

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



## PROYECTO DE GRADO

### SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS CASO: TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON

Para optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Eulogia Mamani Chuca

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Especialista: Ing. Santos Aurelio Limachi Huanca

Tutor Revisor: Ing. Sergio Ramiro Rojas Saire

EL ALTO - BOLIVIA

2020

## **DEDICATORIA**

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto. En primer lugar, doy infinitamente gracias a Dios, por haberme permitido llegar hasta este punto y darme la oportunidad de cursar esta carrera y terminar este proyecto de grado.

A mis padres, Anastasia la persona que más me inspiro desde pequeña a superar cada obstáculo de la vida a mi papá Angelino por sus enseñanzas y palabras sabias en cada momento, A mi esposo Eddy Heberon por el amor brindado cada día por la paciencia y comprensión, a mis hermosos hijos Kheddy Russell y Sharon Alexia por ser mi fuente de inspiración para seguir adelante, a toda mi familia mis hermanas con mucho agradecimiento María, Viviana, Edgar y Cristóbal por sus palabras de aliento que hicieron posible seguir con fuerza y valor para culminar esta etapa de mi vida, a todos ellos siempre los llevo en mi corazón.

## **AGRADECIMIENTO**

A Dios, por darme la vida y el conocimiento, la voluntad y la perseverancia para poder alcanzar la culminación de este proyecto de grado.

A mi Tutor Metodológico, Ing. Marisol Arguedas Balladares por su enseñanza, confianza, paciencia, tiempo, apoyo y motivación que me brindo desde el momento que empecé con este proyecto.

A mi Tutor Especialista, Ing. Santos Aurelio Limachi Huanca por compartir sus conocimientos, brindándome recomendaciones y motivación durante el desarrollo de este proyecto.

A mi Tutor Revisor, Ing. Sergio Ramiro Rojas Saire por su continuo seguimiento y la disponibilidad de tiempo que le dedico a cada sus observaciones y recomendaciones brindadas en la realización del proyecto, por sus sugerencias y observaciones acertadas en la realización del presente proyecto.

A la Universidad Pública de El Alto, por acogerme en sus aulas por los años de estudio, de igual forma a la carrera de Ingeniería de Sistemas y a mis compañeros(as) de carrera por su amistad y apoyo incondicional.

A la empresa TESICON., por brindarme la oportunidad y la confianza para el desarrollo de mi proyecto de grado.

Muchas gracias a todos.

## **RESUMEN**

En el presente documento tiene la siguiente estructura:

### **CAPITULO I: MARCO PRELIMINAR**

Este primer capítulo describe el aspecto general del proyecto de grado como: la problemática, justificación, objetivos y alcances de esta forma dejar en claro los resultados las cuales se desea al concluir el proyecto de grado

### **CAPITULO II: MARCO TEÓRICO**

El segundo capítulo proporcionara los fundamentos teóricos acerca de las herramientas, teoría, métodos y metodologías Web empleados para el desarrollo del presente proyecto de grado, recopilando la información para el sustento de estudio que se realiza, estos elementos extraídos consisten en la base para la descripción y explicación del problema a solucionar.

### **CAPITULO III: MARCO APLICATIVO**

En el marco aplicativo se pone en marcha la ingeniería de requerimientos trabajando también con la metodología XP con esto se vio conveniente utilizar la metodología UWE para la fase de diseño.

### **CAPITULO IV: CALIDAD Y SEGURIDAD**

En este capítulo se realizó las pruebas de calidad de software para comprobar que el mismo es aceptable y cumple con los requisitos de la ISO 9126, en cuanto a seguridad del sistema se realizó la autenticación de usuarios y asignación de roles para cada usuario.

### **CAPITULO V: COSTOS DEL SOFTWARE**

La estimación de costos del software se lo hizo con COCOMO II basándose en los puntos de función.

### **CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES**

Finalmente, este capítulo describe las conclusiones y recomendaciones que se formularon al concluir el proyecto de grado, satisfaciendo las necesidades de la institución, logrando cumplir con los objetivos planteados desarrollando un sistema eficiente, confiable y de fácil interacción con el usuario. Haciendo un cliente satisfecho será igual a proyecto funcional más el tiempo de entrega establecido.

## ÍNDICE DE CONTENIDO

1	MARCO PRELIMINAR .....	1
1.1	INTRODUCCIÓN .....	1
1.2	ANTECEDENTES.....	2
1.2.1	Antecedentes Institucionales .....	2
1.2.2	Antecedentes Académicos .....	3
1.3	PLANTEAMIENTO DEL PROBLEMA .....	4
1.3.1	Problema Principal .....	5
1.3.2	Problemas Específicos .....	5
1.4	OBJETIVOS.....	5
1.4.1	Objetivo General .....	5
1.4.2	Objetivos Específicos .....	6
1.5	JUSTIFICACIÓN.....	6
1.5.1	Justificación Técnica .....	6
1.5.2	Justificación Económica .....	6
1.5.3	Justificación Social .....	7
1.6	METODOLOGÍA.....	7
1.6.1	Método de Ingeniería.....	7
1.6.2	Metodología de Desarrollo.....	8
1.7	HERRAMIENTAS .....	8
1.8	LÍMITES Y ALCANCES .....	10
1.8.1	Limites.....	10
1.9	APORTES.....	11
2	MARCO TEÓRICO.....	12
2.1	DESCRIPCIÓN DE LA EMPRESA .....	12

2.1.1	Misión y Visión .....	12
2.2	SISTEMA.....	13
2.3	INFORMACIÓN .....	13
2.4	SISTEMA DE INFORMACIÓN.....	14
2.5	REGISTRO.....	15
2.6	COMPRAS.....	15
2.7	VENTAS .....	15
2.8	INVENTARIOS .....	15
2.8.1	Métodos de Control de Inventario.....	17
2.8.2	Sistema de Inventarios .....	18
2.9	INGENIERÍA DE SOFTWARE .....	19
2.10	METODOLOGÍA ÁGIL.....	21
2.10.1	Diferencias Entre Metodologías Tradicionales y Agiles .....	22
2.10.2	Programación Extrema XP .....	23
2.10.2.1	Características de la metodología.....	23
2.10.2.2	Valores XP .....	24
2.10.2.3	Fases de la Metodología XP .....	25
2.11	INGENIERÍA DE REQUERIMIENTOS.....	31
2.11.1	Obtención de Requerimientos .....	32
2.11.2	Tipos de requerimientos.....	33
2.11.2.1	Requerimientos funcionales.....	33
2.11.2.2	Requerimientos no funcionales .....	34
2.11.2.3	Requerimientos del usuario.....	35
2.11.2.4	Requerimientos del sistema.....	35
2.12	METODOLOGÍA UWE.....	35

2.12.1	Lenguaje Unificado de Modelado (UML) .....	35
2.12.2	UWE – Based Web Engineering (UML).....	36
2.12.2.1	Modelo de Requisitos.....	39
2.12.2.2	Modelo de Contenido .....	40
2.12.2.3	Modelo de Navegación .....	41
2.12.2.4	Modelo de Presentación .....	43
2.12.2.5	Modelo de Proceso .....	45
2.13	ARQUITECTURA CLIENTE/SERVIDOR .....	48
2.13.1	Tipos de Arquitectura Cliente/Servidor .....	49
2.13.1.1	Arquitectura de dos Capas.....	49
2.13.1.2	Arquitectura de tres Capas .....	49
2.13.1.3	Arquitectura de N Capas.....	50
2.13.2	Seguridad del Modelo Cliente Servidor .....	50
2.14	CICLO DE VIDA DEL SOFTWARE .....	51
2.15	HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE .....	52
2.15.1	PHP .....	52
2.15.2	PostgreSQL .....	52
2.15.3	JavaScript .....	53
2.15.4	Framework Codeigniter .....	53
2.15.4.1	Boostrap .....	53
2.15.5	XAMPP .....	54
2.15.6	Arquitectura de Software MVC .....	54
2.16	PRUEBAS DE SOFTWARE .....	56
2.16.1	Prueba de Caja Blanca .....	56
2.16.2	Prueba de Caja Negra.....	57

2.17	MÉTRICAS DE CALIDAD .....	57
2.17.1	ISO/IEC 9126 .....	57
2.18	MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE .....	61
2.19	MODELO COCOMO II .....	63
2.20	SEGURIDAD DEL SISTEMA .....	68
3	MARCO APLICATIVO .....	71
3.1	INTRODUCCIÓN .....	71
3.2	ANÁLISIS DE REQUERIMIENTOS .....	72
3.2.1	Análisis de Situación Actual.....	72
3.2.2	Ingeniería de Requerimientos.....	72
3.2.3	Obtención de Requerimientos .....	73
3.3	FASE DE PLANIFICACIÓN METODOLOGÍA XP .....	77
3.3.1	Historias de Usuario .....	77
3.3.2	Planificación de Iteraciones .....	81
3.3.2.1	Plan de Entregas del Proyecto (Release Planning).....	83
3.4	DISEÑO .....	84
3.4.1	Modelo Relacional de la Base de Datos .....	84
3.4.2	Modelado UWE .....	84
3.5	FASE DE DESARROLLO Y CODIFICACIÓN .....	95
3.5.1	Modelo de Implementación.....	95
3.6	PRUEBAS.....	102
3.6.1	Pruebas de Caja Blanca .....	102
3.6.2	Pruebas de Caja Negra .....	105
4	CALIDAD Y SEGURIDAD .....	108
4.1	Calidad de Software.....	108



4.1.1	Usabilidad .....	108
4.1.2	Funcionabilidad .....	110
4.1.3	Confiabilidad.....	114
4.1.4	Mantenibilidad .....	115
4.1.5	Portabilidad .....	116
4.1.5.1	Resultados.....	117
4.2	SEGURIDAD DEL SISTEMA .....	117
4.2.1	Seguridad.....	117
4.2.1.1	Seguridad a nivel usuario.....	118
4.2.1.2	Seguridad a nivel aplicación.....	118
4.2.1.3	Seguridad a nivel Base de Datos .....	118
5	COSTO DEL SOFTWARE.....	119
5.1	COCOMO II .....	119
5.1.1	Costo del desarrollo del software.....	119
5.1.2	Costo de Implementación del Software .....	120
5.1.3	Costo por Mes de Proyecto .....	123
5.1.4	Costo de Implementación de la aplicación.....	123
5.1.5	Costo de Elaboración del Proyecto.....	123
5.1.6	Costo total del Software .....	124
6	CONCLUSIONES Y RECOMENDACIONES.....	125
6.1	CONCLUSIONES .....	125
6.2	RECOMENDACIONES .....	125
	BIBLIOGRAFÍA.....	127
	ANEXOS.....	131

## ÍNDICE DE FIGURAS

Figura 1: Organigrama de la Empresa TESICON .....	13
Figura 2: Capas de la Ingeniería de Software .....	20
Figura 3: Características de XP .....	23
Figura 4: Fases de la Metodología XP .....	25
Figura 5: Modelo Propuesto para un Historia de Usuario .....	26
Figura 6: Ciclos Iterativos.....	28
Figura 7: Fase IV Pruebas .....	30
Figura 8: Modelo Propuesto Historia de un Usuario .....	31
Figura 9: Modelos de UWE .....	38
Figura 10: Nombres de Estereotipos y sus iconos de casos de uso.....	39
Figura 11: Modelo de Casos de Uso .....	40
Figura 12: Modelo de Contenido .....	41
Figura 13: Descripción de Estereotipos.....	42
Figura 14: Modelo de Navegación .....	43
Figura 15: Modelo de Presentación .....	44
Figura 16: Pagina de Presentación .....	45
Figura 17: Modelo de Proceso .....	46
Figura 18: Flujo de Proceso de Inicio .....	47
Figura 19: Flujo de Proceso Buscar .....	48
Figura 20: Modelo Cliente/Servidor.....	49
Figura 21: Ciclo de vida XP.....	51
Figura 22: Modelo Vista Controlador.....	55
Figura 23: Características del Modelo de Calidad del Producto .....	58
Figura 24: Modelo de Estimación de Costos COCOMO II.....	63
Figura 25: Conceptualización Básica de COCOMO .....	64
Figura 26: Esquema de Modos de Desarrollo .....	65
Figura 27: Valores Constantes por Modo Desarrollo.....	65
Figura 28: Ecuaciones por tipo de modelo COCOMO: Básico e Intermedio.....	66
Figura 29: Ecuaciones por Tipo de Modelo COCOMO: Básico e Intermedio .....	67

Figura 30: Caso de Uso General del Sistema .....	85
Figura 31: Diagrama Caso de Uso Registrar Ventas.....	86
Figura 32: Diagrama Caso de Uso Registrar Compras .....	86
Figura 33: Diagrama de Contenido del Sistema.....	87
Figura 34: Diagrama Navegacional General .....	88
Figura 35: Diagrama de Presentación Ingreso al Sistema .....	89
Figura 36: Ingreso al Sistema .....	89
Figura 37: Diagrama de Presentación del Administrador .....	90
Figura 38: Diagrama de Presentación Administración de Usuarios.....	91
Figura 39: Diagrama de Presentación Listar Categoría.....	92
Figura 40: Diagrama de Presentación Listar Marca .....	92
Figura 41: Diagrama de Presentación Listar Producto .....	93
Figura 42: Diagrama de Presentación Ventas.....	94
Figura 43: Inicio de Sesión del Sistema .....	95
Figura 44: Página Principal del Sistema.....	96
Figura 45: Registro de Usuario .....	96
Figura 46: Asignación de Rol .....	97
Figura 47: Listado de Usuarios del Sistema.....	97
Figura 48: Lista de Producto .....	98
Figura 49: Registro de Abastecimiento .....	99
Figura 50: Realizar Pedido de Venta .....	99
Figura 51: Confirmar Pedido de la Venta Realizada .....	100
Figura 52: Listar Proveedores.....	100
Figura 53 : Reporte de Venta Diaria.....	101
Figura 54: Reporte de Inventarios.....	101
Figura 55: Prueba de Caja Blanca .....	102
Figura 56: Autenticación Exitosa- Autenticación Fallida .....	105

## ÍNDICE DE TABLAS

Tabla 1: Comparativa Metodología Tradicional y Metodología Ágil .....	22
Tabla 2: Categoría de las Funciones de los Requerimientos del Sistema .....	33
Tabla 3: Categoría de las funciones de los Requerimientos del Sistema .....	74
Tabla 4: Requerimientos Funcionales .....	74
Tabla 5: Referencias No Funcionales .....	76
Tabla 6: Historia de Usuario1 – Administración de Usuarios .....	77
Tabla 7: Historia de Usuario 2 - Administrador de Categoría .....	78
Tabla 8: Historia de Usuario 3 - Administrador de Productos .....	78
Tabla 9: Historia de Usuario 4 - Administrador de Compras.....	79
Tabla 10: Historia de Usuario 5 - Administrador de Ventas .....	79
Tabla 11: Historia de Usuario 6 - Administrador de Proveedores .....	80
Tabla 12: Historia de Usuario 6 - Inventario .....	80
Tabla 13: Historia de Usuario 8 - Reportes .....	81
Tabla 14: Descripción de la Tareas de Iteración .....	81
Tabla 15: Plan de Entrega .....	83
Tabla 16: Valores de Entrada - Proceso de inicio del usuario .....	106
Tabla 17: Prueba de Caja Negra Inicio de Usuario .....	106
Tabla 18: Calculo de Usabilidad.....	109
Tabla 19: Parámetros de Medición .....	110
Tabla 20: Conteo de Parámetros PF.....	111
Tabla 21: Valores de Ajuste de Complejidad .....	112
Tabla 22: Medición de Acuerdo al Resultado Obtenido.....	117
Tabla 23: Modo de Desarrollo COCOMO intermedio .....	120
Tabla 24: Multiplicadores de Esfuerzo ME .....	121
Tabla 25: Costo de Elaboración .....	124
Tabla 26: Resultados Finales.....	124

# **CAPÍTULO I**

## **MARCO PRELIMINAR**

# 1 MARCO PRELIMINAR

## 1.1 INTRODUCCIÓN

En una organización o empresa, el análisis y diseño de sistemas de información es el proceso de estudiar su situación con la finalidad de observar cómo trabaja y así decidir si es necesario realizar una mejora.

La información que puedan proporcionar a la empresa es esencial y de mucha importancia con el fin de lograr objetivos de corto o mediano plazo, y lo más importante el acceso a la información debe ser actualizada, rápida, de calidad, objetiva, completa, aplicable y debe satisfacer las necesidades de la empresa ya que sin ella no podemos tomar decisiones adecuadas, como también debe estar disponible en cualquier momento que se necesite. Los sistemas se han convertido en una pieza fundamental en una empresa, para el crecimiento y desarrollo de toda empresa, a medida que va creciendo la empresa también va creciendo la cantidad de información, por lo tanto, la empresa requiere tener el control y seguimiento de las transacciones diarias, las entradas y salidas de productos, esto es determinante para tomar decisiones estratégicas.

TESICON<sup>1</sup> es una empresa comercial que adquiere mercadería y se dedica a la venta e instalación especializada en servicios de sistemas de control para personal, sistemas de seguridad, sensores sistemas de vigilancia, la cual se dedica a la adquisición y ventas de dichos productos, cuenta con una unidad de almacén, está a su cargo la entrada y salida de productos de manera manual y semi-manual lo cual ocasiona pérdida de tiempo, por esta razón deciden contar con un sistema de información para el registro y control de compras, ventas e inventarios para así obtener un mejor desempeño en el registro y control de productos en el almacén.

---

<sup>1</sup> TESICON: Tecnologías en Sistemas de Control

El presente proyecto de grado, propone solucionar los procedimientos manuales de la compra y venta las cuales generan retraso en la entrega de mercadería requerida, retraso en entrega de informes, errores frecuentes en los registros así mismo errores en las salidas de los mismos, errores en realizar el inventario para futuras solicitudes de abastecimiento y la falta de comunicación con el superior. Con el desarrollo e implementación del sistema, este le permitirá a la empresa registrar y controlar los procesos de entrada y salida de los materiales existentes, además ayudará de gran manera a la empresa a tener toda la información de inventarios así se reducirá el tiempo en la elaboración de reportes, se mantendrá siempre al día, la información que provea será real y oportuna.

El método de inventario que se utilizara es el método PEPS primeros en entrar y primeros en salir, para el desarrollo del presente se utilizará la metodología UWE (UML), esta no limita el número de vistas posibles de una aplicación, también para el desarrollo se utilizará el modelo vista controlador MVC que usa Codeigniter, que es un tipo de diseño que separa en capas definidas del desarrollo de una aplicación y como gestor de base de datos se utilizara Postgresql, que hacen el sistema sea de mejor administración.

## **1.2 ANTECEDENTES**

### **1.2.1 Antecedentes Institucionales**

La empresa TESICON está ubicada en la ciudad de La Paz – Bolivia, en la Plaza Alonzo de Mendoza, Zona San Sebastián, Edificio Santa Anita, la empresa está inscrita con un numero de NIT. 6106543014 fue fundada por el empresario Sr. Reynaldo Franco Ramos un visionario empresario boliviano.

El 3 de agosto del 2010 nace TESICON con la visión de generar fuentes de trabajo y ser una de las mejores empresas reconocidas a nivel nacional en servicios de tecnología biométrica, solida, transparente y confiable ante todos sus clientes y proveer equipos electrónicos de seguridad para toda Bolivia. En la actualidad

TESICON importa, comercializa y distribuye equipos de seguridad electrónica, redes y comunicaciones. Con casa matriz en la ciudad de La Paz, dedica una gran parte de su esfuerzo a la capacitación y entrenamiento de sus canales de distribución.

### **1.2.2 Antecedentes Académicos**

En el presente apartado se muestran los trabajos relacionados al tema objeto de estudio.

(Lopez, 2010). “DISEÑO DE UN SISTEMA DE CONTROL INTERNO DE INVENTARIOS EN LA CAFETERÍA LOZADA”. El objetivo principal es realizar el control riguroso relativo al ingreso y despacho de la mercadería con las cuales se obtendrá como resultado inventarios con menos porcentaje de error. Realizado en la Universidad Tecnológica Equinoccial de Santo Domingo. Ecuador.

(Chiri, 2009). “SISTEMA DE ENTRADAS Y SALIDAS E INVENTARIOS EN BOLITAL S.R.L.”. El objetivo principal es desarrollar e implementar un sistema de entrada y salida de inventarios, de tal forma que se mejore los procesos operativos y administrativos. Realizado en la Universidad Mayor de San Andrés. La Paz.

(Ramos, 2006). “SISTEMA DE CONTROL DE INVENTARIO PARA LABORATORIOS CRESPAL S.A. CASO: REGIONAL SUCRE”. El objetivo principal fue de implantar un sistema para dar la solución a los distintos problemas y satisfacer los requerimientos de la Regional de Sucre, los cuales eran mejorar el control de la entrada y salida de medicamentos desde y hasta almacenes. Realizado en la Universidad Mayor de San Andrés. La Paz.

(Mendoza, 2016) “SISTEMA WEB DE CONTROL DE INVENTARIOS, MANUFACTURACIÓN Y PRODUCTO FINAL PARA LA EMPRESA INDUSTRIAL COMERCIAL DE ALIMENTOS INCADEX S.R.L”. CASO: “INCADEX S.R.L”. El objetivo principal es apoyar a la empresa, mediante la implementación de un



sistema que permita controlar el almacén, manufacturación y producto final a ser expuesto en venta. Realizado en la Universidad Mayor de San Andrés. La Paz.

(Callisaya, 2017) “SOFTWARE DE GESTIÓN Y CONTROL DE INVENTARIOS. CASO: AGADON S.R.L.”. El objetivo Principal es implementar un software de gestión y control de ventas e inventarios, de forma que los gerentes hagan un control adecuado sobre los procesos que se realizan. Realizado en la Universidad Mayor de San Andrés. La Paz.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

En la unidad de almacenes de la empresa TESICON, el principal problema es la gran cantidad de papeles, registros, anotaciones, que se genera en el momento de realizar el almacenamiento de registro, ya que existen errores al realizar informes de inventario, errores de entrada y salida de mercadería en los almacenes, además los registros no cuentan con un ingreso confiable y seguro de las mercaderías existentes ya que se realiza de forma manual, esto causa en muchos casos, pérdida de información y económicas para la empresa lo más importante es el tiempo al momento de procesar el seguimiento de la mercadería.

Teniendo en cuenta la ausencia de datos confiables al mantener un inventario manual y sin un método adecuado, la empresa TESICON no cuenta con un sistema que le permita administrar todos los procesos anteriormente mencionados, por lo que no se puede tomar decisiones estratégicas correctas, por no contar con la información oportuna y confiable.

Por todo lo expuesto anteriormente se plantea la siguiente interrogante en el problema principal.

### **1.3.1 Problema Principal**

¿Cómo mejorar el manejo de la información para registro y control de compras, ventas e inventarios de la empresa TESICON a través del desarrollo del Sistema de Información altamente disponible?

### **1.3.2 Problemas Específicos**

- El manejo de la información en la empresa TESICON es manual lo que ocasiona información imprecisa.
- No se cuenta con registro de abastecimiento el cual ocasiona imprecisión al momento de realizar la provisión.
- No se cuenta con un catálogo de productos lo que genera desorden a la hora de seleccionar por categoría y marcas.
- Se desconoce la información de los clientes y proveedores ocasionando desconocimientos de los mismos.
- No existe un registro y control de ventas lo que causa deficiencias al momento de realizar el balance del día.
- Existe retraso de entrega de informes de stock lo que causa pérdida de tiempo a la hora de realizar la venta.
- Manejo inadecuado del registro de inventarios, ventas del día lo que ocasiona desinformación en los totales y parciales de los productos.

## **1.4 OBJETIVOS**

### **1.4.1 Objetivo General**

Desarrollar un sistema de información para el Registro y Control de Compras, Ventas e Inventarios para la empresa TESICON que facilite información oportuna, actualizada y confiable que coadyuve en la correcta toma de decisiones.

### **1.4.2 Objetivos Específicos**

Los objetivos específicos planteados para el proyecto son:

- Automatizar el registro de todos los movimientos de la empresa TESICON evitando información imprecisa.
- Desarrollar un módulo de Provisión de abastecimiento evitando compras en demasía.
- Implementar un módulo de Productos que registre por categorías y marcas evitando desorden de los mismos.
- Desarrollar un módulo para registro de clientes y proveedores previniendo desconocimiento a la hora de la atención de los mismos.
- Implementar un módulo de ventas para evitar el mal arqueado de las ventas del día.
- Desarrollar un módulo de reportes que permita realizar generables por día, por fechas y de inventarios.

## **1.5 JUSTIFICACIÓN**

### **1.5.1 Justificación Técnica**

La Empresa TESICON actualmente cuenta con las herramientas tecnológicas para implementar el sistema tanto el hardware y software, por tanto, el sistema se implementará y beneficiará de gran manera a la empresa ayudando a realizar las tareas cotidianas con mayor comodidad y facilidad y eficiencia.

