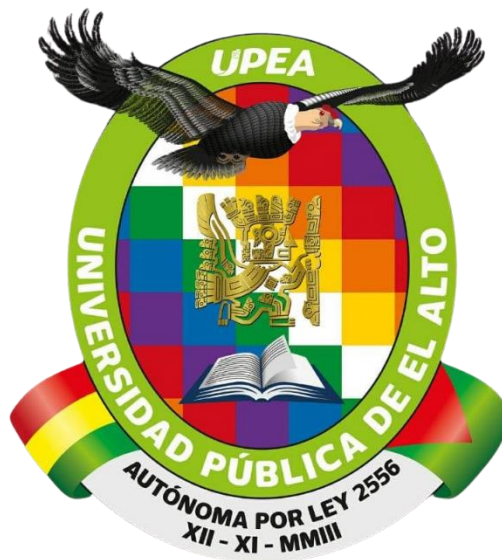


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE VENTAS Y COMPRAS DE DISPOSITIVOS DE SEGURIDAD” CASO: INSTITUCIÓN “VISIÓNTECBOL”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante : Angel Chuquichambi
Tutor Metodológico : M. Sc. Lic. Ing. Marisol Arguedas Balladares
Tutor Especialista : Lic. Margarita Bernarda Lopez Mariaca
Tutor Revisor : Lic. Freddy Salgueiro Trujillo

EL ALTO – BOLIVIA

2024

(TALLER DE GRADO II)

DECLARACIÓN JURADA DE

AUTENTICIDAD Y RESPONSABILIDAD

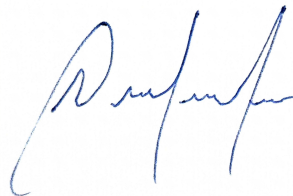
Yo **ANGEL CHUQUICHAMBI** estudiante con **C.I. 13849530 L.P.** mediante la presente declaro de manera pública que la propuesta del **TRABAJO DE GRADO** titulada "**SISTEMA DE INFORMACION WEB PARA LA GESTIÓN DE VENTAS Y COMPRAS DE DISPOSITIVOS DE SEGURIDAD**" **CASO: INSTITUCIÓN VISIONTECBOL**, es original, 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 **TRABAJO 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 do El Alto.

El Alto, Junio del 2024



Angel Chuquichambi
C.I. 13849530 L.P.
angechoque333@gmail.com

DEDICATORIA:

Dedico este proyecto a:

A mi madre querida Albertina, por ser mi mayor inspiración, mi primer y más grande apoyo. Gracias por tu amor incondicional, por inspirarme a ser siempre la mejor versión de mí misma y por alentarme a nunca rendirme.

Tu sacrificio y dedicación han sido fundamentales en mi camino hacia la culminación de este proyecto. Te amo infinitamente.

A mis hermanas, por ser mis compañeras de vida y mis mejores amigas.

A mis padrinos, por ser mis guías y mentores, su apoyo intelectual y emocional ha sido fundamental para alcanzar el éxito en este proyecto.

A mis tutores, por su invaluable ayuda y orientación a lo largo de este proyecto.

AGRADECIMIENTOS

A mi madre

Agradezco profundamente a mi madre, cuyo amor y apoyo incondicional han sido la base de todos mis logros. Gracias por ser mi pilar y por creer en mí en cada paso de este camino.

A mis queridas hermanas, gracias por su constante ánimo y comprensión. Su compañía y palabras de aliento han sido fundamentales para mantenerme motivado y enfocado en mi objetivo.

A mis padrinos, gracias por su guía y consejos sabios. Su respaldo y confianza en mis capacidades han sido una fuente de inspiración para perseverar y alcanzar mis metas.

A mis tutores, agradezco profundamente su dedicación y paciencia. Sus conocimientos y orientación han sido esenciales para el desarrollo de este proyecto, y su compromiso con mi educación ha dejado una huella imborrable en mi formación académica.

A la UPEA y mi carrera de Ingeniería de Sistemas

Agradezco a la Universidad Pública de El Alto por brindarme la oportunidad de crecer y desarrollarme en un entorno académico enriquecedor, y a mi carrera de Ingeniería de Sistemas por desafiarme y expandir mis horizontes. Su compromiso con la educación y las lecciones aprendidas me han preparado para enfrentar el futuro con confianza y habilidad.

ÍNDICE DEL CONTENIDO

CAPÍTULO I: MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES.....	2
1.2.1. Antecedentes Institucionales.....	2
1.2.1.1 Misión.	2
1.2.1.2 Visión.	2
1.2.1.3 Organigrama.....	2
1.2.2. Antecedentes afines al proyecto de grado	3
1.2.2.1 Antecedentes Internacionales.....	3
1.2.2.2 Antecedentes Nacionales	4
1.2.2.3 Antecedentes Locales.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1. Problema Principal.....	5
1.3.2. Problemas Secundarios.....	5
1.4. OBJETIVOS	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos.....	6
1.5. JUSTIFICACIÓN.....	6
1.5.1. Justificación Técnica.....	6
1.5.2. Justificación Económica.....	6
1.5.3. Justificación Social.....	7
1.6. METODOLOGÍA.....	7
1.6.1. Metodología UWE (User-oriented Web Engineering)	7
1.6.2. Métricas de Calidad de Software ISO/IEC 25010	7

1.6.3. Niveles de Prueba.....	8
1.6.3.1 Pruebas unitarias.	8
1.6.3.2 Pruebas de Integración.	8
1.6.3.3 Pruebas de Sistema.....	8
1.6.3.4 Pruebas de Aceptación.	8
1.6.4. Pruebas Unitarias e Integración con PHPUnit	9
1.6.5. Pruebas de Sistema con Selenium	9
1.6.6. Métricas de Seguridad de Software ISO/IEC 27001	9
1.6.7. Estimación de Costos en Software.....	10
1.6.8. Marco de Referencia para el Desarrollo del software ISO/IEC 12207	10
1.7. HERRAMIENTAS	11
1.7.1. Diseño de las Interfaces	11
1.7.2. Base de Datos.....	11
1.7.3. Lenguajes de Programación.....	11
1.7.4. Editor de Códigos.....	12
1.7.5. Servidor Local	12
1.7.6. Framework Laravel Livewire.....	12
1.8. LIMITES Y ALCANCES.....	12
1.8.1. Limites	12
1.8.2. Alcances.....	13
1.9. APORTES.....	14
CAPÍTULO II: MARCO TEÓRICO	16
2.1. INTRODUCCIÓN	16
2.1. DATOS.....	16
2.3. DATOS ESTADÍSTICOS	17

2.3.1. Datos Cualitativos	17
2.3.2. Datos Cuantitativos.....	17
2.3.2.1. Discreto.	17
2.3.2.2 Continuo.	17
2.4. TIPOS DE DATOS EN PROGRAMACIÓN	17
2.4.1. Texto.....	17
2.4.2. Numérico.....	17
2.4.3. Lógico.....	18
2.5. INFORMACIÓN	18
2.5.1. Clasificación	18
2.6. SISTEMA	19
2.7. SISTEMAS DE INFORMACIÓN	19
2.7.1. Características de un Sistema de Información	19
2.7.2. Componentes de un Sistema de Información	19
2.8. WEB	20
2.8.1. Diseño Web.....	20
2.8.1.1 Componentes Principales del Diseño Web.	20
2.8.2. Página Web.....	21
2.8.2.1 Página Web Estática.....	21
2.8.2.2 Página Web Dinámica.....	22
2.8.2.3 Página Web con Base de Datos.	23
2.8.3. Aplicaciones Web.....	24
2.8.3.1 Características de las Aplicaciones Web.....	24
2.8.3.2 Aplicaciones Web Nativas.....	25
2.8.3.3 Aplicaciones Web Progresivas.	25
2.8.4. Servidor Web	25

2.8.4.1 Tipos de Servidores Webs más Utilizados	26
2.8.5. Seguridad Web	26
2.8.5.1 Técnicas y Herramientas de la Seguridad Web.	26
2.9. GESTIÓN.....	27
2.9.1. Elementos de la Gestión.....	27
2.10. GESTIÓN DE VENTAS.....	28
2.10.1. Tipos de Gestión de Ventas.	28
2.11. GESTIÓN DE COMPRAS.....	29
2.11.1. Clasificación de la Gestión de Compras.	30
2.12. INGENIERÍA	30
2.13. SOFTWARE	30
2.14. INGENIERÍA DE SOFTWARE	31
2.14.1. Ciclo de Vida del Desarrollo de Software SDLC.....	31
2.14.1.1 Fase de Requisitos.....	32
2.14.1.2 Fase de Diseño.	32
2.14.1.3 Fase de Implementación.....	32
2.14.1.4 Fase de Pruebas.	32
2.14.1.5 Fase de Despliegue.	32
2.15. UML (LENGUAJE UNIFICADO DE MODELADO)	33
2.15.1. Diagramas UML	33
2.15.1.1 Diagramas de Clase.....	33
2.15.1.2 Diagrama de Implementación.	33
2.15.1.3 Diagrama de Casos de Uso.....	34
2.16. METODOLOGÍA UWE (UML-BASED WEB ENGINEERING).....	35
2.16.1. Características de la metodología UWE	35
2.16.2. Fases de la Metodología UWE.....	36

2.16.3. Modelado UWE	36
2.16.3.1 Análisis de Requisitos	37
2.16.3.2 Diseño Conceptual.....	37
2.16.3.3 Diseño Navegacional.....	38
2.16.3.4 Diseño de Presentación.....	39
2.17. MÉTRICAS DE CALIDAD DE SOFTWARE	40
2.17.1. ISO/IEC 25010	40
2.18. ESTIMACIÓN DE COSTOS.....	41
2.18.1. Cocomo (Constructive Cost Model 1981)	41
2.18.1.1 Cocomo Intermedio.....	41
2.19. SEGURIDAD	43
2.19.1. ISO/IEC 27001	43
2.19.1.1 Protocolos Norma ISO 27001	43
2.20. MODELO EN V	43
2.20.1. Pruebas Unitarias	44
2.20.2. Pruebas de Caja Blanca.....	45
2.20.3. Pruebas de Integración.....	45
2.20.3.1 PHPUnit.....	45
2.20.4. Pruebas de Sistema.....	45
2.20.4.1 Selenium.....	45
2.20.5. Pruebas de Aceptación.....	46
2.21. PRUEBAS DE RENDIMIENTO DE SOFTWARE	46
2.21.1. Pruebas de Estrés	46
2.21.1.1 Apache Jmeter.....	46
2.22. RECOPIACIÓN DE DATOS	47
2.22.1. Entrevistas Personales	47

2.22.2. Observaciones Directa.....	47
2.22.3. Análisis de Documentos	48
2.23. HERRAMIENTAS DE DESARROLLO	48
2.24. BACKEND.....	48
2.24.1. Laravel Livewire	48
2.24.2. PHP	48
2.24.3. MYSQL.....	49
2.25. FRONTEND	49
2.25.1. Bootstrap.....	49
2.25.2. JavaScript.....	49
2.25.3. Html5.....	50
2.25.4. Css3	50
2.25.5. jQuery.....	51
2.25.6. Librería SweetAlert	51
2.26. GESTIÓN DE CÓDIGO Y VISUALIZACIÓN	51
2.26.1. Xampp.....	51
2.26.2. Visual Studio Code (VS Code)	51
2.26.3. Navegador Web	51
CAPÍTULO III: MARCO APLICATIVO	54
3.1. INTRODUCCIÓN	54
3.2. ANÁLISIS DE LA SITUACIÓN ACTUAL.....	54
3.3. ESQUEMA DEL SISTEMA WEB	55
3.4. ANÁLISIS DE LOS REQUERIMIENTOS	55
3.4.1. Obtención de los Requerimientos	55
3.4.2. Requerimientos Funcionales	55

3.4.3. Requerimientos No Funcionales	56
3.5. DISEÑO.....	57
3.5.1. Modelado UWE	57
3.5.1.1 Definición de Actores.	57
3.5.1.2 Definición de Casos de Uso del Sistema.....	59
3.5.1.3 Diseño conceptual.....	66
3.5.1.4 Diseño Navegacional.	67
3.5.1.5 Diseño de Presentación.....	68
3.6. CODIFICACIÓN	69
3.6.1. Configuración e Instalación del Framework Laravel Livewire.....	69
3.6.2. Creación de Modelos y Migraciones	70
3.6.3. Creación de Componentes	72
3.7. PRUEBAS.....	74
3.7.1. Pruebas Unitarias usando el método de Caja Blanca (PHPUnit)	74
3.7.2. Pruebas de Integración usando el método de Caja Blanca (PHPUnit)	76
3.7.3. Pruebas de Sistema usando el método de Caja Negra (Selenium)	78
3.8. IMPLEMENTACIÓN	80
CAPÍTULO IV: CALIDAD, SEGURIDAD Y COSTOS.....	82
4.1. INTRODUCCIÓN	82
4.2. MÉTRICAS DE CALIDAD BASADO EN EL ISO 25010	82
4.2.1. Funcionalidad.....	82
4.2.1.1 Funciones de tipo Entrada Externa (EE).	82
4.2.1.2 Funciones de tipo Salida Externa (SE).	83
4.2.1.3 Funciones de Tipo Consulta Externa (EQ).	83
4.2.1.4 Funciones de Tipo Archivo Lógico Interno (ILF).....	84

4.2.1.5 Funciones de Tipo Archivo Lógico Externo (ELF).....	84
4.2.1.6 Cálculo de los Puntos de Función Sin Ajustar (PFSA).....	84
4.2.1.7 Estimación de Funcionalidad.....	85
4.2.2. Fiabilidad.....	86
4.2.2.1 Definición de Datos.....	86
4.2.3. Compatibilidad.....	87
4.2.3.1 Plan de pruebas de navegador.....	87
4.2.3.2 Plan de pruebas de dispositivos.....	89
4.2.3.3 Estimación de compatibilidad del Sistema.....	90
4.2.4. Usabilidad.....	92
4.2.5. Portabilidad.....	92
4.2.6. Resultados Finales de Calidad.....	94
4.3. ESTIMACIÓN DE COSTOS.....	94
4.3.1. Estimación de costos usando el método COCOMO.....	94
4.3.2. Identificación de los datos iniciales.....	95
4.3.3. Cálculo de FAE (Factor de Ajuste de Esfuerzo).....	96
4.3.4. Cálculo del Esfuerzo (E).....	96
4.3.5. Cálculo del Tiempo de desarrollo (T).....	97
4.3.6. Cálculo de la productividad (PR).....	97
4.3.7. Cálculo del personal promedio (P).....	97
4.3.8. Cálculo del costo del proyecto.....	97
4.4. SEGURIDAD BASADO EN LA ISO 27001.....	98
4.4.1 Seguridad Física.....	98
4.4.2 Seguridad Lógica.....	98
4.5. PRUEBAS DE RENDIMIENTO DE SOFTWARE.....	99
4.5.1. Pruebas de Estrés.....	99

4.5.2. Herramienta Apache JMeter.....	100
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....	105
5.1 CONCLUSIONES.....	105
5.2 RECOMENDACIONES.....	105
REFERENCIAS BIBLIOGRÁFICAS.....	106
ANEXOS.....	110

ÍNDICE DE TABLAS

TABLA 1 DEFINICIÓN DE ACTORES	56
TABLA 2 PLANILLA DE USO DE CASO - GESTIONAR USUARIOS.....	57
TABLA 3 DEFINICIÓN DE ACTORES	58
TABLA 4 PLANTILLA DE CASO DE USO GESTIONAR USUARIOS.....	60
TABLA 5 PLANTILLA DE CASO DE USO GESTIONAR PRODUCTOS	61
TABLA 6 PLANTILLA DE CASO DE USO GESTIONAR CLIENTES.....	62
TABLA 7 PLANTILLA DE CASO DE USO REGISTRAR VENTAS.....	63
TABLA 8 PLANTILLA DE CASO DE USO REGISTRAR COMPRAS.....	64
TABLA 9 PLANTILLA DE CASO DE USO GESTIONAR PROVEEDORES	65
TABLA 10 FUNCIONES DE ENTRADAS EXTERNAS (EE)	82
TABLA 11 FUNCIONES DE SALIDA EXTERNA	83
TABLA 12 FUNCIÓN DE CONSULTA EXTERNA.....	83
TABLA 13 FUNCIÓN DE ARCHIVO LÓGICO INTERNO	84
TABLA 14 FUNCIÓN DE ARCHIVO LÓGICO EXTERNO.....	84
TABLA 15 CÁLCULO DE PUNTOS DE FUNCIÓN SIN AJUSTAR	84
TABLA 16 FACTOR DE AJUSTE	85
TABLA 17 PRUEBA DE NAVEGADORES WEB.....	90
TABLA 18 PRUEBA DE DISPOSITIVOS	91
TABLA 19 EVALUACIÓN DE USABILIDAD.....	92
TABLA 20 EVALUACIÓN DE LA PORTABILIDAD	93
TABLA 21 RESULTADOS FINALES.....	94
TABLA 22 CALCULO DE LA VARIABLE FAE	96

ÍNDICE DE FIGURAS

FIGURA 1 ORGANIGRAMA VISIÓNTECBOL.....	2
FIGURA 2 PROCESO DE TRANSFORMACIÓN DE DATOS EN INFORMACIÓN.....	16
FIGURA 3 TOMA DE DECISIONES: TRANSFORMACIÓN DE INFORMACIÓN EN ACCIÓN	18
FIGURA 4 PÁGINA WEB ESTÁTICA.....	22
FIGURA 5 PÁGINA WEB DINÁMICA.....	23
FIGURA 6 PÁGINA WEB CON BASE DE DATOS	24
FIGURA 7 CAPAS DE LA INGENIERÍA DE SOFTWARE.....	31
FIGURA 8 CICLO DE VIDA CLÁSICO	32
FIGURA 9 DIAGRAMA DE CLASE	33
FIGURA 10 DIAGRAMA DE IMPLEMENTACIÓN	34
FIGURA 11 DIAGRAMA DE CASOS DE USO.....	35
FIGURA 12 DISEÑO DE CASOS DE USO	37
FIGURA 13 EJEMPLO DEL MODELO CONCEPTUAL.....	38
FIGURA 14 EJEMPLO DEL MODELO NAVEGACIONAL	39
FIGURA 15 EJEMPLO DE MODELADO DE PRESENTACIÓN.....	39
FIGURA 16 TIPOS DE PROYECTO Y SU COMPLEJIDAD.....	42
FIGURA 17 CONDUCTORES DE COSTOS FAE.....	42
FIGURA 18 EL MODELO EN V	44
FIGURA 19 VENTAJAS DE APACHE JMETER	47
FIGURA 20 ESTRUCTURA DE UN DOCUMENTO HTML	50
FIGURA 21 ESQUEMA DEL SISTEMA WEB.....	55
FIGURA 22 DIAGRAMA DE CASOS DE USO GENERAL DEL SISTEMA.....	59
FIGURA 23 CASO DE USO GESTIONAR USUARIOS	60
FIGURA 24 CASO DE USO GESTIONAR PRODUCTOS.....	61
FIGURA 25 CASO DE USO GESTIONAR CLIENTES.....	62

FIGURA 26 CASOS DE USO GESTIONAR VENTA.....	63
FIGURA 27 CASO DE USO REGISTRAR COMPRAS.....	64
FIGURA 28 CASO DE USO GESTIONAR PROVEEDORES.....	65
FIGURA 29 DISEÑO CONCEPTUAL DEL SISTEMA.....	66
FIGURA 30 DIAGRAMA DE NAVEGACIÓN DEL SUPERVISOR	67
FIGURA 31 DIAGRAMA DE NAVEGACIÓN DEL ADMINISTRADOR	67
FIGURA 32 MODELO DE PRESENTACIÓN - INICIAR SESIÓN.....	68
FIGURA 33 MODELO DE PRESENTACIÓN - PÁGINA PRINCIPAL.....	68
FIGURA 34 INSTALACIÓN DE LARAVEL.....	69
FIGURA 35 INSTALACIÓN DE LIVEWIRE	69
FIGURA 36 FRAMEWORK LARAVEL LIVEWIRE	70
FIGURA 37 CREACIÓN DE MODELOS	70
FIGURA 38 CONFIGURACIÓN DEL MODELO "VENTAS"	71
FIGURA 39 CREACIÓN DE MIGRACIONES (BASE DE DATOS DEL SISTEMA)	71
FIGURA 40 DEFINICIÓN DE ATRIBUTOS DE LA MIGRACIÓN "VENTAS".....	72
FIGURA 41 CREACIÓN DE COMPONENTES CON LIVEWIRE.....	72
FIGURA 42 CONTENIDO DEL COMPONENTE "VENTAS"	73
FIGURA 43 VISTA DEL COMPONENTE "VENTAS"	73
FIGURA 44 CÓDIGO DE LA FUNCIÓN "STORE" DEL COMPONENTE CLIENTE.....	74
FIGURA 45 PRUEBA DE CÓDIGO A LA FUNCIÓN STORE	75
FIGURA 46 EJECUCIÓN DE LA PRUEBA.....	75
FIGURA 47 LLAMADO DE VARIOS COMPONENTES.....	76
FIGURA 48 PRUEBA A LA FUNCIÓN VENTA.....	77
FIGURA 49 EJECUCIÓN DE LA PRUEBA.....	77
FIGURA 50 CONFIGURACIÓN DE NOMBRE DEL PROYECTO.....	78
FIGURA 51 VISTA CLIENTES DE PRUEBA.....	78

FIGURA 52 CONFIGURACIÓN DE PARÁMETROS URL	79
FIGURA 53 EJECUCIÓN DE PRUEBA CON SELENIUM	79
FIGURA 54 RESULTADOS FINALES DE LA PRUEBA	80
FIGURA 55 COMPATIBILIDAD CON GOOGLE CHROME.....	88
FIGURA 56 COMPATIBILIDAD CON FIREFOX	88
FIGURA 57 COMPATIBILIDAD CON MICROSOFT EDGE	89
FIGURA 58 PRUEBA CON DISPOSITIVO DE ESCRITORIO.....	89
FIGURA 59 PRUEBA CON DISPOSITIVO MÓVIL	90
FIGURA 60 DEFINICIÓN DE PARÁMETROS.....	100
FIGURA 61 CONFIGURACIÓN DEL THREAD GROUP	101
FIGURA 62 PRUEBA EN EJECUCIÓN	101
FIGURA 63 RESULTADOS CON SUMMARY REPORT	102
FIGURA 64 RESULTADOS CON VIEW RESULTS IN TABLE.....	102
FIGURA 65 RESULTADO CON GRAPH RESULTS	103

RESUMEN

En la actualidad, la tecnología web transforma la operación de las instituciones, facilitando el acceso y la compartición de información. El proyecto de grado "SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE VENTAS Y COMPRAS DE DISPOSITIVOS DE SEGURIDAD" aborda los desafíos de VISIÓNTECBOL, una empresa especializada en dispositivos de seguridad. Los problemas incluyen desorganización de información, falta de datos precisos, demora en atención al cliente y pérdida de información valiosa. El objetivo es desarrollar un sistema web para centralizar y organizar las transacciones de ventas y compras, garantizando la seguridad de los datos conforme a los estándares ISO 27001 e ISO 25010. Utilizando la metodología UWE, el proyecto incorpora tecnologías como Laravel Livewire, Bootstrap, JavaScript, jQuery, Html, Css, Visual Studio Code, Xampp y MySQL Workbench, asegurando la participación activa de los usuarios y mejorando la satisfacción del cliente.

ABSTRACT

Currently, web technology transforms the operation of institutions, facilitating access and sharing of information. The degree project "WEB INFORMATION SYSTEM FOR THE MANAGEMENT OF SALES AND PURCHASES OF SECURITY DEVICES" addresses the challenges of VISIÓNTECBOL, a company specialized in security devices. Problems include disorganization of information, lack of accurate data, delay in customer service, and loss of valuable information. The objective is to develop a web system to centralize and organize sales and purchasing transactions, guaranteeing data security in accordance with ISO 27001 and ISO 25010 standards. Using the UWE methodology, the project incorporates technologies such as Laravel Livewire, Bootstrap, JavaScript , jQuery, Html, CSS, Visual Studio Code, Xampp and MySQL Workbench, ensuring active user participation and improving customer satisfaction.

CAPÍTULO I

CAPÍTULO I: MARCO PRELIMINAR

1.1. Introducción

Actualmente, el uso de los Sistemas Web se ha convertido en una herramienta fundamental e importante para la gestión de negocios de diferentes instituciones. Este tipo de Sistema ofrece numerosos beneficios, como ser una gestión automatizada, eficiente y precisa de las ventas y las compras realizadas.

El problema principal que se aborda en este proyecto se basa en la tendencia aún existente de registrar sus ventas y compras a proveedores de forma manual. Este enfoque presenta limitaciones significativas, como errores humanos, pérdida de información y falta de visibilidad en tiempo real sobre el estado real de las ventas. Estas deficiencias pueden tener un impacto negativo en la toma de decisiones, la satisfacción del cliente y la rentabilidad general del negocio.

Para abordar el problema mencionado con anterioridad, se propone desarrollar un Sistema de Información Web que Gestione las Ventas y Compras de Dispositivos de Seguridad. Además, que tiene como fin proporcionar a los usuarios una interfaz fácil de usar y amigable que les permita acceder y utilizar de manera eficiente el sistema. De esta manera, se mejora la eficiencia y precisión en la gestión de ventas y compras, así como también facilitar la toma de decisiones basadas en datos en tiempo real.

Para llevar a cabo el presente proyecto, se utiliza la metodología UWE (Unifying Web Engineering) la cual permite el desarrollo de sistemas web de manera estructurada, tomando en cuenta aspectos como la obtención de los requerimientos, diseño, pruebas e implementación.

En cuanto a las herramientas de desarrollo, se hace uso de tecnologías de desarrollo como ser Framework Livewire Laravel, HTML, CSS, JavaScript, PHP, MySQL, Visual Studio Code y Xampp para crear un Sistema robusto y escalable. Además de utilizar herramientas de

visualización de datos para asegurar el correcto desarrollo del sistema y la generación de informes y análisis de los datos de venta.

1.2. Antecedentes

1.2.1. Antecedentes Institucionales

La institución “VISIONTECBOL” entro en operaciones en enero del 2023, dicha institución se encuentra ubicada en la Zona Faro Murillo, entre calle 6 y 7 de la ciudad de El Alto, está se dedica al comercio de equipos electrónicos de seguridad. Los equipos que la institución ofrece a sus clientes, vienen de distintos proveedores que son a elección de los administradores.

1.2.1.1 Misión. Ser una de las mejores instituciones en ofrecer equipos electrónicos de última tecnología con las mejores calidades y garantías. Además de brindar soluciones de soporte a sus clientes.

1.2.1.2 Visión. Ser una de las mejores elecciones del mercado en la venta de equipos electrónicos de seguridad y para brindar soluciones de soporte.

1.2.1.3 Organigrama

Figura 1

Organigrama VISIÓNTECBOL



1.2.2. Antecedentes afines al proyecto de grado

A continuación, se indicarán algunos antecedentes locales, nacionales e internacionales que han sido de gran utilidad como guía. Estos antecedentes proporcionaran una vista amplia y valiosa para abordar los temas centrales del proyecto de grado.

1.2.2.1 Antecedentes Internacionales.

- Luis Daniel Llerena Yamba, **“Sistema E-commerce Para la Gestión de Ventas en Power PC JR. Ubicada en la Ciudad de Quito”**. El proyecto comprendió el desarrollo de un sistema de Internet para la empresa Power PC Jr. proporciona soluciones informáticas y hardware para el comercio minorista técnica. Desarrollar un sistema en línea con este tipo de actividad; relacionado con las ventas y reservas, un estudio en profundidad de las actividades y procesos que realizadas por los usuarios para brindar una solución tecnológica e innovadora mejorar el flujo de negocios. Así es como la aplicación te permite hacerlo. Compre en línea en cualquier momento, se muestran los productos más populares las imágenes, la información recibida y el estado del pedido se muestran y se usar. (Llerena Yamba, 2019)
- Aldeyr Junior Romero Carhuamaca, **“Sistema de Informacion Web para Mejorar la Gestion de Ventas e Inventario en la Ferreteria R&C”**. La tesis de grado abarca el desarrollo de un sistema informático de Internet, también conocido como aplicación de Internet, que puede realizar actividades o tareas como parte del proceso de ventas (registro, modificación, eliminación) e inventario (introducción de productos, lanzamiento de productos). en la organización; Esto tiene como objetivo mejorar la gestión de la información y potenciar los procesos que se llevan a cabo dentro de la organización. (Romero Carhuamaca, 2023)
- Juan Carlos Castilla, **"Sistema de inventario y ventas en línea para la tienda virtual de la Universidad Nacional de Colombia"**. El objetivo del proyecto de grado fue implementar un Sistema web para el control de ventas e inventario en línea que

permitiera a la tienda virtual de la universidad ofrecer sus productos de manera más eficiente y llegar a más clientes potenciales. El Sistema desarrollado permitió a los usuarios realizar pedidos, realizar pagos en línea y hacer seguimiento a sus compras. Además, el Sistema permitió a la tienda virtual de la universidad tener un mejor control de su inventario y de las ventas realizadas. (Castilla, 2020)

1.2.2.2 Antecedentes Nacionales

- Jhony Mamani, “**Sistema de control de inventarios para la empresa de transporte de carga pesada Trans-River**”. El objetivo del proyecto de Jhony Mamani fue diseñar e implementar un Sistema de control de inventarios para la empresa Trans-River, con el fin de mejorar la eficiencia y precisión en la gestión de sus existencias. (Mamani, 2021)
- Mauricio Choque, “**Sistema de gestión de ventas e inventario para la tienda de productos naturales Ayni**”. El objetivo del proyecto de Mauricio Choque fue diseñar e implementar un Sistema de gestión de ventas e inventario para la tienda Ayni, con el fin de mejorar la eficiencia y precisión en la gestión de sus existencias y ventas, así como la satisfacción de sus clientes. (Choque, 2020)

1.2.2.3 Antecedentes Locales

- Esther Eva Calle Limachi, “**Sistema de Información Para El Control De Inventario De Ventas**” **Caso: (Victory Motors Bolivia)**. Como objetivo se tenía lo siguiente: Desarrollar e implementar el Sistema de información para el control de inventario de venta, como una herramienta en la administración y la optimización de la información en la empresa Victory Motors Bolivia de la ciudad de El Alto. (Calle Limachi, 2020).

1.3. Planteamiento Del Problema

Después de realizar el análisis de la situación actual se ha determinado los siguientes problemas.

1.3.1. Problema Principal

En la institución VISIÓNTECBOL, el proceso de atención, la gestión de sus ventas y compras se realiza de forma manual. Esto implica que el personal administrativo debe buscar y recopilar información específica de diversas fuentes, lo cual puede resultar en demoras y pérdida de tiempo. Además, esta forma de trabajo manual puede llevar a la desorganización de la información, lo que hace más difícil su acceso, pérdidas económicas y toma de decisiones incorrectas.

1.3.2. Problemas Secundarios

- La atención al cliente en VISIONTECBOL es manual, lo que retrasa el tiempo de respuesta y genera insatisfacción debido a la búsqueda de información en múltiples documentos.
- Ausencia de información actualizada sobre precios, disponibilidad y existencias de los productos, provocando retrasos en las ventas y posibles insatisfacciones en los clientes.
- Existe un mayor riesgo de pérdida de información de ventas debido a errores del personal, extravío de documentos y a la falta de organización en la gestión de la información.
- No se cuenta con registros detallados y centralizado de las compras realizadas, lo que dificulta el control, la trazabilidad y la toma de decisiones informadas sobre las operaciones de compra.

1.4. Objetivos

En el presente proyecto se establecen los siguientes objetivos general y específicos:

1.4.1. Objetivo General

Desarrollar un Sistema Web para la Gestión de Ventas y Compras para la institución VISIÓNTECBOL, que permita automatizar los procesos manuales, optimizando la búsqueda y organización de la información y además facilitando la toma de decisiones estratégicas.

1.4.2. Objetivos Específicos

- Realizar un análisis exhaustivo de los requisitos y necesidades del personal administrativo en cuanto a la gestión de ventas y compras, para diseñar un sistema de información web que se ajuste a sus necesidades.
- Desarrollar un módulo de control de inventario que permita llevar un registro detallado y actualizado de los productos disponibles, incluyendo información como cantidades, ubicaciones, fechas, niveles de stock mínimos.
- Implementar un módulo de ventas que permita al personal tener una visión clara de las ventas realizadas, los ingresos generados y completar transacciones de manera rápida y segura.
- Crear un módulo de compras eficiente que permita al personal tener un control detallado sobre las compras realizadas, proveedores de manera ágil y segura.

1.5. Justificación

1.5.1. Justificación Técnica

El presente proyecto se justifica técnicamente por que actualmente la institución posee con los recursos adecuados de software y hardware. Como además de que cuenta con la tecnología necesaria para realizar el desarrollo del Sistema de Información Web para su debida Gestión de sus Ventas y Compras.

1.5.2. Justificación Económica

Se justifica económicamente por que la implementación del Sistema permitirá a la institución reducir costos operativos y aumentar la eficiencia en la gestión de sus productos.

La automatización de los procesos manuales permitirá a la institución ahorrar tiempo y recursos que pueden ser utilizados en otras áreas. Además de reducir costos de almacenamiento y gestión de documentos, ya que la información estará centralizada y con total accesibilidad en línea. Todo ello evitando pérdidas económicas por los posibles errores y malversaciones por la falta control.

Por otra parte, para el desarrollo del Sistema se empleará el uso del software libre, ya que este nos ahorra costos en licencias y además de que nos ofrece herramientas de alta calidad y personalizables.

1.5.3. Justificación Social

El Sistema beneficiara tanto a la institución como al personal administrativo, facilitando el rápido manejo de la información clara y precisa, por lo que garantiza que la toma de decisiones sea correcta.

El desarrollo del Sistema coadyuvará a la institución mejorar la calidad del servicio que ofrece a sus clientes, ya que se podrá tener una mejor gestión de las ventas y de las compras.

1.6. Metodología

En relación con la metodología de este proyecto, se ha optado por emplear una serie de enfoques y procedimientos que permitirán alcanzar los objetivos propuestos de manera eficiente y efectiva. A continuación, se detallan los siguientes puntos:

1.6.1. Metodología UWE (User-oriented Web Engineering)

La metodología de UWE se basa en la premisa de que el diseño y desarrollo de sitios web debe orientarse hacia las necesidades y preferencias del usuario final. Para lograr esto, UWE proporciona un conjunto de métodos y herramientas organizados en un proceso de desarrollo interactivo y orientado al usuario. El proceso UWE consta de cuatro fases principales: especificación de requisitos, diseño conceptual, diseño detallado y generación de código. Cada fase se centra en aspectos específicos del proceso de desarrollo web, desde la definición de requisitos hasta la implementación y prueba del sitio web. (Seidl, Kappel y Proll, 2005)

1.6.2. Métricas de Calidad de Software ISO/IEC 25010

La norma ISO 25010 es un modelo de calidad que incluye ocho características relacionadas con las propiedades estáticas del software y las propiedades dinámicas del sistema informático. El modelo es aplicable tanto a sistemas informáticos como a software.

- Funcionalidad
- Rendimiento
- Compatibilidad
- Usabilidad
- Fiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

(Ormeño Rojas, 2019)

1.6.3. Niveles de Prueba

1.6.3.1 Pruebas unitarias. Implementados en un módulo de software específico por programadores, las pruebas de desarrollo aseguran que cada módulo funcione de manera autónoma como se espera, facilitando las pruebas de integración. Para este tipo de pruebas se utiliza el método de caja blanca, mediante la herramienta PHPUNIT. (Admin, 2020)

1.6.3.2 Pruebas de Integración. Verifican que los subsistemas de un producto interactúan correctamente tras las pruebas unitarias, identificando errores de comunicación y asegurando que cumplen con las funcionalidades y requisitos del cliente. Al igual que las pruebas unitarias, este tipo de pruebas se realiza con el método de caja blanca, utilizando la herramienta PHPUNIT. (Admin, 2020)

1.6.3.3 Pruebas de Sistema. Estas pruebas, conocidas como pruebas funcionales, garantizan el correcto funcionamiento de la integración del sistema, evaluando el comportamiento del producto desde la perspectiva del usuario final. Se realizan en un entorno casi realista para verificar que todo cumple con las especificaciones y requisitos del cliente. La herramienta que se utiliza para este tipo de pruebas es Selenium, mediante el método de caja negra. (Admin, 2020)

1.6.3.4 Pruebas de Aceptación. Se llevan a cabo después de que el área de pruebas

ha finalizado sus pruebas programadas, siendo realizadas principalmente por el cliente o usuario final para confirmar que se han cumplido sus expectativas. (Admin, 2020)

1.6.4. Pruebas Unitarias e Integración con PHPUnit

Las pruebas de PHPUnit permiten a los desarrolladores encontrar problemas lo antes posible en el ciclo de desarrollo, lo que garantiza que el código no se someta a la ingeniería inversa.

Características principales:

- Pruebas de nuevas funciones
- Refactorización de revisión de código reciente
- Se puede automatizar completamente y ejecutar como parte de un proceso de integración continua de su elección
- Depuración
- Documentación automática del uso previsto llamando a funciones y restricciones
- Habilita el desarrollo basado en pruebas (TDD) son el foco, donde los desarrolladores primero crean pruebas y luego funcionalidades.

(Losoviz, 2022)

1.6.5. Pruebas de Sistema con Selenium

Selenium es una herramienta de automatización de pruebas para aplicaciones web que utiliza scripts en diferentes lenguajes de programación. Su principal función es automatizar tareas repetitivas en el navegador, permitiendo realizar pruebas de forma eficiente y precisa.

(hiberus, 2023)

1.6.6. Métricas de Seguridad de Software ISO/IEC 27001

ISO 27001, es una norma internacional que especifica los requisitos para la implementación, mantenimiento y mejora continua de un sistema de gestión de seguridad de la información (SGSI). La norma aborda tanto la seguridad física (Protección de los activos de una organización) como la seguridad lógica (Protección de los activos digitales y de

información). Este sistema sirve para proteger la confidencialidad, integridad y disponibilidad de la información. Este estándar proporciona un marco de seguridad de la información que ayuda a las organizaciones a identificar y administrar de manera efectiva los riesgos de seguridad de la información.

