tabla 3

Requisitos de Software y Hardware mínimo para el Desarrollo

Hardware	Software
PC de escritorio o laptop con las siguientes características: <ul style="list-style-type: none"><li>• RAM 4 GB o superior</li><li>• Disco duro mínimo de 100 GB</li><li>• Procesador AMD, Ryzen 5 o superior, Core i5 o Superior</li><li>• Monitor</li><li>• Teclado</li><li>• Mouse</li><li>• Impresora</li></ul>	<ul style="list-style-type: none"><li>• Sistema Operativo Windows o Linux Programas necesarios:</li><li>• MySQL (Para el servidor donde se realiza las pruebas de la base de datos)</li><li>• Draw.io (Para la creación de diagramas)</li><li>• Sublime o Atom (Editor de código de múltiples Lenguajes de Programación)</li></ul>

Fuente: Elaboración propia

### 3.5. Fase 3 Análisis y diseño de la base de datos

#### 3.5.1. Descripciones de los actores

Para comprender Figura 8 página 63 diagrama de caso de uso general, se detalla un listado de roles y tareas que los usuarios desarrollan en el sistema las cuales son descritas en la siguiente tabla.





**Tabla 4**

*Descripción de los Actores del Sistema*

ACTOR	TAREAS
ADMINISTRADOR	Administrar a los usuarios. Autoriza los roles a los usuarios.
	Generar todos los reportes existentes
	Seleccionar y actualizar las compras
	Seleccionar y actualizar las ventas
	Gestionar los registros paramétricos: Clientes, proveedor, productos, tipos característica, tipo destinatario, tipo línea, tipo producto, tarifas
LÍDER DE GRUPO	Selecciona y actualiza roles, rangos, alcances mes
	Selecciona y actualiza clientes y tarifas
	Selecciona y actualiza alcances mes y reportes
ASESORA	Seleccionar y actualizar las ventas
	Emite reportes de sus ventas
ENCARGADO DE ALMACENES	Seleccionar y actualizar las ventas de su cuenta
	Seleccionar y actualizar las compras
	Gestionar los registros paramétricos: Proveedores, productos, tipos característica, tipo destinatario, tipo línea, tipo producto, tarifas

Fuente: Elaboración Propia

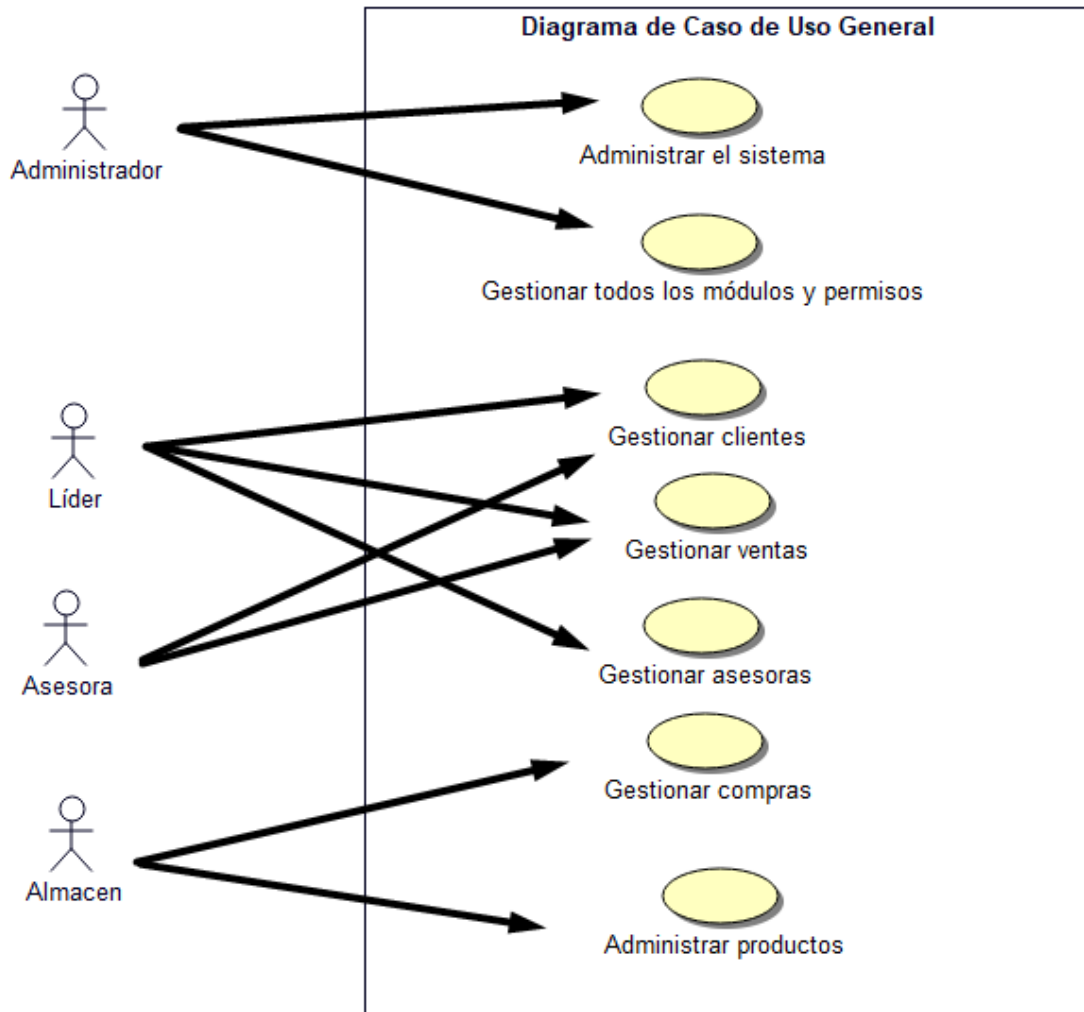
### 3.5.2. Diagrama de caso de uso general

Para comprender el funcionamiento del sistema se utiliza el diagrama de caso de uso de alto nivel la cual nos permitirá observar ordenadamente las funciones que se realizan.



Figura 8

Diagrama de Caso de Uso General



Fuente: Elaboración propia

### 3.5.3. Diseño de base de datos e implementación

En esta parte de la etapa se realizó la tarea de diseñar la base de datos e implementación. En la cual se diseñó el modelo relacional de la base de datos pensando en cumplir todos los requerimientos.



Tabla 5

Modelo relacional de la Base de Datos

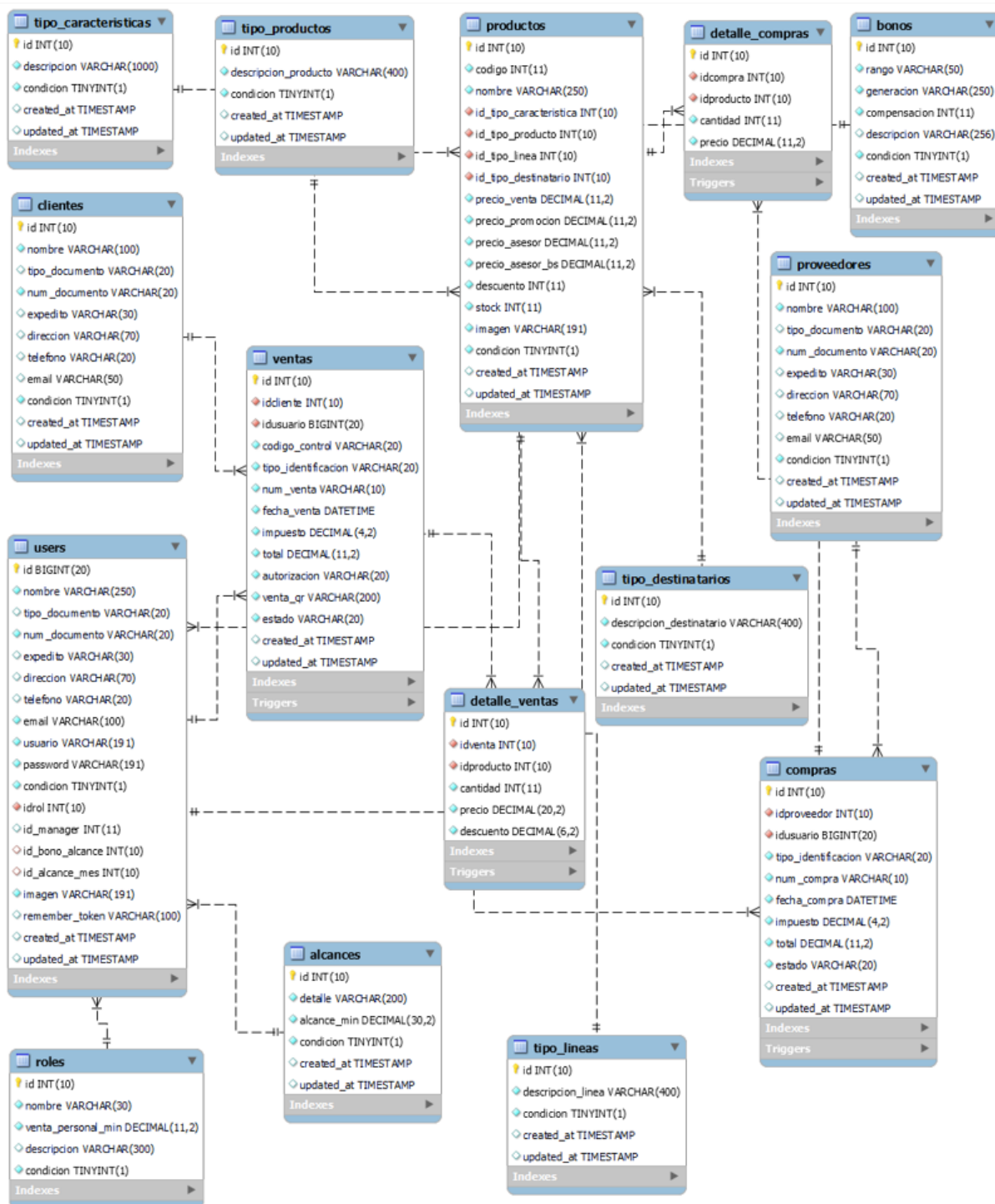
REQUISITO	TAREA	Inicio	Fin	Duraci3n
		03/04/2023	09/04/2023	7 d í as
		Desde	Hasta	Estado
Dise ñ o de la base de datos	Dise ñ o del modelo relacional de la base de datos e Implementaci3n	03/04/2023	09/04/2023	Completado

Fuente: Elaboraci3n propia



Figura 9

Modelo relacional de la Base de Datos



Fuente: Elaboración propia

### 3.6. Fase 4 Programación – Base de datos

Después de diseñar el modelo relacional se realiza la programación de la base de datos y el diccionario de datos correspondiente, la cual contiene nombre de las tablas, atributos, tipo de dato y descripción de los atributos. Se puede observar en la siguiente tabla:

**Tabla 6**

*Diccionario de datos de la Base de Datos*

TABLAS	ATRIBUTOS	TIPO	DESCRIPCIÓN
alcances	id	INT(10)	Llave primaria de la tabla
	detalle	VARCHAR(200)	Descripción breve del detalle
	alcance_min	DECIMAL(30,2)	Monto mínimo de venta
	condicion	TINYINT(1)	Estado de rol activo – no activo
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
bonos	id	INT(10)	Llave primaria de la tabla
	rango	VARCHAR(50)	Categoría líder
	generacion	VARCHAR(250)	Descripción del líder para el bono
	descripcion	VARCHAR(256)	Descripción del bono
	condicion	TINYINT(1)	Estado de rol activo – no activo
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
clientes	id	INT(10)	Llave primaria de la tabla
	nombre	VARCHAR(100)	Nombre completo del cliente
	tipo_documento	VARCHAR(20)	Tipo de documento
	num_documento	VARCHAR(20)	Numero de documento
	expedido	VARCHAR(30)	Lugar de emisión del documento
	direccion	VARCHAR(70)	Dirección
	telefono	VARCHAR(20)	Teléfono
	email	VARCHAR(50)	Correo electrónico
	condicion	TINYINT(1)	Estado activo – no activo
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
compras	Id	INT(10)	Llave primaria de la tabla
	idproveedor	INT(10)	Numero único del proveedor
	idusuario	BIGINT(20)	Numero único del usuario
	tipo.identificacion	VARCHAR(20)	Tipo de identificación del documento de compra



	num_compra	VARCHAR(10)	Número de la compra
	fecha_compra	DATETIME	Fecha y hora del registro de la compra
	impuesto	DECIMAL(4,2)	Registra el impuesto IVA
	Total	DECIMAL(11,2)	Total a pagar de las compras
	estado	VARCHAR(20)	Estado de compra registrado o anulado
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
detalle_compras	id	INT(10)	Llave primaria de la tabla
	idcompra	INT(10)	Número único de la tabla compras
	idproducto	INT(10)	Número único de la tabla producto
	cantidad	INT(11)	Cantidad
	precio	DECIMAL(11,2)	Precio
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
detalle_ventas	Id	INT(10)	Llave primaria de la tabla
	idventa	INT(10)	Número único de venta
	idproducto	INT(10)	Número único de producto
	cantidad	INT(11)	Cantidad de productos
	precio	DECIMAL(11,2)	Precio unitario del producto
	descuento	DECIMAL(4,2)	Descuento unitario del producto
	created_at	TIMESTAMP	Fecha creación del registro
	updated_at	TIMESTAMP	Fecha actualización del registro
productos	id	INT(10)	Llave primaria de la tabla
	codigo	INT(10)	Numero único del producto sin característica
	nombre	VARCHAR(250)	Nombre del producto
	id_tipo_característica	INT(10)	Descripción del producto
	id_tipo_producto	INT(10)	Llave foránea de tipo producto
	id_tipo_linea	INT(10)	Llave foránea de tipo línea
	id_tipo_destinatario	INT(10)	Llave foránea de tipo destinatario
	precio_venta	DECIMAL(11,2)	Precio de venta en \$us
	precio_promocion	DECIMAL(11,2)	Precio de promoción en \$us
	precio_asesor	DECIMAL(11,2)	Precio de asesora en \$us
	precio_asesor_bs	DECIMAL(11,2)	Precio de asesora en Bs
	descuento	DECIMAL(11,2)	% Descuento de acuerdo a la promoción
	stock	INT(11)	Cantidad de producto existentes
	imagen	VARCHAR(200)	Foto del producto
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creación del registro
updated_at	TIMESTAMP	Fecha actualización del registro	
proveedores	id	INT(10)	Llave primaria de la tabla
	nombre	VARCHAR(100)	Nombre completo
	tipo_documento	VARCHAR(20)	Tipo de documento



	num_documento	VARCHAR(20)	Numero de documento
	expedido	VARCHAR(30)	Lugar de emisi ó n del documento
	direccion	VARCHAR(70)	Direcci ó n
	telefono	VARCHAR(20)	Tel é fono
	email	VARCHAR(50)	Correo electr ó nico
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro
roles	id	INT(10)	Numero ú nico del rol
	nombre	VARCHAR(30)	Nombre del rol
	venta_personal_min	DECIMAL(11,2)	Venta m í nima por asesora y l í der
	descripcion	VARCHAR(100)	Descripci ó n breve del rol
	condicion	TINYINT(1)	Estado de rol activo - no activo
tipo_caracteristica	id	INT(10)	Llave primaria de la tabla
	descripcion	VARCHAR(1000)	Descripci ó n de la caracter í stica
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro
tipo_destinatario	id	INT(10)	Llave primaria de la tabla
	descripcion	VARCHAR(400)	Descripci ó n destinatario
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro
tipo_destinatario	id	INT(10)	Llave primaria de la tabla
	descripcion_linea	VARCHAR(400)	Descripci ó n de la l í nea
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro
tipo_producto	id	INT(10)	Llave primaria de la tabla
	descripcion_producto	VARCHAR(400)	Descripci ó n del tipo producto
	condicion	TINYINT(1)	Estado activo - no activo
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro
activity_log	id	INT(10)	Llave primaria de la tabla
	log_name	VARCHAR(191)	Nombre de registro
	description	TEXT	Descripci ó n de acci ó n que realiza
	subject_id	INT(11)	Numero ú nico del modificado
	subject_type	VARCHAR(191)	URL donde fue causado
	causer_id	INT(11)	Numero ú nico de quien modifiko
	causer_type	VARCHAR(191)	Registro de quien modifiko
	properties	TEXT	Registro de datos modificados



users	Id	BIGINT(20)	Llave primaria de la tabla
	nombre	VARCHAR(100)	Nombre completo de usuario
	tipo_documento	VARCHAR(20)	Tipo de documento del usuario
	num_documento	VARCHAR(20)	Numero de documento del usuario
	expedito	VARCHAR(30)	Lugar de emisi ó n del documento
	direccion	VARCHAR(70)	Direcci ó n del usuario
	telefono	VARCHAR(20)	Tel é fono del usuario
	Email	VARCHAR(50)	Correo electr ó nico del usuario
	usuario	VARCHAR(50)	Nombre de usuario para iniciar sesi ó n Contrase ñ a del usuario para iniciar sesi ó n
	password	VARCHAR(30)	
	condicion	TINYINT(1)	Estado del usuario activo - no activo
	Idrol	INT(10)	Numero de rol asignado al usuario
	imagen	VARCHAR(200)	Foto del usuario
ventas	Id	INT(10)	Llave primaria de la tabla
	idcliente	INT(10)	Numero ú nico del cliente
	idusuario	BIGINT(20)	Numero ú nico del usuario
	codigo_control	VARCHAR(20)	C ó digo de control de la venta
	tipo_identificacion	VARCHAR(20)	Tipo de documento
	num_venta	VARCHAR(10)	Numero de venta
	fecha_venta	DATETIME	Fecha de la venta
	impuesto	DECIMAL(4,2)	Impuesto IVA
	total	DECIMAL(11,2)	Total, a pagar
	autorizacion	VARCHAR(20)	N ú mero de autorizaci ó n
	venta_qr	VARCHAR(200)	C ó digo QR de la venta
	estado	VARCHAR(20)	Estado de la venta registrado o anulado
	created_at	TIMESTAMP	Fecha creaci ó n del registro
	updated_at	TIMESTAMP	Fecha actualizaci ó n del registro

Fuente: Elaboraci ó n propia

### 3.7. Fase 5 Implementaci ó n y pruebas - base de datos

Se realiz ó las pruebas de integraci ó n en base de datos implementada en el servidor de pruebas.





Figura 10

Prueba e implementación de Base de datos en el servidor

The screenshot shows the phpMyAdmin interface with the 'db\_venta' database selected. The 'Estructura' (Structure) tab is active, displaying a list of 18 tables. The table 'ventas' is highlighted in blue, indicating it is the current table being viewed. The table summary at the bottom indicates 476 rows and a total size of 832.0 KB.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
activity_log	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	176	InnoDB	utf8mb4_unicode_ci	113.0 KB	-
alcances	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	8	InnoDB	utf8mb4_unicode_ci	26.0 KB	-
bonos	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	7	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
clientes	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	2	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
compras	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	2	InnoDB	utf8mb4_unicode_ci	64.0 KB	-
detalle_compras	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	16	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
detalle_ventas	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
migrations	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	17	InnoDB	utf8mb4_unicode_ci	26.0 KB	-
password_resets	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	6	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
productos	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	12	InnoDB	utf8mb4_unicode_ci	88.0 KB	-
proveedores	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	6	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
roles	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	8	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
tipo_caracteristicas	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	104	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
tipo_destinatarios	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	6	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
tipo_lineas	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	22	InnoDB	utf8mb4_unicode_ci	26.0 KB	-
tipo_productos	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	13	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
users	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	17	InnoDB	utf8mb4_unicode_ci	122.0 KB	-
ventas	[Examinar] [Estructura] [Buscar] [Insertar] [Vaciar] [Eliminar]	2	InnoDB	utf8mb4_unicode_ci	64.0 KB	-

The screenshot shows the phpMyAdmin interface with the 'db\_venta' database selected. The 'Estructura' (Structure) tab is active, displaying the structure of the 'productos' table. The table has 11 columns and 11 rows of data. The columns are: id, codigo, nombre, id\_tipo\_caracteristica, id\_tipo\_producto, id\_tipo\_linea, id\_tipo\_destinatario, precio\_venta, precio\_promocion, precio\_asesor, and precio\_asesor\_bs. The data rows show various product entries with their respective codes, names, and prices.

id	codigo	nombre	id_tipo_caracteristica	id_tipo_producto	id_tipo_linea	id_tipo_destinatario	precio_venta	precio_promocion	precio_asesor	precio_asesor_bs	desuento
6	38133	POLVO VEGANO (CON) H-FAR	1	5	9	6	21.20	21.20	12.72	66.53	0
7	38134	POLVO VEGANO (CON) HENRY	1	10	6	2	21.20	21.20	12.72	67.00	0
8	38135	POLVO VEGANO (CON) A-H	1	10	6	2	21.20	21.20	12.72	66.53	0
9	38136	POLVO VEGANO (CON) H-FAR	1	10	6	2	21.20	21.20	12.72	66.53	0



Fuente: Elaboración propia

### 3.7.1. Resultados

Se realizó las pruebas necesarias en el servidor de pruebas de desarrollo.



Tabla N<sup>o</sup> 3.7 Resultados de la implementación de la Base de Datos

N <sup>o</sup>	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	Crear: ¿cuándo un usuario guarda la información en la base de datos cumple?	Cumple
2	Recuperar: ¿Cuándo el usuario realiza la búsqueda de la información almacenada en la base de datos cumple?	Cumple
3	Actualizar: ¿Cuándo un usuario edita o modifica un registro la información se guarda con los nuevos cambios, cumple?	Cumple
4	Eliminar: ¿Cuándo un usuario elimina o borra una información en la base de datos cumple?	Cumple

Fuente: Elaboración propia

### 3.8. Fase 3 Análisis y diseño - Módulo usuario

Este es prioridad Muy Alto los cuales son: Autenticación de Usuario, Acciones de Usuarios y Roles de Usuarios que pertenecen al Módulo Usuario. Se diseñó los diagramas en la que nos proporciona UML para cada requerimiento, para luego poder codificarlo.

Tabla 7

Módulo de usuario, seguridad y las acciones

REQUISITO	TAREA	Inicio	Fin	Duración
		Desde	Hasta	Estado
Autenticación de usuario	Diseño del diagrama UML	10/04/2023	11/04/2023	10 días
	Codificación	12/04/2023	13/04/2023	Completado
	Pruebas	14/04/2023	14/04/2023	Completado
Acciones de Usuario	Diseño del diagrama UML	15/04/2023	15/04/2023	Completado
	Codificación	16/04/2023	16/04/2023	Completado
	Pruebas	17/04/2023	17/04/2023	Completado
Roles de usuario	Diseño del diagrama UML	18/04/2023	18/04/2023	Completado
	Codificación	19/04/2023	19/04/2023	Completado
	Pruebas	20/04/2023	20/04/2023	Completado

Fuente: Elaboración propia



### 3.8.1. Caso de Uso

En la siguiente tabla se describe como se realiza el módulo.