### **1.5.2 Justificación Económica**

El proyecto que se desarrollará en la empresa TESICON es factible económicamente puesto que permitirá reducir económicamente gastos en el material de escritorio y disminución del tiempo en actividades diarias, ayudará a agilizar los procesos manuales, tendrá un mejor control y seguimiento de la mercadería tanto de entradas como de salidas de productos lo que permitirá obtener una información rápida, oportuna, confiable y fácil de comprender beneficiando a la

empresa y al personal encargado. Además, cabe mencionar que el sistema se realizará con herramientas de software libre y no se realizará gasto extra en la licencia del software y su requerimiento hardware no requiere equipos de última generación y el sistema funcionará en Internet e Intranet.

### **1.5.3 Justificación Social**

Con la implementación del nuevo sistema, se beneficiará a la empresa, a tener un mayor prestigio en la sociedad y tener mayor competitividad en el mercado incrementando la productividad, mejorando el registro y control de material, en cuanto al personal de la empresa tendrán mayor noción en el manejo del sistema puesto que es una herramienta que le permitirá tener un mayor conocimiento en los avances tecnológicos, también el sistema beneficiará:

- Al gerente, porque la información que requiera será de manera inmediata oportuna y confiable.
- Administrativos, les facilitara de gran manera saber el stock de existencia en almacenes para su utilización.
- Población, le beneficiara en la entrega rápida y efectiva del producto, sin demoras.

## **1.6 METODOLOGÍA**

### **1.6.1 Método de Ingeniería**

La metodología que se empleara en el desarrollo del sistema es XP<sup>2</sup> es una metodología ágil, lo que caracteriza a XP, al igual que al resto de métodos Agiles es un ciclo de vida dinámico, realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, por lo que trabaja en equipo entre el cliente y desarrollador. (Beck, 2002).

---

<sup>2</sup> XP: Acrónimo de Extreme Programming – Programación Extrema

## 1.6.2 Metodología de Desarrollo

Para el diseño y modelado se utilizará el método UWE<sup>3</sup> es una propuesta basada en UML<sup>4</sup> y en el proceso unificado para modelar aplicaciones web. Esta propuesta está formada por una notación para especificar el dominio (basada en UML) y un modelo para llevar a cabo el desarrollo del proceso de modelado. (Web Engineering Group, 2016)

El método UWE contempla los siguientes modelos:

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

## 1.7 HERRAMIENTAS

Para el desarrollo del sistema de control de inventarios vía web se tiene planificado implementar la Base de Datos PostgreSQL, como el lenguaje de programación PHP<sup>5</sup>, como plataforma de desarrollo Visual Studio Code, como servidor web se utilizará Apache, para el lado del cliente Bootstrap, CSS y Codeigniter como framework.

**PostgreSQL:** Es un gestor de base de datos Ampliamente popular Ideal para tecnologías Web el cual es fácil de administrar y su sintaxis SQL es estándar y fácil de aprender, multiplataforma con capacidades de replicación de datos además tiene

---

<sup>3</sup> UWE: Acrónimo de Basad Web Engineering

<sup>4</sup> UML: Acrónimo de Lenguaje Unificado de Modelo

<sup>5</sup> PHP: Acrónimo de Hypertext Preprocessor (preprocesador de hipertexto)

soporte empresarial disponible no se encuentra controlado por ninguna corporación. (Chavez, 2012).

**PHP:** PHP es un lenguaje de programación que te permite manipular varios elementos del navegador como la cache o cookies también realizar conexiones con Bases de Datos como Oracle, SQL<sup>6</sup> o MySQL además nos puede ofrecer sacar el máximo provecho al servidor, su rapidez de desarrollo, multiplataforma, código libre, portabilidad, acceso a muchas fuentes de datos, orientación a objetos. (Lerdorf, 2009).

**Visual Studio Code:** Es un editor de código fuente desarrollado por Microsoft para Windows, Linux. Incluye soporte para la depuración, control integrado de Git, resaltando de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

**Servidor Apache:** Es un servidor de páginas Web HTTP<sup>7</sup> de código abierto que sirve para colocar varias plataformas entre ellas Windows, Unix y otras. (McCool, 2018).

**Bootstrap:** bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web con CSS<sup>8</sup> y JavaScript, cuya particularidad es de adaptar la interfaz del sitio web se adapta automáticamente al tamaño de una PC, Tablet u otros dispositivos.

**CSS:** En español hojas de estilo en cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y para muchas aplicaciones.

---

<sup>6</sup> SQL: Acrónimo de Structured Query Language (Lenguaje de Consulta Estructurada)

<sup>7</sup> HTTP: Acrónimo de Hyper Text Transfer Protocol, (protocolo de transferencia de hipertextos)

<sup>8</sup> CSS: Acrónimo de Cascading Style Sheets (Hojas de estilo en cascada)

**Codeigniter:** Es un framework para el desarrollo de aplicaciones en php que utiliza el MVC, permite a los programadores web mejorar la forma de trabajar y hacerlo a mayor velocidad. Al igual que cualquier framework está pensado para gente que tiene un dominio al menos medio del lenguaje de programación PHP. (Álvarez, Codeigniter, 2009).

## 1.8 LÍMITES Y ALCANCES

### 1.8.1 Limites

El presente proyecto se limita, solo a realizar análisis diseño e implementación del sistema de información de compras, ventas e inventarios de la empresa TESICON, los módulos especificados en el proyecto son los requeridos de manera inmediata.

El proyecto no contempla lo relacionado con contabilidad, activos fijos, o todo lo que esté relacionado con los aspectos administrativos

El presente proyecto se enfocará en el control de inventarios y como priorización se propone lo siguiente:

- **Módulo de administración de usuarios:** este módulo se encarga de dar privilegios y se aplicara técnicas de seguridad, en cuanto al ingreso del sistema permite al usuario ingresar con un usuario y contraseña para poder interactuar con el sistema.
- **Módulo de proveedores:** en este módulo el único autorizado es el administrador donde se brindará toda la información del proveedor. Allí se podrá modificar, eliminar y guardar cualquier información.
- **Modulo Provisión:** se podrá ver los datos del abastecimiento las compras que se realizaron con cada detalle y característica del producto y estos quedan actualizados.
- **Módulo de registro de venta:** se genera el registro de venta detallando el producto y sus características para el administrador y así mismo para el encargado de ventas.

- **Módulo de reportes ventas:** se obtendrá datos inmediatos sobre los productos, un balance de las ventas que se realizaron durante el día, el porcentaje y los productos que se vendieron con su respectivo detallado.

## **1.9 APORTES**

El aporte más destacado será el mismo sistema aplicada en la empresa y que no ha sido desarrollado, también podemos mencionar que gracias al desarrollo e implementación del sistema esta fortalece de gran manera a la empresa el cual permite tener una información centralizada en una base de datos donde se logrará el mejor control de existencias y de stock, mejorando así de una manera eficiente y confiable lo que produce un mejor servicio.

Se facilita el trabajo del personal de la empresa brindando información y podrá generar reportes en tiempo real, ayudando a facilitar la búsqueda de información. Así mismo se logrará la sistematización del proceso aplicable y eficiente para mejorar el proceso en la toma de decisiones lo que es vital para la empresa TESICON.



## **CAPÍTULO II**

### **MARCO TEÓRICO**

## 2 MARCO TEÓRICO

### 2.1 DESCRIPCIÓN DE LA EMPRESA

La empresa TESICON es una empresa que desde hace 10 años está establecida en el mercado nacional teniendo gran aceptación en ciudades como: La Paz, Cochabamba, Potosí, Oruro, Santa Cruz, Beni. Se encuentra en la ciudad de La Paz – Bolivia ubicado en la Plaza Alonzo de Mendoza, zona San Sebastián Edificio Santa Anita, la empresa está inscrita con un numero de NIT 6106543014 fue fundada por el empresario Sr. Reynaldo Franco Ramos un visionario empresario boliviano.

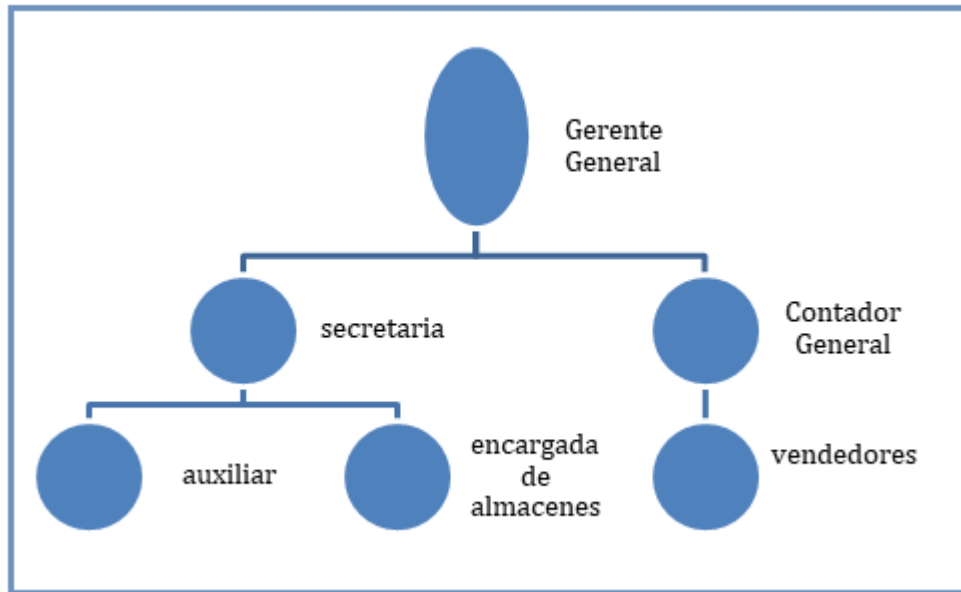
**Objetivo:** Brindar las mejores soluciones a las necesidades de seguridad de la sociedad y mercado a través de la eficiencia y profesionalismo basados en procesos y una cultura de calidad de servicio y atención al cliente.

#### 2.1.1 Misión y Visión

- ✓ **Misión:** Ser una de las mejores empresas reconocidas a nivel nacional en servicios de tecnología innovadora, de calidad y confianza ante todos nuestros clientes, nos caracterizamos por utilizar una metodología de trabajo transparente y muy cercano al cliente brindándoles soluciones inmediatas.
- ✓ **Visión:** Ser la empresa más confiable, satisfaciendo las necesidades tecnológicas de la sociedad a través de nuestros clientes, ayudándolos a alcanzar el éxito.

En la siguiente figura se muestra la estructura orgánica de la empresa TESICON.

**Figura 1:** Organigrama de la Empresa TESICON



Fuente: Tecnología en Sistemas de Control

## **2.2 SISTEMA**

Un sistema es un conjunto de elementos relacionados entre sí para realizar un conjunto de funciones específicas.

También puede mencionarse la noción de sistema informático, muy común en las sociedades modernas. Este tipo de sistemas denominan al conjunto de hardware, software y soporte humano que forman parte de una empresa u organización. Incluyen ordenadores con los programas necesarios para procesar datos y las personas encargadas de su manejo. (Von Bertalanffy, 2001).

## **2.3 INFORMACIÓN**

La información está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. (Laudon & Laudon, 2012).

## 2.4 SISTEMA DE INFORMACIÓN

Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común. En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización, para apoyar a la toma de decisiones.

Sistema de información aplicada a una empresa es un elemento clave y fundamental para el buen funcionamiento de cualquier tipo de organización. Determina la consecución de los objetivos de la empresa repercutiendo en el éxito o fracaso del negocio. Para que la información cumpla con sus objetivos es necesario que posea ciertas características:

- **Debe ser relevante:** tiene que ser importante, que sea la información que nosotros necesitamos.
- **Debe estar actualizada:** debe utilizarse en el momento de ser generada.
- **Debe ser rápida:** el acceso a la información debe realizarse de forma rápida y sencilla.
- **Debe ser económica:** la obtención de la información no debe generar un coste elevado para la empresa.
- **Debe ser de calidad:** es importante que la información carezca de errores y sea completa.
- **Debe ser objetiva:** no cabe opción a subjetividades.
- **Debe ser completa:** el tener la información incompleta es peor que no tener nada.
- **Debe ser aplicable:** la información debe ser adecuada para la toma de una decisión, además de ser importante y pertinente.

Una información que no satisfaga estas características carecerá de valor y, por tanto, no será válida para la organización.

## **2.5 REGISTRO**

Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad. Se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice, aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

## **2.6 COMPRAS**

El término compras se refiere a la acción que realiza una persona de adquirir un bien o servicio mediante dinero. La compra es realizada por el comprador quien interactúa con otra persona, el vendedor, que ofrece cierto producto. Las compras se efectúan, por lo tanto, mediante un sistema de compra-venta donde el determinante es cierto valor en efectivo. Se trata de actividades opuestas; quien compra entrega dinero para recibir un bien o servicio, mientras que quien vende entrega el bien o servicio a cambio del dinero.

## **2.7 VENTAS**

Venta es una acción que se genera de vender un bien o servicio a cambio de dinero, también puede referirse a un objeto o servicio que se encuentra a disposición del público, lo cual quiere decir que aún no está vendido, por tanto, se encuentra en venta.

Desde el punto de vista contable y financiero, la venta es el monto total cobrado por productos o servicios prestados. En cualquier caso, las ventas son el corazón de cualquier negocio, es la actividad fundamental de cualquier aventura comercial.

## **2.8 INVENTARIOS**

El inventario representa la existencia de bienes tangibles almacenados destinados a realizar una operación, sea de compra, alquiler, venta, uso o transformación. Debe aparecer, contablemente, dentro del activo como un activo circulante. Los

inventarios se definen como bienes ociosos almacenados en espera de ser utilizados. (Orellana Funes, 2016)

El inventario constituye las partidas del activo corriente que están listas para la venta, es decir, toda aquella mercancía que posee una empresa en el almacén valorada al costo de adquisición, para la venta, o actividades productivas. En los últimos años, las empresas o instituciones comerciales como las de ventas, conceden mayor atención cada día a los inventarios porque es un proceso recomendable para cualquier compañía, empresa, institución e incluso, para un hogar. Su ejecución es de vital importancia para mantener una mejor organización, evitando así, la pérdida de material y de inversión. (Orellana Funes, 2016)

El inventario también es considerado como uno de los activos más grandes de una empresa. Se registra en el balance general como en el estado de los resultados permitiendo determinar el costo de las mercancías vendidas en un periodo determinado.

**Objetivo de los inventarios:** El objetivo de los inventarios, es identificar, proveer o distribuir adecuadamente los materiales necesarios a la empresa. Colocándolos a disposición en el momento indicado, para así evitar aumentos de costos o pérdidas de los mismos. Permitiendo satisfacer correctamente las necesidades reales de la empresa, a las cuales debe permanecer constantemente adaptado. Por lo tanto, la gestión de inventarios debe ser atentamente controlada y vigilada.

El mantenimiento y control de inventario es importante para el desarrollo de pequeñas, medianas y grandes empresas. Una mala administración de inventario puede provocar una mala atención al cliente y causar problemas financieros.

El inventario cumple una función vital en una empresa, ya que provee a la compañía de la materia prima necesaria para el mantenimiento de sus actividades dentro del mercado.

### 2.8.1 Métodos de Control de Inventario

Existen 3 tipos de Control de Inventarios las cuales son: Método PEPS, método UEPS y promedio ponderado.

- **Método PEPS (Primeras Entradas, Primeras Salidas):** El método PEPS, también conocido como FIFO por sus siglas en inglés, consiste en identificar los primeros artículos en entrar al almacén para que sean los primeros en ser vendidos o utilizados en la producción. Así se evita, por ejemplo, que productos perecederos caduquen, ya que tendrán un movimiento constante y oportuno.

Bajo este método, los primeros costos que entran al inventario son los primeros costos que salen al costo de las mercaderías vendidas. El costo del inventario final se basa en los costos de las compras más recientes.

- **Método UEPS (Ultimas Entradas, Primeras Salidas):** El método ultimas entradas, primeras salidas dependen también de los costos por compras de un inventario en particular. Bajo este método, los últimos costos que entran al inventario son los primeros costos que salen al costo de mercancías vendidas. Este método deja los costos más antiguos (aquellos del inventario inicial y las compras primeras del periodo) en el inventario final.

- **Promedio Ponderado:** El método del costo promedio ponderado, llamado a menudo método del costo promedio se basa en el costo promedio ponderado del inventario durante el periodo. Este método pondera el costo por unidad como el costo unitario promedio durante el periodo, se utiliza el promedio de estos costos. El costo promedio se determina de la manera siguiente: divide el costo de las mercaderías disponibles para la venta (inventario inicial + compras) entre el número de unidades disponibles.

En el presente proyecto se utilizó el método PEPS.

## 2.8.2 Sistema de Inventarios

Existen básicamente dos sistemas para llevar a cabo los registros de inventario: el sistema periódico y el sistema perpetuo.

**Sistema de inventario periódico:** No existe el control permanente de entradas, salidas y saldo de existencias de mercadería, la información no es útil, es ambigua por tanto causa una deficiencia en la toma de decisiones de los usuarios internos y externos.

El sistema de inventarios periódico no rastrea el inventario continuamente, sino que permite a las organizaciones conocer los niveles de inventarios inicial y final, durante un cierto período de tiempo.

Este sistema rastrea el inventario usando conteos físicos. Cuando se completa el inventario físico, el saldo en la cuenta de compras se suma a la cuenta de inventarios y se ajusta para que coincida con el costo del inventario final.

Existen también algunas desventajas de usar un sistema de inventarios periódico. Primero, mientras se completa el recuento de inventario físico, las actividades comerciales normales están casi suspendidas. Como resultado, los trabajadores pueden realizar los conteos físicos apuradamente, debido a las limitaciones de tiempo.

Los errores y fraudes pueden ser más frecuentes en este sistema, porque no hay un control continuo sobre el inventario. También se hace más difícil identificar dónde ocurren las discrepancias en los conteos de inventario, porque transcurre mucho tiempo entre los recuentos.

**Sistema de inventario perpetuo:** Los sistemas de inventarios perpetuo realizan el control y registros de las existencias se mantienen siempre al día. La información que provee es real y oportuna, requiere el mantenimiento de registros completos de



existencias de mercadería y el costo de mercadería vendida; por tanto, es efectiva en el momento de realizar toma de decisiones.

Con un sistema de inventarios perpetuo, se actualizan continuamente los registros de inventarios, contando las adiciones y sustracciones cuando los artículos de inventarios son recibidos, vendidos, trasladados de una ubicación a otra, seleccionados para consumirlos y desechados.

Algunas organizaciones prefieren este tipo de sistema porque entrega información actualizada de los inventarios y maneja mejor los conteos de inventarios físicos.

El sistema de inventarios perpetuo también es el preferido para hacerle seguimiento al inventario, porque brinda continuamente resultados precisos cuando se administra de forma adecuada.

Este tipo de sistema funciona mejor cuando se utiliza, junto con el inventario de almacén, una base de datos con las cantidades de inventario y las ubicaciones de estantes actualizados en tiempo real por los empleados, utilizando escáneres de código de barras.

En el presente proyecto se utiliza el sistema perpetuo ya que la empresa requiere un control continuo de sus inventarios.

## **2.9 INGENIERÍA DE SOFTWARE**

La ingeniería de software es una tecnología con varias capas como se aprecia en la Figura N° 2 cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad. (Pressman, 2010).

Podemos entonces mencionar que la Ingeniería de Software se basa en procesos, métodos y herramientas parcialmente ordenadas que sirven como guía para los

ingenieros del software durante el proceso de desarrollo, con la finalidad de mejorar la calidad de los proyectos y obtener un producto de software de calidad.

**Figura 2:** Capas de la Ingeniería de Software



Fuente: (Pressman, 2010).

- **Procesos**

El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada. (Pressman, 2010).

La base de la ingeniería del software es la capa de proceso. El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software.

- **Métodos**

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas. (Pressman, 2010).

Los métodos de la Ingeniería del Software indican técnicamente como construir el software. Los métodos abarcan una gran gama de tareas que te llevan a elaborar un software de calidad.

- **Herramientas**

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software. La ingeniería de software aplica diferentes normas y métodos que permiten obtener mejores resultados, en cuanto al desarrollo y uso del software mediante la aplicación correcta de estos procedimientos se pueden llegar a cumplir de manera satisfactoria los proyectos que ayudan a la población. (Pressman, 2010).

Las herramientas tienen como finalidad, el disminuir el estrés y los tiempos de cada fase, para además mejorar los resultados obtenidos y dar mejores propuestas al cliente. Además, el nivel de satisfacción con el uso de herramientas es mucho mayor que sin ellas.

## **2.10 METODOLOGÍA ÁGIL**

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento). (Huanca Cantuta, 2015).

Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes desarrolladores y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no

a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Se puede hacer mención dentro de las metodologías ágiles a: XP (por sus siglas en inglés Extreme Programming), Scrum y Crystal Methodologies. (EcuRed, 2011).

### 2.10.1 Diferencias Entre Metodologías Tradicionales y Ágiles

La siguiente tabla muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil. (Navarro, 2013)

Tabla 1: Comparativa Metodología Tradicional y Metodología Ágil

Metodologías Tradicionales	Metodologías Ágiles
Desarrollo lento debido al estricto cumplimiento de los procesos.	Desarrollo rápido gracias al cumplimiento de las características del manifiesto ágil.
El equipo de trabajo sigue una estricta serie de procesos sin posibilidad de saltarse hacia otros roles y responsabilidades.	El equipo de trabajo puede intercambiar roles en cualquier momento y transferir conocimiento de manera colectiva.
Grupos pequeños y trabajando en el mismo sitio.	Grupos grandes y distribuidos.
El cliente interactúa con el equipo de desarrollo pocas veces	El cliente es parte del equipo de desarrollo constantemente.
El contrato es totalmente fijo e inmodificable	El contrato puede estar sujeto a cambios a lo largo del desarrollo.

Fuente: ( Navarro, 2013))

En el presente proyecto nos enfocaremos en las metodologías ágiles y se profundizará el proceso XP.

## 2.10.2 Programación Extrema XP

Es también conocida como Programación Extrema, es una metodología ágil y flexible utilizada para la gestión de proyectos. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. (Ferrer & Robles, 2002).

### 2.10.2.1 Características de la metodología

Según (Bustamante & Rodríguez, 2014) las características de XP son las siguientes:

Figura 3: Características de XP



Fuente: (Bustamante & Rodríguez, 2014)

- Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del software.
- Es capaz de adaptarse a los cambios de requisitos.

- Los individuos e interacciones son más importantes que los procesos y herramientas.
- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.

#### **2.10.2.2 Valores XP**

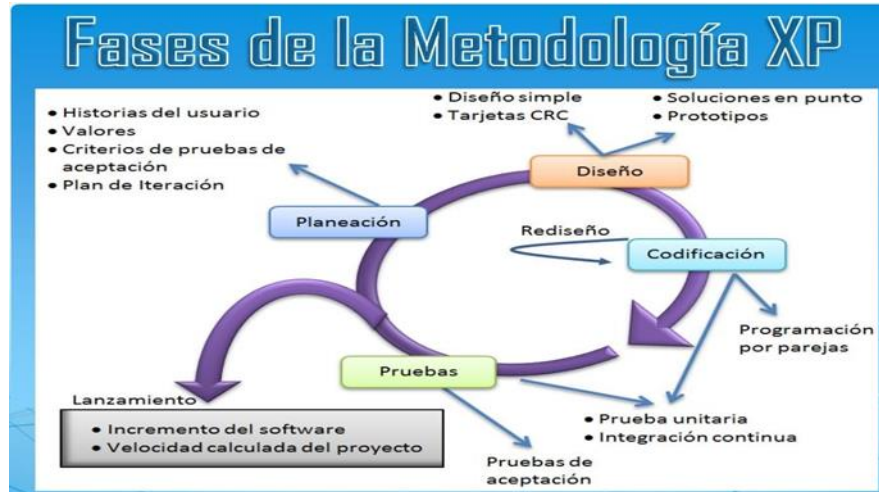
Los valores de la programación extrema se definen como un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado como parte de XP, estos valores son: simplicidad, comunicación, retroalimentación (feedback) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de Extreme Programming Explained. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP.

- **Comunicación:** Eficaz entre los ingenieros de software y otros participantes, XP pone el énfasis en la colaboración estrecha pero informal (verbal) entre los clientes y los desarrolladores.
- **Simplicidad:** Restringe a los desarrolladores para que diseñen solo para las necesidades inmediatas, en lugar de considerar las del futuro.
- **Retroalimentación:** Se obtiene de tres fuentes: del software implementado, del cliente y otros miembros del equipo de software.
- **Valentía:** Un término más apropiado sería disciplina. Por ejemplo, es frecuente que haya mucha presión para diseñar para hoy y reconocer que los requerimientos futuros tal vez cambien mucho, por lo que demandarán repeticiones sustanciales del diseño y del código implementado.
- **Respeto:** Entre sus miembros, entre otros participantes y los integrantes del equipo, e indirectamente para el software en sí mismo. Conforme logra la entrega exitosa de equipo de software.