- **Seguridad Lógica:** Autenticación, Contraseñas Encriptadas, Captcha, Backup's, Seguridad de la red, entre otros más.
- **Seguridad Física:** Sistema de vigilancia, Control de acceso físico, Cerraduras y llaves, entre otros más.

(GlobalSuite Solutions, 2023)

1.6.7. Estimación de Costos en Software

COCOMO (COnstructive COst MOdel), es un modelo de costo objetivo para la planificación e implementación de proyectos de software. Es un componente esencial para la gestión de proyectos de software o negocios de software. El modelo de costos proporciona un marco para el intercambio de decisiones comerciales entre las partes interesadas en el desarrollo de software. COCOMO II incluye una serie de mejoras probadas en el campo que amplían la aplicabilidad y mejoran la precisión de las estimaciones para los métodos modernos de desarrollo de software. El primero es un marco para describir un proyecto de software, incluidos los modelos de proceso, la cultura, las partes interesadas, los métodos, las herramientas y el tamaño/complejidad del software. (Sharma, 2011)

1.6.8. Marco de Referencia para el Desarrollo del software ISO/IEC 12207

La norma ISO 12207 se encarga de todo lo relacionado con el ciclo de vida del software, desde la conceptualización de ideas hasta la retroalimentación, y cubre la compra y entrega de proyectos y servicios de software, estableciendo lineamientos para el control y mantenimiento. El propósito más importante de este estándar es proporcionar un marco común para que compradores, proveedores, desarrolladores, mantenedores, operadores, gerentes y técnicos involucrados en el desarrollo de software utilicen un lenguaje común. (Comejo, 2015)

1.7. Herramientas

Para el presente proyecto se utilizarán las siguientes herramientas en cada fase del proceso de desarrollo siguiendo la metodología UWE:

1.7.1. Diseño de las Interfaces

HTML5 (Lenguaje de Marcado de Hipertexto) es un lenguaje compuesto por entidades que ayudan a estructurar y dar sentido a las diferentes partes de un documento. Cada una de estas entidades, comúnmente denominadas elementos o etiquetas, consta de contenido y cero, uno o más atributos. Cada uno de los atributos tiene una función y puede o no estar asociado a un comportamiento específico. (Casado, 2023)

CSS3 (Hojas de Estilo en Cascada) es un lenguaje de programación muy similar a HTML que le permite aplicar estilos a diferentes elementos de las páginas web para que los encabezados, las listas y los párrafos se vean iguales en todas las páginas. (A. C. Luna, 2019)

1.7.2. Base de Datos

MYSQL es un motor de base de datos relacional de código abierto desarrollado en 1994 por la empresa sueca MYSQL AB. En 2008, Microsystems compró MYSQL AB. MYSQL está escrito en C/C++. Es un motor de base de datos con conexión MYSQL que se ejecuta a través de un cliente/servidor. Hay un servidor MYSQL esperando conexiones y uno o más clientes se están conectando a él. (Urriaga, 2020)

1.7.3. Lenguajes de Programación

PHP (abreviatura de "PHP: preprocesador de hipertexto") es un lenguaje de código abierto altamente interpretado que está incrustado en páginas HTML y se ejecuta en el servidor. Se puede usar en cualquiera de los principales Sistemas operativos del mercado, entre estas están: Windows, Mac OS, Linux y posiblemente algunos otros. Entonces, con PHP, tiene la libertad de elegir el Sistema operativo y el servidor de su elección. (Fossati, 2018)

JavaScript habitualmente abreviado como JS, es un lenguaje de programación de tipo interpretado. Se basa en el estándar ECMAScript y, a pesar de su escritura y dinámica sueltas, por otra parte, se define como un lenguaje orientado a objetos. (F. Luna, 2019)

1.7.4. Editor de Códigos

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es un software gratuito y multiplataforma disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, admite la depuración de código y tiene toneladas de extensiones que básicamente le brindan la capacidad de escribir y ejecutar código en cualquier lenguaje de programación. (Flores, 2022)

1.7.5. Servidor Local

Xampp es una distribución de Apache que incluye varios tipos de software libre. Fue desarrollado por Apache Friends. El programa se distribuye bajo la licencia GNU. El nombre es una abreviatura formada por las iniciales de los respectivos programas: Apache, MYSQL, PHP, Perl. La **X** inicial representa los Sistemas operativos Linux, Windows y Mac OS X. (Bou, 2019)

1.7.6. Framework Laravel Livewire

Laravel Livewire es un framework full-stack que permite construir interfaces de usuario dinámicas y reactivas sin necesidad de escribir una sola línea de JavaScript. Livewire facilita la creación de aplicaciones web modernas y reactivas utilizando solo PHP y Blade (el motor de plantillas de Laravel). (Eseme, 2023)

1.8. Límites Y Alcances

El presente proyecto tiene los siguientes límites y alcances:

1.8.1. Límites

- El Sistema solo se enfocará en la gestión de ventas y compras, por ende, no se encargará de otros procesos de la institución.

- El Sistema estará limitado a la información disponible en la base de datos actual de la institución.
- La manipulación y actualización de los datos solo serán manipulados por el personal administrativo de la institución.
- no podrán acceder al sistema personas no autorizadas.
- no genera la factura en línea.

1.8.2. Alcances

La implementación de este Sistema Web y la Implantación, coadyuvará en el manejo y la gestión de información en las ventas que este opera dentro de la institución "VISIONTECBOL", lo cual esto producirá que se minimice las pérdidas de información, duplicidad, entre otros aspectos más.

Los módulos con los que contará el Sistema son los siguientes:

- **Módulo de altas y bajas para usuarios Administradores:** Permite la incorporación y la desvinculación de los administradores del sistema. Este componente esencial brinda la posibilidad de agregar individuos y, a su vez, realizar las respectivas bajas, otorgando así una gestión impecable y fluida del registro de usuarios.
- **Módulo de altas y bajas para Clientes:** Permite la incorporación y la desvinculación de los clientes del sistema. Este componente esencial brinda la posibilidad de agregar individuos y, a su vez, realizar las respectivas bajas, otorgando así una gestión impecable y fluida del registro de usuarios.
- **Módulo de ventas:** Facilita la realización de ventas, desde el registro de compras hasta la generación de facturas y recibos. Puede incluir funciones como búsqueda de productos, cálculo de precios y descuentos, gestión de pagos y generación de reportes de ventas.

- **Módulo de compras:** Permite realizar compras de productos a proveedores, registrando los detalles de las transacciones, controlando los costos y plazos de entrega, y generando informes de compras.
- **Módulo de clientes:** Proporciona la gestión de datos y seguimiento de los clientes, incluyendo información personal, historial de compras, preferencias, seguimiento de pagos, entre otros.
- **Módulo de proveedores:** Permite mantener registros de los proveedores, gestionar los detalles de las transacciones comerciales, como los precios, plazos y condiciones de pago, y generar informes relacionados con los proveedores.
- **Módulo de reportes:** Ofrece la capacidad de generar informes detallados sobre ventas, compras, clientes, proveedores, entre otros aspectos relevantes del negocio.
- **Módulo de Recibos:** Permite generar recibos de ventas en formato PDF. Puede incluir secciones como creación de facturas internas(recibos), gestión de pagos, y generación de reportes de facturación.
- **Módulo de análisis de ventas:** Proporciona herramientas y gráficos para analizar las ventas realizadas en diferentes períodos, identificar tendencias, comparar productos y generar informes de análisis de ventas.

1.9. Aportes

Los principales aportes que trae el desarrollo de este Sistema son, la optimización de procesos operativos, la toma de decisiones basada en datos precisos y tiempo real, análisis predictivo para la planificación estratégica y mejoramiento en la interacción con los clientes. Por ende, estos permiten cubrir las necesidades de la institución.

CAPÍTULO II

CAPÍTULO II: MARCO TEÓRICO

2.1. Introducción

En la era digital actual, los sistemas de información web desempeñan un papel esencial en la facilitación de la comunicación, la gestión de datos y la interacción en línea. Este marco teórico tiene como objetivo proporcionar una sólida base conceptual para comprender y evaluar la arquitectura y el funcionamiento de los sistemas de información web. Se apoya en teorías y conceptos clave relacionados con la usabilidad, la metodología, herramientas, la seguridad y otros aspectos esenciales que influyen en el diseño y el rendimiento de estos sistemas.

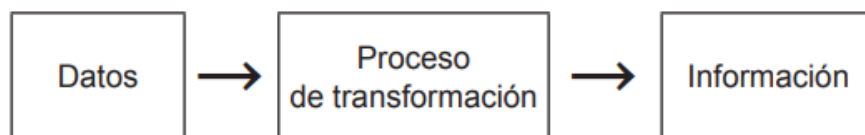
2.1. Datos

Los datos son la representación de una variable, que puede ser cuantitativa o cualitativa, que indica el valor asignado a las cosas y está representada por una cadena de símbolos, números o letras. Los datos describen un evento experimental. Para estudiarlos hay que organizarlos o tabularlos porque un solo dato no puede demostrar mucho; más bien, es necesario evaluar toda la información para verificar los resultados. (Equipo editorial, 2020)

Los datos son caracteres no aleatorios que representan el valor de un atributo o evento. Entonces los datos son hechos que se obtienen leyendo, observando, calculando, midiendo, etc. Por ejemplo, en una organización podemos llamar volúmenes de datos y otros detalles de factura o cheque o detalles de pago de salario, etc. Los datos se recopilan automáticamente de determinadas actividades rutinarias, como procesos de facturación o medición. (Lapiedra Alcamí y otros, 2011)

Figura 2

Proceso de transformación de datos en información



Nota. (Lapiedra Alcamí y otros, 2011)

2.3. Datos Estadísticos

Se diferencian de 2 tipos:

2.3.1. Datos Cualitativos

Son datos relacionados con características o propiedades de un objeto o fenómeno. Estos datos no se pueden medir numéricamente y se recopilan mediante observación, entrevistas o análisis de contenido.

2.3.2. Datos Cuantitativos

Se refieren a cantidades o medidas numéricas. Entre los datos cuantitativos más usados están los siguientes:

2.3.2.1. Discreto. Son datos que solo pueden tomar valores enteros y no pueden ser divididos en partes más pequeñas. Por ejemplo (El número de alumnos en una clase, Número de artículos que hay en stock).

2.3.2.2 Continuo. Son aquellos que pueden tomar cualquier valor dentro de un rango. Por ejemplo (La estatura de una persona, El peso de un grupo de personas).

2.4. Tipos De Datos En Programación

Los tipos de datos se refieren a la clasificación de los datos según sus características. Los tipos de datos son importantes porque definen el conjunto de valores que puede tomar una variable y las operaciones que se pueden realizar sobre ella.

2.4.1. Texto

- **Caracteres:** Son valores que representan un solo carácter, como una letra o un símbolo.
- **Cadenas de Caracteres:** Son valores que representan una secuencia de caracteres.

2.4.2. Numérico

- **Enteros:** Son aquellos números que no tienen decimales.
- **Reales:** Son aquellos números que tienen decimales.

2.4.3. Lógico

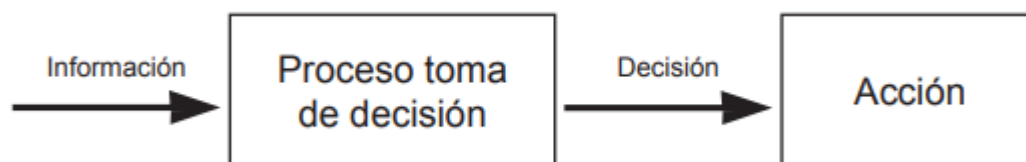
Son aquellos valores que presentan dos casos: verdadero o falso. En programación, estos valores se utilizan para tomar decisiones

2.5. Información

La información es una colección de datos transformados de tal manera que ayudan a reducir la incertidumbre del futuro y así ayudan en la toma de decisiones. La información son datos que se han transformado significativamente. La persona que los recibe, es decir, tiene un valor real o percibido de sus decisiones y por sus acciones. Entonces, la información son datos que han sido interpretados y fáciles de entender para el destinatario del mensaje. (Lapiedra Alcamí y otros, 2011)

Figura 3

Toma de decisiones: transformación de información en acción



Nota. (Lapiedra Alcamí y otros, 2011)

2.5.1. Clasificación

- **Información textual:** Se compone de texto escrito, como artículos de noticias, libros o documentos. Es la forma más común de información.
- **Información numérica:** Presenta datos numéricos o estadísticos, que se utiliza en aplicaciones como hojas de cálculo o bases de datos para procesar grandes volúmenes de datos.
- **Información gráfica:** Representa datos visuales de la información a través de diagramas, gráficos e imágenes. Este enfoque facilita la comprensión de conceptos abstractos y complejos mediante un lenguaje visual claro y sencillo.

(Marujita, 2023)

2.6. Sistema

Un sistema se refiere a un grupo organizado de componentes que están interrelacionados entre sí, ya sea que se trate de elementos físicos o conceptuales. Estos componentes forman una estructura y tienen una composición específica, así como un entorno particular. El término "sistema" se aplica en diferentes campos del conocimiento, como la física, la biología y la informática o computación. (Etecé, 2021)

2.7. Sistemas De Información

Un sistema de información tiene como propósito principal la gestión y administración de los datos y la información que lo conforman. La prioridad es poder recuperar siempre dichos datos y tener un acceso fácil y seguro a ellos, garantizando la integridad y confidencialidad de la información. (Peiró, 2020)

2.7.1. Características de un Sistema de Información

Un sistema de información se caracteriza principalmente por su capacidad eficiente para procesar los datos en relación a su área de aplicación. Los sistemas de información se nutren de procesos y herramientas estadísticas, de probabilidad, de inteligencia de negocios, de producción, de marketing, entre otros, con el fin de llegar a la solución óptima. También se caracteriza por su diseño, facilidad de uso, flexibilidad, mantenimiento automático de registros, apoyo en la toma de decisiones críticas y preservación del anonimato en información no relevante. (Chen, 2019)

2.7.2. Componentes de un Sistema de Información

Los componentes del sistema de información posibilitan una serie de procedimientos que implican:

- **Entrada de los datos:** Son los datos que se recopilan y registran en el sistema.
- **Gestión y procesamiento de datos:** Es la transformación de los datos en información útil.

- **Almacenamiento:** Es la forma en que se guarda la información procesada para su posterior acceso y recuperación.
- **Salida:** Son los resultados del procesamiento de la información.

Los componentes del sistema de información colaboran de forma conjunta y persiguen los mismos propósitos para lograr la utilización y la adecuada gestión de cualquier información específica. (Chen, 2019)

2.8. Web

(World Wide Web, es decir, www) es una colección de documentos (páginas web) interconectados por enlaces de hipertexto, disponibles en Internet, que pueden transmitirse mediante tecnología digital. "Hipertexto" significa la combinación de todo tipo de texto, gráficos y archivos en un solo documento.

Desde su fundación en 1966, cuando apareció la primera red Arpanet, hasta el posterior nacimiento de Internet tal y como la conocemos, no ha dejado de cambiar y mejorar. Con el transcurso del tiempo se tuvo versiones de la Web 1.0 a la 2.0, 3.0 y ahora llega la Web 4.0. (Latorre, 2018)

2.8.1. Diseño Web

El diseño Web es el proceso de creación y diseño de la apariencia y estructura de un sitio web. Implica combinar elementos como diseño gráfico, tipografía, color, imágenes y diseño de contenido para crear una experiencia de usuario estética y funcional. Además del aspecto visual, el diseño web también aborda la arquitectura de la información, la usabilidad y la experiencia del usuario. Esto implica organizar y estructurar el contenido para que se pueda encontrar y navegar fácilmente. También se centra en optimizar el rendimiento del sitio web, la capacidad de respuesta en todos los dispositivos y la accesibilidad para personas con discapacidades.

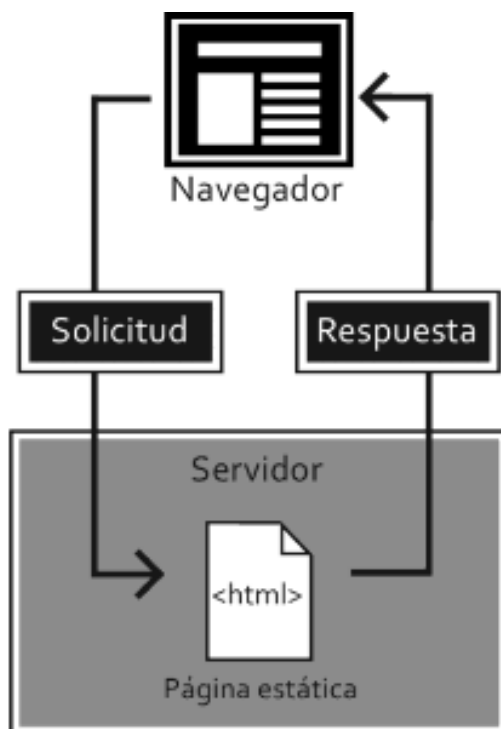
2.8.1.1 Componentes Principales del Diseño Web.

- **Responsivo:** Debe ser responsivo y adaptable a diferentes tamaños de pantalla y dispositivos. Esto garantiza que su sitio web se vea y funcione bien en computadoras de escritorio, tabletas y dispositivos móviles.
- **Contenido Legible:** Debe hacer que el contenido sea fácil de leer y comprender. Esto incluye elegir fuentes que sean fáciles de leer, un espacio adecuado entre líneas y párrafos y presentar la información con claridad.
- **Aspecto Visual y Estético:** Debe ser visualmente atractivo y reflejar la personalidad de su marca o el propósito de su sitio web. Esto incluye elegir colores, tipografía, imágenes y elementos visuales.
- **Velocidad de Carga:** Debe cargarse rápidamente para evitar perder visitantes. Esto incluye optimizar el tamaño de los archivos, utilizar técnicas de compresión y minimizar el uso de scripts y recursos innecesarios.

2.8.2. Página Web

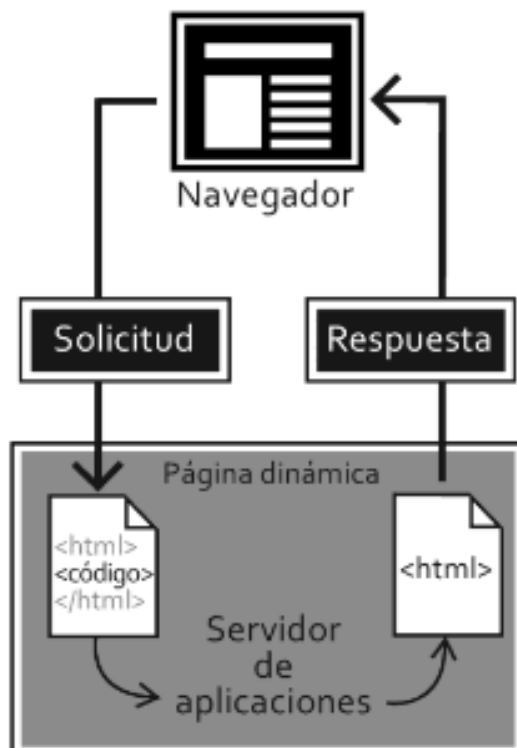
Es una fuente de información compatible con la web que se puede ver a través de un navegador web o un cliente web. Incluye muchos archivos que pueden ser imágenes, vídeos y, por supuesto, archivos de código fuente. Debe estar escrito al menos en HTML y debe indicar en su código dónde se encuentra cada elemento, texto, imagen o vídeo, es decir, la estructura de la página. Existen 2 tipos de páginas: Estáticas y Dinámicas.

2.8.2.1 Página Web Estática. Están contruidos en formato HTML, pueden representar texto estático acompañado de imágenes o archivos multimedia (audio o video) y no permiten la interacción del usuario más allá de enlaces y formularios de envío. Son fáciles de crear, aunque ofrecen pocos beneficios a los visitantes. (Vértice, 2009)

Figura 4*Página Web Estática*

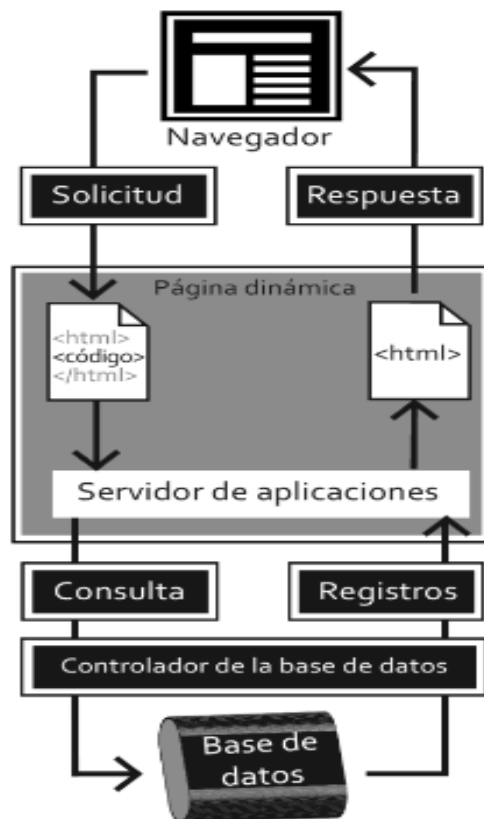
Nota. (Vértice, 2009)

2.8.2.2 Página Web Dinámica. Contienen elementos que permiten a los usuarios interactuar con ellos o crear efectos especiales. Para crearlos, además de HTML, necesitas otro tipo de lenguajes de programación. Pueden ejecutarse en el cliente o en el servidor. Cuando una página HTML contiene un script del lado del cliente, el navegador es responsable de interpretarlo y ejecutarlo. Que se implementen de una forma u otra dependerá de cómo se utilizará el sitio. Las páginas dinámicas del servidor son ejecutadas por el servidor. El archivo HTML del sitio web llega al servidor y ejecuta los scripts contenidos en él, devolviendo únicamente el código HTML al cliente. (Vértice, 2009)

Figura 5*Página Web Dinámica*

Nota. (Vértice, 2009)

2.8.2.3 Página Web con Base de Datos. Utiliza una estructura de almacenamiento de información organizada en una base de datos. Esto permite la recopilación, gestión y acceso eficiente a datos como datos de usuario, datos de productos e información específica para que los usuarios puedan interactuar dinámicamente con esos datos a través del sitio web. Esta combinación de interfaz web y base de datos permite la visualización de información actualizada y personalizada en tiempo real. (Vértice, 2009)

Figura 6*Página Web con Base de Datos*

Nota. (Vértice, 2009)

2.8.3. Aplicaciones Web

Una aplicación web es un software cliente-servidor que le permite realizar ciertas funciones en el Internet, como enviar mensajes, realizar compras, editar fotografías, jugar videojuegos, realizar pagos entre otras actividades.

2.8.3.1 Características de las Aplicaciones Web.

- **Disponibilidad y Compatibilidad:** Se puede acceder a las aplicaciones web desde todos los navegadores web, dispositivos y ubicaciones.
- **Escalabilidad:** Las aplicaciones web pueden agregar usuarios según sea necesario sin necesidad de infraestructura adicional o hardware costoso.

- **Almacenamiento Eficiente:** La mayoría de los datos de las aplicaciones web se almacenan en la nube, por lo que las empresas no tienen que invertir en almacenamiento adicional para ejecutarlas. Por su parte, los usuarios no necesitan descargar aplicaciones y no necesitan espacio en el disco duro para utilizarlas.

(Londoño, 2023)

2.8.3.2 Aplicaciones Web Nativas. Es una aplicación de desarrollo especial para operar en el navegador. A diferencia de las aplicaciones progresivas, las aplicaciones web nativas no pueden funcionar sin conectarse a Internet o enviar notificaciones push. Se crean utilizando tecnologías web estándar como HTML, CSS y JavaScript. Aunque estas aplicaciones se ejecutan en un navegador web, pueden proporcionar una experiencia de usuario similar a la de una aplicación móvil nativa. Esto se logra utilizando técnicas de diseño y programación que aprovechan las capacidades del navegador y garantizan una experiencia fluida y receptiva.

2.8.3.3 Aplicaciones Web Progresivas. Conocido por sus siglas (Progressive web app), están diseñadas para funcionar en todos los dispositivos y plataformas, ya sea de escritorio, portátil, tableta o móvil. Esto se logra mediante el uso de tecnologías web estándar como HTML, CSS y JavaScript, que permiten que la aplicación se ajuste y adapte automáticamente al tamaño de la pantalla y las capacidades del dispositivo.

2.8.4. Servidor Web

Un servidor web es un programa o software que se ejecuta en un servidor y que le permite recibir y responder a las solicitudes de sitios web de los clientes. Su función principal es almacenar y entregar los archivos y recursos necesarios para ver un sitio web en un navegador. Cuando un usuario ingresa la dirección de un sitio web en su navegador, envía una solicitud al servidor web apropiado.

El servidor web procesa la solicitud y devuelve HTML, CSS, JavaScript, imágenes y otros recursos necesarios para mostrar correctamente la página web en el navegador. Los

servidores web utilizan protocolos de comunicación como HTTP (Protocolo de transferencia de hipertexto) o HTTPS (Protocolo seguro de transferencia de hipertexto) para conectarse a los clientes y transmitir datos de forma segura.

2.8.4.1 Tipos de Servidores Webs más Utilizados. En la actualidad los servidores más utilizados son:

HTTP Apache

- Apache es uno de los servidores web más populares y ampliamente utilizados.
- Es de código abierto y gratuito.
- Es compatible con múltiples plataformas, como Unix, Linux, Windows y macOS.
- Tiene una arquitectura modular y flexible, lo que permite la personalización y la adición de módulos según las necesidades.
- Es altamente escalable y puede manejar grandes volúmenes de tráfico.
- Ofrece una amplia gama de características y funcionalidades, como soporte para SSL/TLS, autenticación de usuarios, compresión de datos y balanceo de carga.
- Es altamente configurable y permite ajustar diversos parámetros para optimizar el rendimiento y la seguridad.

2.8.5. Seguridad Web

La seguridad web se refiere a las medidas y prácticas implementadas para proteger los sitios web y las aplicaciones en línea contra amenazas y ataques maliciosos. Su objetivo principal es garantizar la confidencialidad, integridad y disponibilidad de los datos y recursos almacenados en un sitio web.

La seguridad web es un aspecto crucial en la actualidad, ya que los ataques cibernéticos son cada vez más sofisticados. Por lo tanto, es necesario implementar medidas de seguridad sólidas y mantenerse actualizado sobre las últimas amenazas y soluciones de seguridad para proteger adecuadamente los sitios web y las aplicaciones en línea.

2.8.5.1 Técnicas y Herramientas de la Seguridad Web.

- **Autenticación y control de acceso:** se utilizan métodos de autenticación seguros, como contraseñas seguras o autenticación de dos factores, para garantizar que solo los usuarios autorizados puedan acceder al sitio web.
- **Cifrado de datos:** el cifrado se utiliza para proteger la información confidencial durante la transmisión evitando que terceros intercepten y lean los datos.
- **Actualizaciones y corrección de errores:** Es importante mantener actualizados los sistemas y aplicaciones utilizados en el sitio con los últimos parches de seguridad para abordar posibles vulnerabilidades.
- **Copia de seguridad y restauración de datos:** Haga una copia de seguridad de los datos de su sitio web con regularidad y tenga implementado un plan de recuperación de datos para garantizar la continuidad del servicio en caso de pérdida o falla de datos.

2.9. Gestión

La gestión es el proceso de planificar, organizar, dirigir y controlar los recursos de una organización de manera efectiva y eficiente para lograr sus objetivos. Esto implica la toma de decisiones y la asignación óptima de recursos, como el capital humano, financiero, físico y tecnológico.

2.9.1. Elementos de la Gestión

Los principales elementos de la gestión son:

- **Planificación:** Establezca objetivos claros y objetivos a largo plazo, teniendo en cuenta las fortalezas de la organización.
- **Organización:** Discutir las tareas necesarias para lograr objetivos y distribuir completamente las tareas entre los miembros de la organización.
- **Dirección:** Dirige los esfuerzos de los empleados para lograr las metas planificadas a través de la motivación, la influencia y la comunicación efectiva.
- **Monitoreo:** monitorear el desempeño organizacional frente a objetivos previamente establecidos y tomar acciones correctivas si es necesario.

2.10. Gestión De Ventas

Es el proceso de desarrollar la fuerza de ventas mediante la coordinación de las estrategias y tácticas de ventas más efectivas para lograr los objetivos comerciales. Aunque su empresa esté generando ingresos hoy en día, siempre hay una mejor manera de garantizar operaciones comerciales fluidas y aumentar la productividad. Además, una buena gestión de ventas creará un equipo de ventas más comprometido y centrado en lograr resultados. Lo mejor de la gestión de ventas es que no es necesario ser una empresa de talla mundial. Las pequeñas y medianas empresas pueden beneficiarse de este nivel de trabajo. Incluso animamos a las empresas a dominar la gestión de ventas para lograr un crecimiento sostenible, independientemente de su tamaño o sector. (Palacios, 2023)

La gestión de ventas es el proceso de coordinar un grupo de personas y recursos para lograr los objetivos de ventas de una empresa. Contiene un conjunto de políticas y procedimientos que brindan orientación para una amplia gama de actividades. El departamento de ventas es el responsable de gestionar las ventas. Una adecuada gestión de ventas debe identificar nuevas oportunidades, realizar previsiones de ventas y gestionar informes y métodos. Todas estas tareas ayudarán a los participantes a alcanzar sus objetivos. Sin embargo, la gestión de ventas debe incluir una serie de procesos y actividades. Si se hace de forma eficaz, ayudará a la empresa a crecer y obtener más beneficios. (Quiroa, Gestión de ventas, 2021)

La gestión de ventas es la actividad comercial de intercambiar bienes o servicios por dinero u otros artículos de valor. Generalmente, es un proceso o serie de pasos lógicos y ordenados que se siguen para lograr el objetivo final de ambas partes (comprador y vendedor).

2.10.1. Tipos de Gestión de Ventas. Los tipos de ventas más destacados son los siguientes:

- **Ventas por Menor:** Se refiere a la venta de productos directamente al consumidor final, generalmente en establecimientos físicos como tiendas, supermercados o boutiques.

- **Ventas por Mayor:** consiste en vender productos al por mayor a intermediarios, minoristas u otros comercios. Este tipo de ventas se caracteriza por ofrecer precios más bajos al adquirir cantidades mayores.
- **Ventas Directa:** consiste en la venta personalizada por parte del vendedor directamente al comprador, sin intermediarios. Ejemplos de este tipo de ventas incluyen demostraciones de productos en el hogar o ventas de productos en el hogar.

2.11. Gestión De Compras

Es la práctica de una empresa que compra bienes y servicios para respaldar su proceso de producción. Cabe señalar que la gestión de compras es un proceso bastante complejo para las empresas. Antes de realizar un pedido a un proveedor, es necesario investigar el mercado para encontrar el mejor proveedor. Para hacer esto, necesita conocer el estado de calidad y precio de los bienes y servicios que desea comprar. Luego, se implementa el proceso de negociación y los procedimientos adecuados de gestión de compras. Gestionar los suministros de su empresa es fundamental para lograr resultados financieros exitosos. (Quiroa, Gestión de compras, 2021)

Se denomina gestión de compras a las actividades relacionadas con la compra de bienes y servicios, cuya finalidad es realizar un proceso productivo, comercial o administrativo. Es un proceso de examen de las condiciones del mercado, los proveedores y los requisitos internos de una empresa para producir un producto o servicio de forma eficaz. Es un proceso complejo que hoy en día se encuentra mayoritariamente automatizado y sus etapas son más o menos estables, porque es una práctica que toda empresa necesariamente aplica para brindar los insumos que permitan a la empresa realizar sus actividades. (DELSOL, 2023)

En el ámbito personal, las compras se relacionan con la adquisición de bienes de consumo, como alimentos, ropa, electrónicos, entre otros. Este tipo de compras pueden realizarse en tiendas físicas, en línea o a través de otros canales de venta.

2.11.1. Clasificación de la Gestión de Compras.

A continuación, se presentan los tipos de pedidos más comunes:

- **Compras de consumo:** Estas compras están relacionadas con la adquisición de bienes o servicios para uso personal. Incluyen productos como alimentos, ropa, electrónicos, muebles, entre otros.
- **Compras corporativas:** Este tipo de compras se llevan a cabo por empresas u organizaciones para adquirir los suministros, equipos, materiales o servicios necesarios para su funcionamiento. Pueden incluir desde la compra de insumos de oficina hasta la adquisición de maquinaria especializada.
- **Compras por catálogo:** Este tipo de compras se realiza a través de catálogos físicos o en línea, donde se pueden seleccionar y ordenar los productos deseados. Son comunes en la compra de productos por correo o en tiendas en línea.
- **Compras a proveedores locales:** Se trata de adquirir bienes o servicios de proveedores ubicados en la misma región o país que la empresa. Esto puede tener ventajas en términos de logística, comunicación y apoyo a la economía local.

2.12. Ingeniería

La ingeniería es una disciplina que aplica conocimiento científico y matemático para planificar, diseñar, desarrollar y mantener sistemas, producción y procesos que abordan desafíos prácticos y necesidades de la sociedad. Los ingenieros utilizan principios fundamentales de diversas ramas de la ciencia y la tecnología para crear soluciones innovadoras que van desde infraestructuras físicas como puentes y edificios, hasta avances tecnológicos en áreas como la electrónica, la informática y la biotecnología.

2.13. Software

Es un conjunto de programas, datos y ordenes que controlan el funcionamiento de un sistema informático. Esta combinación de elementos permite que la computadora y dispositivos electrónicos realizar diversas tareas, desde procesar información y ejecutar

aplicaciones hasta gestionar software y facilitar la interacción entre usuarios y máquinas. El software puede ser categorizado en dos tipos principales: Software de sistema, que se encarga de la gestión y el funcionamiento del hardware, y software de aplicación, diseñado para realizar tareas específicas, como procesamiento de texto, navegación web o diseño gráfico.

2.14. Ingeniería De Software

El desarrollo de software es una tecnología de múltiples capas. Cualquier enfoque técnico debe basarse en el compromiso de la organización con la calidad. La gestión de calidad total, Six Sigma y otras filosofías similares promueven una cultura de mejora continua, y es esta cultura la que en última instancia conduce al desarrollo de enfoques cada vez más eficaces para el desarrollo de software. La naturaleza del desarrollo de software es prestar atención a la calidad. (Pressman R. S., 2010)

El desarrollo de software es una tecnología de múltiples capas. Cualquier enfoque técnico debe basarse en el compromiso de la organización con la calidad. La naturaleza del desarrollo de software es prestar atención a la calidad.

Figura 7

Capas de la ingeniería de software



Nota. (Pressman R. S., 2010)

2.14.1. Ciclo de Vida del Desarrollo de Software SDLC.

Presenta un enfoque organizado que dirige el proceso completo de construcción de software. Este proceso se divide en una serie de etapas, cada una desempeñando un papel crucial en la gestión y éxito de proyectos de desarrollo de software.

2.14.1.1 Fase de Requisitos. Esto implica una comunicación cercana con las partes interesadas, como clientes y usuarios, para comprender y registrar con precisión sus necesidades y expectativas. La claridad en los requisitos es una piedra angular para el éxito del proyecto.

2.14.1.2 Fase de Diseño. Esto incluye la definición de la arquitectura del sistema, la estructura de datos y la apariencia de la interfaz de usuario. El objetivo es establecer una base sólida que oriente la fase de implementación.

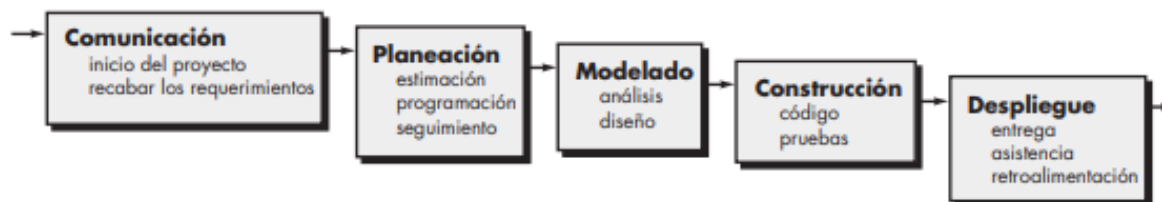
2.14.1.3 Fase de Implementación. En la fase de implementación, se procede con la codificación del software de acuerdo con las especificaciones y diseño previamente establecidos. Aquí es donde se escribe y se prueba el código del programa para asegurar su eficacia y cumplimiento de los requisitos.

2.14.1.4 Fase de Pruebas. Estas pruebas pueden abarcar desde pruebas de unidades, que examinan componentes individuales, hasta pruebas de sistema, que verifican el funcionamiento global del software.

2.14.1.5 Fase de Despliegue. Una vez superadas las pruebas y confirmado que el software satisface los requisitos, se avanza hacia la fase de despliegue. En esta etapa, el software se instala en el entorno de producción y se pone a disposición de los usuarios finales.

Figura 8

Ciclo de vida clásico



Nota. (Pressman R. S., 2010)

2.15. UML (Lenguaje Unificado De Modelado)

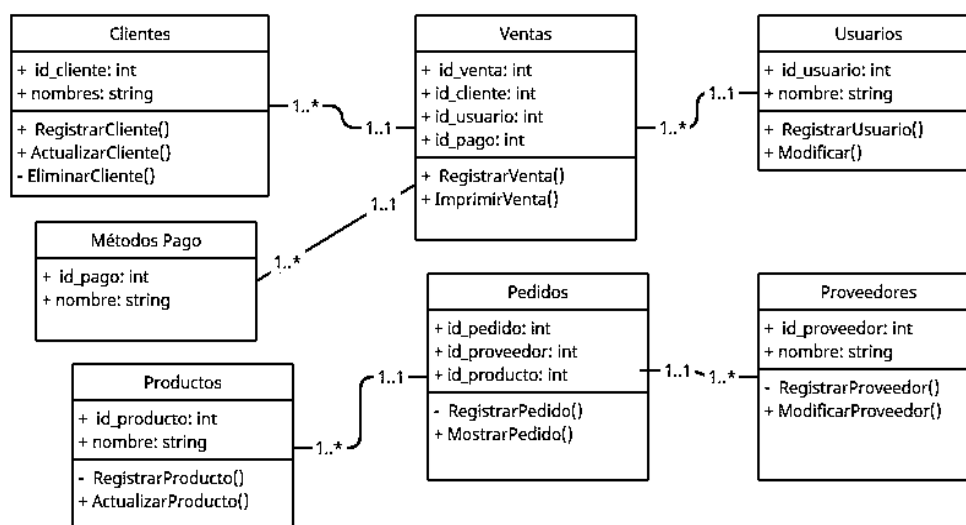
Este es un conjunto de 13 tipos diferentes de diagramas para modelar sistemas de software. Surgió de una investigación sobre modelado orientado a objetos en la década de 1990, cuando se integraron anotaciones orientadas a objetos similares para crear el lenguaje UML. La versión completa (UML 2) se completó en 2004. UML es ampliamente aceptado como enfoque estándar para crear modelos de sistemas de software. Se proponen opciones de modelado de sistemas más generales. (Sommerville, 2011)

2.15.1. Diagramas UML

2.15.1.1 Diagramas de Clase. En este diagrama, cada clase se representa como un rectángulo con tres secciones: la primera sección muestra el nombre de la clase, la segunda sección muestra los atributos de la clase y la tercera sección muestra los métodos de la clase.

