

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y ADMINISTRACIÓN DE ALMACENES Y CAJAS

CASO: CERÁMICA – TITAN BOLIVIA

Para optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Univ. Juan Carlos LLojlla Quispe

Tutor Metodológico: Lic. Ing. Helen Fanny Suntura Escobar

Tutor Revisor: Lic. Ing. Dionicio Henry Pacheco Rios

Tutor Especialista: Lic. Santos Zenon Quispe Apaza

EL ALTO-BOLIVIA

2024

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Juan Carlos Llojlla Quispe, estudiante con C.I. 11089243 LP mediante la presente declaro de manera pública que la propuesta del TRABAJO DE GRADO titulada “SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y ADMINISTRACIÓN DE ALMACENES Y CAJAS. Caso: Cerámica Titan Bolivia” 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 de El Alto.

El Alto, junio de 2024

Univ. Juan Carlos Llojlla Quispe
C.I. 11089243 L.P.
e-mail: jclojllaquispe@gmail.com

DEDICATORIA

este proyecto va dedicado a toda mi familia por ser la principal inspiración y el pilar fundamental en mi formación, por su incondicional apoyo durante estos años. A mis tutores cuya guía fue esencial para la elaboración del proyecto

AGRADECIMIENTOS

En primer lugar, dar las gracias a Dios por estar siempre conmigo, agradecer a mis padres en especial a mi madre por su apoyo incondicional para salir adelante.

Así mismo agradecer a la carrera ingeniería de sistemas y docentes por los años de formación que fueron piezas fundamentales para crecer cada día

Agradecer a la empresa “Cerámica Titan Bolivia” por darme la oportunidad de demostrar mis conocimientos adquiridos durante mi formación.

ÍNDICE GENERAL

1. CAPÍTULO I- MARCO PRELIMINAR.....	2
1.1 INTRODUCCIÓN.....	2
1.2 ANTECEDENTES	2
1.2.1 Antecedentes Institucionales.....	2
1.2.2 Antecedentes Internacionales	3
1.2.3 Antecedentes Nacionales	4
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.3.1 Problema Principal	6
1.3.2 Problemas Secundarios	6
1.4. FORMULACIÓN DEL PROBLEMA.....	6
1.5. OBJETIVOS.....	7
1.5.1. Objetivo General	7
1.5.2. Objetivo Especifico	7
1.6. JUSTIFICACIÓN	7
1.6.1. Justificación Técnica	7
1.6.2. Justificación Económica.....	8
1.6.3. Justificación Social.....	8
1.7. METODOLOGÍA	8
1.7.1. Metodología UWE.....	8
1.8. HERRAMIENTAS	9
1.8.1. Gestor de Base de Datos MySQL.....	9
1.8.2. Codeigniter 4.....	9
1.8.3. PHP 8.1.....	9
1.8.4. Servidor Virtual en la Nube.....	10

1.9. MÉTRICAS DE CALIDAD	10
1.9.1. ISO/IEC 9126	10
1.10. COSTO	10
1.10.1. Cocomo II	10
1.11. LIMITES Y ALCANCES	11
1.11.1. Limites.....	11
1.11.2. Alcances	11
1.12. APORTES	11
2. CAPÍTULO II- MARCO TEÓRICO.....	13
2.4. METODOLOGÍA UWE.....	13
2.4.1. Fases de la Metodología UWE	13
2.4.1.1. Captura, Análisis y Especificación de Requisitos.	13
2.4.1.2. Diseño del Sistema	13
2.4.1.3. Codificación del Software.....	14
2.4.1.4. Pruebas.....	14
2.4.1.5. La Instalación o Fase de Implementación.	14
2.4.1.6. El Mantenimiento.	14
2.4.2. Etapas de la Metodología UWE	14
2.4.2.1. Diseño de Requisitos	14
2.5. SISTEMA	18
2.6. ALMACÉN	18
2.7. PROMEDIO PONDERADO	18
2.8. CAJA	19
2.9. CODEIGNITER 4	19
2.9.1. Modelo MVC	19

2.9.1.1.	Modelo	20
2.9.1.2.	Controlador.....	20
2.9.1.3.	Vista.....	20
2.10.	PHP 8.1.....	21
2.10.1.	Características.....	21
2.10.2.	Ventajas de PHP	22
2.11.	JAVASCRIPT.....	22
2.12.	BOOTSTRAP 5.....	23
2.13.	MySQL.....	23
2.14.	PhpSpreadsheet.....	24
2.15.	CHAR JS.....	24
2.16.	CALIDAD DE SOFTWARE.....	25
2.16.1.	Estándar ISO/IEC 2126.....	25
2.16.1.1.	Características de Calidad de un Software según ISO/IEC 9126: 2001.....	26
2.17.	MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE.....	29
2.17.1.	Cocomo II	29
2.18.	SEGURIDAD DEL SISTEMA.....	31
2.18.1.	Estándar ISO 27000	32
3.	CAPÍTULO III- MARCO APLICATIVO.....	34
3.4.	INTRODUCCIÓN.....	34
3.5.	REQUERIMIENTOS DEL SISTEMA.....	34
3.5.1.	Requerimientos Funcionales	34
3.5.2.	Requerimientos No Funcionales	36
3.6.	IDENTIFICACIÓN DE ACTORES.....	36
3.3.1	Responsable de Compras:.....	36

3.3.2 Tesorería:	36
3.3.3 Cajero:	36
3.3.4 Jefe de almacén:	36
3.7. DIAGRAMA DE CASO DE USO GENERAL DEL SISTEMA	37
3.8. MODELO RELACIONAL.....	48
3.9. DIAGRAMAS DE NAVEGACIÓN.....	49
3.9.1. Diagrama de Navegación Modulo Caja	49
3.9.2. Diagrama de Navegación Modulo Compras	50
3.9.3. Diagrama de Navegación Modulo Tesorería	51
3.9.4. Diagrama de Navegación Administrar Usuarios, Cajas y Almacén	52
3.10. CODIFICACIÓN.....	53
3.10.1. Implementación del Sistema	54
3.10.1.1. Interfaz Inicio Sesión.....	54
3.10.1.2. Panel de Inicio.....	55
3.10.1.3. Administración de Usuarios.	55
3.10.1.4. Administración de Cajas.....	56
3.10.1.5. Administración de Almacenes.	56
3.10.1.6. Tablero de Almacén.	57
3.10.1.7. Formulario de Solicitud.	57
3.10.1.8. Movimientos de Almacén.	58
3.10.1.9. Stock en Almacén.....	58
3.10.1.10. Tablero de Cajas.....	59
3.10.1.11. Flujo de Efectivo por Caja.....	59
3.10.1.12. Formulario de Apertura.	60
3.10.1.13. Formulario de Ingreso en Caja.	60

3.10.1.14.	Formulario de Egresos en Caja.....	61
3.10.1.15.	Formulario de Cierre de Caja.....	61
3.10.1.16.	Reporte de Caja.....	62
3.10.1.17.	Reporte de Almacén.....	63
3.10.1.18.	Reporte de Compras.....	63
4.	CAPÍTULO IV- MÉTRICAS DE CALIDAD, ESTIMACIÓN DE COSTOS Y SEGURIDAD	65
4.4.	MÉTRICAS DE CALIDAD	65
4.4.1.	Estándar ISO 9126	65
4.4.1.1.	Funcionalidad.....	65
4.4.1.2.	Confiabilidad.....	68
4.4.1.3.	Usabilidad.....	69
4.4.1.4.	Mantenibilidad.....	71
4.4.1.5.	Portabilidad.....	72
4.4.2.	Metodología de Estimación de Costo de Software COCOMO II	73
4.5.	PRUEBAS DEL SISTEMA	76
4.5.1.1.	Pruebas de Caja Blanca.....	76
4.5.1.2.	Pruebas de Caja Negra.....	85
4.5.1.2.1.	Prueba de Caja Negra – Inicio de Sesión.....	85
4.5.1.2.2.	Prueba de Caja Negra – Crear Solicitud.....	86
4.5.1.2.3.	Prueba de Caja Negra – Ingreso en Caja.....	87
4.6.	Sistema de Gestión de Seguridad de la Información ISO-27002.....	88
4.6.1.	Seguridad Lógica	88
4.6.1.1.	Copias de Seguridad.....	88
4.6.1.2.	Identificación y autenticación.....	89
4.6.1.3.	Encriptación.....	89

4.6.2.	Seguridad Física.....	89
4.6.3.	Seguridad Organizativa.....	89
5.	CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	91
5.4.	CONCLUSIONES	91
5.5.	RECOMENDACIONES	92
6.	BIBLIOGRAFÍA.....	93

Índice de tablas

Tabla 2.1	Valores para Determinar el punto de fusión.....	26
Tabla 2.2	Coeficientes de Modelo COCOMO II	29
Tabla 2.3	Atributos de Modelo Intermedio COCOMO II.....	30
Tabla 3.1	Requerimientos Funcionales	34
Tabla 3.2	Detalle Administración de Usuarios	38
Tabla 3.3	Descripción de Gestión de Artículos	39
Tabla 3.4	Detalle Administración de Caja	40
Tabla 3.5	Detalle de Gestión de Proveedores	41
Tabla 3.6	Gestión de Categorías.....	42
Tabla 3.7	Gestión de Clientes	43
Tabla 3.8	Gestión de Destinos	44
Tabla 3.9	Detalle Solicitud de Compras	45
Tabla 3.10	Gestión de Movimientos en Caja	46
Tabla 3.11	Gestión de Reportes.....	47
Tabla 4.1	Parámetros de Medición.....	65
Tabla 4.2	Parámetros de Medición con Factor Medio	66
Tabla 4.3	Valores de Ajustes de Complejidad	66
Tabla 4.4	Escala de Valores.....	69
Tabla 4.5	Preguntas para determinar la usabilidad	70
Tabla 4.6	Valores para Determinar la Mantenibilidad	71
Tabla 4.7	Factor de Portabilidad.....	72
Tabla 4.8	Resultados de Calidad.....	73
Tabla 4.9	Constantes para el Cálculo de Cocomo.....	74
Tabla 4.10	Cálculo del Atributo ME	74

Tabla 4.11 Prueba de Caja Negra Inicio de Sesión	85
Tabla 4.12 Valores del Registro Solicitud.....	86
Tabla 4.13 Valores de Registro de Ingreso en Caja	87
Tabla 4.14 Seguridad Lógica	89

Índice de figuras

Figura 1.1	Organigrama Cerámica Titán.....	3
Figura 2.1	Diagrama de Caso de Uso.....	15
Figura 2.2	Diagrama Conceptual	16
Figura 2.3	Diagrama de Navegación	17
Figura 2.4	Diagrama de Presentación	17
Figura 3.1	Diagrama de Caso de Uso General.....	37
Figura 3.2	Caso de uso administración de almacén.....	38
Figura 3.3	Caso de Uso Administración de Artículos	39
Figura 3.4	Caso de uso administración de caja.....	40
Figura 3.5	Caso de uso gestión de proveedores.....	41
Figura 3.6	Caso de Uso Administración de Categorías.....	42
Figura 3.7	Caso de Uso Administración de Clientes	43
Figura 3.8	Caso de uso Administración de Destino.....	44
Figura 3.9	Caso de Uso Solicitud de Compra.....	45
Figura 3.10	Caso de Uso Administración de Caja.....	46
Figura 3.11	Caso de uso administración de reportes	47
Figura 3.12	Modelo Relacional del Sistema	48
Figura 3.13	Diagrama de Navegación Módulo de Caja.....	49
Figura 3.14	Diagrama de Navegación Modulo Compras	50
Figura 3.15	Diagrama de Navegación Modulo Compras	51
Figura 3.16	Diagrama de Navegación Usuarios, Cajas y Almacenes	52
Figura 3.17	Código Fuente Login.....	53
Figura 3.18	Inicio de Sesión.....	54
Figura 3.19	Panel de Inicio del Sistema.....	55
Figura 3.20	Administración de Usuarios	55

Figura 3.21	Administración de Cajas	56
Figura 3.22	Administración de Almacenes	56
Figura 3.23	Almacenes Habilitados en el Sistema	57
Figura 3.24	Formulario de Solicitud de Compra	57
Figura 3.25	Movimientos en Almacén de Artículos	58
Figura 3.26	Existencias Físicas en Almacén	58
Figura 3.27	Tablero de Cajas.....	59
Figura 3.28	Flujo de Caja	59
Figura 3.29	Formulario de Apertura de Caja	60
Figura 3.30	Formulario de Ingresos a Caja	60
Figura 3.31	Egresos Caja.....	61
Figura 3.32	Formulario de Cierre de Caja	62
Figura 3.33	Reporte Caja	62
Figura 3.34	Reporte de Almacén	63
Figura 3.35	Reporte de Compras.....	63
Figura 4.1	Grafo del Sistema	77
Figura 4.2	Prueba de Caja Negra Inicio de Sesión	85
Figura 4.3	Prueba de Caja Negra crear Solicitud	86
Figura 4.4	Prueba de Caja Negra Ingreso en Caja	87

Índice de ecuaciones

(1) Ecuación de Punto de Fusión.....	27
(2) Ecuación de la Función.....	27
(3) Ecuación de la Confiabilidad	27
(4) Ecuación de la Disponibilidad.....	27
(5) Ecuación de la Usabilidad	28
(6) Ecuación de Mantenimiento	28
(7) Ecuación de Esfuerzo	30
(8) Ecuación de Tiempo	30
(9) Ecuación para Determinar el Número de Personas.....	30
(10) Ecuación para Determinar el Número de Líneas de Códigos en Miles.....	73
(11) Ecuación para Determinar el Ciclomática	80

RESUMEN

El presente proyecto se desarrolla para dar conformidad a los requisitos exigidos por la Universidad Pública de El Alto para dar fin a la carrera de Ingeniería de Sistemas mediante el proceso de titulación. El proyecto lleva por nombre “SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y ADMINISTRACIÓN DE ALMACENES Y CAJAS caso: CERÁMICA TITAN BOLIVIA”. La empresa tiene dificultades en la gestión de la información como registros manuales en hojas de cálculo en las distintas áreas como compras, almacén, caja y tesorería.

El proyecto desarrollado trata sobre la gestión de la elaboración de solicitudes de compra, evaluación y aprobación, desembolso de efectivo en caja en función a la solicitud, registro de entradas según el pedido, salida de artículos de los almacenes, control del saldo físico en función al almacén, control del flujo diario por cajas.

Para la elaboración del proyecto se utilizó como framework CodeIgniter 4, gestor de base de datos MySQL, lenguaje de programación PHP 8, a lado de front-end se utilizó Bootstrap 5 y como metodología se empleó UWE que permite el desarrollo de aplicaciones en un entorno WEB.

Así mismo, para determinar la calidad de sistema Web desarrollado se hace uso de la ISO 9126 y finalmente se realiza el análisis de costos se empleó COCOMO II

Palabras Clave: Almacén, Cajas, Solicitudes, Compras, metodología.

ABSTRACT

This project is developed to comply with the requirements demanded by the Public University of El Alto to complete the Systems Engineering degree through the degree process. The project is called "WEB INFORMATION SYSTEM FOR THE CONTROL AND ADMINISTRATION OF WAREHOUSES AND BOXES case: CERÁMICA TITAN BOLIVIA". The company has difficulties in managing information such as manual records in spreadsheets in different areas such as purchasing, warehouse, cash and treasury.

The developed project deals with the management of the preparation of purchase requests, evaluation and approval, disbursement of cash in cash based on the request, registration of receipts according to the order, removal of items from warehouses, control of the physical balance based on to the warehouse, control of the daily flow by boxes.

To develop the project, CodeIgniter 4 was used as a framework, MySQL database manager, PHP 8 programming language, Bootstrap 5 was used as a front-end and UWE was used as a methodology that allows the development of applications in an environment WEB.

Likewise, to determine the quality of the developed Web system, ISO 9126 is used and finally the cost analysis is carried out, COCOMO II is used.

Keywords: Warehouse, Boxes, Requests, Purchases, methodology

LISTADO DE ABREVIATURAS

- **PHP** - Hypertext Preprocessor
- **KLDC** - Kilo Lines of Delivered Code
- **UWE** - UML-Based Web Engineering
- **ISO** - International Organization for Standardization
- **UML** - Unified Modeling Language
- **MVC** - Modelo-Vista-Controlador
- **JS** – JavaScript
- **CSS** - Cascading Style Sheets
- **LCD** – Lineas de codigo
- **GPL** – General Public License
- **HTML** – HyperText Markup Language
- **SVG** - Scalable Vector Graphics
- **SGBD** – Sistema Gestor de Base de Datos
- **DBMS** - Scalable Vector Graphics

CAPÍTULO I

MARCO PRELIMINAR

1. CAPÍTULO I- MARCO PRELIMINAR

1.1 INTRODUCCIÓN

En la era digital actual, los sistemas de información juegan un papel fundamental en el funcionamiento y la competitividad de las empresas. Los sistemas no solo facilitan la gestión eficiente de datos y procesos, sino que también permiten a las organizaciones adaptarse rápidamente a un entorno empresarial.

La problemática principal se manifiesta en la gestión manual de la información en hojas físicas lo cual demora tiempo, gasto de recursos para la empresa y a la vez conduce a una mala administración de los reportes.

Frente a esta situación se plantea el desarrollo de un sistema de información web. El cual tiene como propósito automatizar los registros y brindar reportes al instante y desde el cualquier lugar. las herramientas que se utilizará para el desarrollo del proyecto son Codeigniter 4 como framework, Mysql para la gestión de base de datos y como metodología UWE.

1.2 ANTECEDENTES

1.2.1 *Antecedentes Institucionales.*

La empresa CERÁMICA TITAN BOLIVIA fue creada en el año 1995 en el departamento de La Paz, provincia ingavi, ciudad de VIACHA, con número de identificación tributaria 4889934015 que corresponde al señor JOSE JHASMANI PATZI YUJRA.

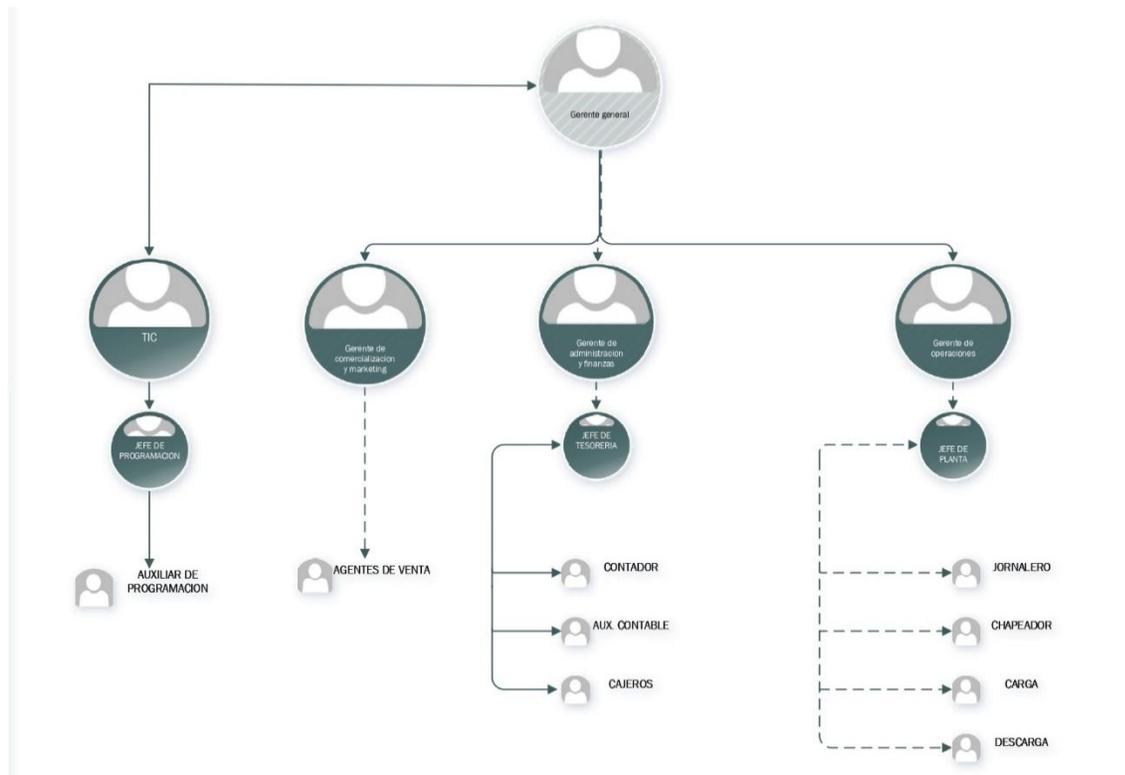
La empresa actualmente cuenta con cuatro líneas de producción de cerámicas(ladrillos) las cuales comercializa a nivel nacional e internacional.

Misión. - Formar parte de los mejores proyectos de vida, comprometidos con superar la calidad de nuestros productos cerámicos para la construcción.

Visión. - liderar la transformación de la industria en el país, superando nuestras propias expectativas hacia la mejor oferta de productos para la construcción dentro y fuera de nuestras fronteras.