### 2.10.2.3 Fases de la Metodología XP

XP está clasificada en cuatro fases estructurales: Planificación, Diseño, Desarrollo y Pruebas, las cuales pueden observarse a continuación.

Figura 4: Fases de la Metodología XP



Fuente: (Bustamante & Rodríguez, 2014).

#### a. Fase I: Planeación del proyecto

XP plantea la planificación como un permanente dialogo entre la parte empresarial y técnica del proyecto, en la que los primeros decidirán el alcance, la prioridad, la composición de las versiones y la fecha de las mismas. En cuanto a los técnicos son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las consecuencias de determinadas decisiones, de organizar el trabajo u finalmente de realizar la planificación detallada dentro de cada versión. (Beck, 2002).

- **Historias de usuario:** El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de

posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

El modelo de historia de usuario propuesto por Kent Beck es el siguiente:

**Figura 5:** Modelo Propuesto para un Historia de Usuario

HISTORIA DE USUARIO	
NUMERO:1	NOMBRE HISTORIA DE USUARIO
USUARIO: Autor	ITERACION ASIGNADA
PRIORIDAD EN NEGOCIO: Alta/Media/Baja)	PUNTOS ESTIMADOS
RIESGOS DE DESARROLLO:(Alta/Media/Baja)	PUNTOS REALES
DESCRIPCION: Breve descripción de lo que el sistema, constan de 3 ó 4 líneas escritas.	
OBSERVACIONES:	

Fuente: (Beck, 2002).

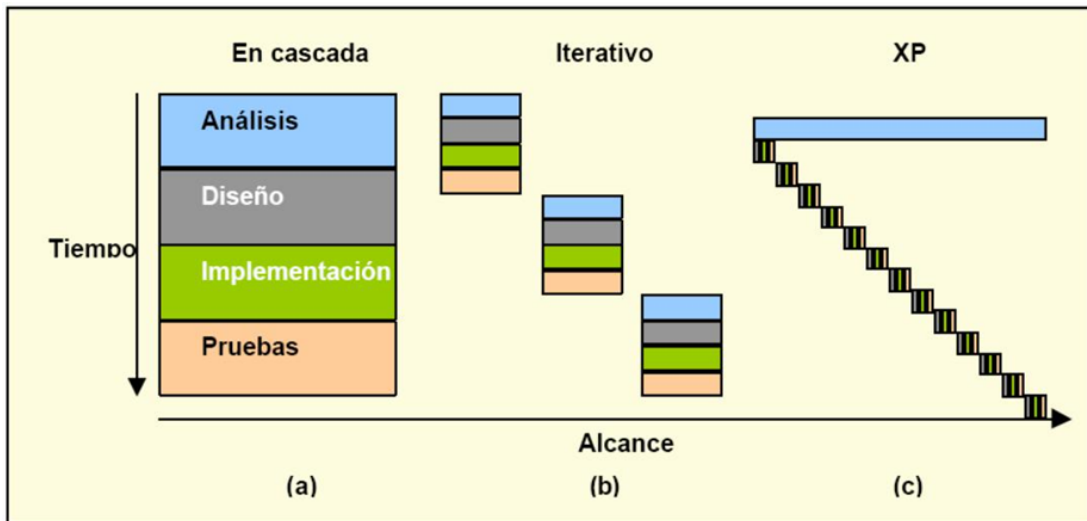
Las historias de usuario tienen tres aspectos:

- ✓ **Tarjeta:** en ella se almacena suficiente información para identificar y detallar la historia.
- ✓ **Conversación:** cliente y programadores número discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera).
- **Pruebas de aceptación:** permite confirmar que la historia ha sido implementada.



- **Reléase planning:** Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Reléase plan", donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un "Reléase plan" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un "Reléase plan" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (Reléase plan: Planificación de publicaciones).
- **Iteraciones:** Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Reléase planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

**Figura 6:** Ciclos Iterativos



Fuente: (Jozkowicz, 2008).

Para cada iteración se define un módulo al conjunto de historia de usuario que se van a implementar, al final de cada iteración se tiene la entrega de un producto, el cual debe superar las pruebas de aceptación que establece el cliente para dar cumplimiento a los requisitos las tareas que no se vean cubiertas por el producto deberán ser tomadas en cuenta para la siguiente iteración.

### **b. Fase II: Diseño**

El diseño XP sigue rigurosamente el principio mantenerlo sencillo. Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después.

XP estimula el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Las tarjetas CRC son el

único producto del trabajo de diseño que se genera como parte del proceso XP. (Pressman, 2010).

### c. Fase III: Codificación

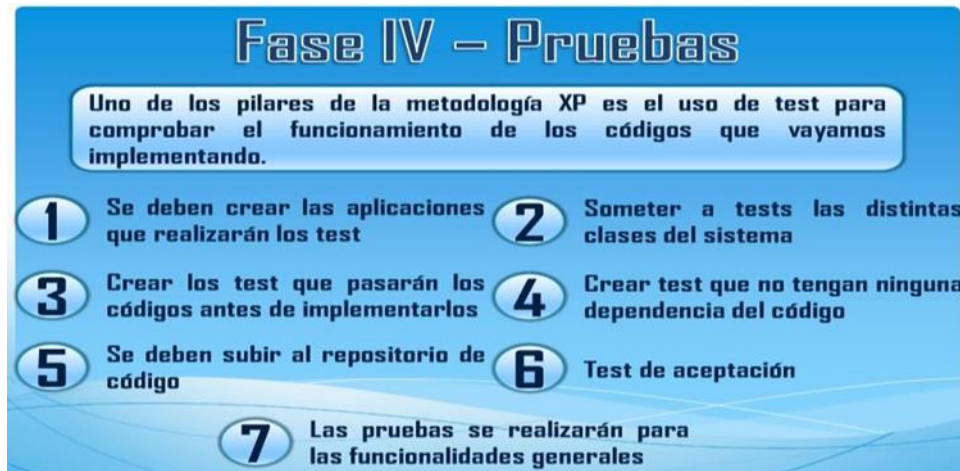
Un concepto clave durante la actividad de codificación (y uno de los aspectos del que más se habla en la XP) es la programación por parejas. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás. En ciertos casos, esto lo lleva a cabo a diario un equipo de integración. En otros, las parejas de programadores tienen la responsabilidad de la integración. Esta estrategia de “integración continua” ayuda a evitar los problemas de compatibilidad de interfaces y brinda un ambiente de “prueba de humo” que ayuda a descubrir a tiempo los errores.

- **Cliente siempre presente:** Uno de los requerimientos de XP es que el cliente este siempre presente y disponible. No solamente para solucionar las dudas del grupo de desarrollo, debería ser parte de este. (Echeverry & Delgado, 2007).
- **Codificar primero la prueba:** Cuando se crea la primera prueba, se ahorra mucho tiempo elaborando el código que la haga pasar, siendo menor el tiempo de hacer ambos procesos que crear el código solamente. (Echeverry & Delgado, 2007)
- **No trabajar más de 40 horas semanales:** Trabajar horas extras absorbe al espíritu y la motivación del equipo. Aquellos proyectos que requiera horas extras para acabarse a tiempo pueden convertirse en un problema en lugar de esto se utilizara las conocidas reuniones también es una mala idea incorporar nueva gente al proyecto. (Echeverry & Delgado, 2007).

#### d. Fase IV Prueba

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. (Bustamante & Rodríguez, 2014), a continuación, se muestra el test que se debe hacer en la fase de pruebas.

Figura 7: Fase IV Pruebas



Fuente: (Bustamante & Rodríguez, 2014).

El uso de los test en XP es el siguiente:

Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

XP enfatiza muchos aspectos relacionados con las pruebas, clasificándolas en diferentes tipos, funcionalidades específicas, indicando quien, cuando y como deben ser implementadas y ejecutadas. (Echeverry & Delgado, 2007).

- **Pruebas Unitarias:** Estas pruebas se aplican a todos los métodos no triviales de todas las clases del proyecto con la condición que no se liberara ninguna clase que tenga asociada su correspondiente paquete de pruebas.

- **Pruebas de Aceptación:** También llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. Las pruebas de aceptación son pruebas de caja negra (“Black box system test”), que representa un resultado esperado de determinada transacción con el sistema.

Los clientes son responsables de verificar que los resultados de éstas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

**Figura 8:** Modelo Propuesto Historia de un Usuario

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>Número:</b>	<b>Usuario:</b>
<b>Historia de usuario:</b>	
<b>Descripción:</b>	
<b>Evaluación de la prueba:</b>	

Fuente: (Beck, 2002)

- **Cuando se Encuentra un Error:** Al momento de escribir un error (“Bug”), debe escribirse una prueba antes de intentar corregirlo, para no volver a cometer el mismo error.

## 2.11 INGENIERÍA DE REQUERIMIENTOS

El diseño y construcción de software de computadora es difícil, creativo y sencillamente divertido. En realidad, elaborar software es tan atractivo que muchos

desarrolladores de software quieren ir directo a él antes de haber tenido el entendimiento claro de lo que se necesita. Argumentan que las cosas se aclararán a medida que lo elaboren, que los participantes en el proyecto podrán comprender sus necesidades sólo después de estudiar las primeras iteraciones del software, que las cosas cambian tan rápido que cualquier intento de entender los requerimientos en detalle es una pérdida de tiempo, que las utilidades salen de la producción de un programa que funcione y que todo lo demás es secundario. Lo que hace que estos argumentos sean tan seductores es que tienen algunos elementos de verdad. Pero todos son erróneos y pueden llevar un proyecto de software al fracaso. (Pressman, 2010).

La ingeniería de requerimientos tiende un puente para el diseño y la construcción. Pero, ¿dónde se origina el puente? Podría argumentarse que principia en los pies de los participantes en el proyecto (por ejemplo, gerentes, clientes y usuarios), donde se definen las necesidades del negocio, se describen los escenarios de uso, se delinear las funciones y características y se identifican las restricciones del proyecto. Otros tal vez sugieran que empieza con una definición más amplia del sistema, donde el software no es más que un componente del dominio del sistema mayor. Pero sin importar el punto de arranque, el recorrido por el puente lo lleva a uno muy alto sobre el proyecto, lo que le permite examinar el contexto del trabajo de software que debe realizarse; las necesidades específicas que deben abordar el diseño y la construcción; las prioridades que guían el orden en el que se efectúa el trabajo, y la información, las funciones y los comportamientos que tendrán un profundo efecto en el diseño resultante. (Pressman, 2010).

### **2.11.1 Obtención de Requerimientos**

Se plantea una clasificación de requerimientos en dos tipos: Los funcionales y los no funcionales. Los requerimientos funcionales definen características sobre el funcionamiento del sistema, es decir, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Por otro lado, se tienen los

requerimientos no funcionales, los cuales determinan características que de una u otra forma puedan limitar el sistema.

**Tabla 2:** Categoría de las Funciones de los Requerimientos del Sistema

Categoría de la función	Significado
Evidente	Debe realizarse y el usuario debería saber que se ha realizado.
Oculto	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, por ejemplo, guardar información en un mecanismo persistente de almacenamiento.
Opcionales	Su inclusión no repercute de forma significativa en costo ni en otras funciones.

Fuente: (Larman, 1999)

### 2.11.2 Tipos de requerimientos

Se clasifican en dos tipos requerimientos estos son:

#### 2.11.2.1 Requerimientos funcionales

Los requerimientos funcionales de un sistema describen con detalle lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. Cuando se expresan como requerimientos del usuario, habitualmente se describen de una forma bastante abstracta. Sin embargo, los requerimientos funcionales del sistema describen con detalle la función de este, sus entradas y salidas, excepciones, etcétera.

La especificación de requerimientos funcionales de un sistema debe ser completa y consistente. La completitud se refiere a que todos los servicios solicitados por el usuario deben estar definidos. La consistencia significa que los requerimientos no deben tener definiciones contradictorias. (Sommerville, 2005).

### **2.11.2.2 Requerimientos no funcionales**

Los requerimientos no funcionales, como su nombre sugieren, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema.

Los requerimientos no funcionales surgen de las necesidades de los usuarios, por la restricción del presupuesto, políticas organizacionales, necesidades interoperabilidad con otros sistemas (software o hardware), o a factores externos como regulaciones de seguridad o legislaciones de privacidad. (Sommerville, 2005).

### **Clasificación de los requerimientos no funcionales**

Según (Sommerville, 2005), clasifica los tipos de requerimientos no funcionales y lo describe de la siguiente manera:

- **Requerimientos del producto:** Especifican el comportamiento del producto, como, por ejemplo, la rapidez de ejecución, cantidad de memoria que se requiere; los requerimientos de fiabilidad que fijan la tasa de fallos para que el sistema sea aceptable; los requerimientos de portabilidad y los de usabilidad.
- **Requerimientos organizacionales:** Derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador, como los requerimientos de implementación (lenguajes de programación o método de



diseño), requerimientos de entrega que especifican la entrega del producto y su documentación.

- **Requerimientos externos:** Derivan de los factores externos del sistema y de su proceso de desarrollo. Puede incluir los requerimientos de interoperabilidad que define la interacción del sistema con otras organizaciones, requerimientos legislativos que aseguran el funcionamiento del sistema dentro de la ley, y los requerimientos éticos.

Además de los requerimientos funcionales y no funcionales, se hace una separación según los niveles de descripción.

### **2.11.2.3 Requerimientos del usuario**

Son los requerimientos abstractos de alto nivel, son declaraciones de los servicios que se espera que el sistema proporcione y las restricciones bajo las cuales debe funcionar.

Los requerimientos del usuario deben describir tanto los requerimientos funcionales como los no funcionales de manera comprensible para el usuario, debe especificar el comportamiento externo del sistema, estos requerimientos deben enfocarse en los recursos principales sin la necesidad de especificar el diseño del sistema.

### **2.11.2.4 Requerimientos del sistema**

Son aquellos requerimientos que describen detalladamente el comportamiento del sistema, las funciones, servicios y las restricciones operativas.

## **2.12 METODOLOGÍA UWE**

### **2.12.1 Lenguaje Unificado de Modelado (UML)**

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y

utilizado en la actualidad; está respaldado por el OMG (Object Management Group). (Aspiazu Gutierrez, 2016).

Es un lenguaje grafico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (Aspiazu Gutierrez, 2016)

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. (Aspiazu Gutierrez, 2016)

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar. (Aspiazu Gutierrez, 2016).

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos. (Aspiazu Gutierrez, 2016).

### **2.12.2 UWE – Based Web Engineering (UML)**

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. (Aspiazu Gutierrez, 2016).

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. (Aspiazu Gutierrez, 2016).

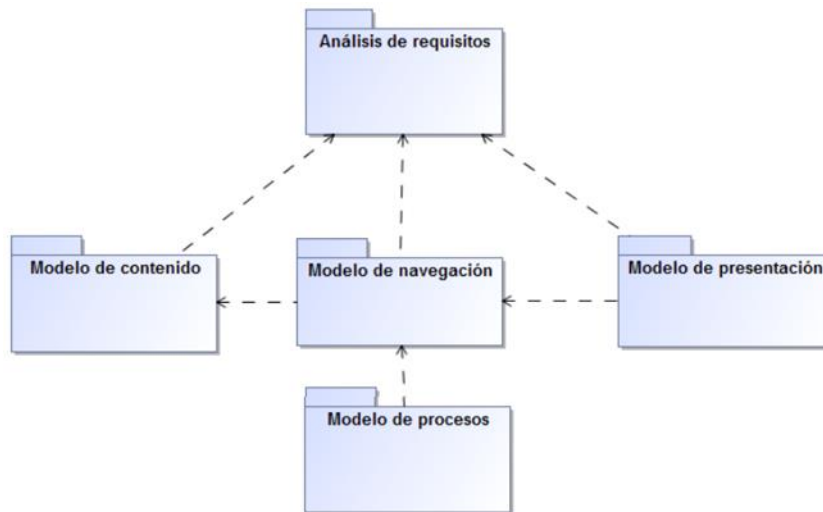
UWE permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo. (Nieves G., Ucan P., & Menendez D., 2014).

En su implementación se deben contemplar las siguientes etapas y modelos:

- **Análisis de requisitos:** Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.
- **Modelo de contenido:** Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- **Modelo de navegación:** Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- **Modelo de presentación:** Representa las interfaces de usuario por medio de vistas abstractas.
- **Modelo de proceso:** Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

Como se hace notar, UWE provee diferentes modelos que permite describir una aplicación Web desde varios puntos de vista abstractos, dichos modelos están relacionados tal como se muestra en la siguiente figura. (Aspiazu Gutierrez, 2016).

**Figura 9:** Modelos de UWE



Fuente: (Nieves G., Ucan P., & Menendez D., 2014).

Cada uno de estos modelos se representa como paquetes UML, dichos paquetes son procesos relacionados que pueden ser refinados en iteraciones sucesivas durante el desarrollo del UWE. (Nieves G., Ucan P., & Menendez D., 2014).

El análisis de requisitos en UWE se modela con casos de uso. Está conformado por los elementos actor y caso de uso. En este sentido, los actores se utilizan para modelar los usuarios de la aplicación Web. (Nieves G., Ucan P., & Menendez D., 2014).

El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases. Basado en el análisis de requisitos y el modelo de contenido se obtiene el modelo de navegación. Este se representa con clases de navegación que serán explicados en el caso de estudio de este artículo. Basado en el modelo de navegación y en los aspectos de interfaz usuario (requisitos), se obtiene el modelo de presentación. Dicho modelo describe la estructura de la interacción del usuario con la aplicación Web. El modelo de navegación puede ser extendido mediante clases de procesos. El modelo del proceso representa el aspecto que

tienen las acciones de las clases de proceso. (Nieves G., Ucan P., & Menendez D., 2014).

### 2.12.2.1 Modelo de Requisitos

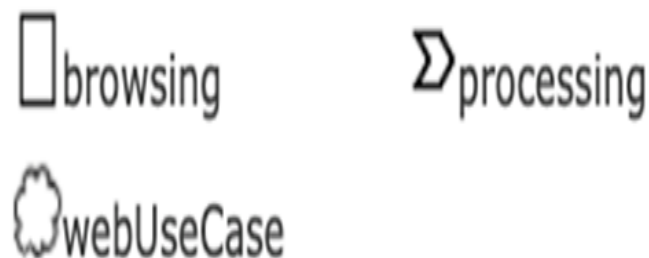
Una de las primeras actividades en la construcción de aplicaciones Web es la identificación de los requisitos, en UWE el modelado de requisitos consiste de dos partes: Casos de uso de la aplicación y sus relaciones y actividades describiendo los casos de uso en detalle.

#### Modelo de Casos de Uso

El diagrama de casos de uso está conformado por los elementos actor y caso de uso. Los actores se utilizan para modelar los usuarios de la aplicación Web. Los casos de uso se utilizan para visualizar las diferentes funcionalidades que la aplicación tiene que proporcionar. (Aspiazu Gutierrez, 2016).

En UWE se distinguen casos de uso estereotipados con «browsing» y con «processing» para ilustrar si los datos persistentes de la aplicación son modificados o no. "SearchContact" por ejemplo, modela la búsqueda de contactos y por ello lleva el estereotipo «browsing» pues los datos son solamente leídos y presentados al usuario. Los otros casos de uso por el contrario modelan cambios, lo que se especifica con el estereotipo «processing». (Web Engineering Group, 2016).

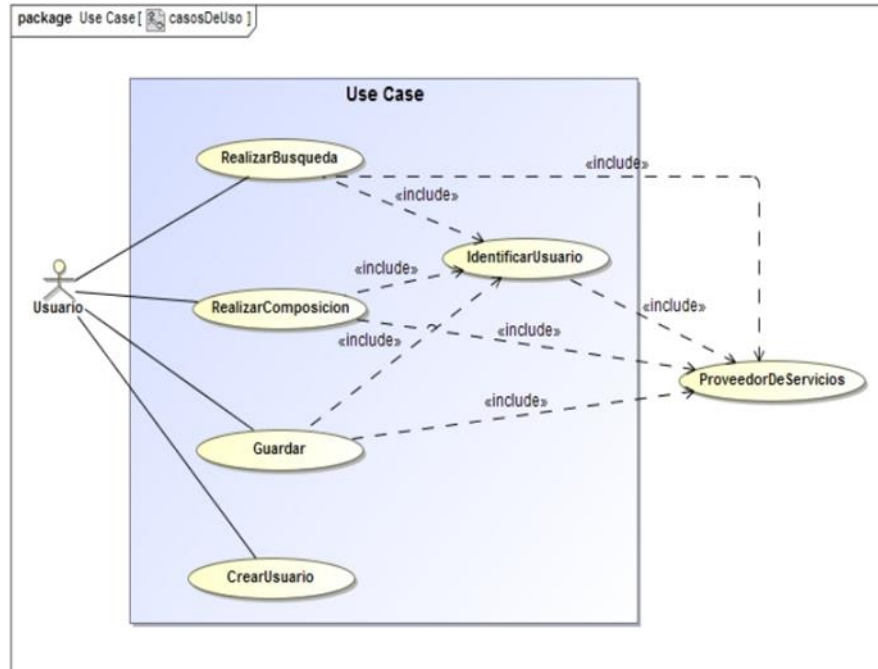
**Figura 10:** Nombres de Estereotipos y sus iconos de casos de uso



Fuente: (Web Engineering Group, 2016).

En la figura 11 se ilustra el diagrama de casos de usos para la aplicación Web. Se debe mencionar que, para cada etapa del modelado, UWE provee diferentes estereotipos.

**Figura 11:** Modelo de Casos de Uso



Fuente: (Pech, Guerrero, & Menéndez, 2014).

El caso de uso "RealizarBusqueda" es del estereotipo explorar («browsing»). Modela la búsqueda de los objetos de aprendizaje por medio de las características de los objetos y de los usuarios para que el sistema pueda proporcionar una recomendación personalizada. (Pech, Guerrero, & Menéndez, 2014).

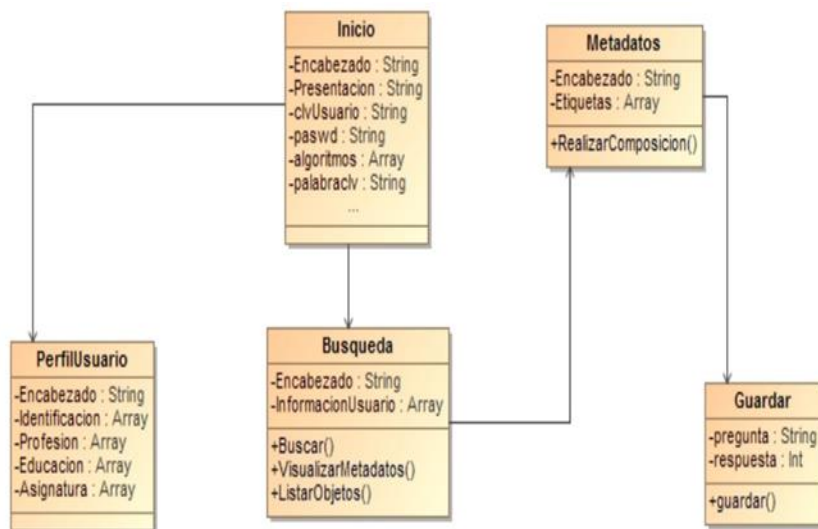
### 2.12.2.2 Modelo de Contenido

El objetivo del modelo de contenido es proporcionar una especificación visual de la información en el dominio relevante para la aplicación Web. Este es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias

para el caso de estudio presentado. El modelo de contenido define, mediante un diagrama de clases, los conceptos detalle involucrado en la aplicación.

En la figura 12 se presenta el diagrama de clases para el modelo de contenido. En particular, la información de los usuarios es modelada por la clase "PerfilUsuario" donde se almacenan las propiedades que describen a los diferentes tipos de usuarios.

**Figura 12: Modelo de Contenido**



Fuente: (Pech, Guerrero, & Menéndez, 2014)

En la clase "Inicio" se modela el inicio de la aplicación Web, se almacenan las credenciales y propiedades que sirven para identificar al usuario que quiere iniciar sesión. La clase "Búsqueda" modela la información que el usuario proporciona para realizar una consulta y los métodos que se ejecutan para generar la lista de recomendación, la selección de los objetos y la recuperación de los mismos con sus metadatos. (Pech, Guerrero, & Menéndez, 2014).









### 2.12.2.3 Modelo de Navegación

El modelo de navegación representa a los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas. Es decir que en el

modelo de navegación se incluyen todas las clases como clases de navegación y se debe aumentar las clases de procesos que representan los métodos.