Figura 9

Diagrama de Clase



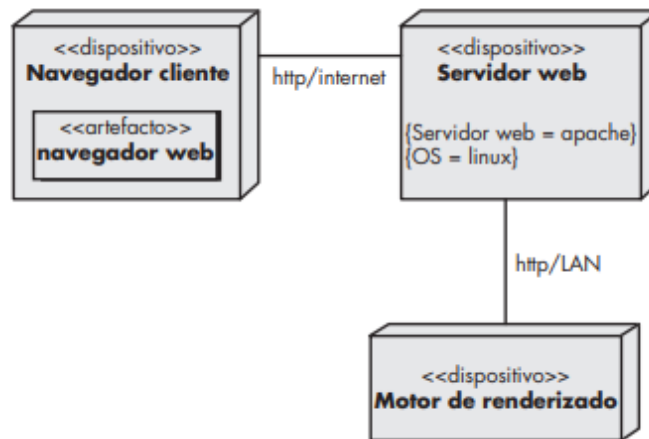
Nota. Elaboración Propia

2.15.1.2 Diagrama de Implementación. Es utilizada para representar la implementación de un sistema en hardware o software. Los componentes físicos se

representan mediante nodos, que pueden ser computadoras, etc. Los componentes lógicos están representados por bloques, que pueden incluir: bases de datos, etc. Las relaciones entre los componentes se representan mediante líneas que conectan los nodos y las cajas.

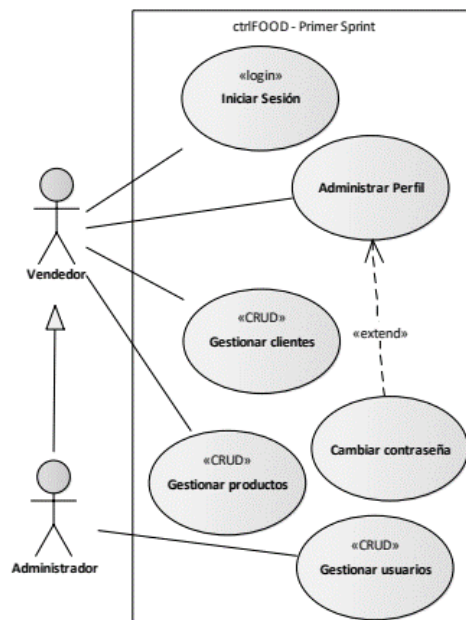
Figura 10

Diagrama de Implementación



Nota. (Pressman R. S., 2010)

2.15.1.3 Diagrama de Casos de Uso. Se utiliza para representar la interacción entre el usuario y el sistema. Los actores están representados como personas externas al sistema. Los casos de uso representan diferentes funciones y procesos del sistema y se representan por líneas.

Figura 11*Diagrama de Casos de Uso*

Nota. (Mamani, 2019)

2.16. Metodología UWE (UML-Based Web Engineering)

La metodología UWE es un método sistemático para diseñar aplicaciones web basadas en técnicas UML, notación UML y mecanismos de extensibilidad UML. UWE se caracteriza porque durante el proceso de desarrollo las fases de análisis y diseño tienen gran importancia. El método UWE es un proceso práctico para la creación de aplicaciones web, centrándose en la aplicación práctica del método UWE. (Pressman & Lowe, 2014)

UWE se basa en el principio de que las aplicaciones web deben diseñarse y desarrollarse teniendo en cuenta las necesidades y expectativas del usuario final. La atención se centra en la creación de sistemas web fáciles de usar, accesibles y eficientes que brinden una experiencia de usuario óptima.

2.16.1. Características de la metodología UWE

- El enfoque UWE coloca a los usuarios en el centro del proceso de desarrollo, garantizando que se cumplan sus requisitos y expectativas en cada etapa del proyecto.

- Promueve la participación activa de los usuarios en todas las etapas del proyecto, desde la definición de requisitos hasta la verificación final.
- UWE contrata a expertos de diversos campos, como diseñadores, desarrolladores y expertos en experiencia de usuario, para garantizar resultados integrales y de alta calidad.
- Se basa en estándares enormemente utilizados como XMI, UML, entre otros.

2.16.2. Fases de la Metodología UWE

Consta de varias fases que se deben seguir para un desarrollo web. Estas etapas incluyen:

- **Análisis y Requerimientos:** En esta fase se recopilan los requisitos funcionales y no funcionales del sistema.
- **Diseño:** Se crea un modelo conceptual para mostrar cómo los usuarios interactuarán con la aplicación y cómo se organizan y estructuran las distintas funciones y páginas.
- **Codificación:** Durante la fase de codificación, los desarrolladores se basan en los diseños y especificaciones definidos en etapas anteriores para escribir el código necesario para que la aplicación funcione correctamente.
- **Pruebas:** En esta fase, se llevan a cabo diferentes tipos de pruebas, como pruebas de unidad, pruebas de integración y pruebas de aceptación.
- **Implementación:** Se utiliza el diseño y la codificación previamente realizados para dar vida a la aplicación y convertirla en un producto funcional.

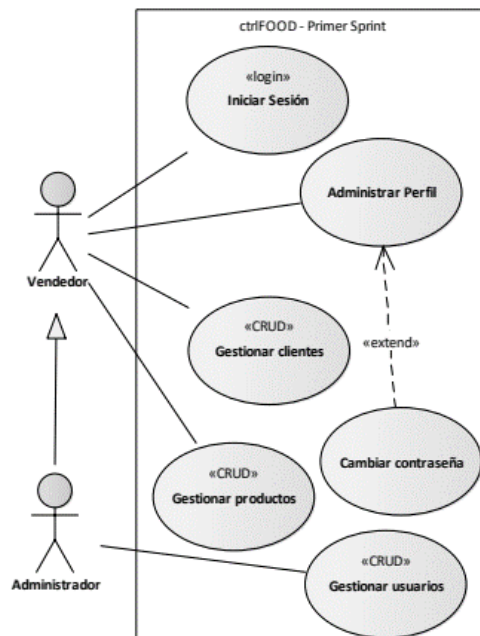
2.16.3. Modelado UWE

El proceso de modelado en UWE incluye varios tipos de modelos, como modelos de requisitos, modelos conceptuales, modelos de navegación, modelos de presentación y modelos de interacción. Estos modelos se crean de forma incremental y recursiva, lo que proporciona una vista detallada y estructurada de la aplicación web que está creando.

2.16.3.1 Análisis de Requisitos. Implica crear una representación clara y organizada de las necesidades y especificaciones de un sistema. En esta etapa, los requerimientos funcionales y no funcionales del sistema, así como las expectativas del usuario, se analizan y registran continuamente.

Figura 12

Diseño de Casos de Uso

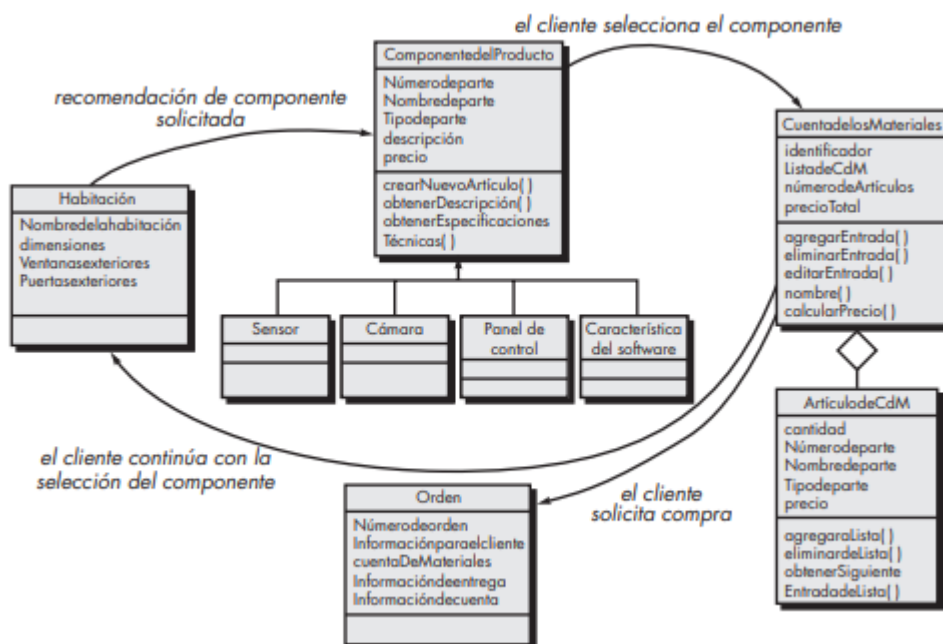


Nota. (Mamani, 2019)

2.16.3.2 Diseño Conceptual. Es el proceso de crear representaciones abstractas clara. Durante esta etapa, se desarrolla un modelo para describir las entidades básicas y cómo interactúan entre sí en el contexto del sistema. El objetivo principal es proporcionar una representación conceptual que facilite la comprensión de la estructura y el propósito del sistema antes de pasar a detalles específicos de diseño e implementación.

Figura 13

Ejemplo del Modelo Conceptual

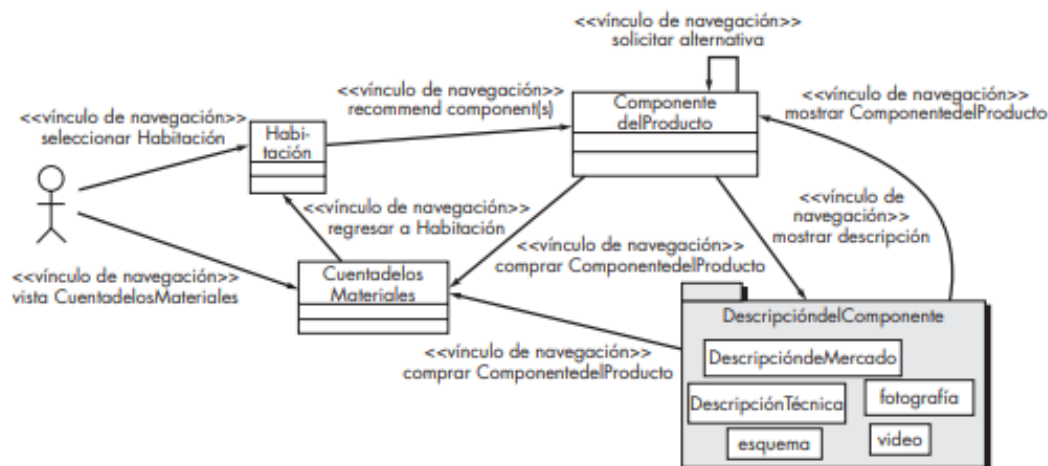


Nota. (Pressman R. S., 2010)

2.16.3.3 Diseño Navegacional. El modelado de navegación es el proceso de creación de representaciones visuales estructuradas que describen cómo los usuarios interactuarán con una aplicación o sistema. En esta etapa, se crea un mapa de navegación para mostrar la estructura y la lógica de cómo los usuarios pueden navegar a través de las páginas o vistas de la aplicación. El objetivo principal es facilitar a los usuarios la comprensión de cómo navegar por el sistema y cómo se conectarán las diferentes partes de la interfaz.

Figura 14

Ejemplo del Modelo Navegacional

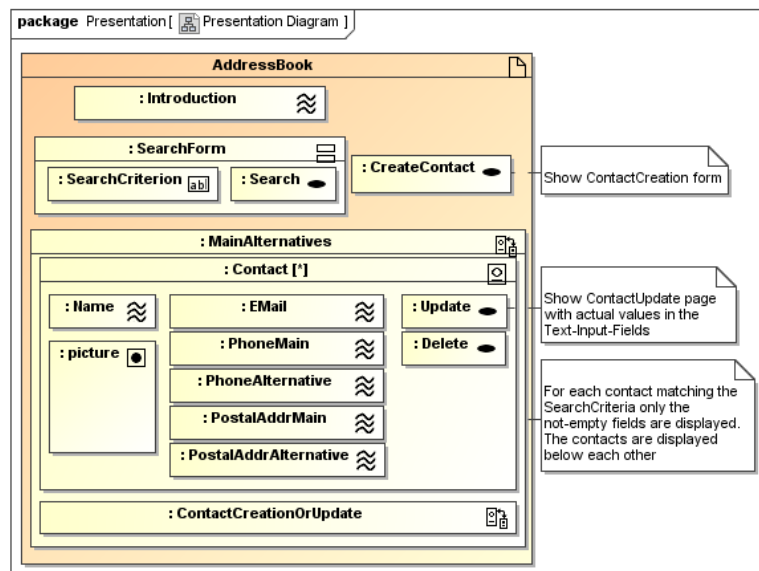


Nota. (Pressman R. S., 2010)

2.16.3.4 Diseño de Presentación. El modelado de presentación se refiere al proceso de creación de vistas estructuradas y detalladas de la interfaz de usuario de una aplicación o sistema. El objetivo principal es determinar el diseño de los elementos, la interfaz gráfica y la experiencia general del usuario en la aplicación.

Figura 15

Ejemplo de Modelado de Presentación



Nota. (LMU – Ludwig-Maximilians-Universität München, s.f.)

2.17. Métricas De Calidad De Software

2.17.1. ISO/IEC 25010

La Norma ISO/IEC 25010, es una normal que representa la calidad, que está relacionada tanto con los aspectos estáticos del software como con las características dinámicas de los sistemas informáticos. Este modelo se aplica tanto a los sistemas informáticos en general como al software en particular.

Define ocho características, las cuales son:

- **Funcionalidad:** La capacidad del software para cumplir con los requisitos funcionales y realizar las tareas esperadas de manera precisa y completa.
- **Desempeño:** La capacidad del software para realizar las funciones requeridas de manera eficiente mientras hace un uso óptimo de los recursos requeridos.
- **Compatibilidad:** La capacidad del software para interactuar y funcionar correctamente con otros entornos, sistemas o componentes se conoce como compatibilidad.
- **Usabilidad:** la facilidad con la que los usuarios pueden usar y comprender el software, incluidos aspectos como la interfaz de usuario, la navegación y la facilidad de aprendizaje
- **Confiabilidad:** La capacidad del software para mantener los niveles esperados de desempeño y evitar fallas o errores en su operación.
- **Seguridad:** Proteger la información y los datos del software del acceso no autorizado, asegurando su confidencialidad, integridad y disponibilidad.
- **Mantenibilidad:** La facilidad con la que el software se puede modificar, reparar o mejorar, asegurando la capacidad de evolucionar y adaptarse a cambios futuros.
- **Portabilidad:** La capacidad de mover o utilizar software en diferentes entornos o plataformas sin pérdida de funcionalidad o rendimiento.

(Ormeño Rojas, 2019)

2.18. Estimación De Costos

La estimación de costos nos ayuda a establecer el presupuesto del proyecto.

2.18.1. Cocomo (*Constructive Cost Model 1981*)

COCOMO es un modelo de costos objetivos para planificar y ejecutar proyectos de desarrollo de software. Propuesto por Barry Boehm en 1981 y posteriormente cambiado a COCOMO 2.0. Originalmente este modelo está compuesto por tres modelos:

- Cocomo Básico
- Cocomo Intermedio
- Cocomo Avanzado

Estos tres modelos se definen para tres tipos de proyectos:

- Modo Orgánico
- Modo Semilibre
- Modo Rígido

(UNIVERSIDAD DE SEVILLA, s.f.)

2.18.1.1 Cocomo Intermedio. Calcula el esfuerzo en función del tamaño estimado y “factores de costo”. Los factores de costo evalúan un conjunto de atributos de producto, equipo, personal y diseño.

La fórmula de este modelo es el siguiente:

$$\text{Esfuerzo} = (\text{personas} - \text{mes}) \rightarrow E = a * KLDC^b * FAE$$

FAE → Factor de Ajuste de Esfuerzo

$$\text{Tiempo (meses)} \rightarrow D = c * E^d$$

$$\text{Personas (P)} = E/D$$

$$\text{Productividad (PD)} = \frac{LCD}{\text{Esfuerzo}}$$

$$\text{Costo Mes (CM)} = \text{Personas} * \text{Salario Minimo Nacional}$$

Figura 16*Tipos de Proyecto y su Complejidad*

Tipo Proyecto	a	b	c	d
Orgánico (pequeños)	2,4	1,05	2,5	0,38
Semiacoplado (intermedios)	3,0	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Nota. (UNIVERSIDAD DE SEVILLA, s.f.)

Figura 17*Conductores de Costos FAE*

CONDUCTORES DE COSTE	VALORACIÓN					
	<i>Muy bajo</i>	<i>Bajo</i>	<i>Nominal</i>	<i>Alto</i>	<i>Muy alto</i>	<i>Extr. alto</i>
Fiabilidad requerida del software	0,75	0,88	1,00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones del tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1,00	1,07	1,15	-
Capacidad del analista	1,46	1,19	1,00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	-
Experiencia en S.O. utilizado	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1,00	0,95	-	-
Prácticas de programación modernas	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1,00	1,04	1,10	-

Nota. (UNIVERSIDAD DE SEVILLA, s.f.)

2.19. Seguridad

2.19.1. ISO/IEC 27001

Es una norma internacional que especifica los requisitos para implementar, mantener y mejorar continuamente un sistema de gestión de seguridad de la información (SGSI). Esta norma sirve para proteger la confidencialidad, integridad y disponibilidad de la información. Además, proporciona un marco de seguridad de la información que ayuda a las organizaciones a identificar y gestionar eficazmente los riesgos de seguridad de la información.

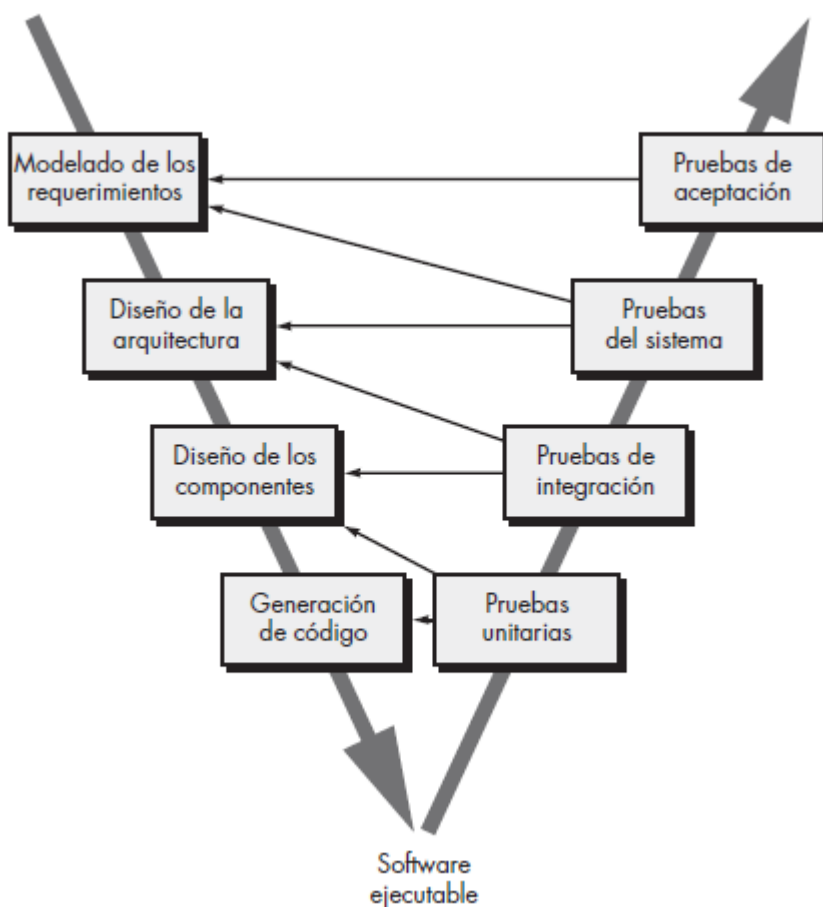
2.19.1.1 Protocolos Norma ISO 27001

- **Acceso controlado:** Limita el acceso a las fuentes de información únicamente a personas autorizadas.
- **Seguridad física:** Seguridad para proteger los activos de información física, como dispositivos de almacenamiento, oficinas, etc.
- **Seguridad lógica:** Se enfoca en salvaguardar la información y los sistemas de información a través de una serie de controles.
- **Criptografía:** Recomendable utilizar métodos de cifrado para proteger la información durante el almacenamiento y la transmisión.
- **Backup y restauración:** Planifica y realiza Backup's periódicas para garantizar que su información esté siempre disponible en caso de un desastre.
- **Seguimiento y auditoría:** Supervisa y revisa periódicamente los sistemas y registros de seguridad para detectar posibles vulnerabilidades y actividades sospechosas.

(GlobalSuite Solutions, 2023)

2.20. Modelo En V

El modelo en V es una metodología de desarrollo de software que organiza las fases de desarrollo y prueba en un patrón en forma de "V". En este modelo, cada fase de desarrollo tiene una fase de prueba asociada, lo que permite realizar pruebas de manera paralela con la construcción del software.

Figura 18*El Modelo en V*

Nota. (Pressman R. S., 2010)

El lado izquierdo de la V representa las necesidades del cliente y las etapas de implementación del sistema, mientras que el lado derecho de la V se refiere a los diferentes niveles de prueba, las cuales se detallan a continuación:

2.20.1. Pruebas Unitarias

Estos se implementan en un módulo de software particular, generalmente por programadores, por lo que también se conocen como pruebas de desarrollo. Su objetivo es asegurar que cada módulo del sistema funcione de manera autónoma como se espera, facilitando así las pruebas de integración. (Admin, 2020)

2.20.2. Pruebas de Caja Blanca

Son un tipo de pruebas de software que se realizan con conocimiento del código fuente y la estructura interna del sistema. Estas pruebas se enfocan en verificar el flujo interno, la lógica, y el control de datos dentro del código para asegurar que cada camino, condición y bucle funcione según lo esperado. El objetivo es garantizar que el código esté libre de errores y que todos los caminos posibles sean probados adecuadamente

2.20.3. Pruebas de Integración

Consisten en verificar que los distintos subsistemas de un mismo producto interactúan correctamente entre sí. Se llevan a cabo después de las pruebas unitarias para identificar posibles errores en la comunicación entre sistemas y asegurar que cumplen con las funcionalidades y requisitos establecidos por el cliente. (Admin, 2020)

2.20.3.1 PHPUnit. Es un marco de prueba PHP diseñado para desarrolladores. Este es el marco más popular para ejecutar pruebas unitarias en código PHP para probar partes específicas del código fuente. Las pruebas de PHPUnit permiten a los desarrolladores encontrar problemas lo antes posible en el ciclo de desarrollo, lo que ayuda a garantizar que no se introduzcan regresiones en el código (es decir, los cambios en una parte del código no introducen problemas). (Losoviz, 2022)

2.20.4. Pruebas de Sistema

Se llevan a cabo para comprobar el correcto funcionamiento de la integración, es decir, pruebas funcionales. Estas pruebas están diseñadas para verificar el comportamiento del producto desde la perspectiva del usuario final y su interacción con el sistema. Deben realizarse en un entorno casi realista, asegurando que todo funciona según las especificaciones y requisitos inicialmente establecidos por el cliente. (Admin, 2020)

2.20.4.1 Selenium. Es una herramienta destinada a realizar pruebas en aplicaciones web mediante scripts en diversos lenguajes de programación. Básicamente, automatiza las interacciones del navegador, lo que amplía sus posibilidades de uso.

Su propósito principal es automatizar pruebas en aplicaciones web, ya que permite automatizar tareas repetitivas en el navegador. (hiberus, 2023)

2.20.5. Pruebas de Aceptación

Se realizan cuando el área de pruebas ha completado sus pruebas programadas. Generalmente, estas pruebas las lleva a cabo el cliente o usuario final para verificar que se hayan cumplido sus expectativas. (Admin, 2020)

2.21. Pruebas De Rendimiento De Software

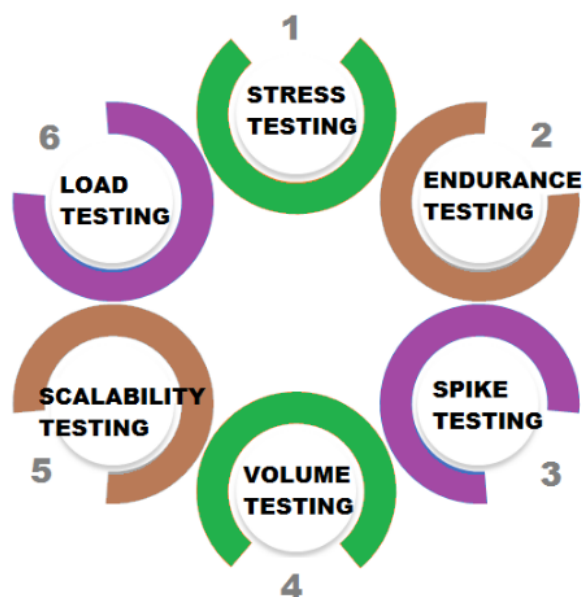
Las pruebas de rendimiento de software son un tipo de pruebas no funcionales que se realizan para evaluar cómo se comporta un sistema bajo ciertas condiciones de carga. Su objetivo es asegurar que el software cumpla con los requisitos de rendimiento esperados y pueda manejar la carga de trabajo prevista de manera eficiente y sin fallos.

2.21.1. Pruebas de Estrés

Tiene como objetivo evaluar el comportamiento del sistema bajo condiciones extremas de carga. El propósito de estas pruebas es determinar los límites del sistema y su capacidad para manejar situaciones de carga pico. Además, se busca identificar el punto de quiebre del sistema y evaluar cómo se recupera después de un fallo.

2.21.1.1 Apache Jmeter. Es un software de código abierto para medir el rendimiento de aplicaciones y sitios web. Se creó originalmente como una herramienta de prueba de carga para el servidor HTTP Apache. Sin embargo, en los últimos 20 años, se ha convertido en una valiosa e importante plataforma de prueba de rendimiento utilizada por desarrolladores.

Actualmente la herramienta de prueba de rendimiento más popular del mundo y es utilizada por empresas como Google, Microsoft y Facebook. Además, JMeter garantiza que sus sistemas funcionen de la manera más eficiente posible. Puede usarlo para cargar aplicaciones web de prueba, medir el rendimiento de aplicaciones web, probar aplicaciones web y más. (Shethi, 2023)

Figura 19*Ventajas de Apache Jmeter*

Nota. (Shethi, 2023)

2.22. Recopilación De Datos

La obtención de datos es un proceso diverso y creativo que abarca desde las conversaciones con personas hasta la exploración de datos masivos. Existen diversas maneras de recopilar datos, cada uno en su enfoque, para el presente caso se utiliza los siguientes tipos de obtención de datos:

2.22.1. Entrevistas Personales

Las entrevistas personales en profundidad permiten una interacción más detallada entre el investigador y el entrevistado. Esto es útil para explorar temas complejos o sensibles en profundidad.

2.22.2. Observaciones Directa

Observar el comportamiento de las personas o eventos en tiempo real puede proporcionar datos valiosos. Esto se utiliza comúnmente en investigaciones de campo, como la sociología o la etnografía.

2.22.3. Análisis de Documentos

Examinar documentos, informes, registros existentes es una forma de obtener datos importantes y valiosos. La precisión y la integridad en este proceso son de vital importancia para garantizar que los datos obtenidos sean fiable y útil.

2.23. Herramientas De Desarrollo

En el presente proyecto, se emplean las siguientes herramientas de desarrollo

2.24. Backend

También llamado como “lado del servidor”, constituye la parte de una aplicación web o sistema que opera detrás de escena. Se ocupa de la lógica de negocio y la interacción con la base de datos, asegurando que este ofrezca un comportamiento coherente y seguro.

2.24.1. Laravel Livewire

Es un framework del lado del servidor que utiliza la arquitectura MVC (modelo-vista-controlador) que nos sirve para desarrollar aplicaciones web simples y complejas empleando el lenguaje de programación PHP, siguiendo estrictamente el patrón arquitectónico MVC.

Las principales características de Laravel son:

- Arquitectura MVC (Modelo-Vista-Controlador)
- ORM Eloquent (Mapeo Objeto-Racional)
- Seguridad
- Migraciones de Base de Datos
- Autenticación y Autorización
- Enrutamiento
- Motor de plantillas Blade

(Eseme, 2023)

2.24.2. PHP

Es un lenguaje de programación de código abierto ampliamente interpretado que está integrado en páginas HTML y se ejecuta en un servidor. Se puede utilizar en cualquier sistema

operativo importante del mercado, incluidos: Windows, Mac OS, Linux y posiblemente algunos otros. Con PHP, tienes la libertad de elegir tu sistema operativo y servidor. (Fossati, 2018)

2.24.3. MYSQL

El software MySQL es un motor de base de datos relacional de código abierto creado en 1994 por la empresa sueca MYSQL AB. En 2008, A finales de 2010, Oracle adquirió MySQL.

Está programado en C/C++, es un motor de base de datos conectado a MySQL y se ejecuta a través de un cliente/servidor. Un servidor MySQL escucha conexiones, y un cliente (o varias clientes) se están conectando a él. Utiliza SQL como lenguaje de consulta que nos permite insertar y recuperar datos de la base de datos. (Urtiaga, 2020)

2.25. Frontend

Conocido también como “lado del cliente”, consiste en la interfaz, que incluye elementos visuales como botones, formularios, etc. Es el encargado de presentar la información de manera atractiva y funcional, permitiendo que los usuarios naveguen, interactúen y realicen acciones en la aplicación.

2.25.1. Bootstrap

Es un marco de diseño de código abierto ampliamente reconocido en el desarrollo web. Proporciona una colección de estilos y componentes predefinidos para crear interfaces modernas. Bootstrap es responsivo, es decir, que se adapta automáticamente a diferentes dispositivos, lo que facilita su uso en sitios web y en aplicaciones móviles.

2.25.2. JavaScript

Es un lenguaje de programación más utilizado en la actualidad. Junto con HTML y CSS, impulsa la mayoría de los sitios web que visitamos. Pero JavaScript ya no es exclusivo de la web: proyectos como NodeJS, Electron y React Native lo han llevado al mundo de los servidores y la programación de escritorio y aplicaciones móviles. Además, cada día aparecen

nuevos frameworks (React, Angular y otros) que amplían las capacidades de JavaScript. (F. Luna, 2019)

2.25.3. *Html5*

HTML es un lenguaje de marcado para crear sitios web. Después de 1995, se publicó el primer estándar HTML oficial, en consecuencia, lo llaman HTML 2.0. En 1997, el W3C intervino y desarrolló tres estándares más hasta llegar a donde estamos hoy lo conocemos como HTML5. Aunque HTML es un lenguaje compuesto por unidades que ayudan a estructurar y proporcionar significado para diferentes partes del documento, cada uno de estos objetos sigue una regla llamados elementos o etiquetas, que contienen una o más propiedades. (Casado, 2023)