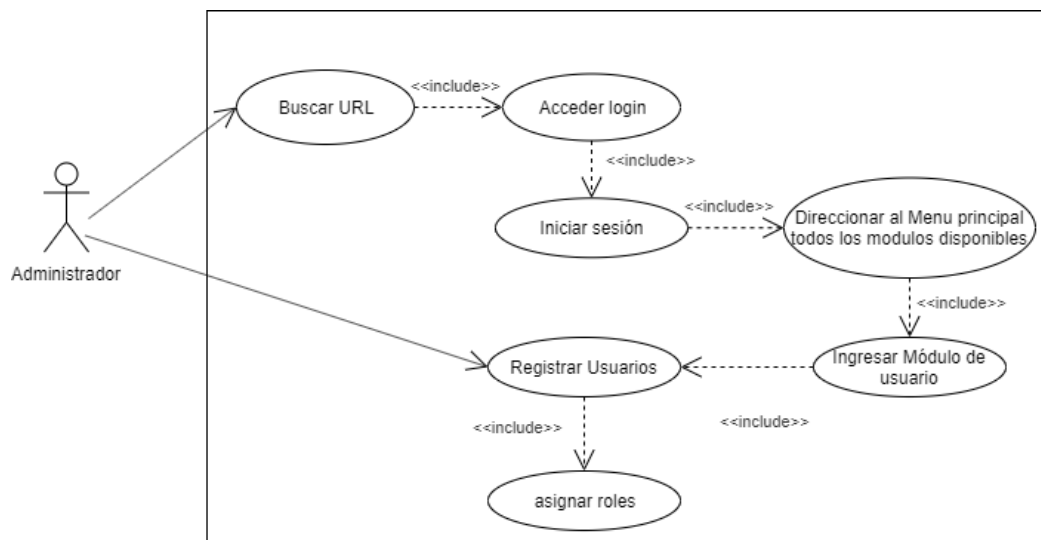
Tabla N<sup>o</sup> 3.9 Descripción de caso de uso del Módulo Usuario en la seguridad

N <sup>o</sup>	Descripción
1	Para la autenticación se debe dirigir a la página principal del dominio a la opción login
2	Se autentifica con los datos del usuario y se direcciona al menú principal
3	Asigna intervalos de tiempo en el acceso de usuario
4	Muestra todos los módulos disponibles para la administración ingresa al módulo usuario y registra usuarios con la debida asignación de roles y se registra las acciones.

Fuente: Elaboración propia

Figura 11

Caso de uso del Módulo usuario en la seguridad



Fuente: Elaboración propia



Tabla 8

Inicio de sesión ingresa al módulo de usuario

Nombre.	Iniciar sesión
Actores	Administrador. Sistema.
Propósitos	Para el usuario pueda ingresar debe en navegador buscar la URL o dirección de la página web y dir la parte del login e iniciar la sesión con sus datos asignados y el sistema verifica que el usuario los permisos para acceder, luego es direccionado al menú principal del sistema la cual el usuario podrá dirigirse y acceder al módulo de usuarios para los debidos registros.
Flujo principal	<ul style="list-style-type: none"> <li>• Evento del actor: Introduce el nombre de usuario y la contraseña</li> <li>• Evento del sistema: Valida ambos parámetros enviados, ingresa al sistema con los permisos correspondientes y puede ingresar al módulo de usuario administrar y registra nuevos usuarios y se registra las acciones.</li> </ul>
Flujo alternativo	Si no se encuentra el usuario o contraseña no coinciden, se envía un mensaje "Usuario o Contraseña incorrecta"
Pre condición	El usuario debe estar registrado en la base de datos, tanto su contraseña.
Pos condición	Solo un usuario autorizado ingresa al sistema.
Presunción	Si el usuario no se encuentra registrado no podrá ingresar al sistema.

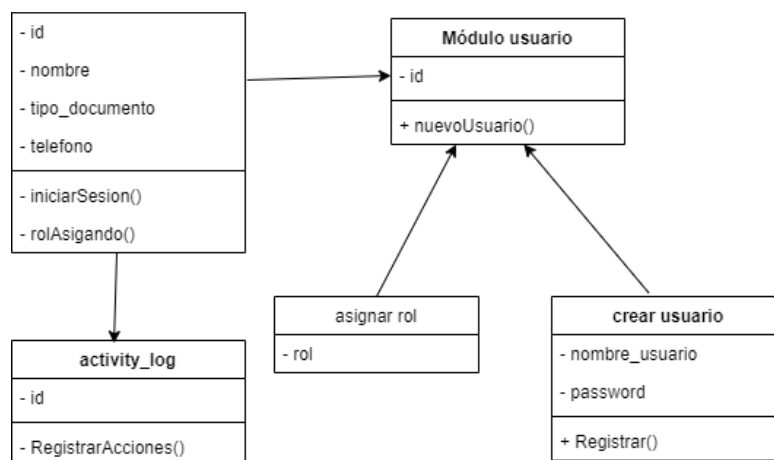
Fuente: Elaboración propia

### 3.8.2. Diagrama de clases

Se diseñó el diagrama de clases para el requisito Autenticación de Usuario, mostrar acciones del usuario y rol el cual se visualiza en la siguiente figura:

Figura 12

Diagrama de clases Modelo Usuario y acceso de seguridad



Fuente: Elaboración propia

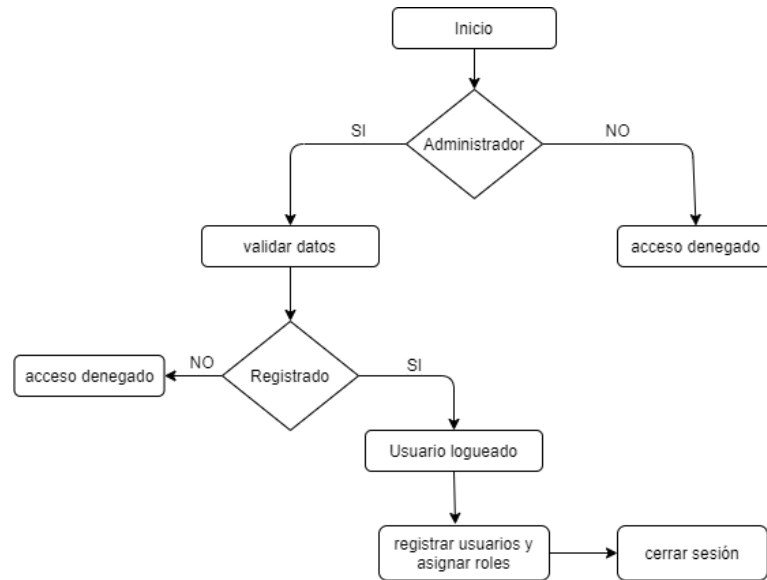


### 3.8.3. Diagrama de actividades o de flujo

Se diseñó el diagrama de actividades los requisitos de autenticación de usuario y las acciones que tiene, lo cual se visualiza en la siguiente figura:

Figura 13

Diagrama de actividades y módulo de usuario y seguridad



Fuente: Elaboración propia

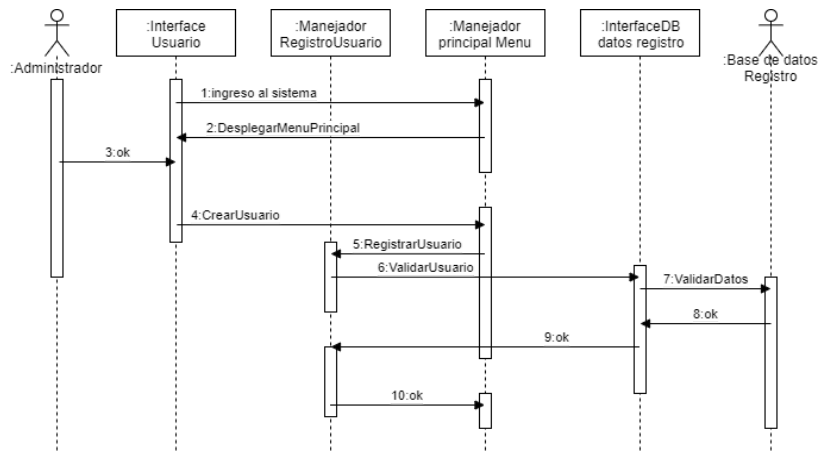
### 3.8.4. Diagrama de secuencia

Se diseñó el diagrama de secuencias para la autenticación de usuario, acciones de usuario y asignación de roles la cual se puede observar en el siguiente gráfico:



Figura 14

Diagrama de secuencia del módulo de usuario y seguridad, registro



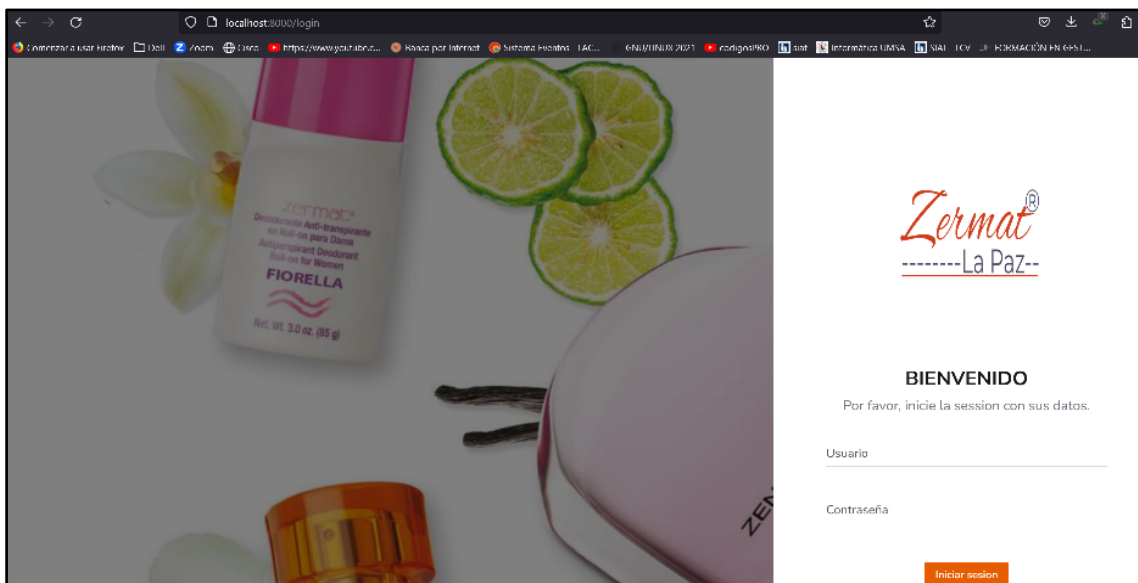
Fuente: Elaboración propia

### 3.9. Fase 4 Programación – Módulo usuario

Después de la programación se obtiene los siguientes resultados como se observa en las siguientes figuras:

Figura 15

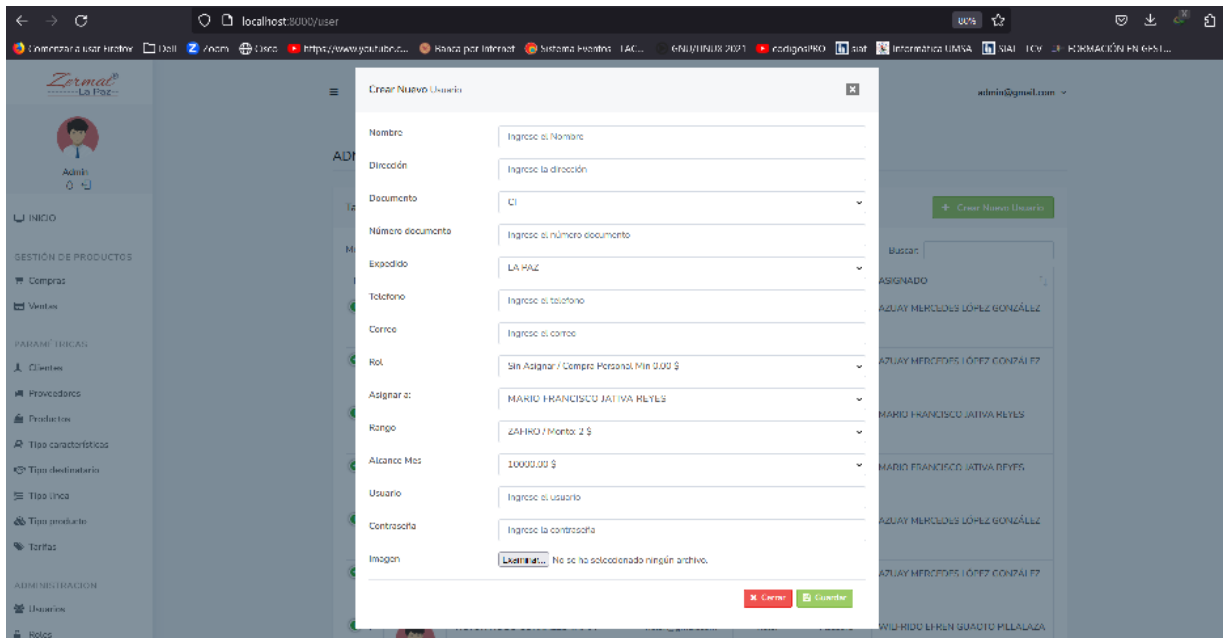
Resultados módulo usuario y seguridad, acciones de registros



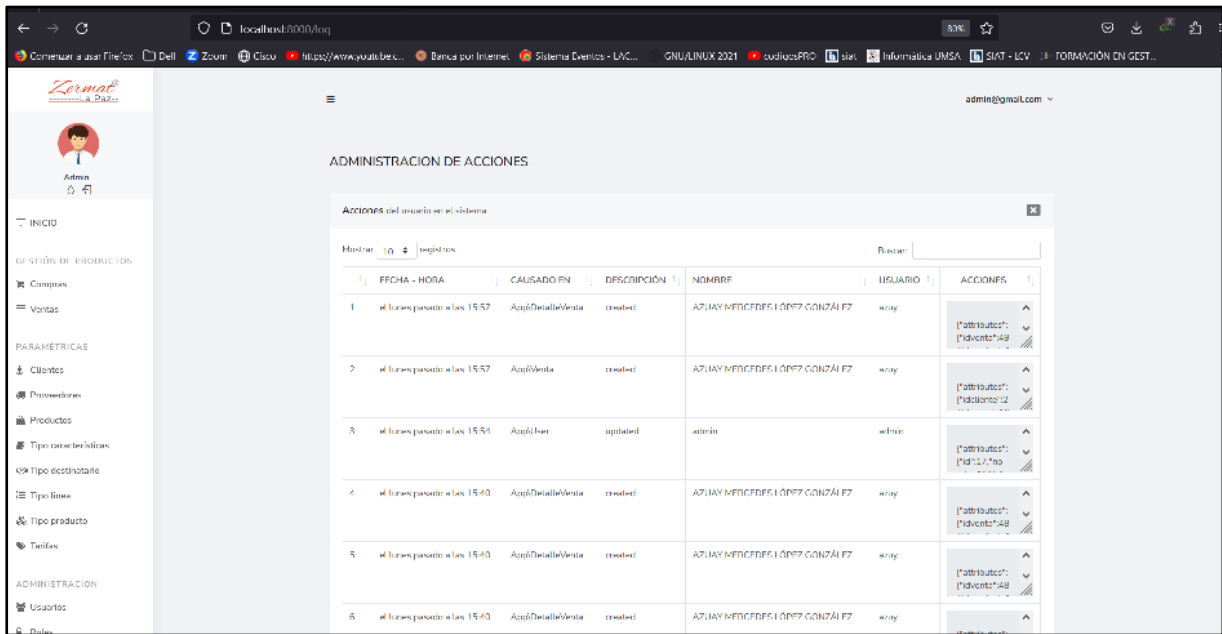
Fuente: Elaboración propia

Figura 16

Acciones de Registros







Fuente: Elaboración propia

### 3.10. Fase 5 Implementación y pruebas – Módulo usuario

Se realiza la implementación una vez finalizado la programación, posteriormente se realizan las pruebas de caja blanca y negra.

#### 3.10.1. Prueba de Caja Blanca

Esta prueba se centra en el comportamiento interno y la estructura del programa, en el cual

Figura 17

Grafo de flujo módulo de usuario y seguridad



se examina la lógica interna del programa sin considerar los aspectos de rendimiento. La prueba de la caja blanca intenta garantizar que se ejecute al menos una vez todos los caminos independientes en cada módulo, se aplican las decisiones verdadera o falsa ejecutando los bucles a sus límites. La prueba que se realizó para este módulo es calcular la complejidad ciclomática del diagrama de flujo de la figura N<sup>o</sup> 3.8.

Fuente: Elaboración propia

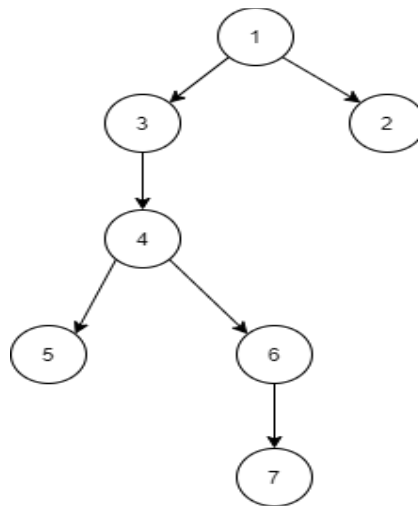


Tabla 9

Cálculo de complejidad

NODOS	ARISTAS	REGIONES	NODOS PREDICADO
7	6	3	1
$V(G) = \text{Aristas} - \text{Nodos} + 2$			$V(G) = 6 - 7 + 2 = 1$
$V(G) = \text{Nodos Predicados} + 1$			$V(G) = 2 + 1 = 3$
$V(G) = \text{Regiones}$			$V(G) = 3$
Complejidad ciclomática de 3 que nos indica realizar 3 pruebas			

Fuente: Elaboración propia

Tabla 10

Caminos independientes

CAMINO	ENTRADA DATOS	PRUEBA	SALIDA
1,3,4,6,7	-> Administrador	-> Si usuario es administrador está registrado en la base de datos. -> Si usuario está registrado, usuario conectado puede registrar nuevos usuarios y asignar roles -> Si es administrador valida datos.	Registro de usuarios y asignación de roles con éxito  Acceso denegado
1,3,4,5	->Administrador	-> No registrado en la base de datos acceso denegado	
1,2	->Administrador	-> No administrador acceso denegado	Acceso denegado

Fuente: Elaboración propia

### 3.10.2. Prueba de Caja Negra

Esta prueba de software verifica la funcionabilidad sin tomar en cuenta la estructura interna del código, la cual se enfoca solamente en las entradas y salidas, únicamente nos enfocamos en los requerimientos de software y especificaciones funcionales.



Tabla 11

Prueba de Caja negra del módulo usuario, seguridad y acciones

Nº	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	¿ El formulario de login para iniciar sesión tiene validadores?	Cumple
2	¿ Después de que el usuario inicia sesión nos direcciona al menú principal?	Cumple
3	¿ Cuando ingresa al módulo usuario registra nuevos usuarios y asignando roles?	Cumple
4	¿ Al registrar un nuevo usuario es validado en la base de datos?	Cumple
5	¿ Las acciones del usuario son registrados, cumple?	Cumple

Fuente: Elaboración propia

### 3.10.3. Resultados

Después de codificar, se realizó las pruebas necesarias al Módulo de Usuario y de seguridad en el momento de la implementación pasando primero por la prueba de integración, luego se aplicaron las pruebas de caja blanca y caja negra, obteniendo resultados satisfactorios.

### 3.11. Fase 3 Análisis y diseño - Módulo de Productos

Es considerado como de prioridad alto la cual es Crear el Registro de productos con todas las acciones necesarias de CRUD, se diseñó los diagramas que UML nos proporciona en cada requisito, para luego pasar a la codificación.

Tabla 12

Módulo Productos

REQUISITO	TAREA	Inicio	Fin	Duración
		Desde	Hasta	Estado
Crear registro de productos.	Diseño del diagrama UML	01/02/2023	02/05/2023	Completado
	Codificación	03/05/2023	05/05/2023	Completado
	Pruebas	06/05/2023	06/05/2023	Completado

Fuente: Elaboración propia



### 3.11.1. Caso de Uso

En la siguiente tabla se describe como realiza el registro de productos

Tabla 13

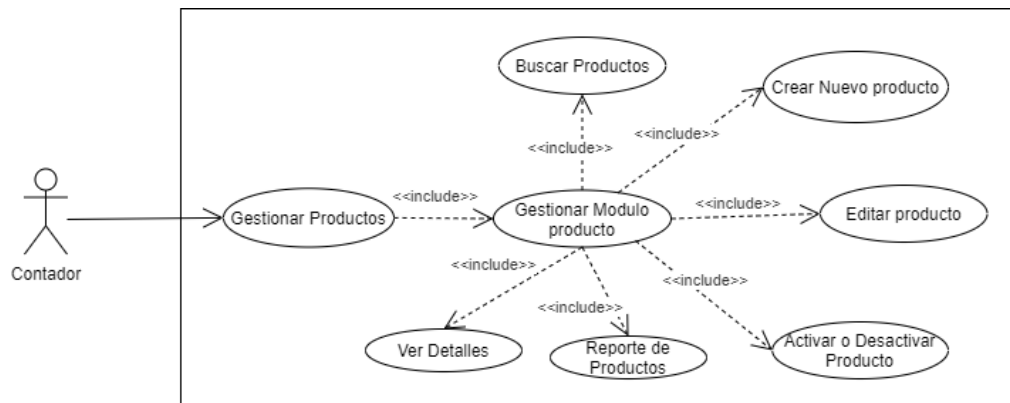
Descripción de caso de uso del módulo productos

Nro.	Descripción
1	Registro y actualización productos
2	Y las acciones necesarias que contendrá

Fuente: Elaboración propia

Figura 18

Caso de uso del módulo productos



Fuente: Elaboración propia



Tabla 14

Inicio de sesión, ingresar al módulo de productos

Nombre.	Registrar y actualizar productos
Actores	Usuario, Sistema.
Propósitos	Para que el usuario pueda registrar un nuevo producto deberá estar autorizado y con el rol asignado e iniciar sesión en el sistema. <ul style="list-style-type: none"> <li>Evento del actor: deberá iniciar sesión con el nombre de usuario y la contraseña</li> </ul>
Flujo principal	<ul style="list-style-type: none"> <li>Evento del sistema: Valida ambos parámetros enviados, ingresa al sistema con los permisos correspondientes y puede ingresar al módulo de productos para que pueda realizar los debidos registros.</li> </ul>
Flujo alternativo	Si no se encuentra el usuario o contraseña no coinciden, se manda un mensaje "Usuario o Contraseña incorrecta"
Precondición	El usuario debe estar registrado en la base de datos, tanto su contraseña.
Poscondición	Solo un usuario autorizado ingresa al sistema.
Presunción	Si el usuario no se encuentra registrado no podrá ingresar al sistema.