UWE provee estereotipos para el modelado de navegación, en la Figura 13 se presentan los usados en este caso de estudio y seguidamente se da una descripción de cada uno de ellos.

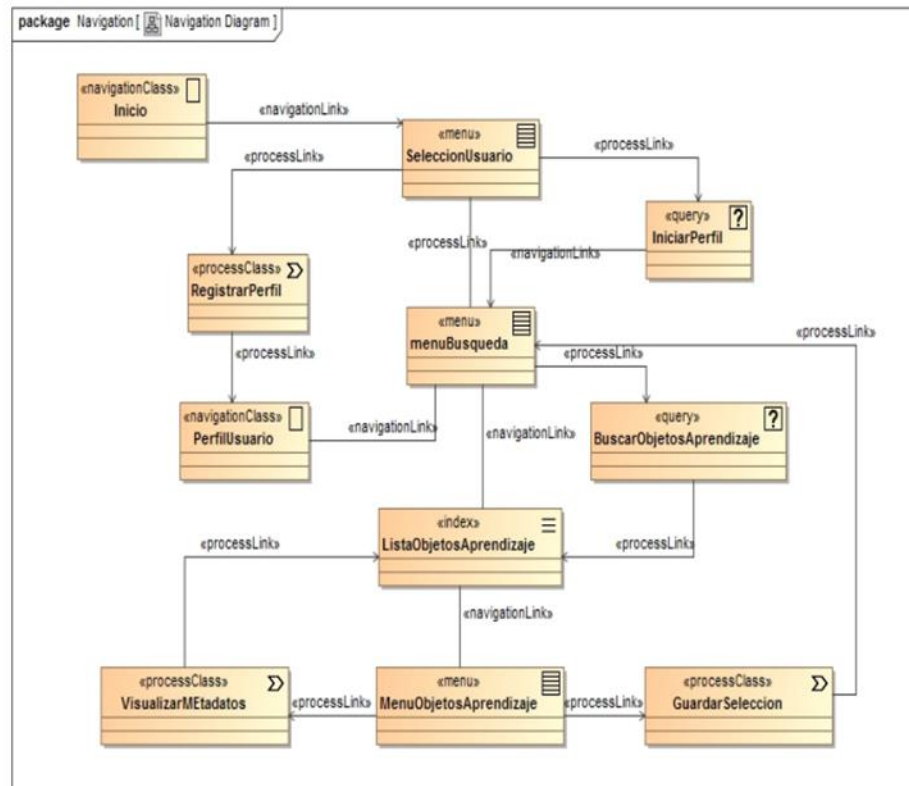
Figura 13: Descripción de Estereotipos

	navigationClass		guidedTour
	navigationLink		query
	menu		processClass
	index		processLink

Fuente: (Pech, Guerrero, & Menéndez, 2014)



Figura 14: Modelo de Navegación



Fuente: (Pech, Guerrero, & Menéndez, 2014).

La navegación por diferentes alternativas es representada por las clases «menu» ("SeleccionUsuario, MenuBusqueda y MenuObjetosAprendizaje") que se añaden a cada clase de navegación que tiene más de una asociación saliente. (Pech, Guerrero, & Menéndez, 2014).

#### 2.12.2.4 Modelo de Presentación

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación Web.

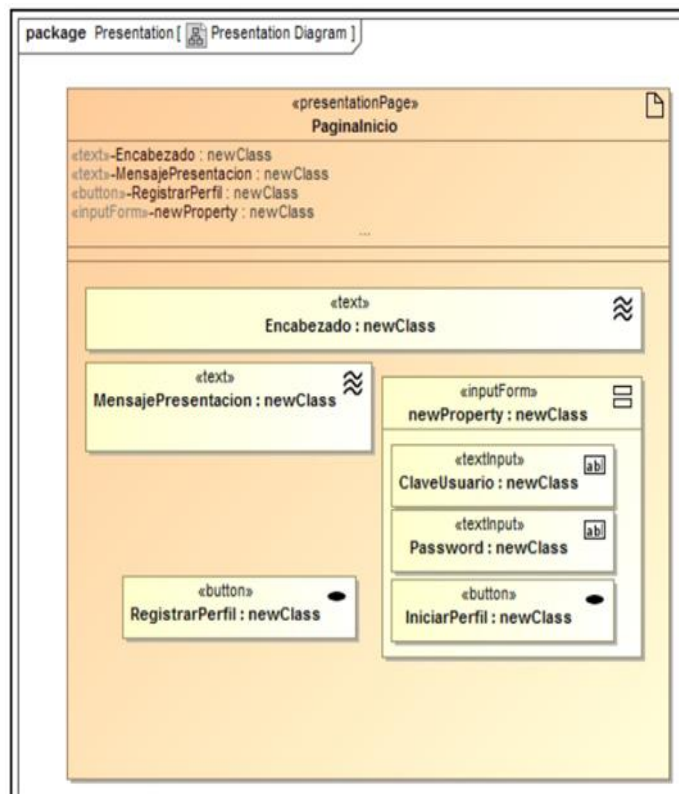
Se basa en el modelo de navegación y en los aspectos concretos de la IU. Describe la estructura básica de la IU. Su ventaja es que es independiente de las técnicas actuales que se utilizan para implementar un sitio Web, lo que permite a las partes

interesadas discutir la conveniencia de la presentación antes de que realmente se aplique.

Una clase de presentación está compuesta de elementos de IU como texto («text»), enlaces («anchor»), botones («button»), imágenes («image»), formularios («form») y colecciones de enlaces («anchored collection»).

En la siguiente figura se muestra un ejemplo de la clase de presentación para la clase de Navegación Inicio.

**Figura 15:** Modelo de Presentación

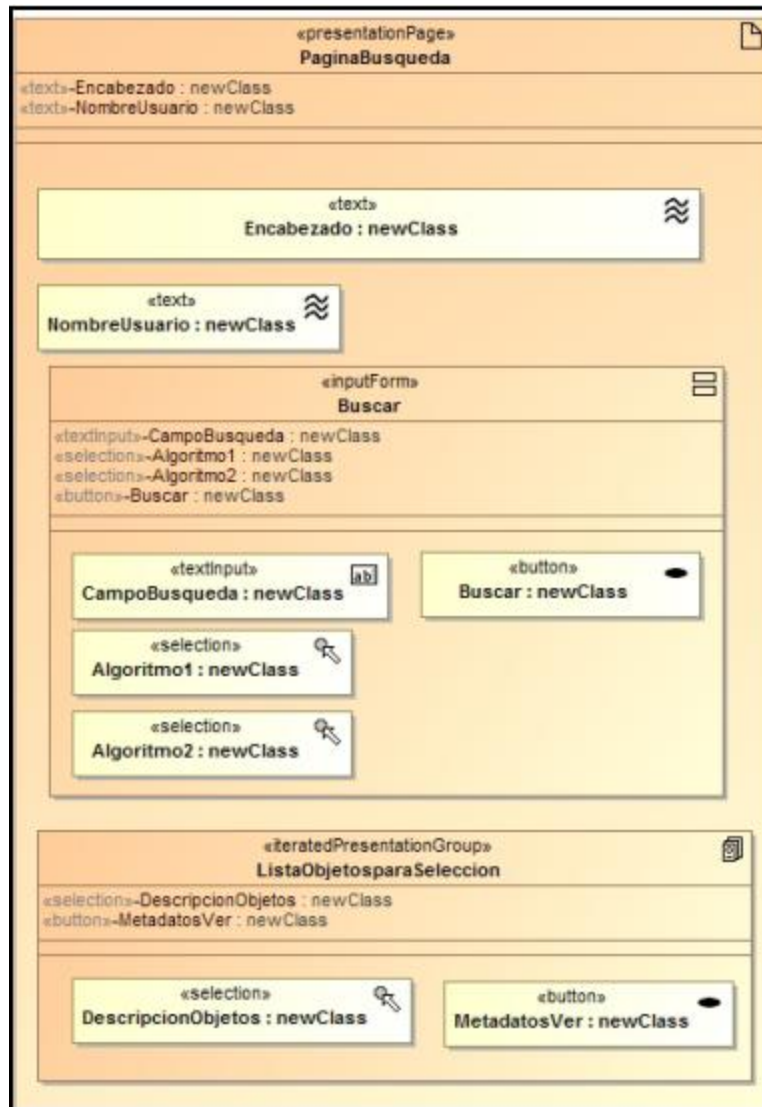


Fuente: (Pech, Guerrero, & Menéndez, 2014).

En la Figura 17 se modela la página de presentación "paginaBusqueda" donde se representa como texto un encabezado y el nombre del usuario. Existe un formulario donde se puede introducir las palabras clave de búsqueda, así como seleccionar los algoritmos que se pueden aplicar. Esta página de presentación contiene un

grupo de presentación para modelar las listas de objetos candidatos a la composición y los botones de buscar y ver metadatos.

**Figura 16:** Pagina de Presentación



Fuente: (Pech, Guerrero, & Menéndez, 2014).

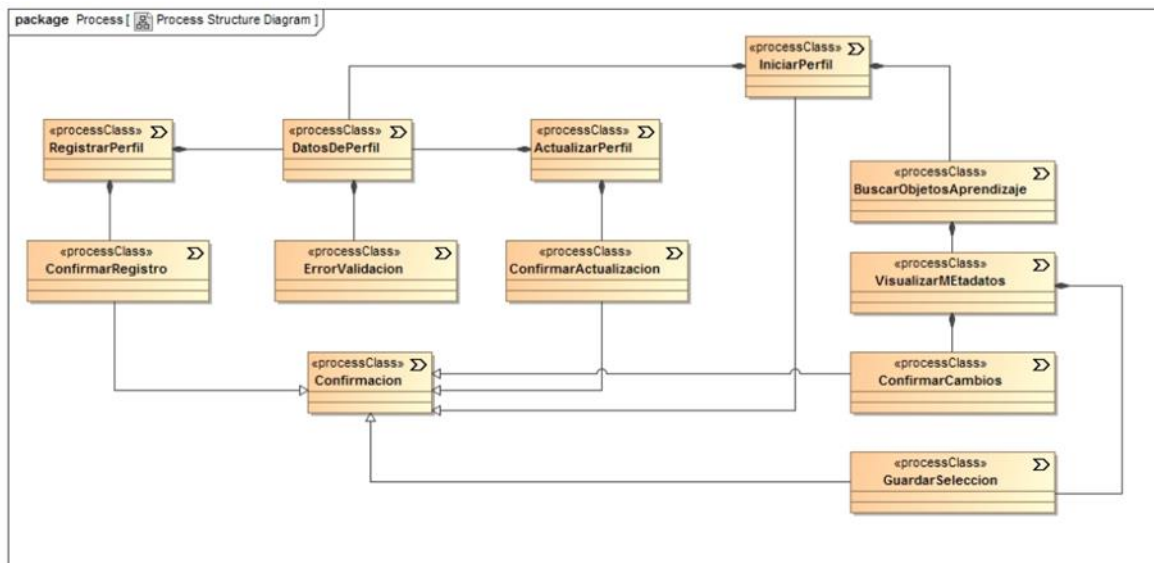
### 2.12.2.5 Modelo de Proceso

La estructura de navegación puede ser extendida mediante clases de procesos que representan la entrada y la salida de procesos de negocio. El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso. En este modelo se tienen dos tipos de modelos:

**Modelo de estructura del proceso:** Es representado por un diagrama de clases donde se describen las relaciones entre las diferentes clases de proceso. La Figura 17 presenta la aplicación del modelo para el caso de estudio analizado.

**Modelo del flujo del proceso:** Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Ellos pueden ser enriquecidos con flujos de objetos que muestran objetos relevantes para la entrada o salida de esas actividades.

Figura 17: Modelo de Proceso

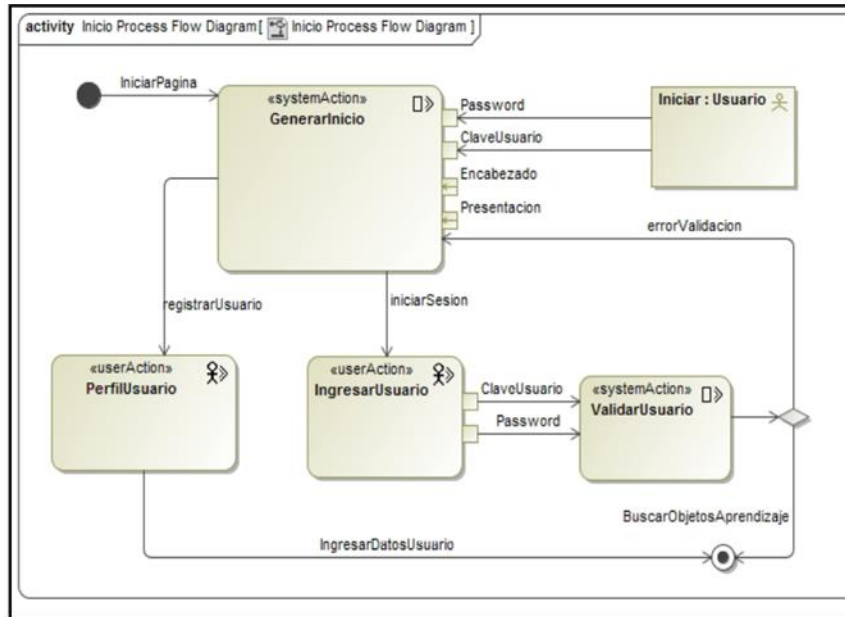


Fuente: (Pech, Guerrero, & Menéndez, 2014)

Estos diagramas representan el flujo del proceso, describiendo el comportamiento de una clase de proceso. En la Figura 17 se muestra el diagrama de actividad para el proceso "Inicio". El diagrama muestra que al generar la página de inicio el usuario puede optar por dos opciones:

- Proporcionar su clave de usuario y contraseña si es un usuario registrado, Activar el botón para registrarse como nuevo usuario.

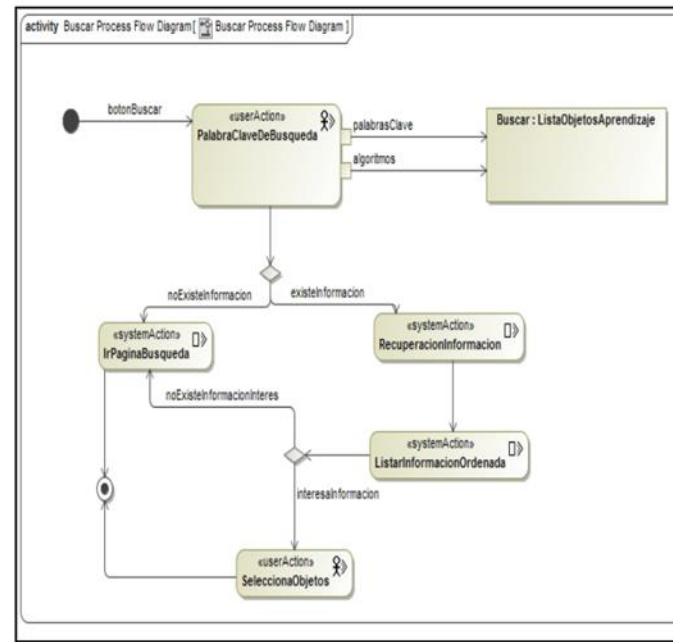
**Figura 18:** Flujo de Proceso de Inicio



Fuente: (Pech, Guerrero, & Menéndez, 2014).

En la Figura 19 se ilustra el diagrama de actividad para el proceso "Buscar". El diagrama muestra que se activa con el botón buscar y el usuario proporciona las palabras clave para iniciar la búsqueda. La aplicación regresa una lista de objetos de aprendizaje candidatos a ser seleccionados por el usuario. Si existe la información se recupera la misma desde el repositorio, en caso contrario se regresa a la página de búsqueda. Si la información listada es de interés para el usuario, este selecciona la misma, en caso contrario cambia sus parámetros de búsqueda. Adicionalmente a estos modelos es requerido conformar la documentación requerida para la descripción de los modelos, así como los diccionarios de datos necesarios para clarificar el conocimiento representado. (Pech, Guerrero, & Menéndez, 2014).

**Figura 19:** Flujo de Proceso Buscar



Fuente: (Pech, Guerrero, & Menéndez, 2014).

El desarrollo de aplicaciones requiere de metodologías acordes a las características de la plataforma donde estas sean ejecutadas. La Ingeniería Web propone nuevas metodologías orientadas al desarrollo y modelación de los procesos asociados a aplicaciones que se ejecuten en la W.W.W.

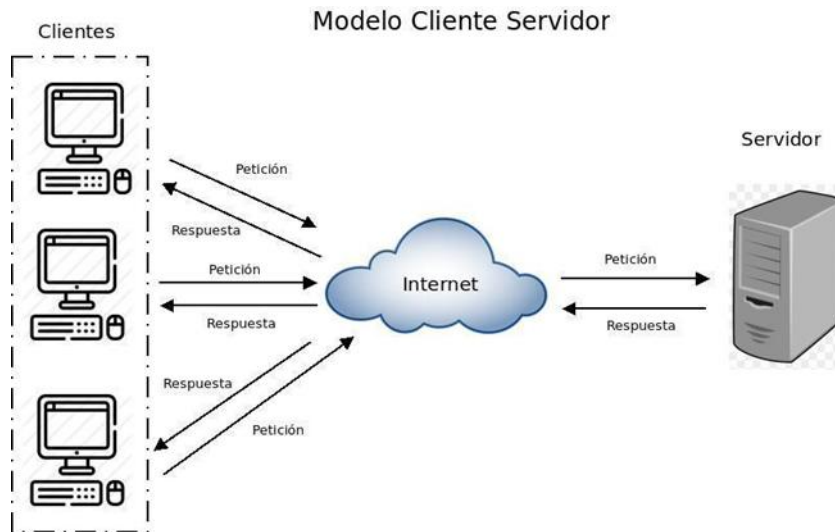
### 2.13 ARQUITECTURA CLIENTE/SERVIDOR

La tecnología Cliente/Servidor, es un modelo que implica productos y servicios enmarcados en el uso de la tecnología de punta y que permite la distribución de la información en forma ágil y eficaz a las diversas áreas de una organización (empresa o institución pública o privada), así como también fuera de ella.

Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

**Figura 20:** Modelo Cliente/Servidor



Fuente: (Schiaffarino, 2019)

### 2.13.1 Tipos de Arquitectura Cliente/Servidor

Dentro de la arquitectura cliente servidor existen tres tipos que son los que se detalla a continuación.

#### 2.13.1.1 Arquitectura de dos Capas

Esta se utiliza para describir los sistemas cliente servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos. Eso significa que el servidor no requiere de una aplicación extra para proporcionar parte del servicio.

#### 2.13.1.2 Arquitectura de tres Capas

En la arquitectura de tres capas existe un nivel intermediario, eso significa que la arquitectura generalmente está compartida por un cliente que como hablamos más

arriba es el que solicita los recursos equipado con una interfaz de usuario o mediante un navegador Web.

La capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados pero que requiere de otro servidor para hacerlo. La última capa es el servidor de datos que proporciona al servidor de aplicaciones los datos necesarios para poder procesar y generar el servicio que solicitó el cliente en un principio.

### **2.13.1.3 Arquitectura de N Capas**

En la arquitectura de tres capas, los servidores dos y tres realizaron una tarea específica por lo tanto un servidor Web puede usar los servicios de otros servidores para poder proporcionar su propio servicio.

Por consiguiente, la arquitectura en tres niveles es potencialmente una arquitectura en N capas ya que, así como está contemplado en tres niveles como el caso anterior puede estar compuesto por N servidores donde cada uno de ellos brindan su servicio específico.

### **2.13.2 Seguridad del Modelo Cliente Servidor**

Es de gran importancia por el valor intrínseco para los usuarios. Tiene tres componentes:

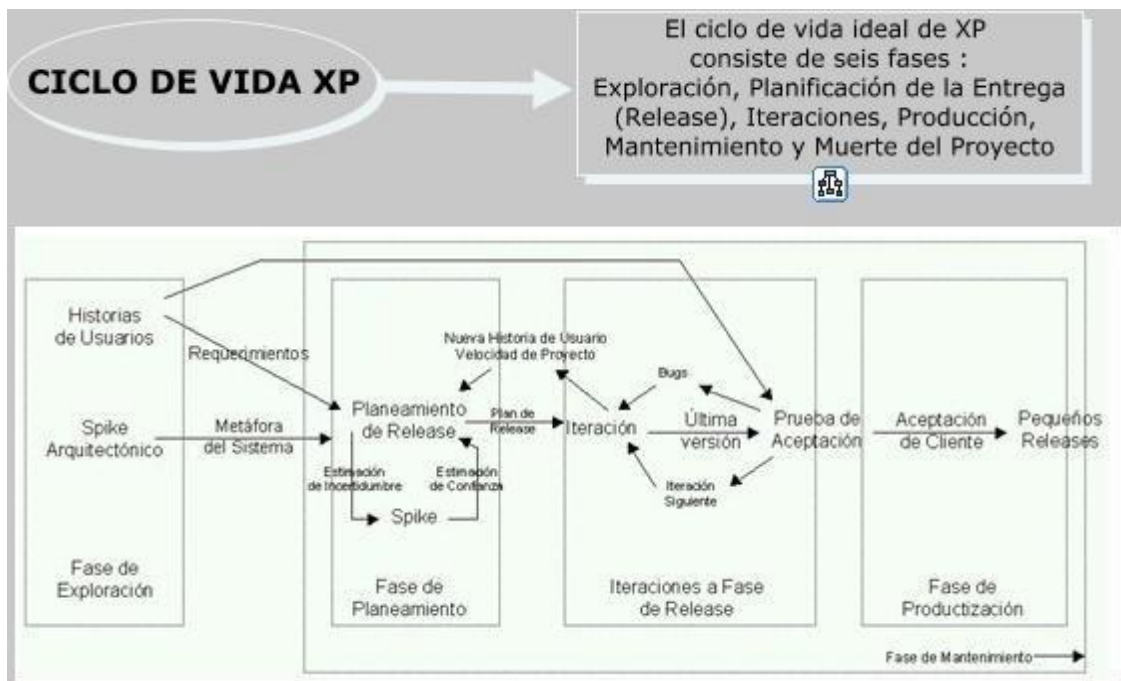
- ✓ **Confidencialidad:** Protección contra individuos no autorizados.
- ✓ **Integridad:** Protección contra la alteración o corrupción.
- ✓ **Disponibilidad:** Protección contra la interferencia con los procedimientos de acceso a los recursos.



## 2.14 CICLO DE VIDA DEL SOFTWARE

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla con los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados. (Villagómez, 2017).

Figura 21: Ciclo de vida XP



Fuente: (Ferrer & Robles, 2002)

El ciclo de vida de XP es de carácter interactivo e incremental del desarrollo. Las iteraciones son relativamente cortas ya que se piensa que entre más rápido se le entreguen desarrollados al cliente, más retroalimentación se va a obtener y esto se va a representar una mejor calidad del producto a largo plazo.

## **2.15 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE**

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software. La ingeniería de software aplica diferentes normas y métodos que permiten obtener mejores resultados, en cuanto al desarrollo y uso del software mediante la aplicación correcta de estos procedimientos se pueden llegar a cumplir de manera satisfactoria los proyectos que ayudan a la población. (Pressman, 2010).

### **2.15.1 PHP**

Es un lenguaje de programación de código abierto ideal para el desarrollo de aplicaciones web, Page Hypertext Pre-processor, más conocido por el acrónimo PHP, es muy extendido especialmente en el entorno del desarrollo web. La primera versión de PHP fue creada por Rasmus Lerdorf en 1995, y fue uno de los primeros lenguajes de programación del lado del servidor, es decir: al ser ejecutado en el servidor web directamente tiene acceso a bases de datos, conexiones en red y otras tareas antes de que se envíe la página final a la cual el cliente tendrá acceso. Además, PHP tenía la peculiaridad de poder incorporar directamente un documento HTML sin la necesidad de tener que llamar a un archivo externo que procese los datos. (Arias, 2017).

En el presente proyecto se utilizará php7.4

### **2.15.2 PostgreSQL**

Es PostgreSQL es un potente sistema gestor de bases de datos relacionales, está orientado a objetos, es multiplataforma y Open Source, es orientado a objetos, es decir, todos los elementos de nuestra base de datos van a poder tratarse como objetos, utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más

complicadas puedes usarlo en cualquier sistema operativo y bajo licencia libre. (Grupo Global, 2020)

### **2.15.3 JavaScript**

JavaScript es un lenguaje de programación 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.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

### **2.15.4 Framework Codeigniter**

Es un Framework para el Desarrollo de Aplicaciones orientado para crear webs usando PHP. Su meta es permitir desarrollar proyectos de manera más ágil y rápido que hacerlo escribiendo el código desde cero, proporcionando una gran variedad de librerías para las tareas más corrientes, así como una interfaz simple y una estructura lógica para acceder a estas librerías. Codeigniter es un framework para el desarrollo de aplicaciones, que utiliza el MVC. Esto permite a los programadores o desarrolladores web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs. (Vergara, 2016)

#### **2.15.4.1 Bootstrap**

Es un Framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla de un dispositivo utilizado por el usuario. El framework combina CSS y JavaScript para estilizar los

elementos de una página HTML. Permite mucho más que, cambiar el color de los botones y los enlaces. (Guajardo, 2020).