Figura 20

Estructura de un documento HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!--Contenido de la página-->
</body>
</html>
```

Nota. Elaboración Propia

2.25.4. *Css3*

(Hojas de Estilo en Cascada) es un lenguaje de programación muy similar a HTML que le permite aplicar estilos a diferentes elementos de las páginas web para que los encabezados, las listas y los párrafos se vean iguales en todas las páginas. La forma más adecuada de aplicar estilos CSS a una página HTML es utilizar un archivo CSS externo con la extensión “.css”. (A. C. Luna, 2019)

2.25.5. jQuery

jQuery es una biblioteca JavaScript con múltiples funciones. Se utiliza para simplificar la manipulación de documentos HTML, el manejo de eventos, el manejo de animaciones y la interacción con el servidor. jQuery nos permite seleccionar elementos HTML, aplicar estilos, realizar animaciones y realizar llamadas AJAX de manera fácil y eficiente.

2.25.6. Librería SweetAlert

Es una librería de JavaScript de código abierto para crear ventanas modales, personalizadas y atractivas. Esta librería se destaca por su facilidad de uso y su capacidad para mostrar mensajes interactivos y diálogos, mejorando la experiencia del usuario en aplicaciones web.

2.26. Gestión De Código Y Visualización

2.26.1. Xampp

Es una distribución de Apache que incluye varios tipos de software gratuito. Está desarrollado por Amigos de Apache. El programa se distribuye bajo la licencia GNU. Este nombre es una abreviatura creada a partir de las abreviaturas de los programas correspondientes: Apache, MYSQL, PHP, Perl. La primera X representa los sistemas operativos Linux, Windows y Mac OS X. (Bou, 2019)

2.26.2. Visual Studio Code (VS Code)

Es un editor de código fuente desarrollado por Microsoft. Este es un software gratuito multiplataforma para Windows, GNU/Linux y macOS. VS Code se integra bien con Git, admite la depuración de código y tiene muchas extensiones que le brindan la capacidad de escribir y ejecutar código en cualquier lenguaje de programación. (Flores, 2022)

2.26.3. Navegador Web

Es un software que permite a los usuarios acceder y ver sitios web en Internet. El navegador actúa como intermediario entre el usuario y el servidor web, interpretando el código HTML, CSS y JavaScript de las páginas web para mostrar el contenido con precisión en la

pantalla del dispositivo del usuario. Además, los navegadores suelen ofrecer funciones adicionales, como la capacidad de guardar marcadores, administrar el historial de navegación, bloquear anuncios y habilitar extensiones para personalizar su experiencia de navegación.

CAPÍTULO III

CAPÍTULO III: MARCO APLICATIVO

3.1. Introducción

En el capítulo anterior, se ha explorado las bases teóricas y conceptuales necesarias para comprender a fondo el proyecto “Sistema de Información Web para la Gestión de Ventas y Compras”. Ahora en este capítulo (III), se sumergen en la aplicación práctica de los principios y enfoques previamente discutidos, llevando a cabo un análisis detallado de cómo esta solución puede transformar y optimizar el proceso de ventas y compras en el entorno institucional. A través de casos de uso y escenarios de implementación, se explora cómo el sistema web se convierte en una herramienta esencial para mejorar la eficiencia, la precisión y la toma de decisiones en la gestión de transacciones comerciales.

3.2. Análisis De La Situación Actual

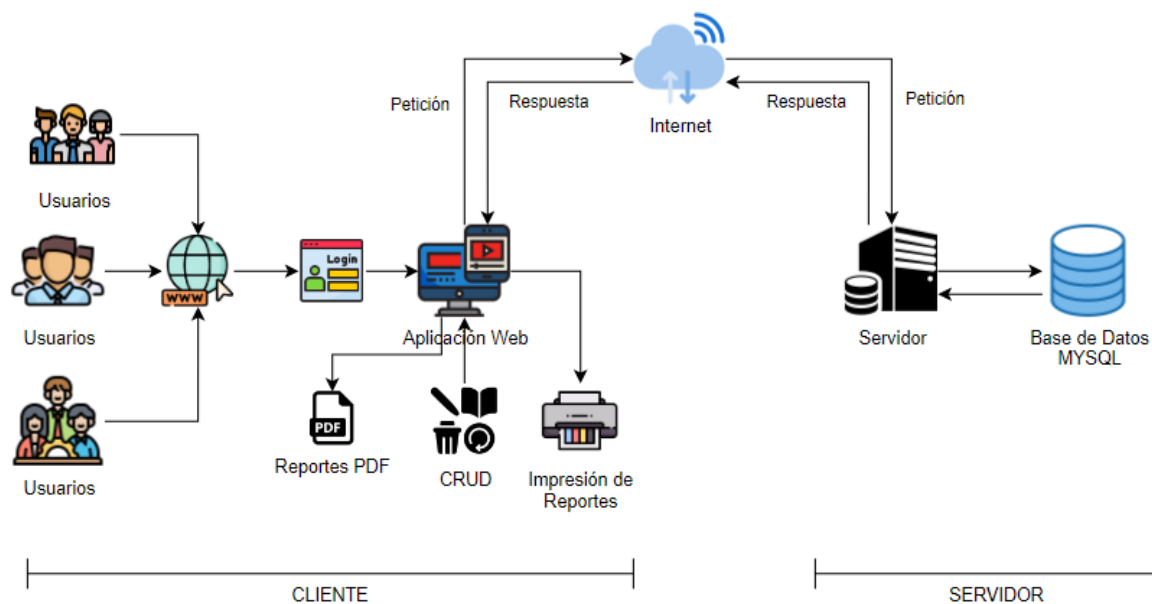
VISIÓNTecbol se destaca como una institución integral en el ámbito de la seguridad electrónica. Su amplio catálogo de servicios incluye la instalación de cámaras de seguridad, sistemas de alarmas, y una variedad de soluciones de seguridad adicionales. Estas ofertas permiten a la institución VISIÓNTECBOL abordar de manera efectiva las necesidades de seguridad de sus clientes en diversos entornos.

Además de sus servicios, la institución VISIÓNTECBOL desempeña un papel fundamental como comerciante de dispositivos electrónicos de seguridad. La institución adquiere cuidadosamente productos de seguridad de alta calidad de sus proveedores de preferencia. Estos dispositivos incluyen desde cámaras de vigilancia hasta sistemas de control de acceso y alarmas avanzadas. La capacidad de seleccionar productos de calidad de sus proveedores contribuye significativamente a la oferta de VISIÓNTECBOL y a su reputación en el mercado.

3.3. Esquema Del Sistema Web

Figura 21

Esquema del Sistema Web



3.4. Análisis de los Requerimientos

En este punto, se detallan los requerimientos funcionales y no funcionales claves que definen el comportamiento y que funciones debe cumplir el sistema para satisfacer las necesidades del usuario.

3.4.1. Obtención de los Requerimientos

3.4.2. Requerimientos Funcionales

En la siguiente tabla que se verá a continuación, se detallan los requerimientos funcionales para el sistema:

Tabla 1*Definición de Actores*

Nro. REQ	FUNCIÓN	DESCRIPCIÓN	EVIDENTE
RF-01	Gestionar Usuarios	Permite al administrador registrar, modificar y eliminar usuarios.	Evidente
RF-02	Iniciar Sesión	Permite a los usuarios iniciar sesión en el sistema.	Evidente
RF-03	Gestionar Productos	Permite registrar, modificar y eliminar productos al sistema.	Evidente
RF-04	Gestionar Ventas	Permite a los usuarios registrar ventas y generar recibos.	Evidente
RF-05	Gestionar Compras	Permite a los usuarios registrar compras.	Evidente
RF-06	Generar Reportes de Ventas	Genera informes detallados de ventas en un período específico.	Evidente
RF-07	Gestionar Clientes	Permite registrar, editar y eliminar información de los clientes.	Evidente
RF-08	Gestionar Proveedores	Permite registrar, editar y eliminar información de los proveedores.	Evidente
RF-09	Configurar Perfil	Permite al usuario personalizar su perfil.	Evidente
RF-10	Cerrar Sesión	Permite al usuario cerrar su sesión actual y regresar a la página de inicio de sesión.	Evidente

3.4.3. Requerimientos No Funcionales

En tabla que se verá a continuación, se describen los requerimientos no funcionales para el sistema:

Tabla 2*Planilla de Uso de Caso - Gestionar Usuarios*




Nro. REQ	FUNCIÓN	EVIDENTE
RNF-01	El sistema debe ser compatible con los navegadores más utilizados: Google Chrome, Mozilla Firefox, Microsoft Edge, Brave, Opera y Safari.	Evidente
RNF-02	El sistema debe cargar páginas en un tiempo máximo de 5 segundos.	Evidente
RNF-03	El sistema debe contar con información cifrada de las contraseñas de los usuarios.	Evidente
RNF-04	La interfaz de usuario debe ser intuitiva y fácil de navegar para los usuarios.	Evidente
RNF-05	El sistema debe contar con autenticación para garantizar el acceso a la información.	Evidente

3.5. Diseño

3.5.1. Modelado UWE

3.5.1.1 Definición de Actores. Los "actores" en los casos de uso son elementos fundamentales en el análisis de sistemas y desarrollo de software. Estos actores representan las diferentes entidades, ya sean usuarios o sistemas externos, que interactúan con el sistema que está siendo analizado o diseñado. Para el proyecto los actores que se definen son los siguientes:

Tabla 3*Definición de Actores*

TIPO DE ACTOR	DETALLE
Administrador 	Es el usuario con amplios privilegios y responsabilidades en la gestión y supervisión del sistema. Su función principal es garantizar el correcto funcionamiento, la seguridad y el mantenimiento de la plataforma.
Supervisor de Ventas 	Es responsable de supervisar y dirigir las actividades de venta, asegurando que se cumplan los objetivos y se brinde un servicio de calidad a los clientes.
Cliente 	Es el usuario final o consumidor de un producto o servicio proporcionado por el sistema.

3.5.1.2 Definición de Casos de Uso del Sistema.

Figura 22

Diagrama de Casos de Uso General del Sistema

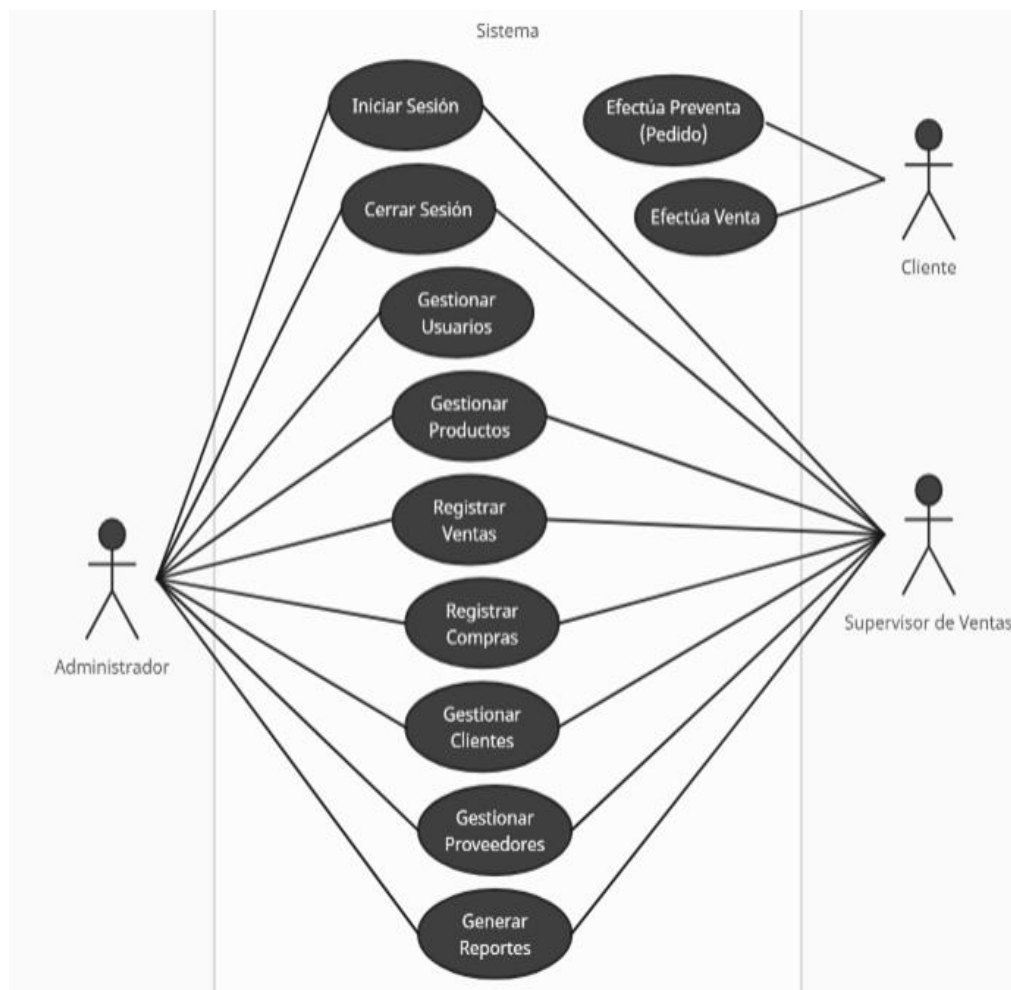
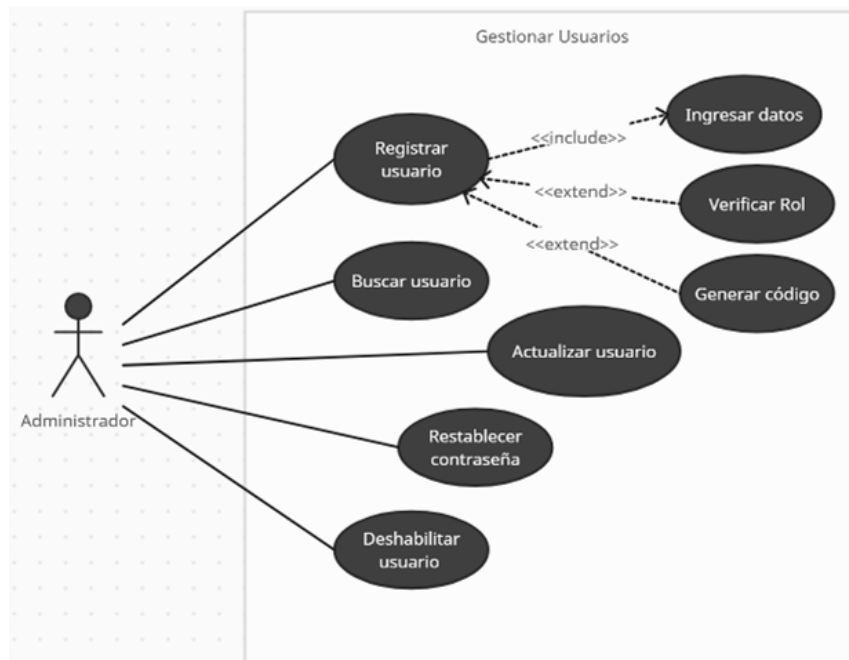
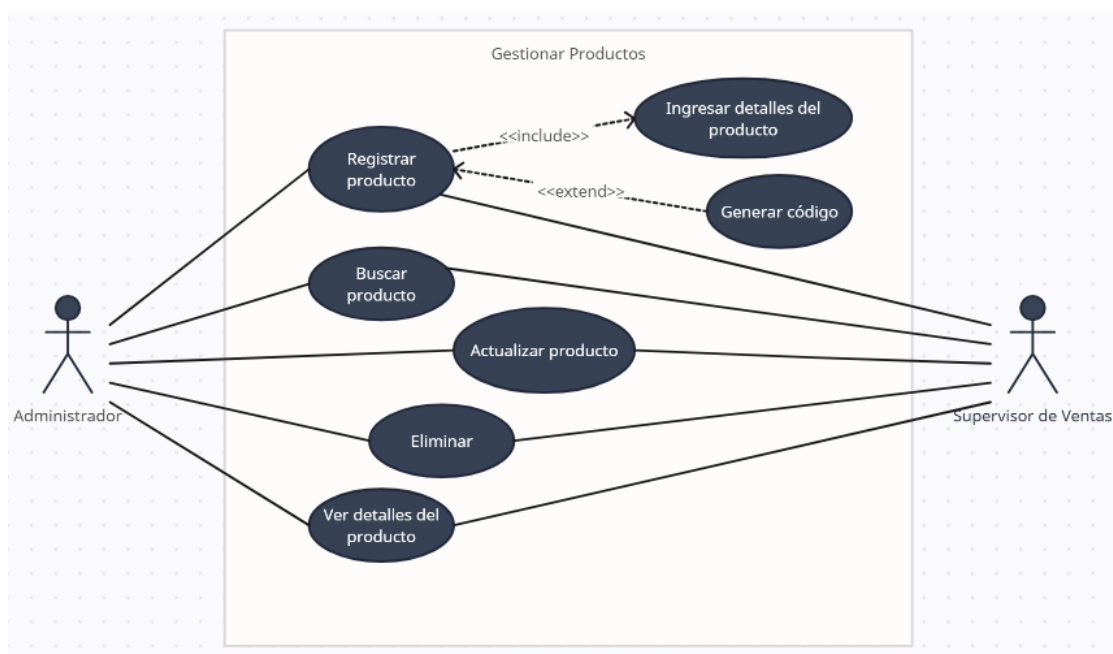
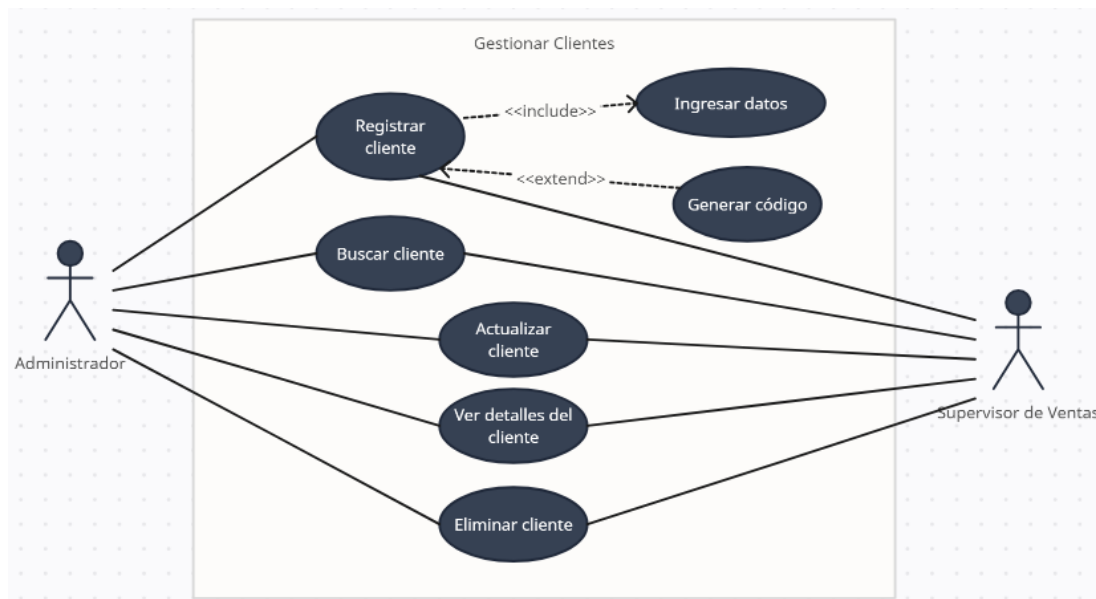


Figura 23*Caso de Uso Gestionar Usuarios***Tabla 4***Plantilla de Caso de Uso Gestionar Usuarios*

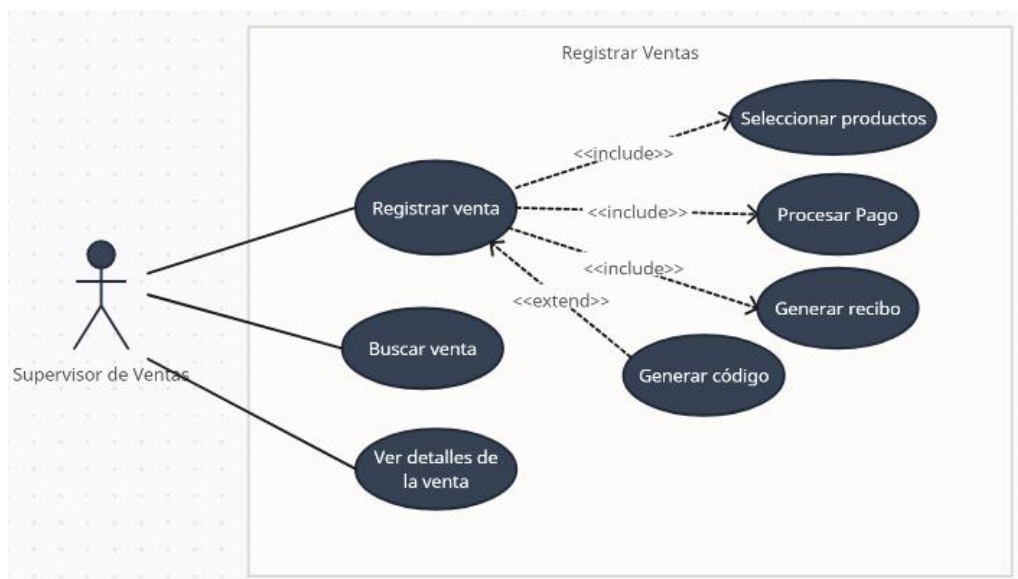
Caso de Uso	Gestionar Usuarios
Actores:	Administrador
Propósito:	Permite a los administradores del sistema gestionar las cuentas de los usuarios en el sistema. Concede a los administradores realizar acciones relacionadas con la gestión de cuentas de usuario, como crear, editar, deshabilitar, restablecer contraseñas, asignar roles y visualizar la lista de usuarios registrados.
Resumen:	registrados.
Tipo:	Primario Esencial

Figura 24*Caso de Uso Gestionar Productos***Tabla 5***Plantilla de Caso de Uso Gestionar Productos*

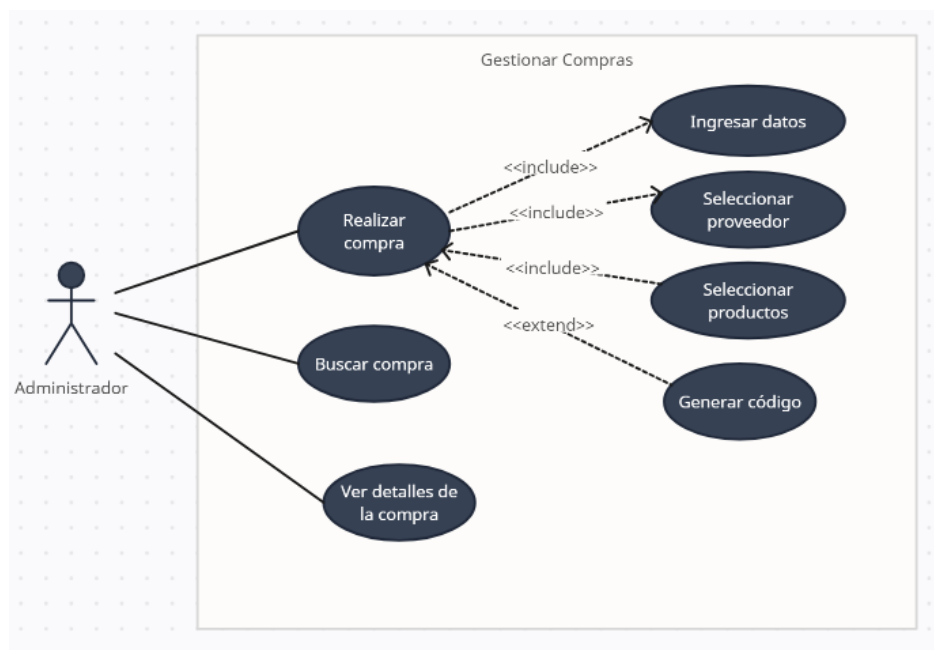
Caso de Uso	Gestionar Productos
Actores:	Administrador y Supervisor de Ventas
Propósito:	Tiene como propósito permitir a los actores gestionar la información de productos en el sistema. El administrador y el supervisor de ventas pueden realizar acciones relacionadas con la gestión de productos, que incluyen la creación, modificación y eliminación de registros de productos, así como la actualización de sus detalles y categorización.
Resumen:	actualización de sus detalles y categorización.
Tipo:	Primario (Supervisor de Ventas) Secundario (Administrador) Esencial

Figura 25*Caso de Uso Gestionar Clientes***Tabla 6***Plantilla de Caso de Uso Gestionar Clientes*

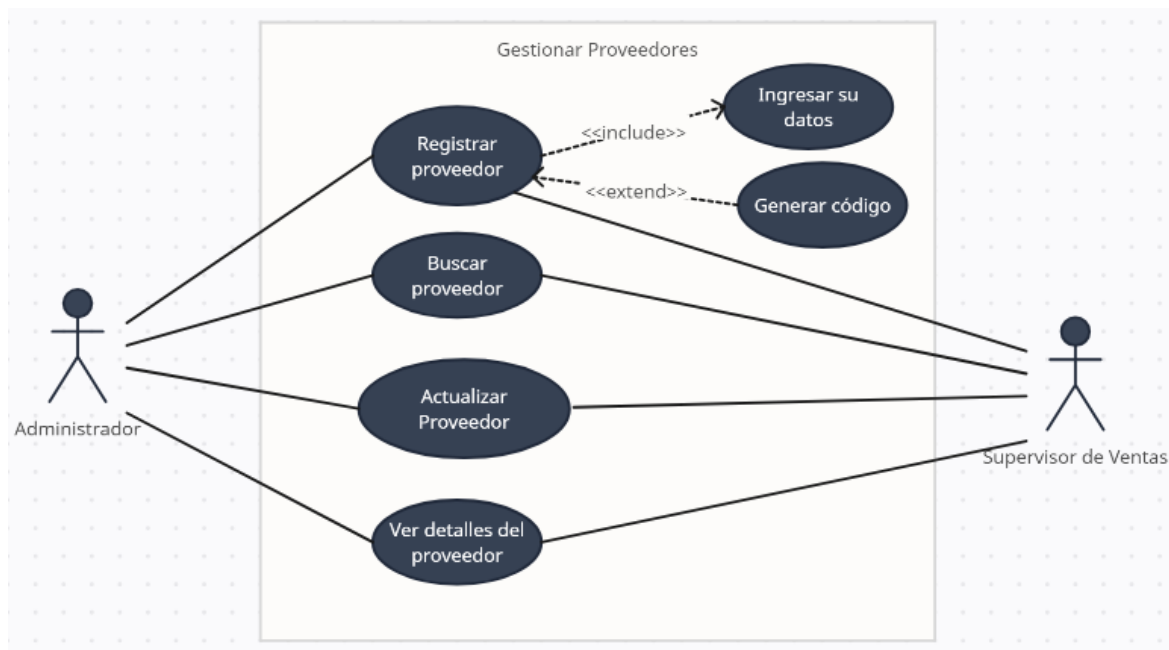
Caso de Uso	Gestionar Clientes
Actores:	Administrador y Supervisor de Ventas
Propósito:	Tiene como propósito permitir a los actores gestionar la información de clientes en el sistema. El administrador y supervisor de ventas pueden llevar a cabo acciones relacionadas con la gestión de clientes. Esto incluye la creación, modificación y eliminación de registros de clientes, así como la actualización de sus datos personales, preferencias y necesidades.
Resumen:	
Tipo:	Primario (Supervisor de Ventas) Secundario (Administrador) Esencial

Figura 26*Casos de Uso Gestionar Venta***Tabla 7***Plantilla de Caso de Uso Registrar Ventas*

Caso de Uso	Registrar Ventas
Actores:	Supervisor de Ventas
Propósito:	Este caso de uso tiene como propósito permitir a los vendedores registrar ventas en el sistema. Los supervisores de venta pueden llevar a cabo la acción principal de registrar ventas. Esto incluye la selección de productos vendidos, la asignación de clientes correspondientes, el registro de cantidades y detalles de transacción, y la generación de recibos de venta.
Resumen:	
Tipo:	Primario Esencial

Figura 27**Caso de Uso Registrar Compras****Tabla 8****Plantilla de Caso de Uso Registrar Compras**

Caso de Uso	Registrar Compras
Actores:	Administrador
Propósito:	Tiene como propósito permitir al administrador registrar compras de productos en el sistema. En este caso de uso, los administradores pueden llevar a cabo la acción principal de registrar compras. Esto incluye la selección de productos adquiridos, la asignación de proveedores correspondientes, el registro de cantidades y detalles de transacción, y la generación de registros de
Resumen:	compra.
Tipo:	Primario Esencial

Figura 28*Caso de Uso Gestionar Proveedores***Tabla 9***Plantilla de Caso de Uso Gestionar Proveedores*

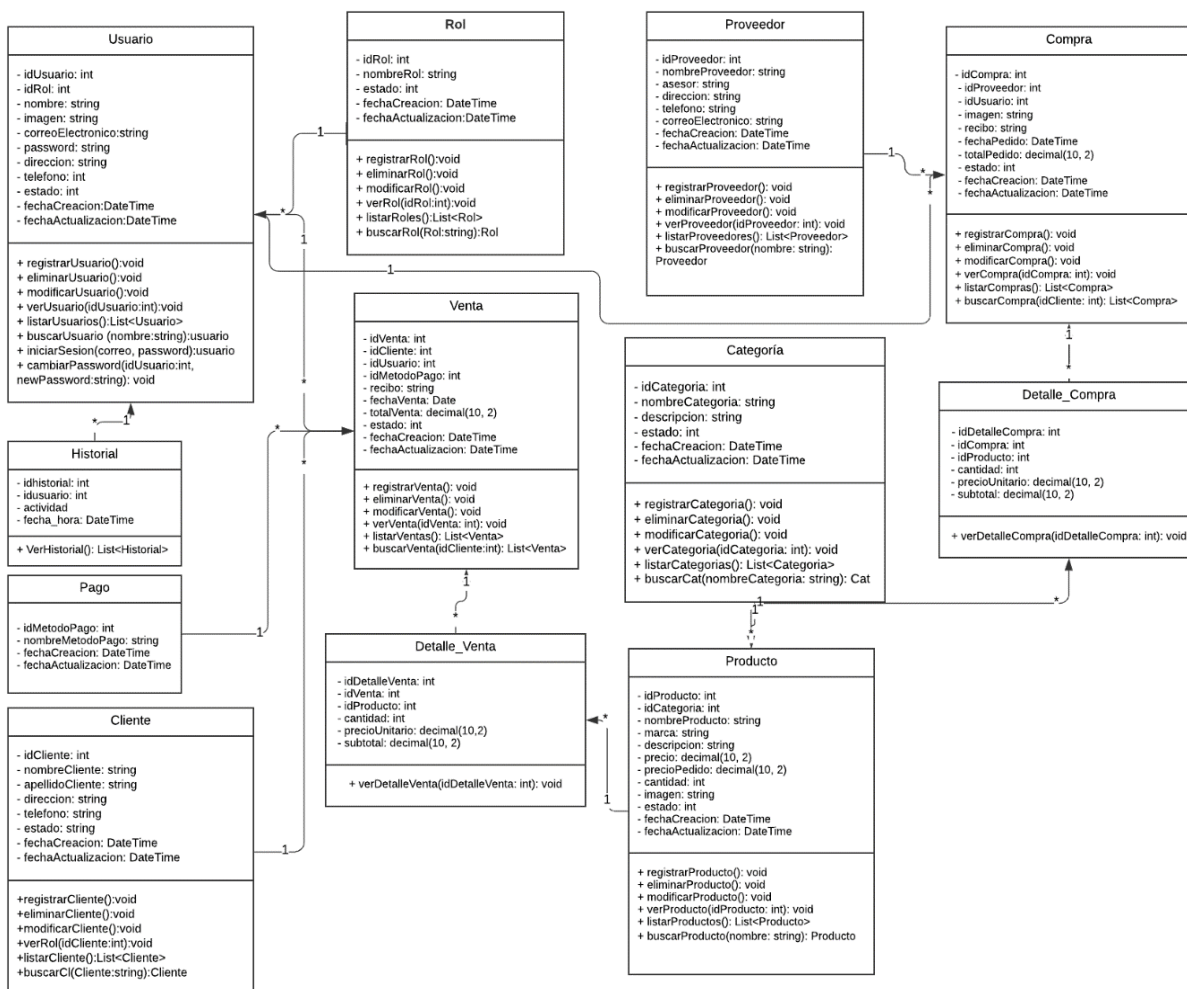
Caso de Uso	Gestionar Proveedores
Actores:	Administrador y Supervisor de Ventas
Propósito:	Tiene como propósito permitir a los actores gestionar la información de proveedores en el sistema. Los administradores y supervisores de venta pueden llevar a cabo acciones relacionadas con la gestión de proveedores. Esto incluye la creación, actualización y eliminación de registros de proveedores y demás información pertinente.
Resumen:	
Tipo:	Primario Esencial

3.5.1.3 Diseño conceptual. En la siguiente figura se muestran las clases relacionadas

que contiene el Sistema:

Figura 29

Diseño Conceptual del Sistema



Nota. Diseño conceptual mediante el diagrama de clases, donde contiene todas las entidades del Sistema con sus atributos y métodos.

3.5.1.5 Diseño de Presentación.

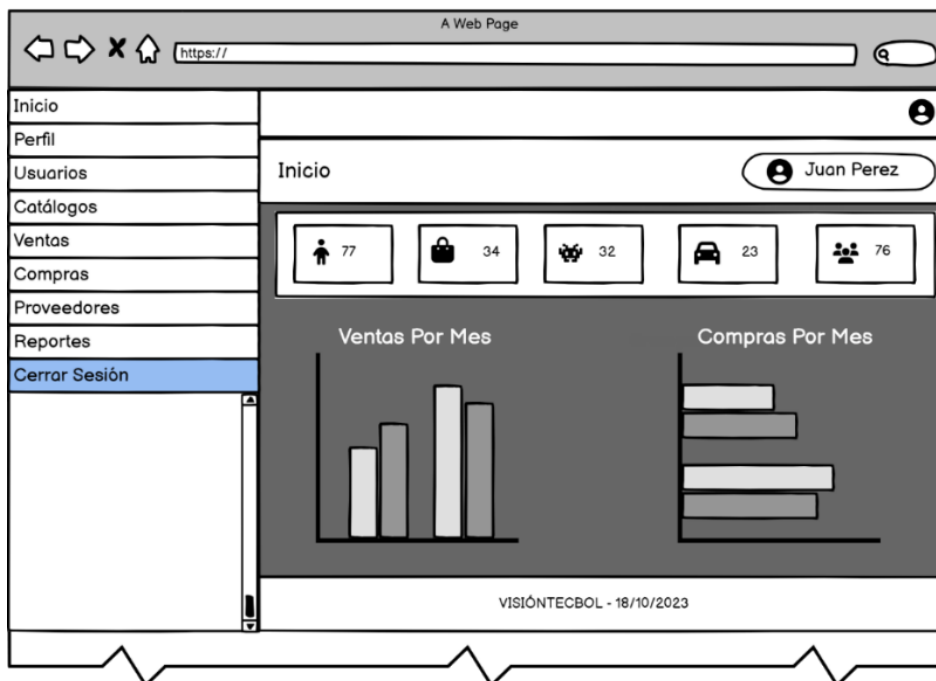
Figura 32

Modelo de Presentación - Iniciar Sesión



Figura 33

Modelo de Presentación - Página Principal



3.6. Codificación

3.6.1. Configuración e Instalación del Framework Laravel Livewire

A continuación, se instala y configura el Framework Laravel y Livewire:

Figura 34

Instalación de Laravel

```

Administrador Símbolo del sistema - composer create-project laravel/laravel:10.0 SistemaInformacion
Microsoft Windows [Versión 10.0.19045.4412]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>cd/

C:\>cd sistemas

C:\Sistemas>composer create-project laravel/laravel:^10.0 SistemaInformacion
Creating a "laravel/laravel:10.0" project at "./SistemaInformacion"
Installing laravel/laravel (v10.0.0)
- Downloading laravel/laravel (v10.0.0)
- Installing laravel/laravel (v10.0.0): Extracting archive
Created project in C:\Sistemas\SistemaInformacion
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.1)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.10)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.3)
- Locking egulias/email-validator (4.0.2)
- Locking fakerphp/faker (v1.23.1)
- Locking filp/whoops (2.15.4)
- Locking fruitcake/php-cors (v1.3.0)
- Locking graham-campbell/result-type (v1.1.2)
- Locking guzzlehttp/guzzle (7.8.1)
- Locking guzzlehttp/promises (2.0.2)
- Locking guzzlehttp/psr7 (2.6.2)
- Locking guzzlehttp/uri-template (v1.0.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.48.11)
- Locking laravel/pint (v1.16.0)
- Locking laravel/prompts (v0.1.22)
- Locking laravel/sail (v1.29.2)
- Locking laravel/sanctum (v3.3.3)
- Locking laravel/serializable-closure (v1.3.3)
- Locking laravel/tinker (v2.9.0)
- Locking league/commonmark (2.4.2)
- Locking league/config (v1.2.0)
- Locking league/flysystem (3.27.0)
- Locking league/flysystem-local (3.25.1)

```

Figura 35

Instalación de Livewire

```

Administrador Símbolo del sistema

C:\Sistemas\SistemaInformacion>composer require livewire/livewire
./composer.json has been updated
Running composer update livewire/livewire
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking livewire/livewire (v3.4.12)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing livewire/livewire (v3.4.12): Extracting archive
Generating optimized autoload files
> illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
livewire/livewire ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

95 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

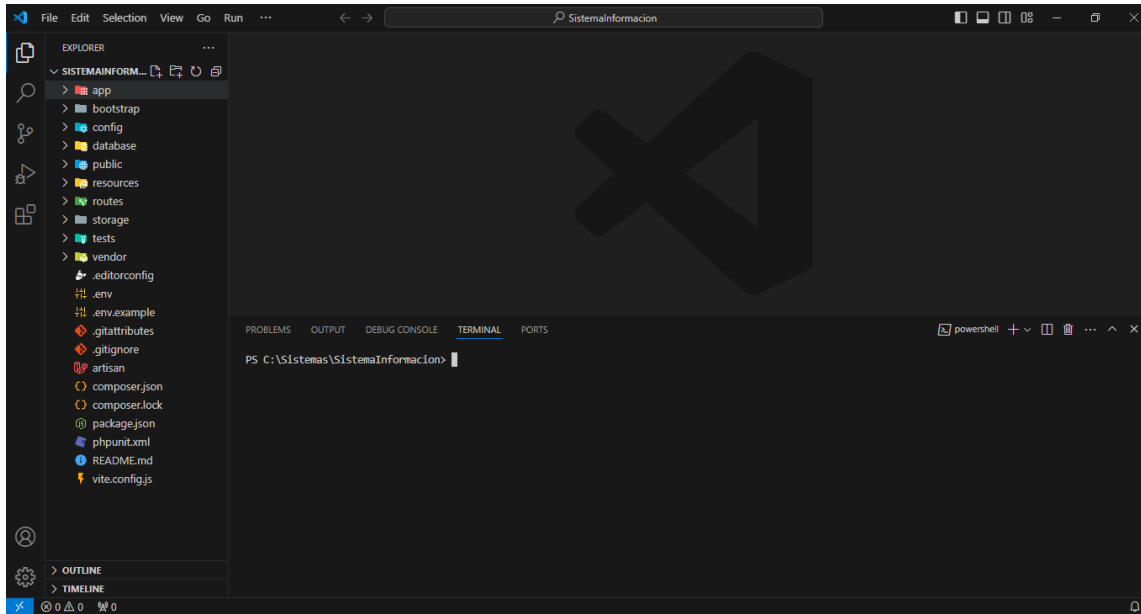
No security vulnerability advisories found
Using version ^3.4 for livewire/livewire

C:\Sistemas\SistemaInformacion>

```

Figura 36

Framework Laravel Livewire

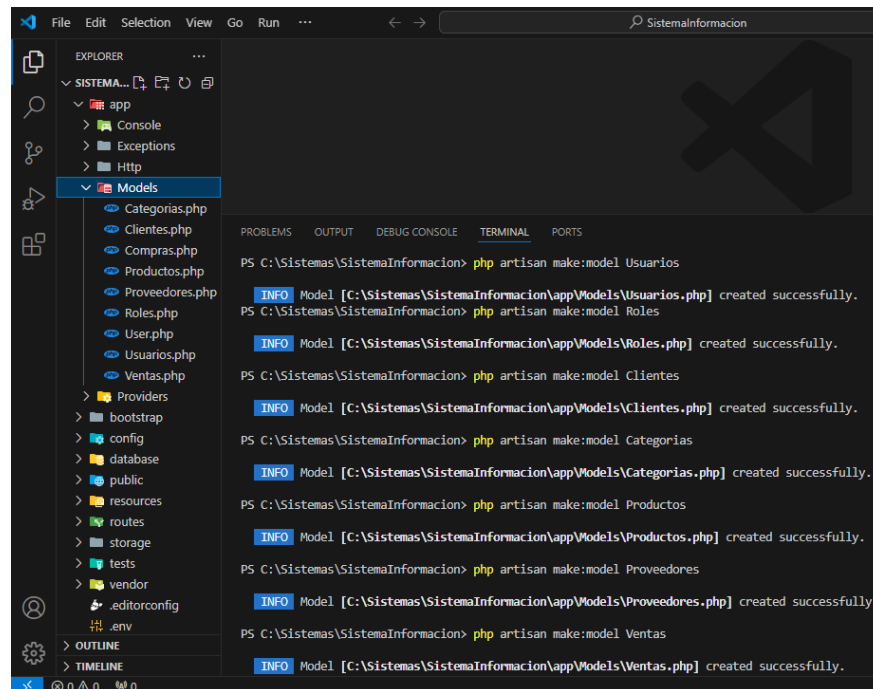


3.6.2. Creación de Modelos y Migraciones

Se crea las modelos y migraciones para las siguientes entidades:

Figura 37

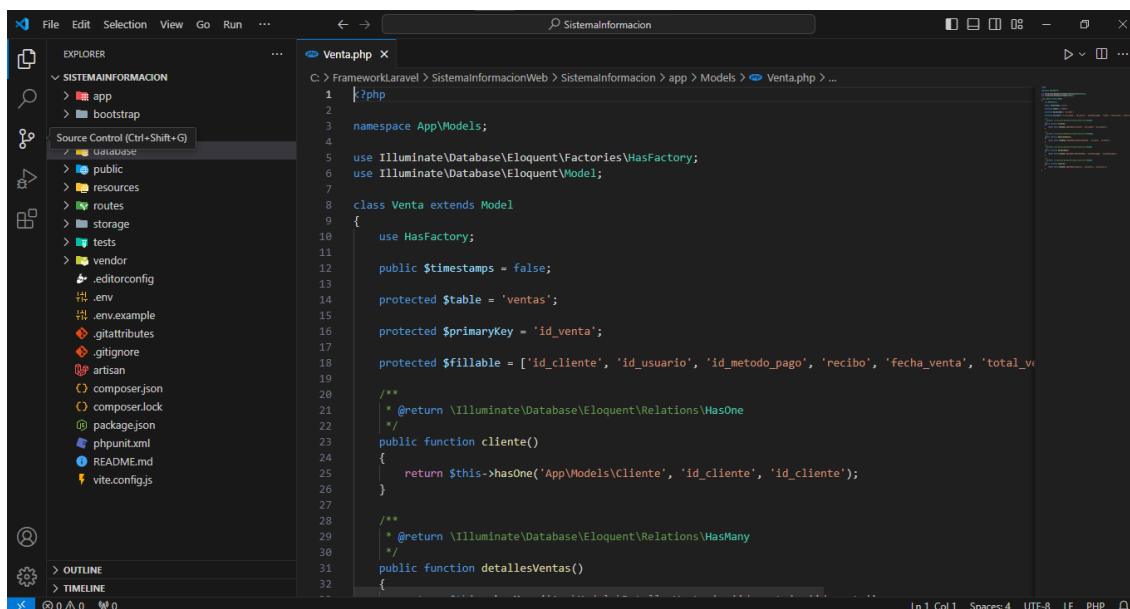
Creación de Modelos



En el modelo “Ventas” se configura los atributos y las relaciones necesarias que se ve a continuación:

Figura 38

Configuración del Modelo "Ventas"

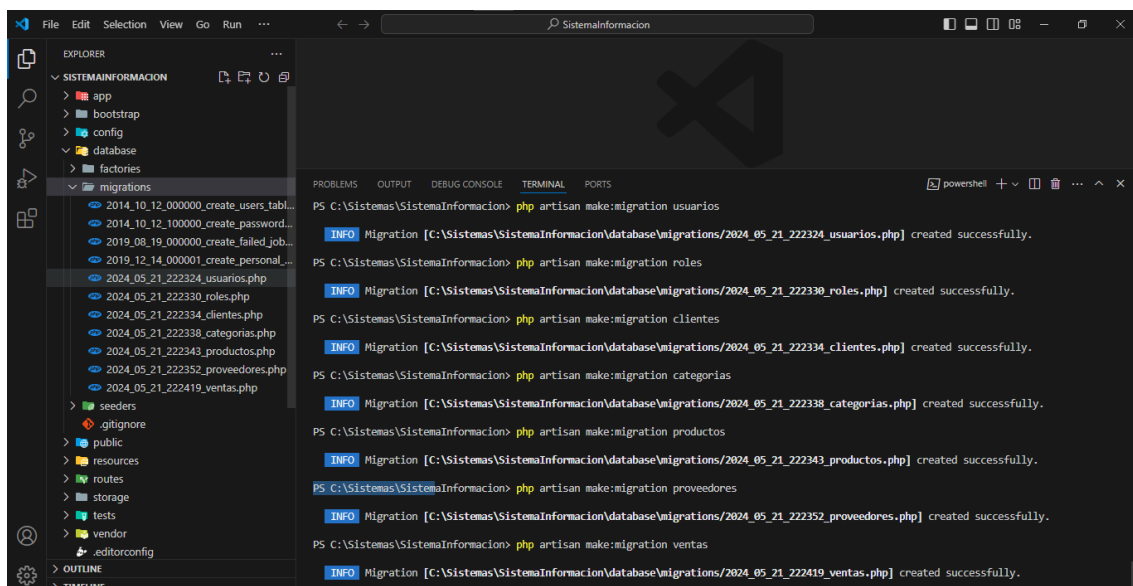


```

1  {?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Venta extends Model
9  {
10     use HasFactory;
11
12     public $timestamps = false;
13
14     protected $table = 'ventas';
15
16     protected $primaryKey = 'id_venta';
17
18     protected $fillable = ['id_cliente', 'id_usuario', 'id_metodo_pago', 'recibo', 'fecha_venta', 'total_v
19
20     /**
21      * @return \Illuminate\Database\Eloquent\Relations\HasOne
22      */
23     public function cliente()
24     {
25         return $this->hasOne('App\Models\Cliente', 'id_cliente', 'id_cliente');
26     }
27
28     /**
29      * @return \Illuminate\Database\Eloquent\Relations\HasMany
30      */
31     public function detallesVentas()
32     {
  
```

Figura 39

Creación de Migraciones (Base de Datos del Sistema)



```

PS C:\Sistemas\SistemaInformacion> php artisan make:migration usuarios
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222324_usuarios.php] created successfully.