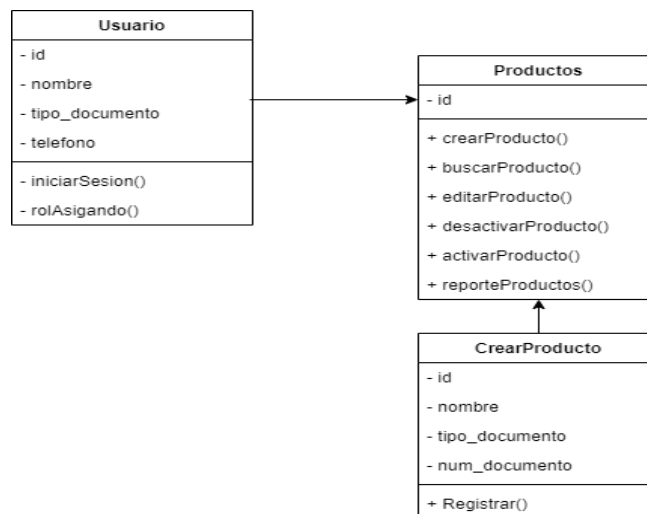
Fuente: Elaboración propia

### 3.11.2. Diagrama de clases

Se diseñó el diagrama de clases para el requerimiento registrar productos.

Figura 19

Diagrama de clases del Módulo de productos



Fuente: Elaboración propia

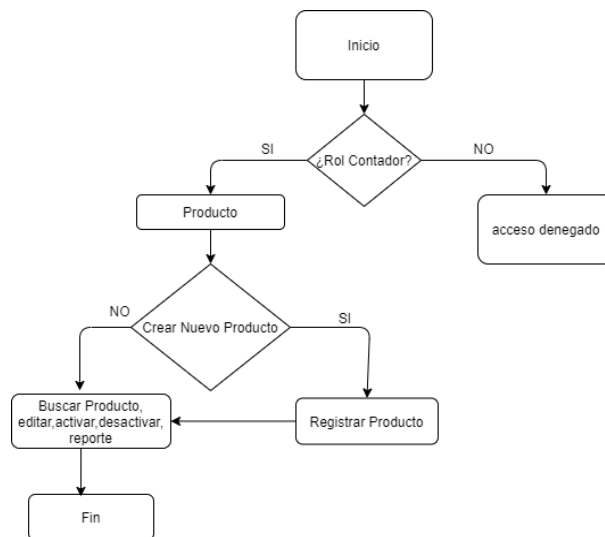


### 3.11.3. Diagrama de actividades o de flujo

Se diseñó el diagrama de actividades o flujo para registrar modulo productos, el cual se observa en la siguiente figura:

Figura 20

Diagrama de actividades o flujo modulo productos



Fuente: Elaboración propia

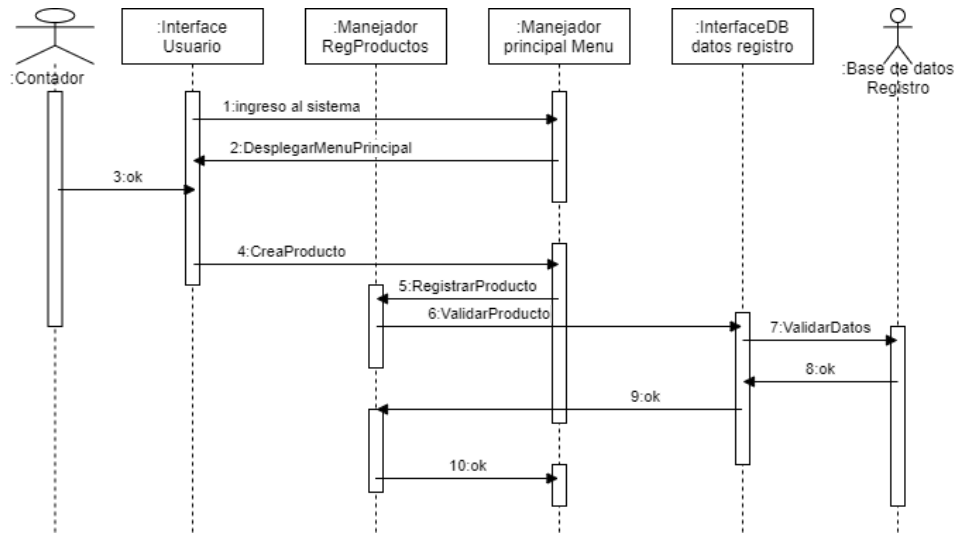
### 3.11.4. Diagrama de secuencia

Se diseñó el diagrama de secuencias para los requerimientos del módulo de registro de productos / actualizar como se observa en la siguiente figura:



Figura 21

Diagrama de secuencia módulo de productos



Fuente: Elaboración propia

### 3.12. Fase 4 Programación - Módulo de Productos

Se obtiene el siguiente resultado como se observa en la siguiente figura.





Figura 22

Entornos gráficos del módulo de productos/Registros

Crear Nuevo Producto

Código: Ingrese el Código

Stock: 0

Nombre: Ingrese el nombre de producto

Tipo de Característica: Fórmula vegana, libre de parabenos, de sulfatos, de ingredientes de origen animal

Tipo de Producto: Crema Corporal

Tipo de Línea: Body Evolution

Tipo de Destinatario: Niño

Precio Venta Unitario \$us.: Ingrese el precio venta

Precio Promoción \$us.: Ingrese el precio de promoción

Precio Asesor \$us.: Ingrese el precio de asesor

Precio Asesor Bs.: Ingrese el precio de asesor en Bs.

Imagen: Examinar... No se ha seleccionado ningún archivo.

ADMINISTRACION DE PRODUCTOS

Tabla de Productos Registrados

Mostrar 11 registros

N°	CODIGO	PRODUCTO
1	98128	MAQUILLAJE FIN CMA PORCHAIN CL-FAR AGH-FSS HX

Reporta PDF + Crear Nuevo Producto

Buscar:

Tipo Característica: Base de maquillaje de cobertura de moderada a intensa, textura cremosa fácil aplicación la cual se edifica y ayuda a disminuir imperfecciones y líneas de expresión. Fórmula con anti

Tipo Producto: Maquillaje

Tipo de Línea: Hegan77a

Tipo Destinatario: Mujer

Precio Venta \$us.: 27.50

Precio Promoción \$us.: 27.50

Precio Asesor \$us.: 16.50

Precio Asesor Bs. Hs.: 114.84

Stock: 0

Imagen: NO IMAGEN PRODUCTO

Fuente: Elaboración propia

### 3.13. Fase 5 Implementación y pruebas – Módulo de productos

Se realiza la implementación una vez finalizado la programación, posteriormente se realizan las pruebas de caja blanca y negra.

#### 3.13.1. Prueba de Caja Blanca

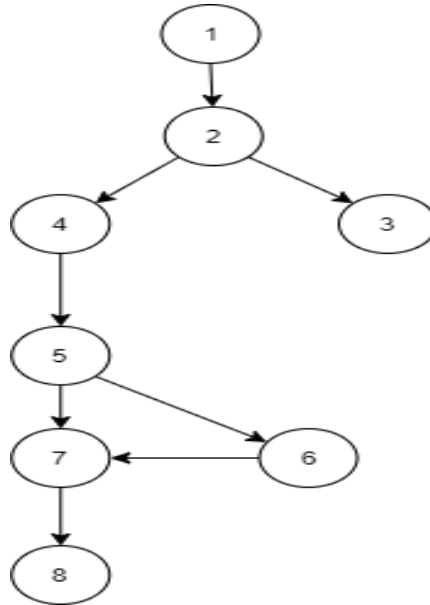


La prueba que se realizó para este módulo es calcular la complejidad ciclomática del diagrama de flujo de la figura 19.



Figura 23

Grafo de flujo m ó dulo de productos



Fuente: Elaboraci ó n propia

Tabla 15

C á lculo de complejidad

NODOS	ARISTAS	REGIONES	NODOS PREDICADO
8	8	2	1
$V(G) = \text{Aristas} - \text{Nodos} + 2$		$V(G) = 8 - 8 + 2 = 2$	
$V(G) = \text{Nodos Predicados} + 1$		$V(G) = 2 + 1 = 3$	
$V(G) = \text{Regiones}$		$V(G) = 2$	
Complejidad ciclom á tica de 3 que nos indica realizar 3 pruebas			

Fuente: Elaboraci ó n propia



Tabla 16

Caminos independientes

CAMINO	ENTRADA DATOS	PRUEBA	SALIDA
1,2,4,5,6,7,8	->Rol Almacén	-> Si usuario debe estar registrado y tener el rol de Almacén. -> Si usuario está registrado, usuario logueado puede registrar nuevos productos.	Registra nuevos productos con éxito
1,2,4,5,7,8	-> Rol Almacén	->Si es almacenero ->No crea nuevos productos	Buscar productos, editar, desactivar, desactivar y reporte.
1,2,3	-> Rol Almacén	->No es almacenero acceso denegado	Acceso denegado

Fuente: Elaboración propia

### 3.13.2. Prueba de Caja Negra

En esta prueba únicamente nos enfocamos en los requerimientos de software y especificaciones funcionales.

Tabla 17

Resultados de la implementación del módulo de productos

Nº	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	¿ Solo un usuario autorizado puede registrar los productos?	Cumple
2	¿ Después de que el usuario inicia sesión puede ingresar al módulo de productos?	Cumple
3	¿ Cuando ingresa al módulo productos puede registrar?	Cumple
4	¿ Al registrar un producto es validado en la base de datos?	Cumple
5	¿ Tiene el CRUD necesario el módulo productos, cumple?	Cumple

Fuente: Elaboración propia

### 3.13.3. Resultados

Después de codificar, se realizó las pruebas necesarias al Módulo de productos en el momento de la implementación pasando primeramente por la prueba de integración, luego se aplicaron las pruebas de caja blanca y caja negra, obteniendo resultados satisfactorios.



### 3.14. Fase 3 Análisis y diseño - Módulo de clientes

Es considerado como de prioridad alto la cual es Crear el Registro de Clientes con todas las acciones necesarias de CRUD, se diseñó los diagramas que UML nos proporciona en cada requisito, para luego pasar a la codificación.

Tabla 18

Módulo clientes

REQUISITO	TAREA	Inicio	Fin	Duración
		09/05/2023	13/05/2023	5 días
		Desde	Hasta	Estado
Crear registro de clientes.	Diseño del diagrama UML	09/05/2023	10/05/2023	Completado
	Codificación	11/05/2023	12/05/2023	Completado
	Pruebas	13/05/2023	13/05/2023	Completado

Fuente: Elaboración propia

#### 3.14.1. Caso de Uso

En la siguiente tabla se describe como realiza el registro de clientes.

Tabla 19

Descripción de caso de uso del módulo cliente

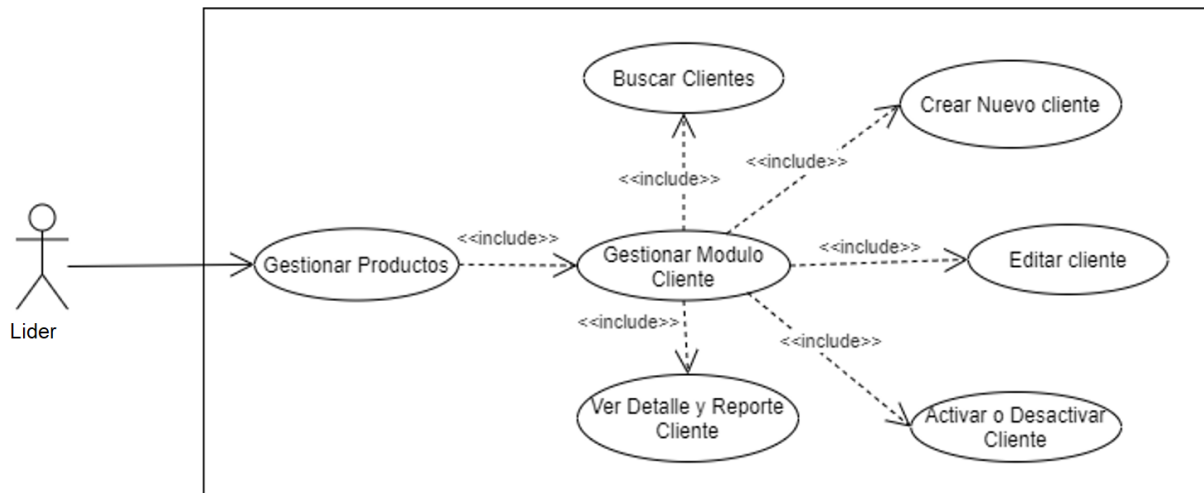
Nro.	Descripción
1	Registro y actualización de clientes
2	Y las acciones necesarias que contendrá

Fuente: Elaboración propia



Figura 24

Caso de uso del M ó dulo cliente



Fuente: Elaboraci ó n propia

Figura 25

Inicio de sesi ó n, ingresar al m ó dulo cliente

Nombre.	Registrar y actualizar plan de cuentas
Actores	Usuarios l í der y asesora
Prop ó sitios	Para que el usuario pueda registrar un nuevo cliente deber á estar autorizado y con el rol asignado e iniciar sesi ó n en el sistema.
Flujo principal	<ul style="list-style-type: none"> <li>Evento del actor: deber á iniciar sesi ó n con el nombre de usuario y la contrase ñ a</li> <li>Evento del sistema: Valida ambos par á metros enviados, ingresa al sistema con los permisos correspondientes y puede ingresar al m ó dulo de clientes para que pueda realizar los debidos registros.</li> </ul>
Flujo alternativo	Si no se encuentra el usuario o contrase ñ a no coinciden, se manda un mensaje "Usuario o Contrase ñ a incorrecta"
Precondici ó n	El usuario debe estar registrado en la base de datos, tanto su contrase ñ a.
Pos condici ó n	Solo un usuario autorizado ingresa al sistema.
Presunci ó n	Si el usuario no se encuentra registrado no podr á ingresar al sistema.

Fuente: Elaboraci ó n propia

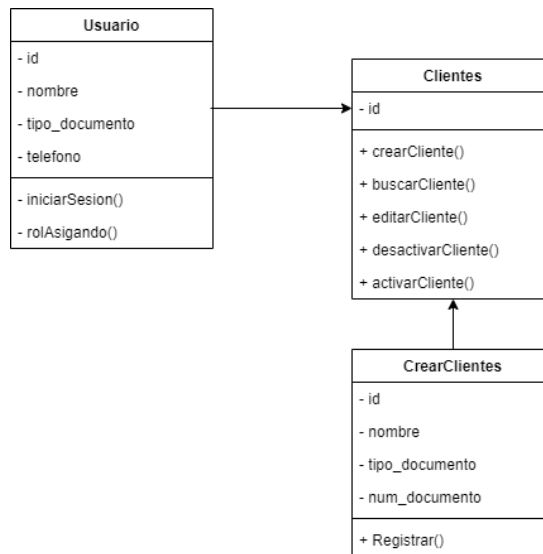
### 3.14.2. Diagrama de clases

Se dise ñ a el diagrama de clases para el requerimiento Registrar clientes



Figura 26

Diagrama de clases del módulo de cliente



Fuente: Elaboración propia

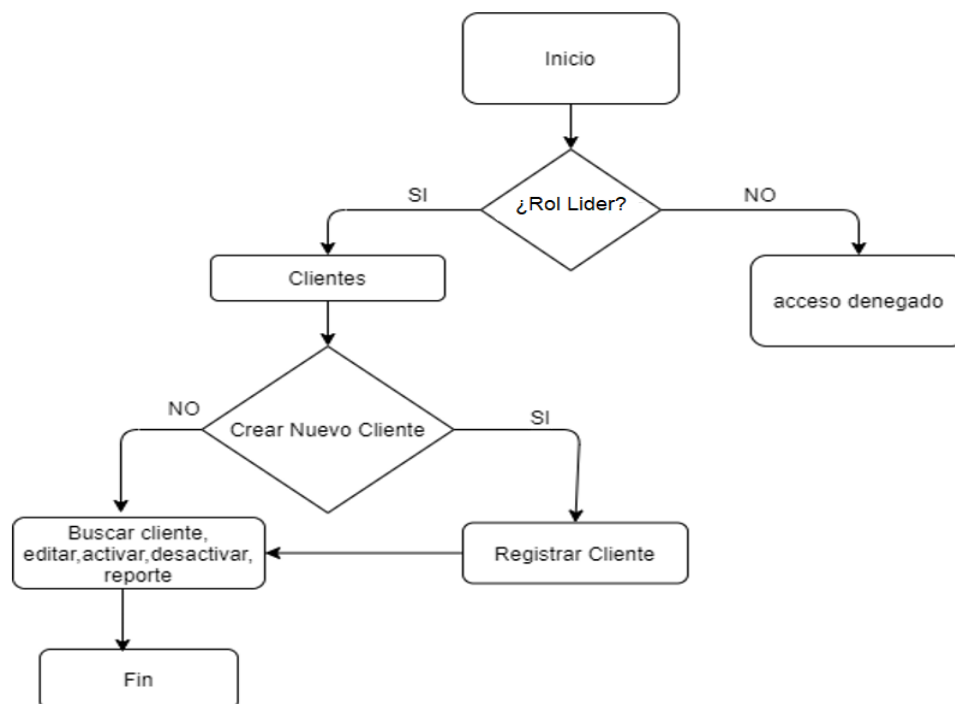
### 3.14.3. Diagrama de actividades o de flujo

Se diseña el diagrama de actividades o flujo para registrar modulo cliente, el cual se observa en la siguiente figura:



Figura 27

Diagrama de actividades o flujo m ó dulo cliente



Fuente: Elaboración propia

#### 3.14.4. Diagrama de secuencia

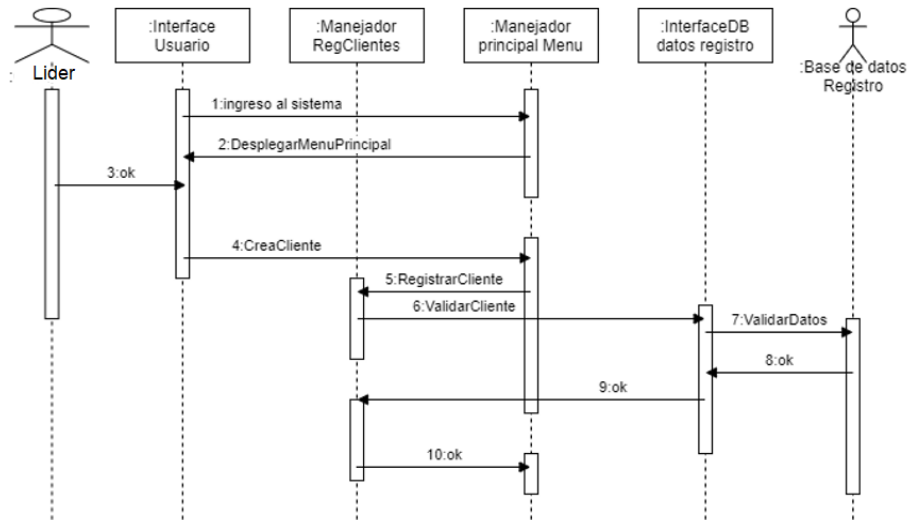
Se diseñ ó el diagrama de secuencias para los requerimientos del m ó dulo de registro de clientes / actualizar como se observa en la siguiente figura:





Figura 28

Diagrama de secuencia módulo de registro cliente



Fuente: Elaboración propia

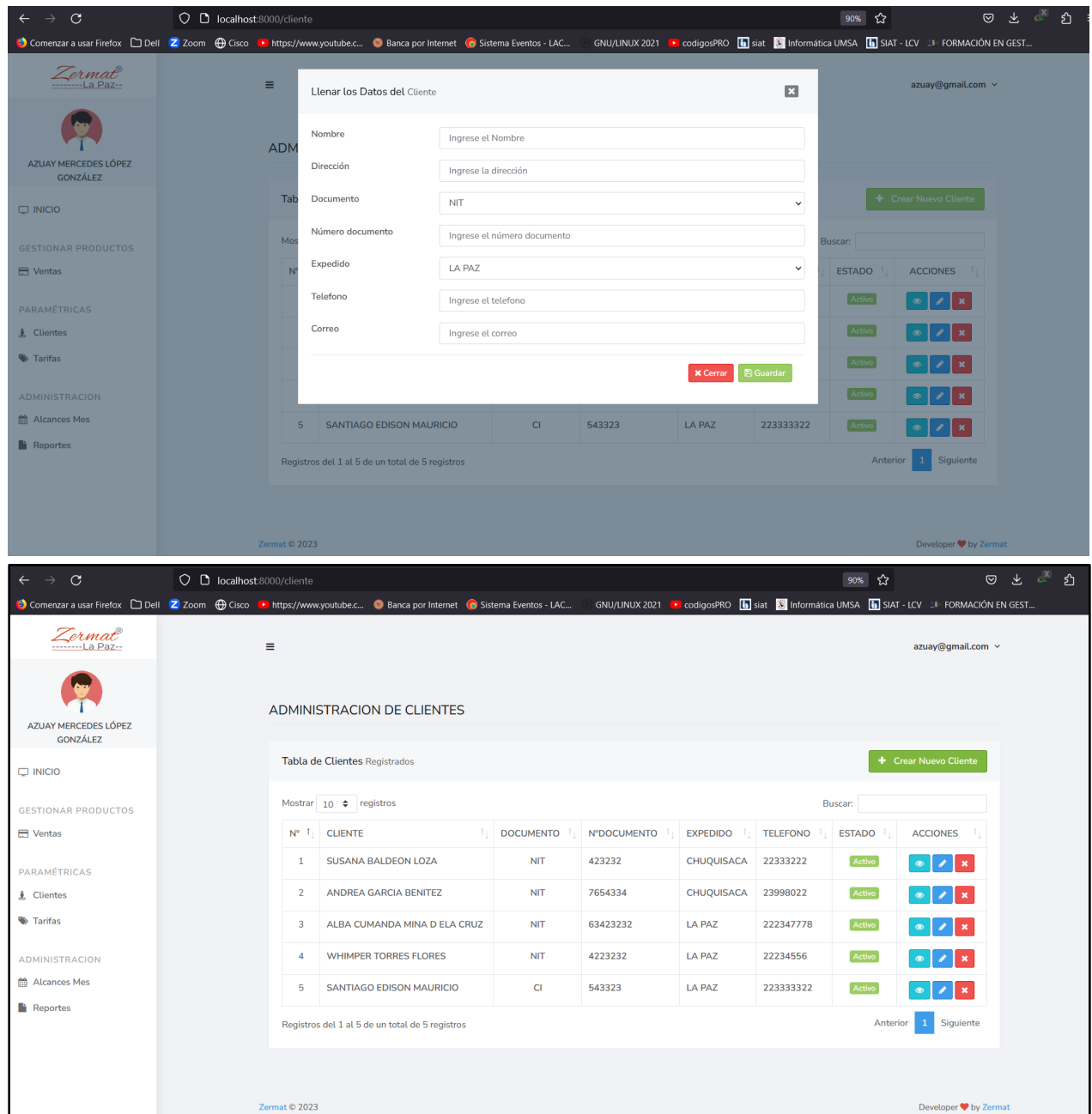
### 3.15. Fase 4 Programación - Módulo de clientes

Se obtiene el siguiente resultado como se observa en la siguiente figura.



Figura 29

Entornos gráficos del Módulo de clientes/registros



Fuente: Elaboración propia

### 3.16. Fase 5 Implementación y pruebas – Módulo de Clientes

Se realiza la implementación una vez finalizado la programación como se puede observar en las figuras N<sup>o</sup> 3.20, posteriormente se realizan las pruebas de caja blanca y negra.