Figura 1.1

Organigrama Cerámica Titán



Nota. Organigrama, Fernando,2023, Cerámica Titan

1.2.2 Antecedentes Internacionales

“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE VENTAS E INVENTARIO”
CASO: EMPRESA KARISHMAS.[PROYECTO DE GRADO], Para la construcción del sistema se utilizó como herramienta de desarrollo Java Netbeans, especializada en herramientas de escritorio, y para la gestión de manejo de datos se utiliza SQL en la plataforma PhpMyAdmin de Xampp, El sistema cuenta con la gestión de 4 pilares fundamentales de funcionamiento que son: Gestión de ventas, Control de productos o

inventario, Control de clientes y Control de proveedores. Universidad Técnica Federico Santa María Sede Viña Del Mar Chile (Echeverría, 2017).

“DESARROLLO DE UN SISTEMA DE CONTROL DE INVENTARIO PARA LA GESTIÓN DE INSUMOS EN LA EMPRESA “MINIMARKET DON LUCHO” UBICADA EN LA PARROQUIA VIRGEN DE FÁTIMA MEDIANTE DATOS ESTADÍSTICOS” [PROYECTO DE GRADO], El sistema contiene seis módulos proveedores, inventario, producto, tipo de producto, compra y venta también constando de reportes. Se emplearon herramientas de desarrollo como PHP, MySQL y JavaScript. Universidad Estatal de milagro Ecuador (Jimenez, 2020).

“DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL CONTROL DE INVENTARIO Y ALQUILER DE MAQUINARIAS DE LA EMPRESA S.A.”[PROYECTO DE GRADO]

Universidad Politécnica Salesiana Sede Guayaquil. nos apoyamos en la metodología de desarrollo de software ágil XP Vásquez rudas, J. F. (2014)”

Diseño de un sistema basado en tecnología web para el control y gestión de venta de unidades móviles” [TESIS] Para alcanzar este objetivo, nos apoyamos en la metodología de desarrollo de software ágil XP (Xtreme Programming - Programación Extrema(Vera, 2019)

1.2.3 Antecedentes Nacionales

“Sistema Web de Control de Ventas e Inventario” Caso: MICHELLINE [PROYECTO DE GRADO]

Para la implementación se utilizó como gestor de base de datos MySQL, además como lenguaje de programación se utiliza PHP. La calidad del sistema se lo realizó bajo el estándar ISO 9126 que evalúa aspectos como usabilidad, funcionalidad, confiabilidad,

mantenibilidad y portabilidad, proporcionando una evaluación tras la implementación del Sistema Web.

El proyecto fue desarrollo utilizando ciertas metodologías, es así que como metodología de desarrollo ágil se utilizó XP (Extreme Programming) UMSA, La Paz (Aduviri, 2016).

“Aplicación Web para el Registro y Control Compras, Ventas e Inventarios Caso: ASOCIACIÓN DE PRODUCTORES EN METALMECÁNICA SUR METAL SUR” [PROYECTO DE GRADO].

Para alcanzar este objetivo se utilizó la metodología de desarrollo de software ágil XP (Xtreme Programming - Programación Extrema), Con la implementación del nuevo sistema, se logró mejorar el proceso de registro de compras, ventas y el manejo de su inventario de los materiales y artículos existentes UMSA, La Paz (Aquino, 2018).

“Sistema Web de Control de Compras, Ventas e Inventario” Caso: COMERCIAL ARIANA. [PROYECTO DE GRADO]

El desarrollo del proyecto se basó por la Metodología de Desarrollo Ágil XP (Extreme Programming – Programación Extrema). Como herramienta de desarrollo de aplicaciones web se utilizó el framework Laravel en su versión 5.4 complementado con el gestor de base de datos MySQL. Para el diseño responsivo se utilizó el framework Bootstrap, acompañado de Javascript, JQuery y PHP. UMSA, La Paz (Carrillo, 2017).

1.3 PLANTEAMIENTO DEL PROBLEMA

En la actualidad la empresa CERÁMICA TITAN BOLIVIA presenta dificultades en el control del manejo de la información. En almacén se logró identificar algunos problemas cotidianos como el registro de entradas y salidas de los artículos en papel y posteriormente por la tarde pasan a un archivo en Excel, las solicitudes de compras son elaboradas en

hojas y posteriormente llevadas en físico a tesorería para que la compra sea aprobada lo cual es muy moroso.

En caja el ingreso y egreso se registra de forma escrita en talonarios y después del cierre de caja todos los movimientos se pasan a un archivo en Excel lo cual genera pérdida de tiempo.

1.3.1 Problema Principal

La empresa CERÁMICA TITAN BOLIVIA realiza el proceso de gestión de almacenes de forma manual, desde generar y aprobar solicitudes de compras hasta el registro del desembolso de efectivo en caja lo cual provoca deficiencias en el manejo de la información.

1.3.2 Problemas Secundarios

Al realizar la recopilación de la información, se identificaron los siguientes problemas secundarios:

- ✓ Las solicitudes de compra de materiales, su verificación y aprobación se realizan manualmente en hojas impresas
- ✓ Registro de entradas y salidas de almacén se realiza en hojas físicas y posteriormente se pasa a hojas de cálculo.
- ✓ Los ingresos y egresos de caja son registrados en talonarios.
- ✓ El cierre de caja diario se realiza de forma manual.
- ✓ No existe la información exacta sobre el inventario de los almacenes.

1.4. FORMULACIÓN DEL PROBLEMA

De los problemas mencionados anteriormente surge la siguiente interrogante:

¿de qué manera se podrá realizar un control eficiente de los procesos de entrada y salida de los artículos en los almacenes, solicitud, verificación, aprobación de compra de artículos y desembolso del importe solicitado en caja para la compra de los materiales?

1.5. OBJETIVOS

1.5.1. *Objetivo General*

Desarrollar un SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y ADMINISTRACIÓN DE ALMACENES Y CAJAS para un control adecuado de la información y mejor eficiencia en los procesos administrativos de la empresa CERÁMICA TITAN BOLIVIA.

1.5.2. *Objetivo Especifico*

- ✓ Diseñar la base de datos para almacenar la información de los registros.
- ✓ Desarrollar módulos para generar solicitudes de compras, verificación y aprobación en función al almacén.
- ✓ Implementar un módulo para el registro de entradas y salidas de artículos
- ✓ Automatizar el control de saldos mínimos disponibles en cualquier articulo registrado en el sistema de los almacenes.
- ✓ Diseñar el módulo para el control de flujo de caja, donde los comprobantes de ingresos y egreso se generen de forma automática en base formato pdf.
- ✓ Automatizar el cierre de caja para un mejor control del efectivo y evitar errores al momento de entregar el efectivo a tesorería.
- ✓ Diseñar módulos para generar reportes del flujo diario en función a caja, saldo físico del inventario

1.6. JUSTIFICACIÓN

1.6.1. *Justificación Técnica*

La empresa CERÁMICA TITAN BOLIVIA cuenta con un servidor privado, internet y con la infraestructura necesaria para implementar el sistema, su vez el proyecto se desarrollará con herramientas software libre para la obtención de un producto de calidad, utilizando interfaces adecuadas facilitando la interacción con el usuario y brindando medidas de seguridad para la información.

1.6.2. Justificación Económica

El presente proyecto implica un ahorro en tiempo y dinero al momento de la manipulación de datos, sustituyendo los procesos manuales realizados con herramientas ofimáticas, por otro lado, colaborará al personal de la empresa a desarrollar sus actividades con facilidad.

La empresa se beneficiará del sistema de información en la optimización de tiempos y el manejo eficiente de la información respecto a los procesos existentes dentro de la empresa y ahorrará el uso de materiales de escritorio.

1.6.3. Justificación Social

Con la implementación del proyecto, beneficiará directamente a la empresa CERÁMICA TITAN BOLIVIA, simplificando el trabajo tedioso e incrementando la eficiencia y mejorando la calidad de servicio a los clientes por parte de la empresa.

El proyecto se justifica socialmente debido a que los consumidores finales gozaran los beneficios de una atención sin demora.

1.7. METODOLOGÍA

1.7.1. Metodología UWE

UWE nació a finales de la década de los 90 con la idea de encontrar una forma estándar para analizar y diseñar modelos de sistemas web. El objetivo por el cual nació esta metodología fue utilizar un lenguaje común o por lo menos definir un metamodelo basado en el mapeo a lo largo de las diferentes etapas. En esa época UML prometía convertirse en un estándar para el modelamiento de sistemas. Por este motivo, UWE se adhirió a UML y no a otra técnica de modelado. UWE se ha adaptado a las nuevas características de los sistemas web como transacciones, personalizaciones y aplicaciones asíncronas, y por otro lado ha evolucionado para incorporar técnicas de ingeniería de

software como el modelamiento orientado a aspectos y nuevos lenguajes de transformación para mejorar la calidad del diseño (Olsina, 2008).

1.8. HERRAMIENTAS

Existen una infinidad de herramientas para el desarrollo de sistemas de información, para el uso de este software emplearemos las siguientes herramientas:

1.8.1. Gestor de Base de Datos MySQL

MySQL es la base de datos de código abierto más popular del mercado. Según DB-Engines, MySQL se clasifica como la segunda base de datos más popular, detrás de Oracle Database. MySQL potencia muchas de las aplicaciones más accesibles (Oracle, 2024)

1.8.2. Codeigniter 4

Es un marco de desarrollo de aplicaciones (un conjunto de herramientas) para personas que crean sitios web utilizando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría hacerlo si estuviera escribiendo código desde cero, proporcionando un amplio conjunto de bibliotecas para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas (Codeigniter, 2024).

Codeigniter le permite concentrarse creativamente en su proyecto minimizando la cantidad de código necesario para una tarea determinada.

1.8.3. PHP 8.1

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (PHP, 2024).

1.8.4. Servidor Virtual en la Nube

Un servidor virtual en la nube es una potente infraestructura virtual que lleva a cabo el almacenamiento de procesamiento de aplicaciones e información. Los servidores en la nube se crean utilizando software de virtualización para dividir un servidor físico en varios servidores virtuales. Las organizaciones utilizan un modelo de infraestructura como servicio para procesar cargas de trabajo y almacenar información.

1.9. MÉTRICAS DE CALIDAD

1.9.1. ISO/IEC 9126

En 1991 se publica la primera edición de la norma ISO/IEC 9126 con objeto de promover un entorno que permita la evaluación de la calidad del software. La norma ISO/IEC 9126 según su modelo de calidad propone: (Ballesteros, 2003, p. 3)

- ✓ Funcionalidad
- ✓ Fiabilidad
- ✓ Facilidad de uso
- ✓ Eficiencia
- ✓ Mantenimiento
- ✓ Movilidad

1.10. COSTO

1.10.1. Cocomo II

COCOMO es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software (González, 2014, p. 6).

1.11. LIMITES Y ALCANCES

1.11.1. Límites

En cuanto a los límites del presente proyecto se lo describe en el siguiente punto:

- ✓ El proyecto no podrá realizar transacciones monetarias vía internet.
- ✓ El proyecto no contemplara facturación electrónica.
- ✓ El sistema no realiza ventas en línea.

1.11.2. Alcances

Los alcances del proyecto orientado a la administración de recursos humanos engloban los siguientes módulos:

El presente proyecto contempla los siguientes alcances:

- ✓ Módulo solicitud de compras
- ✓ Módulo almacén de artículos
- ✓ Módulo caja
- ✓ Modulo tesorería
- ✓ Reportes en formato EXCEL y PDF
- ✓ Consulta de saldos físicos actuales de los distintos almacenes de la empresa

1.12. APORTES

Los aportes que proporciona el Sistema brindara la elaboración inmediata de reportes del área almacén, caja y solicitudes de compra. Además de brindar una interfaz amigable para los usuarios.

CAPÍTULO II

MARCO TEÓRICO

2. CAPÍTULO II- MARCO TEÓRICO

En este capítulo se describirán conceptos y definiciones de las herramientas que serán utilizadas para el desarrollo del software. Estos conceptos serán de ayuda para entender el objeto de estudio del presente proyecto de grado. Se hará uso de la metodología UWE para el desarrollo del software en un entorno web.

2.4. METODOLOGÍA UWE

UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace, además, una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. Según (Galiano, 2012, p. 5)

2.4.1. Fases de la Metodología UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

2.4.1.1. Captura, Análisis y Especificación de Requisitos.

Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipo de la interfaz de usuario.

2.4.1.2. Diseño del Sistema.

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

2.4.1.3. Codificación del Software.

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.4.1.4. Pruebas.

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

2.4.1.5. La Instalación o Fase de Implementación.

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

2.4.1.6. El Mantenimiento.

En esta fase se realiza mejoras y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.4.2. Etapas de la Metodología UWE

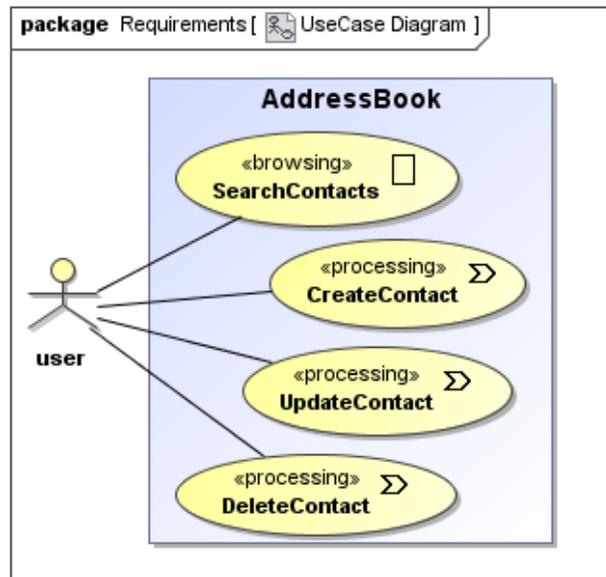
2.4.2.1. Diseño de Requisitos

- **Caso de uso.** - En UWE los casos de uso se distinguen por los estereotipos «navegación» y «procesamiento» para indicar si una aplicación modifica los datos persistentes de la aplicación o no. "SearchContact", por ejemplo, modela la búsqueda de contactos y tiene el estereotipo de «navegar» porque durante una

búsqueda la base de datos sólo se lee y se presenta al usuario. Los otros casos de uso, sin embargo, modelan cambios de datos y por lo tanto se escriben con «procesamiento». (UWE, 2016)

Figura 2.1

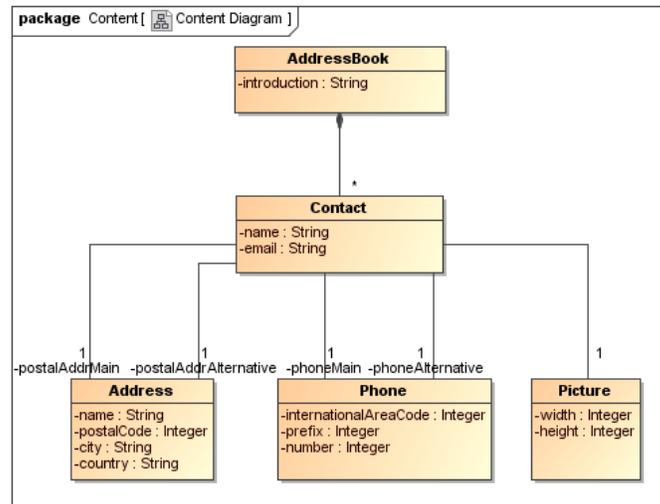
Diagrama de Caso de Uso



Nota: (UWE, 2016)

- **Diseño Conceptual.** - UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizarán en la aplicación Web.

Figura 2.2

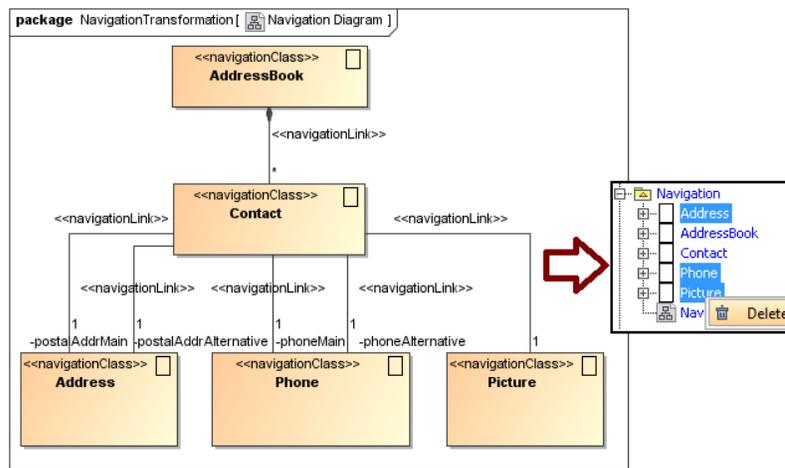
Diagrama Conceptual

Nota: (UWE, 2016)

- **Diseño de Navegación.** - Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. Para un sistema de páginas web, sería bueno saber cómo están vinculadas entre sí. Eso significa que necesitamos un diagrama que contenga nodos y enlaces. (UWE, 2016)

Figura 2.3

Diagrama de Navegación

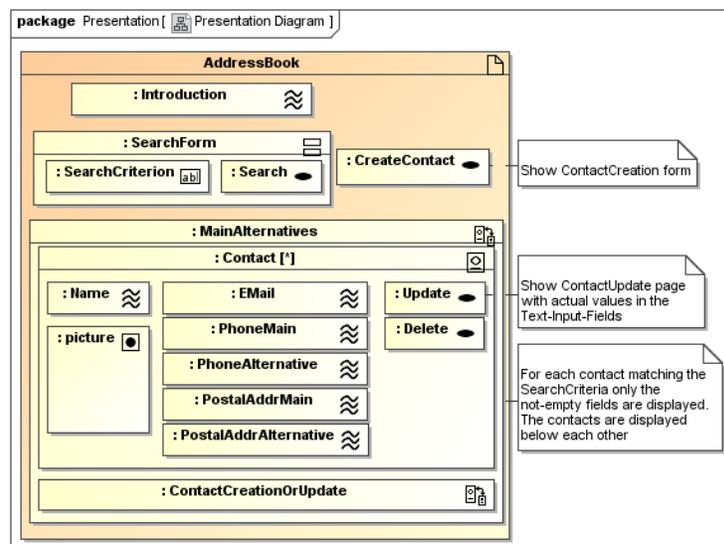


Nota: (UWE, 2016)

- **Diseño de Presentación.** - Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario

Figura 2.4

Diagrama de Presentación



Nota: (UWE, 2016)

2.5. SISTEMA

Un sistema informático permite almacenar y procesar información. Por la utilidad que propone, puede adaptarse a varios sectores de actividad económica. Un sistema resulta de la interacción entre los componentes físicos que se denominan Hardware y los lógicos que se denominan Software. A estos hay que agregarles el recurso humano, parte fundamental de un sistema. Este componente es llamado Humanware. En un sistema informático, la información es introducida a través de los periféricos de entrada, luego es procesada y mostrada por los periféricos de salida (Eduteka, 2018).

2.6. ALMACÉN

El almacén es el espacio físico de la empresa donde se efectúa la función de almacenaje. Esta facilidad para definir con una sola palabra como “almacenaje” la función que desempeña dentro de la empresa uno de sus departamentos, podría ser la causa de esa especie de invisibilidad que padece. Es en el almacén donde en la mayoría de las empresas acaba el personal de menor cualificación, es donde se producen la mayor parte de las tensiones por los esfuerzos del último momento, horas extraordinarias, trabajos en festivos... Sin embargo y curiosamente es precisamente donde la empresa consigue hacer efectivo el servicio al cliente y en demasiadas ocasiones con pocos medios o con medios poco actualizados. (DATADEC, 2024)

2.7. PROMEDIO PONDERADO

Se conoce como promedio a la cifra que resulta idéntica o que es la más cercana a la media aritmética. El promedio también puede ser el punto en el que una cosa se divide al medio.

La noción de promedio ponderado se utiliza para nombrar a un método de cálculo que se aplica cuando, dentro de una serie de datos, uno de ellos tiene una importancia

mayor. Hay, por lo tanto, un dato con mayor peso que el resto. El promedio ponderado consiste en establecer dicho peso, también conocido como ponderación, y utilizar dicho valor para realizar el cálculo del promedio (Gardey., 2017)

2.8. CAJA

Se entiende por caja la tenencia de dinero por parte de un agente económico, el cual lleva un registro contable, en el que se anotan las entradas y salidas de efectivo.

En la contabilidad de una empresa, la cuenta contable de caja es la que refleja el saldo de dinero en efectivo o en cheques sin cobrar, existente en la empresa en un momento determinado. En una empresa puede haber dos tipos, la de la moneda funcional, que es la de uso corriente en el país, y una de moneda extranjera. Esta última se utilizará si la empresa realiza cobros y pagos de manera relativamente habitual en una moneda extranjera.

Esta cuenta pertenece al activo corriente de la empresa. En el balance de situación figura dentro del apartado de efectivo y otros activos líquidos equivalentes. Junto con los saldos de las cuentas de los bancos y otras inversiones a corto plazo de gran liquidez. (Sage Software, 2021)

2.9. CODEIGNITER 4

CodeIgniter, es un entorno de desarrollo rápido, con un grupo específico de herramientas para programadores que generan aplicaciones (EllisLab 2011).

2.9.1. Modelo MVC

El MVC o Modelo Vista Controlador es un patrón de arquitectura de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el

controlador, es decir por un lado define los componentes para la representación de la información y por otro lado la interacción del usuario.