PS C:\Sistemas\SistemaInformacion> php artisan make:migration roles
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222330_roles.php] created successfully.

PS C:\Sistemas\SistemaInformacion> php artisan make:migration clientes
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222334_clientes.php] created successfully.

PS C:\Sistemas\SistemaInformacion> php artisan make:migration categorias
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222338_categorias.php] created successfully.

PS C:\Sistemas\SistemaInformacion> php artisan make:migration productos
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222343_productos.php] created successfully.

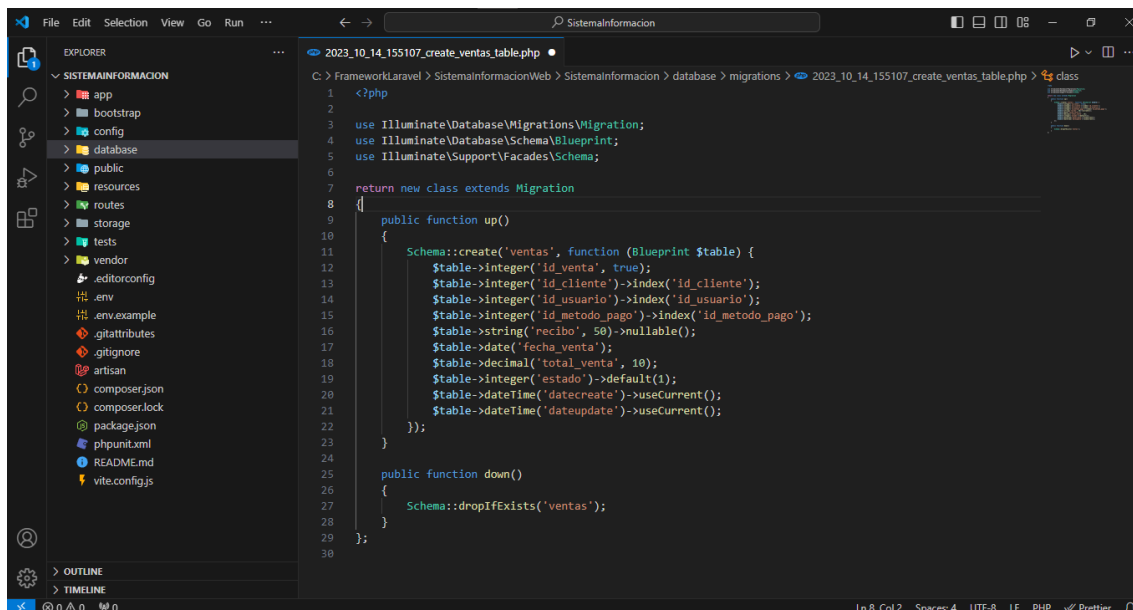
PS C:\Sistemas\SistemaInformacion> php artisan make:migration proveedores
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222352_proveedores.php] created successfully.

PS C:\Sistemas\SistemaInformacion> php artisan make:migration ventas
[INFO] Migration [C:\Sistemas\SistemaInformacion\database\Migrations\2024_05_21_222419_ventas.php] created successfully.
  
```

En el archivo de migración “Ventas”, se define los atributos según el diagrama de clases:

Figura 40

Definición de Atributos de la Migración "Ventas"



```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     public function up()
10    {
11        Schema::create('ventas', function (Blueprint $table) {
12            $table->integer('id_venta', true);
13            $table->integer('id_cliente')->index('id_cliente');
14            $table->integer('id_usuario')->index('id_usuario');
15            $table->integer('id_metodo_pago')->index('id_metodo_pago');
16            $table->string('recibo', 50)->nullable();
17            $table->date('fecha_venta');
18            $table->decimal('total_venta', 10);
19            $table->integer('estado')->default(1);
20            $table->dateTime('datecreate')->useCurrent();
21            $table->dateTime('dateupdate')->useCurrent();
22        });
23    }
24
25    public function down()
26    {
27        Schema::dropIfExists('ventas');
28    }
29 };
30

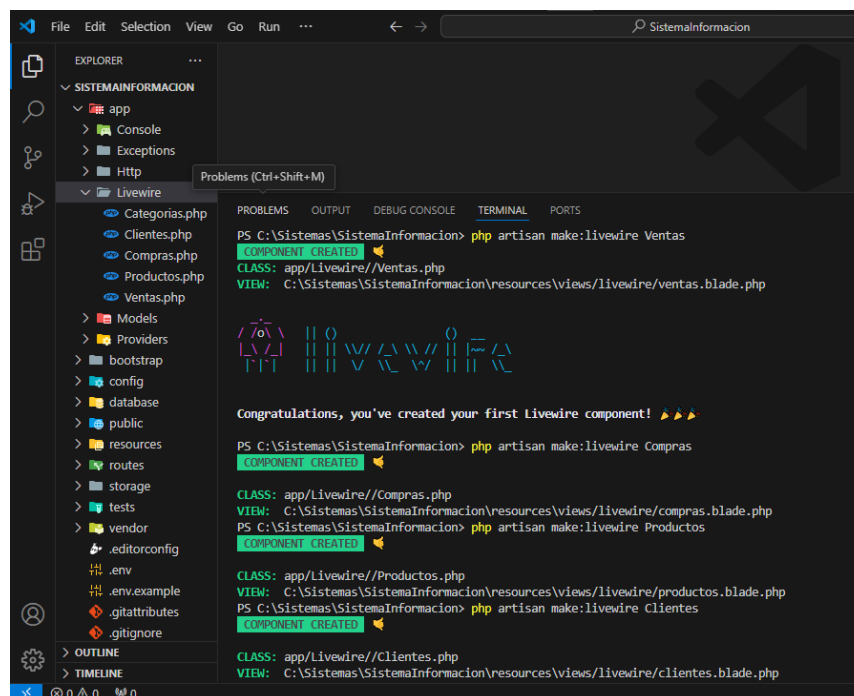
```

3.6.3. Creación de Componentes

Se crea los componentes para listar la información de cada entidad y sus funciones:

Figura 41

Creación de Componentes con Livewire



```

PS C:\Sistemas\SistemaInformacion> php artisan make:livewire Ventas
COMPONENT CREATED
CLASS: app/Livewire/Ventas.php
VIEW: C:\Sistemas\SistemaInformacion\resources\views/livewire/ventas.blade.php

PS C:\Sistemas\SistemaInformacion> php artisan make:livewire Compras
COMPONENT CREATED
CLASS: app/Livewire/Compras.php
VIEW: C:\Sistemas\SistemaInformacion\resources\views/livewire/compras.blade.php

PS C:\Sistemas\SistemaInformacion> php artisan make:livewire Productos
COMPONENT CREATED
CLASS: app/Livewire/Productos.php
VIEW: C:\Sistemas\SistemaInformacion\resources\views/livewire/productos.blade.php

PS C:\Sistemas\SistemaInformacion> php artisan make:livewire Clientes
COMPONENT CREATED
CLASS: app/Livewire/Clientes.php
VIEW: C:\Sistemas\SistemaInformacion\resources\views/livewire/clientes.blade.php

```

Figura 42

Contenido del Componente "Ventas"

```

1 | ?php
2 |
3 | namespace App\Http\Livewire;
4 |
5 | use App\Exports\VentaExport;
6 | use App\Models\Venta;
7 | use App\Models\Cliente;
8 | use App\Models\Usuario;
9 | use Livewire\Component;
10 | use App\Models\Producto;
11 | use App\Models\MetodoPago;
12 | use Livewire\WithPagination;
13 | use App\Models\DetalleVenta;
14 | use Maatwebsite\Excel\Facades\Excel;
15 | use Barryvdh\DomPDF\Facade\Pdf as PDF;
16 | use SimpleSoftwareIO\QrCode\Facades\QrCode;
17 |
18 | class Ventas extends Component
19 | {
20 |     use WithPagination;
21 |
22 |     protected $paginationTheme = 'bootstrap';
23 |     public $select_id, $keyword, $id_venta, $id_cliente, $id_usuario, $id_metodo_pago, $recibo, $fecha_venta, $total_venta, $estado, $datec;
24 |     public $show = 10, $total;
25 |
26 |     public function render()
27 |     {
28 |         $keyword = 'X'; $this->keyword = 'X';
29 |         $this->total = Venta::where('estado', '=', 1)->count();
30 |         $usuarios = Usuario::all();
31 |         $pagos = MetodoPago::all();
32 |         $productos = Producto::all();
33 |         $clientes = Cliente::all();
34 |         $detalles = DetalleVenta::all();
35 |
36 |         return view('livewire.ventas.view', [
37 |             'ventas' => Venta::where('estado', '=', 1)
38 |                 ->where(function ($query) use ($keyword) {
39 |                     $query->where('id_cliente', 'LIKE', $keyword)

```

Vista correspondiente al Componente "Ventas"

Figura 43

Vista del Componente "Ventas"

```

2 | <div class="container-fluid">
3 |     <div class="row justify-content-center">
4 |         <div class="col-md-12">
5 |             <div class="card">
29 |                 <div class="card-body">
35 |                     <option value="25">25</option>
36 |                     <option value="50">50</option>
37 |                     <option value="100">100</option>
38 |                 </select>
39 |             </div>
40 |             <label for="mostrar" class="col-sm-1 mt-1 col-form-label">Mostrar</label>
41 |             <div class="col-md-3 offset-md-7">
42 |                 <input wire:model="keyword" type="text" class="form-control bg-white" name="search" id="sea
43 |             </div>
44 |         </div>
45 |     </div>
46 |     <div class="table-responsive" style="border-radius: 10px;">
47 |         <table class="table table-hover table-secondary table-bordered table-sm">
48 |             <thead class="thead table-primary text-capitalize">
49 |                 <tr>
50 |                     <th class="text-center align-middle" style="padding: 11px; width="50">Código</th>
51 |                     <th class="text-center align-middle" style="padding: 11px;">Cliente</th>
52 |                     <th class="text-center align-middle" style="padding: 11px;">Responsable</th>
53 |                     <th class="text-center align-middle" style="padding: 11px;">Método Pago</th>
54 |                     <th class="text-center align-middle" style="padding: 11px;">Nro. Recibo</th>
55 |                     <th class="text-center align-middle" style="padding: 11px;">Fecha</th>
56 |                     <th class="text-center align-middle" style="padding: 11px;">Ganacia Total (Bs)</th>
57 |                     <th class="text-center align-middle" style="padding: 11px;">Estado</th>
58 |                     <th class="text-center align-middle" style="padding: 11px;">Acciones</th>
59 |                 </tr>
60 |             </thead>
61 |         </table>

```

3.7. Pruebas

Las pruebas son un proceso esencial que tienen como objetivo detectar errores, identificar problemas y garantizar el funcionamiento correcto del software las cuales se describen los métodos a utilizar:

3.7.1. Pruebas Unitarias usando el método de Caja Blanca (PHPUnit)

Para este proceso, se utiliza la herramienta PHPUnit, que este ya viene integrado por defecto en el Framework Laravel.

Para este proceso, se utiliza el componente Clientes para guardar la información de un nuevo cliente mediante la función "store".

Figura 44

Código de la Función "store" del Componente Cliente

```
public function store()
{
    $this->validate([
        'nombre_cliente' => 'required',
        'apellido_cliente' => 'required',
        'direccion' => 'required',
        'telefono' => 'required'
    ]);
    Cliente::create([
        'ci' => 123456789,
        'nombre_cliente' => $this->nombre_cliente,
        'apellido_cliente' => $this->apellido_cliente,
        'direccion' => $this->direccion,
        'telefono' => $this->telefono,
        'estado' => 1,
    ]);

    $ultimo_id = DB::getPdo()->lastInsertId();

    Historial_Usuario::create([
        'id_usuario' => $this->id_usuario,
        'tipo_actividad' => 'Creación',
        'tabla_afectada' => 'Clientes',
        'id_registro_afectado' => $ultimo_id
    ]);

    $this->resetInput();
    $this->dispatchBrowserEvent('closeModal');
    session()->flash('message', '¡Tu registro se ha insertado con éxito!');
}
```

Figura 45

Prueba de código a la función store

```

12 class ClienteTest extends TestCase
13 {
14     use RefreshDatabase;
15
16     /** @test */
17     public function testStoreCliente()
18     {
19         Livewire::test(Clientes::class,)
20             ->set('ci', 123456789)
21             ->set('nombre_cliente', 'John')
22             ->set('apellido_cliente', 'Doe')
23             ->set('direccion', '123 Main St')
24             ->set('telefono', '1234567890')
25             ->call('store');
26
27         // Verificar que se hayan creado los registros en la base de datos
28         $this->assertDatabaseHas('clientes', [
29             'ci' => 123456789,
30             'nombre_cliente' => 'John',
31             'apellido_cliente' => 'Doe',
32             'direccion' => '123 Main St',
33             'telefono' => '1234567890'
34         ]);
35
36         // Obtener los datos insertados en la tabla clientes
37         $cliente = Cliente::where('ci', 123456789)->first();
38
39         // Verificar que los datos se han insertado correctamente
40         $this->assertEquals('John', $cliente->nombre_cliente);
41         $this->assertEquals('Doe', $cliente->apellido_cliente);
42         DB::commit();
43     }

```

Se realiza la prueba a la función “store”, esta verifica si el proceso de almacenamiento de clientes funciona correctamente, estableciendo valores, almacenándolos en la base de datos y luego verificando si los datos se insertaron correctamente. Y Por último se procede a ejecutar.

Figura 46

Ejecución de la Prueba

```

17 public function testStoreCliente()
18 {
19     Livewire::test(Clientes::class,)
20         ->set('ci', 123456789)
21         ->set('nombre_cliente', 'John')
22         ->set('apellido_cliente', 'Doe')
23         ->set('direccion', '123 Main St')
24         ->set('telefono', '1234567890')
25         ->call('store');
26
27     // Verificar que se hayan creado los registros en la base de
28     $this->assertDatabaseHas('clientes', [

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Sistema Web\SistemaPruebas> vendor\bin\phpunit tests/Unit/ClienteTest.php
PHPUnit 9.6.17 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 00:03.379, Memory: 38.00 MB
OK (1 test, 3 assertions)
PS D:\Sistema Web\SistemaPruebas>

```

Después de esperar con atención los resultados, confirmamos que todo está funcionando correctamente. Esto demuestra que la codificación y la función "store" han sido implementadas adecuadamente.

3.7.2. Pruebas de Integración usando el método de Caja Blanca (PHPUnit)

Este tipo de pruebas se realiza una vez concluida las pruebas unitarias, a continuación, se muestra el proceso de dicha prueba.

Figura 47

Llamado de Varios Componentes

```
class VentaTest extends TestCase
{
    use RefreshDatabase;

    public function testSaveSale()
    {
        // Crear un cliente ficticio para evitar la violación de
        $cliente = Cliente::create([
            'nombre_cliente' => 'Cliente de Prueba',
            'ci' => 12345456,
            'apellido_cliente' => 'Apellido de Prueba'
        ]);

        // Crear un rol
        $rol = Role::create([
            'id_rol' => 1,
            'nombre_rol' => 'admin',
        ]);

        // Crear un usuario real en la base de datos de pruebas
        $usuario = Usuario::create([
            'id_rol' => 1,
            'nombre' => 'Usuario de Prueba',
            'email' => 'usuarioprueba@example.com',
            'password' => bcrypt('password')
        ]);

        $categoria = Categoria::create([
            'id_categoria' => 1,
            'nombre_categoria' => 'camaras',
            'descripcion' => 'Camaras Dahua'
        ]);

        $producto = Producto::create([
            'id_categoria' => 1,
            'nombre_producto' => 'camara dahua 1mp',
            'descripcion' => 'color blanco de metal',
            'precio' => 150,
            'cantidad' => 25,
            'precio_pedido' => 120
        ]);

        $recibo = Recibo::create([
            'id' => 1,
            'numero' => 8003
        ]);
    }
}
```

Se procede a llamar a los componentes clientes, usuarios, roles, métodos de pago, categorías y productos, que estos están relacionados con la función ventas, para luego realizar una simulación de venta con los componentes anteriormente descritos. Y por último ejecutar la prueba.

Figura 48

Prueba a la Función Venta

```

// Mock de Recibo
$reciboMock = m::mock('App\Models\Recibo');
$reciboMock->shouldReceive('save');
$reciboMock->shouldReceive('setAttribute')->andReturnSelf();
$reciboMock->id = 1;

// Mock de Usuario
$usuarioMock = m::mock('App\Models\Usuario')->shouldReceiveIgnoreMissing();
$usuarioMock->shouldReceive('getAttribute')->with('id_usuario')->andReturn($usuario->id_usuario);
Auth::shouldReceive('guard->user')->andReturn($usuarioMock);

// Mock de Venta
$ventaMock = m::mock('App\Models\Venta')->shouldReceiveIgnoreMissing();
$ventaMock->shouldReceive('save');
$ventaMock->id_venta = 1;

// Mock de Producto
$productoMock = m::mock('App\Models\Producto')->shouldReceiveIgnoreMissing();
$productoMock->shouldReceive('findOrFail')->andReturn((object) ['id_producto' => 1, 'cantidad' => 10]);
$productoMock->shouldReceive('update');

// Mock de DetallesVenta
$dDetalleVentaMock = m::mock('App\Models\DetallesVenta')->shouldReceiveIgnoreMissing();
$dDetalleVentaMock->shouldReceive('save');

// Mock de HistorialUsuario
$historialUsuarioMock = m::mock('App\Models\HistorialUsuario')->shouldReceiveIgnoreMissing();
$historialUsuarioMock->shouldReceive('create');

// Instancia de la clase que contiene el método saveSale
$ventaService = new Ventasave();
$ventaService->selectedClient = $cliente->id_cliente; // Usar el cliente ficticio creado
$ventaService->selectedPago = $pago->id_metodo_pago;
$ventaService->cart = [
    [
        'id_producto' => 1, // Id de producto existente en la base de datos
        'cantidad' => 2,
        'precio' => 100,
        'subtotal' => 200
    ]
];

```

Figura 49

Ejecución de la Prueba

```

VentaTest.php X
ests > Unit > VentaTest.php > VentaTest > testSaveSale
20 class VentaTest extends TestCase
24 public function testSaveSale()
73 // Mock de Recibo
74 $reciboMock = m::mock('App\Models\Recibo');
75 $reciboMock->shouldReceive('save');
76 $reciboMock->shouldReceive('setAttribute')->andReturnSelf();
77 $reciboMock->id = 1;
78
79 // Mock de Usuario
80 $usuarioMock = m::mock('App\Models\Usuario')->shouldReceiveIgnoreMissing();
81 $usuarioMock->shouldReceive('getAttribute')->with('id_usuario')->andReturn($usuario->id_usuario);
82 Auth::shouldReceive('guard->user')->andReturn($usuarioMock);
83
84 // Mock de Venta
85 $ventaMock = m::mock('App\Models\Venta')->shouldReceiveIgnoreMissing();
86 $ventaMock->shouldReceive('save');
87 $ventaMock->id_venta = 1;
88
89 // Mock de Producto
90 $productoMock = m::mock('App\Models\Producto')->shouldReceiveIgnoreMissing();
91 $productoMock->shouldReceive('findOrFail')->andReturn((object) ['id_producto' => 1, 'cantidad' => 10]);
92 $productoMock->shouldReceive('update');
93
94 // Mock de DetallesVenta
95 $dDetalleVentaMock = m::mock('App\Models\DetallesVenta')->shouldReceiveIgnoreMissing();
96 $dDetalleVentaMock->shouldReceive('save');
97
98 // Mock de HistorialUsuario
99 $historialUsuarioMock = m::mock('App\Models\HistorialUsuario')->shouldReceiveIgnoreMissing();
100 $historialUsuarioMock->shouldReceive('create');

```

PS D:\Sistema Web\SistemaPruebas> vendor\bin\phpunit tests\Unit\VentaTest.php
 PHPUnit 9.6.17 by Sebastian Bergmann and contributors.

```

[] // tests\Unit\VentaTest.php:125
.
1 / 1 (100%)