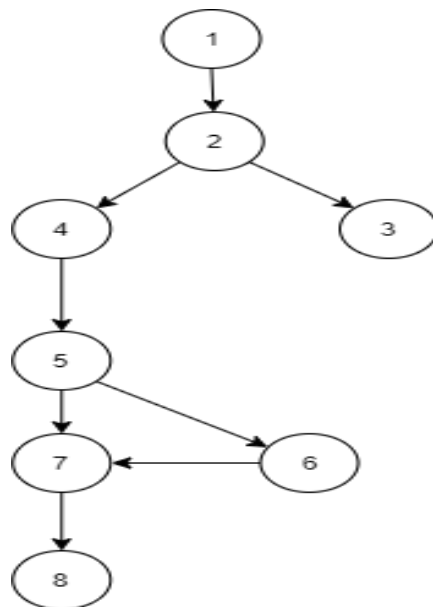


### 3.17. Prueba de Caja Blanca

La prueba que se realizó para este módulo es calcular la complejidad ciclomática del diagrama de flujo de la figura N<sup>o</sup> 27.

Figura 30

Grafo de flujo módulo cliente



Fuente: Elaboración propia

Tabla 20

Cálculo de complejidad

NODOS	ARISTAS	REGIONES	NODOS PREDICADO
8	8	2	1

$V(G) = \text{Aristas} - \text{Nodos} + 2$   $V(G) = 8 - 8 + 2 = 2$



$$V(G) = \text{Nodos Predicados} + 1$$

$$V(G) = 2 + 1 = 3$$

$$V(G) = \text{Regiones}$$

$$V(G) = 2$$

Complejidad ciclomática de 3 que nos indica realizar 3 pruebas

Fuente: Elaboración propia

Tabla 21

Caminos independientes

CAMINO	ENTRADA DATOS	PRUEBA	SALIDA
1,2,4,5,6,7,8	-> Rol Líder	-> Si usuario debe estar registrado y tener el rol de líder. -> Si usuario está registrado, usuario conectado puede registrar nuevo cliente.	Registra nuevo cliente con éxito
1,2,4,5,7,8	-> Rol Líder	-> Si es líder -> No registra nuevo cliente	Buscar cliente, editar, desactivar, activar y reporte
1,2,3	-> Rol Líder	-> No es líder acceso denegado	Acceso denegado

Fuente: Elaboración propia

### 3.17.1. Prueba de Caja Negra

En esta prueba únicamente nos enfocamos en los requerimientos de software y especificaciones funcionales.

Tabla 22

Resultados de la implementación del módulo cliente

Nº	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	¿ Solo un usuario autorizado puede registrar clientes?	Cumple
2	¿ Después de que el usuario inicia sesión puede ingresar al módulo del cliente?	Cumple
3	¿ Cuando ingresa al módulo de cliente puede generar reporte?	Cumple
4	¿ Al registrar un nuevo cliente es validado en la base de datos?	Cumple
5	¿ Tiene el CRUD necesario el módulo cliente, cumple?	Cumple

Fuente: Elaboración propia



### 3.17.2. Resultados

Después de codificar, se realizó las pruebas necesarias al Módulo de cliente en el momento de la implementación pasando primeramente por la prueba de integración, luego se aplicaron las pruebas de caja blanca y caja negra, obteniendo resultados satisfactorios.

### 3.18. Fase 3 Análisis y diseño - Módulo de compras

Es considerado como de prioridad alto la cual es Crear el Registro de las compras con todas las acciones necesarias de CRUD, se diseñó los diagramas que UML nos proporciona en cada requisito, para luego pasar a la codificación.

Tabla 23

Módulo de Compras

REQUISITO	TAREA	Inicio	Fin	Duración
		15/05/2023	18/05/2023	4 días
		Desde	Hasta	Estado
Crear registro de compras	Diseño del diagrama UML	15/05/2023	15/05/2023	Completado
	Codificación	16/05/2023	17/05/2023	Completado
	Pruebas	18/05/2023	18/05/2023	Completado

Fuente: Elaboración propia

#### 3.18.1. Caso de Uso

En la siguiente tabla se describe como realiza el registro de compras.

Tabla 24

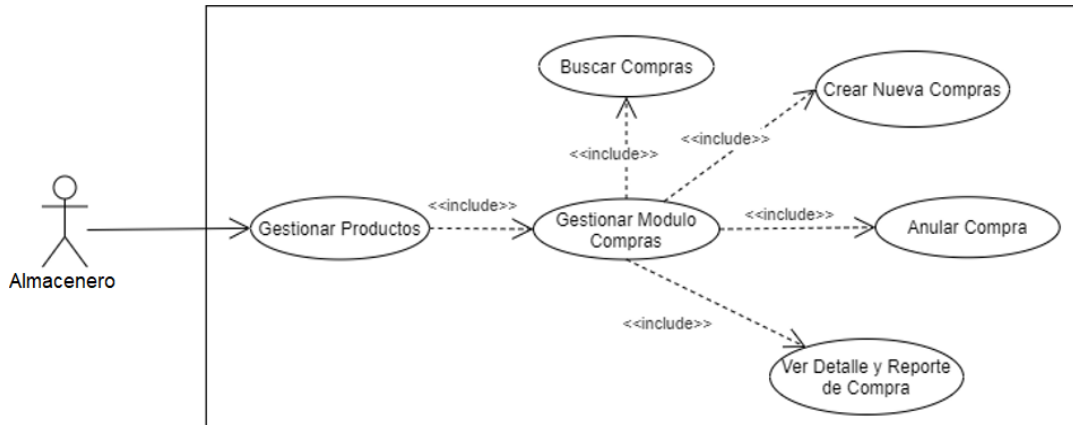
Descripción de caso de uso del Módulo de compras

Nº	Descripción
1	Registro y anulación las compras
2	Las acciones necesarias que contendrá

Fuente: Elaboración propia



Figura 31  
Caso de uso del M ó dulo de compras



Fuente: Elaboraci ó n propia

Tabla 25

Inicio de sesi ó n, ingresar al m ó dulo de compras

Nombre.	Registrar y actualizar plan de cuentas
Actores	Almacenero
Prop ó sitios	Para que el usuario pueda registrar una nueva compra deber á estar autorizado y con el rol asignado e iniciar sesi ó n en el sistema. <ul style="list-style-type: none"> <li>• Evento del actor: deber á iniciar sesi ó n con el nombre de usuario y la contrase ñ a</li> </ul>
Flujo principal	<ul style="list-style-type: none"> <li>• Evento del sistema: Valida ambos par á metros enviados, ingresa al sistema con los permisos correspondientes y puede ingresar al m ó dulo de compras para que pueda realizar los debidos registros.</li> </ul>
Flujo alternativo	Si no se encuentra el usuario o contrase ñ a no coinciden, se manda un mensaje "Usuario o Contrase ñ a incorrecta"
Precondici ó n	El usuario debe estar registrado en la base de datos, tanto su contrase ñ a.
Pos condici ó n	Solo un usuario autorizado ingresa al sistema.
Presunci ó n	Si el usuario no se encuentra registrado no podr á ingresar al sistema.

Fuente: Elaboraci ó n propia

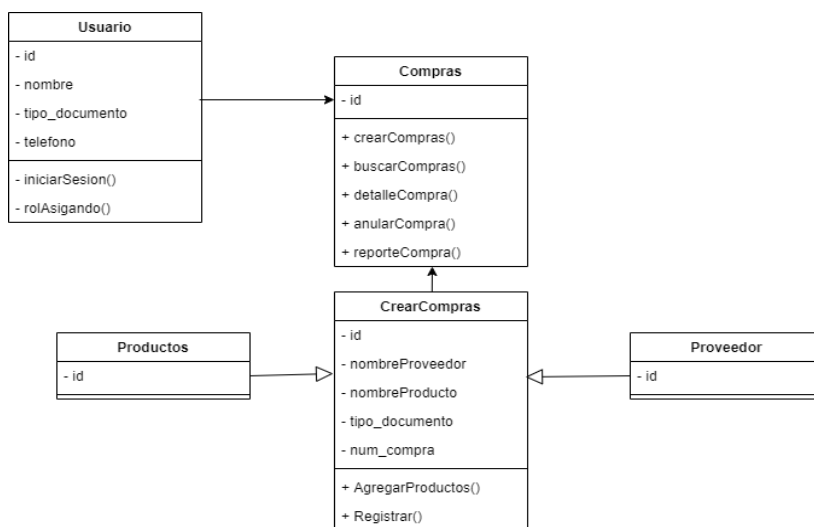
### 3.18.2. Diagrama de clases



Se diseña el diagrama de clases para el requerimiento Registrar compras.

Tabla 26

Diagrama de clases del módulo de compras



Fuente: Elaboración propia

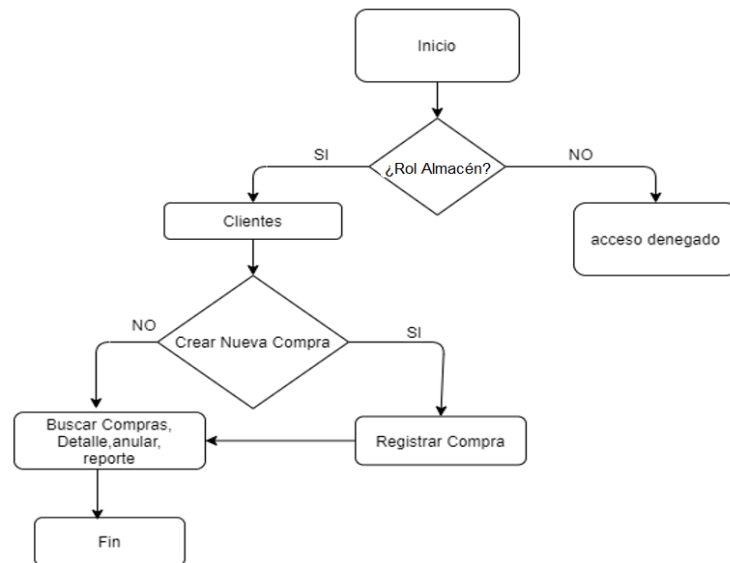
### 3.18.3. Diagrama de actividades o de flujo

Se diseña el diagrama de actividades o flujo para registrar módulo de compras y autenticación del usuario, el cual se observa en la siguiente figura:



Figura 32

Diagrama de actividades o flujo m ó dulo de compras



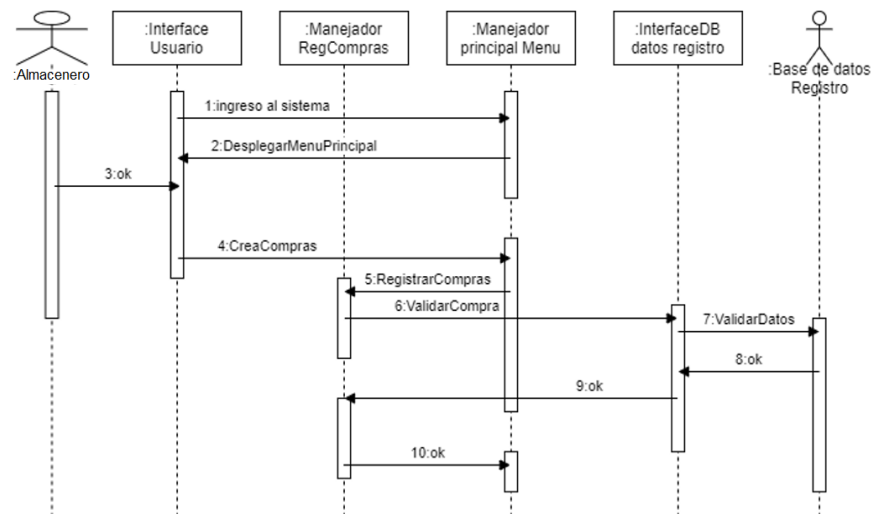
Fuente: Elaboraci ó n propia

### 3.18.4. Diagrama de secuencia

Se dise ñ a el diagrama de secuencias para los requerimientos del m ó dulo de registro de compras de acuerdo a la siguiente figura:

Figura 33

Diagrama de secuencia m ó dulo de compra





Fuente: Elaboración propia

### 3.19. Fase 4 Programación – Módulo de compras

Se obtiene el siguiente resultado como se observa en la siguiente figura.



Figura 34

Entornos gráficos del Módulo de compras

**ADMINISTRAR NUEVAS COMPRAS**

Tabla de Registro de Compras

Nota: Todos los campos con (\*) son obligatorios

Número de Compra (\*)  Documento (\*)

Nombres Paterno Materno: Proveedor (\*)  Producto (\*)

Destinados a:  Línea  Tipo de Producto

Características

Cantidad (\*)  Precio de Compra por Unidad (\*)

[+ Agregar Detalle Compra](#)

DETALLE DE COMPRAS DE LOS DIVERSOS PRODUCTOS

ELIMINAR	PRODUCTO	PRECIO (BS)	CANTIDAD	SUBTOTAL (BS)
<input type="button" value="X"/>	93027 MAQUILLAJE LIQUIDO HUMECTANTE 30 ML BRONZE	<input type="text" value="98.14"/>	<input type="text" value="5"/>	Bs. 490.7
<b>TOTAL:</b>				Bs. 490.70
<b>TOTAL IVA (13%):</b>				Bs. 63.79

**ADMINISTRACION DE COMPRAS**

Tabla de Compras Registradas [+ Nueva Compra](#)

Mostrar  registros

N°	FECHA	PROVEEDOR	N°COMPRA	IVA	TOTAL	COMPRADOR	ESTADO	ACCIONES
1	06 de noviembre 2023	Zermat México	2	0.13	Bs. 5,323.11	ALAN MONTALVO QUISPE	Registrado	<input type="button" value="V"/> <input type="button" value="X"/> <input type="button" value="E"/>
2	06 de noviembre 2023	Zermat México	1	0.13	Bs. 9,363.87	ALAN MONTALVO QUISPE	Registrado	<input type="button" value="V"/> <input type="button" value="X"/> <input type="button" value="E"/>

Registros del 1 al 2 de un total de 2 registros Anterior  Siguiente

Zermat © 2023 Developer  by Zermat

Fuente: Elaboración propia



### 3.20. Fase 5 Implementación y pruebas – Módulo de compras

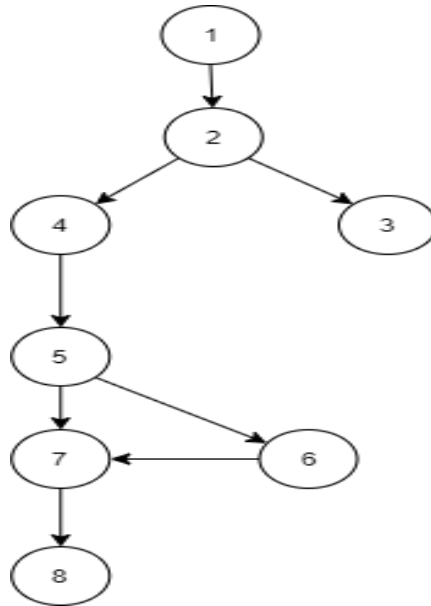
Se realiza la implementación una vez finalizado la posteriormente se realizan las pruebas de caja blanca y negra.

#### 3.20.1. Prueba de Caja Blanca

La prueba que se realizó para este módulo es calcular la complejidad ciclomática del diagrama de flujo de la figura N<sup>o</sup> 3.26.

Figura 35

Grafo de flujo módulo de compras



Fuente: Elaboración propia



Tabla 27

*Cálculo de complejidad*

NODOS	ARISTAS	REGIONES	NODOS PREDICADO
8	8	2	1
$V(G) = \text{Aristas} - \text{Nodos} + 2$		$V(G) = 8 - 8 + 2 = 2$	
$V(G) = \text{Nodos Predicados} + 1$		$V(G) = 2 + 1 = 3$	
$V(G) = \text{Regiones}$		$V(G) = 2$	
Complejidad ciclomática de 3 que nos indica realizar 3 pruebas			

Fuente: Elaboración propia

Tabla 28

*Caminos independientes*

CAMINO	ENTRADA DATOS	PRUEBA	SALIDA
1,2,4,5,6,7,8	-> Rol Almacén	-> Si usuario debe estar registrado y tener el rol de almacenero. -> Si usuario está registrado, usuario conectado puede registrar nuevas compras.	Registra nueva compra con éxito
1,2,4,5,7,8	-> Rol Almacén	-> Si es almacenero -> No realiza una compra	Buscar compras, detalle, anular y reporte
1,2,3	-> Rol Almacén	-> No es almacenero acceso denegado	Acceso denegado

Fuente: Elaboración propia

### 3.20.2. Prueba de Caja Negra

En esta prueba únicamente nos enfocamos en los requerimientos de software y especificaciones funcionales.



Tabla 29

Resultados de la implementación del módulo de compras

Nº	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	¿ Solo un usuario autorizado puede registrar nuevas compras?	Cumple
2	¿ Después de que el usuario inicia sesión puede ingresar al módulo de compras?	Cumple
3	¿ Cuando ingresa al módulo de compras puede ver los detalles y generar reporte?	Cumple
4	¿ Al registrar una nueva compra es validado en la base de datos?	Cumple
5	¿ Tiene el CRUD necesario el módulo de compras, cumple?	Cumple

Fuente: Elaboración propia

### 3.2.0.3. Resultados

Después de codificar, se realizó las pruebas necesarias al Módulo de compras en el momento de la implementación pasando primeramente por la prueba de integración, luego se aplicaron las pruebas de caja blanca y caja negra, obteniendo resultados satisfactorios.

### 3.2.1. Fase 3 Análisis y diseño - Módulo de ventas

Es considerado como de prioridad alto la cual es Crear el Registro de ventas con todas las acciones necesarias de CRUD, se diseñó los diagramas que UML nos proporciona en cada requisito, para luego pasar a la codificación.

Tabla 30

Módulo de ventas

REQUISITO	TAREA	Inicio	Fin	Duración
		Desde	Hasta	Estado
Crear registro de ventas.	Diseño del diagrama UML	01/06/2023	02/06/2023	4 días
	Codificación	05/06/2023	05/06/2023	Completado
	Pruebas	06/06/2023	06/06/2023	Completado

Fuente: Elaboración propia

#### 3.2.1.1. Caso de Uso



En la siguiente tabla se describe como realiza el registro de ventas.

Tabla 31

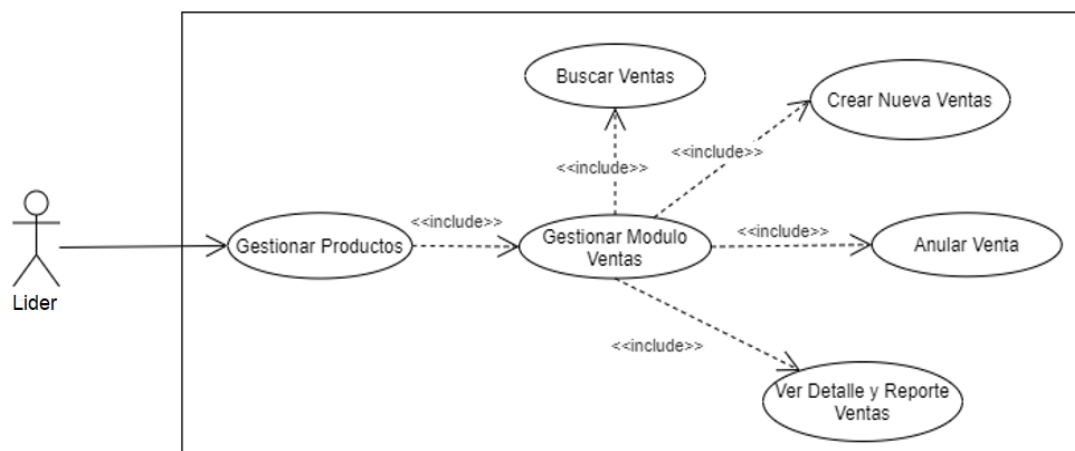
Descripción de caso de uso del Módulo de ventas

Nº	Descripción
1	Registro y anulación de las ventas
2	Las acciones necesarias que contendrá

Fuente: Elaboración propia

Figura 36

Descripción de caso de uso del Módulo de ventas



Fuente: Elaboración propia



Tabla 32

Inicio de sesión, ingresar al módulo de ventas

Nombre.	Registrar y anular ventas
Actores	Usuarios del Sistema
Propósitos	Para que el usuario pueda registrar una nueva venta o anular deberá estar autorizado y con el rol asignado e iniciar sesión en el sistema. <ul style="list-style-type: none"> <li>Evento del actor: deberá iniciar sesión con el nombre de usuario y la contraseña</li> </ul>
Flujo principal	<ul style="list-style-type: none"> <li>Evento del sistema: Valida ambos parámetros enviados, ingresa al sistema con los permisos correspondientes y puede ingresar al módulo de ventas para que pueda realizar los debidos registros.</li> </ul>
Flujo alternativo	Si no se encuentra el usuario o contraseña no coinciden, se manda un mensaje "Usuario o Contraseña incorrecta"
Precondición	El usuario debe estar registrado en la base de datos, tanto como su contraseña.
Pos condición	Solo un usuario autorizado ingresa al sistema.
Presunción	Si el usuario no se encuentra registrado no podrá ingresar al sistema.

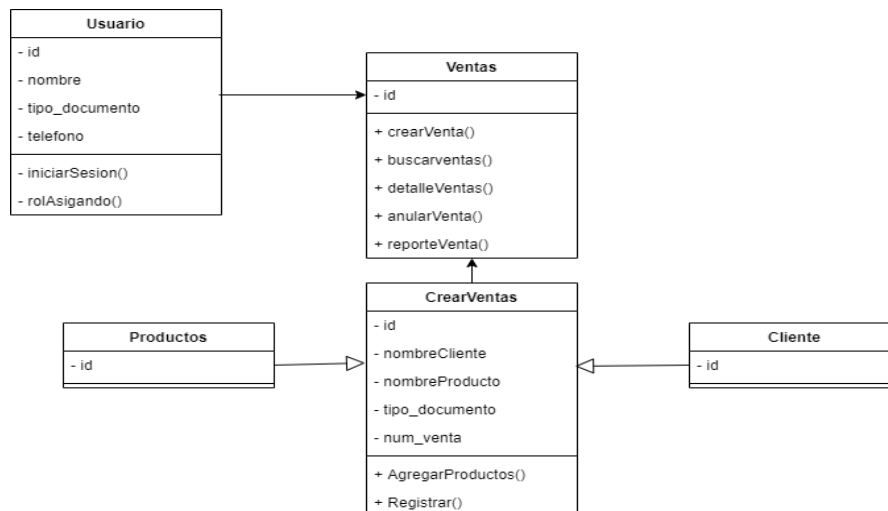
Fuente: Elaboración propia

### 3.2.1.2. Diagrama de clases

Se diseñó el diagrama de clases para el requerimiento Registrar ventas realizadas.