### **2.15.5 XAMPP**

Es un servidor independiente de plataforma, software libre, el objetivo de XAMPP es crear una distribución fácil de instalar para desarrolladores que se están iniciando en el mundo de Apache. XAMPP viene configurado por defecto con todas las opciones activadas. Es posible realizar todas las tareas de administración necesarias en cualquier base de datos: crear, eliminar, importar, exportar y administrar las tablas: añadir, eliminar o modificar campos, ejecutar secuencias de comandos SQL, entre otros XAMPP es gratuito tanto para usos comerciales como no comercial.

XAMPP es una distribución de Apache que incluye varios softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen: el servidor web Apache, los sistemas relacionales de administración de bases de datos MySQL y MariaDB, así como los lenguajes de programación Perl y PHP. La inicial X se usa para representar a los sistemas operativos Linux, Windows y Mac OS X. (Martin, 2020)

### **2.15.6 Arquitectura de Software MVC**

El modelo de vista controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Modelos y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC.

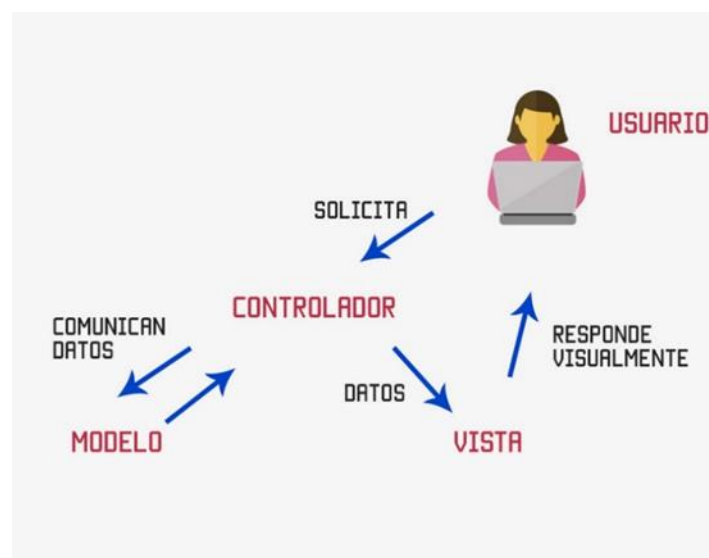
El MVC es muy usado en el desarrollo Web porque a diferencia de la creación de programas locales donde se usa por lo general un mismo lenguaje para desarrollar toda un aplicación en el desarrollo Web se requiere por lo menos usar dos lenguajes

(HTML y php) para crear una página simple, y todo esto en una sola página aunado a las consultas en la base de datos que son imprescindibles pues crean hasta páginas de 300 líneas que después para depurar, mejorar o saber dónde nos estamos equivocando es un poco difícil. Se realizará un breve ejemplo de una petición de usuario:

- El usuario realiza una petición. En nuestro caso desea modificar el nombre de una tabla.
- El controlador recibe dicha petición (de la interfaz gráfica) y procede a ejecutar la acción pertinente enviando los datos al Modelo.
- El Modelo realiza las modificaciones con la base de datos (mysql, postgresql, sqlite).
- Luego, devuelve al controlador los cambios solicitados.
- Cuando el Controlador recibe todos los datos del cambio, envía una respuesta a la Vista.
- La Vista aplica la modificación en la interfaz del navegador mostrando al usuario.

A continuación, una explicación un poco más detallada del modelo, vista y controlador:

**Figura 22:** Modelo Vista Controlador



Fuente: (Hernandez, 2015).

**Modelo:** El modelo se encarga de gestionar la información que es enviada por el controlador. Puede hacer peticiones de consulta, actualización o búsqueda en la base de datos.

Otra función importante del modelo es implementar los privilegios de acceso, como por ejemplo permisos de usuario, asignación de roles.

**Controlador:** Es el encargado de controlar y relacionar el modelo con la vista. Genera respuestas a través de una solicitud, en general dada por el usuario.

Cumple la función de solicitar datos al modelo. Dicha solicitud puede ser de modificar un registro en la base de datos. Estos cambios luego son comunicados a la vista.

**Vistas:** Las vistas se encargan de mostrar los datos a través de una interfaz. Aquí es donde el usuario puede interactuar con nuestra aplicación.

Tanto el controlador como los modelos procesan los datos para luego generar dicha respuesta al usuario, como por ejemplo en una página web.

## **2.16 PRUEBAS DE SOFTWARE**

### **2.16.1 Prueba de Caja Blanca**

Según (Pressman, 2010), las pruebas de caja blanca, denominada a veces como prueba de caja de cristal es un método de diseño de caso de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Al usar los métodos de prueba de caja blanca puede derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras operativas.

- Revisen estructuras de datos internas para garantizar su validez.

### **2.16.2 Prueba de Caja Negra**

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software; es decir las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisaran por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra son una alternativa para las técnicas de caja blanca. En vez de ello es un enfoque complementario es probable que descubra una clase de errores diferentes de los métodos de caja blanca complementarias que muestran errores distintos que el método de la caja negra. (Pressman, 2010).

Según (Pressman, 2010), estas pruebas intentan encontrar errores en las siguientes categorías:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externos.
- Errores de comportamiento o rendimiento
- Errores de inicialización y terminación

## **2.17 MÉTRICAS DE CALIDAD**

Éstas proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir, cómo se va a medir para que el sistema se adapte a los requisitos que pide el cliente. (Pressman, 2010).

### **2.17.1 ISO/IEC 9126**

Es un estándar internacional para la evaluación de la calidad del software. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

El modelo de calidad del producto definido por la ISO 9126 se encuentra compuesto por las cinco características de calidad que se muestran en la siguiente figura:

**Figura 23:** Características del Modelo de Calidad del Producto



**Fuente:** (ISO 9126, 2016)

### a) Funcionabilidad

Funcionabilidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. A continuación, se detallará las sub características:

- **Adecuación:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- **Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- **Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.
- **Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan



acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

- **Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

## b) Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. A continuación, se detallará las sub características:

- **Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores
- **Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- **Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- **Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

## c) Usabilidad

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. Esta característica se subdivide a su vez en las siguientes sub características:

- **Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- **Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- **Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.
- **Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- **Conformidad de uso:** La capacidad del software de cumplir a los estándares o normas relacionadas a la usabilidad.

#### d) Mantenibilidad

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales. A continuación, se detallará las sub características:

- **Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- **Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.
- **Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- **Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

- **Conformidad de mantenimiento:** La capacidad del software de cumplir a los estándares de facilidad de mantenimiento.

#### e) Portabilidad

La capacidad que tiene el software para ser trasladado de un entorno a otro. A continuación, se detallará las sub características:

- **Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).
- **Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- **Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.
- **Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- **Conformidad de portabilidad:** La capacidad del software de cumplir a los estándares relacionados con la portabilidad.

### 2.18 MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE

La estimación de costo y esfuerzo del software nunca será una ciencia exacta. Demasiadas variables (humanas, técnicas, ambientales, políticas) pueden afectar el costo final del software y el esfuerzo aplicado para su desarrollo. Sin embargo, la estimación del proyecto de software puede transformarse de un arte oscuro a una

serie de pasos sistemáticos que proporcionen estimaciones con riesgo aceptable. (Pressman, 2010). Para lograr estimaciones confiables de costo y esfuerzo, surgen algunas opciones:

- Retrase la estimación hasta ir avanzado el proyecto (obviamente se puede lograr estimaciones 100 por ciento precisas después de que el proyecto esté completo).
- Base las estimaciones en proyectos similares que ya estén completos.
- Use técnicas de descomposición relativamente simples para generar estimaciones de costo y esfuerzo de proyecto.
- Use uno o más modelos empíricos para estimación de costo y esfuerzo de software.

Desafortunadamente, la primera opción, aunque atractiva, no es práctica. Las estimaciones de costo deben proporcionarse por anticipado. No obstante, debe reconocer que mientras más espere, más conocerá, y mientras más conozca, menos probabilidades tendrá de cometer errores serios en sus estimaciones. (Pressman, 2010).

La segunda opción puede funcionar razonablemente bien si el proyecto actual es muy similar a esfuerzos anteriores y otros factores que influyen en el proyecto (por ejemplo, el cliente, condiciones del negocio, entorno de ingeniería de software, fechas límite) son aproximadamente equivalentes. Desafortunadamente, la experiencia pasada no siempre es buen indicador de resultados futuros. (Pressman, 2010).

Las opciones restantes son enfoques viables para la estimación del proyecto de software. De manera ideal, las técnicas anotadas para cada opción deben aplicarse en cascada y cada una es una comprobación cruzada para las demás. Las técnicas de descomposición tienen un enfoque de “divide y vencerás” para la estimación del proyecto. Al descomponer un proyecto en funciones principales y actividades de

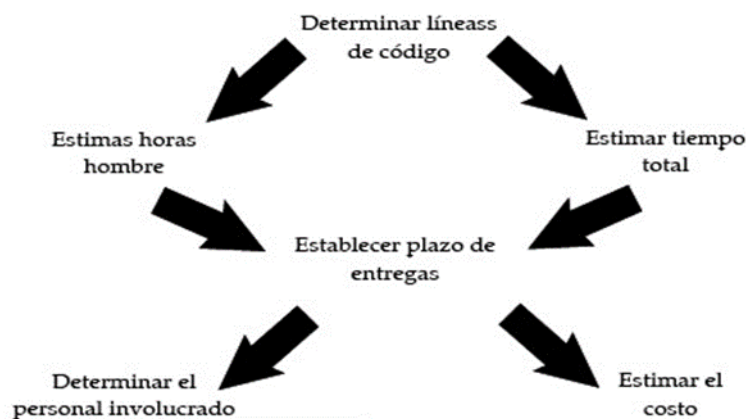
ingeniería de software relacionadas, la estimación de costo y esfuerzo puede realizarse en forma escalonada. Los modelos de estimación empírica pueden usarse para complementar las técnicas de descomposición y ofrecer un enfoque de estimación potencialmente valioso por derecho propio. (Pressman, 2010).

Las herramientas de estimación automatizadas implementan una o más técnicas de descomposición o modelos empíricos y proporcionan una atractiva opción para estimar. (Pressman, 2010).

## 2.19 MODELO COCOMO II

Modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por (Modelo Constructivo de Coste). El modelo Cocomo original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria. Desarrollado en 1981 por el Dr. Barry Bohem, para dar una estimación del número de meses hombre que tomará para desarrollar un producto software.

**Figura 24:** Modelo de Estimación de Costos COCOMO II

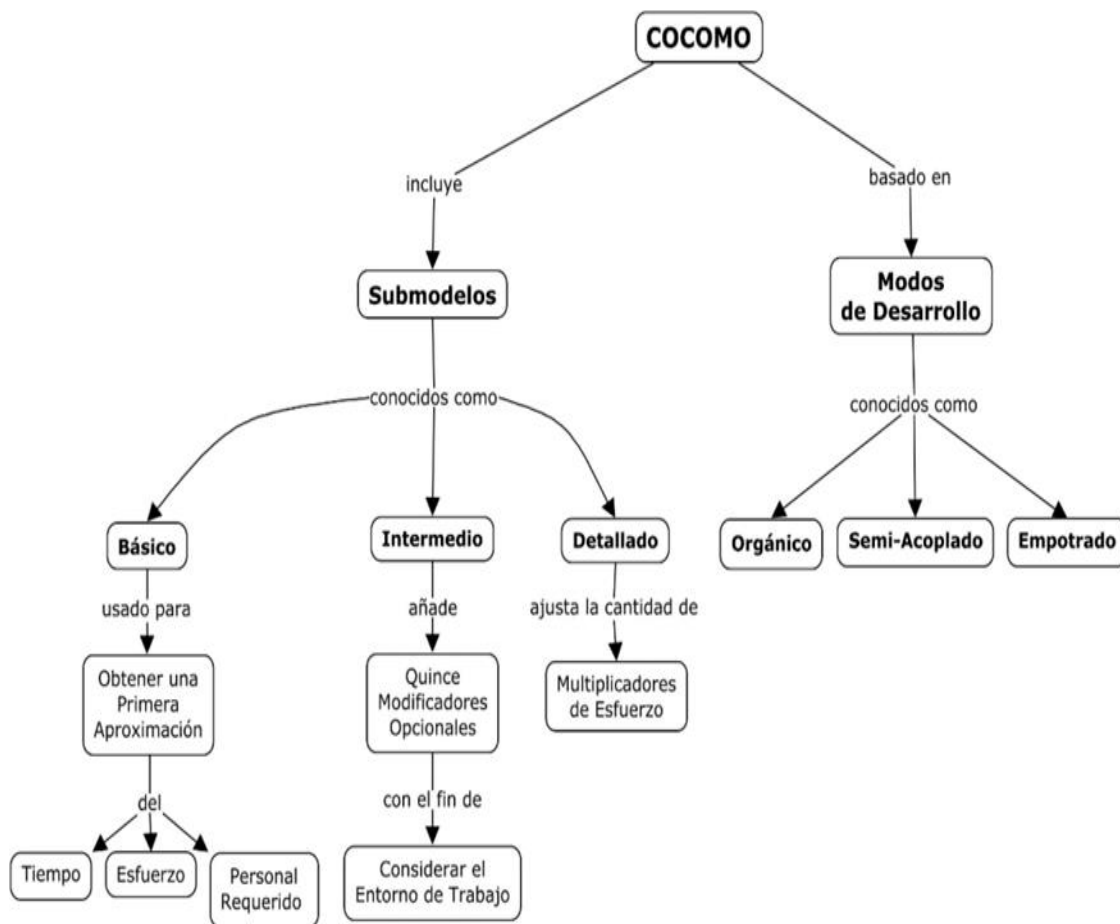


Fuente: (Aparicio Gil, 2012)

En la siguiente figura se muestran los principales conceptos relacionados con el modelo COCOMO. Este modelo, propuesto por Barry W. Bohem, fue introducido a finales de los años 70 y comienzos de los 80 del siglo pasado en su trabajo, Software

Engineering Economics. Entre otras características, el modelo COCOMO está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el “tamaño” del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo. (Garita Gonzalez & Lizano Madriz, 2018).

**Figura 25:** Conceptualización Básica de COCOMO



Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

Los submodelos son tres: básico, intermedio y detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado. (Garita Gonzalez & Lizano Madriz, 2018).

**Figura 26:** Esquema de Modos de Desarrollo

<i>Modo de desarrollo</i>	<i>Requisitos</i>	<i>Tamaño</i>	<i>Complejidad</i>	<i>Personas</i>	<i>Experiencia</i>
Orgánico	Poco rígidos	Pequeño (<50KLDC)	Pequeña	Pocas	Mucha
Semiacoplado	Poco/medio	Medio (50 a 300KLDC)	Medio	Medio	Medio
Empotrado	Alto	Grande (>300KLDC)	Alta	Alta	Poca

Fuente: (Garita Gonzalez & Lizano Madriz, 2018).

En la figura 26 se muestra el esquema de modos de desarrollo de software con sus principales características que ayudan a elegir el tipo de modo de desarrollo para un proyecto en particular. En esta tabla, en el caso del tamaño, se consideran las líneas de código fuente del software en unidades de miles de líneas 3 de código (KLDC, por sus siglas en inglés). (Garita Gonzalez & Lizano Madriz, 2018).

Estos modos de desarrollo permiten utilizar cuatro valores constantes. En la siguiente tabla se muestran los modos de desarrollo y los valores constantes respectivos. Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo. (Garita Gonzalez & Lizano Madriz, 2018).

**Figura 27: Valores Constantes por Modo Desarrollo**

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

Las ecuaciones incluidas en este artículo, son las utilizadas para los submodelos básico e intermedio. Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P). 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. (Garita Gonzalez & Lizano Madriz, 2018).

**Figura 28:.** Ecuaciones por tipo de modelo COCOMO: Básico e Intermedio

Ecuación	Submodelo básico	Submodelo intermedio
Esfuerzo (E)	$( E ) = a * (KLDC)b$	$( E ) = a * (KLDC)b * ME$
Tiempo (T)	$( T ) = c * (E)d$	$( T ) = c * (E)d$
Personal (P)	$( P ) = E/T$	$( P ) = E/T$

Fuente: (Garita Gonzalez & Lizano Madriz, 2018).

En la Figura 28, se muestran las ecuaciones para esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P) así como los multiplicadores de esfuerzo (ME), utilizados solo en la ecuación de esfuerzo del submodelo intermedio. (Garita Gonzalez & Lizano Madriz, 2018).

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. (Garita Gonzalez & Lizano Madriz, 2018).



**Figura 29: Ecuaciones por Tipo de Modelo COCOMO: Básico e Intermedio**

Multiplicadores de esfuerzo (ME)			Valoración					
			Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
<b>Atributos del producto</b>								
1.	RELY	Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	
2.	DATA	Tamaño de la base de datos		0,94	1.00	1,08	1,16	
3.	CPLX	Complejidad del producto	0,70	0,85	1.00	1,15	1,30	1,65
<b>Atributos de la computadora</b>								
4.	TIME	Restricciones del tiempo de ejecución			1.00	1,11	1,30	1,66
5.	STOR	Restricciones del almacenamiento princ.			1.00	1,06	1,21	1,56
6.	VIRT	Inestabilidad de la máquina virtual		0,87	1.00	1,15	1,30	
7.	TURN	Tiempo de respuesta del computador		0,87	1.00	1,07	1,15	
<b>Atributos del personal</b>								
8.	ACAP	Capacidad del analista	1,46	1,19	1.00	0,86	0,71	
9.	AEXP	Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	
10.	PCAP	Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70	
11.	VEXP	Experiencia en S.O. utilizado	1,21	1,10	1.00	0,90		
12.	LEXP	Experiencia en el lenguaje de progr.	1,14	1,07	1.00	0,95		
<b>Atributos del proyecto</b>								
13.	MODP	Uso de prácticas de programación modernas	1,24	1,10	1.00	0,91	0,82	
14.	TOOL	Uso de herramientas software	1,24	1,10	1.00	0,91	0,83	
15.	SCED	Restricciones en la duración del proy.	1,23	1,08	1.00	1,04	1,10	

Fuente: (Garita Gonzalez & Lizano Madriz, 2018).

En la figura 29, se muestran los multiplicadores de esfuerzo. 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. Dos ejemplos nos ayudan a entender el objeto de los multiplicadores de esfuerzo. Primero, en caso de que no se desee ajustar el esfuerzo nominal, se debe utilizar el valor nominal (es decir, el valor “1”) para cualquier multiplicador de esfuerzo. Segundo; si, por el contrario, se desea hacer algún ajuste en particular al esfuerzo nominal por razones de, en este caso, poca experiencia en la aplicación, se podría utilizar el multiplicador AEXP en valor nominal “muy bajo” (1,29). (Garita Gonzalez & Lizano Madriz, 2018).

## **2.20 SEGURIDAD DEL SISTEMA**

Actualmente 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. En ese orden, existe una triada de seguridad de la información que toda empresa, negocio, organización y persona particular debe conocer y respaldar.

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo. (Pressman, 2010).

- **Seguridad a nivel base de datos:** Es la capacidad del sistema para proteger Datos, Servicios y Recursos de usuarios no autorizados. El fin de la seguridad es garantizar la protección o estar libre de todo peligro y/o daño, y que en cierta manera es infalible
  
- Confidencialidad, integridad y disponibilidad.

- **Seguridad a nivel de software:** La seguridad del software aplica los principios de la seguridad de la información al desarrollo de software. Esto se refiere a la seguridad de información contra el acceso desautorizado y la modificación de información, se está en una fase de procesamiento, almacenamiento o tránsito.
- **Seguridad en el desarrollo de software:** Sin importar el tipo de software que se construya, siempre se debe considerar la seguridad como una prioridad para continuar con el desarrollo del sistema y así evitar factores externos que se puedan dañar el trabajo.
- **Seguridad a nivel aplicación:**
  - ✓ Control de acceso
  - ✓ Validación de datos de entrada
  - ✓ Pruebas de código
- **Copias de seguridad**
  - ✓ Backup de la base de datos

A continuación, se muestra las vulnerabilidades del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor.

- ✓ Vulnerabilidad en los navegadores
- ✓ Vulnerabilidad en los correos electrónicos
- ✓ Acceso no autorizado a cookies
- ✓ Simulación en la comunicación con el servidor

Para dar solución a estas vulnerabilidades podemos aplicar los siguientes mecanismos para emitir confianza y seguridad en la aplicación Web.

- **Cortafuegos:** Mecanismo de filtrado combinación de hardware y software para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas examinando cada paquete de información entrante para garantizar que no sea un dato sospechoso.
- **Uso de Autenticación:** Esta debe ser difícil de averiguar, la contraseña debe tener la suficiente complejidad como para que un atacante no pueda deducirla por medio de programas una contraseña o clave es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso.

## **CAPÍTULO III**

### **MARCO APLICATIVO**

### **3 MARCO APLICATIVO**

#### **3.1 INTRODUCCIÓN**

En el presente capítulo se hará uso de las metodologías y herramientas descritas anteriormente, las mismas nos servirán para el desarrollo del sistema y todos sus módulos.

En el desarrollo del sistema se observó que la comunicación con el equipo de trabajo era muy importante, desde el cliente que forma parte del equipo haciendo conocer la forma del trabajo actual, y captar las necesidades de los mismos. Por estas razones se hará el uso de la metodología XP, donde claramente pone en comunicación directa y continua a clientes y desarrolladores, este aspecto fue aplicado ya que se mantuvieron reuniones frecuentes con el personal del cliente, quienes se integraron al proyecto para establecer prioridades y resolver dudas. Dicha metodología fue elegida porque responde de manera favorable a un ambiente de requerimientos dinámicamente cambiantes.

El sistema contará con una metodología de desarrollo de software, como se mencionó, en el Capítulo II, se usa la metodología ágil XP, de manera que se siga los pasos y así logre la producción de un software de calidad, que ayude a definir las etapas del proyecto, los pasos que deben seguirse, los tiempos y las tareas que se realizaran en cada etapa. Conjuntamente con la herramienta del Lenguaje de Modelado UWE.

En la parte de planificación se agregaron algunos pasos de análisis de la situación actual y se adicionará modelos de la notación UWE en la fase de diseño, como: modelo de contenido o conceptual, navegación y presentación.

## **3.2 ANÁLISIS DE REQUERIMIENTOS**

### **3.2.1 Análisis de Situación Actual**

Se tuvieron reuniones con los responsables de la Empresa TESICON, en las cuales permitió la observación de sus procesos cotidianos. El producto de estos encuentros dio lugar al siguiente análisis de la situación actual en la que se encuentra TESICON.

- ✓ El personal encargado de ventas registra en un cuaderno los productos vendidos.
- ✓ No existe un registro de precios de los productos que hay en almacén, lo que ocasiona constantes llamadas al gerente para consulta de precios.
- ✓ Al finalizar en día lo registrado en el cuaderno de ventas no concuerda con el efectivo que se encuentra en caja, debido al posible olvido de anotación de la encargada.
- ✓ En cada compra, la verificación de la existencia del producto es realizada de forma manual comparándolo con una lista en un documento físico.
- ✓ No existe datos de los proveedores para poder realizar una nueva adquisición.
- ✓ No se lleva un registro de los productos entrantes y salientes por lo tanto no existe un control del inventario existente, y muchas veces se llega al extravió de productos en almacén.

### **3.2.2 Ingeniería de Requerimientos**

El diseño y construcción de software de computadora es difícil, creativo y sencillamente divertido. En realidad, elaborar software es tan atractivo que muchos desarrolladores de software quieren ir directo a él antes de haber tenido el entendimiento claro de lo que se necesita. Argumentan que las cosas se aclararán a medida que lo elaboren, que los participantes en el proyecto podrán comprender sus necesidades sólo después de estudiar las primeras iteraciones del software, que las cosas cambian tan rápido que cualquier intento de entender los requerimientos en detalle es una pérdida de tiempo, que las utilidades salen de la

producción de un programa que funcione y que todo lo demás es secundario. Lo que hace que estos argumentos sean tan seductores es que tienen algunos elementos de verdad. Pero todos son erróneos y pueden llevar un proyecto de software al fracaso. (Pressman, 2010).