Time: 00:03.456, Memory: 40.00 MB

OK (1 test, 3 assertions)
PS D:\Sistema Web\SistemaPruebas>

```

3.7.3. Pruebas de Sistema usando el método de Caja Negra (Selenium)

Una vez concluidas las pruebas unitarias y de integración se procede a realizar las pruebas de sistemas, utilizando la herramienta Selenium que a continuación se muestran los detalles del proceso.

Figura 50

Configuración de Nombre del Proyecto

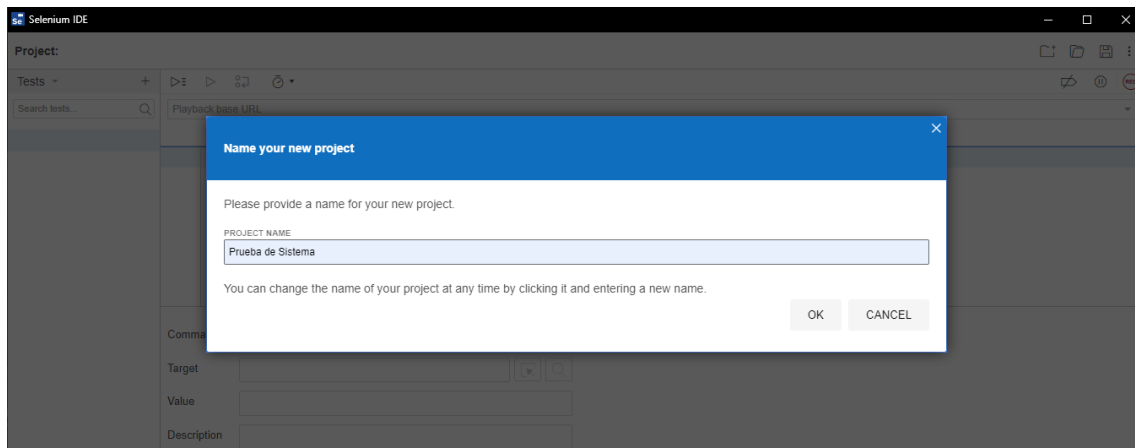


Figura 51

Vista Clientes De Prueba

Nro	C.I.	Nombre Completo	Direccion	Whatsapp	Estado	Fecha Creada	Fecha Actualizada	Acciones
1	1234567	Juan Perez	Calle 123, La Paz, Bolivia	71234567	Activo	2024-05-25 01:47:45	2024-05-25 02:02:49	[Edit] [Delete]
2	2345678	Maria Gomez	Av. America, Cochabamba, Bolivia	61234567	Activo	2024-05-25 01:47:45	2024-05-25 02:02:49	[Edit] [Delete]
3	3456789	Carlos Martinez	Calle 456, Santa Cruz, Bolivia	71234568	Activo	2024-05-25 01:47:45	2024-05-25 02:02:49	[Edit] [Delete]
4	4567890	Laura Rodriguez	Av. 7 de Octubre, La Paz, Bolivia	61234568	Activo	2024-05-25 01:47:45	2024-05-25 02:02:49	[Edit] [Delete]
5	5678901	Pedro Lopez	Calle Sucre, Cochabamba, Bolivia	71234569	Activo	2024-05-25 01:47:45	2024-05-25 02:02:49	[Edit] [Delete]

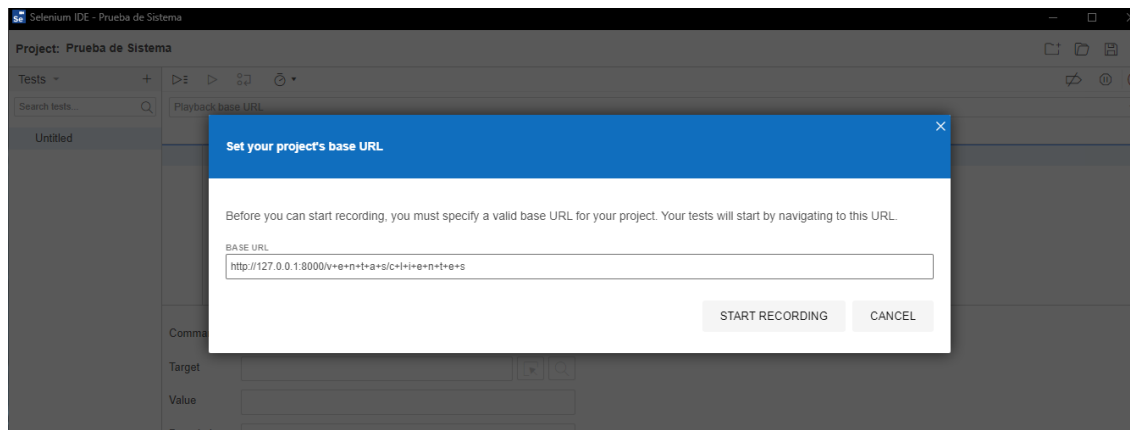
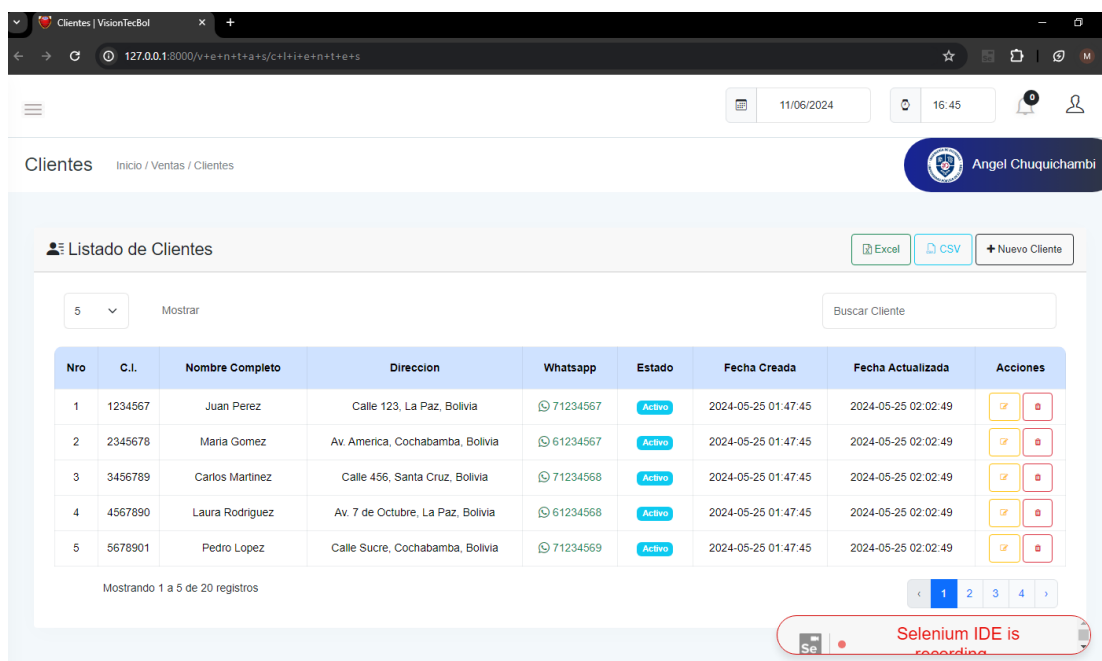
Figura 52**Configuración de Parámetros URL****Figura 53****Ejecución de Prueba con Selenium**

Figura 54

Resultados Finales de la Prueba

The screenshot displays the Selenium IDE interface for a test named 'Untitled'. The test is currently running, as indicated by the 'Executing' status and the 'Runs: 1 Failures: 1' indicator. The test steps are listed in a table below:

Step	Command	Target	Value
1	open	/v+e+n+t+a+s/c+h+i+e+n+t+e+s	
2	set window size	1456x876	
3	click	css=div:nth-child(2) > .btn-outline-dark	
4	click	id=nombre_cliente	
5	type	id=nombre_cliente	Dania
6	type	id=apellido_cliente	Flores
7	click	id=direccion	

Below the table, the current step configuration is shown: Command: type, Target: id=nombre_cliente, Value: 13849533. The Log window at the bottom shows the execution progress, with the first seven steps completed successfully (OK) and the eighth step (click on id=telefono) failing (FK).

3.8. Implementación

En esta fase marca la culminación del desarrollo del Sistema. Tras una serie de rigurosas fases o etapas, se pone en marcha el Sistema en un entorno de Hosting “000webhost”, disponible 24/7.

Para acceder al Sistema copie el siguiente enlace:

<https://visiotecboliviana.000webhostapp.com/>

CAPÍTULO IV

CAPÍTULO IV: CALIDAD, SEGURIDAD Y COSTOS

4.1. Introducción

En el desarrollo de cualquier sistema, el capítulo pruebas y resultados desempeña un papel importante para garantizar su funcionamiento óptimo y su capacidad de cumplir con los objetivos establecidos. En el contexto del presente proyecto, se presentan en detalle las pruebas exhaustivas a las que ha sido sometido el sistema, evaluando su rendimiento, su usabilidad y su capacidad de satisfacer las necesidades de los usuarios finales. Además, en este capítulo se muestran los resultados obtenidos en las pruebas, tanto funcionales como de usabilidad, destacando los logros alcanzados y los desafíos superados.

4.2. Métricas De Calidad Basado En El ISO 25010

4.2.1. Funcionalidad

La Funcionalidad del presente este proyecto se centra en ofrecer una amplia gama de capacidades destinadas a satisfacer las necesidades y requerimientos de nuestros usuarios finales. Esto incluye, pero no se limita a, la implementación de las siguientes funcionalidades clave:

4.2.1.1 Funciones de tipo Entrada Externa (EE).

Tabla 10

Funciones de Entradas Externas (EE)

Nro.	Descripción	Ponderación
1	Registro de Cliente	6
2	Registro de Producto	7
3	Registro de Proveedor	6
5	Registro de Venta	10
6	Registro de Usuarios	8
7	Registro de Compras	10
Total		56

4.2.1.2 Funciones de tipo Salida Externa (SE).

Tabla 11

Funciones de Salida Externa

Nro.	Descripción	Ponderación
1	Generación de Recibos de Venta	10
2	Generación de Proforma de Venta	10
3	Generación de Reporte de Ventas	6
4	Generación de Reporte de Compras	5
5	Generación de Reporte de Productos	5
6	Generación de Reporte de Productos más Vendidos	6
7	Generación de Reporte de Compras	5
8	Exportación de Datos en CSV	5
9	Exportación de Datos en EXCEL	5
Total		52

4.2.1.3 Funciones de Tipo Consulta Externa (EQ).

Tabla 12

Función de Consulta Externa

Nro.	Descripción	Ponderación
1	Consulta de Productos Disponibles	5
2	Consulta de Productos Agotados	5
3	Búsqueda de Recibos de Ventas	5
4	Consulta de Información de Clientes	4
5	Consulta de Información de Proveedores	4
6	Consulta de Información de Ventas	5
7	Consulta de Estado de Ventas y Pedidos	5
8	Consulta de Información de Usuarios	4
9	Consulta de Información de Compras	4
Total		41

4.2.1.4 Funciones de Tipo Archivo Lógico Interno (ILF).

Tabla 13

Función de Archivo Lógico Interno

Nro.	Descripción	Ponderación
1	Módulo de Usuarios	6
2	Módulo de Productos	8
3	Módulo de Ventas	12
4	Módulo de Compras	8
5	Módulo de Proveedores	6
6	Módulo de Reportes	8
Total		48

4.2.1.5 Funciones de Tipo Archivo Lógico Externo (ELF).

Tabla 14

Función de Archivo Lógico Externo

Nro.	Descripción	Ponderación
1	Importación de Datos de Proveedor	1
2	Exportación de Datos de Ventas	1
3	Exportación de Informe de Ventas	1
4	Importación de Datos de Productos	1
5	Importación de Datos de Compras	1
Total		5

4.2.1.6 Cálculo de los Puntos de Función Sin Ajustar (PFSA).

Tabla 15

Cálculo de Puntos de Función Sin Ajustar

Tipo	Sumatoria	Simple	Medio	Complejo	Total
Entrada Externa	56	x3	x4	x6	168
Salida Externa	52	x4	x5	x7	208
Consulta Externa	41	x3	x10	x15	123
Archivo Lógico Interno	48	x7	x7	x10	336
Archivo Lógico Externo	5	x5	x4	x6	25
Total					857

4.2.1.7 Estimación de Funcionalidad.

La estimación se realiza tomando en cuenta

las ponderaciones definidas que son:

- 1 => Mínimo
- 2 => Baja Prioridad
- 3 => Moderado
- 4 => Alto
- 5 => Muy Alto

Tabla 16

Factor de Ajuste

Nro.	Factor de Ajuste	Ponderación
1	Comunicación de Datos	4
2	Proceso Distributivo	4
3	Objetivos de Rendimiento	4
4	Configuración usada por otros sistemas	3
5	Tasa de transacciones	3
6	Entrada de datos en línea	4
7	Eficiencia con el usuario final	4
8	Actualización en línea	5
9	Lógica del proceso Interno Compleja	3
10	Reusabilidad del código	5
11	Contempla la conversión e instalación	1
12	Facilidad de Operación	4
13	Instalaciones Múltiples	2
14	Facilidad de Cambios	5
Factor de Ajuste Total (FAT)		51

Para la estimación se utiliza la siguiente formula:

$$PF = PFSA * (0,65 + 0,01 * FAT)$$

$$PF = 857 * (0,65 + 0,01 * 51)$$

$$PF = 994$$

Tomando como valor máximo 70 del Factor de Ajuste nos queda de la siguiente manera:

$$PFA = PFSA * (0,65 + 0,01 * FAT)$$

$$PFA = 857 * (0,65 + 0,01 * 70)$$

$$PFA = 1156$$

Con estos datos, se procede a estimar la funcionalidad del proyecto en términos de porcentaje.

$$F = \frac{PF}{PFA} * 100$$

$$F = \frac{994}{1156} * 100$$

$$F = 84,1591 \Rightarrow 84\%$$

Por tanto, la funcionalidad del proyecto es del 84%.

4.2.2. Fiabilidad

Para realizar el cálculo de la fiabilidad del proyecto, primero se debe obtener la estimación de la disponibilidad, para luego realizar la estimación de la fiabilidad. Entonces para el presente caso, se utiliza las siguientes formulas:

$$D = (TOT - TI)/TOT$$

Donde:

- D => Disponibilidad
- TOT => Tiempo de Operación Total
- TI => Tiempo de Inactividad

4.2.2.1 Definición de Datos. Para la presente estimación, se toma en cuenta que el sistema entra en funcionamiento durante un mes, por lo cual se observa problemas de tiempos de carga moderados de un total de 15 minutos por día aproximadamente.

Entonces con los datos mencionados con anterioridad, se procede a estimar la Fiabilidad:

$$TI = 15 \frac{\text{minutos}}{\text{días}} * 30 \text{ días} = 450 \text{ minutos}$$

Conversión de TI, formato de unidades de días a horas:

$$TI = \frac{900 \text{ minutos}}{60 \text{ minutos}} = 7,5 \text{ horas} \Rightarrow 8 \text{ horas}$$

Conversión de TOT, formato de unidades de días a horas:

$$TOT = 30 \text{ días} * 24 \frac{\text{horas}}{\text{día}} = 720 \text{ horas}$$

Estimación de la Fiabilidad Total:

$$D = \frac{(720 \text{ horas} - 8 \text{ horas})}{720 \text{ horas}}$$

$$D = 0,97 * 100\% = 97\%$$

Dado este resultado en promedios generales, los usuarios pueden acceder y utilizar el sistema sin problemas el 97% del tiempo durante un mes.

En este contexto, el sistema muestra una fiabilidad solida en su primer mes de funcionamiento, ya que los problemas de carga fueron breves, y la disponibilidad general fue alta. Sin embargo, la estimación de la fiabilidad debe considerarse de manera continua a medida que el sistema continuo en funcionamiento y se enfrente a futuros desafíos.

4.2.3. Compatibilidad

El proceso de la compatibilidad, se realiza en diferentes entornos, dispositivos y navegadores, para garantizar el funcionamiento correcto y adecuado del sistema.

4.2.3.1 Plan de pruebas de navegador. El Sistema es evaluado en diferentes navegadores web. Los navegadores más utilizados por el personal son: Google Chrome, Mozilla Firefox, Microsoft Edge.

Figura 55

Compatibilidad con Google Chrome

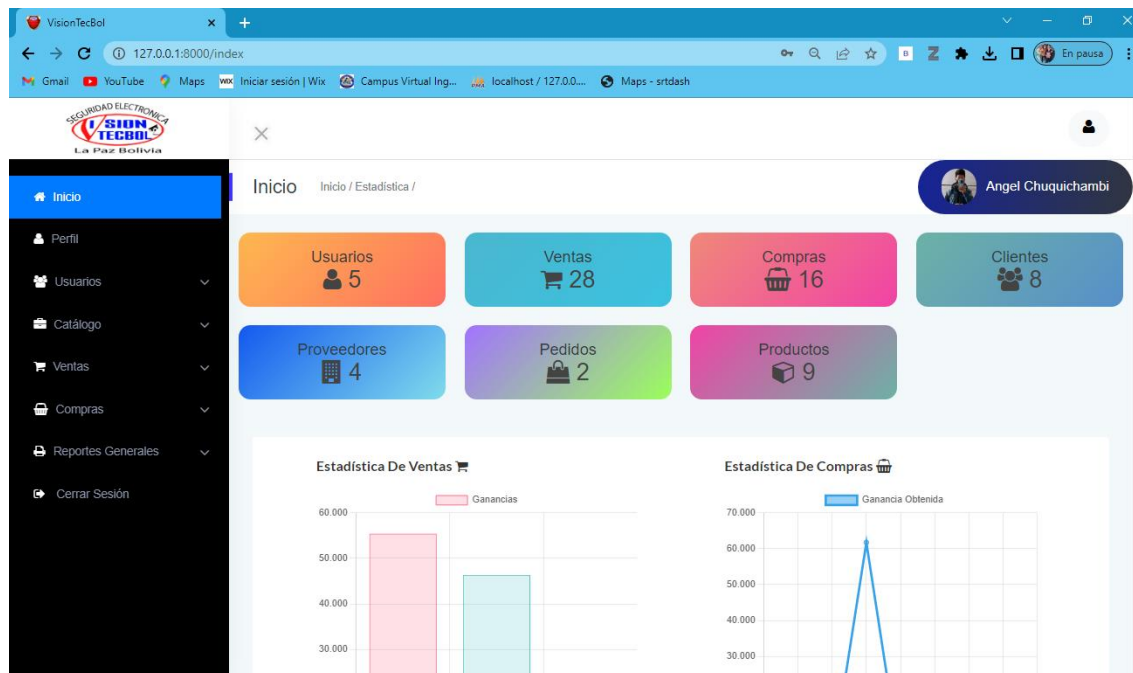


Figura 56

Compatibilidad con Firefox

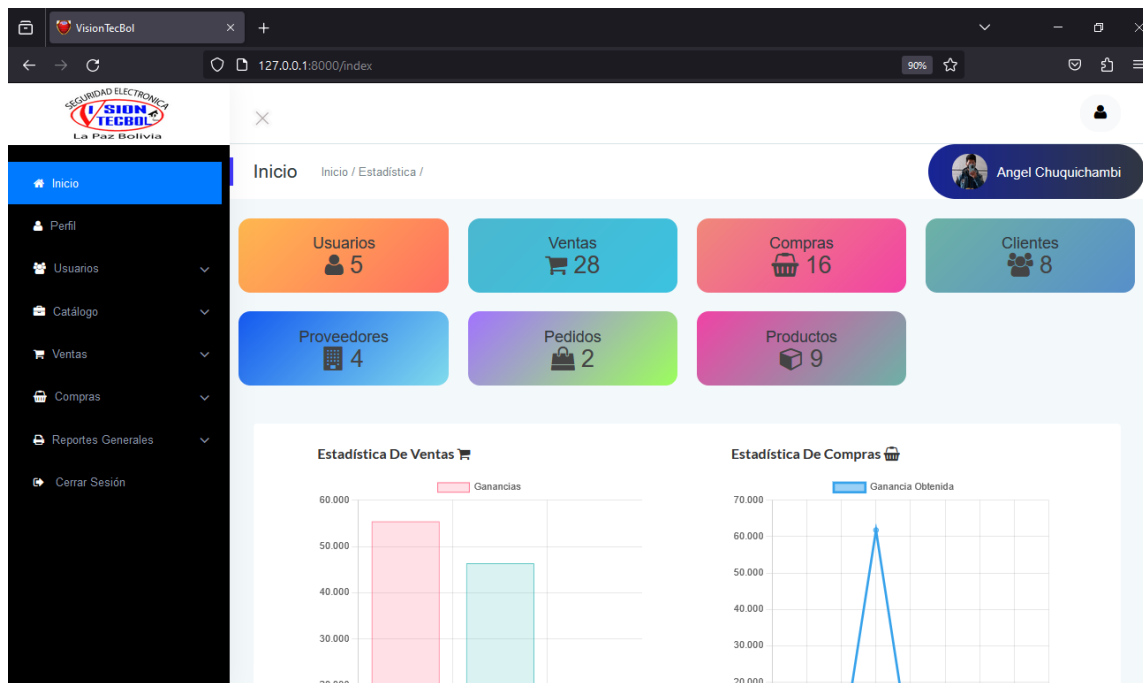
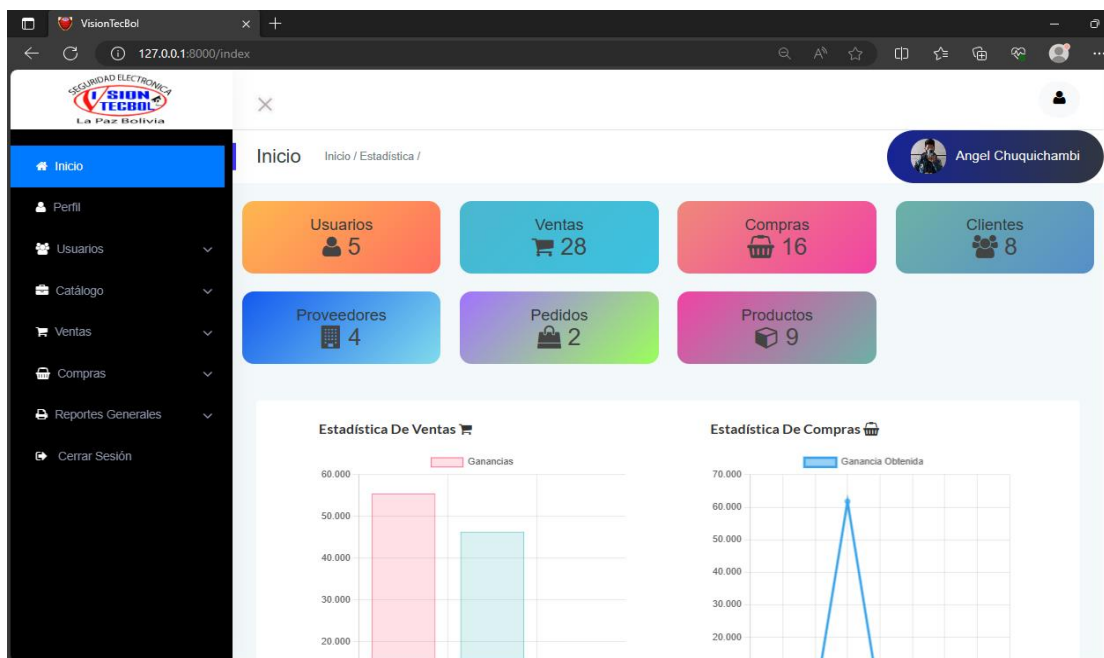


Figura 57

Compatibilidad con Microsoft Edge



4.2.3.2 Plan de pruebas de dispositivos. El sistema es evaluado en diferentes dispositivos, entre los dispositivos están las computadoras de escritorio (Windows) y teléfonos móviles (Android y iOS).

Figura 58

Prueba con dispositivo de Escritorio

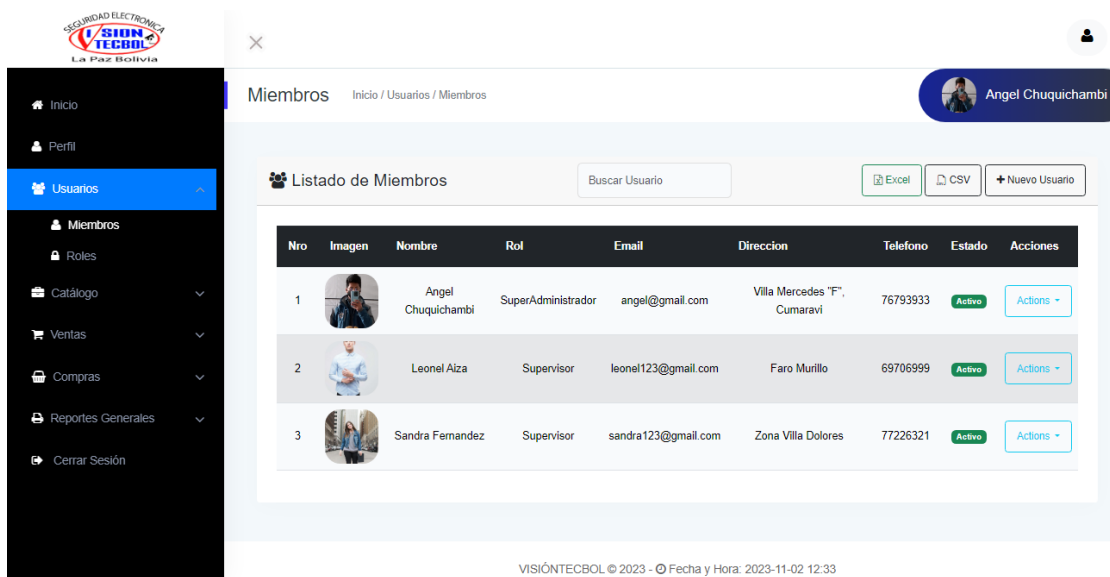
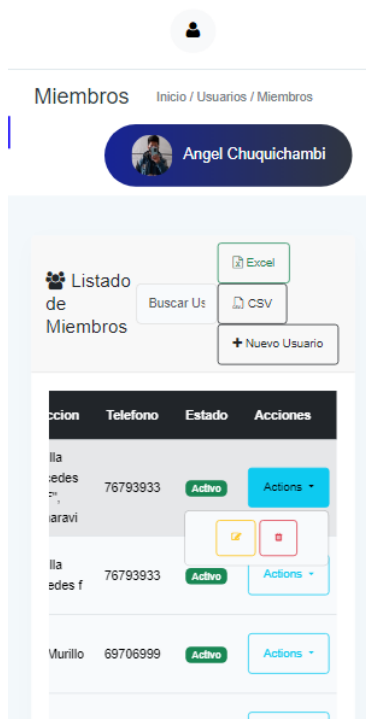


Figura 59*Prueba con dispositivo móvil***4.2.3.3 Estimación de compatibilidad del Sistema.**

- Ambientes de Pruebas: Navegadores Web
- Ponderación de 1 a 10

Tabla 17*Prueba de Navegadores Web*

Navegador	Ponderación	Descripción
Google Chrome	9	El sistema funciona sin problemas en Google Chrome.
Firefox	8	La aplicación tuvo un retraso mínimo en la carga inicial, pero funcionó correctamente en general.
Microsoft Edge	9	Funcionamiento eficiente en Microsoft Edge.
Total	26	

- Ambiente de Pruebas: Dispositivos
- Ponderación de 1 a 10

Tabla 18*Prueba de Dispositivos*

Dispositivo	Ponderación	Descripción
Computadora de Escritorio (Windows)	9	El sistema es compatible con Windows en todos los aspectos probados.
Tablet	9	El sistema se adapta de manera efectiva a las Tablet Android y se ejecutó sin problemas.
Teléfonos Móviles	9	La interfaz de usuario fue receptiva y todas las funciones funcionan sin problemas en iPhone y teléfonos Android
Total	27	

Por último, el **Tiempo de carga** del sistema en todos los entornos se mantiene dentro de los límites aceptables, garantizando una respuesta rápida, la cual tiene una ponderación de 8. Entonces, con los datos obtenidos con anterioridad, se realiza la estimación total de la compatibilidad, de la siguiente manera:

$$\text{Compatibilidad} = \frac{\text{PNT} + \text{PDT} + \text{TC}}{\text{VM}}$$

$$\text{Compatibilidad} = \frac{26 + 27 + 8}{70}$$

$$\text{Compatibilidad} = \frac{61}{70}$$

$$\text{Compatibilidad} = 0,87 * 100\% = 87\%$$

El sistema ha superado con éxito las pruebas de compatibilidad con una ponderación total del 87%.

4.2.4. Usabilidad

La usabilidad se evaluará en términos de la facilidad de uso, la eficiencia y la satisfacción del usuario al interactuar con el sistema.

Escala de Ponderación: 1 (Deficiente) - 10 (Excelente)

Tabla 19

Evaluación de Usabilidad

PREGUNTAS	RESPUESTA		PONDERACIÓN
	SI	NO	
¿La navegación es intuitiva?	X		9
¿La interfaz es clara y comprensible?	X		7
¿El diseño es consistente en toda la aplicación?	X		7
¿Las tareas se pueden realizar eficientemente?	X		9
¿La aplicación proporciona retroalimentación efectiva?	X		8
¿Es fácil de aprender a usar la aplicación?	X		8
¿La documentación y ayuda son útiles?		X	6
¿Los usuarios pueden personalizar su experiencia?		X	6
¿La aplicación es accesible para todos los usuarios?	X		10
¿El tiempo de respuesta es rápido?	X		8
Total			78 %

La evaluación de usabilidad del sistema revela una calificación total de 78%. Esta puntuación refleja una evaluación general positiva de la usabilidad, destacando aspectos como la navegación intuitiva, la claridad de la interfaz y la eficiencia en la realización de tareas.

4.2.5. Portabilidad

Por último, se procede a calcular el valor en porcentajes (%) la portabilidad de Sistema:

Tabla 20*Evaluación de la Portabilidad*

CARACTERÍSTICAS	DESCRIPCIÓN	PONDERACIÓN
Adaptabilidad	El sistema se ajusta automáticamente a los sistemas operativos más populares (Windows, macOS, Linux) y es compatible con diversos navegadores.	7
Facilidad de Instalación y Desinstalación	La instalación del sistema se realiza con un asistente sencillo y no deja archivos temporales. La desinstalación es igualmente fácil y no deja rastros.	8
Coexistencia	El sistema se integra sin problemas con otras aplicaciones en el mismo servidor web, evitando conflictos de puerto y recursos.	3
Reemplazabilidad	El sistema tiene la capacidad de actualizarse en su versión actual por una nueva versión sin problemas significativos.	9
Total		27

Por lo tanto, la puntuación de portabilidad obtenida es de 27. Ahora obtendremos esta puntuación en términos de porcentajes de la siguiente manera:

$$P = \frac{PT}{PM} * 100\%$$

Donde:

- P => Portabilidad
- PM => Puntaje Máximo (40)
- PT => Puntaje Total (27)

Entonces, reemplazando los datos, queda de la siguiente manera:

$$P = \frac{27}{40} * 100\%$$

$$P = 0,675 => 68\%$$

En conclusión, la puntuación final de la portabilidad que se obtuvo es de 68 %, esto indica un nivel de preparación que se encuentra por encima del promedio, pero que aún tiene margen de mejora.

4.2.6. Resultados Finales de Calidad

Tras un minucioso análisis de múltiples aspectos contenidos en la norma ISO 25010, se procede a llevar a cabo una estimación del puntaje definitivo, haciendo uso de los datos previamente recopilados de la manera siguiente:

Tabla 21

Resultados Finales

ELEMENTO	PONDERACIÓN
Funcionalidad	84
Fiabilidad	97
Compatibilidad	87
Usabilidad	78
Portabilidad	68
Estimación Total	82%

Se ha obtenido una puntuación del 83% en la calidad del software, lo que indica un nivel destacado de cumplimiento de los estándares de calidad según la norma ISO 25010. Esta evaluación proporciona una base óptima para futuras mejoras y la continuación de esfuerzos destinados a optimizar la calidad del software.

4.3. Estimación De Costos

La utilización de COCOMO en el presente proyecto tiene como propósito fundamental estimar el esfuerzo, el tiempo, el costo y personas requeridas para el desarrollo del sistema.

4.3.1. Estimación de costos usando el método COCOMO

Para estimar el costo, se toma en cuenta que el Sistema se asemeja a un Tipo de Proyecto "Orgánico". Las variables a, b, c, d y sus respectivos valores son los siguientes:

$$a = 2.4 \quad b = 1.05 \quad c = 2.5 \quad d = 0.38$$

Para calcular la estimación se utiliza la siguiente formula general del modelo COCOMO:

$$E = a * KLOC^b * FAE$$

Donde:

- E = esfuerzo requerido en Personas-Mes.
- a y b son variables que dependen de la complejidad del proyecto (Básico).
- KLPC = Kilo líneas de código.
- FAE = Factor de ajuste externo.

Además de las siguientes formulas:

$$C = E * \text{Salario Minimo (Expresado en Bolivianos)}$$

$$\text{Productividad} = \frac{E}{T}$$

$$T = \frac{E}{\text{Productividad}}$$

Donde:

- C = Costo estimado expresado en bolivianos.
- T = Tiempo estimado expresado meses.
- Productividad = Productividad de equipo expresado en Persona – Mes.

4.3.2. Identificación de los datos iniciales

Primero, recopilamos la información inicial necesaria para la estimación. Esta información hace referencia al tamaño estimado del software en miles de líneas de código (4290 LOC), para el caso convertimos los miles de líneas de código (LOC) a KLOC bajo la siguiente conversión:

$$LOC = 4290$$

$$KLOC = \frac{4290}{1000} = 4.290$$

$$KLOC = 4.290$$

4.3.3. Cálculo de FAE (Factor de Ajuste de Esfuerzo)

Tabla 22

Calculo de la variable FAE

CONDUCTORES DE COSTE	VALOR					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos de Software:						
Fiabilidad	0,75	0,88	1	1,15	1,4	
Tamaño de la Base de datos		0,94	1	1,08	1,16	
Complejidad	0,70	0,85	1	1,15	1,3	1,65
Atributos de Hardware:						
Restricciones de tiempo de ejecución			1	1,11	1,3	1,66
Restricciones de almacenamiento			1	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1	1,15	1,3	
Tiempo de respuesta		0,87	1	1,07	1,15	
Atributos de Personal:						
Capacidad de análisis	1,46	1,19	1	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1	0,91	0,82	
Calidad de los programadores	1,42	1,17	1	0,86	0,7	
Experiencia en el S.O.	1,21	1,1	1	0,9		
Experiencia en el lenguaje	1,14	1,07	1	0,95		
Atributos del Proyecto:						
Técnicas actualizadas de programación	1,24	1,1	1	0,91	0,82	
Utilización de herramientas de software	1,24	1,1	1	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1	1,04	1,1	
TOTAL						0,69

FAE = 1,15 * 1 * 1,15 * 1,11 * 1 * 1 * 1 * 1,07 * 0,86 * 0,91 * 0,7 * 0,9 * 1 * 0,91 * 0,91 * 1,08 =>
0,69.

Por ende, el cálculo del FAE es **0,69**.

4.3.4. Cálculo del Esfuerzo (E)

Utilizamos la fórmula que se describió anteriormente.

$$E = a * KLOC^b * FAE$$

$$E = 2.4 * (4.290)^{1,05} * 0.69$$

$E = 7.6408 \Rightarrow E = 8$ personas – mes.

4.3.5. Cálculo del Tiempo de desarrollo (T)

Se realiza el cálculo el tiempo estimado (T) en meses utilizando la relación de la variable “c” multiplicado por esfuerzo elevado a la variable “d”.

$$T = c * E^d$$

$$T = 2,5 * 7,6408^{0,38}$$

$T = 5.4141 \Rightarrow 5$ meses.

4.3.6. Cálculo de la productividad (PR)

Se elabora el cálculo para la productividad (PR) en LCD/personas – mes, teniendo en cuenta la siguiente formula:

$$PR = LCD/E$$

$$PR = \frac{4290}{7.6408} \text{ personas – mes}$$

$PR = 561.459$ personas – mes.

4.3.7. Cálculo del personal promedio (P)

El cálculo del personal se realiza mediante la siguiente formula:

$$P = \frac{E}{T}$$

$$P = \frac{7.6408 \text{ personas – mes}}{5.4141 \text{ mes}}$$

$P = 1.5712$ personas $\Rightarrow 2$

4.3.8. Cálculo del costo del proyecto

Para realizar el costo total, primero se calcula el costo por mes para luego finalmente obtener el costo total del proyecto, tomando en cuenta el tiempo de desarrollo que se calculó anteriormente que equivale a 8 meses.

➤ Costo por Mes

$$CM = P * \text{Salario Minimo Nacional (En bolivianos)}$$

$$CM = 2 * 2500 \Rightarrow 5000 \text{ Bs.}$$

- Costo Total del Proyecto en bolivianos (Bs)

$$CT = CM * T$$

$$CT = 5000 * 5 = 25000 \text{ Bs}$$

- Costo Total del Proyecto en dólares (\$)

Tomando en cuenta que la tasa de cambio es:

$$\mathbf{\$1 = Bs 6.90 / USD}$$

Conversión de Unidades

$$\text{Dolares (\$)} = \frac{\text{Bolivianos}}{\text{Tasa de Cambio}}$$

$$\text{Dolares (\$)} = \frac{25.000 \text{ Bs}}{6.90 \text{ Bs/USD}}$$

$$\text{Dolares (\$)} = 3623.19 \Rightarrow \$ 3623.19$$

En conclusión, con los datos obtenidos se requieren 2 personas durante 5 meses, con un salario básico de Bs. 2500 por persona. El Costo por mes es de Bs. 5000, lo que implica que el Costo total estimado del Sistema en bolivianos es Bs. 25000 y en dólares equivalente a \$ 3623,19.

4.4. Seguridad Basado En La ISO 27001

Para garantizar la protección integral de los datos y el cumplimiento de las mejores prácticas de seguridad, se ha adoptado una estrategia basada en la norma ISO 27001.

4.4.1 Seguridad Física

La estrategia de seguridad física aborda la protección de los recursos físicos que respaldan el sistema:

- **Acceso a las Instalaciones del Centro de Datos:** Hostinger nos garantiza la seguridad física de las instalaciones de los centros de datos donde se aloja el sistema.

4.4.2 Seguridad Lógica

La estrategia de seguridad lógica se centra en proteger la información y los datos en línea.

- **Autenticación de Usuarios:** Se ha implementado controles de acceso lógico para garantizar que solo los usuarios autorizados tengan acceso al sistema. Esto incluye autenticación de usuarios y contraseñas seguras.
- **Cifrado de Datos:** La confidencialidad de los datos transmitidos y almacenados se protege mediante el cifrado. Utilizando HTTPS y certificados SSL para garantizar la seguridad de la información.
- **Acceso a Documentación Confidencial:** La información y registros confidenciales dentro del sistema, relacionados con la institución están seguros y protegidos contra el acceso no autorizado.
- **Respaldo y Recuperación de Datos (Backup's):** Se crean copias de seguridad de la base de datos periódicamente y procedimientos de recuperación para garantizar la disponibilidad de los datos en caso de pérdida.
- **Contraseñas Seguras:** Se exige al usuario establecer contraseñas fuertes y su cambio regular, con el fin de prevenir el uso de contraseñas débiles.

4.5. Pruebas De Rendimiento De Software

Estas pruebas son esenciales para garantizar que una aplicación funcione de manera eficiente bajo diversas condiciones de carga. Para el sistema, se realiza pruebas utilizando el siguiente método:

4.5.1. Pruebas de Estrés

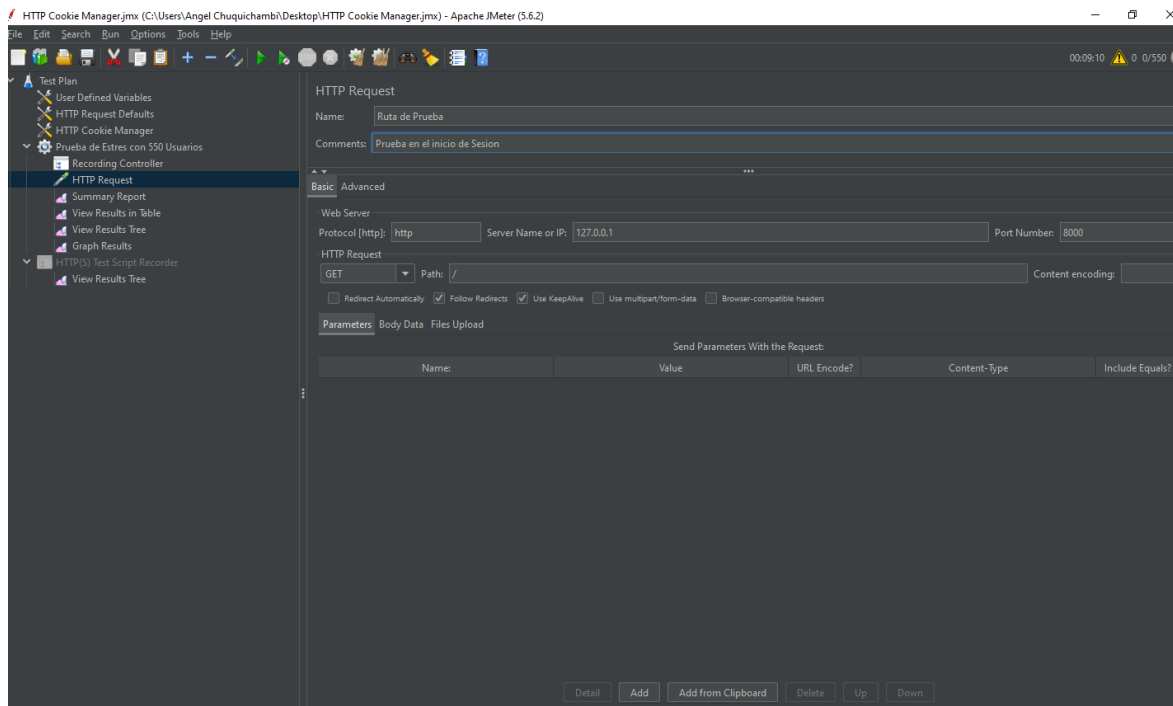
Este método de pruebas maneja condiciones extremas de carga y determinar su punto de ruptura. Para el sistema se utiliza la siguiente herramienta:

4.5.2. Herramienta Apache JMeter

Para la utilización de esta herramienta, se configura los siguientes parámetros para definir la configuración básica de la solicitud HTTP, incluyendo la URL de la página a la cual se le va realizar la prueba.

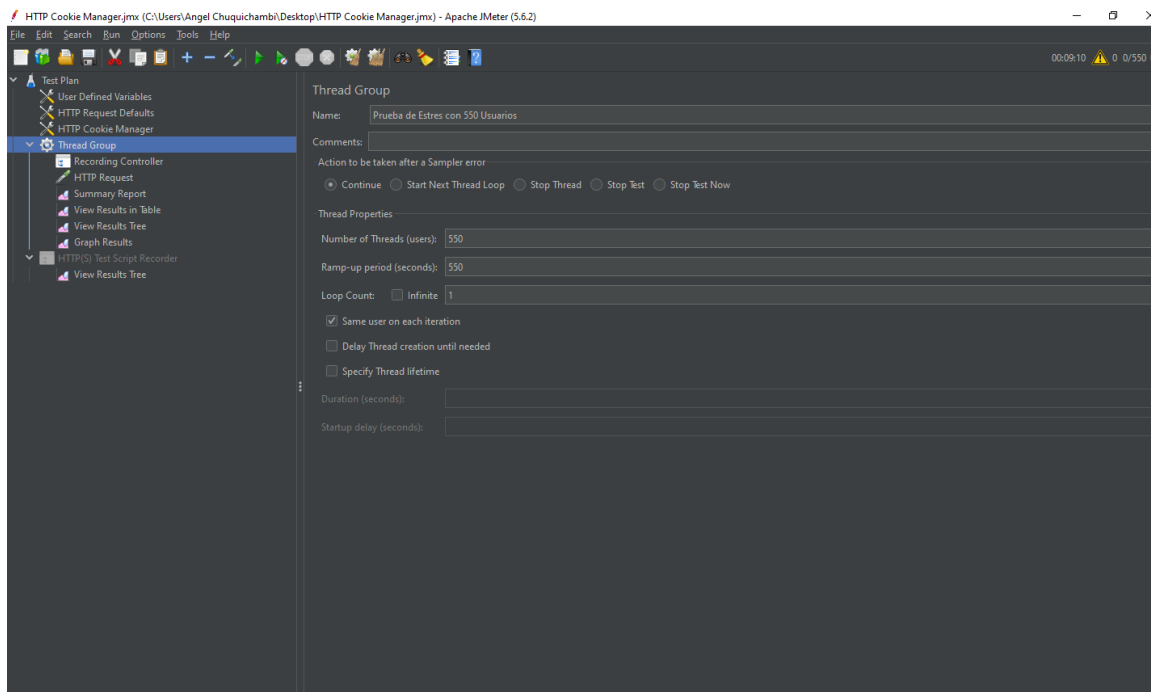
Figura 60

Definición de Parámetros

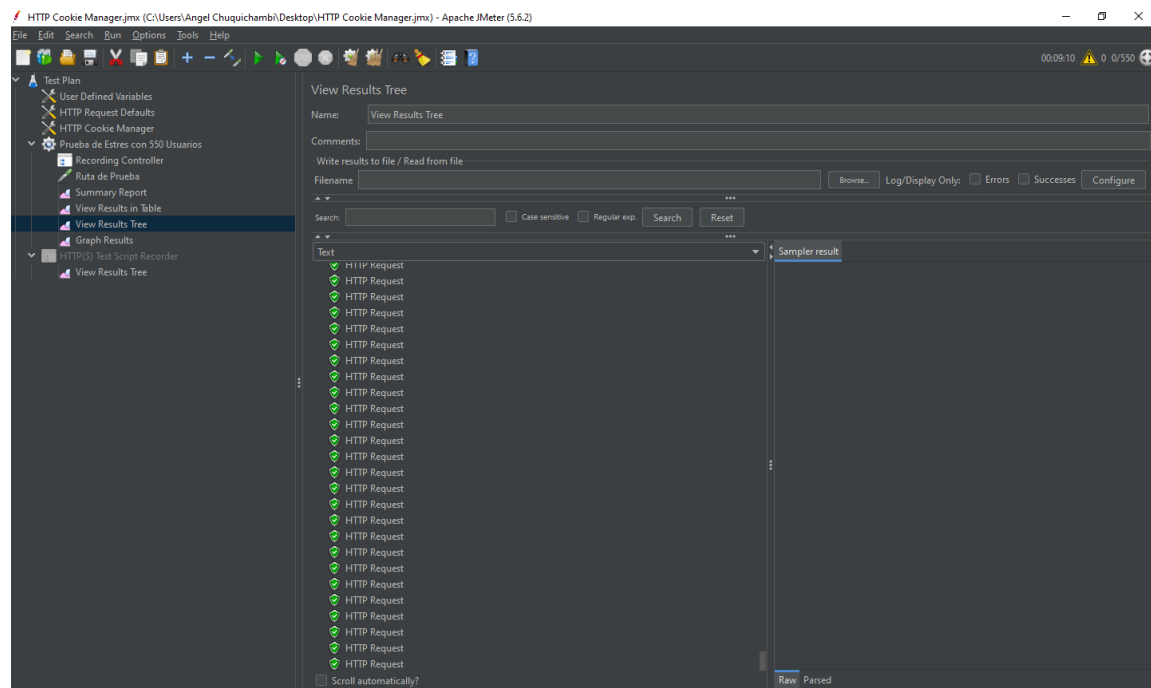


Seguidamente se configura el **Thread Group** donde, a continuación, se establece los siguientes parámetros

- **Número de Hilos (Usuarios):** Establecemos el número de usuarios simulados, para el caso se establece 550.
- **Periodo de Ramp-Up:** Establecemos el tiempo en segundos que JMeter debe esperar antes de lanzar todos los hilos, para el caso se establece 550.
- **Número de Ciclos (Loop Count):** Configuramos el número de veces que cada hilo ejecutará la prueba, que es 1.

Figura 61*Configuración del Thread Group*

Después de establecer todos los parámetros, se procede a ejecutar la prueba.

Figura 62*Prueba en Ejecución*

Finalizado la prueba, se observa los resultados en los siguientes elementos:

Figura 63

Resultados con Summary Report

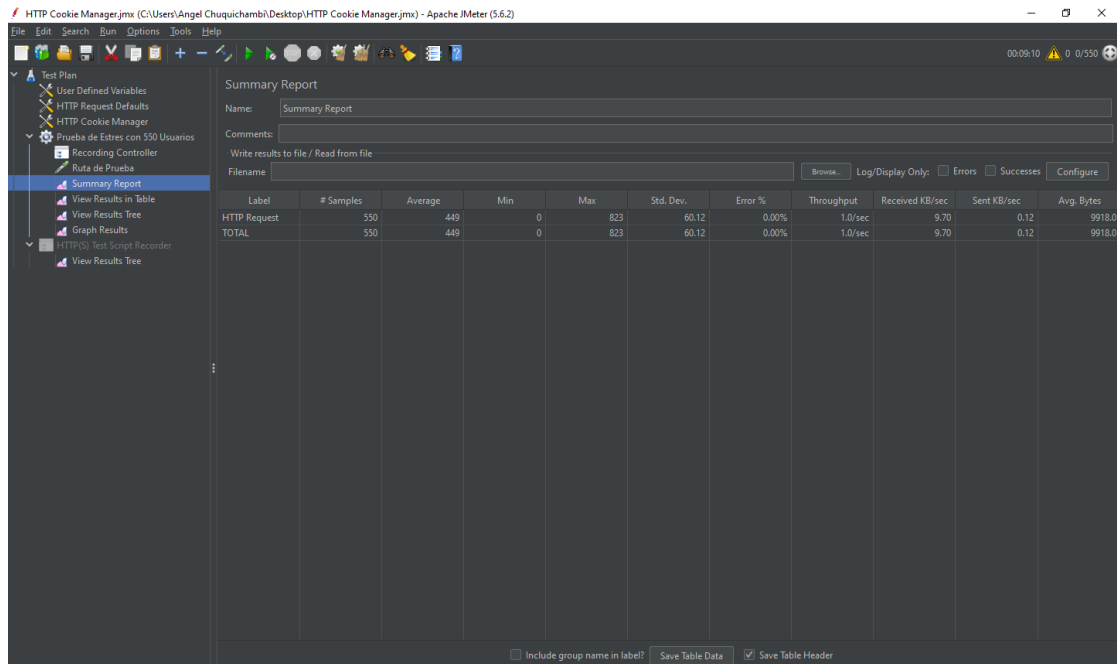


Figura 64

Resultados con View Results in Table

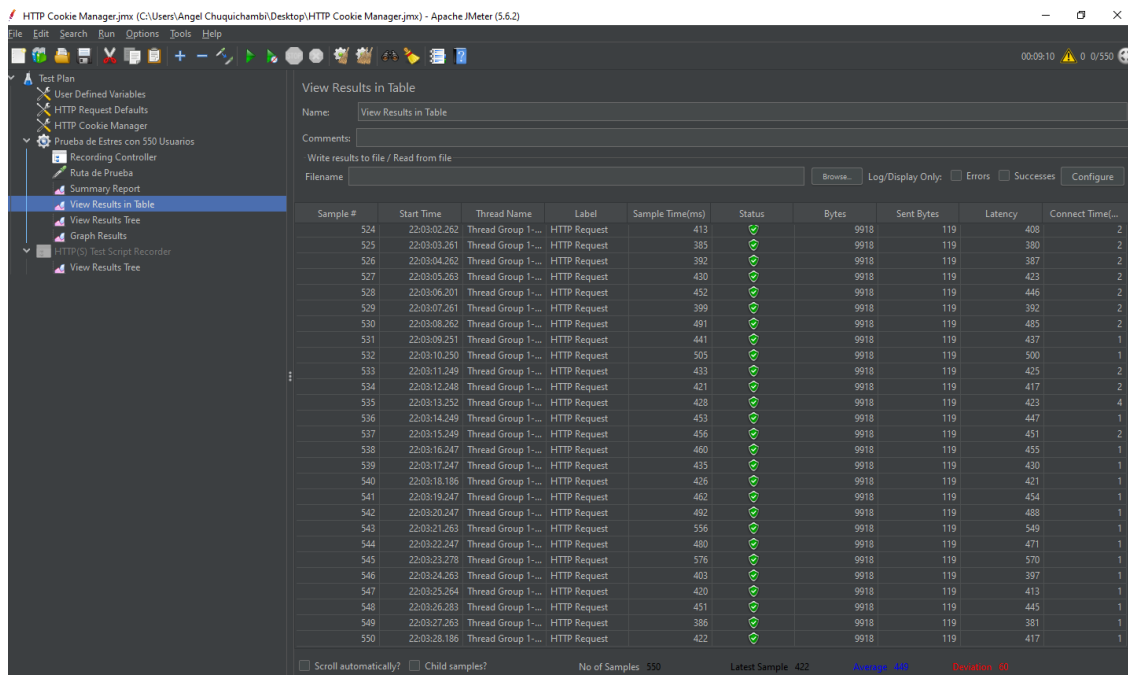
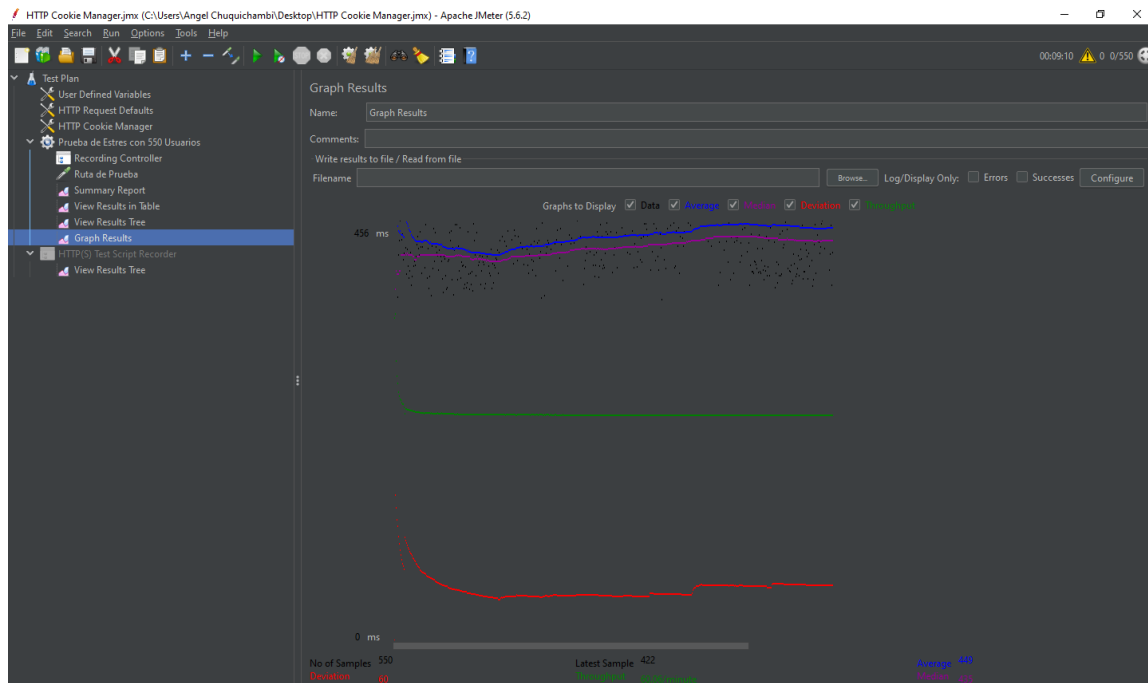


Figura 65

Resultado con Graph Results



- **View Results in Table:** Se analiza los resultados en la tabla para identificar solicitudes individuales que pudieron haber fallado, lo cual se puede apreciar en la siguiente figura que todas las peticiones tuvieron una respuesta exitosa.
- **Summary Report:** Revisamos el informe resumido para obtener una visión general del rendimiento, estos son el número total de solicitudes que son 550, tiempos medios de respuesta y tasas de error con un total de 0.00%.
- **Graph Results:** Observamos los gráficos para identificar tendencias y patrones en el rendimiento del sistema a lo largo del tiempo. Lo cual, para las peticiones solicitadas se aprecia que se tiene una estabilidad en las peticiones HTTP REQUEST GET.

CAPÍTULO V

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

Se ha llegado a cumplir de manera exitosa con los objetivos planteados en el presente proyecto.

5.1 Conclusiones

- **Automatización de Procesos Manuales:** El sistema ha permitido automatizar tareas previamente manuales, reduciendo drásticamente el tiempo empleado en la gestión de ventas y compras.
- **Organización y Accesibilidad de Información:** La información ahora se encuentra organizada de manera centralizada y es fácilmente accesible, lo que ha eliminado la pérdida de tiempo asociada con la búsqueda de datos dispersos.
- **Información Actualizada:** La disponibilidad de información actualizada sobre precios, disponibilidad y existencias de productos ha sido un avance significativo.
- **Reducción de Riesgos:** El sistema ha reducido los riesgos asociados con la pérdida de información y la toma de decisiones incorrectas.

5.2 Recomendaciones

- **Capacitación Continua del Personal:** Proporcionar capacitación regular al personal para garantizar un uso efectivo del sistema y fomentar una adopción completa de sus funcionalidades.
- **Mantenimiento y Actualización:** Establecer un programa de mantenimiento periódico para garantizar el funcionamiento óptimo del sistema y su alineación con las necesidades en evolución de la institución.
- **Seguridad y Acceso a la Información:** Para garantizar el acceso al Sistema es importante establecer contraseñas complejas y estas actualizarlas cada cierto tiempo.
- **Herramientas:** Se recomienda utilizar herramientas similares en proyectos venideros para facilitar la evolución del proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- Admin. (14 de diciembre de 2020). *Quassure*. Retrieved 5 de junio de 2024, from Testing en Modelo V: <https://quassure.com/testing-en-modelo-v/>
- Carrion, J. (2017). Diferencia entre dato información y conocimiento.
- Chen, C. (21 de mayo de 2019). *Significado de Sistema de información*. Retrieved 7 de septiembre de 2023, from Significados: <https://www.significados.com/sistema-de-informacion/#:~:text=Caracter%C3%ADsticas%20de%20un%20sistema%20de%20informaci%C3%B3n&text=Un%20sistema%20de%20informaci%C3%B3n%20se%20destaca%20por%20su%20dise%C3%B1o%2C%20facilidad,anonimato%20en%20informaciones%2>