Figura 37

Diagrama de clases del Módulo de ventas



Fuente: Elaboración propia

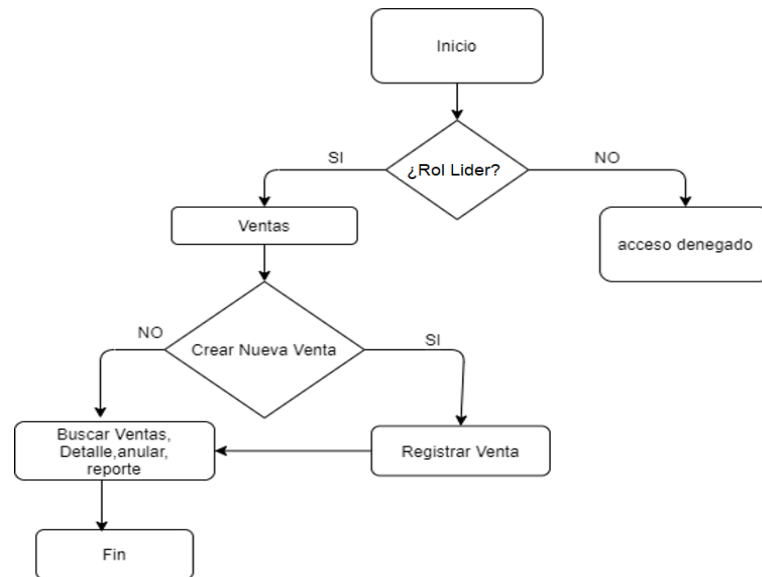


### 3.21.3. Diagrama de actividades o de flujo

Se diseñó el diagrama de actividades o flujo para registrar módulo de ventas, el cual se observa en la siguiente figura:

Figura 38

Diagrama de actividades o flujo módulo de ventas



Fuente: Elaboración propia

### 3.21.4. Diagrama de secuencia

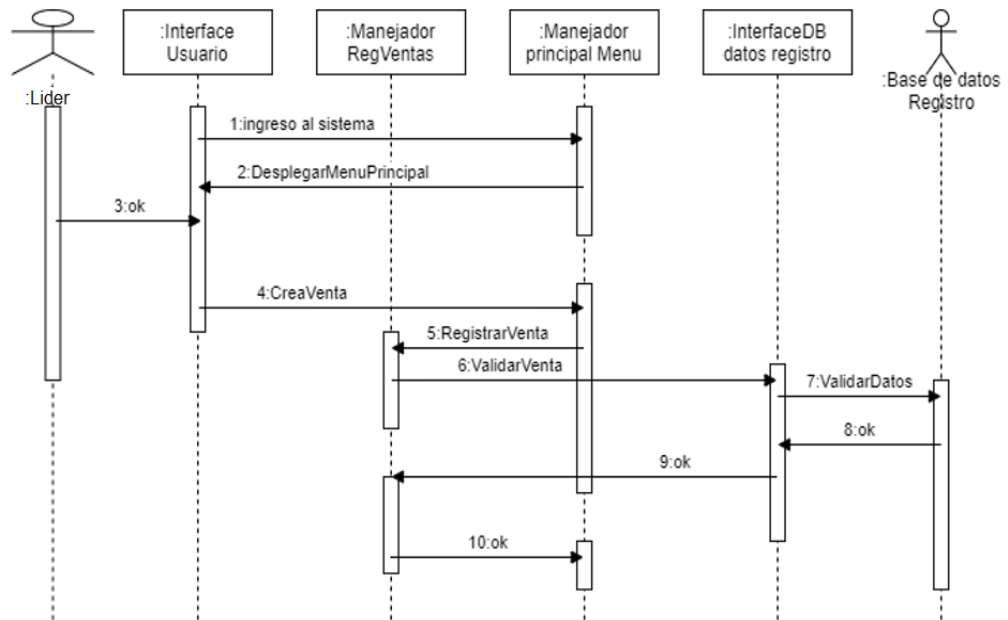
Se diseñó el diagrama de secuencias para los requerimientos del módulo de registro de nuevas compras como se observa en la siguiente figura:





Figura 39

Diagrama de secuencia módulo de registro ventas



Fuente: Elaboración propia

### 3.22.Fase 4 Programación - Módulo de ventas

Se obtiene el siguiente resultado como se observa en la siguiente figura.



Figura 40

Entornos gráficos del Módulo de ventas

Tabla de Registro de Ventas

Nota: Todos los campos con (\*) son obligatorios

Número de Venta: 3 Documento (\*): FACTURA

Nombres Paterno Materno: Cliente (\*): WHIMPER TORRES FLORES Producto (\*): 36132 POLVO VEGANO TONO TOFFEE ELEGANZZA CONT. NET. 10 G

Destinados a: Destinatarios Línea Tipo de línea Tipo de Producto Tipo de producto

Características: Breve descripción de producto

Cantidad (\*): Ingrese cantidad Stock: Stock Actual Precio Asesor Bs: Precio de venta asesor Descuento en %: 0

+ Agregar Detalle Venta

ELIMINAR	PRODUCTO	PRECIO ASESOR (BS.)	DESCUENTO	CANTIDAD	SUBTOTAL (BS.)
	36132 POLVO VEGANO TONO TOFFEE ELEGANZZA CONT. NET. 10 G	88,53	0,00	1	Bs. 88,53
TOTAL:					Bs. 88,53



localhost:8000/venta 110%

**Zermat**  
-----La Paz-----

AZUJAY MERCEDES LÓPEZ GONZÁLEZ

azuay@gmail.com

### ADMINISTRACION DE VENTAS

Tabla de Ventas Registrados Facturas Anulados Nueva Venta

Mostrar 10 registros

N°	FECHA	CLIENTE	N°VENTA	IVA	TOTAL	VENDEDOR
1	06 de noviembre 2023	WHIMPER TORRES FLORES	2	0.16	Bs. 102.69	AZUJAY MERCEDES LÓPEZ GONZÁLEZ
Estado <span>Registrado</span>						
Acciones <span>📄</span> <span>✖</span> <span>🔒</span>						
2	06 de noviembre 2023	SANTIAGO EDISON MAURICIO	1	0.16	Bs. 205.39	AZUJAY MERCEDES LÓPEZ GONZÁLEZ

Registros del 1 al 2 de un total de 2 registros Anterior 1 Siguiente

Zermat © 2023 Developer by Zermat

Fuente: Elaboración propia



### 3.23. Fase 5 Implementación y pruebas – Módulo de Ventas

Se realiza la implementación una vez finalizado la programación como se puede observar en las figuras N.º 3.33, posteriormente se realizan las pruebas de caja blanca y negra.

#### 3.23.1. Prueba de Caja Blanca

La prueba que se realizó para este módulo es calcular la complejidad ciclomática del

Figura 41

Grafo de flujo módulo de ventas  
diagrama de flujo de la figura N.º 3.31.

Fuente: Elaboración propia

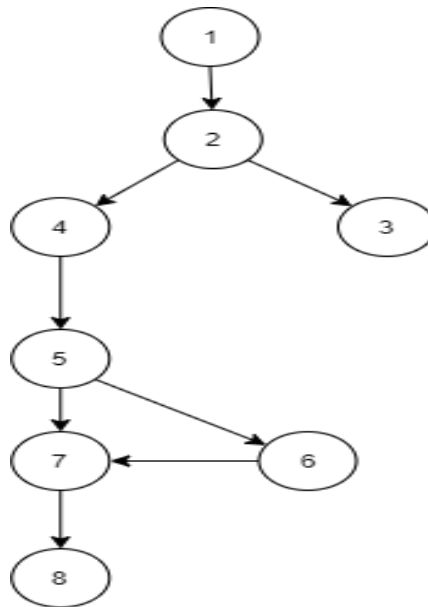


Tabla 33

Calculo de complejidad

NODOS	ARISTAS	REGIONES	NODOS PREDICADO
8	8	2	1
$V(G) = \text{Aristas} - \text{Nodos} + 2$		$V(G) = 8 - 8 + 2 = 2$	
$V(G) = \text{Nodos Predicados} + 1$		$V(G) = 2 + 1 = 3$	
$V(G) = \text{Regiones}$		$V(G) = 2$	
Complejidad ciclomática de 3 que nos indica realizar 3 pruebas			

Fuente: Elaboración propia

Tabla N<sup>o</sup> 3.37 Caminos independientes

Tabla 34

Caminos independientes

CAMINO	ENTRADA DATOS	PRUEBA	SALIDA
1,2,4,5,6,7,8	-> Rol L í der	-> Si usuario debe estar registrado y tener el rol de L í der. -> Si usuario está registrado, usuario conectado puede registrar nuevas ventas.	Registra nueva venta con éxito
1,2,4,5,7,8	-> Rol L í der	-> Si es contador -> No registra nueva venta.	Buscar ventas, detalle, anular y facturar
1,2,3	-> Rol L í der	-> No es L í der acceso denegado	Acceso denegado

Fuente: Elaboración propia

### 3.23.2. Prueba de Caja Negra

En esta prueba únicamente nos enfocamos en los requerimientos de software y especificaciones funcionales.



Tabla 35

Resultados de la implementación del módulo de ventas

N <sup>o</sup>	DESCRIPCIÓN DE LOS CASOS DE PRUEBA	RESULTADOS
1	¿ Solo un usuario autorizado puede registrar nuevas ventas?	Cumple
2	¿ Después de que el usuario inicia sesión puede ingresar al módulo de ventas?	Cumple
3	¿ Cuando el usuario ingresa al módulo de ventas puede emitir facturas?	Cumple
4	¿ Al registrar una nueva venta es validado en la base de datos?	Cumple
5	¿ Tiene el CRUD necesario el módulo de ventas, cumple?	Cumple

Fuente: Elaboración propia

### 3.2.3.3. Resultados

Después de codificar, se realizó las pruebas necesarias al Módulo de ventas en el momento de la implementación pasando primeramente por la prueba de integración, luego se aplicaron las pruebas de caja blanca y caja negra, obteniendo resultados satisfactorios.



CAPÍTULO IV



Editar con WPS Office

## 4. MÉTRICAS DE CALIDAD, SEGURIDAD Y ESTIMACIÓN DE COSTOS

### 4.1. Norma ISO/IEC 25010

#### 4.1.1. Adecuación funcional

Es la capacidad del software de proveer las funciones para satisfacer las necesidades explícitas e implícitas, este atributo del sistema no puede ser calculado de forma directa, por esa razón se utiliza el cálculo de métrica de punto función, que contiene cinco características de información. Los valores de información son definidos de la siguiente forma:

- **Número de entradas de usuario:** Cada entrada de usuario proporciona diferentes datos orientados a la aplicación. Las entradas se diferencian de las peticiones de forma separadas.
- **Número de salidas de usuario:** Cada salida proporciona al usuario información orientada a la aplicación. Las salidas se refieren a informes, pantallas, mensajes de error entre otros.
- **Número de peticiones de usuario:** Es una entrada interactiva que es producida por la respuesta del software inmediata en forma de salida interactiva. Las peticiones son contadas de forma separada.
- **Numero de archivos:** Un grupo lógico de datos que puede ser parte de una base de datos (tablas de base de datos).
- **Numero de interfaces externas:** Se cuenta todas las interfaces legibles que se utilizan para transmitir la información.

Para calcular los puntos función se emplea la siguiente fórmula:

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum F_i)$$

Donde:

PF: Medida de la adecuación funcional.





**Cuenta Total:** Es la suma de los siguientes datos: N° de entradas, N° de salidas, N° de peticiones, N° de archivos y N° de interfaces externas.

0.65: Confiabilidad del proyecto, varia del 1% al 100% (0 a 1).

0.01: Error mínimo aceptable de complejidad.

$F_i$ : Son los valores de ajuste de complejidad, donde  $(1 \leq i \leq 14)$ .

Analizando todas las interfaces que tiene el sistema se obtuvieron los siguientes datos:

**Tabla 36**

*Parámetros de medida y su cantidad*

NRO.	PARÁMETROS DE MEDIDA	CANTIDAD
1	N° de entradas de usuario	26
2	N° de salidas de usuario	16
3	N° de peticiones de usuario	31
4	N° de archivos	18
5	N° de interfaces externas	2

Fuente: Elaboración propia

Una vez obtenida la información se procede a calcular la cuenta total con el factor de ponderación media que se muestra en la siguiente tabla:

**Tabla 37**

*Parámetros de medida y su cantidad*

PARÁMETROS DE MEDIDA	CANTIDAD		FACTOR DE PONDERACIÓN		TOTAL
N° de entradas de usuario	26	*	4	=	10
N° de salidas de usuario	16	*	5	=	8
N° de peticiones de usuario	31	*	5	=	15
N° de archivos	18	*	10	=	18
N° de interfaces externas	2	*	7	=	1
			TOTAL	=	52

Fuente: Elaboración propia



La cuenta total de los puntos función obtenida se debe ajustar en función a las características ambientales del sistema. Los valores de ajuste de complejidad  $F_i$  basados en las respuestas a las preguntas formuladas de la siguiente tabla:

Tabla 38

Valores de ajuste de complejidad

NRO.	FACTORES						$F_i$
		0=Sin influencia	1=Incidental	2=Medio	3=Sgningificativo	4=Esencial	
1	¿ Requiere el sistema copias de seguridad y de recuperación fiables?					X	5
2	¿ Se requiere comunicación de datos?				X		4
3	¿ Existen funciones de procesos distribuidos?			X			3
4	¿ Es crítico el rendimiento?		X				2
5	¿ Sera ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?				X		4
6	¿ Requiere el sistema entrada de datos interactiva?				X		4
7	¿ Requiere la entrada de datos interactiva que se utilicen varias pantallas o varias operaciones?			X			3
8	¿ Se utilizaron los archivos maestros de forma interactiva?				X		4
9	¿ Son complejas las entradas, las salidas y/o peticiones?				X		4
10	¿ Es complejo el procesamiento interno?					X	5
11	¿ Se ha diseñado el código para ser reutilizable?			X			3
12	¿ Están incluidas en el diseño la conversión y la instalación?			X			3
13	¿ Se ha diseñado el sistema para soportar diferentes instalaciones en diferentes organizaciones?					X	5
14	¿ Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?					X	5
Factor ajuste de complejidad							54

Fuente: Elaboración propia

Una vez que se consiguió los valores correspondientes a las variables de la fórmula de los puntos función se procedió a realizar el cálculo del mismo.

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum F_i)$$

$$PF = 533 * (0.65 + 0.01 * 54)$$

$$PF = 533 * (1.19)$$



$$PF = 634.27$$

Para comparar los puntos función con su valor máximo, se calculó los puntos función con valores de ajuste de complejidad al máximo que es un total de 70:

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum F_i)$$

$$PF = 533 * (0.65 + 0.01 * 70)$$

$$PF = 533 * (1.35)$$

$$PF = 719.55$$

Después de haber calculado ambos valores se tiene que la funcionalidad real es:

$$\text{Adecuación Funcional} = \left( \frac{634.27}{719.55} \right) * 100\%$$

$$\text{Adecuación Funcional} = 88.15 \%$$

Este resultado demuestra que la funcionabilidad del sistema web es de un 88.15 %, esto quiere decir que el sistema tiene un 88.15 % de funcionar sin riesgos a fallar con una operatividad constante y un 11.85 % aproximadamente que suceda un colapso de sistema.

#### 4.1.2. Usabilidad

Es la capacidad del software que su comprensión sea fácil e intuitivo para el uso, satisfaciendo los requerimientos establecidos por el usuario, se realiza una evaluación en base a encuestas a 10 personas del uso del sistema, las cuales se evaluarán es ponderada al 100 %, los usuarios tienen el conocimiento de los procesos y los resultados son reflejados en la siguiente tabla:



Tabla 39

Encuesta sobre Usabilidad del Sistema

NRO.	PREGUNTAS	RESPUESTAS		% de SI
		SI	NO	
1	¿ Aprendí ó a usar rápido el sistema?	8	2	80
2	¿ Las vistas de pantalla que vio fueron de su agrado?	9	1	90
3	¿ Las pantallas que vio fueron de su agrado?	10	0	100
4	¿ El sistema responde rápido a sus solicitudes?	9	1	90
5	¿ El sistema le facilitara el trabajo?	9	1	90
6	¿ El sistema reduce su tiempo de trabajo?	10	0	100
7	¿ Es fácil navegar por las distintas opciones?	10	0	100
8	¿ Las operaciones que se realizan no son complicadas?	10	0	100
9	¿ El sistema le proporciono las respuestas requeridas?	9	1	90
10	¿ El sistema no presento errores?	9	1	90
El resultado de la usabilidad es:				93%

Fuente: Elaboración propia

#### 4.1.3. Fiabilidad

Es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas por cierto tiempo. Para la fiabilidad del sistema se considera, calcula la fiabilidad de cada módulo o subsistema de forma independiente.

Para calcular la fiabilidad de cada módulo se usó la fórmula:  $R(t) = e^{-\lambda t}$

Donde:

$R(t)$ : Fiabilidad de un componente o subsistema  $t$ .

$\lambda$ : Tasa de constantes de fallo ( $\lambda = N^\circ$  de fallas de acceso /  $N^\circ$  total de accesos al sistema).

$t$ : Periodo de operación de tiempo.

$e^{-\lambda t}$ : Probabilidad de falla de un componente o subsistema en el tiempo  $t$ .



$$e = 2,72$$

Luego de realizar pruebas de cada módulo en un tiempo de 4 horas continuas se obtiene la siguiente tabla:

**Tabla 40**

Valores de fiabilidad de cada módulo

MÓDULO	$\lambda$	t	R(t)
Compras	0,018	4 Horas	0,93
Ventas	0,018	4 Horas	0,93
Clientes	0,018	4 Horas	0,93
Proveedores	0,005	4 Horas	0,92
Productos	0,018	4 Horas	0,93
Tipo			
característica	0,005	4 Horas	0,92
Tipo destinatario	0,005	4 Horas	0,92
Tipo línea	0,005	4 Horas	0,92
Tipo producto	0,005	4 Horas	0,92
Tarifas	0,012	4 Horas	0,95
Usuarios	0,025	4 Horas	0,90
Roles	0,005	4 Horas	0,92
Rangos	0,005	4 Horas	0,92
Alcances mes	0,005	4 Horas	0,92
Reportes	0,025	4 Horas	0,90

Fuente: Elaboración propia

Para calcular la fiabilidad del sistema completo, en el módulo de usuario y seguridad se vio que si una autentificación de ingreso falla no se puede acceder al resto de los módulos por lo tanto está conectado en serie según el rol que se les asigna y los demás módulos tienen función independiente. Es por eso que la fiabilidad del sistema estará dada por la siguiente fórmula:

$$\text{Fiabilidad} = R_s * R_p$$

Donde:

$$R_s = R_i$$

$$R_p = \frac{\sum_{i=1}^n (R_i * R_j)}{\sum_{i=1}^n R_i}$$



$i$   
 $=$   
 $2$        $i$   
 $i$   
 $i$   
 $=$   
 $2$   
 $P$   
 $i$



La fórmula de  $R_p$ , la variable  $P_i$  es la participación en el equipo de desarrollo del módulo y como la participación fue al 100% entonces  $P_i = 1$ , así se tiene el siguiente resultado:

$$R_p = \frac{\sum_{i=2}^5 R}{12} = \frac{0.92+0.93+0.95+0.95+0.90+0.92+0.92+0.92+0.93+0.93+0.99+0.99}{12} = \frac{11.25}{12} = 0.94$$

Se comprueba que la fiabilidad del sistema está dada por:

$$\text{Fiabilidad} = 0.95 * 0.94 = 0.893 * 100 = 89.3 \%$$

De lo cual se puede establecer que un 10.7 % de que el sistema presente algún fallo, que pueda ser debido a que exista fallas con la conexión del sistema a la base de datos, servidor o uso incorrecto por parte del usuario.

#### 4.1.4. Mantenibilidad

Es la cualidad que tiene el software para poder ser modificado, incluyendo correcciones o mejoras del software, en los cambios de entorno o especificaciones de requerimientos funcionales. Para calcular la calidad de mantenimiento del sistema se emplea el Índice de Madurez del Software (IMS), que indica la estabilidad de un producto de software. El índice de madurez se calcula con la siguiente fórmula:

$$IMS = \frac{M_t - (F_a + F_b + F_c)}{M_t}$$

Donde:

$M_t$ : Número de módulos en la versión actual.

$F_a$ : Número de módulos en la versión actual que se han cambiado.

$F_b$ : Número de módulos en la versión actual que se han añadido.



$F_c$ : Número de módulos en la versión anterior que se han borrado en la versión actual.

Toda la recopilación de la información requerida por la fórmula se obtuvo en la siguiente tabla:

**Tabla 41**

*Información requerida por el IMS*

INFORMACIÓN	VALOR
$M_t$	13
$F_a$	0
$F_b$	0
$F_c$	0

Fuente: Elaboración propia

Se realiza el cálculo del IMS con los valores obtenidos:

$$IMS = \frac{13 - (0 + 0 + 0)}{13} = \frac{13}{13} = 1$$

$$IMS = 1 * 100 \% = 100 \%$$

Con ese resultado se concluye que la madurez del “Sistema de Información Web para la gestión de entrega de productos y su respectivo pago para mejorar la promoción de líderes y asesoras” Caso: Zermat La Paz, la cual nos indica que tiene un índice de madurez de software del 100 %.

#### 4.1.5. Portabilidad

Es la capacidad del software para ser trasladado de un entorno de operación a otro. Para medir la portabilidad del sistema se empleará la siguiente fórmula que indica el grado de portabilidad del software.

$$GP = 1 - \frac{ET}{ER}$$





Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si  $GP > 0$ : La portabilidad es más rentable que el re-desarrollo.

Si  $GP = 1$ : La portabilidad es perfecta.

Si  $GP < 0$ : El re-desarrollo es más rentable que la portabilidad.

Para llevar el sistema web a otro entorno se necesita una memoria USB 1 GB mínimo de capacidad, para crear el sistema en el entorno residente se necesita contar con un equipo de sistema operativo que pueda ser (Windows, Linux o MacOS) en las cuales se requiere tener instalado servidor Xampp, Composer, editor de código, lenguaje de programación PHP, gestor de base de datos MySQL instalado y navegador web Google Chrome, Mozilla Firefox, Safari, Opera y Brave.

Con esta información requiere por la fórmula, se procede a calcular el grado de portabilidad.

$$GP = 1 - \frac{1}{13} = 1 - 0.07 = 0.93 * 100 = 93 \%$$

Se concluye que el sistema web tiene un grado de portabilidad de un 93%.

#### 4.1.6. Calidad General

Una vez calculado todos los atributos con sus respectivos porcentajes según el estándar de la ISO/IEC 25010, se realiza un resultado general de la calidad del software como se visualiza en la siguiente tabla:



Tabla 42

Información de Calidad Global

ATRIBUTOS	VALOR EN %
Adecuación funcional	88.15
Usabilidad	93
Fiabilidad	89.3
Mantenibilidad	100
Portabilidad	93
Calidad General	92.69

Fuente: Elaboración propia

La calidad general del “Sistema de Información Web para la gestión de entrega de productos y respectivo pago para mejorar la promoción de Líderes y Asesoras” Caso: Zermat La Paz es del 92.69%.

## 4.2. Seguridad

La seguridad lógica fue muy importante durante el desarrollo del sistema, se implementó la seguridad lógica en el sistema web, como ser la autenticación de usuario, asignación de roles y acciones realizadas por el usuario, las que pertenecen al módulo de usuario, para proteger la información que se traslada entre el servidor se emplea la encriptación, también se implementó validación de datos del Framework Laravel y finalmente la seguridad de la base de datos.