La ingeniería de requerimientos tiende un puente para el diseño y la construcción. Pero, ¿dónde se origina el puente? Podría argumentarse que principia en los pies de los participantes en el proyecto (por ejemplo, gerentes, clientes y usuarios), donde se definen las necesidades del negocio, se describen los escenarios de uso, se delinear las funciones y características y se identifican las restricciones del proyecto. Otros tal vez sugieran que empieza con una definición más amplia del sistema, donde el software no es más que un componente del dominio del sistema mayor. Pero sin importar el punto de arranque, el recorrido por el puente lo lleva a uno muy alto sobre el proyecto, lo que le permite examinar el contexto del trabajo de software que debe realizarse; las necesidades específicas que deben abordar el diseño y la construcción; las prioridades que guían el orden en el que se efectúa el trabajo, y la información, las funciones y los comportamientos que tendrán un profundo efecto en el diseño resultante. (Pressman, 2010).

### **3.2.3 Obtención de Requerimientos**

Se plantea una clasificación de requerimientos en dos tipos: Los funcionales y los no funcionales. Los requerimientos funcionales definen características sobre el funcionamiento del sistema, es decir, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Por otro lado, se tienen los requerimientos no funcionales, los cuales determinan características que de una u otra forma puedan limitar el sistema.



**Tabla 3:** Categoría de las funciones de los Requerimientos del Sistema

Categoría de la función	Significado
Evidente	Debe realizarse y el usuario debería saber que se ha realizado.
Oculto	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, por ejemplo, guardar información en un mecanismo persistente de almacenamiento.
Superflua	Opcionales; su inclusión no repercute de forma significativa en costo ni en otras funciones.

Fuente: (Larman, 1999)

Analizando toda la información recolectada de la entrevista realizada, se elaboró una lista de requerimientos las cuales se las agrupa por módulos, como se detalla en los siguientes módulos:

- **Requerimientos funcionales**

Empezaremos a desarrollar el sistema satisfaciendo las necesidades del cliente.

**Tabla 4:** Requerimientos Funcionales

Referencia	Función	Categoría
<b>R1-1</b> <b>Administrar usuarios</b>	Permite la asignación de uno o más roles a un usuario.	Evidente
<b>R1-2</b> <b>Administrar roles</b>	Encargado de limitar el acceso a los recursos del sistema.	Evidente

---

<b>R1-3 Administración de las ventas</b>	Las ventas deben adicionar, modificar, eliminar.	Evidente
<b>R1-4 Relacionar las ventas con los clientes</b>	Toda venta debe estar relacionada con el registro de los clientes, de manera que sea posible contabilizar las compras del cliente.	Evidente
<b>R1-5 Administración de clientes</b>	Crear, listar, buscar y eliminar registro de clientes.	Evidente
<b>R1-6 Administración de compras</b>	Hacer el registro correspondiente de cada compra, adicionar modificar, eliminar.	Evidente
<b>R1-7 Administración de proveedores:</b>	Adicionar, modificar, eliminar y listar los registros de los proveedores.	Evidente
<b>R1-8 Administración de productos</b>	Adicionar, modificar, eliminar listar y buscar todos los productos. Cada producto debe tener una imagen referencial. Además, cada producto debe estar asignada a una categoría.	Evidente
<b>R 1-9 Administrar categorías</b>	Clasificar los productos de las categorías, adicionar, modificar, eliminar y listar el registro de categoría.	Evidente

---

<b>R 1-10</b> <b>Administración de marca</b>	Clasificar los productos del tipo de marca, adicionar, modificar, eliminar y listar el registro de categoría.	Evidente
<b>R 1-11</b> <b>Reportes</b>	Debe generar el reporte de venta, generar el reporte de compra que se realiza en el día.	Evidente
<b>R 1-12</b> <b>Inicio y cierre del sistema</b>	El sistema debe dar inicio y cierre de sesión en el sistema.	Oculto

Fuente: Elaboración propia

- **Requerimientos no funcionales**

**Tabla 5:** Referencias No Funcionales

Referencia	Función	Categoría
<b>R2-1</b>	El sistema debe visualizarse y funcionar correctamente en cualquier navegador Mozilla, Chrome.	Evidente
<b>R2-2</b>	El sistema debe ser totalmente funcional en los equipos que posea el cliente, debe ser accesible de una interfaz gráfica agradable y de fácil manejo.	Evidente
<b>R2-3</b>	Mantenimiento periódico del sistema para asegurar el buen rendimiento del sistema.	Evidente

Fuente: Elaboración propia

### 3.3 FASE DE PLANIFICACIÓN METODOLOGÍA XP

Aplicaremos la metodología XP el objetivo es siempre la satisfacción del cliente. El proyecto tendrá un tiempo aproximado de tres semanas por iteración.

Se podrá mostrar la forma de trabajo actual de la empresa TESICON, mediante las historias de usuario e iteraciones para el presente proyecto.

#### 3.3.1 Historias de Usuario

Las historias de usuario tienen el mismo propósito que los casos de uso podemos mencionar que son especificaciones escritas por el cliente expresando las funciones que debería realizar el sistema, son breves y no tienen un lenguaje técnico.

A partir del conjunto de requerimientos, se construyó un conjunto de historias de usuario, las mismas que cuentan con prioridades, riesgos e iteraciones.

#### Historia de Usuario 1

La tabla 6 detalla la Historia de Usuario 1 – Administración de Usuarios

**Tabla 6:** Historia de Usuario1 – Administración de Usuarios

HISTORIA DE USUARIO	
<b>Número:</b> 1	<b>Usuario:</b> Gerente
<b>Nombre de la historia:</b> Administración de usuarios	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> Los usuarios u operadores del sistema deben tener acceso solo a determinadas áreas según su cargo en la empresa TESICON, se determina la gestión de usuarios con roles y contraseñas.	
<b>Observaciones:</b> como administrador se designara al Gerente de la empresa.	

Fuente: Elaboración Propia

## Historia de Usuario 2

La tabla 7 detalla la Historia de Usuario 2 - Administrador de Categoría

**Tabla 7:** Historia de Usuario 2 - Administrador de Categoría

HISTORIA DE USUARIO	
<b>Número:</b> 2	<b>Usuario:</b> Gerente y Administrador
<b>Nombre de la historia:</b> Administración de categoría	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> El sistema debe tener la opción de crear, listar, modificar y eliminar registros de categorías.	
<b>Observaciones:</b> esta acción los realiza tanto el Administrador y Encargado de Almacén .	

Fuente: Elaboración Propia

## Historia de Usuario 3

La tabla 8 se detalla la Historia de Usuario 3 - Administrador de Productos.

**Tabla 8:** Historia de Usuario 3 - Administrador de Productos

HISTORIA DE USUARIO	
<b>Número:</b> 3	<b>Usuario:</b> Gerente, Encargado de Ventas
<b>Nombre de la historia:</b> Administración de productos	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> El registro de productos nuevos y actualización de productos, se lo realizará de acuerdo a las características de los productos de la empresa tomando los datos necesarios de los productos los cuales son: código del producto, nombre del producto, categoría del producto, imagen del producto, debe tener la opción de crear, modificar, eliminar y listar.	
<b>Observación:</b> Los datos de los productos deben ser proporcionado por el administrador y encargado de almacén.	

Fuente: Elaboración Propia

#### Historia de Usuario 4

La tabla 9 detalla la Historia de Usuario 4 - Administrador de Compras

**Tabla 9:** Historia de Usuario 4 - Administrador de Compras

HISTORIA DE USUARIO	
<b>Número:</b> 4	<b>Usuario:</b> Gerente, Encargado de almacén.
<b>Nombre de la historia:</b> Administración de compras	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> El administrador o el encargado de almacén podrá realizar el registro de los datos de los productos que ingresan y el stock que ingresa.	
<b>Observación:</b> debe realizarlo el administrador y encargado de almacén debe realizar el registro de ingreso de abastecimiento.	

Fuente: Elaboración Propia

#### Historia de Usuario 5

La tabla 10 se detalla la Historia de Usuario 5 - Administrador de Ventas.

**Tabla 10:** Historia de Usuario 5 - Administrador de Ventas

HISTORIA DE USUARIO	
<b>Número:</b> 5	<b>Usuario:</b> Gerente, Encargado de Ventas.
<b>Nombre de la historia:</b> Administración de ventas	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> Se realizará el registro de las ventas de los productos teniendo en cuenta los datos necesarios para la operación. El encargado de ventas consulta el stock y precio de los pedidos del cliente una vez verificada los productos y la cantidad que dispone, procede a registrar la venta.	
<b>Observación:</b> lo debe realizar el Administrador y el encargado de ventas.	

Fuente: Elaboración Propia

## Historia de Usuario 6

La tabla 11 se detalla la Historia de Usuario 6 Administrador de Proveedores

**Tabla 11:** Historia de Usuario 6 - Administrador de Proveedores

HISTORIA DE USUARIO	
<b>Número:</b> 6	<b>Usuario:</b> Gerente
<b>Nombre de la historia:</b> Administración de proveedores	
<b>Prioridad del negocio:</b> Media	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> Se realizará el registro de las Proveedores los debe realizar el gerente luego de su respectivo ingreso al sistema, tiene la opción de registrar nuevo proveedor, modificar, eliminar y listar.	
<b>Observación:</b> lo debe realizar el Administrador y el encargado de ventas.	

---

Fuente: Elaboración Propia

## Historia de Usuario 7

La tabla 12 detalla la Historia de Usuario 6 – Inventario

**Tabla 12:** Historia de Usuario 6 - Inventario

HISTORIA DE USUARIO	
<b>Número:</b> 7	<b>Usuario:</b> Gerente, Encargado de Ventas.
<b>Nombre de la historia:</b> Inventario	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> El administrador y el personal de venta podrán tener acceso a esta información para que puedan realizar sus actividades respectivamente con distintas acciones o preferencias. Para la creación de un nuevo ingreso o salida de artículo del almacén.	

---

Fuente: Elaboración Propia

## Historia de Usuario 8

La tabla 13 detalla la Historia de Usuario 7 - Reportes

**Tabla 13:** Historia de Usuario 8 - Reportes

HISTORIA DE USUARIO	
<b>Número:</b> 8	<b>Usuario:</b> Gerente, Encargado de Ventas.
<b>Nombre de la historia:</b> Reportes	
<b>Prioridad del negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Eulogia Mamani Chuca	
<b>Descripción:</b> Se desarrollarán reportes diarios de ventas que se realizaran en el día	

Fuente: Elaboración Propia

### 3.3.2 Planificación de Iteraciones

Se realizará las tareas de iteración según la descripción de las historias de usuario, para poder definir el trabajo que se realice.

**Tabla 14:** Descripción de la Tareas de Iteración

No. de Iteración	Historia de Usuario	Tipo de Tarea
1ra Iteración	Gestión de usuario	Permitir la asignación de uno o más roles a un usuario. Asigna los tipos de validación de los usuarios. Realizara los siguientes roles para adicionar, modificar, eliminar y listar.
2da Iteración		Realiza la validación de Categorías.



---

	Administrador de categoría	de	El sistema debe tener la opción de crear, listar, modificar y eliminar y registros de categorías.
3ra Iteración	Administración Productos	de	Realiza la validación de Productos. El sistema debe tener la opción de crear, listar, modificar y eliminar y registros de productos
4ta Iteración	Gestión Compras	de	Diseña la interfaz de las compras. Realiza la validación de las compras, como también podrá adicionar, modificar, eliminar y listar.
5ta Iteración	Gestión de Ventas		Diseña la interfaz de las ventas. Realiza la validación de las ventas, como también podrá adicionar, modificar, eliminar y listar.
6ta Iteración	Gestión proveedores	de	Diseña la interfaz para los proveedores. Realiza la validación de proveedores, como también podrá adicionar, modificar, eliminar y listar.
7ma Iteración	Gestión inventario	de	Diseñar la interfaz para el inventario. Como también podrá adicionar, modificar, eliminar y listar.

---

8va Iteración	Gestión Inventario	de	Diseñar la interfaz de inventario. Realizar la interfaz para el reporte del inventario.
------------------	-----------------------	----	--

Fuente: Elaboración propia

### 3.3.2.1 Plan de Entregas del Proyecto (Release Planning)

Basándonos en las historias de usuario definidas para el desarrollo del sistema web, se ha elaborado el siguiente plan de entrega, el cual muestra las historias de usuario que se llevarán a cabo en cada iteración. Para este plan de entrega se ha tomado en cuenta la prioridad y el esfuerzo de cada historia de usuario.

**Tabla 15:** Plan de Entrega

Iteración	Historia de Usuario	Duración	Fecha de Inicio	Fecha Final
1	Administración de Usuarios	de 4 semanas	6/12/2020	7/01/2020
2	Administración de Categoría	de 4 semanas	10/01/2020	9/02/2020
3	Administración de Producto	de 2 semanas	10/02/2020	23/02/2020
4	Administración de Compra	de 3 semanas	24/02/2020	13/03/2020
5	Administración de Venta	de 4 semanas	15/03/2020	23/05/2020
6	Administración de Inventario	de 3 semanas	25/05/2020	10/06/2020
7	Administración de Reportes	de 3 semanas	15/06/2020	06/07/2020

Fuente: Elaboración Propia

## **3.4 DISEÑO**

### **3.4.1 Modelo Relacional de la Base de Datos**

Se diseñó e implementó la base de datos analizando todos los datos de los requerimientos del cliente para su cumplimiento y el adecuado funcionamiento del sistema. Ver Anexo A.

### **3.4.2 Modelado UWE**

En la parte de modelado haremos una combinación con UWE, ya que nos brinda un conjunto de modelos especializados para el diseño web.

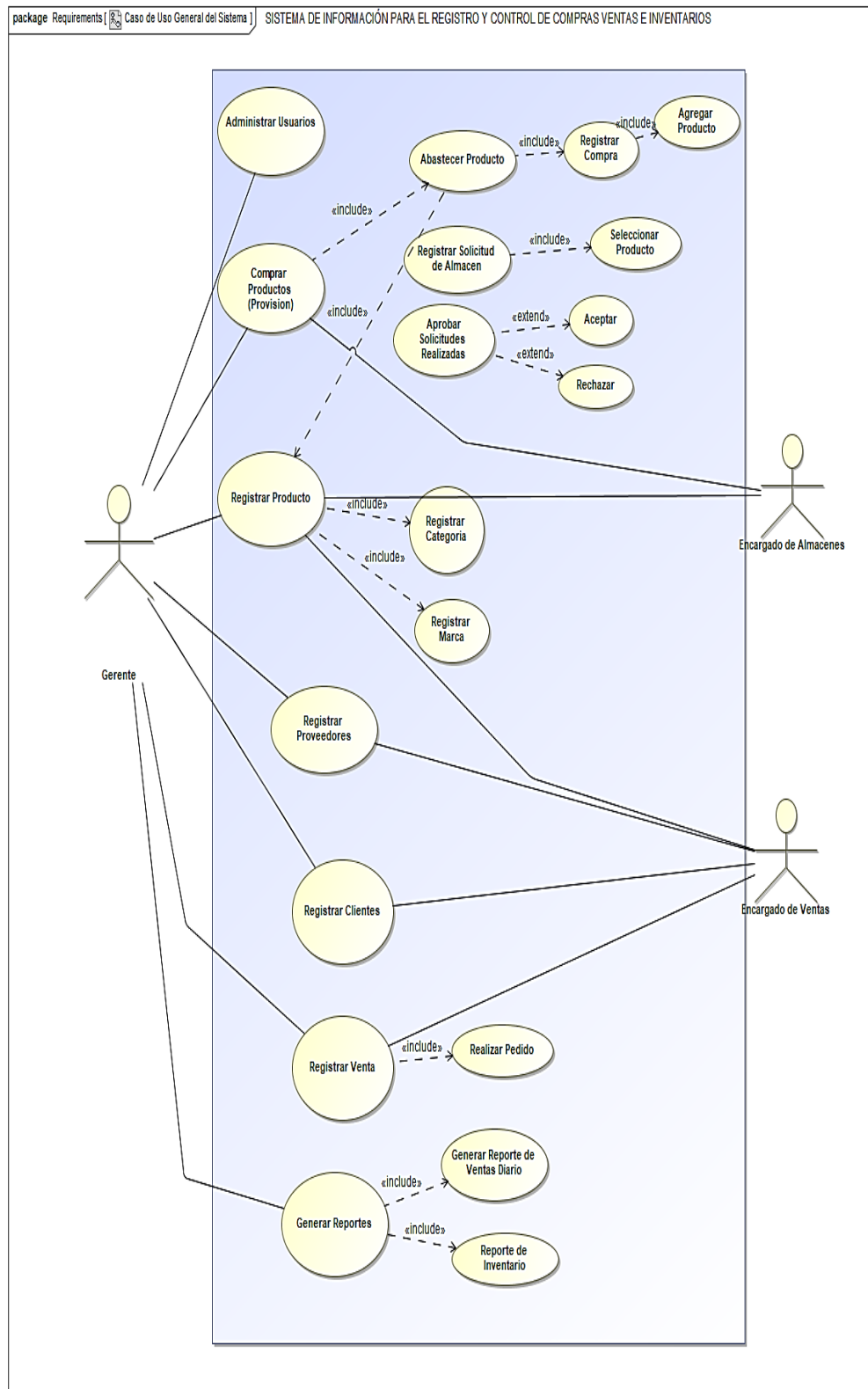
- **Diagrama de Casos de Uso**

Dentro de la etapa de planeación de XP se utilizará también los diagramas de caso de uso de UWE, serán realizados para cada módulo con los que contará el sistema en los cuales se observará la interacción que tienen los usuarios con el sistema.

#### **Diagrama caso de uso general**

En la figura 30 muestra el caso de uso general del sistema web de la empresa TESICON.

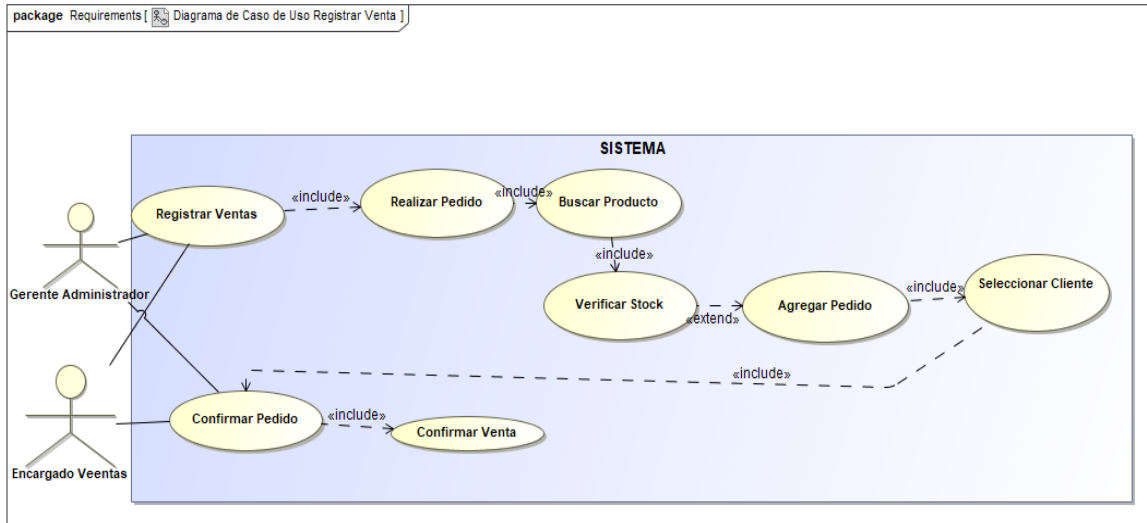
Figura 30: Caso de Uso General del Sistema



Fuente: Elaboración Propia

## Descripción Caso de Uso Registrar Ventas

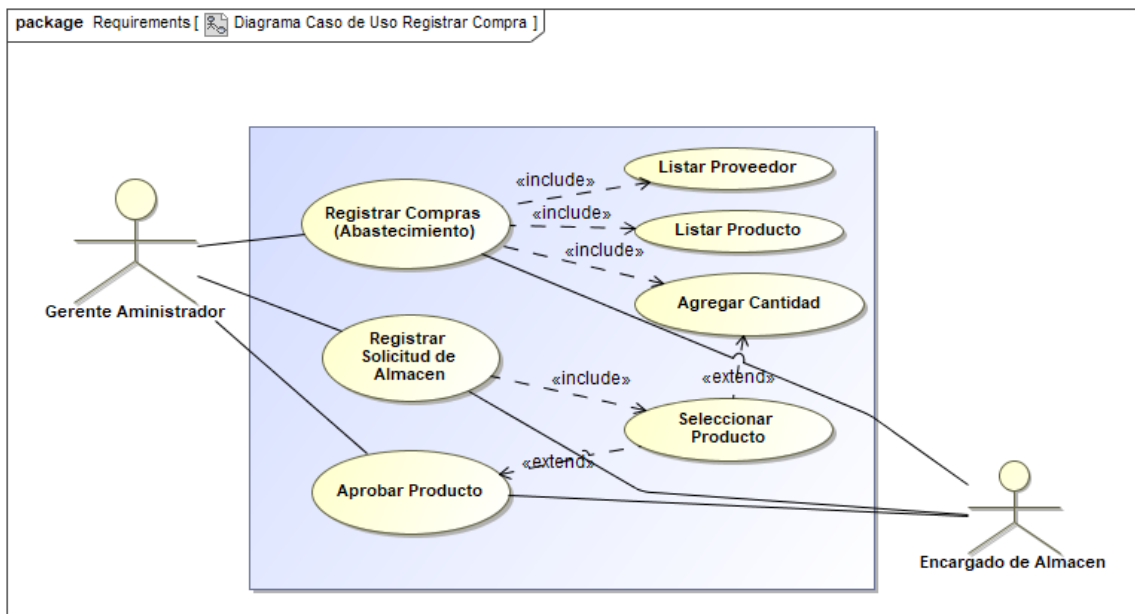
**Figura 31:** Diagrama Caso de Uso Registrar Ventas



Fuente: Elaboracion Propia

## Diagrama Caso de Uso Registrar Compras

**Figura 32:** Diagrama Caso de Uso Registrar Compras

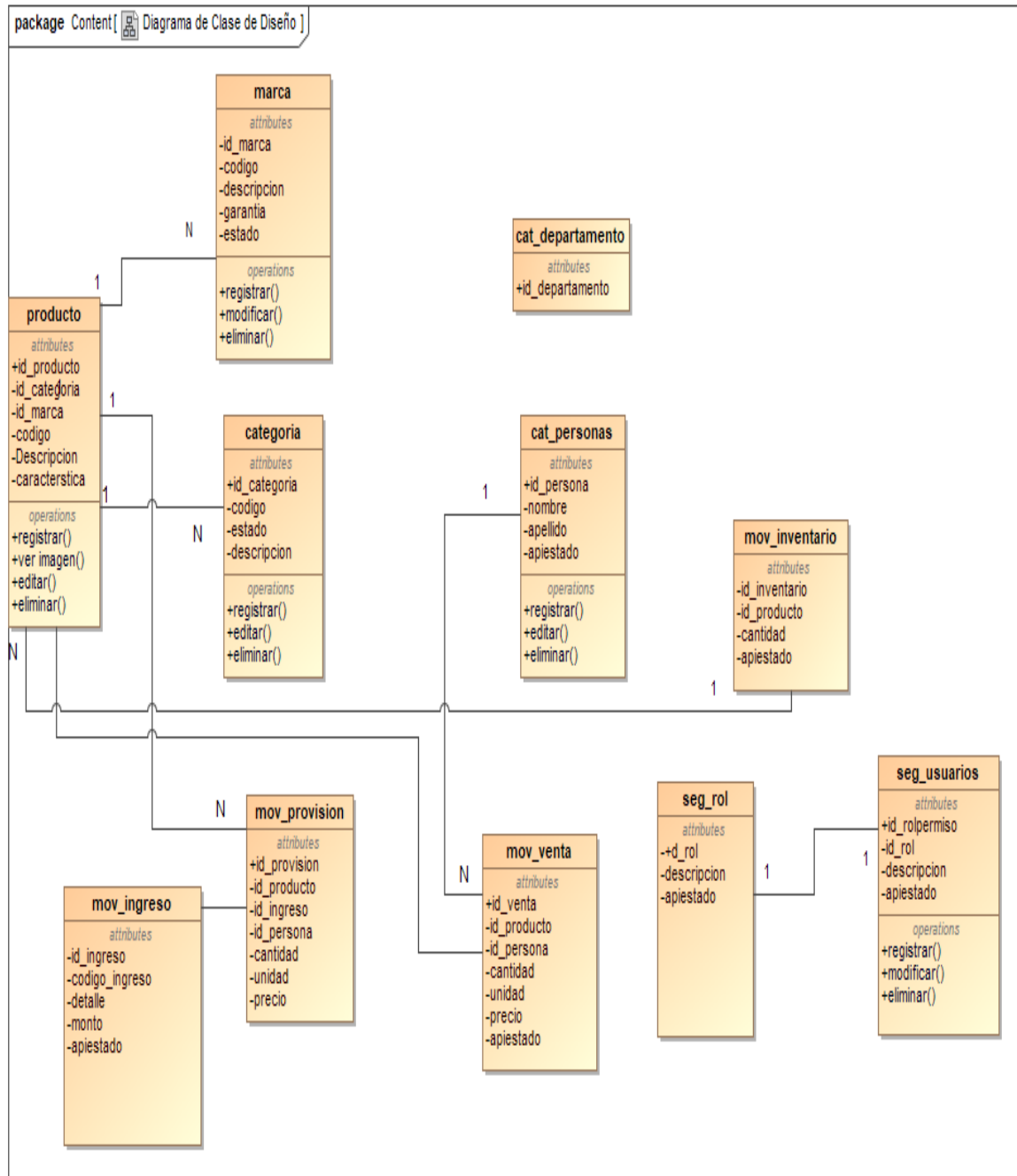


Fuente: Elaboración Propia

## Modelo de Contenido

El modelo de contenido mostrara las relaciones entre las entidades.