Esta arquitectura se encuentra incluso en frameworks modernos, como ser: Ruby on rails, Django, Angular js, etc.

2.9.1.1. Modelo

Por lo general, se encarga de peticiones de información a la base de datos, realizando consultas, actualizaciones, búsquedas, etc. (pero no obligatoriamente).

2.9.1.2. Controlador

Es el intermediario entre la vista y el modelo. Recibe órdenes del usuario y se encarga de solicitar los datos al modelo.

2.9.1.3. Vista

Es la presentación visual de los datos, en una interfaz gráfica que sea entendible por el usuario.

Ventajas

- ✓ Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- ✓ Es sencillo de instalar, basta con subir los archivos al ftp y tocar un archivo de configuración para definir el acceso a la base de datos.
- ✓ Existe abundante documentación en la red.
- ✓ Facilidad de edición del código ya creado.
- ✓ Facilidad para crear nuevos módulos, páginas o funcionalidades.
- ✓ Estandarización del código
- ✓ Separación de la lógica y arquitectura de la web, el MVC.
- ✓ Cualquier servidor que soporte PHP+MySQL sirve para CodeIgniter.

- ✓ CodeIgniter se encuentra bajo una licencia open source, es código libre. CORIAWEB (2020).

2.10. PHP 8.1

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl). Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga (PHP, 2024).

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

2.10.1. Características

El sitio oficial de PHP, destaca las siguientes características:

- ✓ Autenticación HTTP con PHP
- ✓ Manejo de Cookies
- ✓ Manejo de sesiones
- ✓ Subida de ficheros
- ✓ Manejo de conexión con gestores de base de datos
- ✓ Modo seguro

2.10.2. Ventajas de PHP

- ✓ Lenguaje multiplataforma
- ✓ Es libre
- ✓ Documentación amplia
- ✓ Se puede aplicar diferentes técnicas y patrones de programación
- ✓ Tiene variedades en cuanto a extensiones y frameworks

2.11. JAVASCRIPT

Es un lenguaje de programación interpretado, se define como orientado a objetos, se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas (GAUCHAT, 2012)

Características

- ✓ JavaScript está influenciado sobre todo por la sintaxis de Java, pero también de Awk, Perl y Python.
- ✓ JavaScript es case-sensitive (distingue mayúsculas y minúsculas) y utiliza el conjunto de caracteres Unicode.
- ✓ En JavaScript, las instrucciones son llamadas Sentencias y son separadas por punto y coma (;).
- ✓ El lenguaje de scripting es seguro y fiable porque está en claro y hay que interpretarlo.
- ✓ El código de JavaScript se ejecuta en el cliente, por lo que el servidor no es solicitado más de lo debido.

2.12. BOOTSTRAP 5

Bootstrap es un framework de desarrollo web gratuito y de código abierto. Está diseñado para facilitar el proceso de desarrollo de los sitios web responsivos y orientados a los dispositivos móviles, proporcionando una colección de sintaxis para diseños de plantillas. (Deimar, 2023)

Ventajas

- ✓ Es de código abierto, y todo su código actualizado se encuentra en un repositorio de Github.
- ✓ Esta mantenida y actualizado por Twitter.
- ✓ Es compatible con la mayoría de los navegadores (Chrome, Safari, Mozilla, etc.).
- ✓ Se integra con librerías JavaScript.
- ✓ Sus plantillas son de sencilla adaptación responsive.

2.13. MySQL

MySQL es un gestor de bases de datos, y actualmente uno de los más usados y reconocidos del mercado. Especialmente en lo que se refiere a desarrollo web, está clasificada como la base de datos de código abierto más popular del mundo. MySQL es un sistema de gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de educación precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y a su libre distribución en internet bajo licencia GPL le otorgan como beneficios adicionales, contar con un alto grado de estabilidad y un rápido desarrollo. (Pérez, 2014)

Características

- ✓ Escrito en C y C++.
- ✓ Probado con una amplia gama de compiladores diferentes.
- ✓ Funciona en muchas plataformas diferentes.
- ✓ Utiliza un diseño de servidor multicapa con módulos independientes.
- ✓ Proporciona motores de almacenamientos transaccionales y no transaccionales.

2.14. PhpSpreadsheet

PhpSpreadsheet es una biblioteca escrita en PHP puro y ofrece un conjunto de clases que le permiten leer y escribir varios formatos de archivos de hojas de cálculo, como Excel y LibreOffice Calc. (PhpSpreadsheet, 2016)

El requisito para el uso de esta librería es PHP versión 8.1 o posterior para desarrollar usando PhpSpreadsheet. El soporte para las versiones de PHP solo se mantendrá durante un período de seis meses después del final de vida de esa versión de PHP.

El soporte para las versiones de PHP solo se mantendrá durante un período de seis meses después del final de vida de esa versión de PHP.

Instalación

Utilice Composer para instalar PhpSpreadsheet en su proyecto, composer requiere phpoffice/phpspreadsheet o también descarga la documentación y los ejemplos si piensas utilizarlos

2.15. CHAR JS

Chart.js viene con tipificaciones TypeScript incorporadas y es compatible con todos los marcos de JavaScript populares, incluidos React, Vue, Svelte y Angular. Puede utilizar

Chart.js directamente o aprovechar paquetes contenedores bien mantenidos que permitan una integración más nativa con los marcos de su elección. (chartjs.org, 2024).

Chart.js representa elementos de gráficos en un lienzo HTML5 a diferencia de otras bibliotecas de gráficos, en su mayoría basadas en D3.js, que se representan como SVG. La representación de lienzo hace que Chart.js tenga un gran rendimiento, especialmente para grandes conjuntos de datos y visualizaciones complejas que, de otro modo, requerirían miles de nodos SVG en el árbol DOM. Al mismo tiempo, la representación del lienzo no permite el estilo CSS, por lo que tendrás que usar opciones integradas para eso, o crear un complemento personalizado o un tipo de gráfico para representar todo a tu gusto. (chartjs.org, 2024)

Chart.js es muy adecuado para grandes conjuntos de datos. Dichos conjuntos de datos se pueden ingerir de manera eficiente utilizando el formato interno, por lo que puede omitir el análisis y la normalización de datos. Alternativamente, la destrucción de datos se puede configurar para muestrear el conjunto de datos y reducir su tamaño antes de renderizarlo.

2.16. CALIDAD DE SOFTWARE

La calidad de software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia pudiendo medirse después de elaborado el producto.

2.16.1. Estándar ISO/IEC 2126

La Norma ISO/IEC 9126 es un estándar internacional para la evaluación del software que surge debido a la necesidad de un modelo único para expresar la calidad de un software. Fue publicado en 1992 con el nombre de “Information technology – Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software con el propósito de lograr una mayor confiabilidad, mantenibilidad, facilidad de prueba y uso (Pressman, 2013)

2.16.1.1. Características de Calidad de un Software según ISO/IEC 9126: 2001.

- **Funcionalidad:** se evalúa la adecuación, el cumplimiento funcional, idoneidad, corrección, interoperabilidad, conformidad y seguridad de acceso. Por lo que es posible afirmar que la funcionalidad determina la capacidad del software de funcionar en términos de lo que el usuario necesita, de interactuar con otros sistemas y que permita el acceso de diferentes personas pero que cumpla con las regulaciones de las leyes de protección de datos.

Ecuaciones para determinar la funcionalidad.

Tabla 2.1

Valores para Determinar el punto de fusión

Type	Low	Average	High	Total
EI	x3+	x4+	x6=	
EO	x4+	x5+	x7=	
EQ	x7+	x10+	x15=	
ILF	x5+	x7+	x10=	
EIF	x3+	x4+	x6=	

Nota. traducción de la complejidad se muestra la transformación de valores.

(Busquelle, 2010)

Ecuación de Punto de Fusión

$$PF = Cuenta\ total \left[X + Y * \sum FI \right] \quad (1)$$

Ecuación de la Función

$$Funcionalidad = \frac{PF}{PF_{max}} \quad (2)$$

- **Confiabilidad:** se tienen en cuenta aspectos como la capacidad y facilidad de recuperación, la mitigación de fallos, cantidad de tiempo que el software está disponible para su uso y la tolerancia. Por lo que esto tiene en cuenta todo lo relacionado a los fallos que podría dar el producto de software.

Ecuación para determinar la confiabilidad.

Ecuación de la Confiabilidad

$$TMEF = TMDF + TMDR \quad (3)$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Ecuación de la Disponibilidad

$$Disponibilidad = \frac{TMDF}{TMDR} * 100 \quad (4)$$

- **Usabilidad:** mide el grado en que el software es fácil de usar, qué tan intuitivo es, el manejo que el usuario le da al sistema y si este presenta menús sencillos, lectura de textos ágil, cuenta con funciones de forma clara y puntual, entre otros.

Para medir la usabilidad del sistema se utiliza la siguiente ecuación.

Ecuación de la Usabilidad

$$FU = \left[\sum (X_{in}) * 100 \right] \quad (5)$$

Donde:

Xi: Es la sumatoria de valores

n: Es el número de preguntas.

- **Facilidad de mantenimiento:** Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad (Figuroa, 2012). Esto hace que el producto de software sea escalable ya que es posible hacerle mejoras constantes sin que este no tenga ningún problema al querer realizarle alguna modificación o incorporación.

Ecuación de Mantenimiento

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt} \quad (6)$$

Donde:

Mt= Número de módulos de la versión actual

Fc= Número de módulos en la versión actual que se han modificado

Fa= Número de módulos en la versión actual que se han añadido

Fd= Número de módulos de la anterior versión que se han borrado en la versión actual.

- **Portabilidad:** En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos
 - ✓ Adaptabilidad. Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
 - ✓ Facilidad de Instalación. Es el esfuerzo necesario para instalar el software en un ambiente determinado.

- ✓ Conformidad. Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- ✓ Capacidad de reemplazo. Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares. (Figuroa, 2012)

2.17. MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE

2.17.1. Cocomo II

COCOMO II es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software (González, 2014).

Las ecuaciones mostradas, son las utilizadas para los submodelos básico e intermedio, se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P) así como los multiplicadores de esfuerzo (ME). No se incluyen las ecuaciones para el submodelo detallado, por razones de espacio dentro del desarrollo de la propuesta del enfoque pedagógico descrita en este trabajo.

Coeficientes para la estimación de costo

Tabla 2.2

Coeficientes de Modelo COCOMO II

MODO	A	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semi – Orgánico	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.33

Nota: coeficiente de modelo (Boehm, 1981)

SUBMODELO INTERMEDIO

Ecuación de Esfuerzo

$$(E) = a * (KLDC)^b * ME \quad (7)$$

Ecuación de Tiempo

$$(T) = c * E^d \quad (8)$$

Ecuación para Determinar el Número de Personas

$$(P) = \frac{E}{T} \quad (9)$$

Los multiplicadores de esfuerzo, utilizados en la ecuación de esfuerzo del submodelo intermedio, son quince agrupados en cuatro grandes categorías: atributos de producto, atributos de computador, atributos personales y atributos del proyecto. Cada uno de estos multiplicadores de esfuerzo, tiene una valoración que se clasifica en una escala de 6 valores desde “muy bajo”, “bajo”, “nominal”, “alto”, “muy alto” y “extraordinariamente alto”. Estos multiplicadores de esfuerzo ajustan el valor real del esfuerzo

Tabla 2.3

Atributos de Modelo Intermedio COCOMO II

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						

Restricciones de tiempo de ejecución		1,00	1,11	1,30	1,66
Restricción de memoria virtual		1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	0,87	1,00	1,15	1,30	
Tiempo de respuesta	0,87	1,00	1,07	1,15	
Atributos de personal					
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90	
Experiencia en el lenguaje	1,14	1,07	1,00	0,95	
Atributos en el proyecto					
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10

Nota: Atributos del modelo intermedio (Boehm, 1981)

2.18. SEGURIDAD DEL SISTEMA

Cuando hablamos de seguridad en sistemas de información nos referimos a un conjunto de medidas y uso de herramientas para prevenir, resguardar, proteger y reaccionar ante cualquier movimiento que atente contra la información. Con esto, se busca mantener la confidencialidad, mantener íntegros los datos y disponibles según sea necesario.

2.18.1. Estándar ISO 27000

ISO/IEC 27000 es parte de una familia en crecimiento de estándares sobre Sistemas de Gestión de la Seguridad de la Información (SGSI) de ISO/IEC, el ISO 27000 series. ISO/IEC 27000 es un grupo de estándares internacionales titulados: Tecnología de la Información - Técnicas de Seguridad - Sistemas de Gestión de la Seguridad de la Información - Visión de conjunto y vocabulario. Tiene como fin ayudar a organizaciones de todo tipo y tamaño a implementar y operar un Sistema de Gestión de la Seguridad de la Información (SGSI). La norma ISO/IEC 27000 fue preparada por el Comité Técnico conjunto ISO/IEC JTC 1 Tecnología de la Información, SC 27 Técnicas de Seguridad.

ISO/IEC 27000 proporciona:

- ✓ Una visión general de normas sobre Sistemas de Gestión de la Seguridad de la Información (SGSI).
- ✓ Una introducción a los Sistemas de Gestión de la Seguridad de la Información (SGSI).
- ✓ Una breve descripción del proceso para Planificar - Hacer - Verificar - Actuar (Plan- Do - Check - Act, PDCA).
- ✓ Los términos y las definiciones utilizadas en la familia de normas Sistemas de Gestión de la Seguridad de la Información (SGSI). Wikipedia (ISO/IEC 27000, 2018).

CAPÍTULO III

MARCO APLICATIVO

3. CAPÍTULO III- MARCO APLICATIVO

3.4. INTRODUCCIÓN

Este capítulo tiene como finalidad describir el análisis de requerimientos, diseño, codificación, pruebas y la implementación del sistema web para el control y la administración de los registros de la empresa “Cerámica Titan Bolivia”.

3.5. REQUERIMIENTOS DEL SISTEMA

A continuación, detallamos de forma general los requerimientos funcionales de la empresa Cerámica Titan Bolivia.

3.5.1. *Requerimientos Funcionales*

los requerimientos funcionales del proyecto se detallan a continuación:

Tabla 3.1

Requerimientos Funcionales

ÍTEM	FUNCIÓN
1	Administración de usuarios, debe ser posible la creación de nuevos usuarios, modificación y bloqueo según se requiera. Dichos procesos solo deberán ser realizado por el administrador.
2	Administración de almacenes, se debe realizar la creación de nuevos almacenes, modificación. Las modificaciones solo deben realizar el administrador
3	Administración de cajas, solo el administrador debe crear cajas, asignar cajeros, realizar modificaciones, dar de baja a los cajeros.

4	Artículos, se debe realizar la creación de nuevos artículos, modificación. Estos procesos lo pueden realizar el jefe de almacén y el administrador
5	Proveedores, debe ser posible el registro, actualización. Estos procesos lo pueden realizar jefe de almacén, responsables de compras y el administrador
6	categorías lo pueden agregar, modificar jefe de almacén y el administrador
7	Los clientes pueden agregar al sistema, modificar los cajeros, tesorería, responsables de compras, jefes de almacén y el administrador
8	solicitudes de compras, solo pueden generar nuevas solicitudes de compras el administrador, responsable de compras
9	Aprobar solicitudes de compra, esta operación solo puede realizar tesorería y el administrador
10	Gestión de almacén, estos movimientos solo pueden realizar el administrador y el jefe de almacén
11	ingresos, egresos, apertura y cierre en caja. Estos movimientos solo pueden realizar los cajeros y el administrador
12	Generar reportes de cierres, reporte de existencias en almacenes, visualizar panel de control. Dichos procesos pueden realizar tesorería, jefe de almacén y el administrador

3.5.2. *Requerimientos No Funcionales*

- ❖ El sistema debe tener login
- ❖ El sistema debe ser compatible con los principales navegadores web (Chrome, Firefox, Safari, Edge)
- ❖ Debe ser posible implementar el sistema en diferentes sistemas operativos, como Windows, Linux y macOS

3.6. IDENTIFICACIÓN DE ACTORES.

Implementando la metodología UWE detallamos la lista de roles con sus respectivas funciones de acuerdo a la organización de la empresa

3.3.1 Responsable de Compras: son las personas encargadas de realizar la gestión de compras de los artículos de los distintos proveedores con las cuales cuenta la empresa Cerámica Titan Bolivia

3.3.2 Tesorería: responsable de aprobar o rechazar las solicitudes de compras.

3.3.3 Cajero: personas responsables de realizar los movimientos en caja como desembolso de efectivo para la compra de artículos.

3.3.4 Jefe de almacén: son las personas responsables de gestionar el almacén, registrar entradas y salida de los artículos

3.7. DIAGRAMA DE CASO DE USO GENERAL DEL SISTEM

Figura 3.1

Diagrama de Caso de Uso General



Tabla 3.2

Detalle Administración de Usuarios

Nombre	Administrar Almacén
Actor	Administrador
Descripción	Se realizará las operaciones de crear nuevos Almacenes, modificar, buscar, asignar y suspender a responsables de almacén

Figura 3.2

Caso de uso administración de almacén

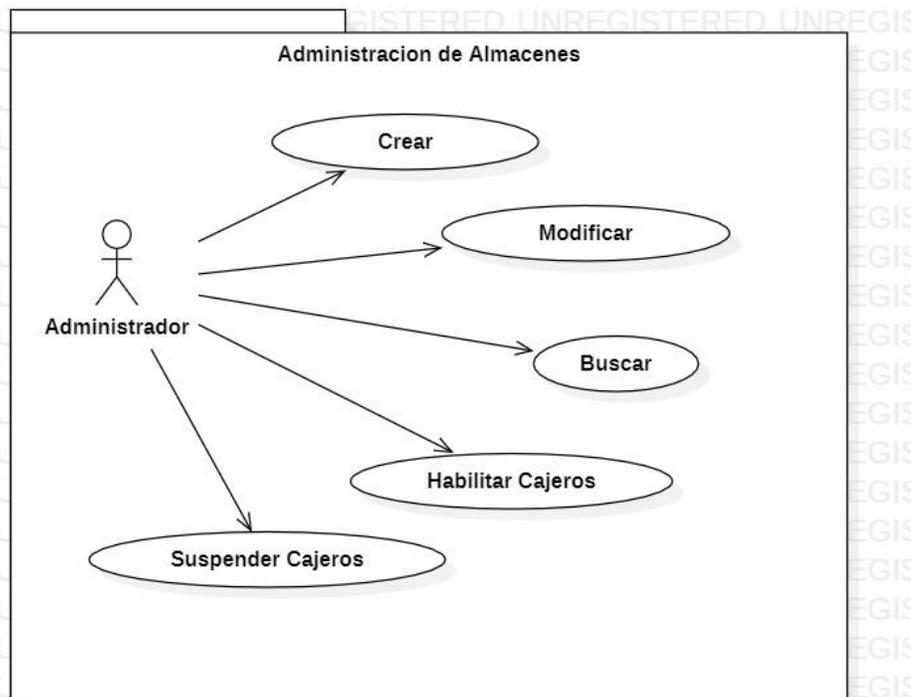


Tabla 3.3*Descripción de Gestión de Artículos*

Nombre	Gestión de Artículos
Actor	Administrador, jefe de Almacén
Descripción	Se realizará las operaciones de registrar, modificar artículos, buscar, ingresar nuevas compras, salidas en función al almacén

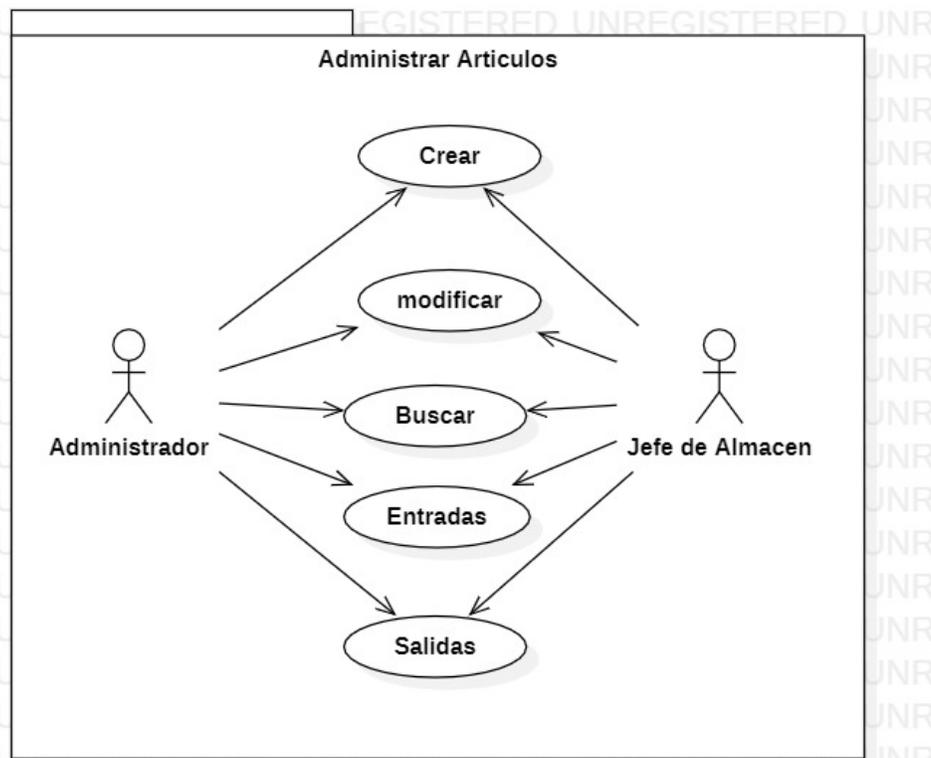
Figura 3.3*Caso de Uso Administración de Artículos*