### 4.2.1. Autenticación

El acceso al sistema es controlado por la autenticación, en la cual un usuario debe introducir los datos correctos como ser: nombre de Usuario, Contraseña y la transferencia de datos encriptadas, estos datos son validados en el lado del cliente y servidor.

En la siguiente figura se puede observar un fragmento de código de validación:



Figura 42

Código de validación de autenticación

```
class LoginController extends Controller
{
    0 references | 0 overrides
    public function showLoginForm(){
        return view('auth.login');
    }

    0 references | 0 overrides
    public function login(Request $request){

        $this->validateLogin($request);

        if (Auth::attempt(['usuario' => $request->usuario, 'password' => $request->password, 'condicion'=>1])){ "usuario": Unknown word.
            return redirect('/home');
        }

        return back()->withErrors(['usuario' => trans('auth.failed')]) "usuario": Unknown word.
        ->withInput(request(['usuario'])); "usuario": Unknown word.

    }

    1 reference | 0 overrides
    protected function validateLogin(Request $request){
        $this->validate($request, [
            'usuario' => 'required|string', "usuario": Unknown word.
            'password' => 'required|string'
        ],
        [
            'password.required' => 'El campo contraseña es obligatorio' "contraseña": Unknown word.
        ]);
    }

    0 references | 0 overrides
    public function logout(Request $request){
        Auth::logout();
        $request->session()->invalidate();
        return redirect('/');
    }
}
```

Fuente: Elaboración propia

#### 4.2.2. Asignación de roles

La asignación de roles permite trabajar con un grupo de usuarios y que este permita accesos y visualización de elementos independientes, permitiendo que cada grupo visualiza su información asignada, siendo invisible el contenido del otro grupo. Los roles permiten un nivel de permisos con una serie de privilegios. Esta es una medida de seguridad que se implementa para proteger que usuarios no autorizados tengan acceso a ciertos módulos del sistema y se registra todas las acciones realizadas por cada usuario.



Figura 43

Roles de asignación y acciones

ADMINISTRACION DE ROLES DISPONIBLES

Tabla de Roles Registrados

Mostrar 10 registros

Buscar:

Nº	ROL	DESCRIPCION	ESTADO	ACCIONES
1	Administrador	Administrador	Activo	X
2	Almacenes	Encargado de almacenes	Activo	X
3	Asesora	Asesora	Activo	X
4	Lider	Lider de Grupo	Activo	X
5	Sin Asignar	El Usuario no tiene ningún rol asignado	Activo	X

Registros del 1 al 5 de un total de 5 registros

Anterior 1 Siguiente

Fuente: Elaboración propia

#### 4.2.3. Encriptación

La encriptación de datos es muy importante para proteger los paquetes de información que se transmiten entre servidores, en el presente sistema se utilizó algoritmos de encriptación que ofrece Laravel OpenSSL y cifrado AES -256-CBC y AES-128-CBC que los valores encriptados están firmados por un código de autenticación de mensajes que detecta si el mensaje encriptado fue alterado.



Figura 44

Encriptación Laravel

```
namespace App\Http\Controllers;

use App\Cliente;      "Cliente": Unknown word.
use DB;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use App\Http\Requests\ClienteRequest;      "Cliente": Unknown word.
use App\Http\Requests\ClienteUpdateRequest;      "Cliente": Unknown word.

0 references | 0 implementations
class ClienteController extends Controller      "Cliente": Unknown word.
{
    //
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    0 references | 0 overrides
    public function index()
    {
        $clientes = DB::table('clientes')      "clientes": Unknown word.
            ->orderBy('id', 'desc')
            ->paginate(100000000000);
        $cont = 0;
        return view('cliente.index', ["clientes" => $clientes, "cont" => $cont]);
    }
}
```

Fuente: (Laravel.com, 2023)

#### 4.2.4. Seguridad en la base de datos

El gestor base de datos MySQL, tiene sus propias medidas de seguridad como una básica autenticación. Uno de los ataques más frecuentes realizados a una base de datos son las inyecciones SQL.

Las inyecciones SQL son cadena de instrucciones SQL que un usuario puede introducir en cualquier campo de un formulario. El framework Laravel proporciona la protección contra ataques de inyección de CSRF, XSS y SQL todas las consultas se realizan pasando por los modelos y



siendo enviados a los controladores para validan los datos y envían al servidor.

### 4.3. Estimación de costos del software

#### 4.3.1. Cocomo II

Para el cálculo del costo total con COCOMO II se emplea el uso de la herramienta “COCOMO II – Constructive Cost Model” que pertenece al Centro de Sistemas e Ingeniería de Software de la Universidad de California del Sur.

En el presente proyecto se realiza la estimación de costo total por LDC (Líneas de Código), que se realiza con los datos obtenidos del software la cantidad de líneas de código que se empleó en el desarrollo. Además de la información de líneas de código el modelo requiere varias características como la flexibilidad del desarrollo, la arquitectura, el trabajo de equipo, tamaño de la base de datos, complejidad del producto, capacidad del personal, la limitación de tiempo, la restricción de almacenamiento, uso de herramientas de software entre otros.

Tabla 43

Líneas de código total

CODIGO	TOTAL, LDC
Código utilizado	2777
Códigos reutilizados	716
Código modificado	659

Fuente: Elaboración propia

Una vez calculado todas las Líneas de código que se emplearon en el presente proyecto se realiza la evaluación de estimación de costo, tomando en cuenta el salario mínimo que es de 2362.00 Bs la cual nos da 339.00 Dólares, en la siguiente figura se observa el llenado de datos:



Figura 45

Inserción de datos para COCOMO I

**COCOMO II - Modelo de Costos Constructivo**

Riesgo de Montecarlo  
 Apagado ▾  
 Calcular automáticamente  
 Apagado ▾

Método de dimensionamiento del tamaño del software:

SLOC	% Diseño modificado	% Código modificado	% de integración requerida	Evaluación y Asimilación (0% - 8%)	Comprensión del software (0% - 50%)	Desconocimiento (0-1)
Nuevo	<input type="text" value="2777"/>					
Reutilizado	<input type="text" value="716"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="80"/>	<input type="text" value="6"/>	
Modificado	<input type="text" value="659"/>	<input type="text" value="23"/>	<input type="text" value="8"/>	<input type="text" value="83"/>	<input type="text" value="7"/>	<input type="text" value="47"/>

**Controladores de báscula de software**

Precedente  Arquitectura / Resolución de Riesgos  Madurez del proceso

Flexibilidad de desarrollo  Cohesión del equipo

**Controladores de costos de software**

**Producto**

Confiabilidad del software requerido

Tamaño de la base de datos

Complejidad del producto

Desarrollado para la reutilización

La documentación coincide con las necesidades del ciclo de vida

**Personal**

Capacidad de analista

Capacidad del programador

Continuidad del personal

Experiencia de aplicación

Experiencia de plataforma

Experiencia en lenguaje y conjunto de herramientas

**Plataforma**

Limitación de tiempo

Restricción de almacenamiento

Volatilidad de la plataforma

**Proyecto**

Uso de herramientas de software

Desarrollo multisitio

Calendario de desarrollo requerido

**Mantenimiento**

**Tarifas de mano de obra de software**

Costo por persona-mes (dólares)

Fuente: (<http://softwarecost.org/tools/COCOMO/>)

Una vez llenada la información requerida la herramienta nos proporciona los resultados y se muestra en la siguiente figura:



Figura 46

Resultado de estación de costo

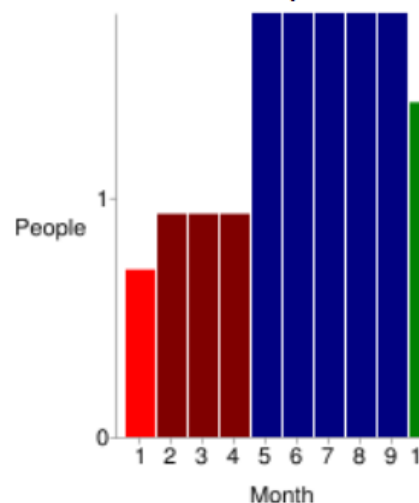
**Resultados**

**Esfuerzo de desarrollo de software (elaboración y construcción)**

= 11,7 meses-persona  
 Programación = 8,0 meses  
 Costo = \$3958

Tamaño total equivalente = 3504 SLOC  
 Factor de ajuste del esfuerzo (EAF) = 1,00

**Perfil de dotación de personal**



**Distribución de la fase de adquisición**

Fase	Esfuerzo (Personas-mes)	Horario (Meses)	Personal promedio	Costo (Dólares)
Comienzo	0,7	1.0	0,7	\$237
Elaboración	2.8	3.0	0,9	\$950
Construcción	8.9	5.0	1.8	\$3008
Transición	1.4	1.0	1.4	\$475

**Distribución del esfuerzo de software para RUP/MBASE (persona-meses)**

Fase/Actividad	Comienzo	Elaboración	Construcción	Transición
Gestión	0.1	0.3	0,9	0,2
Medio ambiente/CM	0.1	0,2	0,4	0.1
Requisitos	0.3	0,5	0,7	0.1
Diseño	0.1	1.0	1.4	0.1
Implementación	0.1	0,4	3.0	0.3
Evaluación	0.1	0.3	2.1	0.3
Despliegue	0.0	0.1	0.3	0,4

Fuente: (<https://softwarecost.org/tools/COCOMO/>)

Los resultados que nos proporciona la herramienta son datos importantes que se observan en la siguiente tabla:





Tabla 44

Resultados de COCOMO II

RESULTADOS	DETALLE
Esfuerzo de desarrollo de software (elaboración y construcción)	11,7Personas/Mes
Tiempo estimado de desarrollo	8,3 Meses
Cantidad de personas para desarrollar el proyecto	1 Persona
Costo total del proyecto final	3958 \$us

Fuente: Elaboración propia





CAPÍTULO V



Editar con WPS Office

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

El objetivo general del proyecto se ha logrado satisfactoriamente con el desarrollo del Sistema de Información Web para la Gestión de Productos y Pagos para mejorar la promoción de Líderes y Asesoras para Zermat La Paz utilizando las herramientas de diseño y desarrollo propuestos para la culminación del proyecto.

- Se elaboró una base de datos confiable y segura para el almacenamiento de toda la información que conlleva la administración de Zermat La Paz.
- Se logró realizar un análisis del proceso, lo que permitió diseñar los módulos del sistema de manera ordenada para cada proceso.
- Se realizó una ingeniería de requerimientos para la recolección de datos del sistema para tener datos precisos.
- Los diseños de interfaz gráfica del sistema fueron diseñados de forma amigable, donde los usuarios realizan los registros con facilidad mostrando una información rápida.
- El sistema permite generar reportes de acuerdo al requerimiento de los usuarios del sistema.
- La introducción del sistema ha resultado en una mejora significativa en la gestión de datos relacionados con los procesos de compra y venta.



## 5.2. Recomendaciones

A partir del presente proyecto se proponen las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Se recomienda capacitar al personal para que puedan realizar operaciones del sistema y así poder tener mejor control de las ventas y compras que ofrece el sistema.
- Se recomienda cambiar continuamente las contraseñas para la seguridad del sistema y proteger el acceso a personas ajenas.
- Cuando se requiera la ampliación y creación de nuevos módulos, se recomienda primero revisar la documentación para tomar una buena decisión, ya que el sistema presenta elementos reutilizables para el desarrollo de nuevos módulos



## BIBLIOGRAFIA

APACHE.ORG. (2 de 1995). *APACHE FOUNDATION*. Obtenido de APACHE FOUNDATION:

[https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)

CALLE LIMACHI, E. E. (2020). *REPOSITORIO INSTITUCIONAL UPEA*. Obtenido de REPOSITORIO INSTITUCIONAL

UPEA-CARRERA INGENIERÍA DE SISTEMAS: <http://repositorio.upea.bo/handle/123456789/222>

Castellanos, L. (2011). *Desarrollo de Sistemas de Información bajo un enfoque incremental*. maracaibo:

Universidad Nacional Experimental de la Fuerza Armada.

CHOQUE RODRIGUES, P. (2020). *REPOSITORIO INSTITUCIONAL UPEA*. Obtenido de

<http://repositorio.upea.bo/handle/123456789/162>

CodeIgniter. (2023). *CodeIgniter*. Obtenido de CodeIgniter: <https://codeigniter.com/>

FLORES, M., DOLZ, F., & CONTRERAS, J. (2016). *Biblioteca de Informática, UMSA*. Obtenido de Carrera de

Informática: [https://bibliotecas.umsa.bo/cgi-bin/koha/opac-search.pl?q=venta+de+productos+&idx=ti&branch\\_group\\_limit=branch%3AINF](https://bibliotecas.umsa.bo/cgi-bin/koha/opac-search.pl?q=venta+de+productos+&idx=ti&branch_group_limit=branch%3AINF)

Gómez, M., Cervantes, J., & Gonzáles, P. (2019). *Fundamentos de Ingeniería de Software*. México:

Universidad Autónoma Metropolitana .

HOSTINGER. (2023). *HOSTINGER TUTORIALES*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-css>

IBCE. (1 de 3 de 2017). *Instituto Boliviano de Comercio Exterior IBCE*. Obtenido de IBCE:

<https://ibce.org.bo/principales-noticias-bolivia/noticias-nacionales-detalle.php?id=74363&idPeriodico=8&fecha=2017-03-01>

ISO 25000. (2022). *ISO 25000 CALIDAD DEL SOFTWARE Y DATOS*. Obtenido de

<https://iso25000.com/index.php/normas-iso-25000>

JQUERY. (2023). *QUERY*. Obtenido de QUERY: <https://jquery.com/>

KINSTA. (2023). *KINSTA*. Obtenido de KINSTA: <https://kinsta.com/es/blog/herramientas-pruebas->

[rendimiento/](https://kinsta.com/es/blog/herramientas-pruebas-rendimiento/)

López, M. A. (15 de 1 de 2015). *MIALTO WEB*. Obtenido de MIALTO WEB: <https://mialtoweb.es/definicion-de>

[-aplicacion-web/](https://mialtoweb.es/definicion-de-aplicacion-web/)

MariaDB Foundation. (2023). *Oracle*. Obtenido de Oracle: <https://mariadb.org/documentation/>

MariaDB.org. (2023). *MariaDB*. Obtenido de MariaDB: <https://mariadb.org/es/>

MONTOYA ALARCÓN, M. (2020). *UNIVERSIDAD CATOLICA DE PEREIRA FACULTAD DE CIENCIAS*. Obtenido de





# ANEXOS



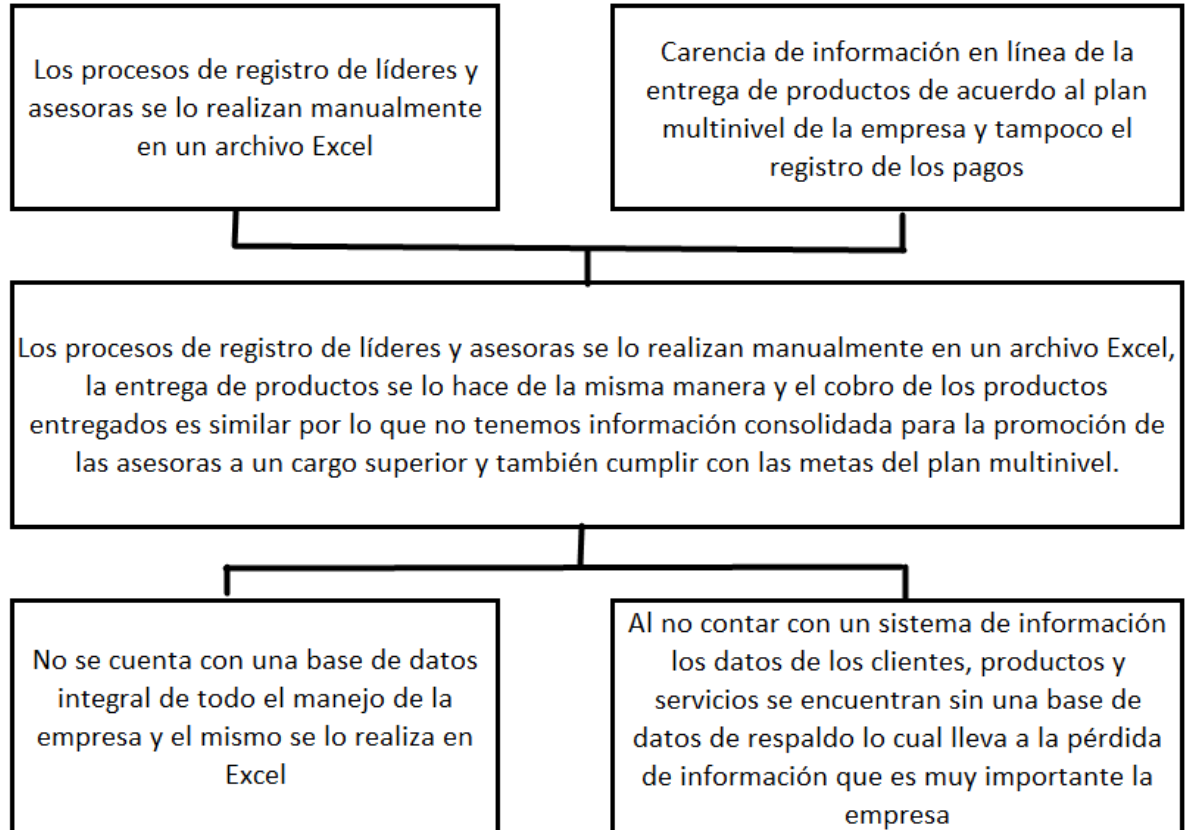
Editar con WPS Office





Editar con WPS Office

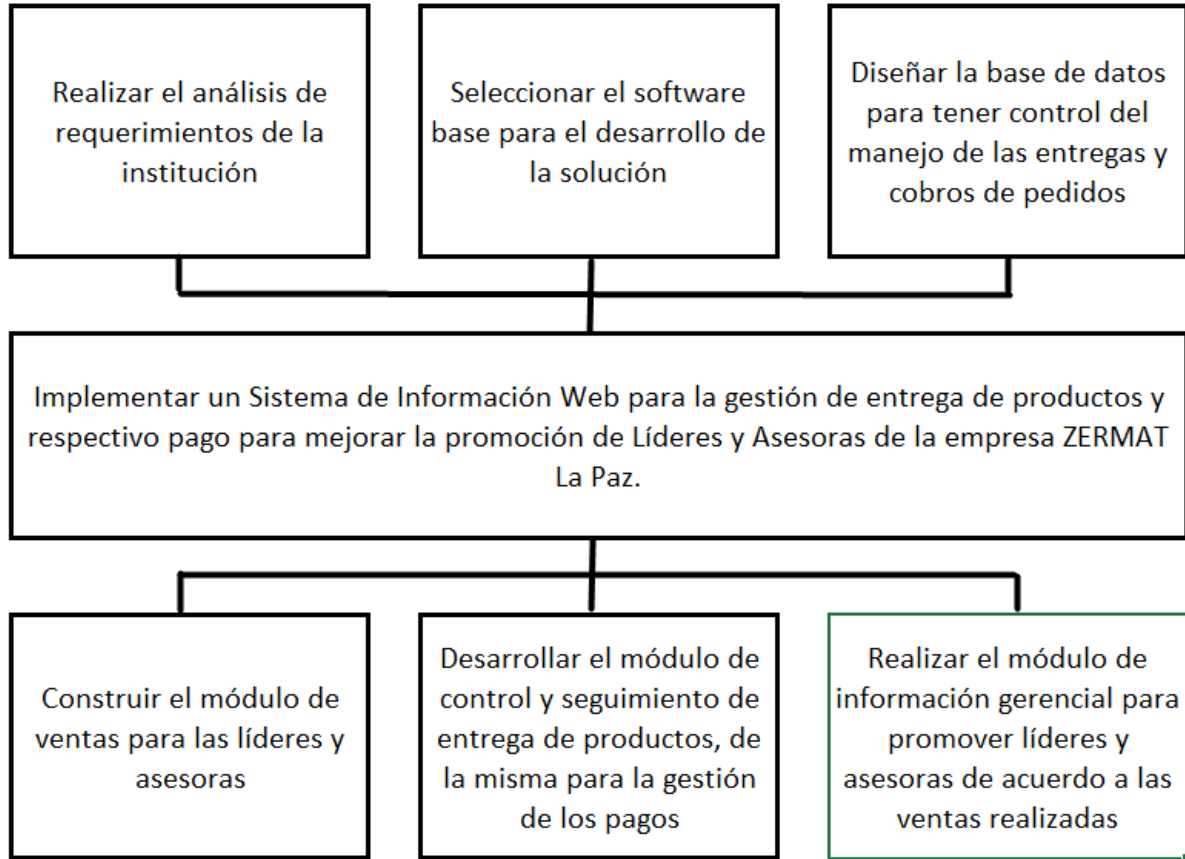
## ÁRBOL DE PROBLEMAS





Editar con WPS Office

## ÁRBOL DE OBJETIVOS



# MANUAL TÉCNICO

“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE ENTREGA DE PRODUCTOS Y RESPECTIVO PAGO PARA MEJORAR LA PROMOCIÓN DE LÍDERES Y ASESORAS”

CASO: ZERMAT LA PAZ

*Zermat*<sup>®</sup>  
-----La Paz-----



Editar con WPS Office

## 1. PRINCIPALES REQUERIMIENTOS

### 1.1 Requerimientos Funcionales

- *Gestión de productos: Compras y ventas.*
- *Seguimiento: Seg. Ventas, Seg. Comisión % Líder, Seg. Líder, Seg. Asesoras.*
- *Paramétricas: Clientes, Proveedores, Productos, Tipo características, Tipo línea, Tipo producto*
- *Acceso al Sistema mediante verificación de datos de acceso.*
- *Administrar los roles y permisos de acuerdo a los usuarios registrados.*
- *Administración de Usuarios.*