Figura 33: Diagrama de Contenido del Sistema

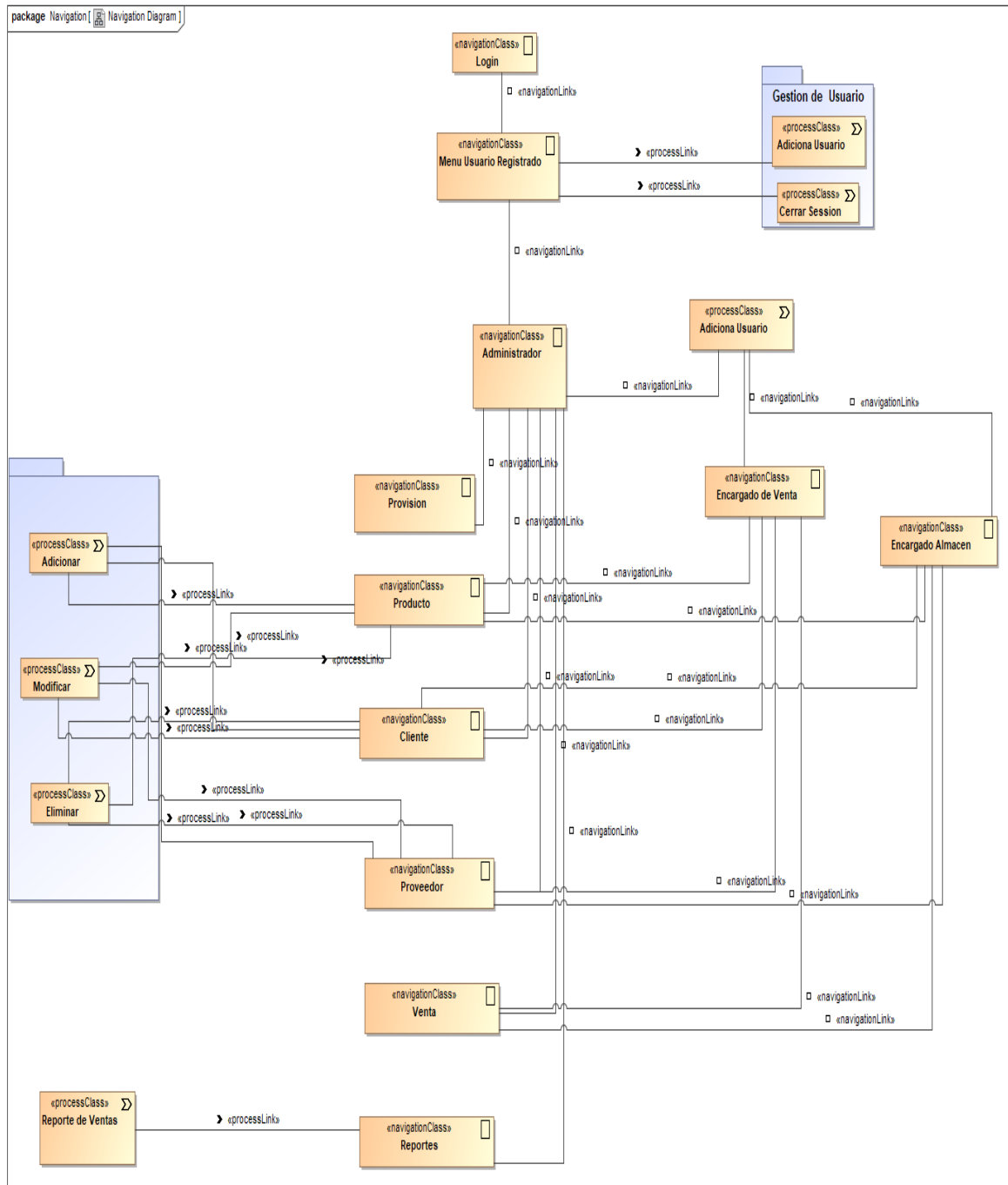


Fuente: Elaboración Propia

## Modelo de Navegación

Como segundo modelo UWE tenemos el modelo de navegación, de tal forma que la navegación sea fácil y usable.

Figura 34: Diagrama Navegacional General

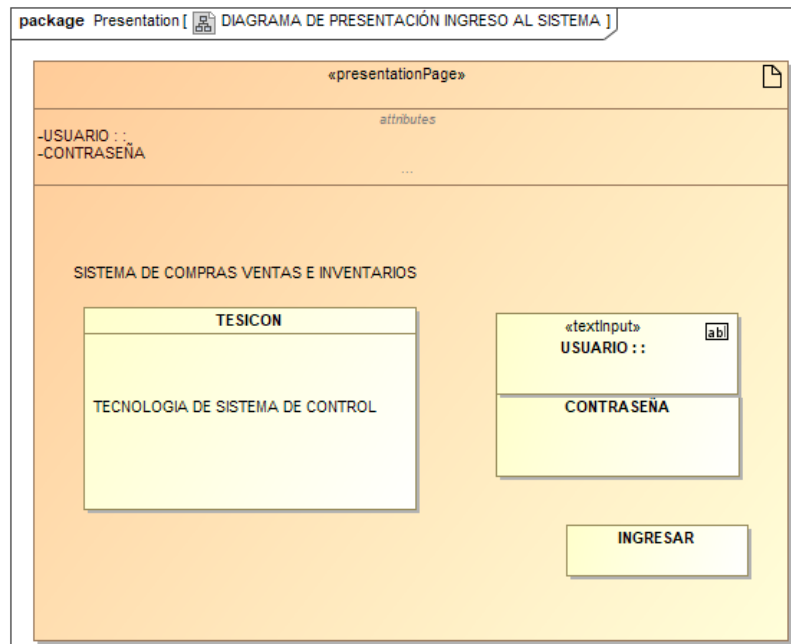


Fuente: Elaboración Propia

## Modelo de Presentación

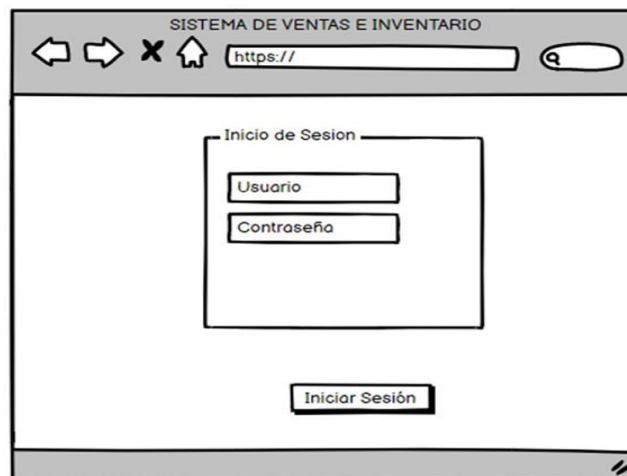
A continuación, presentamos modelos simples donde se puede apreciar como lucirán las vistas de la aplicación, el usuario debe acceder a su perfil e interactuar con el sistema.

**Figura 35:** Diagrama de Presentación Ingreso al Sistema



Fuente: Elaboración Propia

**Figura 36:** Ingreso al Sistema

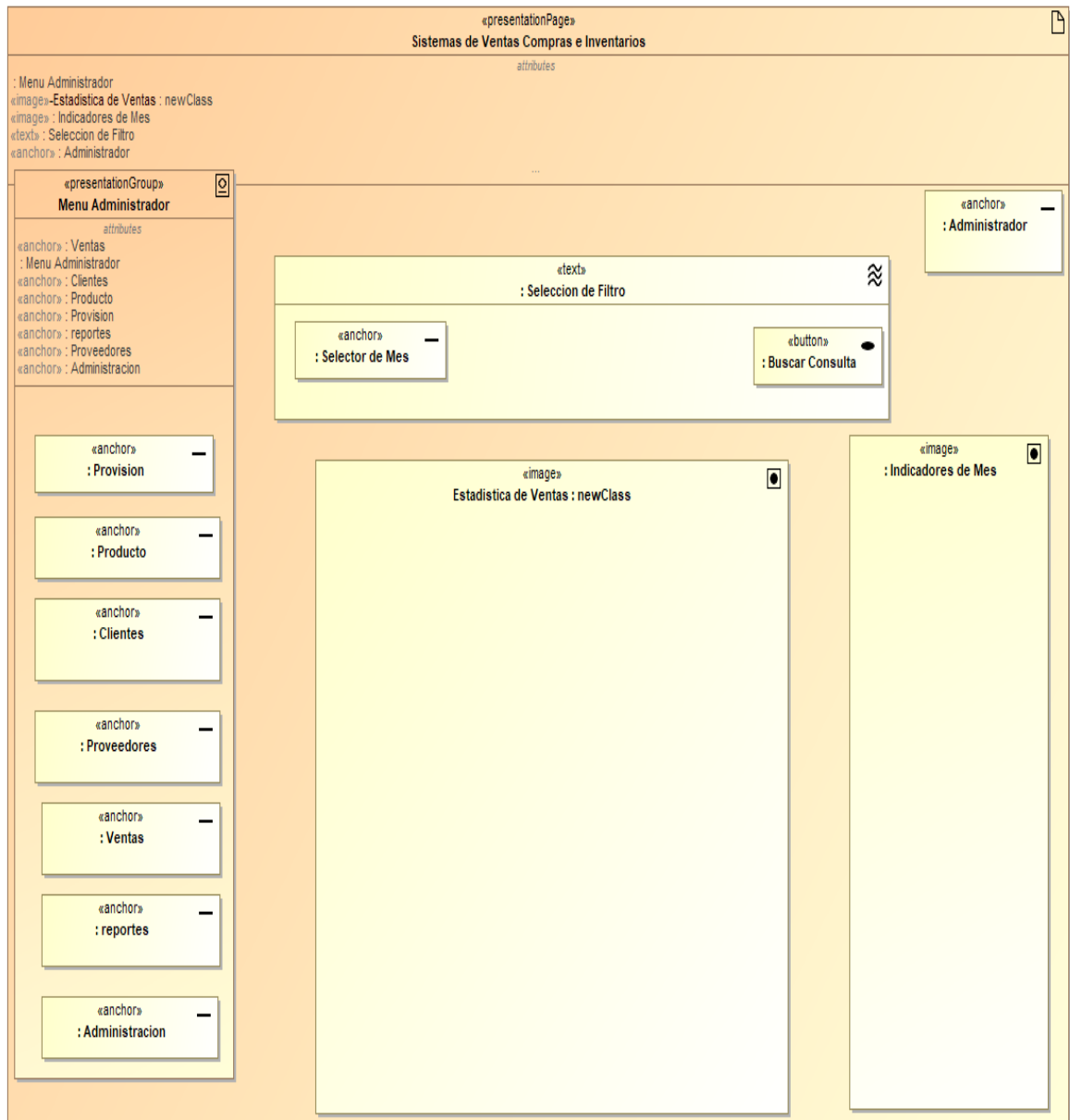


Fuente: Elaboración Propia



## Diagrama de presentación del administrador

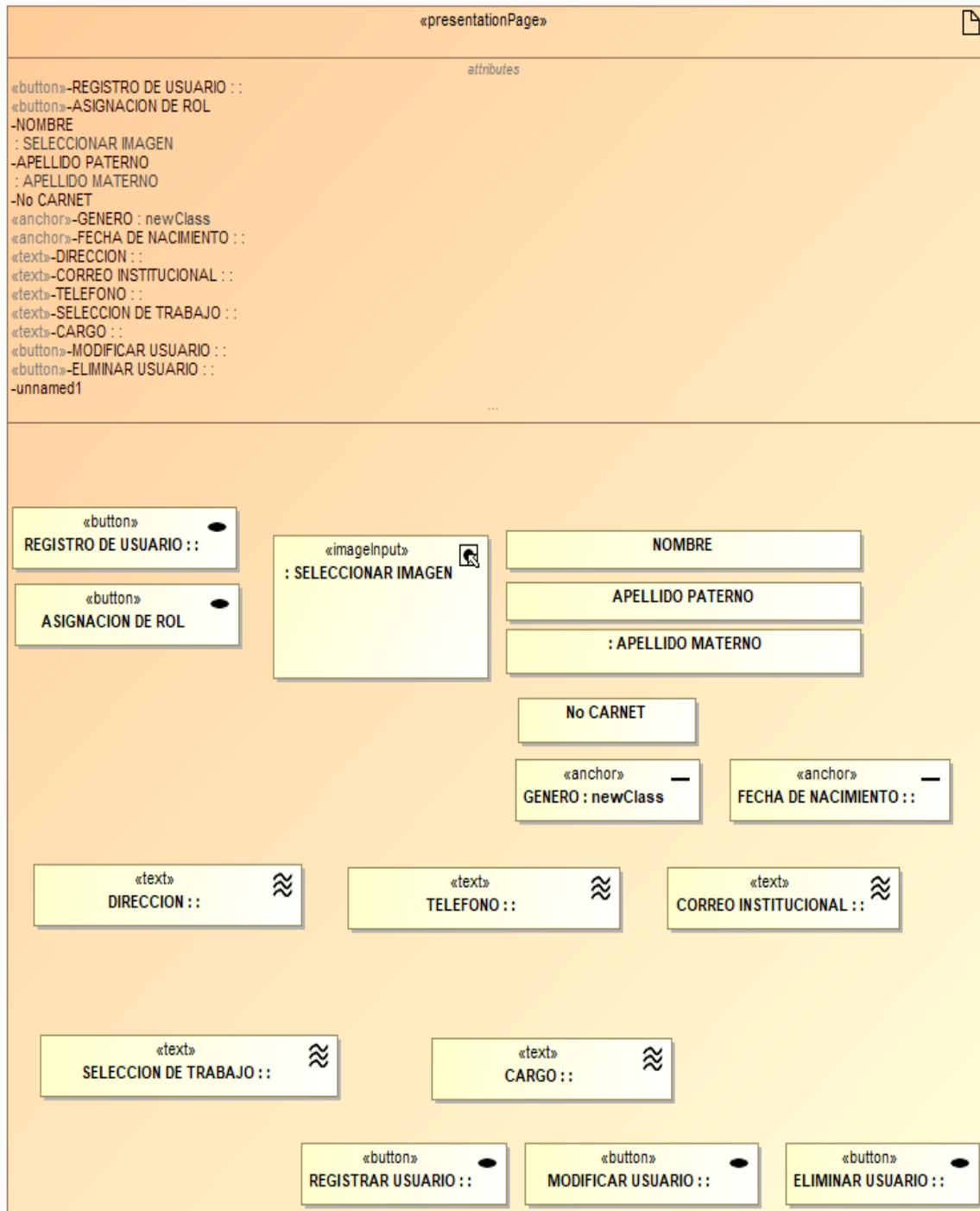
Figura 37: Diagrama de Presentación del Administrador



Fuente: Elaboración Propia

## Administración de usuarios

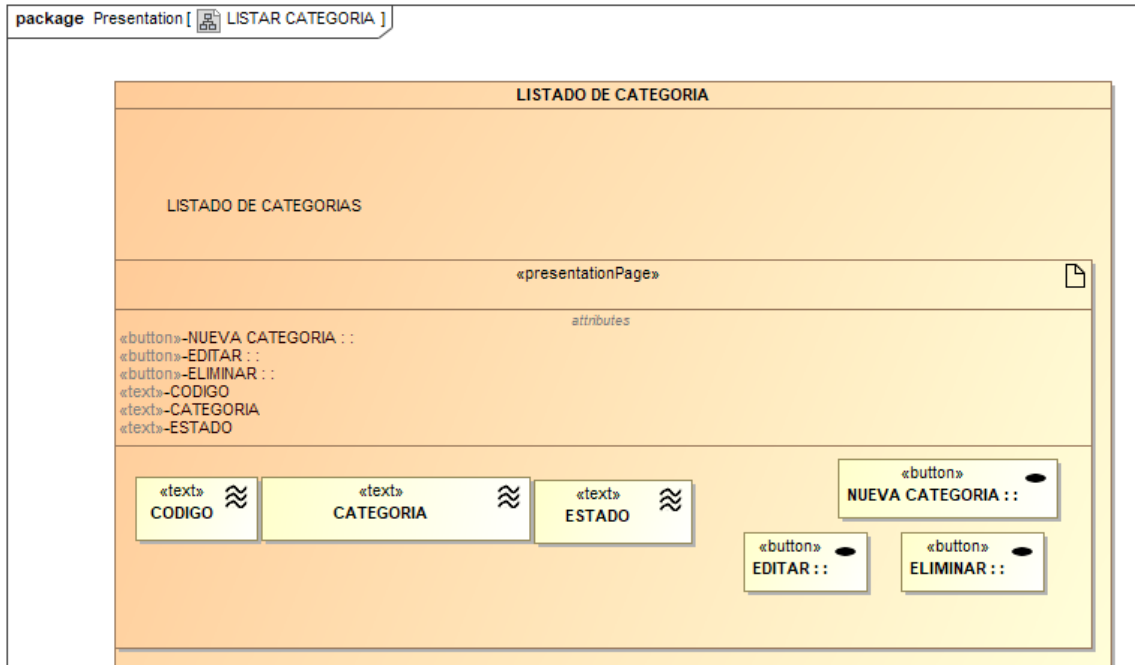
Figura 38: Diagrama de Presentación Administración de Usuarios



Fuente: Elaboración Propia

## Listar Categoría

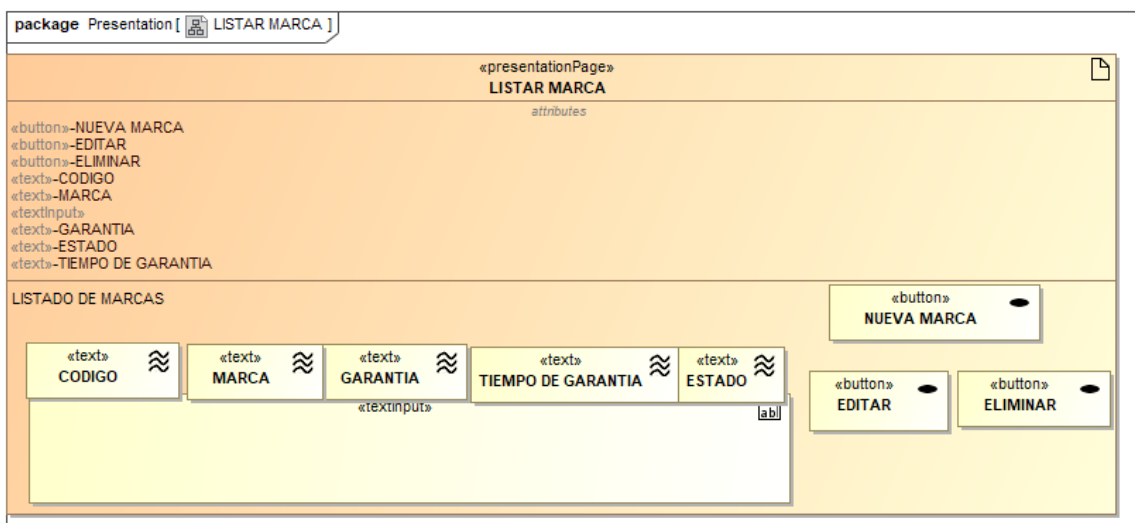
Figura 39: Diagrama de Presentación Listar Categoría



Fuente: Elaboración Propia

## Listar Marca

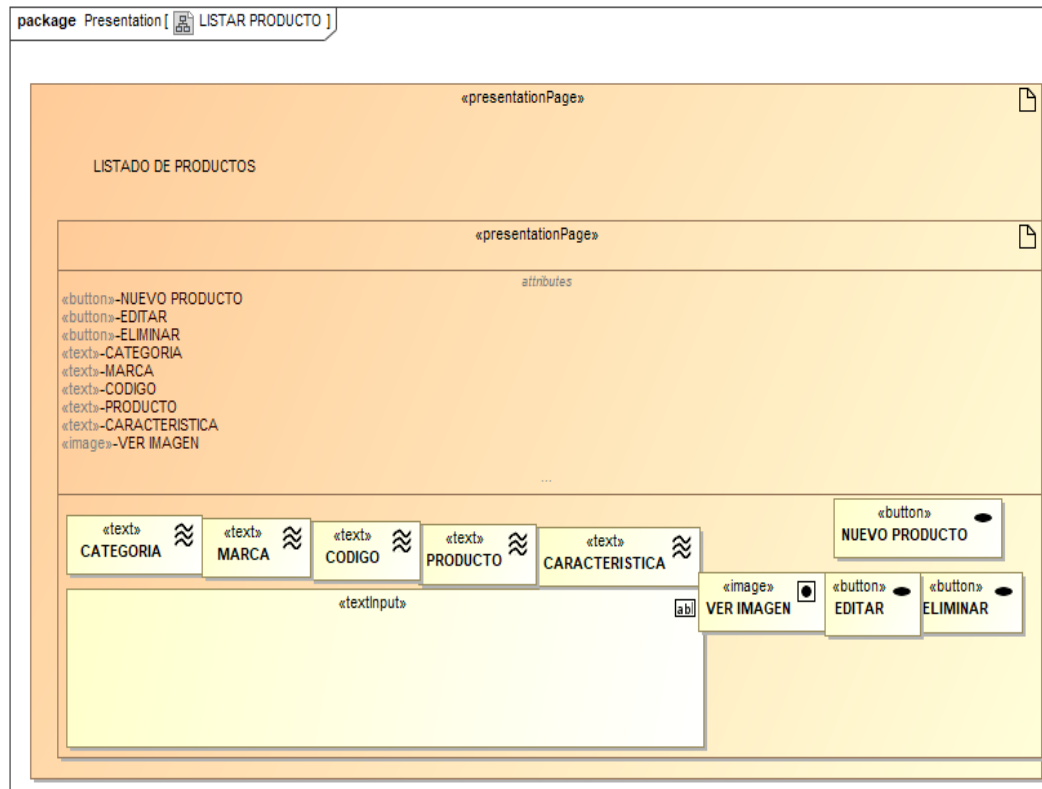
Figura 40: Diagrama de Presentación Listar Marca



Fuente: Elaboración Propia

## Listar Producto

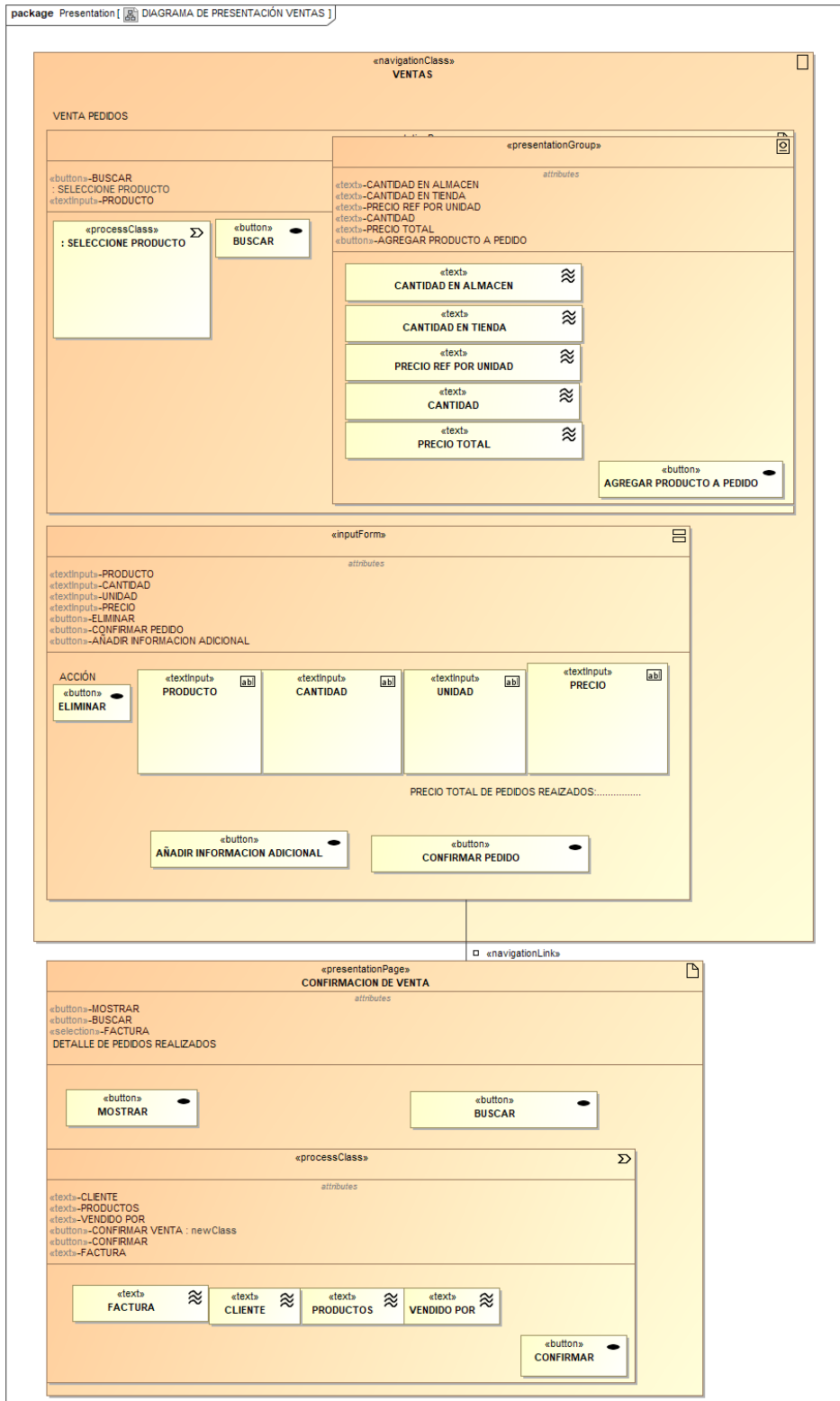
Figura 41: Diagrama de Presentación Listar Producto



Fuente: Elaboración Propia

## Diagrama De Ventas

**Figura 42:**Diagrama de Presentación Ventas



Fuente: Elaboración Propia

Se diseñaron los diagramas de presentación más relevantes, en todo caso el sistema tiene el mismo modelo de presentación, para todas las tablas de la base de datos.