Tabla 3.4*Detalle Administración de Caja*

Nombre	Administrar Caja
Actor	Administrador
Descripción	Se realizará las operaciones de crear nuevos Cajas, modificar, buscar, asignar y suspender a Cajeros

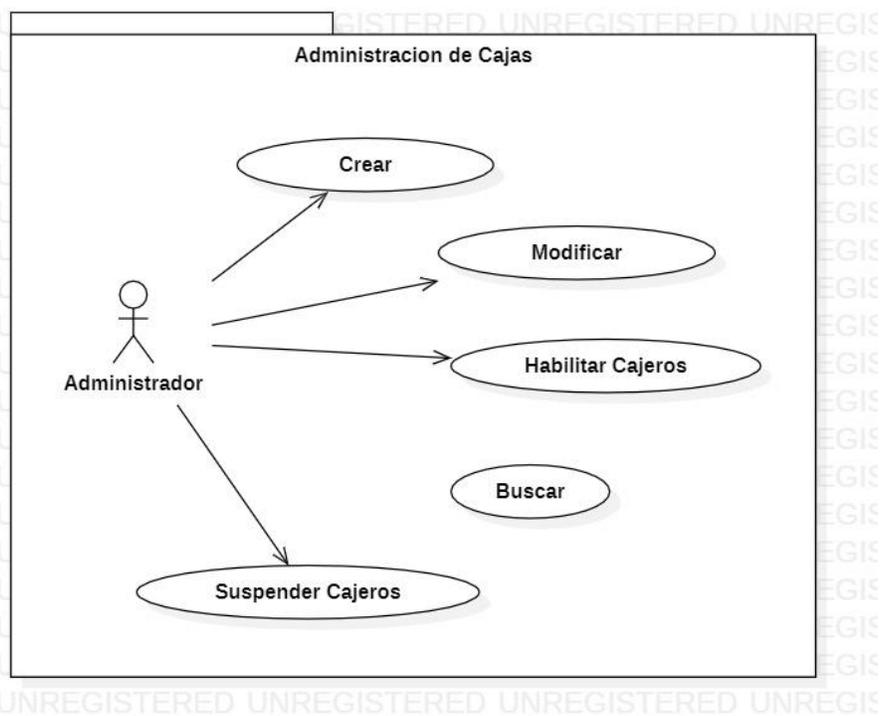
Figura 3.4*Caso de uso administración de caja*

Tabla 3.5

Detalle de Gestión de Proveedores

Nombre	Gestión de Proveedores
Actor	Administrador, jefe de Almacén, Jefe de compras
Descripción	Se realizará las operaciones de registrar, modificar, buscar proveedores

Figura 3.5

Caso de uso gestión de proveedores

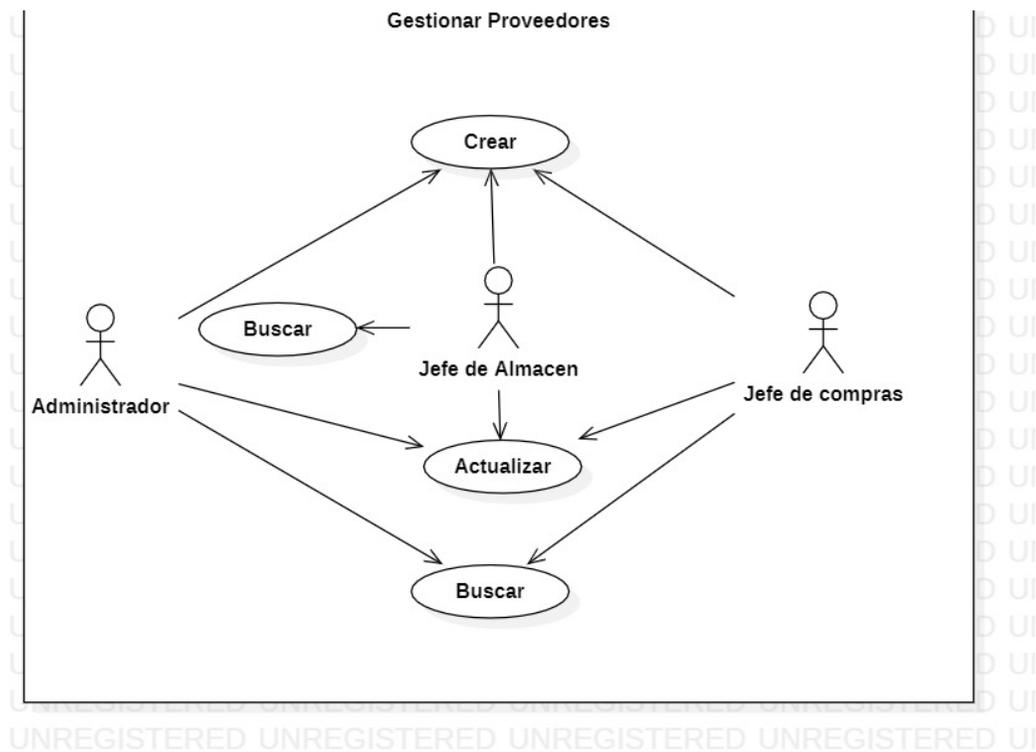


Tabla 3.6

Gestión de Categorías

Nombre	Administrar Categorías
Actor	Administrador, jefe de Almacén
Descripción	Se realizará las operaciones de registrar, modificar, buscar categorías

Figura 3.6

Caso de Uso Administración de Categorías

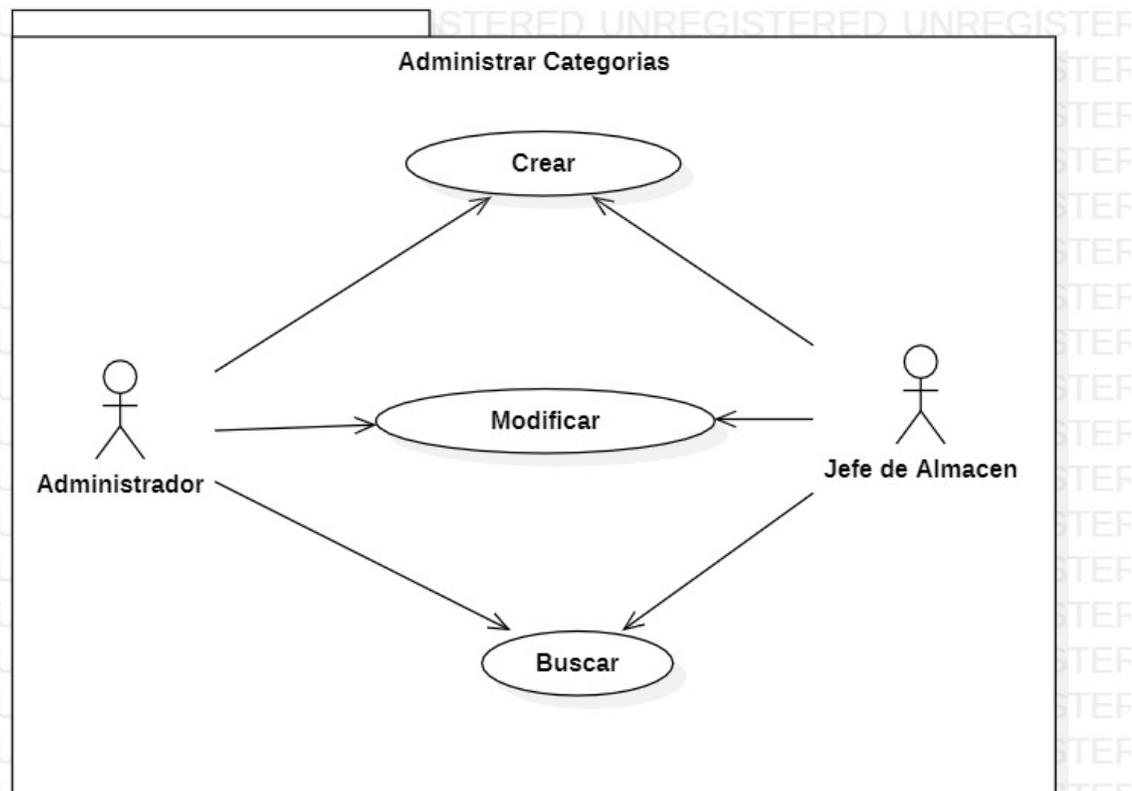


Tabla 3.7

Gestión de Clientes

Nombre	Administrar Clientes
Actor	Administrador, jefe de Almacén, Tesorería, jefe de compras
Descripción	Se realizará las operaciones de registrar, modificar, búsqueda de clientes

Figura 3.7

Caso de Uso Administración de Clientes

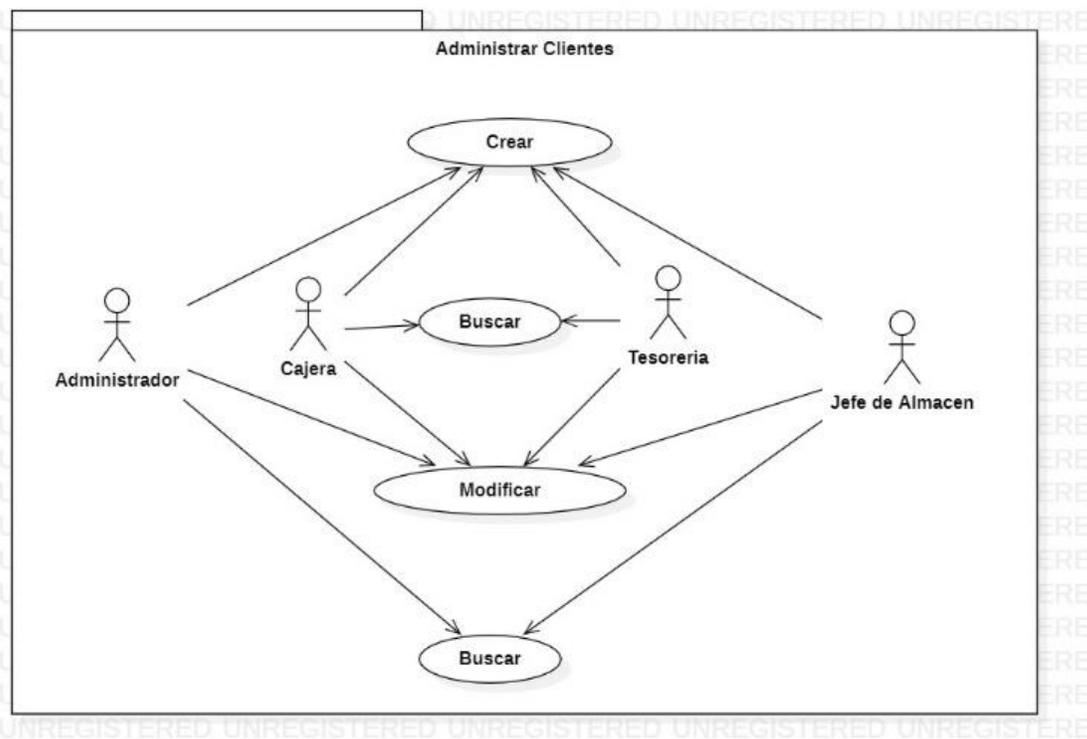


Tabla 3.8

Gestión de Destinos

Nombre	Administrar Destinos
Actor	Administrador, jefe de Almacén
Descripción	Se realizará las operaciones de registrar, modificar, buscar destinos

Figura 3.8

Caso de uso Administración de Destino

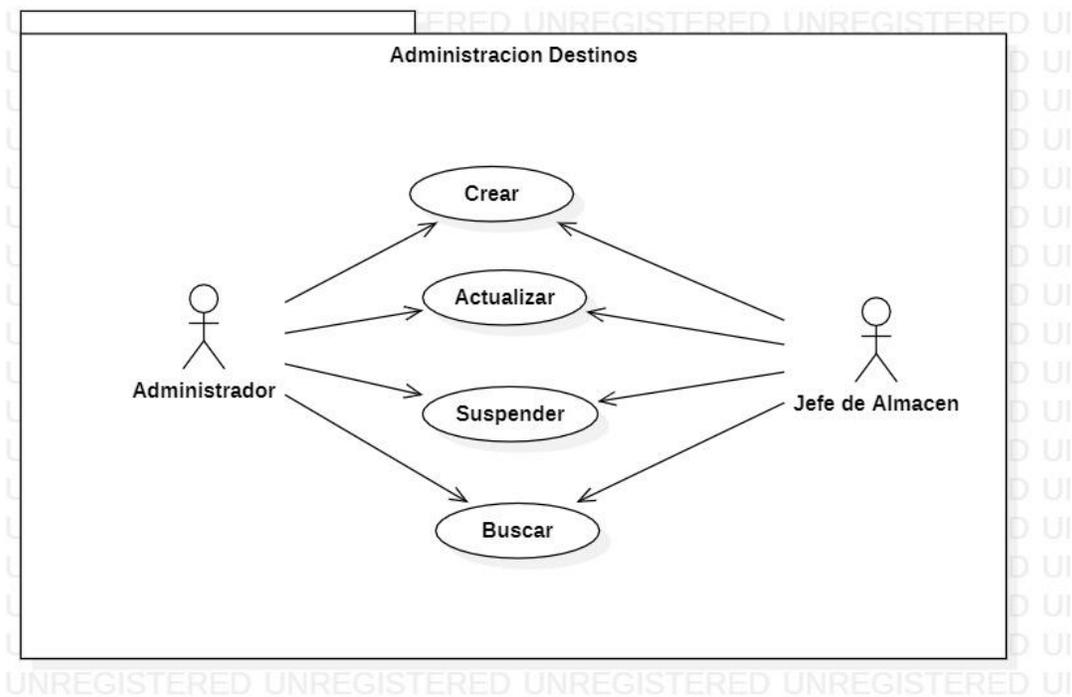


Tabla 3.9

Detalle Solicitud de Compras

Nombre	Solicitud de compras
Actor	Administrador, jefe de compras y tesorería
Descripción	Se realizará las operaciones de generar solicitud, aprobar, rechazar y buscar

Figura 3.9

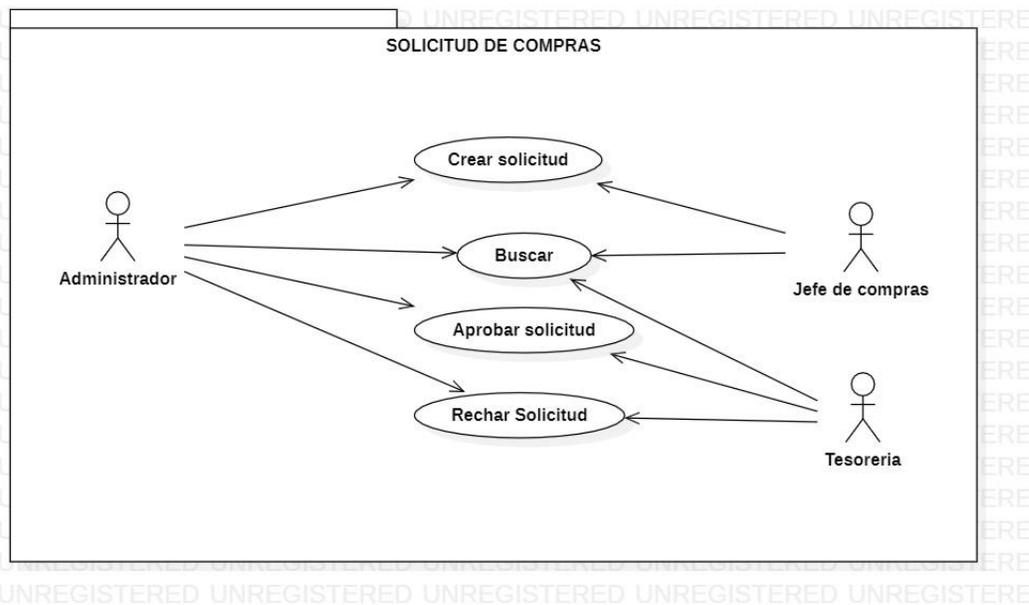
Caso de Uso Solicitud de Compra

Tabla 3.10

Gestión de Movimientos en Caja

Nombre	Gestión de movimientos caja
Actor	Administrador, Cajero
Descripción	Se realizará las operaciones de registrar ingresos, egresos, apertura, cierre y buscar

Figura 3.10

Caso de Uso Administración de Caja

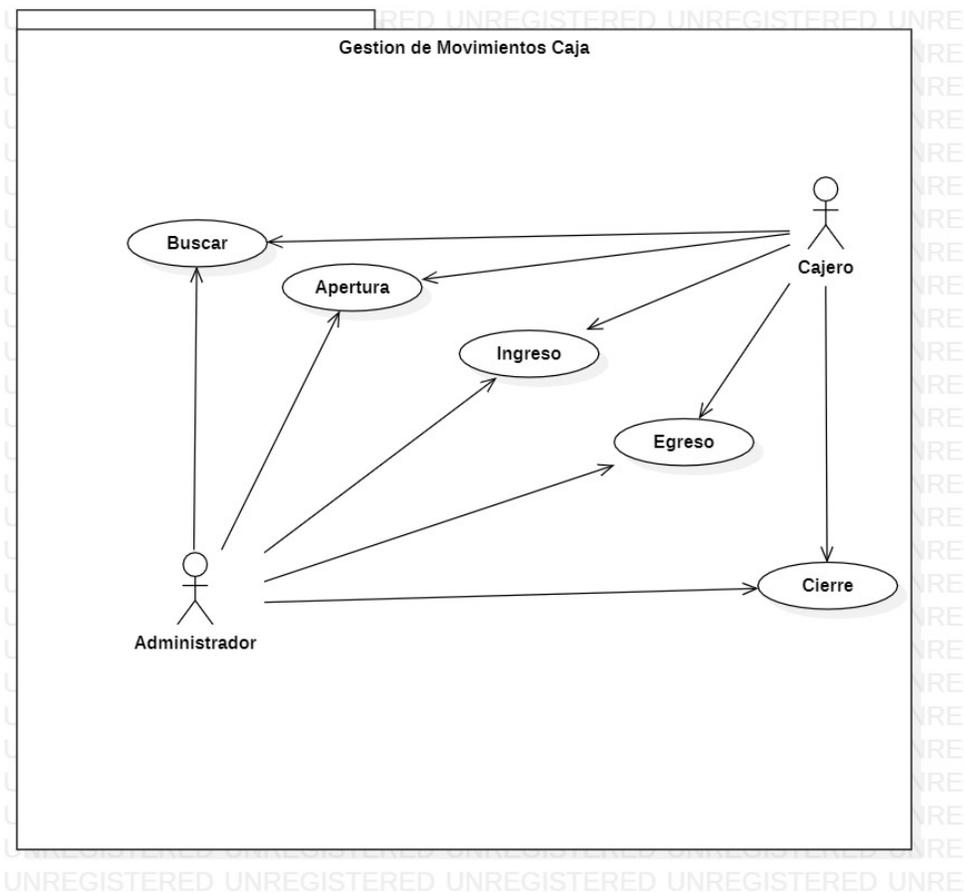


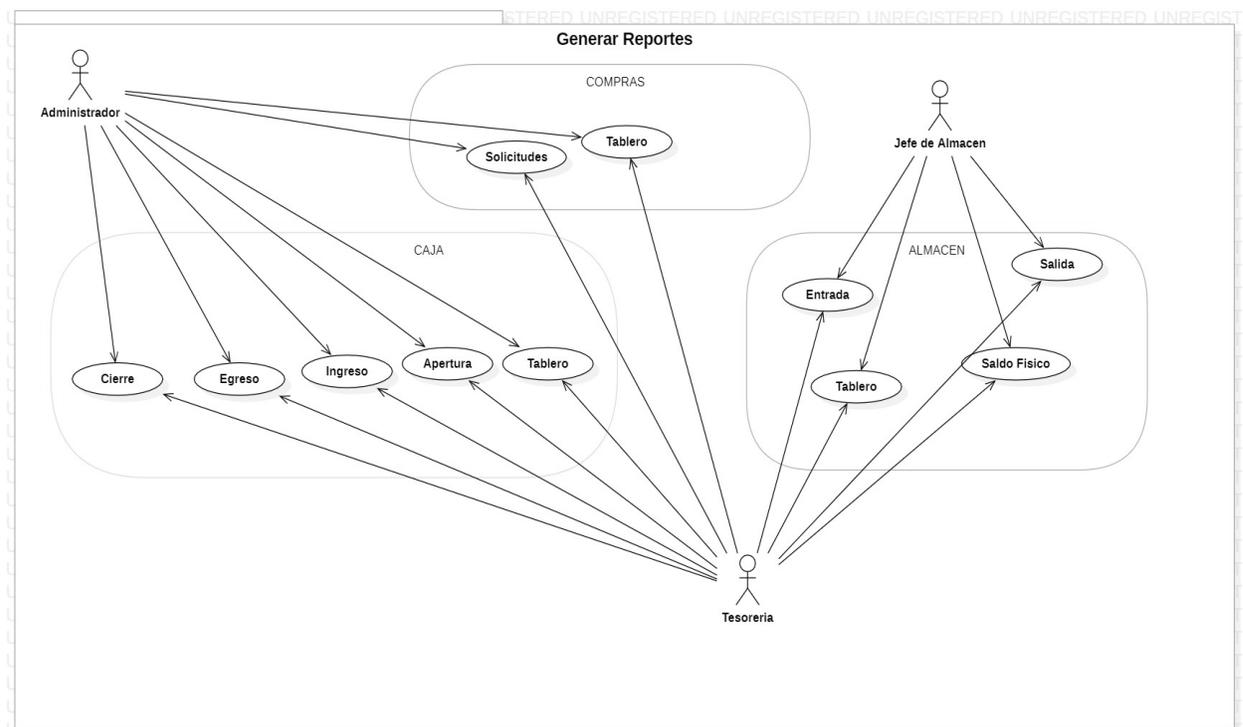
Tabla 3.11

Gestión de Reportes

Nombre	Reportes
Actor	Administrador, Tesorería, jefe de Almacén
Descripción	Se realizará las operaciones de ver reportes graficos, reportes en pdf y Excel