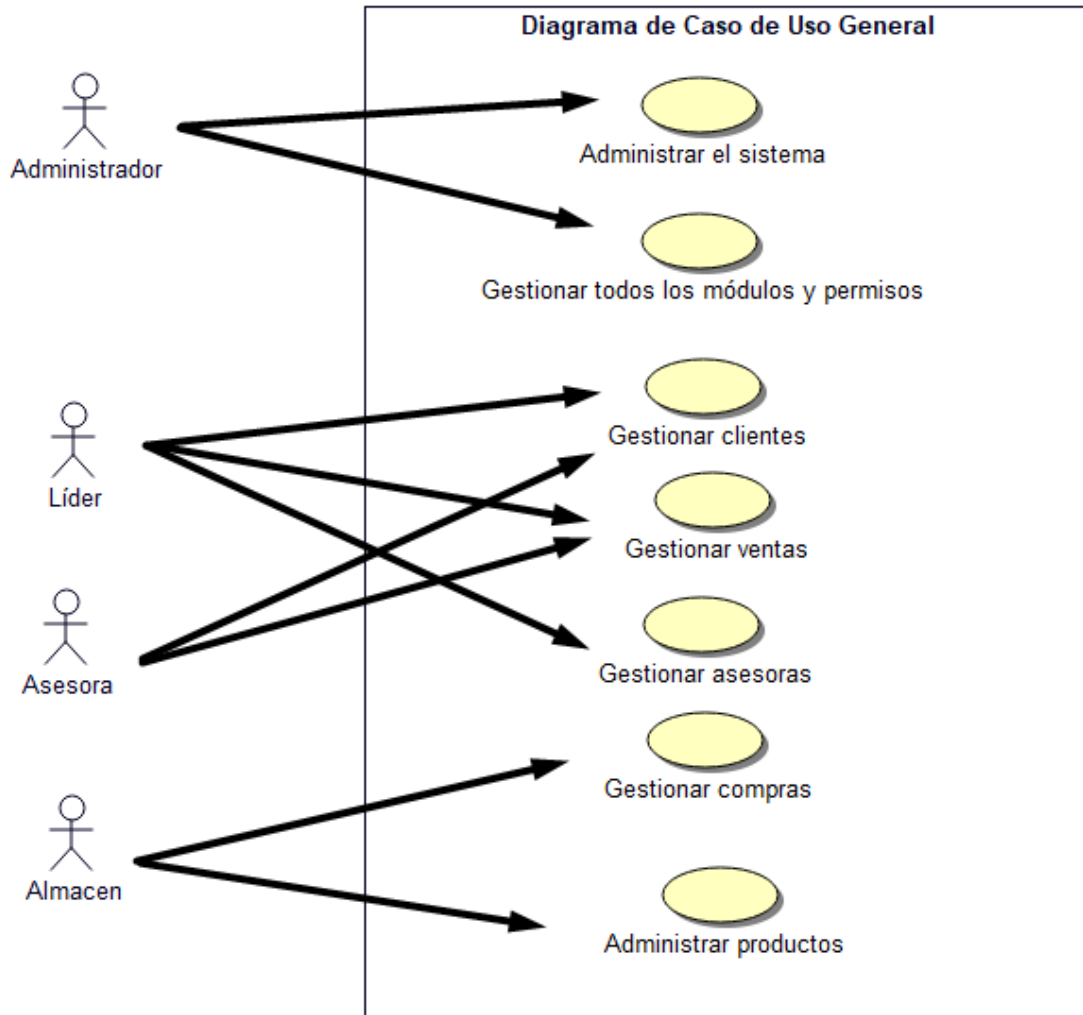
### 1.2 Requerimientos No Funcionales

- *Sistema Compatible con los diferentes navegadores.*
- *Interfaz gráfica debe responder al usuario cuando lo requiera*
- *Transacciones deben responder las necesidades de los usuarios.*
- *Soporte y Mantenimiento, para disponibilidad del sistema.*



## 2 DISEÑO DEL SISTEMA

### 2.1 Caso de Uso General



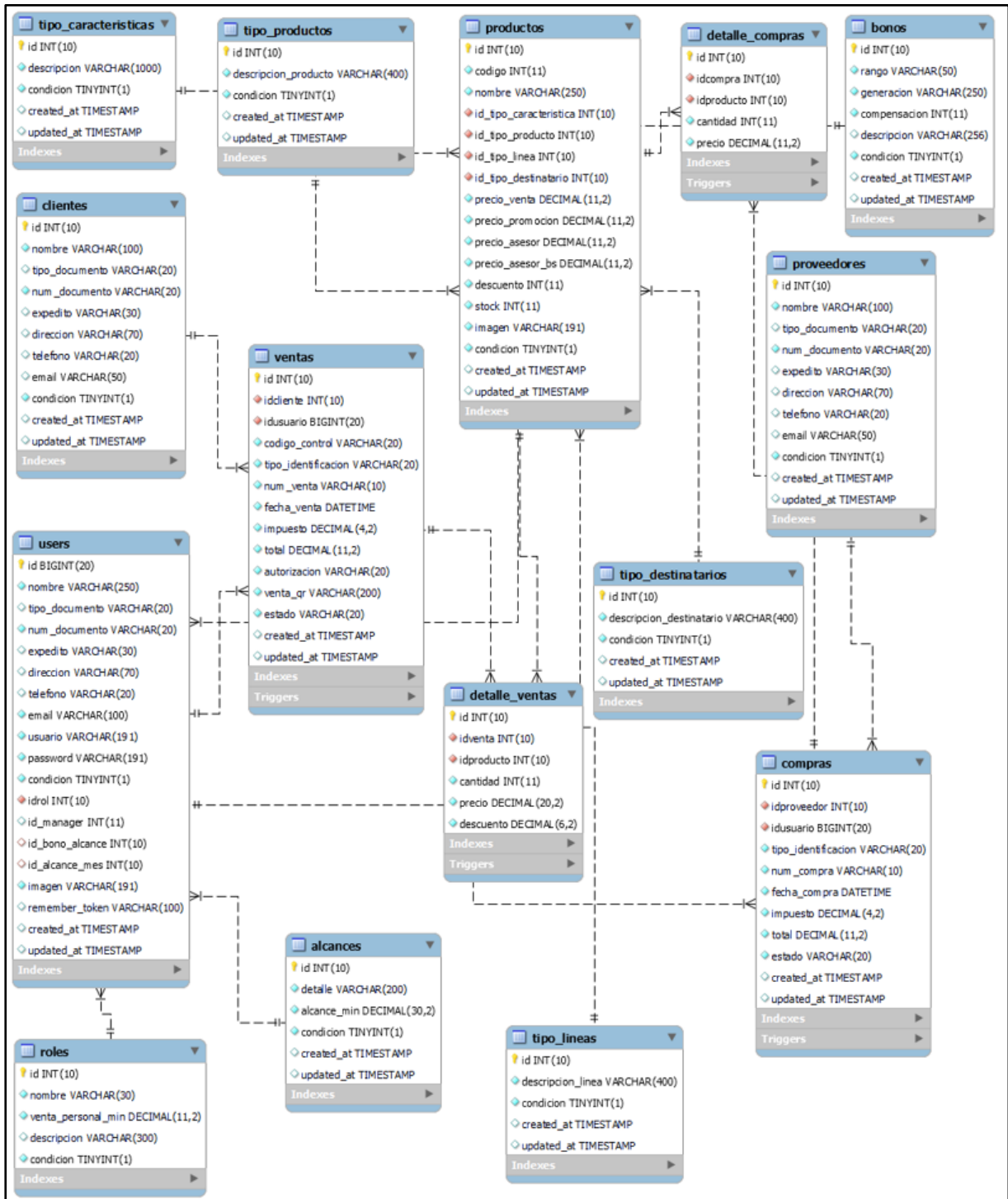


Editar con WPS Office



### 3 BASE DE DATOS

#### 3.1 Modelo de Base de Datos Relacional



#### 4 HERRAMIENTA DE DESARROLLO

Las herramientas para el desarrollo del sistema serán las siguientes:

- Hypertext Pre.Procesor “PHP”

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

- MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation.

- Laravel

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP, basado en el modelo vista controlador, intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP

#### 5 REQUERIMIENTO E INSTALACIÓN

##### 5.1 Requerimientos

A continuación, se lista los requerimientos del sistema para la implementación:

- PHP versión 7.
- Servidor Apache instalado.
- Acceso a Terminal para ejecución de comandos.
- Composer.

El sistema cuenta con un flujo considerable de archivos de imágenes y pdf por lo que se recomienda tener la capacidad de acuerdo a la siguiente especificación:

- 200 GB de almacenamiento como mínimo.



# MANUAL DEL USUARIO

“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE ENTREGA DE PRODUCTOS Y RESPECTIVO PAGO PARA MEJORAR LA PROMOCIÓN DE LÍDERES Y ASESORAS”

CASO: ZERMAT LA PAZ

*Zermat*<sup>®</sup>  
-----La Paz--



Editar con WPS Office

## Introducción

Es esencial revisar antes el Manual de Usuario la cual será una guía de paso a paso en el manejo del sistema, y todo esto para un mejor entendimiento.

El presente sistema se encarga de realizar un seguimiento de la gestión administrativa de Zermat La Paz.

## Objetivos del Manual

Guiar al usuario encargado sobre el manejo del “SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE ENTREGA DE PRODUCTOS Y RESPECTIVO PAGO PARA MEJORAR LA PROMOCIÓN DE LÍDERES Y ASESORAS” para así obtener resultados eficientes.

- Dar a entender a los usuarios encargados las características y las formas del funcionamiento del sistema brindado
- Proporcionar asistencia al usuario encargado del sistema

## Conocimientos Básicos del Usuario

- Conocimiento básico en la Navegación Web

## Requisitos previos

- Equipo de computación, con una capacidad de RAM 2GB o superior, Disco duro Mínimo de 50 GB, Procesador Core i3 o Superior
- Sistema Operativo Windows, Linux o MacOS.
- Estar conectado a internet.
- Tener instalado un navegador ya sea Firefox, Chrome, Ópera entre otros navegadores.
- Tabletas, celular con conexión a internet y un navegador instalado.



## Interfaces de navegación para el Usuario

Abrir el Navegador y anotar la siguiente URL <https://zermatlapaz.com>

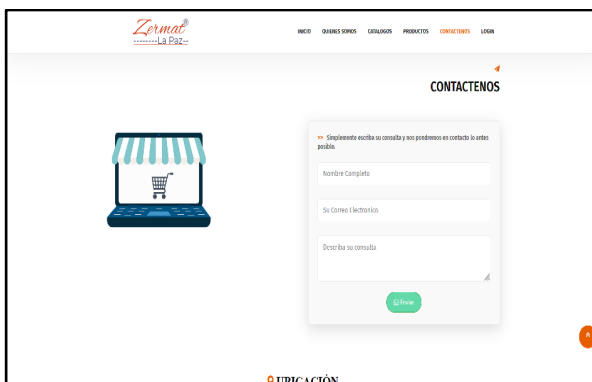
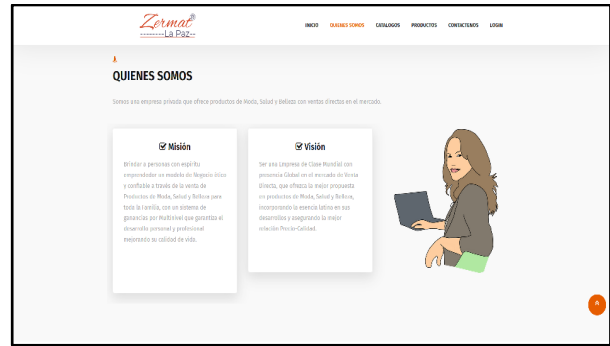
Ingresado a la URL, aparecerá la página principal.



Barra de menú principal disponibles:

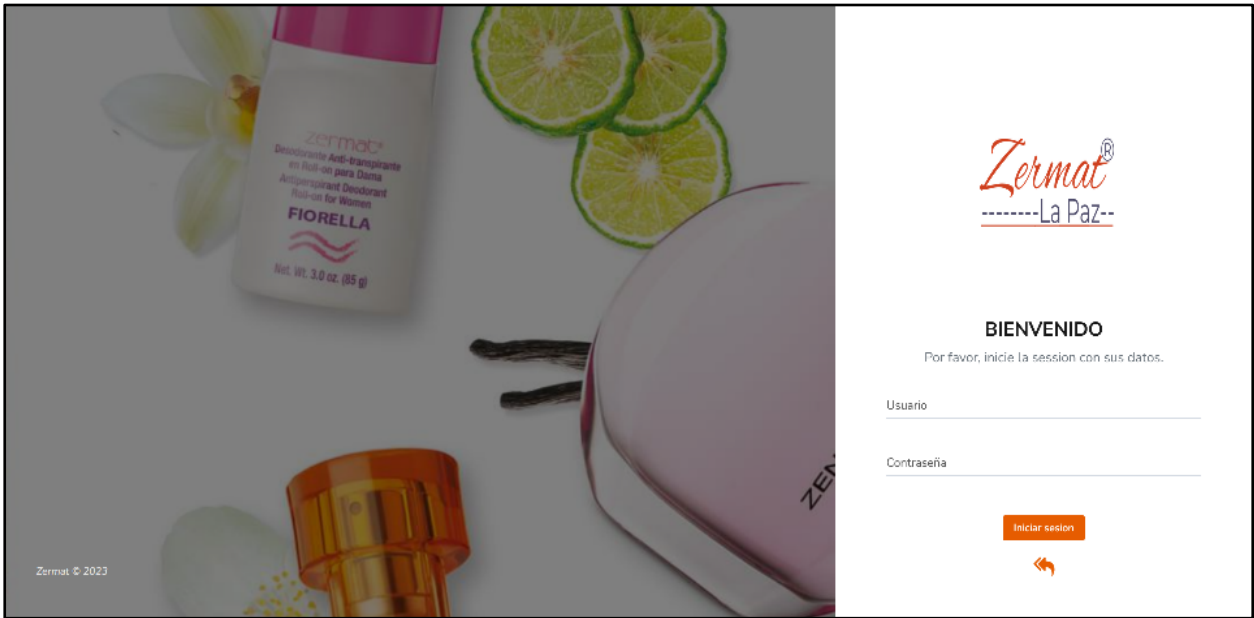
- Seleccionar la pestaña INICIO la cual nos devuelve a la página principal.
- Seleccionar la pestaña QUIENES SOMOS, lleva a misión y visión.
- Seleccionar la pestaña CATALOGOS, la cual describe el catálogo que se ofrece.
- Seleccionar la pestaña PRODUCTOS, la lleva muestra los productos más la existencia de los mismos para su venta.
- Selecciona la pestaña CONTACTENOS, la cual el cliente envía su consulta al correo corporativo.



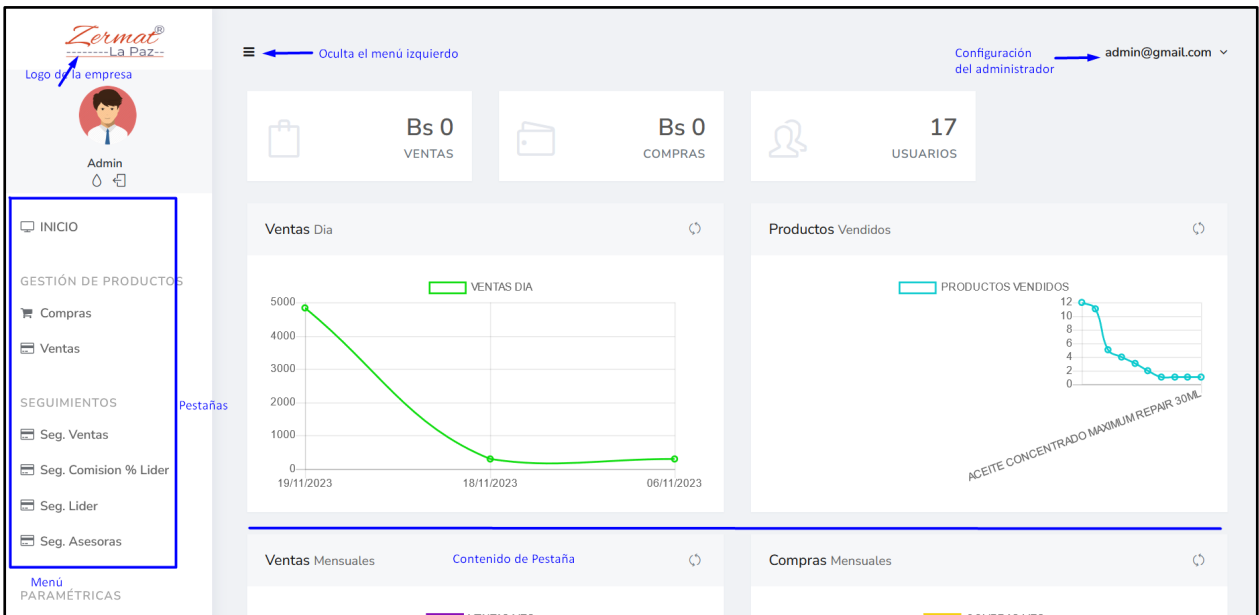


Selección de la pestaña **LOGIN**: la cual lleva al formulario de inicio de sección del usuario que deberá introducir el nombre de Usuario y Contraseña, la cual es asignado por el Administrador, luego presionar el botón “Iniciar sesión” .



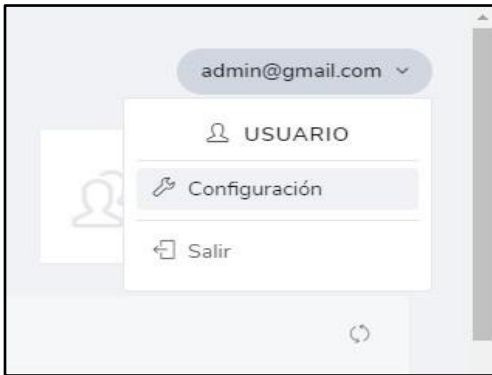


A continuación, ingresaré como Administrador a la página principal, donde tenemos:

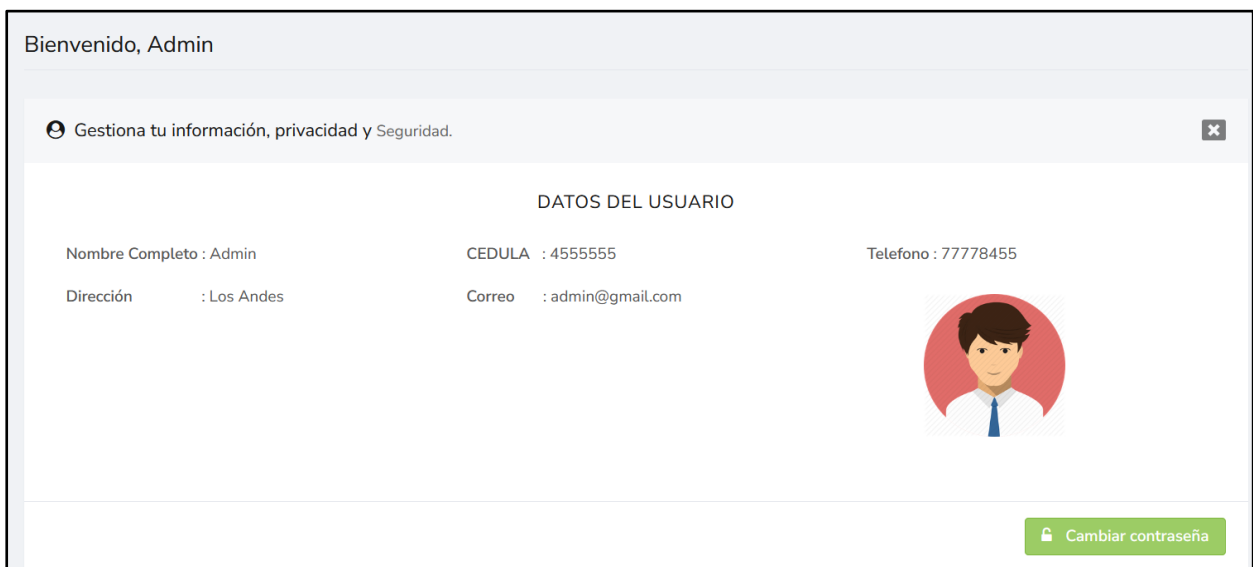


Seleccionar el botón Configuración del Administrador, donde aparece la opción de Configuración y el de Salir del modo Administrador.





A continuación, ingresa a Configuración: donde aparece la opción de cambiar la información del Administrador, en el botón Cambiar contraseña, además de editar la foto dando clic sobre la imagen.



Para cambiar contraseña primero debe ingresar la contraseña actual con el que inicio la sesión y luego establecer la nueva contraseña y confirmarlo según indica el formulario.





Bienvenido, admin

Actualizar contraseña. ✕

Nota: Al actualizar su contraseña con éxito será automáticamente redireccionado al login y deberá iniciar sesión con su nueva contraseña.

Contraseña actual:

Nueva contraseña:

Confirma contraseña:

Cambiar contraseña

Seleccionar la pestaña INICIO: donde muestra el resumen de las Ventas del Día, Productos Vendidos, Ventas Mensuales y Compras Mensuales.

En el menú GESTIÓN DE PRODUCTOS seleccionar Compras: Son todas las compras realizadas.

ADMINISTRACION DE COMPRAS

Tabla de Compras Registrados [+ Nueva Compra](#)

Mostrar  registros Buscar:

N°	FECHA	PROVEEDOR	N°COMPRA	IVA	TOTAL	COMPRADOR	ESTADO
1	21 de noviembre 2023	Zermat México	8	0.13	Bs. 648.85	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
2	21 de noviembre 2023	Zermat México	7	0.13	Bs. 500.19	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
3	19 de noviembre 2023	Zermat México	6	0.13	Bs. 1,090.08	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
4	19 de noviembre 2023	Zermat México	5	0.13	Bs. 11,997.82	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
5	19 de noviembre 2023	Zermat México	4	0.13	Bs. 1,663.47	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
6	19 de noviembre 2023	Zermat México	3	0.13	Bs. 4,258.86	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
7	06 de noviembre 2023	Zermat México	2	0.13	Bs. 5,323.11	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>
8	06 de noviembre 2023	Zermat México	1	0.13	Bs. 9,363.87	ALAN MONTALVO QUISPE	<span style="background-color: #c6e0b4;">Registrado</span>

Registros del 1 al 8 de un total de 8 registros Anterior **1** Siguiente

### ADMINISTRAR NUEVAS COMPRAS

**Tabla de Registro de Compras** ✕

Nota: Todos los campos con (\*) son obligatorios

Número de Compra (*)	Documento (*)	
9 <span>Tipo documento</span>	SELECCIONE	
Nombres Paterno Materno: Proveedor (*)	Producto (*)	
SELECCIONE <span>Datos del proveedor</span>	SELECCIONE <span>Selección de producto</span>	
Destinados a:	Línea	Tipo de Producto
Destinatarios	Tipo de línea	Tipo de producto
Características		
Breve descripción de producto		
Cantidad (*)	Precio de Compra por Unidad (*)	
Ingrese cantidad <span>Cantidad a dar de alta</span>	Ingrese precio de compra <span>Precio del producto</span>	

+ Agregar Detalle Compra

En el menú **GESTIÓN DE PRODUCTOS** seleccionar **Ventas**: Son todas las ventas realizadas.