- Cornejo, A. (29 de enero de 2015). *ISO 12207*. Retrieved 3 de julio de 2023, from Normas y Estándares en Proyectos T.I.: <https://normasyestandaresproyectosti.wordpress.com/2015/01/29/iso-12207/>
- DARIAS PÉREZ, S. (7 de enero de 2022). *POWER BI: LA INTELIGENCIA DE NEGOCIO QUE NECESITA TODA ORGANIZACIÓN*. Retrieved 3 de julio de 2023, from intelequia: <https://intelequia.com/blog/post/power-bi-la-inteligencia-de-negocio-que-necesita-toda-organizaci%C3%B3n>
- DELSOL, S. (2023). *Gestión de compras: Qué es y qué etapas tiene*. Retrieved 19 de septiembre de 2023, from Software DELSOL: <https://www.sdelsol.com/blog/facturacion/gestion-de-compras-que-es/>
- Equipo editorial, E. (1 de octubre de 2020). *Dato*. Retrieved 20 de septiembre de 2023, from concepto: <https://concepto.de/dato/>
- Eseme, S. (23 de agosto de 2023). *Novedades de Laravel 9: Una Inmersión Profunda en la Próxima Gran Versión*. Retrieved 23 de agosto de 2023, from Kingsta : [https://kinsta.com/es/blog/laravel-9/#:~:text=instalar%20Laravel%209-,%C2%BFQu%C3%A9%20es%20Laravel%3F,modelo%2Dvista%2Dcontrolador\).](https://kinsta.com/es/blog/laravel-9/#:~:text=instalar%20Laravel%209-,%C2%BFQu%C3%A9%20es%20Laravel%3F,modelo%2Dvista%2Dcontrolador).)

- Etecé, E. (22 de octubre de 2021). *Sistema*. Retrieved 1 de septiembre de 2023, from concepto: <https://concepto.de/sistema/>
- Flores, F. (22 de julio de 2022). *Qué es Visual Studio Code y qué ventajas ofrece*. Retrieved 27 de junio de 2023, from OpenWebinars.
- GlobalSuite Solutions. (20 de marzo de 2023). *¿Qué es la norma ISO 27001 y para qué sirve?* Retrieved 27 de junio de 2023, from <https://www.globalsuitesolutions.com/es/que-es-la-norma-iso-27001-y-para-que-sirve/#:~:text=La%20norma%20ISO%2027001%20es,y%20disponibilidad%20de%20a%20informaci%C3%B3n>.
- hiberus. (10 de enero de 2023). *Cómo automatizar pruebas funcionales con Selenium*. Retrieved 25 de mayo de 2024, from hiberus blog: <https://www.hiberus.com/crecemos-contigo/como-automatizar-pruebas-funcionales-con-selenium/>
- Lapiedra Alcamí, R., Devece Carañana, C., & Guiral Herrando, J. (2011). *Introducción a la gestión de sistemas de información en la empresa*. Castelló de la Plana : Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions, 2011.
- Latorre, M. (2018). *Historia de las web, 1.0, 2.0, 3.0 y 4.0*. Universidad Marcelino Champagnat, 1.
- Llerena Yamba, L. D. (2019). SISTEMA E-COMMERCE PARA LA GESTIÓN DE VENTAS EN POWER PC JR. UBICADA EN LA CIUDAD DE QUITO. *Tesis de Grado*. Quito, Ecuador: Universidad Israel.
- LMU – Ludwig-Maximilians-Universität München. (s.f.). *UWE – UML-based Web Engineering*. Retrieved 2 de mayo de 2024, from <https://uwe.pst.ifi.lmu.de/teachingTutorialPresentationSpanish.html>
- Londoño, P. (21 de enero de 2023). *Qué son las aplicaciones web y 8 ejemplos*. Retrieved 12 de septiembre de 2023, from HubSpot: <https://blog.hubspot.es/website/que-es-aplicacion-web>

- Losoviz, L. (18 de septiembre de 2022). *10 Herramientas de Prueba de PHP para Plugins de WordPress*. Retrieved 27 de junio de 2023, from Kinsta: <https://kinsta.com/es/blog/herramientas-de-prueba-php/>
- Mamani, S. M. (2019). *APLICACIÓN DE SCRUM Y UML PARA EL DESARROLLO DE UN SISTEMA DE VENTAS*. Cochabamba, Bolivia: Universidad Mayor de San Simon.
- Marujita. (1 de febrero de 2023). *Tipos De Información*. Retrieved 20 de septiembre de 2023, from Muy Tecnológicos: <https://muytecnologicos.com/diccionario-tecnologico/tipos-de-informacion>
- Ormeño Rojas, N. F. (15 de mayo de 2019). *ISO 25010 y el desarrollo de software*. Retrieved 29 de junio de 2023, from Medium: <https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341>
- Palacios, D. (20 de enero de 2023). *Gestión de ventas: qué es, por qué importa y cómo implementarla*. Retrieved 19 de septiembre de 2023, from HubSpot: <https://blog.hubspot.es/sales/gestion-de-ventas>
- Peiró, R. (05 de mayo de 2020). *Sistema de Información*. Retrieved 1 de septiembre de 2023, from Economipedia: <https://economipedia.com/definiciones/sistema-de-informacion.html>
- Pressman, R. S. (2010). *Ingeniería del software un enfoque práctico* (Séptima ed.). México: McGraw-Hill.
- Pressman, R. S., & Lowe, D. (2014). *Web Engineering: A Practitioner's Approach*. McGraw-Hill Education.
- Quiroa, M. (1 de enero de 2021). *Gestión de compras*. Retrieved 19 de septiembre de 2023, from economipedia: <https://economipedia.com/definiciones/gestion-de-compras.html>
- Quiroa, M. (1 de abril de 2021). *Gestión de ventas*. Retrieved 19 de septiembre de 2023, from economipedia: <https://economipedia.com/definiciones/gestion-de-ventas.html>

Romero Carhuamaca, A. J. (2023). SISTEMA DE INFORMACION WEB PARA MEJORAR LA GESTION DE VENTAS E INVENTARIO EN LA FERRETERIA R&C . *Tesis de Grado*. Huancayo, Peru: Universidad Nacional del Centro del Perú.

Shethi, S. (25 de septiembre de 2023). *Apache Jmeter: Todo lo que necesita saber*. Retrieved 3 de noviembre de 2023, from GEEKFLARE: <https://geekflare.com/es/apache-jmeter-guide/>

Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE* (Novena ed.). México: PEARSON EDUCACIÓN.

UNIVERSIDAD DE SEVILLA. (s.f.). *PLANIFICACIÓN Y GESTIÓN DE PROYECTOS INFORMÁTICOS*. Retrieved 8 de mayo de 2024, from <http://www.lsi.us.es/docencia/get.php?id=7350>

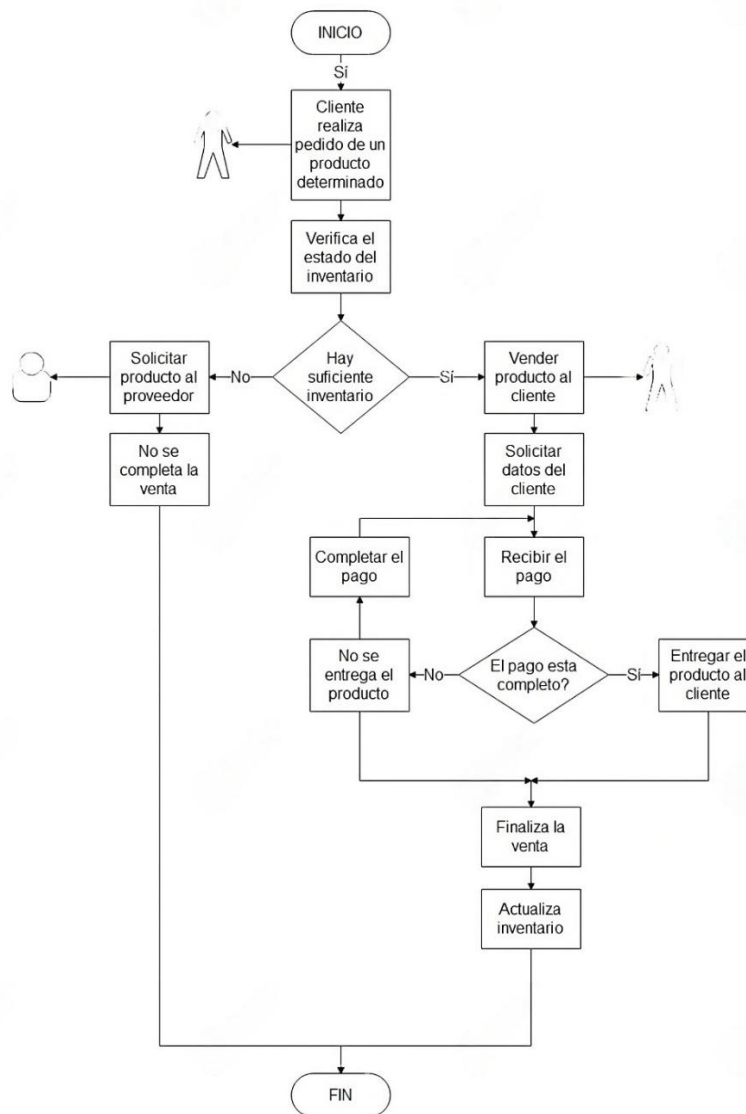
Vértice, E. (2009). *Técnicas avanzadas de diseño web*. Editorial Vértice, 2009.

ANEXOS

ANEXOS

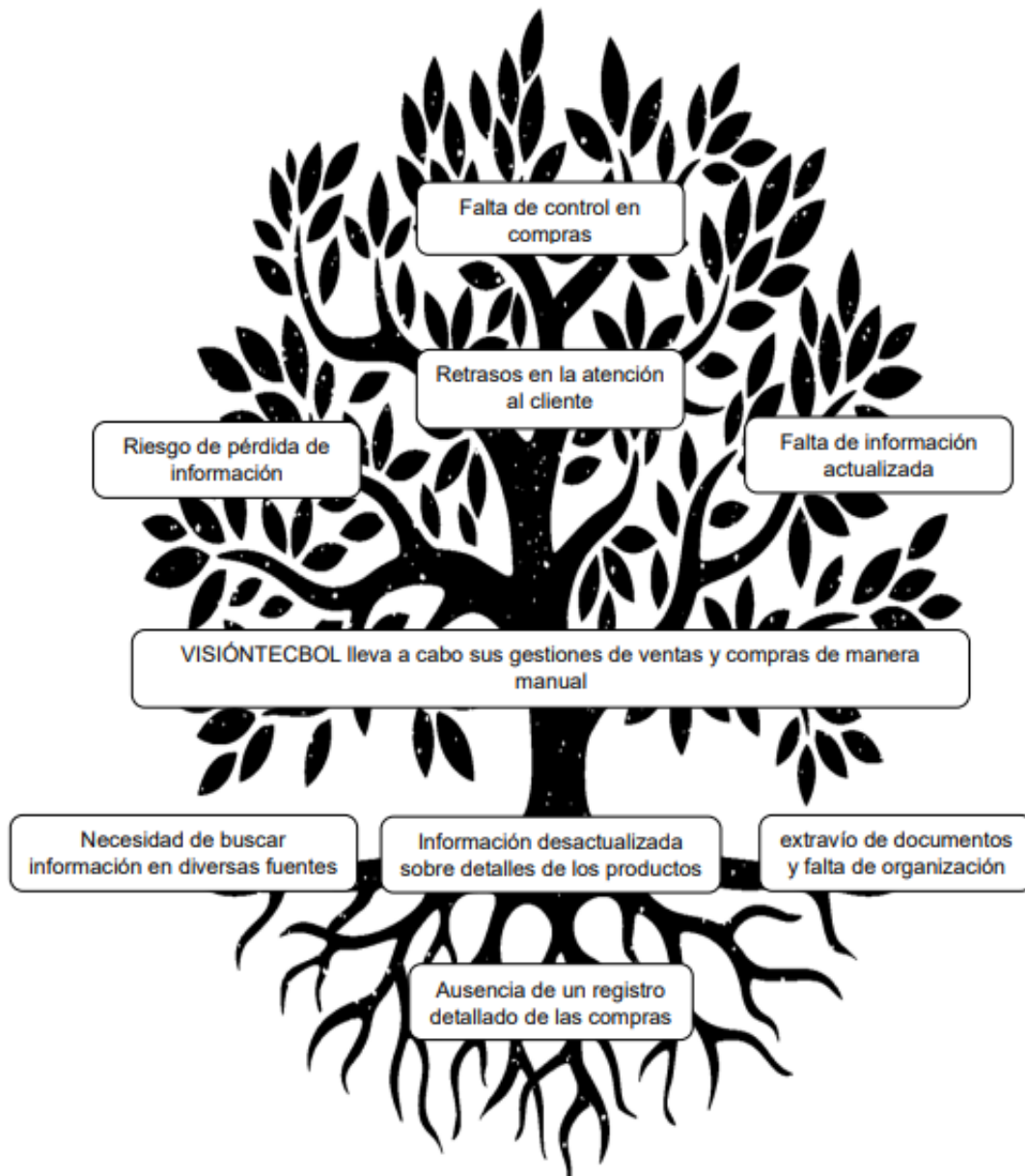
ANEXO A

DIAGRAMA DE LA SITUACIÓN ACTUAL



Nota. Elaboración Propia

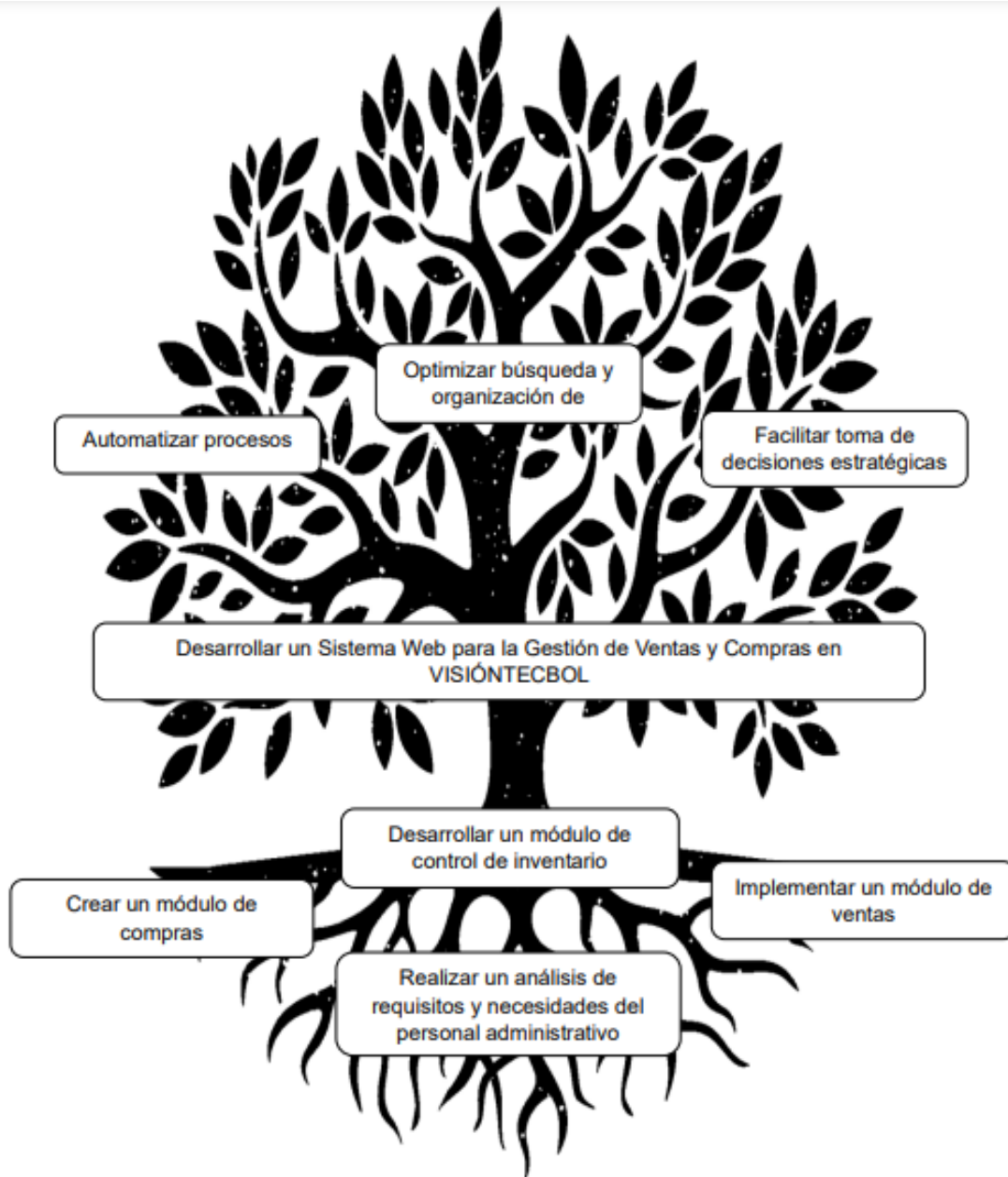
ANEXO B
ARBOL DE PROBLEMAS



Nota. Elaboración Propia

ANEXO C

ÁRBOL DE OBJETIVOS








Nota. Elaboración Propia

ANEXO D

Se realizó el estudio de factibilidades centrados específicamente en el HARDWARE y SOFTWARE, que a continuación se detallan los siguientes:




ESTUDIO DE FACTIBILIDADES (FACTIBILIDAD TÉCNICA EN SOFTWARE)

Aspecto Técnico	Descripción	Imagen	Viabilidad
Sistema Operativo	Uso de un sistema operativo Windows 10, compatible con las tecnologías de desarrollo,		Alta
Plataforma de Desarrollo	Selección de entorno de desarrollo Visual Studio Code para la codificación y depuración del sistema.		Alta
Lenguajes de Programación	Elección de lenguajes de programación utilizados para el desarrollo web, entre estas están: JavaScript, PHP, JQuery.		Alta
Gestor de Bases de Datos	Para el almacenamiento y recuperación de datos del sistema, se opta por utilizar la herramienta de gestor de base de datos relacional MYSQL.		Alta
Framework y Bibliotecas	Para agilizar el desarrollo y mantener la calidad del código, se toma la decisión de trabajar con Framework Laravel Livewire, además de utilizar bibliotecas como DomPDF, Laravel-Excel, Captcha, Laravel-Qr.		Alta

Nota. Elaboración Propia

ANEXO E

ESTUDIO DE FACTIBILIDADES (FACTIBILIDAD TÉCNICA EN HARDWARE)

Aspecto Técnico	Descripción	Imagen	Viabilidad
Redes	La institución dispone de routers para garantizar la conexión a internet y conectividad interna y externa del sistema.		Media
Computadora de Escritorio	La institución cuenta con computadoras de escritorio, laptops y teléfonos móviles, para garantizar la accesibilidad del sistema mediante el internet.		Alta
Seguridad	La institución cuenta con medidas de seguridad necesarias para el sistema, como ser cerraduras y cámaras de Seguridad para proteger los equipos contra robos o daños.		Media

Nota. Elaboración Propia

ANEXO F

VISIÓNTECBOL cuenta con todos los recursos necesarios para desarrollar, implementar y operar el sistema en línea. Estos recursos incluyen computadoras, enrutadores conectados a Internet y cámaras de seguridad. Como ya están disponibles, no hay costos adicionales asociados con la compra de estos recursos.

ESTUDIO DE FACTIBILIDADES (FACTIBILIDAD ECONÓMICA EN SOFTWARE)

Aspecto	Descripción	Costo Estimado
Licencia de Windows 10	Licencia que vino incluido en la adquisición de las Computadoras para el sistema operativo Windows 10.	Gratuito
Visual Studio Code	Utilización de editor de código fuente Visual Studio Code para el desarrollo del sistema.	Gratuito
Framework Laravel Livewire	Uso del framework Laravel Livewire para el desarrollo de componentes interactivos en tiempo real.	Gratuito
Lenguajes de Programación	Empleo de lenguajes de programación PHP, JavaScript para el desarrollo del sistema.	Gratuito
Bibliotecas Adicionales (DomPDF, Laravel-Excel, Captcha)	Utilización de bibliotecas adicionales como DomPDF para la generación de PDFs, Laravel-Excel para la importación y exportación de datos y Captcha para la seguridad del sistema.	Gratuito
Gestor de Base de Datos MySQL	Utilización de MySQL como gestor de base de datos para almacenar y gestionar los datos del sistema.	Gratuito

Nota. Elaboración Propia

ANEXO G

ESTUDIO DE FACTIBILIDADES (FACTIBILIDAD ECONÓMICA EN HARDWARE)

Aspecto	Descripción	Cantidad
Computadora de Escritorio	Equipos de escritorio utilizado para el desarrollo, prueba y uso diario del sistema. Este cumple con los requisitos de hardware y software necesarios para ejecutar eficientemente el sistema.	1
Red a Internet	Dispositivo de red existente en la institución que proporciona conectividad a internet para el equipo computacional de desarrollo.	2
Seguridad con Cámaras de Vigilancia	Dispositivos de seguridad instalados en áreas clave para monitorear y grabar actividades, garantizando la protección física de los equipos.	4

Nota. Elaboración Propia

ANEXO H

MANUALES





Zona Faro Murillo
Av. Panorámica, Calle 13 "B"
Contactos: 79115564 -79115565

MANUAL DE FUNCIONES DEL USUARIO

**“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE
VENTAS Y COMPRAS DE DISPOSITIVOS DE SEGURIDAD”**

LA PAZ – BOLIVIA

2024

ÍNDICE GENERAL

1. INTRODUCCIÓN.....	1
2. VISTA INICIAR SESION.....	1
3. PANEL GENERA DEL SISTEMA.....	1
4. ADMINISTRACIÓN DE USUARIOS	2
5. VISTA DE ROLES	2
5.1 CREAR UN NUEVO ROL.....	3
6. VISTA DE CATEGORÍAS	3
6.1 CREAR UNA NUEVA CATEGORÍA.....	4
7. VISTA DE PRODUCTOS.....	4
7.1 CREAR UN NUEVO PRODUCTO	5
8. VISTA DE CLIENTES.....	6
8.1 CREAR UN NUEVO CLIENTE	7
9. VISTA HISTORIAL DE VENTAS	7
9.1 LECTURA DE VENTA	8
10. VISTA GENERAR VENTA.....	9
11. VISTA DE PEDIDOS.....	9
12. VISTA PROVEEDORES.....	10
12.1 CREAR UN NUEVO PROVEEDOR	10
13. VISTA DE COMPRAS	11
13.1 CREAR UNA NUEVA COMPRA.....	11
14. VISTA PARA GENERAR REPORTE.....	12
14.1 REPORTE DE VENTAS	12
14.2 REPORTE DE COMPRAS.....	12
15. VISTA DE REPORTE PERSONALIZADO	13
16. IMPRESIÓN DE REPORTE.....	13
17. VISTA HISTORIAL DE ACTIVIDADES	13
18. CERRAR SESIÓN.....	14

1. INTRODUCCIÓN

Este manual tiene como objetivo describir las funciones y procedimientos asociados al uso del sistema. Se detallan los usuarios, roles involucrados y las funcionalidades que cada rol puede ejecutar, proporcionando una guía clara para la operación eficiente del sistema.

2. VISTA INICIAR SESION

Pasos a seguir:

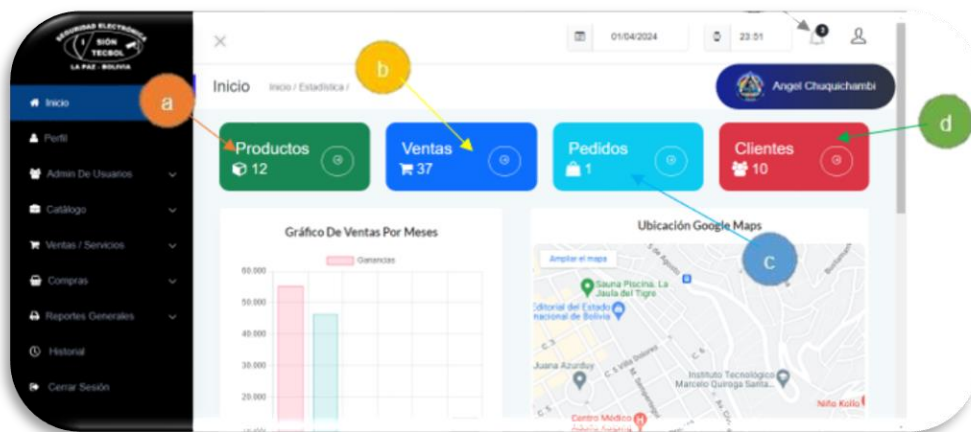
- Ingresar los datos registrados por el administrador
- Ingresar la contraseña
- Con este botón podemos ingresar al sistema



3. PANEL GENERA DEL SISTEMA

Datos Generales

- Podemos ver todos los productos restantes
- Podemos ver todas las ventas realizadas
- Se puede observar los pedidos realizados
- En esta opción se puede observar a todos los clientes
- En esta opción podemos observar todas las notificaciones

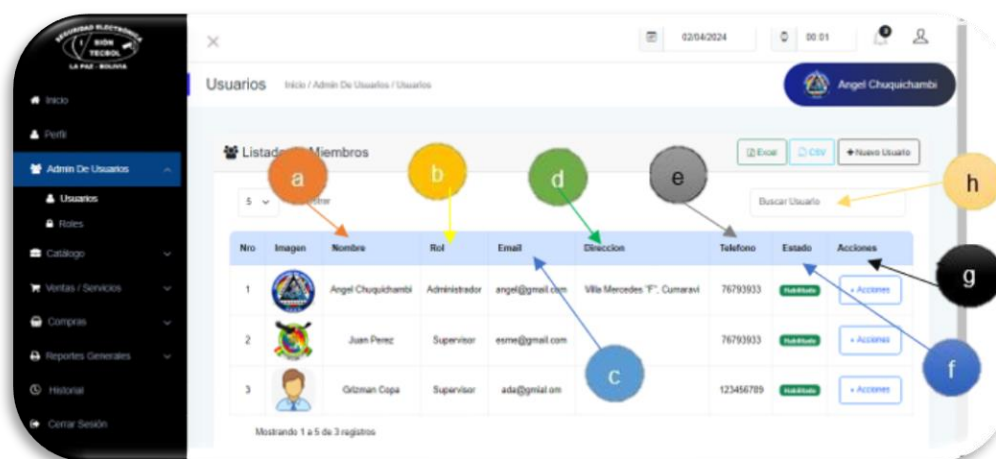


4. ADMINISTRACIÓN DE USUARIOS

Permite a los administradores crear, editar y eliminar usuarios, asignar roles y gestionar permisos.

Acciones y Datos Generales:

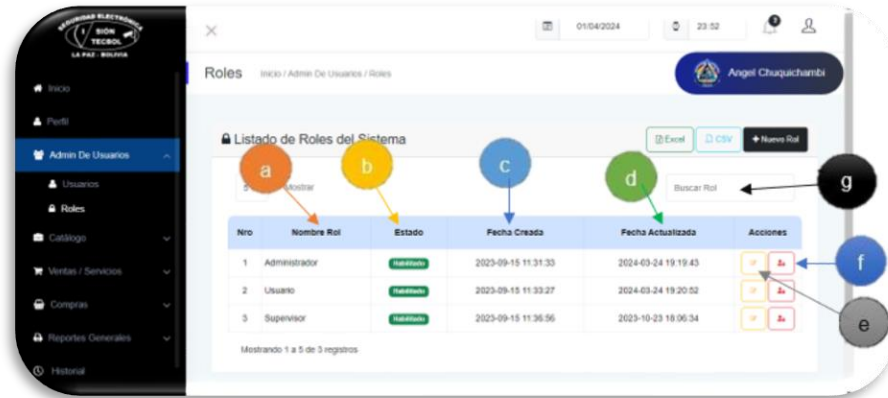
- Podemos ver todos los nombres de los usuarios
- Podemos ver con que rol esta designado cada usuario
- Se puede observar el email de cada usuario
- En esta opción se puede observar la dirección de cada usuario
- En esta opción podemos observar los contactos
- Se puede ver el estado de cada usuario
- Podemos analizar todas las acciones realizadas
- En esta opción podemos buscar de forma más directa a los usuarios



5. VISTA DE ROLES

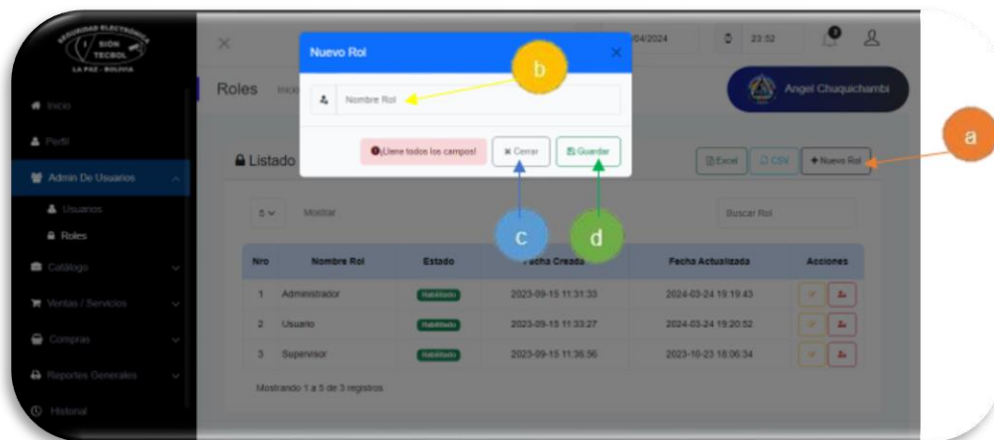
Datos generales y Acciones:

- Podemos observar los nombres de rol
- Se puede ver el estado
- Se puede observar la fecha creada
- Se puede observar la fecha actualizada
- En esta opción podemos editar
- En esta opción podemos dar de baja un rol del sistema
- En esta opción podemos buscar de forma más directa



5.1 Crear un Nuevo Rol

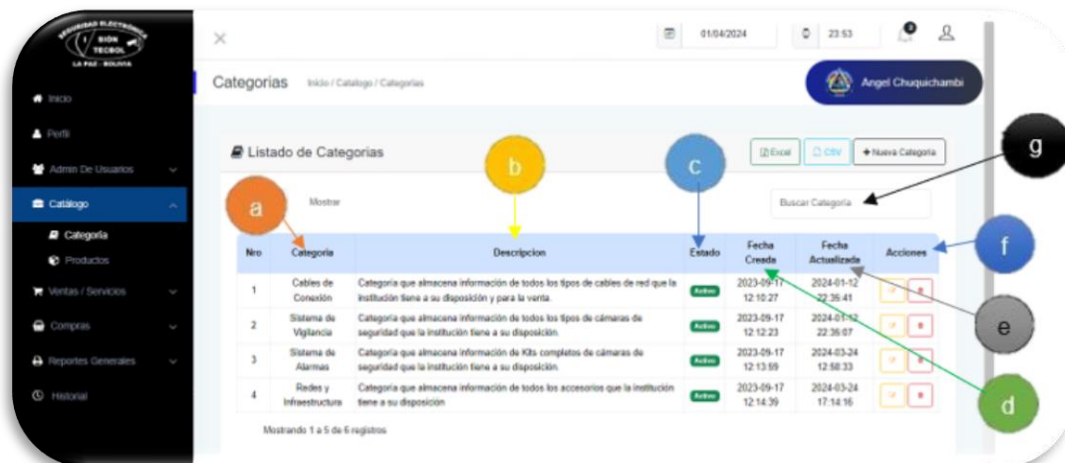
- Esta opción nos ayuda a crear un nuevo rol
- Ponemos el nombre del rol
- La opción nos facilita cerrar la ventana
- La opción nos facilita guardar el nombre del rol



6. VISTA DE CATEGORÍAS

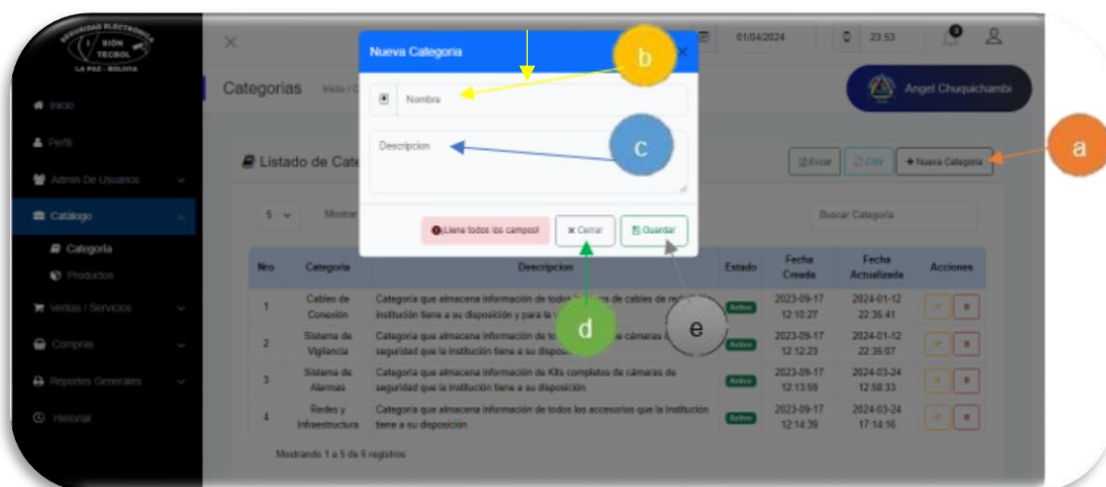
Acciones y Datos Generales

- Podemos observar las categorías
- Se puede ver la descripción de cada categoría
- Se puede observar el estado de la categoría
- d) Se puede observar la fecha creada
- Se puede observar la fecha actualizada
- Podemos observar las acciones que podemos realizar
- En esta opción podemos buscar de forma más directa las categorías



6.1 Crear una Nueva Categoría

- Esta opción nos ayuda a crear una nueva categoría
- Ponemos el nombre de la categoría
- Ponemos la breve descripción de la categoría
- La opción nos facilita cerrar la ventana
- La opción nos facilita guardar la categoría

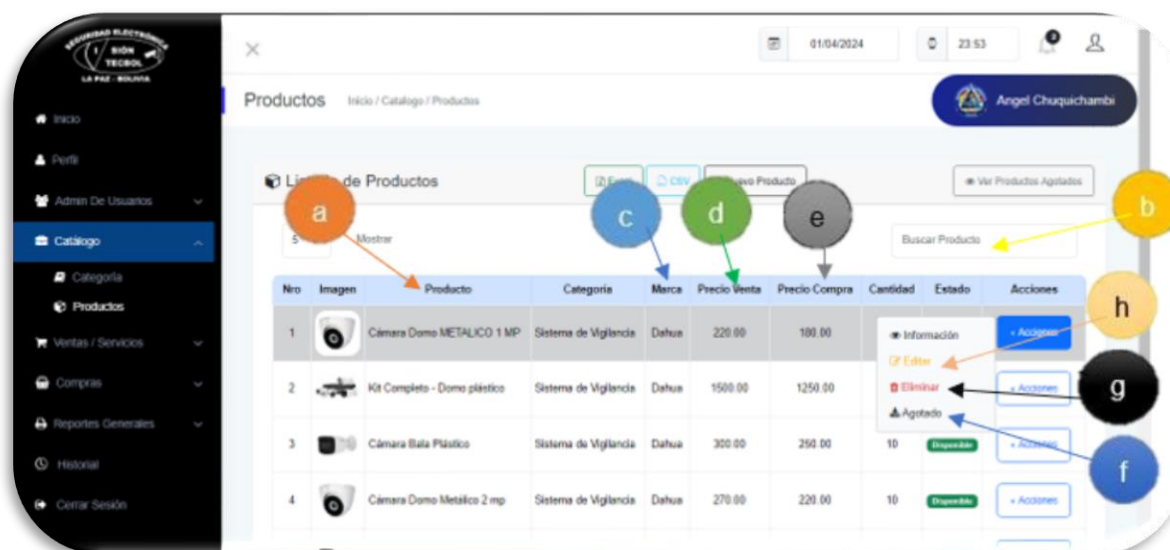


7. VISTA DE PRODUCTOS

- Podemos observar el nombre del producto
- En esta opción podemos buscar de forma más directa el producto
- Se puede ver la marca del producto
- Se puede observar el precio de venta
- Se puede observar el precio de compra
- Se puede ver si está agotado el producto

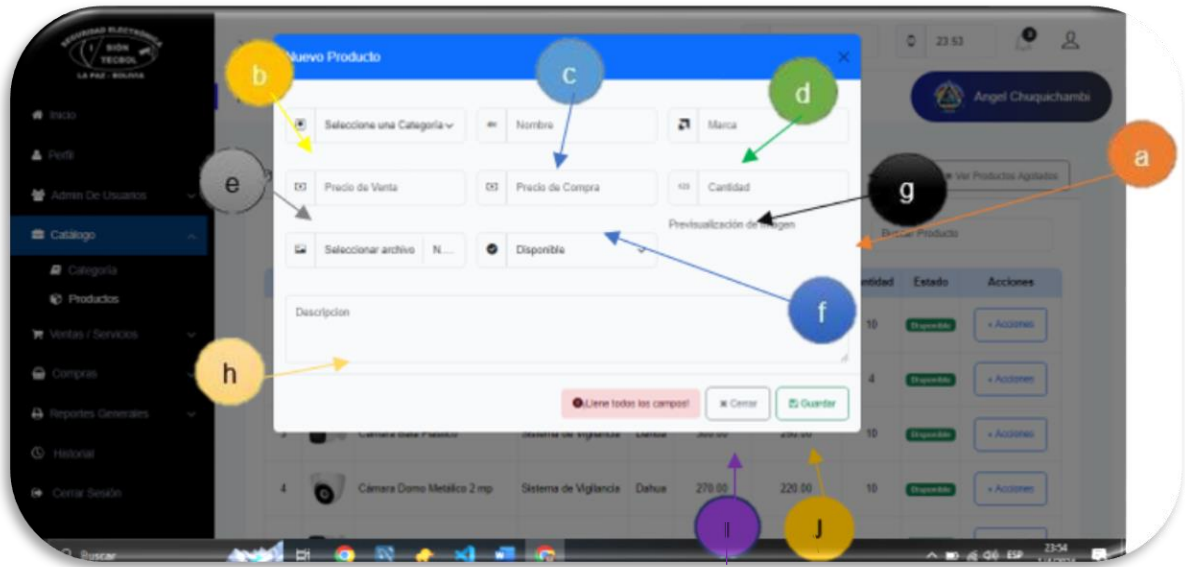
g) Se puede eliminar

h) Se puede editar



7.1 Crear un Nuevo Producto

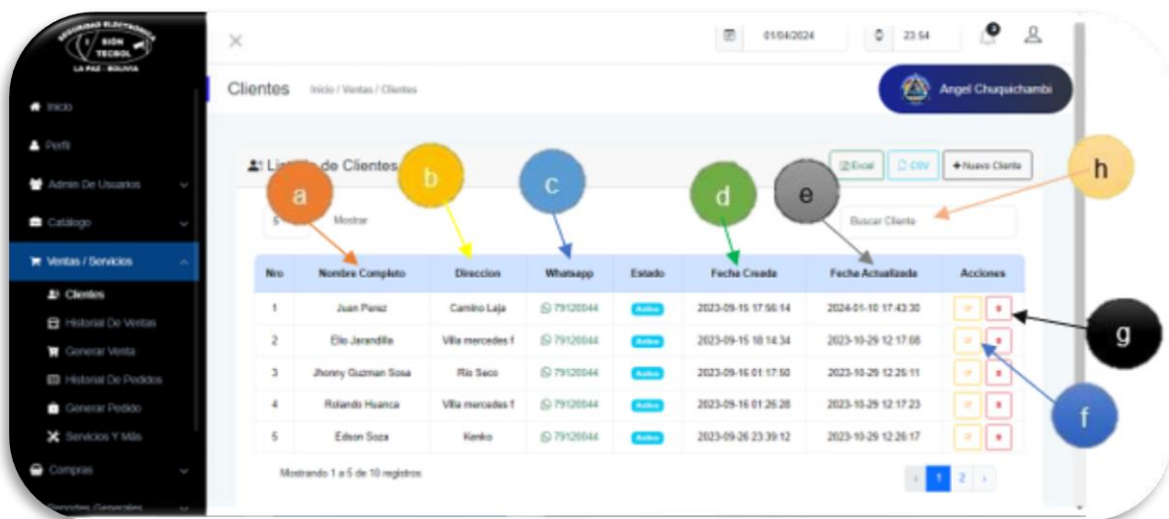
- Esta opción nos ayuda a crear un nuevo producto
- seleccionamos un a categoría
- Ponemos el nombre del producto
- Ponemos la marca del producto
- En esta opción ponemos el precio de venta
- En esta opción ponemos el precio de compra
- Ponemos la cantidad
- Ponemos la breve descripción del producto
- La opción nos facilita cerrar la ventana
- La opción nos facilita guardar el nombre del producto



8. VISTA DE CLIENTES

Datos generales y Acciones:

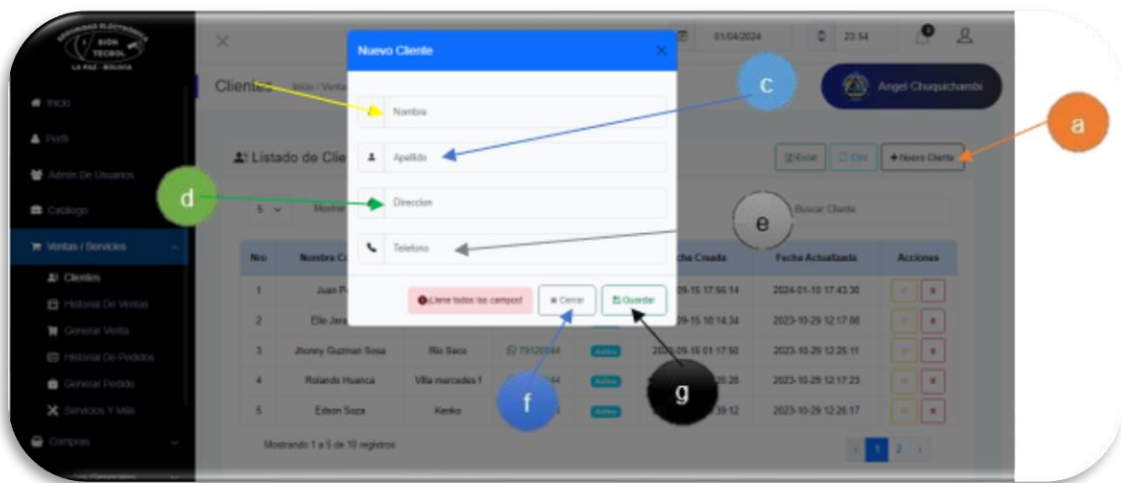
- Podemos observar el nombre del cliente
- Se puede ver la dirección del cliente
- Se puede observar el número de WhatsApp
- Se puede observar la fecha creada
- Se puede observar la fecha actualizada
- Se puede eliminar
- En esta opción podemos buscar de forma más directa el nombre del cliente



8.1 Crear un Nuevo Cliente

Pasos a seguir:

- Esta opción nos ayuda a crear un nuevo cliente
- Agregamos el nombre del cliente
- Agregamos el apellido del cliente
- Agregamos una dirección del cliente
- Agregamos un teléfono del cliente
- La opción nos facilita cerrar la ventana
- La opción nos facilita guardar al cliente



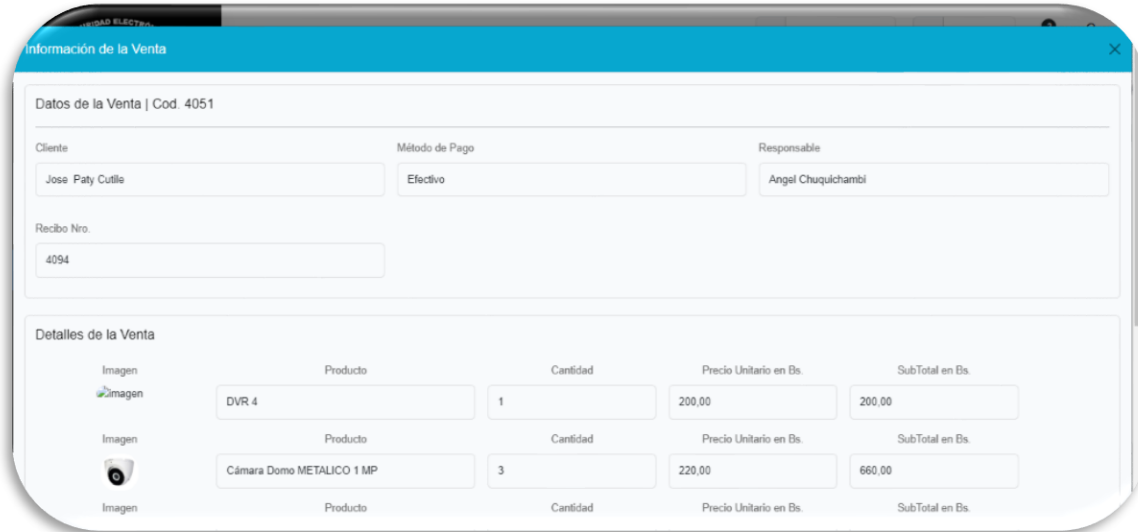
9. VISTA HISTORIAL DE VENTAS

Datos Generales y Acciones:

- Podemos observar el nombre del cliente
- Se puede ver la persona responsable
- Se puede observar cual fue el método de pago realizado
- Podemos ver el número del recibo
- Se puede observar la fecha realizada
- Se puede observar toda la información
- Se imprimir la venta
- En esta opción podemos buscar de forma más directa la venta
- Podemos observar toda la ganancia total



9.1 Lectura de Venta



10. VISTA GENERAR VENTA

Transacción de Venta

Datos Generales

Cliente: Elio Jarandilla

Método de Pago: Efectivo

Producto: Seleccione

Cantidad: 1

+ Agregar a la Lista

¡Llene todos los campos!

Productos Seleccionados

Borrar	Producto	Cantidad	PIU	SubTotal
<input type="checkbox"/>	DVR 4	1	200.00	200
<input type="checkbox"/>	Cámara Domo METALICO 1 MP	2	220.00	440
<input type="checkbox"/>	Cámara Domo Metálico 2 mp	2	270.00	540
<input type="checkbox"/>	Balum	4	25.00	100
Total Bs				1,260.00

Finalizar Venta

11. VISTA DE PEDIDOS

Datos Generales y Acciones:

- Podemos observar el nombre del cliente
- Se puede ver al responsable
- Se puede observar la fecha
- Se puede visualizar el pedido
- Se puede editar los datos del proveedor
- En esta opción podemos buscar de forma más directa el pedido

Ventas

Inicio / Ventas / Estado De Ventas

Angel Churuchambi

Buscar Pedido

Código	Cliente	Responsable	Método Pago	Fecha	Total (Bs)	Estado	Acciones
4047	Elio Jarandilla	Angel Churuchambi	Efectivo	2024-01-15	3400.00	En proceso	<input type="checkbox"/> <input type="checkbox"/>

Mostrando 1 a 10 de 1 registros

Bolivia - Visión TecBol © 2023

12. VISTA PROVEEDORES

Datos Generales y Acciones:

- Podemos observar el nombre del proveedor
- Se puede ver la dirección
- Se puede observar el número de WhatsApp
- Se puede observar la fecha creada
- Se puede observar la fecha actualizada
- Se puede editar
- Se puede eliminar
- En esta opción podemos buscar de forma más directa al proveedor

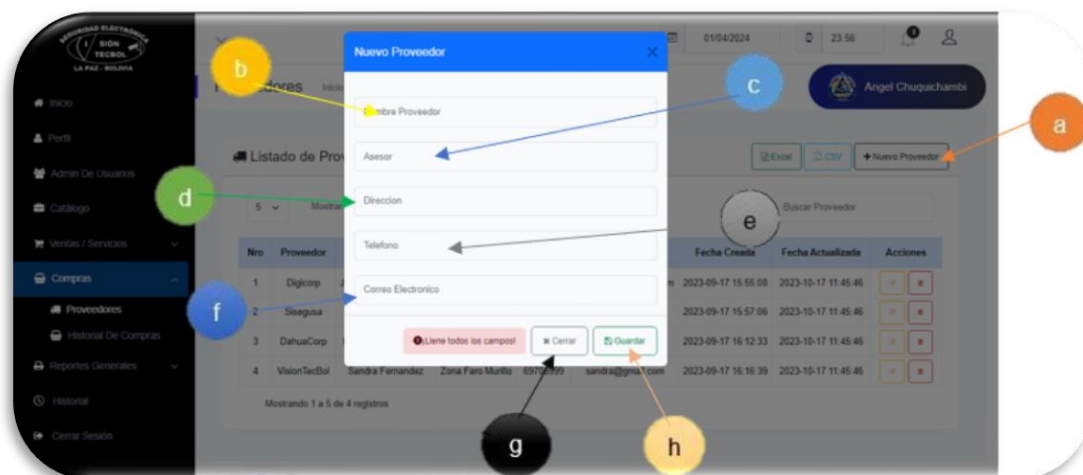
No	Proveedor	Asesor	Dirección	Teléfono	Correo Electrónico	Fecha Creada	Fecha Actualizada	Acciones
1	Digicoop	Juan Perez Fernandez	Zona Satelite	76793933	digicoop123@gmail.com	2023-09-17 16:55:08	2023-10-17 11:45:46	[+]
2	Segura	Angel Perez	Zona Obelisco	79120344	segura@gmail.com	2023-09-17 15:57:06	2023-10-17 11:45:46	[+]
3	DaniaCorp	Dania Esmeralda Pico	Villa Ingiero	61224645	dania@gmail.com	2023-09-17 16:12:33	2023-10-17 11:45:46	[+]
4	VisionTechSol	Sandra Fernandez	Zona Faro Marillo	69706999	sandra@gmail.com	2023-09-17 16:16:38	2023-10-17 11:45:46	[+]

Mostrando 1 a 5 de 4 registros

12.1 Crear un Nuevo Proveedor

Pasos a seguir:

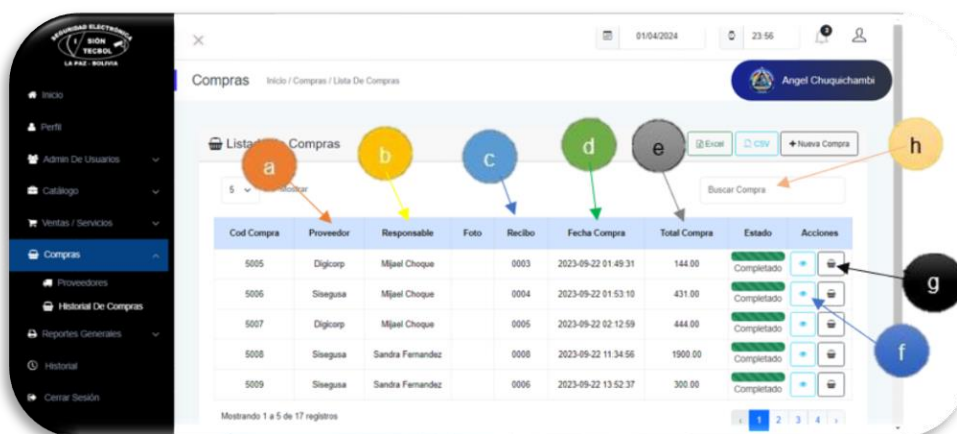
- Esta opción nos ayuda a crear un nuevo proveedor
- Agregamos el nombre del proveedor
- Agregamos un asesor
- Agregamos una dirección
- Agregamos un teléfono del proveedor
- agregamos el correo electrónico
- La opción nos facilita cerrar la ventana
- La opción nos facilita guardar al proveedor



13. VISTA DE COMPRAS

Datos Generales y Acciones:

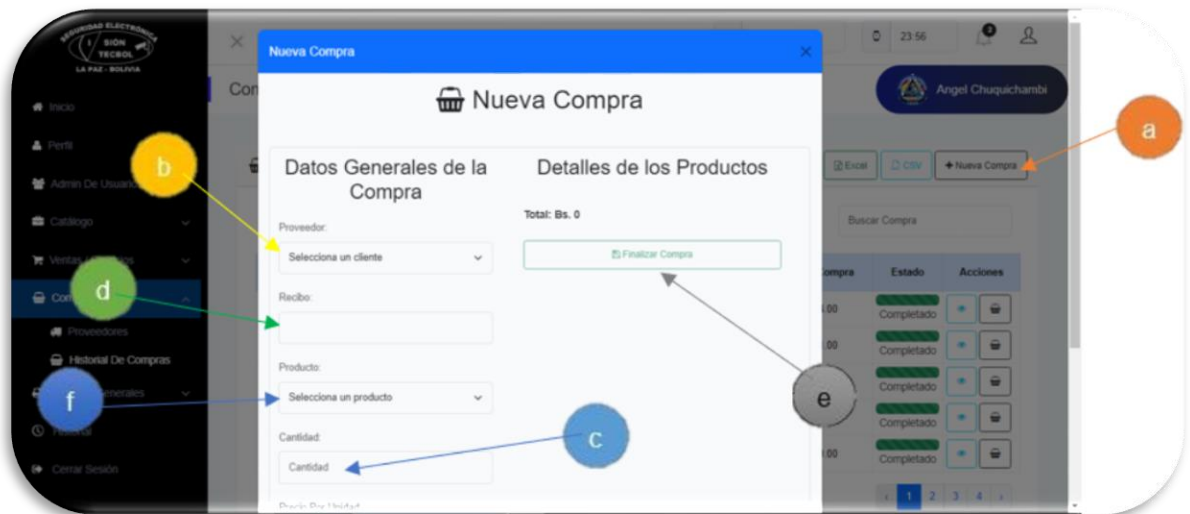
- Podemos observar el nombre del proveedor
- Se puede ver la persona responsable
- Se puede observar el recibo
- Podemos ver la fecha de compra
- Se puede observar el monto Total de compra
- Se puede observar toda la información
- podemos imprimir la compra
- En esta opción podemos buscar de forma más directa la compra



13.1 Crear una Nueva Compra

- Esta opción nos ayuda a crear una nueva compra
- Seleccionamos un cliente
- Agregamos el número del recibo

- d) Seleccionamos el producto
- e) Agregamos una cantidad
- f) Con esta opción finalizamos la compra



14. VISTA PARA GENERAR REPORTES

14.1 Reporte de ventas

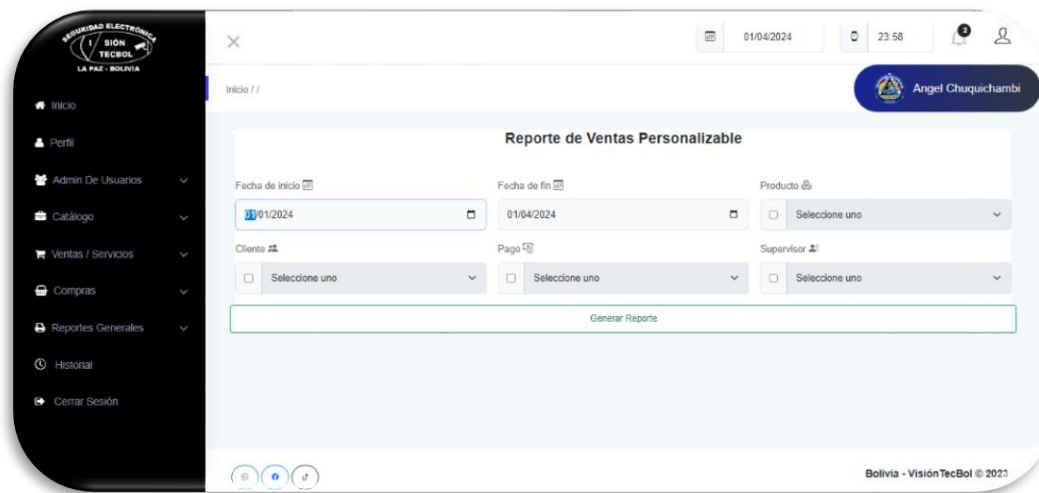
- a) Se puede descargar un reporte general del día de actual
- b) Se puede descargar un reporte personalizado

14.2 Reporte de compras

- a) Se puede descargar un reporte general del día de actual
- b) Se puede descargar un reporte personalizado



15. VISTA DE REPORTE PERSONALIZADO



16. IMPRESIÓN DE REPORTES

PRODUCTOS VENDIDOS DEL DIA						
CODIGO	FECHA	CLIENTE	VENDEDOR	PRODUCTO	CANTIDAD	SUBTOTAL
4949	2024-03-08	Rolando Huanca	Angel Chuquichambi	Kit Completo - Domo plástico	2	3000.00
4949	2024-03-08	Rolando Huanca	Angel Chuquichambi	Cámara Bala Plástico	1	300.00
4949	2024-03-08	Rolando Huanca	Angel Chuquichambi	Cable UTP Cat5	3	1800.00
4949	2024-03-08	Rolando Huanca	Angel Chuquichambi	DVR 4	2	400.00
					GANANCIA TOTAL	5.500.00

Nota: Reporte de ventas elaborado por Angel Chuquichambi

REPORTE DE VENTAS

VISIONTECBOL
Zona Faro Murrillo | Calle 1ro B
Cel. 76739333

PRODUCTOS VENDIDOS DEL 2024-01-01 AL 2024-04-01

Nro.	CLIENTE	VENDEDOR	PAGO	FECHA DE VENTA	PRODUCTO	CANTIDAD	SUBTOTAL
1	Jhonny Guzman Sosa	Angel Chuquichambi	Transferencia Bancaria	2024-01-09	Cámara Bala Plástico	4	1200.00
2	Jhonny Guzman Sosa	Angel Chuquichambi	Transferencia Bancaria	2024-01-09	Cámara Domo METALICO 1 MP	4	880.00
3	Jhonny Guzman Sosa	Angel Chuquichambi	Transferencia Bancaria	2024-01-09	Balun	8	200.00
4	Elio Jarandilla	Angel Chuquichambi	Efectivo	2024-01-15	DVR 4	1	200.00
5	Elio Jarandilla	Angel Chuquichambi	Efectivo	2024-01-15	Camara PTZ	2	2400.00
6	Elio Jarandilla	Angel Chuquichambi	Efectivo	2024-01-15	Cámara Bala Metálico	2	700.00
7	Elio Jarandilla	Angel Chuquichambi	Efectivo	2024-01-15	Balun	4	100.00
8	Edson Soza	Angel Chuquichambi	Efectivo	2024-03-07	Kit Completo - Domo plástica	1	1500.00

17. VISTA HISTORIAL DE ACTIVIDADES

Datos generales y Acciones:

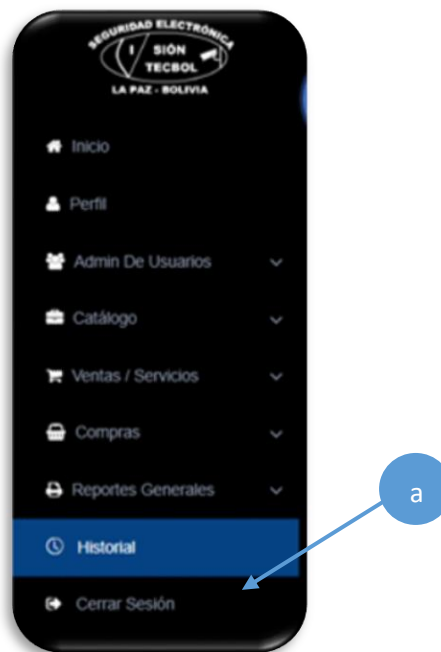
- Se observa información del usuario
- Se observa información del tipo de actividad
- Se observa información de la tabla afectada
- Se observa información del código que sufrió cambios
- Se observa la fecha de la acción
- Campo para buscar por usuarios

Historial de Actividades

Nro.	Usuario	Tipo de Actividad	Tabla	Cod. Registro	Fecha
1	Juan Perez	Modificación	Productos	3009	2024-03-24 18:00:20
2	Angel Chuquichambi	Modificación	Productos	3005	2024-03-24 17:25:38
3	Juan Perez	Eliminación	Categorías	12	2024-03-24 17:14:45
4	Juan Perez	Creación	Categorías	12	2024-03-24 17:14:38
5	Juan Perez	Modificación	Categorías	4	2024-03-24 17:14:16
6	Angel Chuquichambi	Modificación	Categorías	3	2024-03-24 12:58:33
7	Angel Chuquichambi	Modificación	Categorías	3	2024-03-24 12:58:26
8	Angel Chuquichambi	Eliminación	Categorías	11	2024-03-24 12:57:50
9	Angel Chuquichambi	Modificación	Categorías	11	2024-03-24 12:57:45
10	Angel Chuquichambi	Creación	Categorías	11	2024-03-24 12:56:46
11	Angel Chuquichambi	Creación	Categorías	10	2024-03-24 12:37:46
12	Angel Chuquichambi	Creación	Categorías		2024-03-24 12:34:23

18. CERRAR SESIÓN

Para Cerrar Sesión, lo único que se debe realizar es hacer click en la opción “Cerrar Sesión”.





Zona Faro Murillo
Av. Panorámica, Calle 13 "B"
Contactos: 79115564 -79115565

MANUAL TÉCNICO



“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE VENTAS Y COMPRAS DE DISPOSITIVOS DE SEGURIDAD”

LA PAZ – BOLIVIA

2024

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	1
2. REQUISITOS PREVIOS Y MÍNIMOS.....	1
2.1 REQUISITOS GENERALES	1
2.2 REQUISITOS DE HARDWARE MÍNIMOS	1
3. INSTALACIÓN DE XAMPP.....	1
3.1 DESCARGAR XAMPP	1
3.2 INSTALAR XAMPP	1
3.3 CONFIGURAR XAMPP	1
4. INSTALACIÓN DE VISUAL STUDIO CODE	2
4.1 DESCARGAR VISUAL STUDIO CODE.....	2
4.2 INSTALAR VISUAL STUDIO CODE	2
5. INSTALACIÓN DE COMPOSER	3
5.1 DESCARGAR COMPOSER	3
5.2 INSTALAR COMPOSER	3
5.3 VERIFICAR LA INSTALACIÓN DE COMPOSER	3
6. INSTALACIÓN DE LARAVEL LIVEWIRE.....	4
6.1 CREAR UN PROYECTO LARAVEL.....	4
6.2 INSTALAR LIVEWIRE	4
6.3 CONFIGURAR EL SERVIDOR LOCAL.....	4
7. INSTALACIÓN DE MYSQL WORKBENCH.....	4
7.1 DESCARGAR MYSQL WORKBENCH.....	5
7.2 INSTALAR MYSQL WORKBENCH	5
7.3 CONFIGURAR MYSQL WORKBENCH	5

1. Introducción

Este manual técnico proporciona instrucciones detalladas para la instalación y configuración de un entorno del desarrollo web completo utilizando XAMPP, Visual Studio Code, Laravel Livewire, MySQL Workbench, librerías y Composer. Este entorno es ideal para el desarrollo de aplicaciones web modernas con PHP y MySQL.

2. Requisitos Previos y Mínimos

Antes de todo, asegurarse de contar con:

1. Requisitos Generales

- **Un sistema operativo:** Windows 10 o superior.
- **Conexión a Internet**
- **Permisos** de administrador en tu computadora
- **Disco Sólido**
- **Espacio en Disco:** Al menos 5 GB de espacio libre.

2. Requisitos de Hardware Mínimos

- **Procesador:** Intel Core i3 o RYZE 3
- **Memoria RAM:** 4 GB (8 GB recomendado).
- **Espacio en Disco:** 5 GB de espacio libre (10 GB recomendado).

3. Instalación de Xampp

XAMPP es una distribución de Apache que incluye PHP, MySQL y Perl.

Pasos de Instalación:

a. Descargar XAMPP

- Visita el sitio web oficial de XAMPP: <https://www.apachefriends.org/index.html>
- Descarga el instalador para Windows.

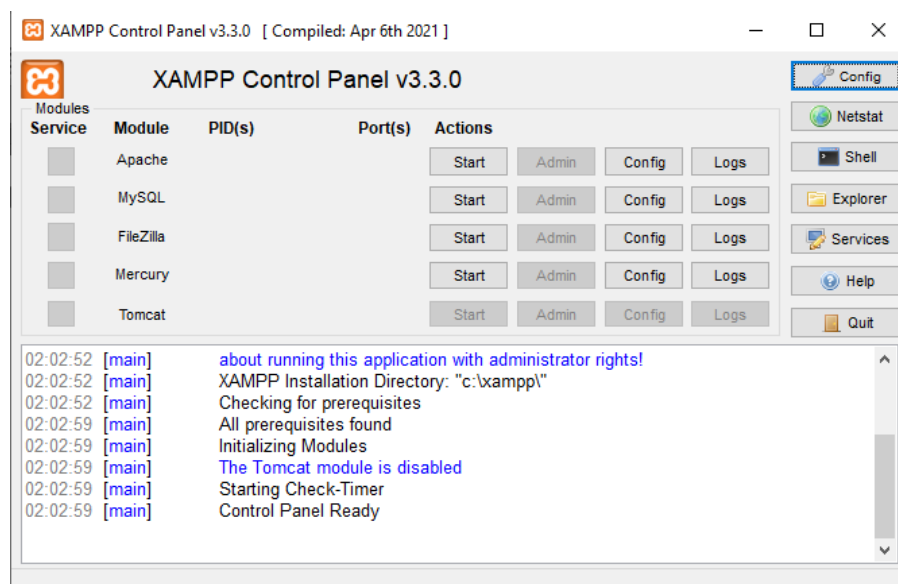
b. Instalar XAMPP

- Ejecuta el instalador descargado.
- Sigue las instrucciones del asistente de instalación.
- Selecciona los componentes a instalar (por defecto están seleccionados Apache, MySQL, PHP y Perl).
- Elige la carpeta de destino para la instalación.
- Completa la instalación y abre el panel de control de XAMPP.

c. Configurar XAMPP

- Inicia los servicios de Apache y MySQL desde el panel de control de XAMPP.

- Verifica la instalación abriendo un navegador web e ingresando <http://localhost>. Deberías ver la página de bienvenida de XAMPP.



4. Instalación de Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft.

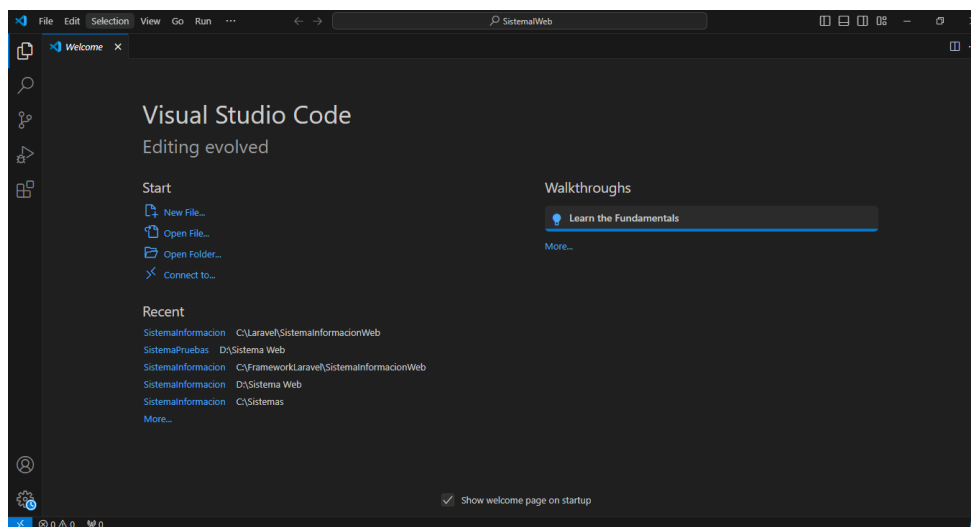
Pasos para la Instalación:

a. Descargar Visual Studio Code

- Visita el sitio web oficial de Visual Studio Code: <https://code.visualstudio.com/>
- Descarga el instalador adecuado para tu sistema operativo.

b. Instalar Visual Studio Code

- Ejecuta el instalador descargado.
- Sigue las instrucciones del asistente de instalación.
- Selecciona las opciones adicionales que desees (por ejemplo, crear un acceso directo en el escritorio).
- Completa la instalación y abre Visual Studio Code.



5. Instalación de Composer

Composer es un gestor de dependencias para PHP.

Pasos de Instalación:

a. Descargar Composer

- Visita el sitio web oficial de Composer: <https://getcomposer.org/>
- Descarga el instalador adecuado para tu sistema operativo.

b. Instalar Composer

- Ejecuta el instalador descargado.
- Sigue las instrucciones del asistente de instalación.
- Durante la instalación, asegúrate de que el instalador detecte correctamente el ejecutable de PHP (usualmente en la carpeta de XAMPP).
- Completa la instalación.

c. Verificar la Instalación de Composer

- Abre una terminal o línea de comandos.
- Ejecuta el comando `composer --version`. Deberías ver la versión de Composer instalada.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.4529]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Angel Chuquichambi>composer --version
Composer version 2.5.8 2023-06-09 17:13:21

C:\Users\Angel Chuquichambi>_