Figura 3.11

Caso de uso administración de reportes

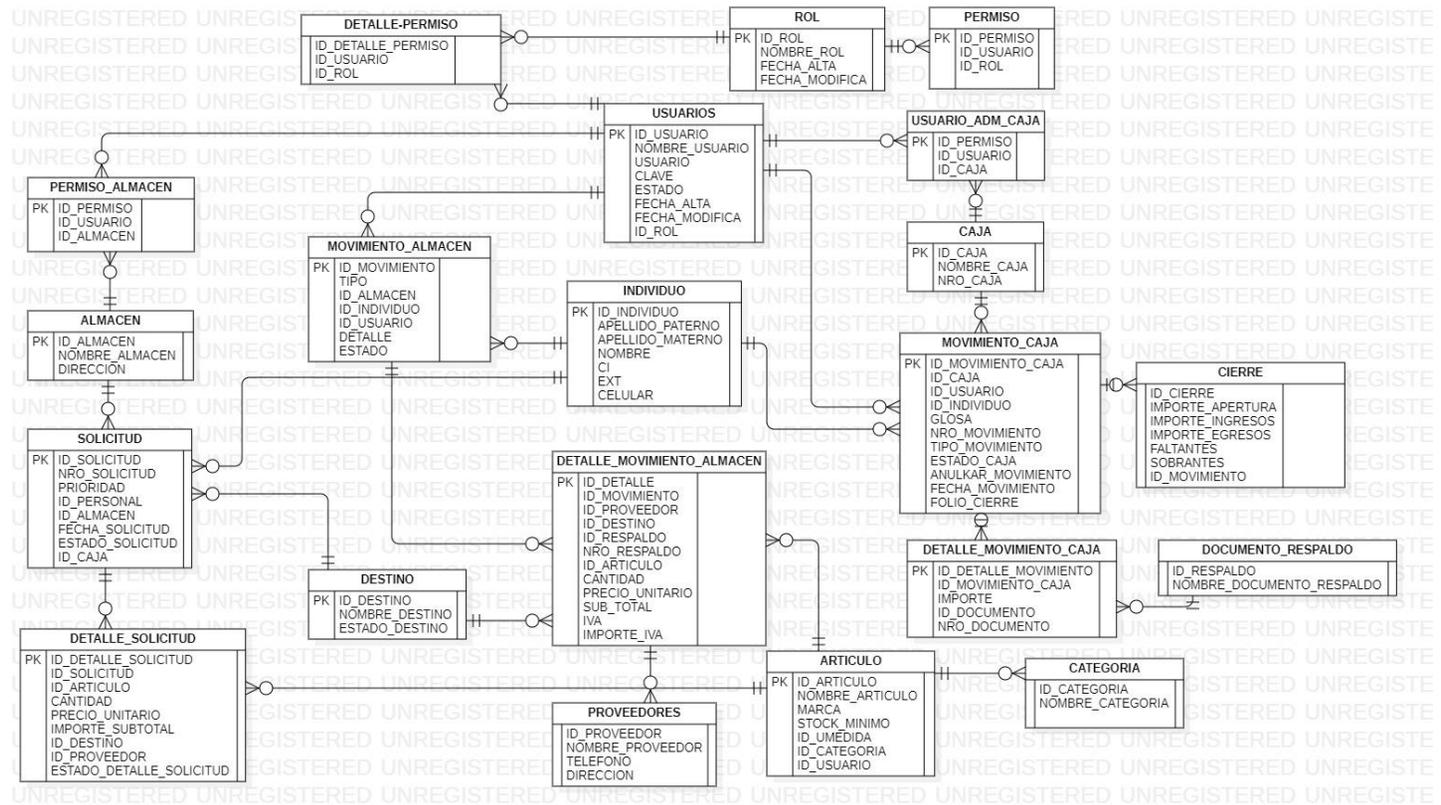


3.8. MODELO RELACIONAL

El modelo relacional, para el modelado y la gestión de bases de datos, es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos (FRANK, 1970)

Figura 3.12

Modelo Relacional del Sistema

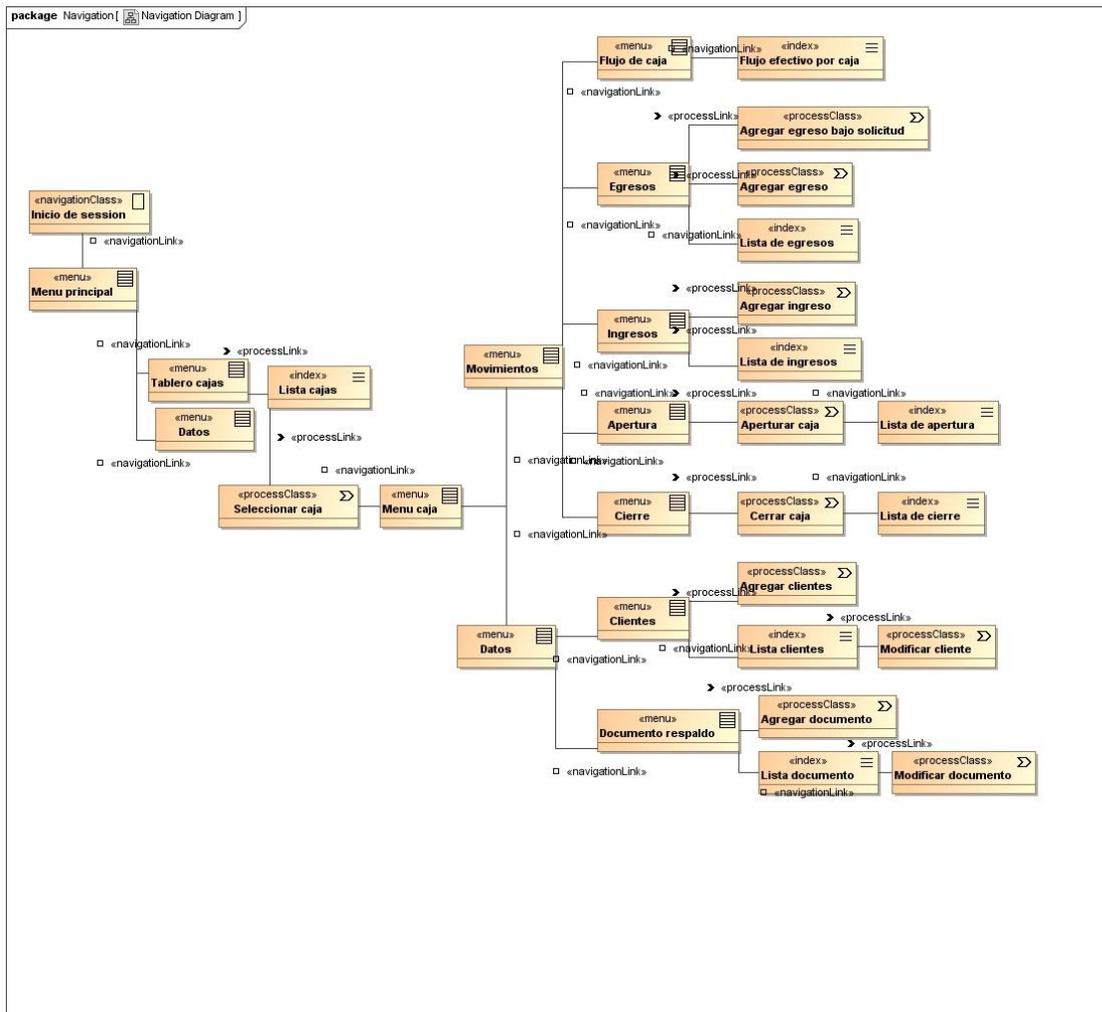


3.9. DIAGRAMAS DE NAVEGACIÓN

3.9.1. Diagrama de Navegación Modulo Caja

Figura 3.13

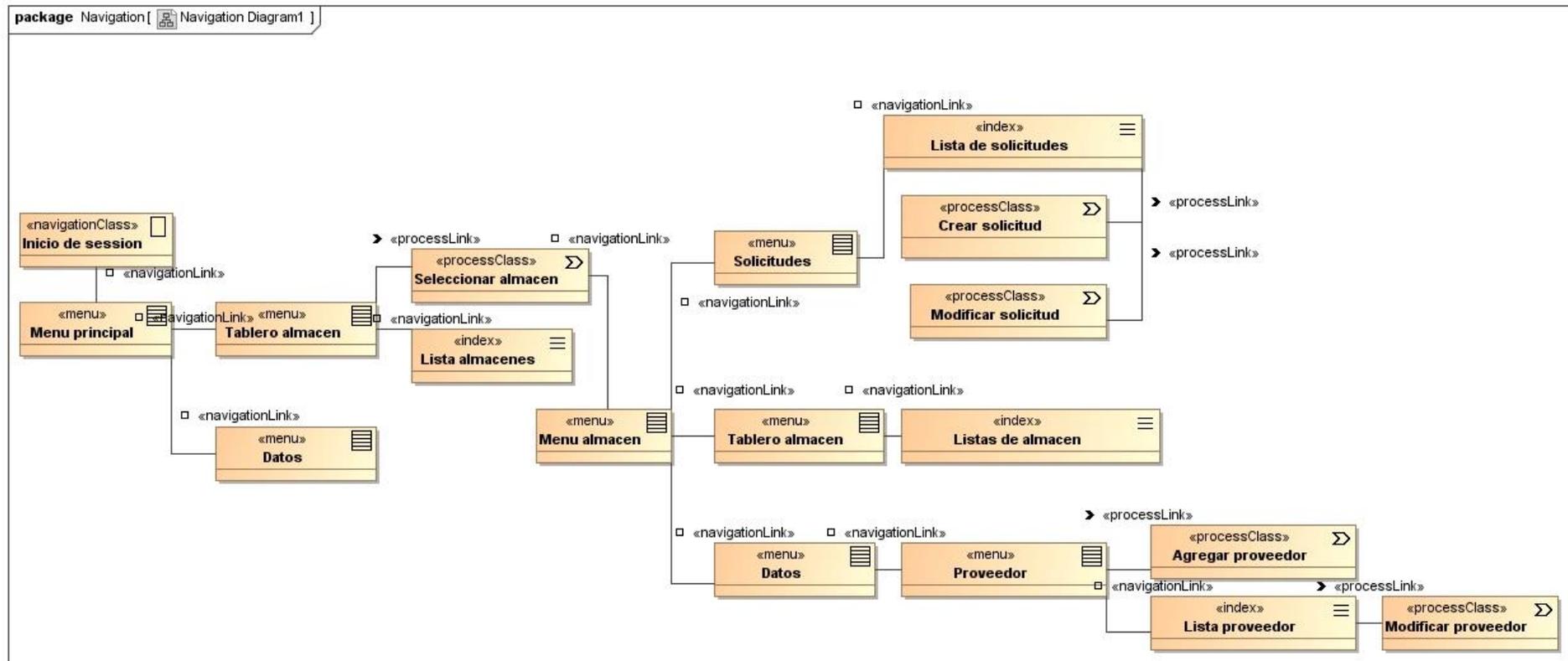
Diagrama de Navegación Módulo de Caja



3.9.2. Diagrama de Navegación Modulo Compras

Figura 3.14

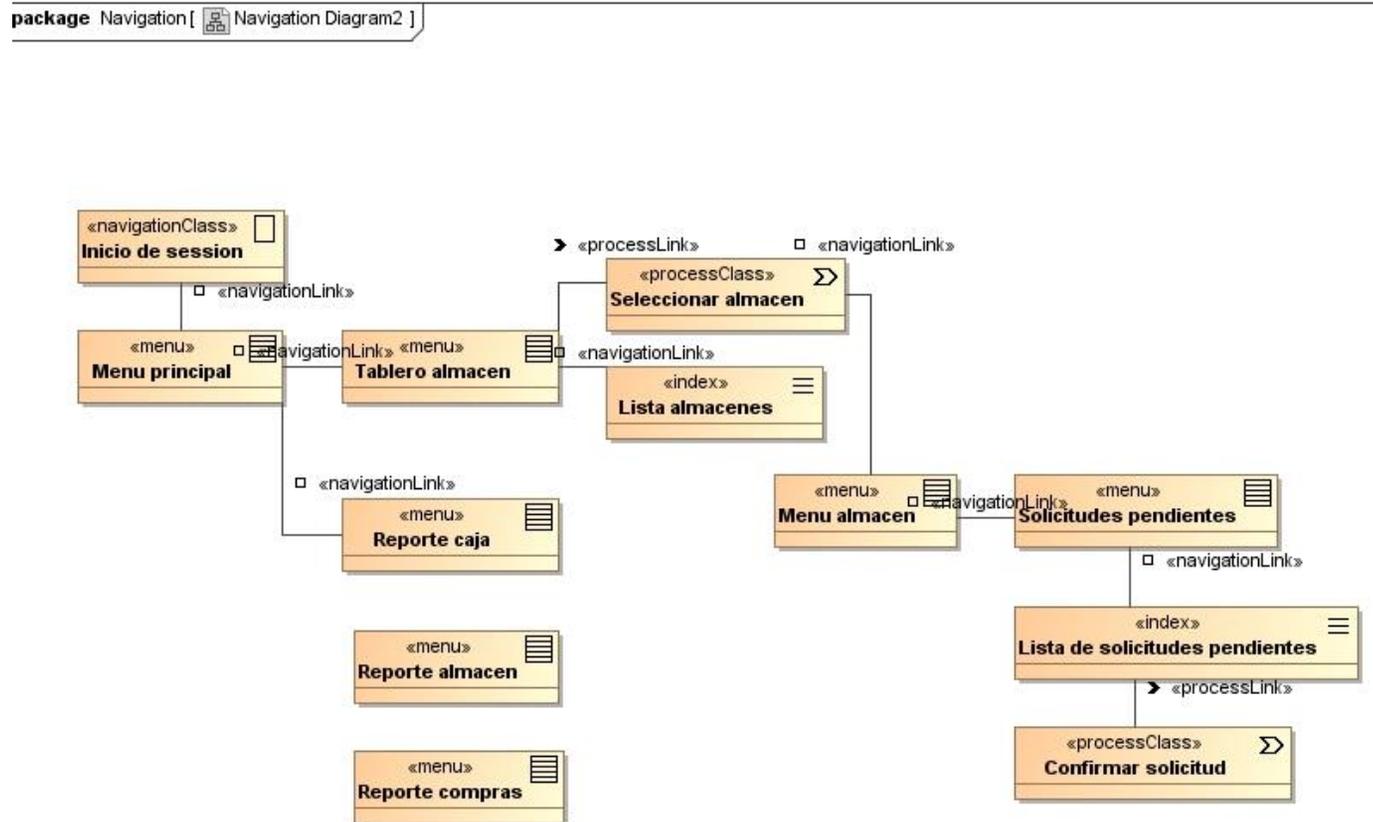
Diagrama de Navegación Modulo Compras



3.9.3. Diagrama de Navegación Modulo Tesorería

Figura 3.15

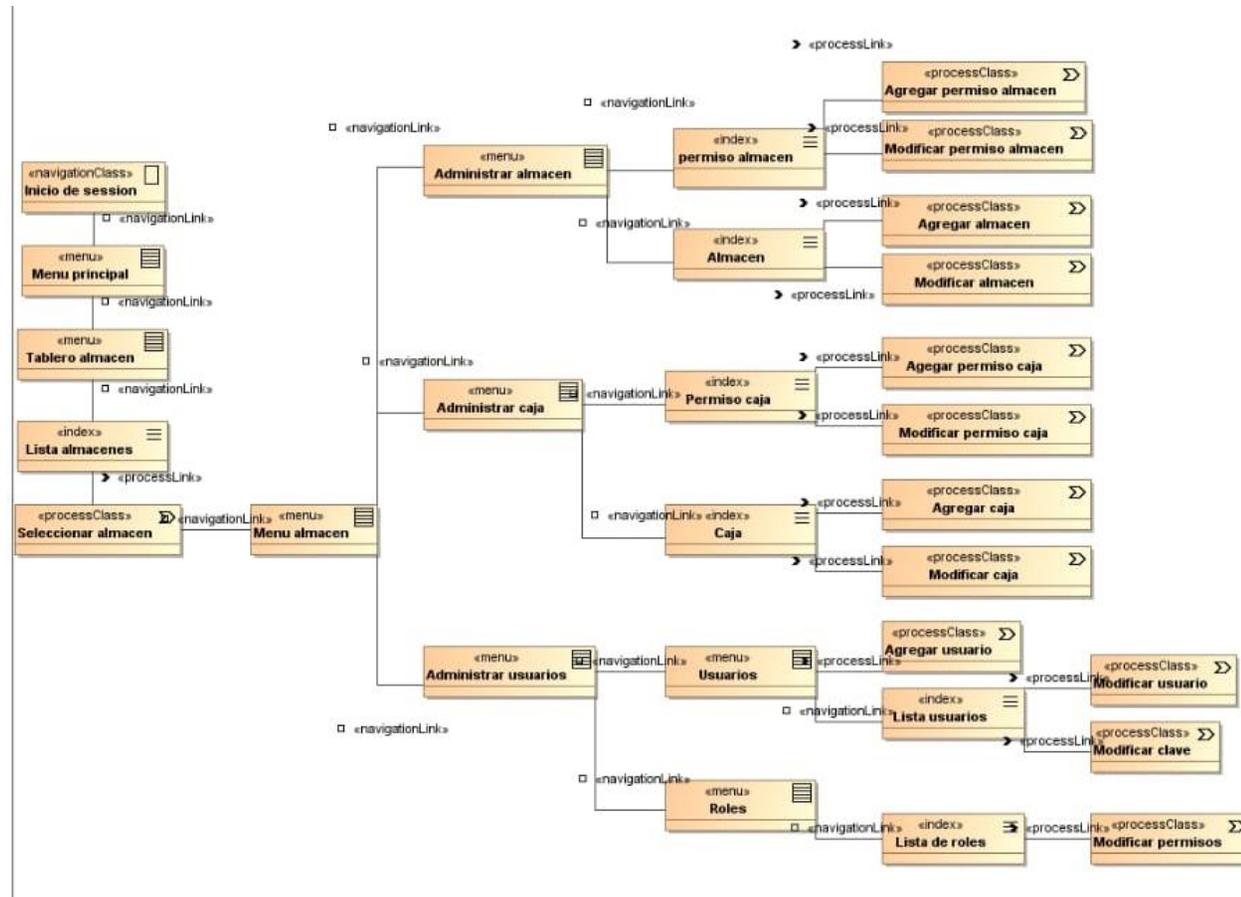
Diagrama de Navegación Modulo Compras



3.9.4. Diagrama de Navegación Administrar Usuarios, Cajas y Almacén

Figura 3.16

Diagrama de Navegación Usuarios, Cajas y Almacenes



3.10. CODIFICACIÓN

Figura 3.17

Código Fuente Login

```

1  public function valida()
2  {
3
4      $recaptchaResponse = $this->request->getPost('g-recaptcha-response');
5
6      $url = 'https://www.google.com/recaptcha/api/siteverify';
7      $data = [
8          'secret' => env('RECAPTCHA_SECRET_KEY'),
9          'response' => $recaptchaResponse
10     ];
11
12     $options = [
13         'http' => [
14             'header' => "Content-type: application/x-www-form-urlencoded\r\n",
15             'method' => 'POST',
16             'content' => http_build_query($data)
17         ]
18     ];
19
20     $context = stream_context_create($options);
21     $result = file_get_contents($url, false, $context);
22     $response = json_decode($result);
23
24     if ($response->success) {
25         if ($this->request->getMethod() == "post" && $this->validate($this->reglaslogin)) {
26             $usuario = $this->request->getPost('usuario');
27             $clave = $this->request->getPost('clave');
28             $datosUsuario = $this->usuarios->datosSession($usuario);
29             if ($datosUsuario != null) {
30                 if (password_verify($clave, $datosUsuario['clave'])) {
31                     $datosSesion = [
32                         'id_usuario' => $datosUsuario['id_usuario'],
33                         'nombre_usuario' => $datosUsuario['nombre_usuario'],
34                         'rol' => $datosUsuario['nombre_rol'],
35                         'id_rol' => $datosUsuario['id_rol']
36                     ];
37
38                     $session = session();
39                     $session->set($datosSesion);
40                     return redirect()->to(base_url() . '/usuarios');
41                 } else {
42                     $data['error'] = "Las contraseñas no coinciden";
43                     echo view('login', $data);
44                 }
45             } else {
46                 $data['error'] = "El usuario no existe";
47                 echo view('login', $data);
48             }
49         } else {
50             $data = ['validation' => $this->validator];
51
52             echo view('login', $data);
53         }
54     } else {
55         // reCAPTCHA inválido, manejar el error
56         return redirect()->back()->with('error', 'reCAPTCHA verification failed');
57     }
58 }
59