### ADMINISTRACION DE VENTAS

Tabla de Ventas Registrados Facturas Anulados + Nueva Venta

Mostrar 10 registros Buscar:

Nº	FECHA	CLIENTE	NºVENTA	IVA	TOTAL	VENDEDOR
1	19 de noviembre 2023	ANDREA GARCIA BENITEZ	12	0.16	Bs. 3.086.23	RUTH MARIBEL ALMEIDA CORTEZ
2	19 de noviembre 2023	SUSANA BALDEON LOZA	11	0.16	Bs. 260.12	MARIO FRANCISCO JATIVA REYES
3	19 de noviembre 2023	ANDREA GARCIA BENITEZ	10	0.16	Bs. 357.01	TANIA PATRICIA ANDRADE ULLAURI
4	19 de noviembre 2023	ALBA CUMANDA MINA D ELA CRUZ	9	0.16	Bs. 159.86	TANIA PATRICIA ANDRADE ULLAURI
5	19 de noviembre 2023	WHIMPER TORRES FLORES	8	0.16	Bs. 274.65	TANIA PATRICIA ANDRADE ULLAURI
6	19 de noviembre 2023	SANTIAGO EDISON MAURICIO	7	0.16	Bs. 274.65	TANIA PATRICIA ANDRADE ULLAURI
7	19 de noviembre 2023	WHIMPER TORRES FLORES	6	0.16	Bs. 102.69	RUTH MARIBEL ALMEIDA CORTEZ
8	19 de noviembre 2023	ALBA CUMANDA MINA D ELA CRUZ	5	0.16	Bs. 308.08	RUTH MARIBEL ALMEIDA CORTEZ
9	18 de noviembre 2023	ALBA CUMANDA MINA D ELA CRUZ	4	0.16	Bs. 203.61	AZUAY MERCEDES LÓPEZ GONZÁLEZ



# ADMINISTRAR NUEVAS VENTAS

## Tabla de Registro de Ventas

Nota: Todos los campos con (\*) son obligatorios

Número de Venta: 13

Documento (\*): FACTURA [Seleccionar opción factura](#)

Nombres Paterno Materno: Cliente (\*): SUSANA BALDEON LOZA [← Datos del cliente](#)

Producto (\*): 93027 MAQUILLAJE LÍQUIDO HUMECTANTE 30 ML BRONZE [← Producto](#)

Destinados a: Destinatarios

Línea: Tipo de línea

Tipo de Producto: Tipo de producto

Características: Breve descripción de producto

Cantidad (\*): Ingrese cantidad [← Cantidad](#)


Stock: Stock Actual

Precio Asesor Bs: Precio de venta asesor

Descuento en %: 0 [← % descuento](#)

[+ Agregar Detalle Venta](#)

### DETALLE DE VENTA DE LOS DIVERSOS PRODUCTOS

ELIMINAR	PRODUCTO	PRECIO ASESOR (BS.)	DESCUENTO	CANTIDAD	SUBTOTAL (BS.)
	93027 MAQUILLAJE LÍQUIDO HUMECTANTE 30 ML BRONZE	98,14	0,00	1	Bs. 98,14



Men ú SEGUIMIENTOS, opci ó n Seg. Ventas con el rol de LIDER

**CONSULTE SUS VENTAS**

Fecha Inicio >> << Fecha Fin Imprimir Buscar

Fecha inicio consulta      Fecha final consulta

Mostrar 10 registros Buscar:

Nº	AÑO	MES	NOMBRE	VENTA	PAGAR	FECHA
1	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	193.76	129.1776	2023-11-06 15:40:40
2	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	88.53	59.0208	2023-11-06 15:57:04
3	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	88.53	59.0208	2023-11-18 20:25:24
4	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	175.53	118.0416	2023-11-18 20:26:21

Registros del 1 al 4 de un total de 4 registros Anterior 1 Siguiente

Men ú SEGUIMIENTOS, opci ó n Seg. L í der con el rol de LIDER

**Seguimiento Lider**

**Tabla de Seguimiento Lider**

Mostrar 10 registros Buscar:

Nº	AÑO	MES	LIDER	ASESORA	VENTAS	PAGO A RECIBIR
1	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	AZUAY MERCEDES LÓPEZ GONZÁLEZ	546.35	365.2608

Registros del 1 al 1 de un total de 1 registros Anterior 1 Siguiente



Menú SEGUIMIENTOS, opción Seg. Asesoras con el rol de LIDER

ADMINISTRACION DE VENTAS

Tabla de Seguimiento Asesora

Mostrar 10 registros Buscar:

N°	AÑO	MES	NOMBRE	VENTAS	PAGO A RECIBIR
1	2023	11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	546.35	365.2608
2	2023	11	MARIO FRANCISCO JATIVA REYES	224.24	146.1600
3	2023	11	RUTH MARIBEL ALMEIDA CORTEZ	354.12	236.0832
4	2023	11	RUTH MARIBEL ALMEIDA CORTEZ	574.20	382.8000
5	2023	11	RUTH MARIBEL ALMEIDA CORTEZ	708.24	472.1664
6	2023	11	RUTH MARIBEL ALMEIDA CORTEZ	1378.10	918.7200
7	2023	11	TANIA PATRICIA ANDRADE ULLAURI	919.12	606.0768

Registros del 1 al 7 de un total de 7 registros Anterior **1** Siguiente

En el menú PARAMÉTRICAS tenemos los botones de ACCIONES y aplica a todas las opciones del menú.

TELEFONO	ESTADO	ACCIONES
75812075	Activo	  
223377776	Activo	  
22334234	Activo	  



Menú PARAMÉTRICAS, opción Clientes.

ADMINISTRACION DE CLIENTES

Tabla de Clientes Registrados + Crear Nuevo Cliente

Mostrar  registros Buscar:

Nº	CLIENTE	DOCUMENTO	NºDOCUMENTO	EXPEDIDO	TELEFONO	ESTADO	ACCIONES
1	SUSANA BALDEON LOZA	NIT	423232	CHUQUISACA	22333222	Activo	
2	ANDREA GARCIA BENITEZ	NIT	7654334	CHUQUISACA	23998022	Activo	
3	ALBA CUMANDA MINA D ELA CRUZ	NIT	63423232	LA PAZ	222347778	Activo	
4	WHIMPER TORRES FLORES	NIT	4223232	LA PAZ	22234556	Activo	
5	SANTIAGO EDISON MAURICIO	CI	543323	LA PAZ	223333322	Activo	

Registros del 1 al 5 de un total de 5 registros Anterior **1** Siguiente

Actualizar Datos del Cliente

Nombre:

Dirección:

Documento:

Número documento:

Expedido:

Telefono:

Correo:

Cerrar Guardar

Registros del 1 al 5 de un total de 5 registros Anterior **1** Siguiente



admin@gmail.com

### Cambiar el Estado del Cliente

Estas seguro de cambiar el estado?

Cerrar Aceptar

ADMINISTRACION DE CLIENTES

Tabla de Clientes Registrados + Crear Nuevo Cliente

Mostrar 10 registros Buscar:

N°	CLIENTE	DOCUMENTO	N°DOCUMENTO	EXPEDIDO	TELEFONO	ESTADO	ACCIONES
1	SUSANA BALDEON LOZA	NIT	423232	CHUQUISACA	22333222	Activo	
2	ANDREA GARCIA BENITEZ	NIT	7654334	CHUQUISACA	23998022	Activo	
3	ALBA CUMANDA MINA D ELA CRUZ	NIT	63423232	LA PAZ	222347778	Activo	
4	WHIMPER TORRES FLORES	NIT	4223232	LA PAZ	22234556	Activo	
5	SANTIAGO EDISON MAURICIO	CI	543323	LA PAZ	223333322	Activo	

Registros del 1 al 5 de un total de 5 registros Anterior 1 Siguiente

admin@gmail.com

### Llenar los Datos del Cliente

Nombre

Dirección

Documento

Número documento

Expedido

Telefono

Correo

Cerrar Guardar

+ Crear Nuevo Cliente

Buscar:

ESTADO	ACCIONES
Activo	
Activo	
Activo	
Activo	
Activo	

5 SANTIAGO EDISON MAURICIO CI 543323 LA PAZ 223333322

Registros del 1 al 5 de un total de 5 registros Anterior 1 Siguiente























Menú PARAMÉTRICAS, opción Proveedores.

ADMINISTRACION DE PROVEEDORES

Tabla de Proveedores Registrados + Crear Nuevo Proveedor

Mostrar 10 registros Buscar:

N°	CLIENTE	DOCUMENTO	N°DOCUMENTO	EXPEDIDO	TELEFONO	ESTADO	ACCIONES
1	Zermat México	NIT	1299812012	LA PAZ	75812075	Activo	  
2	BELEN LOOR SEGOVIA	NIT	87434322	LA PAZ	223377776	Activo	  
3	PABLO SAÚL SOLÓRZANO SALINAS	NIT	12323133	LA PAZ	22334234	Activo	  
4	JOSE FRIERE ALMEIDA	NIT	65322232	BENI	77775554	Activo	  
5	ANA MERO DELGADO	NIT	35232342	COCHABAMBA	22333444	Activo	  
6	JAIME DANIEL SANTOS BALON	NIT	2234232332	COCHABAMBA	22333333	Activo	  













Registros del 1 al 6 de un total de 6 registros Anterior **1** Siguiente

Llenar los Datos del Proveedor admin@gmail.com

+ Crear Nuevo Proveedor

Buscar:

ESTADO ACCIONES

Activo	  
Activo	  
Activo	  
Activo	  

Nombre  ←

Dirección  ←

Documento  ←

Número documento  ←

Expedido  ←

Telefono  ←

Correo  ←

Cerrar Guardar



Menú PARAMÉTRICAS, opción Productos.

### ADMINISTRACION DE PRODUCTOS

Tabla de Productos Registrados Reporte PDF + Crear Nuevo Producto

Mostrar 10 registros

Nº	CODIGO	PRODUCTO
1	35053	SUERO HUMECTANTE FACIAL 30 G
2	37252	CREMA DE NOCHE ENERGIZANTE 50 G
3	37251	CREMA DE DÍA ENERGIZANTE 50 G
4	37250	CREMA CONTORNO DE OJOS ENERGIZANTE 15G
5	92118	ACEITE CONCENTRADO MAXIMUM REPAIR 30ML
6	35051	SUERO CONTORNO DE OJOS 25 ML
7	39476	MAXIMUM REPAIR CREMA DIA 50G
8	39739	MAXIMUM REPAIR CREMA NOCHE 30G
9	98128	MAQUILLAJE EN CMA PORCELAIN CLEAR AGELESS RFLX
10	93019	MAQUILLAJE LÍQUIDO HUMECTANTE 30 ML IVORY

Registros del 1 al 10 de un total de 20 registros Anterior 1 2 Siguiente

### Crear Nuevo Producto

No se ha seleccionado ningún archivo.















Menú PARAMÉTRICAS, opción Tipo Características.

### ADMINISTRACION DE TIPOS DE CARACTERISTICAS

Tabla de Característica Registrados + Agregar Nuevo tipo de Característica

Mostrar  registros Buscar:



Nº	DESCRIPCIÓN	ESTADO	ACCIONES
1	Ideal para piel mixta a grasa, formula que apoya en el equilibrio del brillo durante el día. Efecto mate incluso en zona "T" del rostro.	Activo	 
2	Sweet Like Candy by Ariana Grande	Activo	 
3	Creación de Zermart	Activo	 
4	Labial en barra con acabado brillante. Textura sedosa. Con extracto de Coco y Karité que cuidan tus labios.	Activo	 
5	Rehidrata la piel a profundidad. Suaviza y previene líneas de expresión.	Activo	 
6	Sensación de extra humectación. Deja tersa y luminosa la piel. Con propiedades antioxidantes. Previene líneas de expresión. Uso de día y noche.	Activo	 

### Crear Nueva Característica

Descripción

Ingrese descripción







Solo se acepta 500 caracteres

admin@gmail.com

+ Agregar Nuevo tipo de Característica

Buscar:

Nº	ESTADO	ACCIONES
1	Activo	 
2	Activo	 
3	Activo	 















Menú PARAMÉTRICAS, opción Tipo Destinatarios.

ADMINISTRACION:TIPO DESTINATARIOS

Tabla de Destinatario Registrados + Agregar Nuevo tipo de Destinatario

Mostrar 10 registros Buscar:

N°	DESCRIPCIÓN DESTINATARIO	ESTADO	ACCIONES
1	Unisex	Activo	 
2	Niña	Activo	 
3	Familia	Activo	 
4	Hombre	Activo	 
5	Mujer	Activo	 
6	Niño	Activo	 

Registros del 1 al 6 de un total de 6 registros Anterior **1** Siguiente



Crear Nueva Destinatario admin@gmail.com

Descripción de destinatario

Cerrar Guardar

+ Agregar Nuevo tipo de Destinatario

Mostrar 10 registros Buscar:

N°	DESCRIPCIÓN DESTINATARIO	ESTADO	ACCIONES
1	Unisex	Activo	 























Menú PARAMÉTRICAS, opción Tipos de Líneas.

ADMINISTRACION DE TIPOS DE LINEAS EXISTENTES

Tabla de Línea Registrados + Agregar Nuevo tipo de Línea

Mostrar  registros Buscar:

N°	DESCRIPCIÓN DE LINEA	ESTADO	ACCIONES
1	Zermat Skin	Activo	 
2	Zermat Care	Activo	 
3	Zentimiento	Activo	 
4	Wake Up	Activo	 
5	Virazel	Activo	 
6	Status	Activo	 
7	Senzuel	Activo	 
8	Ronal Jelly	Activo	 
9	Reflex	Activo	 
10	Premium Niños	Activo	 

Registros del 1 al 10 de un total de 23 registros Anterior **1** 2 3 Siguiente

ADM

Crear Nueva Línea ✕





Descripción de línea

✕ Cerrar Guardar

+ Agregar Nuevo tipo de Línea

Buscar:

ACCIONES

1	Zermat Skin	Activo	 
2	Zermat Care	Activo	 













Menú PARAMÉTRICAS, opción Tipo Producto.

ADMINISTRACION DE TIPOS DE PRODUCTOS

Tabla de tipo de Producto Registrados + Agregar Nuevo tipo Producto

Mostrar  registros Buscar:



N°	DESCRIPCIÓN TIPO DE PRODUCTO	ESTADO	ACCIONES
1	Refrescante	Activo	 
2	Perfumería	Activo	 
3	Para Piernas	Activo	 
4	Maquillaje	Activo	 
5	Desodorante Perfumado	Activo	 
6	Desodorante en Barra	Activo	
7	Desodorante	Activo	
8	Cuidado Personal	Activo	
9	Cuidado Facial	Activo	
10	Cuidado del Cabello	Activo	

Registros del 1 al 10 de un total de 13 registros Anterior **1** 2 Siguiente

ADM Crear Nuevo Tipo de Producto ✕

Tab Descripción de tipo de producto  + Agregar Nuevo tipo Producto

Mos Cerrar Guardar

1 Refrescante Activo  








Menú ADMINISTRACIÓN, opción Usuarios.

ADMINISTRACION DE USUARIOS

Tabla de Usuarios Registrados [+ Crear Nuevo Usuario](#)

Mostrar  registros  Buscar:

N°	IMAGEN	NOMBRE	CORREO	USUARIO	ROL	ASIGNADO
1		ALAN MONTALVO QUISPE	alan@gmail.com	alan	Almacenes	AZUAY MERCEDES LÓPEZ GONZÁLEZ
2		TANIA PATRICIA ANDRADE ULLAURI	tannia@gmail.com	tannia	Asesora	AZUAY MERCEDES LÓPEZ GONZÁLEZ
3		JUAN JOSE MONTUFAR MARCAYATA	jose@gmail.com	juanj	Asesora	MARIO FRANCISCO JATIVA REYES
4		LUIS HERMINIO MULLO CEPEDA	luis@gmail.com	luis	Asesora	MARIO FRANCISCO JATIVA REYES
5		RUTH MARIBEL ALMEIDA CORTEZ	ruth@gmail.com	ruth	Asesora	AZUAY MERCEDES LÓPEZ GONZÁLEZ



admin@gmail.com

### Crear Nuevo Usuario

Nombre  ←

Dirección  ←

Documento  ←

Número documento  ←

Expedido  ←

Telefono  ←

Correo  ←

Rol  ←

Asignar a:  ←

Rango  ←

Alcance Mes  ←

Usuario  ←

Contraseña  ←

Imagen  No se ha seleccionado ningún archivo.

+ Crear Nuevo Usuario

Buscar:

SIGNADO

QUAY MERCEDES LÓPEZ GONZÁLEZ

QUAY MERCEDES LÓPEZ GONZÁLEZ

MARIO FRANCISCO JATIVA REYES

MARIO FRANCISCO JATIVA REYES

QUAY MERCEDES LÓPEZ GONZÁLEZ

QUAY MERCEDES LÓPEZ GONZÁLEZ

Menú ADMINISTRACIÓN, opción Roles.





## ADMINISTRACION DE ROLES DISPONIBLES

Tabla de Roles Registrados

Mostrar 10 registros

Buscar:

Nº	ROL	DESCRIPCION	ESTADO	ACCIONES
1	Administrador	Administrador	Activo	
2	Almacenes	Encargado de almacenes	Activo	
3	Asesora	Asesora	Activo	
4	Lider	Lider de Grupo	Activo	
5	Sin Asignar	El Usuario no tiene ningún rol asignado	Activo	

Registros del 1 al 5 de un total de 5 registros

Anterior **1** Siguiente

Menú ADMINISTRACIÓN, opción Rangos.

## ADMINISTRACION DE RANGOS

Tabla de Rangos Registrados

[+ Crear Nuevo Rango](#)

Mostrar 10 registros

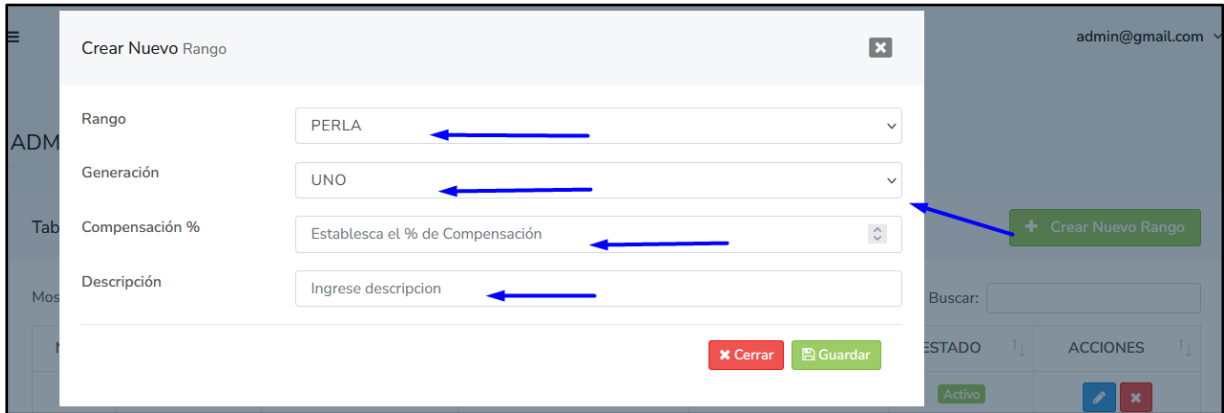
Buscar:

Nº	RANGO	GENERACIÓN	COMPENSACIÓN	DESCRIPCIÓN	ESTADO	ACCIONES
1	ZAFIRO	TRES	2 %	Ninguna	Activo	
2	ZAFIRO	DOS	5 %	Ninguna	Activo	
3	TURQUESA	DOS	5 %	Ninguna	Activo	
4	PERLA	DOS	4 %	Ninguna	Activo	
5	ZAFIRO	UNO	12 %	Ninguna	Activo	
6	TURQUESA	UNO	10 %	Ninguno	Activo	
7	PERLA	UNO	8 %	Ninguna	Activo	

Registros del 1 al 7 de un total de 7 registros

Anterior **1** Siguiente





Menú ADMINISTRACIÓN, opción Alcances Mes.

**ADMINISTRACION DE ALCANCES**

Tabla de Alcances Registrados + Crear Nuevo Alcance

Mostrar  registros Buscar:

Nº	DETALLE	MONTO MINIMO	ESTADO	ACCIONES
1	Hasta la Quinta Generación	10000.00	Activo	
2	Hasta la Quinta Generación	50000.00	Activo	
3	Hasta la Cuarta Generación	25000.00	Activo	
4	Hasta la Cuarta Generación	15000.00	Activo	
5	Hasta la Cuarta Generación	6000.00	Activo	
6	Hasta la Tercera Generación	3000.00	Activo	
7	Hasta la Segunda Generación	1250.00	Activo	
8	Grupo Personal	600.00	Activo	

Registros del 1 al 8 de un total de 8 registros Anterior **1** Siguiente



admin@gmail.com

Crear Nuevo Alcance

Detalle

Monto minimo

Nº	DETALLE	MONTO MINIMO	ESTADO	ACCIONES
1	Hasta la Quinta Generación	10000.00	Activo	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Menú ADMINISTRACIÓN, opción Reportes.

CONSULTA POR FECHA

Fecha Inicio << >> Fecha Fin

Mostrar 10 registros

Buscar: lider

Nº	NONBRE VENDEDOR	CARGO	CODIGO CONTROL	TOTAL VENTA	FECHA	ACCIONES
2	MARIO FRANCISCO JATIVA REYES	Lider	05-0F-EF-47-16	260.12	2023-11-19 17:48:37	<input type="button" value="Imprimir"/>
9	AZUAY MERCEDES LÓPEZ GONZÁLEZ	Lider	E0-4F-11-8B-FC	203.61	2023-11-18 20:26:21	<input type="button" value="Imprimir"/>
10	AZUAY MERCEDES LÓPEZ GONZÁLEZ	Lider	67-E6-93-C2-65	102.69	2023-11-18 20:25:24	<input type="button" value="Imprimir"/>
11	AZUAY MERCEDES LÓPEZ GONZÁLEZ	Lider	9F-FC-73-1C-98	102.69	2023-11-06 15:57:04	<input type="button" value="Imprimir"/>
12	AZUAY MERCEDES LÓPEZ GONZÁLEZ	Lider	D2-AF-03-93-C2	205.39	2023-11-06 15:40:40	<input type="button" value="Imprimir"/>

Registros del 1 al 5 de un total de 5 registros (filtrado de un total de 12 registros)

Anterior **1** Siguiente



Dando click a Acciones.

**DATOS DEL USUARIO**

NOMBRE COMPLETO : MARIO FRANCISCO JATIVA REYES		CI : 1333223	EXPEDIDO : LA PAZ
TELEFONO : 73332233	DIRECCIÓN : ACHACACHI	AAAAAAAAA : AAAAAAAAAAAAAA	
CORREO : mario@gmail.com	COMPENSACION : 4 %	ASIGNADO : AZUAY MERCEDES LÓPEZ GONZÁLEZ	
FUNCIÓN : LIDER	ALCANCE MIN. GRUPO : 600.00	VENTAS MES : NOVEMBER 260.12	
ESTADO: <span style="background-color: #90EE90; padding: 2px;">ACTIVO</span>	VENTA GRUPO EXITOSO: <span style="background-color: #90EE90; padding: 2px;">ACTIVO</span>		

Imprimir

Mostrar  registros
Buscar:

Nº	FUNCIÓN	NONBRE TITULAR	ASESORAS ASIGNADAS	DOCUMENTO	TELEFONO	CORREO
<span style="color: green;">+</span> 1	LIDER	MARIO FRANCISCO JATIVA REYES	LUIS HERMINIO MULLO CEPEDA	213313123 - NIT	72323233	luis@gmail.com
<span style="color: green;">+</span> 2	LIDER	MARIO FRANCISCO JATIVA REYES	IVÁN FRANCISCO RIVADENEIRA TORRES	644535345 - CI	76543322	ivan@gmail.com
<span style="color: green;">+</span> 3	LIDER	MARIO FRANCISCO JATIVA REYES	JUAN JOSE MONTUFAR MARCAYATA	2231323 - NIT	23133132	jose@gmail.com
<span style="color: green;">+</span> 4	LIDER	MARIO FRANCISCO JATIVA REYES	NANCY ISABEL ESCOBAR PAREDES	2766554 - CI	6666563	nancy@gmail.com

Registros del 1 al 4 de un total de 4 registros
Anterior 1 Siguiente