```

6. Instalación de Laravel Livewire

Laravel es un framework PHP para el desarrollo de aplicaciones web.

Pasos de Instalación:

a. Crear un Proyecto Laravel

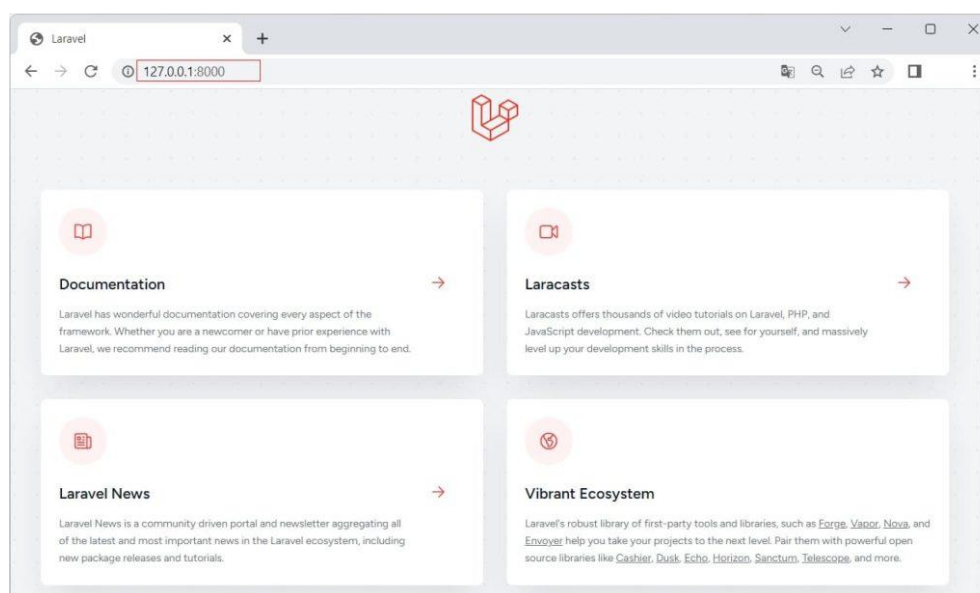
- Abre una terminal o línea de comandos.
- Navega al directorio donde deseas crear tu proyecto Laravel.
- Ejecuta el comando “**composer create-project --prefer-dist laravel/laravel nombre_del_proyecto**”.

b. Instalar Livewire

- Navega al directorio de tu proyecto Laravel.
- Ejecuta el comando “composer require livewire/Livewire”.

c. Configurar el Servidor Local

- Navega al directorio de tu proyecto Laravel.
- Ejecuta el comando “php artisan serve”.
- Abre un navegador web e ingresa <http://localhost:8000> para ver tu aplicación Laravel.



7. Instalación de MYSQL WORKBENCH

MySQL Workbench es una herramienta visual para la administración de bases de datos.

Pasos de Instalación:

a. Descargar MySQL Workbench

- Visita el sitio web oficial de MySQL: <https://dev.mysql.com/downloads/workbench/>
- Descarga el instalador adecuado para tu sistema operativo.

b. Instalar MySQL Workbench

- Ejecuta el instalador descargado.
- Sigue las instrucciones del asistente de instalación.
- Completa la instalación y abre MySQL Workbench.

c. Configurar MySQL Workbench

- Abre MySQL Workbench.
- Crea una nueva conexión a tu servidor MySQL local (usualmente, el host es localhost y el puerto 3306).
- Ingresa las credenciales de tu base de datos (usuario y contraseña).