```

3.10.1. Implementación del Sistema

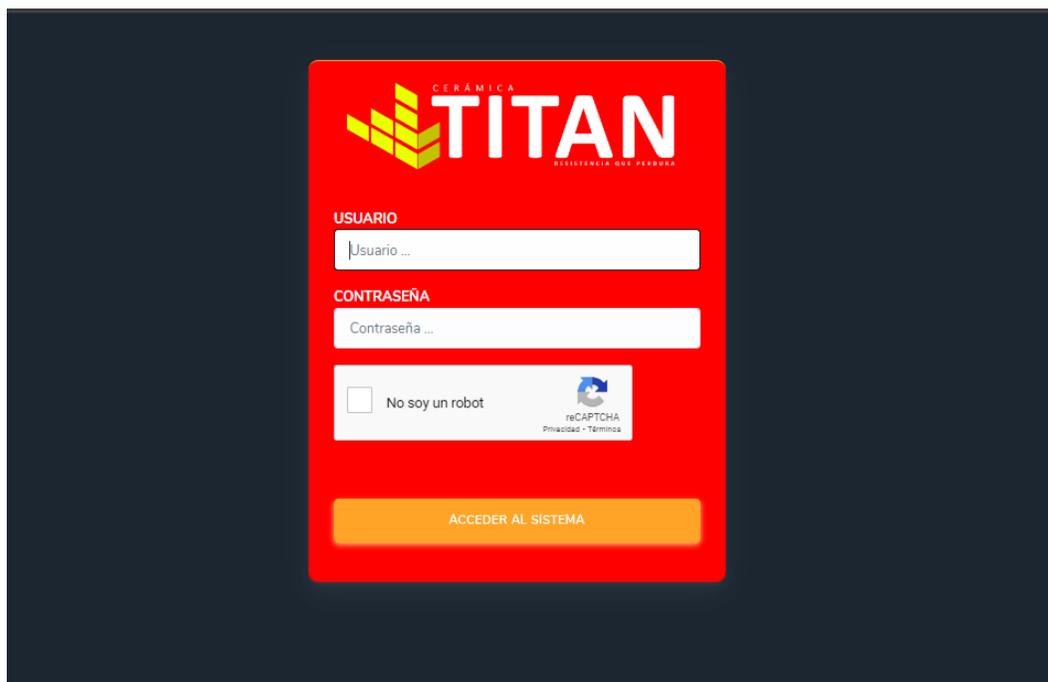
En esta etapa se realiza puesta en marcha del sistema

3.10.1.1. Interfaz Inicio Sesión.

Se deberá ingresar al sistema con el usuario, contraseña y presionando el check box no soy robot en caso de que no se encuentre registrado como usuario en el sistema se re direccionará a inicio de sesión.

Figura 3.18

Inicio de Sesión



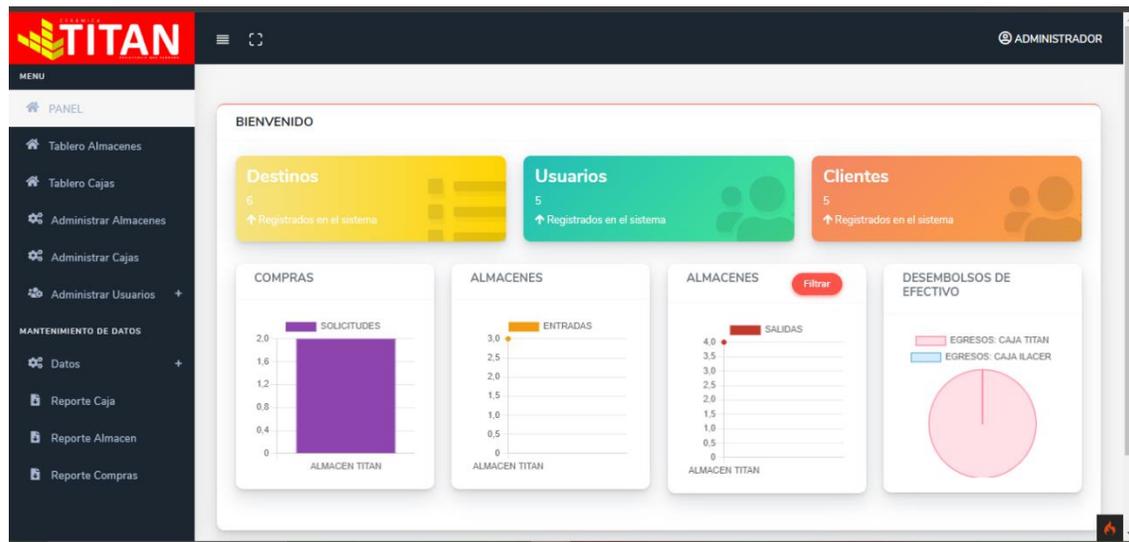
The image shows a login form for 'CERÁMICA TITAN'. The form is contained within a red rectangular box. At the top of the box is the company logo, which consists of a stylized 'T' made of yellow and green blocks, followed by the word 'TITAN' in large white letters and 'CERÁMICA' in smaller letters above it. Below the logo, there are two input fields: one for 'USUARIO' and one for 'CONTRASEÑA'. Below these fields is a reCAPTCHA checkbox with the text 'No soy un robot' and a small robot icon. At the bottom of the form is an orange button with the text 'ACCEDER AL SISTEMA'.

3.10.1.2. Panel de Inicio.

Ingresando al sistema observamos el panel de inicio en el sistema.

Figura 3.19

Panel de Inicio del Sistema



3.10.1.3. Administración de Usuarios.

en el siguiente formulario se observa el registro de todos los usuarios en el cual puedo agregar, modificar y buscar

Figura 3.20

Administración de Usuarios

The screenshot shows the 'REGISTRO DE USUARIOS' interface. It includes a 'Nuevo Usuario' button, a search bar, and a table with 10 entries. The table has columns for ITEM, USUARIO, ROL, ESTADO, and OPERACIONES. All users are currently 'Activo'.

ITEM	USUARIO	ROL	ESTADO	OPERACIONES
1	ADMINISTRADOR	ADMINISTRADOR	Activo	[editar] [Clave]
2	CAJA	CAJA	Activo	[editar] [Clave]
3	ALMACEN	ALMACEN	Activo	[editar] [Clave]
4	COMPRAS	COMPRAS	Activo	[editar] [Clave]
5	TESORERIA	TESORERIA	Activo	[editar] [Clave]

3.10.1.4. Administración de Cajas.

En este panel se puede crear, modificar y buscar cajas. También se crea, modifica y busca cajeros

Figura 3.21

Administración de Cajas

The screenshot displays the 'Administración de Cajas' interface. On the left is a dark sidebar menu with options like 'Tablero Almacenes', 'Tablero Cajas', 'Administrar Almacenes', 'Administrar Cajas', 'Administrar Usuarios', and 'MANTENIMIENTO DE DATOS'. The top navigation bar shows the 'TITAN' logo and the user role 'ADMINISTRADOR'. The main content area is divided into two panels:

- PERMISO CAJA:** A table listing permissions for three items. Each row includes an 'ITEM' number, 'USUARIO' (ADMINISTRADOR), 'CAJA' name, and 'OPERACIONES' (ACTIVO and a delete icon). A 'Nuevo Responsable Caja' button is at the top right.
- CAJA:** A table listing three cash registers. Each row includes an 'ITEM' number, 'CAJA' name, 'NRO CAJA' (1, 2, 3), and 'OPERACIONES' (delete icon). A 'Nuevo Caja' button is at the top right.

Both panels include a search bar, a 'Show 10 entries' dropdown, and pagination controls at the bottom.

3.10.1.5. Administración de Almacenes.

En este panel puede realizar la creación, modificación y búsqueda de responsables de almacén

Figura 3.22

Administración de Almacenes

The screenshot displays the 'Administración de Almacenes' interface. On the left is a dark sidebar menu with options like 'Tablero Cajas', 'Administrar Almacenes', 'Administrar Cajas', 'Administrar Usuarios', and 'MANTENIMIENTO DE DATOS'. The top navigation bar shows the 'TITAN' logo and the user role 'ADMINISTRADOR'. The main content area is divided into two panels:

- PERMISO ALMACEN:** A table listing permissions for three warehouses. Each row includes an 'ITEM' number, 'USUARIO' (ADMINISTRADOR), 'ALMACEN' name, and 'OPERACIONES' (ACTIVO, SUSPENDIDO, or delete icon). A 'Nuevo Responsable Almacen' button is at the top right.
- ALMACENES:** A table listing five warehouses. Each row includes an 'ITEM' number, 'NOMBRE', 'DIRECCION', and 'OPERACIONES' (delete icon). A 'Nuevo Almacen' button is at the top right.

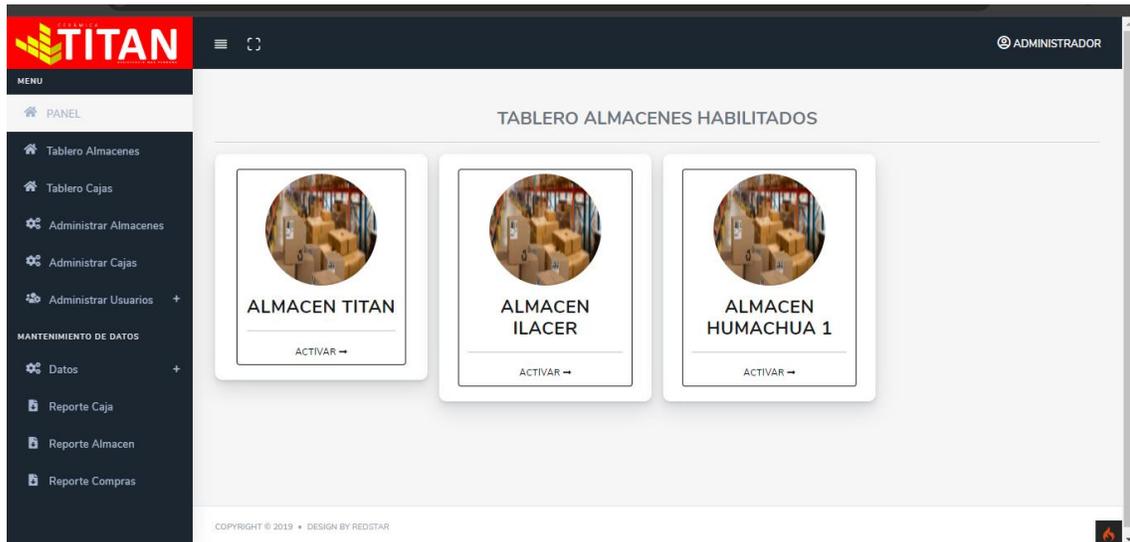
Both panels include a search bar, a 'Show 10 entries' dropdown, and pagination controls at the bottom.

3.10.1.6. Tablero de Almacén.

Se observa los almacenes que tiene habilitado el usuario en el sistema.

Figura 3.23

Almacenes Habilitados en el Sistema



3.10.1.7. Formulario de Solicitud.

Al realizar una solicitud de compra el sistema nos muestra la siguiente interfaz.

Figura 3.24

Formulario de Solicitud de Compra

The screenshot shows the 'Formulario de Solicitud de Compra' (Purchase Request Form) in the TITAN system. The form is titled 'ALMACEN TITAN' and includes a sidebar menu. The main form area contains several input fields and a table. At the top, there is a red note: '(*) Campos Necesarios'. The form fields include:

- NOMBRE ARTICULO * (dropdown menu with 'Seleccione')
- UNIDAD MEDIDA * (text input)
- CANTIDAD SOLICITADA * (text input)
- PROVEEDOR * (dropdown menu with 'SELECCIONAR PROVEEDOR')
- PRECIO UNIDAD * (text input)
- DESTINO * (dropdown menu with 'SELECCIONAR DESTINO')
- TOTAL * (text input)

 Below these fields are two buttons: '← ATRAS' and '+ AGREGAR'. A table is displayed with the following columns: ITEM, NOMBRE ARTICULO, U. MEDIDA, PROVEEDOR, CANTIDAD, PU, TOTAL, and OPERACIONES. At the bottom, the 'IMPORTE TOTAL(BS)' is shown as '0' in a text box, with a 'TERMINAR PROCESO' button below it. The top right corner shows the user role as 'ADMINISTRADOR'.

3.10.1.8. Movimientos de Almacén.

En la interfaz podemos realizar las entradas y salidas de artículos en almacén.

Figura 3.25

Movimientos en Almacén de Artículos

The screenshot shows the 'MOVIMIENTOS' section of the TITAN system. The interface includes a sidebar menu with options like 'Tablero Almacenes', 'Tablero Cajas', 'Administrar Almacenes', 'Solicitudes', 'Solicitudes Pendientes', 'Movimientos', 'Stock', 'Administrar Cajas', and 'Administrar Usuarios'. The main content area displays a table with columns: ITEM, ANULAR, FECHA, T. MOVIMIENTO, RECIBIDO DE, and DETALLE. The table is currently empty, showing 'No data available in table'. There are search and pagination controls.

3.10.1.9. Stock en Almacén.

En la interfaz se visualizará las existencias de artículos en almacén.

Figura 3.26

Existencias Físicas en Almacén

The screenshot shows the 'STOCK' section of the TITAN system. The interface includes a sidebar menu with options like 'Tablero Almacenes', 'Tablero Cajas', 'Administrar Almacenes', 'Solicitudes', 'Solicitudes Pendientes', 'Movimientos', 'Stock', 'Administrar Cajas', and 'Administrar Usuarios'. The main content area displays a table with columns: ITEM, ARTICULO, MARCA, UNIDAD MEDIDA, STOCK MINIMO, and SALDO FISICO. The table shows three items:

ITEM	ARTICULO	MARCA	UNIDAD MEDIDA	STOCK MINIMO	SALDO FISICO
1	CORREA B-50	BANDO	UNIDADES	5	90
2	GRAMPAS 1/2	NINGUNO	UNIDADES	50	200
3	ALAMBRE DE AMARRO	NINGUNO	metros	100	50

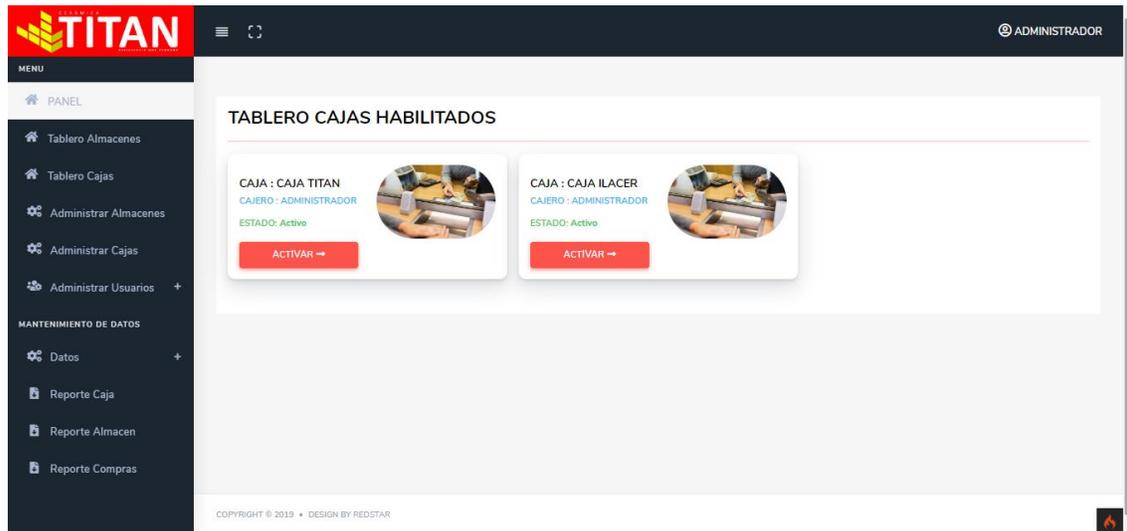
There are search and pagination controls.

3.10.1.10. Tablero de Cajas.

En el tablero de cajas se visualiza todas las cajas habilitadas en función al cajero.

Figura 3.27

Tablero de Cajas



3.10.1.11. Flujo de Efectivo por Caja.

El flujo de efectivo nos muestra los ingresos, egresos y aperturas de las cajas existentes dentro del sistema.

Figura 3.28

Flujo de Caja

The screenshot shows the 'FLUJO DE EFECTIVO' section in the TITAN system. The view is set to 'Flujo de caja'. Two tables are displayed, one for 'CAJA TITAN' and one for 'CAJA ILACER'. Each table shows financial metrics: SALDO INICIAL, INGRESOS, EGRESOS, FLUJO NETO, and SALDO FINAL.

CAJA TITAN		CAJA ILACER	
SALDO INICIAL	100,000.00	SALDO INICIAL	0.00
INGRESOS	0.00	INGRESOS	0.00
EGRESOS	43,500.00	EGRESOS	0.00
FLUJO NETO	-43,500.00	FLUJO NETO	0.00
SALDO FINAL	56,500.00	SALDO FINAL	0.00

3.10.1.12. Formulario de Apertura.

Para iniciar los movimientos deseados en caja, debemos iniciar con apertura de caja.

Figura 3.29

Formulario de Apertura de Caja

3.10.1.13. Formulario de Ingreso en Caja.

En la interfaz podemos realizar el registro de un ingreso a caja.

Figura 3.30

Formulario de Ingresos a Caja

3.10.1.14. Formulario de Egresos en Caja.

En la interfaz se realiza los registros de egresos que tenemos en caja.

Figura 3.31

Egresos Caja

The screenshot displays the 'FORMULARIO NUEVO EGRESO EN BOLIVIANOS' interface. At the top, it shows the 'SALDO ACTUAL' as 56,500.00 Bs. The form contains three input fields: 'DOCUMENTO DE RESPALDO' (with a dropdown menu), 'Nro. DOCUMENTO' (with a text input), and 'IMPORTE BS.' (with a text input). To the right of these fields are two buttons: 'Registrar' (red) and '+ Agregar' (black). Below the form is a table with the following columns: 'ITEM', 'IMPORTE BS.', 'DOCUMENTO RESPALDO', and 'NUMERO DOCUMENTO RESPALDO'. The left sidebar contains a menu with options like 'PANEL', 'Tablero Almacenes', 'Tablero Cajas', 'Movimientos', 'Administrar Almacenes', 'Administrar Cajas', 'Administrar Usuarios', 'MANTENIMIENTO DE DATOS', 'Datos', 'Reporte Caja', and 'Reporte Almacen'. The top right corner shows the user role 'ADMINISTRADOR'.

3.10.1.15. Formulario de Cierre de Caja.

Una vez finalizando las transacciones, iniciamos con el cierre de caja para lo cual debemos llenar el siguiente formulario.

Figura 3.32

Formulario de Cierre de Caja

The screenshot displays the 'CIERRE DE CAJA EN BOLIVIANOS' form. It includes a summary table and a form with the following fields:

CIERRE DE CAJA EN BOLIVIANOS	
APERTURA	0
INGRESOS	43500
EGRESOS	-43500
FLUJO NETO	0
SALDO FINAL	56500

Form fields:

- ENTREGADO A: SELECCIONAR
- USUARIO: ADMINISTRADOR
- FECHA CIERRE: dd/mm/aaaa
- SALDO EN BS.: 56500
- FALTANTE EN BS.: ingrese faltante
- SOBRANTE EN BS.: ingrese sobrante
- GLOSA: CIERRE DE CAJA TITAN

A 'GUARDAR' button is located at the bottom left of the form area.

3.10.1.16. Reporte de Caja.

En la interfaz se realiza el filtrado para observar los movimientos en caja.

Figura 3.33

Reporte Caja

The screenshot displays the 'REPORTES CAJAS' form. It includes a search and filter section with the following fields:

REPORTES CAJAS

Movimientos Cierre

DESDE + dd/mm/aaaa

HASTA + dd/mm/aaaa

TIPO MOVIMIENTO

CAJA SELECCIONE

FILTRAR

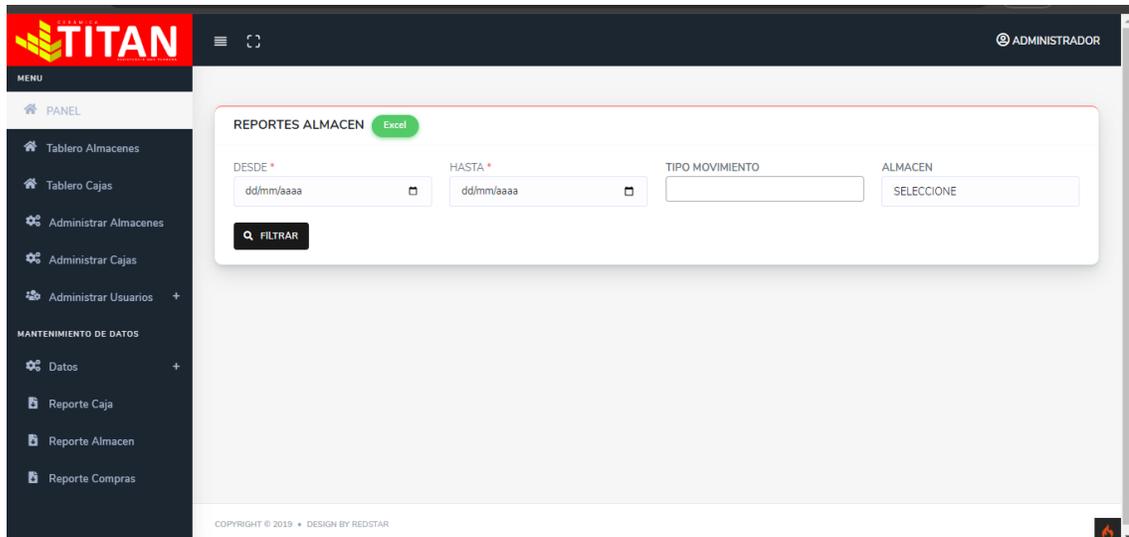
COPYRIGHT © 2019 • DESIGN BY REDSTAR

3.10.1.17. Reporte de Almacén.

En la interfaz se realiza el filtrado para observar los movimientos de entradas y salidas en almacén.

Figura 3.34

Reporte de Almacén

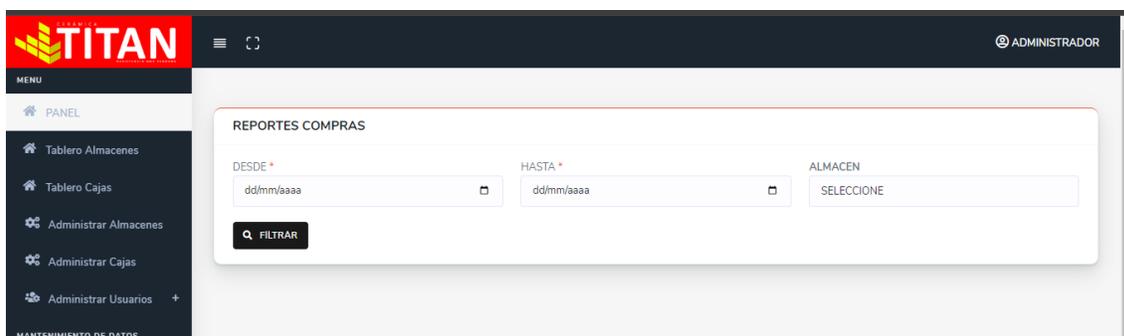


3.10.1.18. Reporte de Compras.

En la interfaz se realiza el filtrado para observar las solicitudes pendientes y aprobadas de las compras en almacén

Figura 3.35

Reporte de Compras



CAPÍTULO IV

CALIDAD, COSTOS Y SEGURIDAD DEL SOFTWARE

4. CAPÍTULO IV- MÉTRICAS DE CALIDAD, ESTIMACIÓN DE COSTOS Y SEGURIDAD

4.4. MÉTRICAS DE CALIDAD

Las métricas de calidad se refieren a un conjunto de indicadores que se utilizan para medir y evaluar la calidad de un producto, servicio o proceso. Estándar de calidad es el que reúne los requisitos mínimos en busca de la excelencia dentro de una organización industrial.

4.4.1. Estándar ISO 9126

Los factores ISO 9126 no necesariamente son utilizados para medidas directas, en cualquier caso, facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad de un sistema.

4.4.1.1. Funcionalidad.

funcionalidad es la capacidad del software para cumplir y proporcionar las funciones que satisfacen las necesidades explícitas e implícitas cuando se utiliza en condiciones específicas.

Tabla 4.1

Parámetros de Medición

PARÁMETROS DE MEDICIÓN	CUENTAS
Número de entrada de usuarios	21
Número de salidas de usuario	25
Número de peticiones de usuario	3
Número de archivos	26
Número de interfaces externas	1

Para calcular el punto de función se tiene que realizar el cálculo de la cuenta total con los factores de ponderación especificados en la siguiente tabla

Tabla 4.2

Parámetros de Medición con Factor Medio

PARÁMETRO DE MEDICIÓN	CUENTA	FACTOR	TOTAL
Número de entrada de usuarios	21	4	84
Número de salidas de usuario	25	5	125
Número de peticiones de usuario	3	4	12
Número de archivos	26	10	260
Número de interfaces externas	1	7	7
CONTEO TOTAL			488

Tabla 4.3

Valores de Ajustes de Complejidad

IMPORTANCIA	0%	20%	40%	60%	80%	100%	FI
Escala	No influencia	Incidencia	Moderado	Medio	Significativo	Esencial	
Factor	0	1	2	3	4	5	
1. ¿Requiere el sistema copia de seguridad y fiable?						X	5
2. ¿Se requiere comunicación de datos?						X	5
3. ¿Existen funciones de procesamiento distribuido?					X		3
4. ¿Es crítico el rendimiento?			X				2
5. ¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?						X	4

6. ¿Requiere el sistema entrada de datos interactiva?	X	4
7. ¿Requiere la entrada de datos interactivas que las transacciones de entrada se llevan a cabo sobre múltiples pantallas o variadas opciones?	X	3
8. ¿Se actualizan los archivos maestros de forma interactiva?	X	3
9. ¿Son complejos las entradas, las salidas, los archivos o peticiones?	X	3
10. ¿Es complejo el procesamiento interno?	X	3
11. ¿Se ha utilizado el código para ser reutilizable?	X	4
12. ¿Están incluidas en el diseño la conversión y la instalación?	X	3
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	X	4
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	X	5
TOTAL		51

Calculando el punto de fusión reemplazando en la ecuación (1):

$$PF = Cuenta\ total [X + Y * \sum FI]$$

$$PF = 488 * [0.65 + 0.01 * 51]$$

$$PF = 566.08$$

Por otro la considerando el máximo ajuste de complejidad que se puede alcanzar se obtiene:

$$PF = 488 * [0.65 + 0.01 * 70]$$

$$PF = 658,8$$

Por lo tanto, se tiene que la funcionalidad real reemplazando en la ecuación (2) es:

$$Funcionalidad = \frac{PF}{PF_{max}}$$

$$Funcionalidad = \frac{566,08}{658.8}$$

$$Funcionalidad = 0.8592 * 100$$

$$Funcionalidad = 85.92 \%$$

Con el análisis realizado, se observa que la funcionalidad del software tiene un 85.92%, sin riesgo de fallar.

4.4.1.2. Confiabilidad.

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos de un programa o computadora.

Se estima que un fallo puede ocurrir cada 15 días hábiles y su reparación promedio puede tomar 1 hora, entonces:

Reemplazando en ecuación (3):

$$TMEF = TMDF + TMDR$$

$$TMEF = (dias * hora) + 1$$

$$TMEF = (15 * 8) + 1$$

$$TMEF = 121 horas$$

$$TMDF = (dias * hora)$$

$$TMDF = (15 * 8)$$

$$TMDF = 120$$

Remplazando en la ecuación (4):

$$Disponibilidad = \frac{TMDF}{TMEF} * 100$$

$$Disponibilidad = \frac{120}{121} * 100$$

$$Disponibilidad = 99.17 \%$$

Por lo tanto, la confiabilidad del sistema es de 99,17 % en un periodo de 15 días como tiempo de prueba.

4.4.1.3. Usabilidad.

Es la facilidad de uso, un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios

Tabla 4.4

Escala de Valores

ESCALA	VALOR
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Tabla 4.5*Preguntas para determinar la usabilidad*

Nro.	PREGUNTAS	SI	NO	EVALUACIÓN
1	¿Puede utilizar con fiabilidad el sistema?	5	0	1
2	¿Puede controlar operaciones que el sistema solicite?	4	1	0.8
3	¿El sistema permitió la retroalimentación de información?	4	0	0.8
4	¿El sistema cuenta con interfaz gráfica agradable a la vista?	5	0	1
5	¿La respuesta del sistema es satisfactoria?	4	1	0.8
6	¿Le parecen accesibles las funciones del sistema?	5	0	1
7	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	1	1
TOTAL				6.4

$$FU = [\sum (X_{in}) * 100]$$

$$FU = \left[\frac{6.4}{7} * 100 \right]$$

$$FU = 91.43\%$$

Por lo tanto, existe un 91,43 % de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

4.4.1.4. Mantenibilidad.

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la imprenta desee implementar para su uso posterior.

Ecuación de la mantenibilidad

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Tabla 4.6

Valores para Determinar la Mantenibilidad

DESCRIPCIÓN	VALOR
Mt = Numero de módulos de la versión actual	6
Fc = Numero de módulos en la versión actual que se han modificado	0
Fa = Numero de módulos en la versión actual que se han añadido	1
Fd = Numero de módulos de la anterior versión que se han borrado en la versión actual	0

$$IMS = \frac{[6 - (0 + 1 + 0)]}{6}$$

$$IMS = 0.833 * 100 = 83.3 \%$$

$$Mantenibilidad = 83.3\%$$

Por lo tanto, se puede decir que el sistema tiene un índice de mantenibilidad de 83,3 % que es la facilidad de mantenimiento, el 16,7% restante es el margen de error corresponde a los cambios y modificaciones que se realizan al sistema

4.4.1.5. Portabilidad.

Un sistema es considerado portable, si el costo de transportar y adaptar a un ambiente es menor al costo de rediseñar el sistema para el mismo ambiente.

Para el cálculo de la portabilidad se tomó en cuenta la siguiente tabla que contiene las características para la evaluación de la portabilidad.

Tabla 4.7

Factor de Portabilidad

FACTOR DE PORTABILIDAD	VALOR%
Puede ser transferido en un entorno a otro	100%
Se puede adaptar a otros ambientes con facilidad	100%
Es fácil de instalar	100%
Es capaz de reemplazar a una aplicación similar	80%
TOTAL	95%

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente.

Tabla 4.8*Resultados de Calidad*

CARACTERÍSTICAS	RESULTADOS
Funcionalidad	85.92%
Confiabilidad	99.17%
Usabilidad	91.43%
Mantenibilidad	83.30%
Portabilidad	95%
EVALUACIÓN FINAL	91%

4.4.2. Metodología de Estimación de Costo de Software COCOMO II

En este capítulo se desarrolla el método análisis de costo y empleando las fórmulas para el análisis de costos del sistema descritos en el capítulo II, El presente trabajo se considerará como un proyecto de software orgánico por contener unos pocos miles de líneas de código.

Ecuación para Determinar el Número de Líneas de Códigos en Miles

$$KLDC = \frac{\text{lineas de codigo}}{1000} \quad (10)$$

$$KLDC = \frac{12,924}{1000}$$

$$KLDC = 12.92$$

Los valores de coeficientes (a, e) y exponentes (b, d) por el tipo de proyecto son los siguientes:

Tabla 4.9*Constantes para el Cálculo de Cocomo*

MODO	a	B	C	d
Orgánico	2.4	1.05	2.5	0.38
Semi – Orgánico	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.33

Tabla 4.10*Cálculo del Atributo ME*

ATRIBUTOS	VALORE					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos del personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos de proyecto						

Técnicas actualizadas de programación	1,24	1,1	1,00	0,91	0,82
Utilización de herramientas de software	1,24	1,1	1,00	0,91	0,83
Restricción de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10

El valor de ME es el resultado de la multiplicación de los valores evaluados en la tabla anterior teniendo:

$$ME = 0,48$$

Cálculo Esfuerzo.

$$(E) = a * (KLDC)^b * ME$$

$$E = 2,4 * (12,9)^{1.05} * 0.48$$

$$E = 16.87 \text{ [persona/mes]}$$

Cálculo de tiempo.

$$(T) = c * E^d$$

$$T = 2.5 * 16.87^{0.38}$$

$$T = 7.25 \text{ (meses)}$$

Cálculo de la productividad.

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

$$P = \frac{16.87}{7.25}$$

$$P = 2.32 \text{ personas}$$

Costo total del proyecto.

Salario de un programador en Bolivia tomando en cuenta el valor de 2500 bs

$$\text{Costo mes} = P * \text{salario}$$

$$\text{Costo mes} = 2 * 2500$$

$$\text{Costo mes} = 5000 \text{ bs}$$

$$\text{Costo total} = \text{costo mes} * T$$

$$\text{Costo total} = 5000 * 7$$

$$\text{Costo total} = 35000 \text{ bs}$$

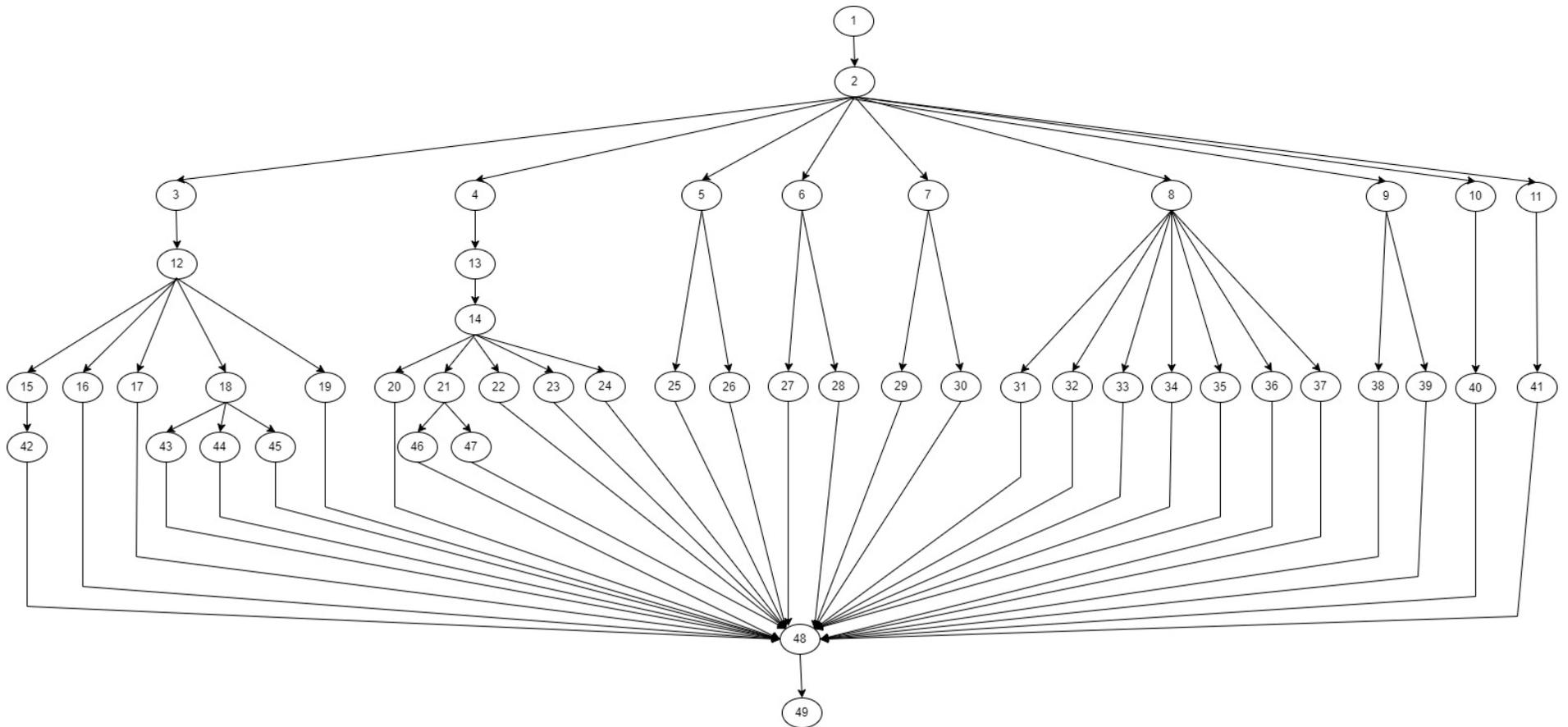
Tomando en cuenta la tasa de cambio del dólar a 6,91 el costo del sistema es aproximadamente en 5065 \$.

4.5. PRUEBAS DEL SISTEMA

4.5.1.1. Pruebas de Caja Blanca.

Esta prueba se orienta al cálculo de las regiones que deben ser consideradas como partes independientes del sistema, asegurando así que cada región se ejecuta al menos una vez. De forma general, se debe seguir.

El objetivo principal de las pruebas de caja blanca es verificar que el código funcione correctamente y cumpla con los requisitos establecidos

Figura 4.1*Grafo del Sistema*

Donde:

Inicio de sesión (1)

Menú principal (2)

Tablero de almacén (3)

Tablero de cajas (4)

Administración de almacén (5)

Administración de cajas (6)

Administración de usuarios (7)

Datos (8)

Reporte caja (9)

Reporte almacén (10)

Reporte compra (11)

Lista de almacenes (12)

Lista de cajas (13)

Movimientos almacén (14)

Solicitudes (15)

Solicitudes pendientes (16)

Solicitudes confirmadas (17)

Movimientos caja (18)

Stock (19)

Flujo de caja (20)

Egresos (21)

Ingresos (22)

Apertura (23)

Cierre (24)

Registro almacén (25)

Asignación almacén usuario (26)

Registro caja (27)

Asignación caja usuario (28)

Usuarios (29)

Roles (30)

Clientes (31)

Categoría (32)

Artículos (33)

Unidad medida (34)

Destino (35)

Documento respaldo (36)

Proveedor (37)

Filtro movimiento (38)

Filtro cierre (39)

Filtro almacén (40)

Filtro compras (41)

Registro solicitud (42)

Movimientos (43)

Entradas (44)

Salidas (45)

Registro egresos (46)

Registro egresos bajo solicitud de almacén (47)

Fin ciclo sistema (48)

Fin del sistema (49)

Posteriormente analizamos el grafo generado a partir de las características del sistema, ahora determinaremos la complejidad del grafo mediante la ecuación:

Ecuación para Determinar el Ciclo matica

$$V(G) = A - N + 2 \quad (11)$$

Remplazando:

$$V(G) = 77 - 49 + 2 = 30$$

Los caminos que deben ser probados son los siguientes:

Camino 1: 1-2-3-12-15-42-48-49

Camino 2: 1-2-3-12-16-48-49

Camino 3: 1-2-3-12-17-48-49

Camino 4: 1-2-3-12-18-43-48-49

Camino 5: 1-2-3-12-18-44-48-49

Camino 6: 1-2-3-12-18-45-48-49

Camino 7: 1-2-3-12-19-48-49

Camino 8: 1-2-4-13-14-20-48-49

Camino 9: 1-2-4-13-14-21-46-48-49

Camino 10: 1-2-4-13-14-21-47-48-49

Camino 11: 1-2-4-13-14-22-48-49

Camino 12: 1-2-4-13-14-23-48-49

Camino 13: 1-2-4-13-14-24-48-49

Camino 14: 1-2-5-25-48-49

Camino 15: 1-2-5-26-48-49

Camino 16: 1-2-6-27-48-49

Camino 17: 1-2-6-28-48-49

Camino 18: 1-2-7-29-48-49

Camino 19: 1-2-7-30-48-49

Camino 20: 1-2-8-31-48-49

Camino 21: 1-2-8-32-48-49

Camino 22: 1-2-8-33-48-49

Camino 23: 1-2-8-34-48-49

Camino 24: 1-2-8-35-48-49

Camino 25: 1-2-8-36-48-49

Camino 26: 1-2-8-37-48-49

Camino 27: 1-2-9-38-48-49

Camino 28: 1-2-9-39-48-49

Camino 29: 1-2-10-40-48-49

Camino 30: 1-2-11-41-48-49

Camino 1: El tablero de almacén se muestra los almacenes registrados dentro de la empresa en el cual se realiza el registro de solicitudes de compras.

Camino 2: Se selecciona el almacén en el tablero y se observa las solicitudes pendientes de compras donde podemos realizar la autorización de compra.

Camino 3: Se selecciona el almacén elegido en el cual se visualizan las solicitudes confirmadas de compras.

Camino 4: Entrando al almacén elegido se observa los movimientos de entradas y salidas que se realizó en el almacén.

Camino 5: Entrando al almacén elegido podemos realizar las entradas de artículos en almacén.

Camino 6: Entrando al almacén elegido podemos realizar las salidas de artículos en almacén.

Camino 7: Entrando al almacén elegido se visualiza el stock de artículos dentro de almacén.

Camino 8: Se ingresa al listado de cajas donde se selecciona la caja donde se visualiza el flujo de efectivo existente de las cajas donde existan aperturas.

Camino 9: Se ingresa al listado de cajas donde se selecciona la caja donde se registran los egresos de caja.

Camino 10: Se ingresa al listado de cajas donde se selecciona la caja donde se registran los egresos bajo solicitudes de compras mandadas por almacén.

Camino 11: Se ingresa al listado de cajas donde se selecciona la caja donde se registran los ingresos de efectivo en caja.

Camino 12: Se ingresa al listado de cajas donde se selecciona la caja donde se registran la apertura de caja para iniciar con las operaciones.

Camino 13: Se ingresa al listado de cajas donde se selecciona la caja donde se registra el cierre de caja para finalizar con los movimientos dentro de la caja.

Camino 14: al ingresar al sistema se realiza el registro de almacenes.

Camino 15: al ingresar al sistema se realiza el registro de responsables de almacén en el sistema.

Camino 16: Al ingresar al sistema se realiza el registro de cajas de la empresa.

Camino 17: Al ingresar al sistema en administración de cajas se realiza la asignación de cajeros a cada caja.

Camino 18: El módulo de administración de usuarios se realiza el registro de nuevos usuarios que interactuaran con el sistema.

Camino 19: El módulo de administración de usuarios se gestiona la asignación de permiso a los roles o creación de roles.

Camino 20: El módulo de datos se realiza el registro y modificaciones de los clientes que interactúan dentro del sistema.

Camino 21: El módulo de datos se realiza el registro y modificaciones de las categorías de los artículos que interactúan dentro del sistema.

Camino 22: El módulo de datos se realiza el registro y modificaciones de los artículos que interactúan dentro del almacén.

Camino 23: El módulo de datos se realiza el registro y modificaciones de unidades de medida de los artículos que interactúan dentro del almacén.

Camino 24: El módulo de datos se realiza el registro y modificaciones de los destinos que se les asignara a los artículos en las salidas o solicitudes.

Camino 25: El módulo de datos se realiza el registro y modificaciones de los documentos de respaldo utilizados en caja para realizar los movimientos.

Camino 26: El módulo de datos se realiza el registro y modificaciones de los proveedores de donde se adquieren los artículos de almacén.

Camino 27: Ingresando al sistema se realiza la consulta de los movimientos realizados en caja filtrando las fechas y la caja que se desea a ver los movimientos.

Camino 28: Ingresando al sistema se realiza la consulta de los cierres realizados en caja filtrando las fechas y la caja.

Camino 29: Ingresando al sistema se realiza la consulta de los movimientos realizados en almacén filtrando las fechas deseadas a ver los movimientos de entradas y salidas de artículos.

Camino 30: Al ingresar al sistema vemos los reportes de compras realizando un filtrado por fechas y almacén donde se desea realizar la consulta.

4.5.1.2. Pruebas de Caja Negra.

4.5.1.2.1. Prueba de Caja Negra – Inicio de Sesión.

Figura 4.2

Prueba de Caja Negra Inicio de Sesión

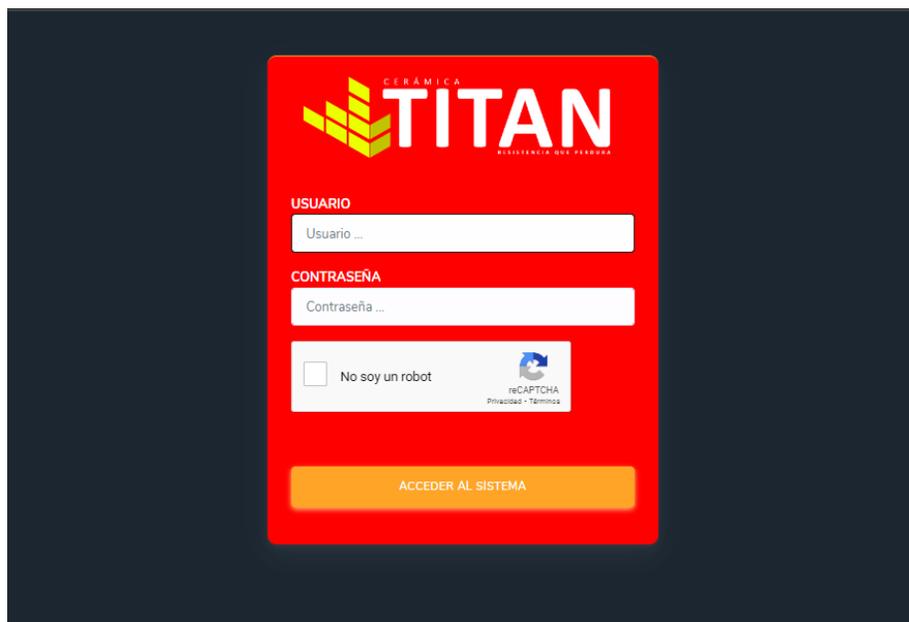


Tabla 4.11

Prueba de Caja Negra Inicio de Sesión

ENTRADAS		SALIDAS	RESULTADOS
Usuario	Contraseña	"Ingrese usuario y contraseña"	El sistema valida que no ingresen datos en blanco
Administrador	Administrador2024.	"Bienvenido al sistema"	El sistema concede acceso al ingreso de datos correcto

4.5.1.2.2. Prueba de Caja Negra – Crear Solicitud.

Figura 4.3

Prueba de Caja Negra crear Solicitud

The screenshot shows the TITAN system interface for creating a request. The form includes the following fields and controls:

- NOMBRE ARTICULO *:** A dropdown menu with the text "Seleccione".
- UNIDAD MEDIDA *:** A text input field.
- CANTIDAD SOLICITADA *:** A text input field.
- PROVEEDOR *:** A dropdown menu with the text "SELECCIONAR PROVEEDOR".
- PRECIO UNIDAD *:** A text input field.
- DESTINO *:** A dropdown menu with the text "SELECCIONAR DESTINO".
- TOTAL *:** A text input field.

Below the form, there is a table with the following columns: ITEM, NOMBRE ARTICULO, U. MEDIDA, PROVEEDOR, CANTIDAD, P/U, TOTAL, OPERACIONES. The table is currently empty.

At the bottom left of the form, there is a button labeled "TERMINAR PROCESO".

The top right of the interface shows the user role "ADMINISTRADOR" and the date "miércoles, 5 de junio de 202".

Tabla 4.12

Valores del Registro Solicitud

OPERACIÓN	CAMPO	DATOS
Entrada	Nombre articulo	TUERCA DE 5/8 316-128
	Unidad Medida	KILOGRAMOS
	Cantidad solicitada	20
	Proveedor	PERNOVOL
	Precio unidad	5
	Destino	SALA DE MAQUINA, DESTINO ESPECIFICO: DOSADORA
	Total	100

Salida	"Ingreso los campos requeridos"	El registro se muestra en el listado de solicitudes
Resultados	El sistema valida que no se registre datos en blanco	Cuando se llenan los datos válidos, el sistema registra la información en la base de datos

4.5.1.2.3. Prueba de Caja Negra – Ingreso en Caja.

Figura 4.4

Prueba de Caja Negra Ingreso en Caja

Tabla 4.13

Valores de Registro de Ingreso en Caja

OPERACIÓN	CAMPO	DATOS
	Documento respaldo	RECIBOS
	Nro. Documento	67
	Importe bs.	500

Entrada	Recibido de	ENRIQUE PEREZ ULO
	Fecha	5/06/2024
	Glosa	INYECCIÓN DE RECURSOS A CAJA TITÁN
Salida	"Ingreso los campos requeridos"	El registro se muestra en el listado de solicitudes
	El sistema valida que no se registre datos en blanco	Cuando se llenan los datos válidos, el sistema registra la información en la base de datos
Resultados		

4.6. Sistema de Gestión de Seguridad de la Información ISO-27002

En la ISO – 27002 evalúa y rectifica la implementación mediante el cumplimiento de normas, así como la mejora continua de un conjunto de controles que permiten reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento de la imprenta en cuanto a la seguridad de la información. La seguridad de la información se define en el estándar como “la preservación de la confidencialidad, integridad y disponibilidad., para lo cual se tomó los siguientes tipos de seguridad:

4.6.1. Seguridad Lógica

4.6.1.1. Copias de Seguridad.

Back-up de la base de datos del sistema se debe realizar de acuerdo a la siguiente tabla:

Tabla 4.14*Seguridad Lógica*

Descripción	Duración	Observaciones
Registro de solicitudes	1 vez por semana	No es de mucha importancia
Movimientos en almacenes	1 vez por semana	De mucha importancia
Movimientos en cajas	1 vez por semana	De mucha importancia
Administración de almacenes	1 vez por semana	No de mucha importancia
Administración de cajas	1 vez por semana	No de mucha importancia

4.6.1.2. Identificación y autenticación.

Los usuarios cuentan con usuario y contraseña estrictamente controlada para acceder al sistema, después del proceso de llenar se debe pasar el capchap, de esta manera se previene el ingreso de personas que no son usuarios.

4.6.1.3. Encriptación.

Para la seguridad del password estará encriptado con el algoritmo HASH que es una encriptación de alta seguridad para el ingreso al sistema.

4.6.2. Seguridad Física

Los back-up de la base de datos deberán ser protegidas en áreas seguras, que solo permita el acceso a personal autorizado, deben ser almacenados por fechas.

4.6.3. Seguridad Organizativa

Manejar el back-up de acuerdo a la fecha que se realizó para mayor organización.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5. CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.4. CONCLUSIONES

Se logro desarrollar un SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y ADMINISTRACIÓN DE ALMACENES Y CAJAS, logrando que sea de gran ayuda en los procesos administrativos dentro de la empresa.

La implementación del sistema alcanzo los siguientes objetivos:

- ✓ Se diseñó la base de datos escalable para almacenar la información de manera eficiente de los registros.
- ✓ Se desarrollo módulos para generar solicitudes de compras y la verificación, aprobación en función al almacén correspondiente de manera satisfactoria.
- ✓ Se implementó un módulo para el registro de entradas y salidas de artículos, obteniendo un control de existencias por almacén de los artículos existentes dentro de los mismos.
- ✓ Se diseñó el módulo para el control de flujo de caja, teniendo un manejo y control eficiente del efectivo en caja, a su vez se realizó los comprobantes por cada movimiento desempleado en caja.
- ✓ Se diseñó módulos para generar reportes de solicitudes de compra, movimientos en caja, del flujo diario y movimientos el almacén en saldos físicos del inventario.

En conclusión, se llegó a solucionar los problemas específicos y consecuentemente la problemática central y los objetivos planteados fueron cumplidos de manera satisfactoria.

5.5. RECOMENDACIONES

En base a la observación, se elaboran las siguientes recomendaciones:

- ✓ Cambiar las contraseñas de acceso al sistema periódicamente, para mejorar la seguridad y evitar accesos no autorizados.
- ✓ Realizar copias de seguridad de la base de datos, con el fin de evitar pérdida de información.
- ✓ Capacitar al nuevo personal de forma correcta, para que haga el uso correcto del sistema.
- ✓ Revisar el manual de usuario si no se tiene claro cómo realizar una operación específica.

Bibliografía

PHP. (2024). Obtenido de <https://www.php.net/manual/es/intro-what-is.php>

Aduviri, P. P. (2016). *SISTEMA WEB DE CONTROL DE VENTAS E INVENTARIOS*[Proyecto de Grado, Universidad Mayor de San Andres]. Repositorio institucional, La Paz.

Obtenido de

<https://repositorio.umsa.bo/bitstream/handle/123456789/9987/T.3231.pdf?sequence=1&isAllowed=y>

Aquino, A. H. (2018). Sistema Web para el control y administracion de recusus humanos [proyecto de grado,Universidad Mayor de San Andres]. *Repositorio Institucional*.

Universidad Mayor de San Andres. Obtenido de

<https://repositorio.umsa.bo/bitstream/handle/123456789/17599/T-3442.pdf?sequence=1&isAllow>

Ballesteros. (2003). *ISO/IEC 9126*.

Ballesteros, M. M. (2003). *ISO/IEC 9126*.

Boehm, B. (1981). *Software Engineering Economics*.

Busquelle. (2010). "Análisis de Puntos de Función".

Carrillo, C. E. (2017). *SISTEMA WEB DE CONTROL COMPRAS, VENTAS E*

INVENTARIO[Proyecto de Grado, Universidad Mayor de San Andres]. repositorio

institucional, La Paz. Obtenido de

<https://repositorio.umsa.bo/bitstream/handle/123456789/12558/T.3290.pdf?sequence=1&isAllow>

Casa, T. V. (2013). *Sistema informático para la de gestión de información sobre el personal*

técnico de la Universidad de Granma[Tesis,Universidad de Granma]. repositorio

institucional. Obtenido de

https://cc.bingj.com/cache.aspx?q=Sistema+inform%3%a1tico+para+la+de+gesti%3%b3n+de+informaci%3%b3n+sobre+el+personal+t%3%a9cnico+de+la+Universid+ad+de+Granma&d=4802718597579498&mkt=es-XL&setlang=es-ES&w=ByPlerOBa5-IBgvCy79o03tBqUmRIG_u

chartjs.org. (17 de Mayo de 2024). *chartjs.org*. Obtenido de chartjs.org:

<https://www.chartjs.org/docs/latest/>

Codeigniter. (abril de 2024). *Manual de usuario*. Obtenido de codeigniter 4:

https://codeigniter.com/user_guide/intro/index.html

DATADEC. (8 de Enero de 2024). *DATADEC*. Obtenido de DATADEC:

<https://www.datadec.es/blog/que-es-un-almacen-y-por-que-es-tan-importante>

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

desarrollo web.com. (Noviembre de 2009). *desarrollo web.com*. Obtenido de desarrollo web.com: <https://desarrolloweb.com/articulos/codeigniter.html>

Echeverria, F. J. (2017). *SISTEMA INFORMÁTICO PARA LA GESTIÓN DE VENTAS E INVENTARIO*[Trabajo de grado, Universidad TÉCNICA FEDERICO SANTA MARÍA]. repositorio institucional. Obtenido de <https://repositorio.usm.cl/handle/11673/46548>

EduTEKA. (2018). *que es sistemas*.

Figuerola, M. A. (2012). *Nacion multicultural*. Obtenido de Nacion multicultural:

<https://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>

FRANK, E. (1970). *MODELO RELACIONAL*. Obtenido de

<https://bookdown.org/paranedagarcia/database/el-modelo-relacional.html>

Galiano. (2012). *metodologia UWE*.

Galiano, L. (octubre de 2012). *Planificacion de mi proyecto*. Obtenido de Planificacion de mi proyecto: <https://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>

Gardey., J. P. (2017). Promedio ponderado. *Definbiciones*.

González, G. G. (2014). *Métodos analíticos y métricas de calidad del software*. San José. Costa Rica: Editorial EUNED.

Jimenez, O. R. (2020). *DESARROLLO DE UN SISTEMA DE CONTROL DE INVENTARIO LA GESTIÓN DE INSUMOS EN LA EMPRESA "MINIMARKET DON LUCHO"* [Trabajo de grado, Universidad Estatal de Milagro]. repositorio institucional. Obtenido de <http://repositorio.unemi.edu.ec/handle/123456789/5132>

Olsina. (2008). *METODOLOGIA UWE*.

Oracle. (2024). Obtenido de Oracle: <https://www.oracle.com/co/mysql/what-is-mysql/#:~:text=MySQL%20es%20la%20base%20de,%2C%20Shopify%20y%20Booking.com>.

Oracle. (2024). *oracle*. Obtenido de <https://www.oracle.com/co/mysql/what-is-mysql/#:~:text=MySQL%20es%20la%20base%20de,%2C%20Shopify%20y%20Booking.com>.

Pérez, O. (2014). *Mysql*.

PHP. (ABRIL de 2024). *PAGINA OFICIAL PHP*. Obtenido de PHP: <https://www.php.net/manual/es/intro-what-is.php>

PhpSpreadsheet. (3 de Diciembre de 2016). *PhpSpreadsheet*. Obtenido de PhpSpreadsheet: <https://phpspreadsheet.readthedocs.io/en/latest/#>

Pressman, R. (2013). *ISO/IEC 2126*.

Remon, M. A. (2012). *DESARROLLO DE APLICACIONES WEB CON PHP*. Lima: Macro.

Rossi. (2018). *Web Engineering - Modelling and Implementing Web Applications*.

Sage Software. (6 de Mayo de 2021). *Sage Software*. Obtenido de Sage Software:

<https://www.sage.com/es-es/blog/diccionario-empresarial/caja/>

UWE. (25 de Mayo de 2016). *UWE – Ingeniería web basada en UML*. Obtenido de UWE –

Ingeniería web basada en UML:

<https://uwe.pst.ifi.lmu.de/teachingTutorialRequirements.html>

ANEXOS Y MANUAL DE USUARIO

ANEXO 1 REGISTRO DE ENTRADAS Y SALIDAS EN ALMACÉN

41594 584

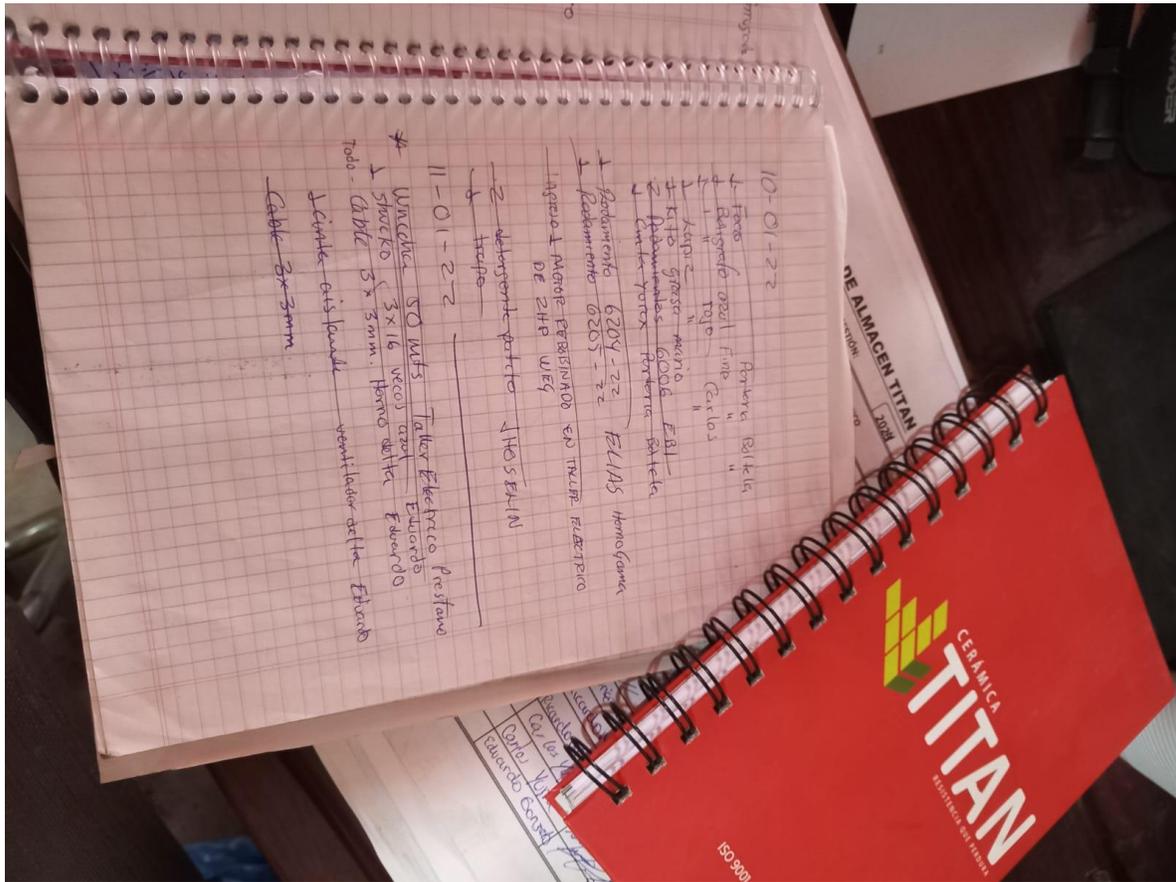


REGISTRO DE ENTRADAS Y SALIDAS DE ALMACEN TITAN

RESPONSABLE: N°	FECHA	CODIGO	DATOS DEL MATERIAL			MIES:		GESTIÓN:		SCUPTANTE NOMBRE	FIRMA
			NOMBRE DEL ARTICULO	U/M	P/U	INGRESOS CANTIDAD	SALIDA CANTIDAD	DESTINO GENERAL	DESTINO ESPECIFICO		
	25-10		fosible laminadora	U						Ignacio Sombra	
	25-10		B-53	U						Ignacio Sombra	
	25-10		B-68	U						Ignacio Sombra	
	25-10		Alambre corte	U							
	25-10		Disco de desbaste	U							
	25-10		tambores	Par			1			Mario Calizaya	
	25-10		Foco 100 w	U						Alex Colliage	
	26-10		Combustible	Lts						Jose Rodrigo	
	26-10		"	Lts						"	
	26-10		"	U						Eduardo	
	26-10		Perno 5/16 x 1/2	U						Raymar B	
	26-10		Foco 100	U						Elias Alvin	
			Teflon	U						Gerardo	
			Cinta aislante	U						José	
28/10			Traverse	U						Francisco	
			Cuente	Par						"	
			Foco	U						Javier Vela	
			Ropa de trabajo	U							
			Guantes de wero	Par							
			Folier Amarillo	U						Aurelia Cruz	

1180

ANEXO 2 REGISTRO MANUAL DE SOLICITUDES DE COMPRA



ANEXO 3 ALMACÉN DE LA EMPRESA



ANEXO 4 COMPROBANTES DE CAJA