### 3.5 FASE DE DESARROLLO Y CODIFICACIÓN

En esta fase la codificación de cada uno de las historias de usuarios se realizará con el cliente, permitiendo de esa forma una retroalimentación de lo que el cliente quiere.

#### 3.5.1 Modelo de Implementación

En este modelo se muestra la implementación en el sistema mostrando el desarrollo de las interfaces de los diferentes módulos y sus elementos construidos de acuerdo al seguimiento de la metodología UWE, a continuación, se muestra algunas capturas de pantalla más sobresalientes, que son el resultado de las historias de usuarios.

A continuación, se muestra la captura de pantalla para el inicio de sesión en el sistema.

**Figura 43:** Inicio de Sesión del Sistema



Fuente: Elaboración Propia

## Página principal administración del sistema

Figura 44: Página Principal del Sistema



Fuente: Elaboración Propia

## Modulo Administración de Usuarios

Figura 45: Registro de Usuario

The registration form for a user named EULOGIA EYLIN includes the following details:

- Nombre:** EULOGIA EYLIN
- Apellido Paterno:** MAMANI
- Apellido Materno:** CHUCA
- DNI:** 4527850013
- Sexo:** FEMENINO
- Fecha de Nacimiento:** 1985-11-24
- Ubicación:** LA PAZ
- Correo Electrónico:** eyn\_mam\_mai@hotmail.com
- Identificación:** 24098011
- Área de Trabajo:** ALMACEN
- Categoría de Usuario:** ENCARGADO DE ALMACEN

Fuente: Elaboración Propia

**Figura 46:** Asignación de Rol

Administración / Usuarios

PERSONAL - TESICON

Registro de Usuario

Asignación de Rol

**Asignación de Rol**

Usuario  
EULOGIA EYLIN MAMANI CHUCA

Nombres: EULOGIA EYLIN | Apellido Paterno: MAMANI | Apellido Materno: CHUCA

Carnet de Identidad: 4927650013 LPZ | Login Usuario: eulogia.mamani | Permisos: ENCARGADO DE ALMACENES

REGISTRAR ROL

Fuente: Elaboración Propia

## Lista de Usuarios del Sistema

**Figura 47:** Listado de Usuarios del Sistema

Detalle de Usuarios ..

Mostrar 10 filas por página

N°	Login	Nombre Completo	Carnet de Identidad	Correo Institucional	Telefono	Ubicacion de Trabajo	Accion
1	admin.admin	ADMIN ADMIN CHUCA	4927650 LPZ	eylinchuca@gmail.com	69795345	TIENDA	
2	ana.chuca	ANA CHUCA MAMANI	2180975 LPZ	mariachuca@gmail.com	73290860	TIENDA	
3	eddy.choque	EDDY CHOQUE TITO	6050358 LPZ	eddy.heberson.ing@gmail.com	74909001	TIENDA	
4	eulogia.mamani	EULOGIA EYLIN MAMANI CHUCA	4927650013 LPZ	eylin_cruz_azul@hotmail.com	74098011	ALMACEN	
5	santos.lima	SANTOS LIMA HUANCA	602020 LPZ	asdfg@hotmail.com	234567	TIENDA	

Mostrando página 1 de 1

Fuente: Elaboración Propia

## Módulo de Productos

Administrador de Categoría, Marca y Producto



Figura 48: Lista de Producto

Nº	Categorías	Marcas	Código	Producto	Características	Estado	Acción
1	GRABADORES 4, 8, 16 CANALES	LONGISE	XVR204D	Grabador XVR de 4CH 5-IN-1	Grabador XVR de 4CH 5-IN-1 4KDCU/TV/Audio/PT/ICH, Compresión H.264, Resolución 1080P/720P, HD, 1080N/720P, P/C, 2MP, Audio IN/4CH OUT/1CH, 1 SATA/Ases. ETE, 1 RUAS 10/100Mbps, Network P/P 1 HDMI, 1 VGA, 2 USB, Cámara Médica.	ELABORADO	
2	GRABADORES 4, 8, 16 CANALES	LONGISE	XVR203D	Grabador XVR de 4CH 5-IN-1	Grabador XVR de 4CH 5-IN-1 4KDCU/TV/Audio/PT/ICH, Compresión H.264, Resolución 1080P/720P, HD, 1080N/720P, P/C, 2MP, Audio IN/4CH OUT/1CH, 1 SATA/Ases. ETE, 1 RUAS 10/100Mbps, Network P/P 1 HDMI, 1 VGA, 2 USB, Cámara Médica.	ELABORADO	
3	GRABADORES 4, 8, 16 CANALES	LONGISE	XVR203A	Grabador XVR HD-TVCU/4WD 1080P	Grabador XVR de 4CH 5-IN-1 4KDCU/TV/Audio/PT/ICH, Compresión H.264, Resolución 1080P/720P, HD, 1080N/720P, P/C, 2MP, Audio IN/4CH OUT/1CH, 2 SATA/Ases. ETE, 1 RUAS 10/100Mbps, Network P/P, 1 HDMI, 1 VGA, 1 RS422, Entrada PTZ, 1 IR Control, 2 USB Backup/Mouse, DC 12V.	ELABORADO	
4	GRABADORES 4, 8, 16 CANALES	LONGISE	XVR211D	Grabador XVR de 16CH 5-IN-1	Grabador XVR de 16CH 5-IN-1 4KDCU/TV/Audio/PT/ICH, Compresión H.264, Resolución 1080P/720P, HD, 1080N/720P, P/C, 2MP, Audio IN/4CH OUT/1CH, 1 SATA/Ases. ETE, 1 RUAS 10/100Mbps, Network P/P, 1 HDMI, 1 VGA, 2 USB, Cámara Médica.	ELABORADO	
5	GRABADORES 4, 8, 16 CANALES	LONGISE	XVR113L	Grabador XVR de 32CH 5-IN-1	Grabador XVR de 32CH 5-IN-1 4KDCU/TV/PT/ICH, Audio, Compresión H.264, Audio IN/32CH, OUT/1CH, Alarmas IN/1CH OUT/1CH, Resolución 1080P/720P, HD, 1080N/720P/P/C, 2MP, 8 SATA, Hard Disk Interface (3.5" SATA, Up to 4TB) + 1 6-SATA Interface, 1 RUAS 10/100/1000 Mbps, Network P/P, 1 HDMI, 1 VGA, 1 RS422, Entrada PTZ, 2 USB Backup/Mouse (Total incoming bandwidth 1500M).	ELABORADO	
7	KITS DE 4 CANALES	LONGISE	PLC200A/15100	KIT PLC COMUNICACION POR LINEA ELECTRICA	KIT PLC COMUNICACION POR LINEA ELECTRICA, 4 Cámaras PLC IP 720P/Plástico IP66, IR 60M, ONVIF, 1 NVR PLC R 2 SATA/Ases. ETE, 2 RUAS 10/100 Mbps, Network P/P, 1 HDMI, 1 VGA, 2 USB, Auto IN/4CH OUT/1CH, AC 100-240V, Disco Duro de 1TB.	ELABORADO	
8	ACCESORIOS	SIN MARCA	ABS-LTPHD	Video Ballun HD 1 Canal Pasivo TVCVI/4WD 1MP, 2MP	Video Ballun HD 1 Canal Pasivo TVCVI/4WD 1MP, 2MP	ELABORADO	
9	CAMARAS HD 720P 1MP	LONGISE	LBNDTV100A	Cámara Bala Médica 1MP HDTVI	Cámara Bala Médica 1MP HDTVI, 1/4" Sensor CMOS Active, 720P/960H, Filtro IR-CUT, DiafNicho, Iluminación 0.01 Lux@1.2, DNR, IR LED 20PCS, IR 20M, IP66, Lente f/3.6mmF2.0, Salida de Video TV/Video.	ELABORADO	
10	CAMARAS HD 720P 1MP	LONGISE	LPNDHTC100B	Cámara Dome Médica 1MP HDTVI	Cámara Dome Médica 1MP HDTVI, 1/4" Sensor CMOS CVI, 720P/960H, DiafNicho, 0.01 Lux@1.2, Control OSD, IR LED 14x, IR 10PCS, IR 20M, Lente f/3.6mm, Salida de Video TVCVI/4WD/Analog.	ELABORADO	

Fuente: Elaboración Propia

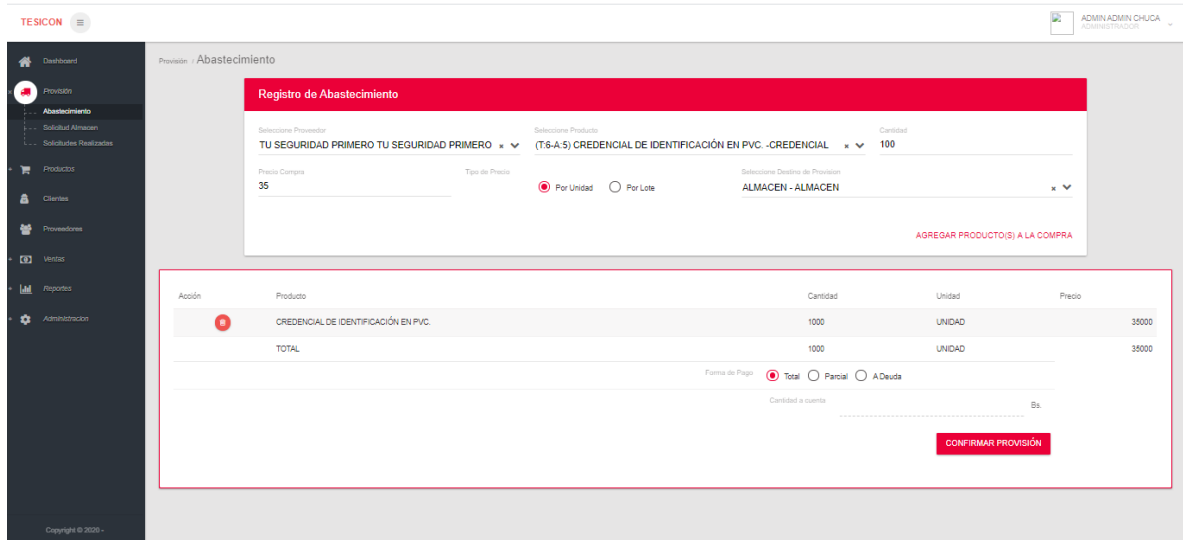
## Lista de Categoría

Nº	Código	Categoría	Estado	Acción
1	001	CAMARAS HD 720P 1MP	ELABORADO	
2	002	CAMARAS HD 1080P 2MP	ELABORADO	
3	003	GRABADORES 4, 8, 16 CANALES	ELABORADO	
4	005	ACCESORIOS	ELABORADO	
5	006	ENERGIA	ELABORADO	
6	007	ALARMAS	ELABORADO	
7	008	BIOMETRICOS	ELABORADO	
8	009	REDES	ELABORADO	
9	010	CREDENCIALES	ELABORADO	
10	011	OTROS	ELABORADO	

## Módulo Provision

### Administrador de Compras Abastecimiento

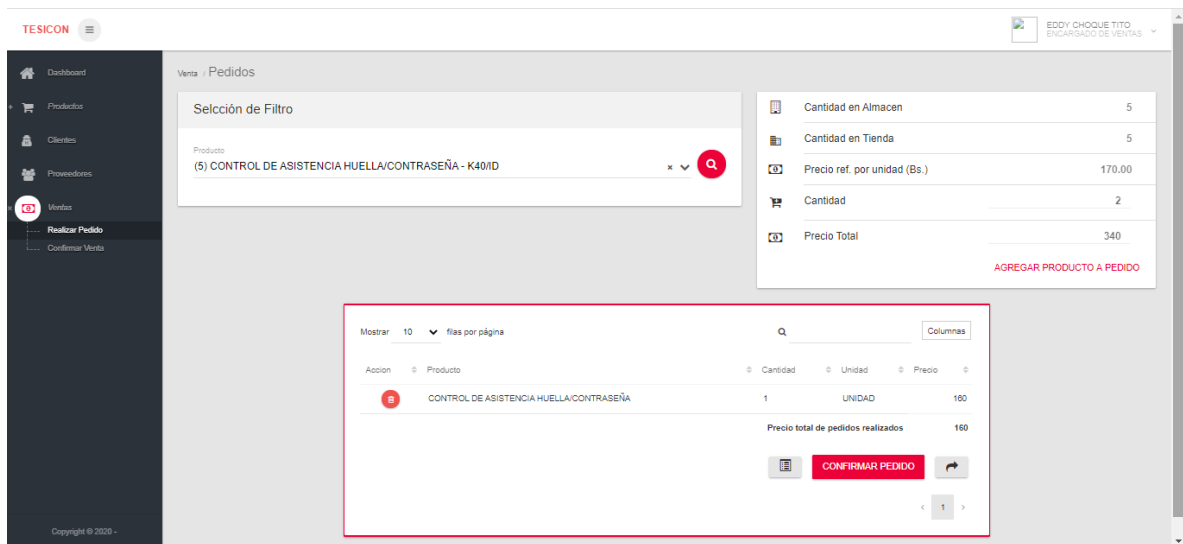
**Figura 49: Registro de Abastecimiento**



Fuente: Elaboración Propia

## Módulo de Ventas

**Figura 50: Realizar Pedido de Venta**



Fuente: Elaboración Propia

## Confirmar Pedido

**Figura 51: Confirmar Pedido de la Venta Realizada**

Venta / Confirmación de Venta

Detalle de pedidos realizados

Mostrar 10 filas por página

Nº	Lote	Factura	Cliente	Productos	Vendido por	Acción
Pedido 1	18	Sin Factura	Sin Cliente	Cámara Bala Metálica 2.4MP VARIFOCAL HDTV	admin.admin	Ver
Pedido 2	20	Con Factura	VIVIAN FRANCO RAMOS	CONECTOR PLUG DC PARA ALIMENTACIÓN (UNIDAD), Grabador 8CH HD-TV/CV/IAHD 1080P, CAPUCHÓN PARA RJ45 COLOR AZUL, Cámara Domo Metálica 2.1MP HDTV	admin.admin	Ver
Pedido 3	21	Sin Factura	Sin Cliente	CONTROL DE ASISTENCIA HUELLA/CONTRASEÑA, CONTROL DE ASISTENCIA HUELLA/CONTRASEÑA	eddy.choque	Ver

Fuente: Elaboración Propia

## Modulo Proveedores

**Figura 52: Listar Proveedores**

Listado de Proveedores

Registrar Proveedor

Razón social  
CASA IDEAS S

Sigla  
CCI

NIT  
34567890

MODIFICAR PROVEEDOR REGISTRAR PROVEEDOR

+ NUEVO PROVEEDOR

Mostrar 10 filas por página

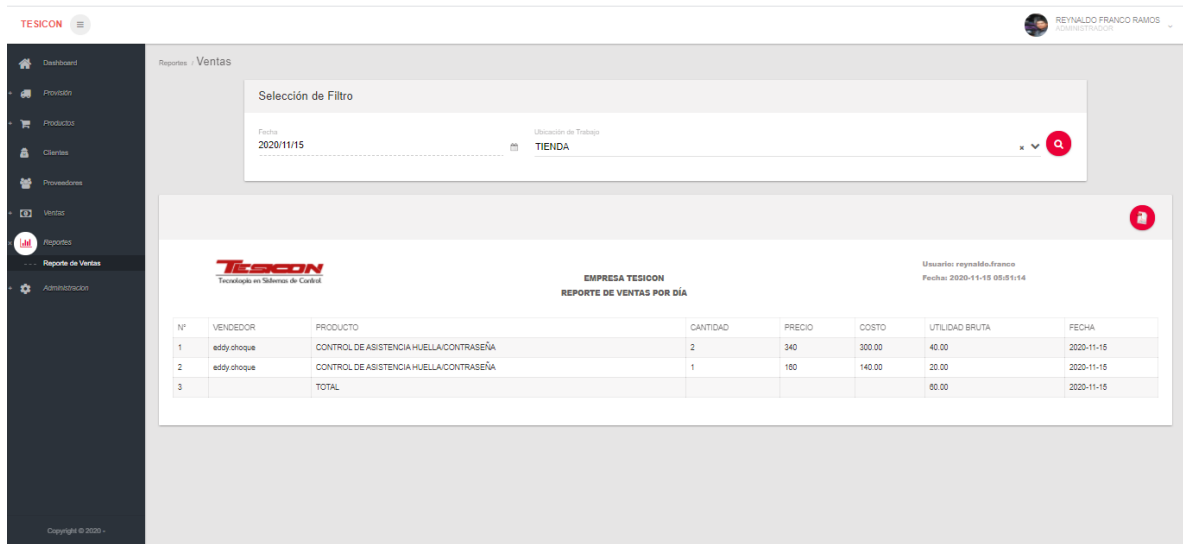
Nº	Razón Social	Sigla	NIT	Estado	Acción
1	CASA SEGURA	CASA SEGURA	1234568015	ELABORADO	Ver Editar
2	ALIBABA	ALIBABA	2345412017	ELABORADO	Ver Editar
3	XTREME - COMERCIAL	XTREME - COMERCIAL	10958789430108	ELABORADO	Ver Editar
4	TU SEGURIDAD PRIMERO	TU SEGURIDAD PRIMERO	9087854015	ELABORADO	Ver Editar
5	CENTER INFORMATIC	CENTINF	9876543012	ELABORADO	Ver Editar

Mostrando página 1 de 1

Fuente: Elaboración Propia

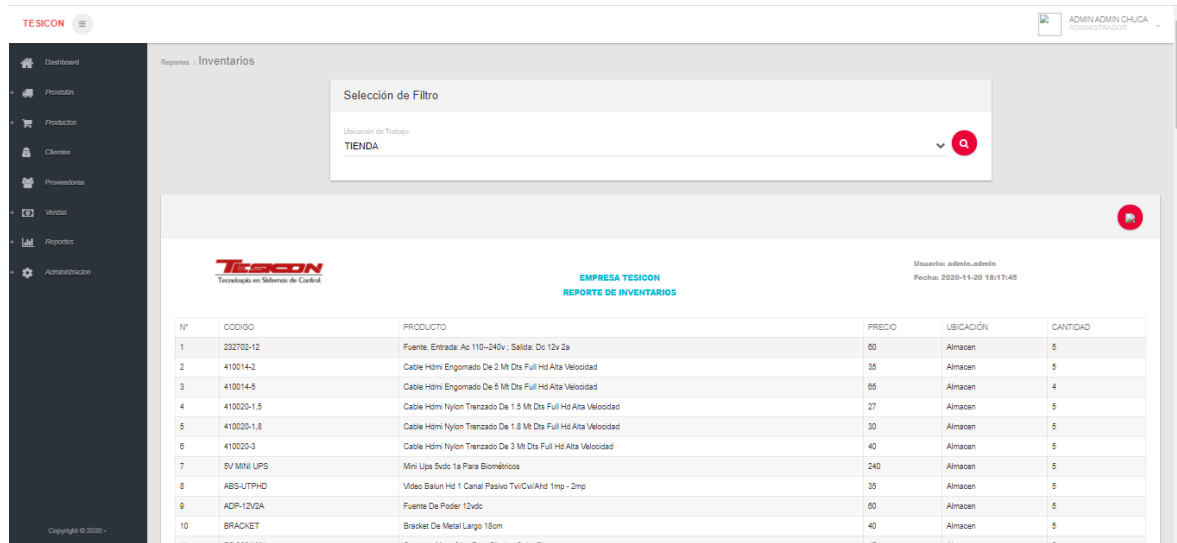
## Módulo de Reportes

**Figura 53 : Reporte de Venta Diaria**



Fuente: Elaboración Propia

**Figura 54: Reporte de Inventarios**



Fuente: Elaboración Propia

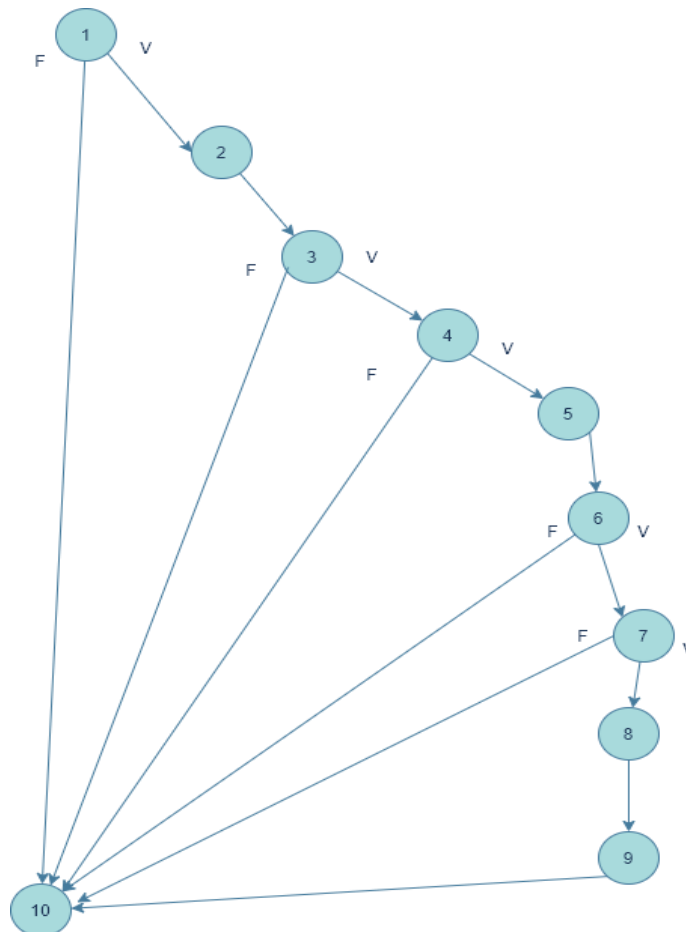
### 3.6 PRUEBAS

Se realizará las pruebas de software a los módulos desarrollados.

#### 3.6.1 Pruebas de Caja Blanca

Esta prueba se centrará en los detalles del cálculo de las regiones que deben ser consideradas como caminos independientes del sistema, estableciendo cuales son las entradas para que se ejecuten, asegurando así que cada región se ejecute al menos una vez. Para la prueba de caja blanca se tomará la forma general, se debe seguir:

**Figura 55:** Prueba de Caja Blanca



Fuente: Elaboración propia

Donde:

- Inicio del sistema 1
- Validación de datos 2
- Crear datos del cliente 3
- Registrar cliente 4
- Crear venta 5
- Registrar venta 6
- Buscar venta 7
- Agregar producto de venta 8
- confirmar venta 9
- fin 10

Número de nodos = 10

Número de aristas = 14

Analizado el grafo generado a partir de las características del sistema, ahora se procede a determinar la complejidad ciclomática del grafo mediante:

$$V(G) = A - N + 2$$

Dónde:

**A:** 14 (Aristas)

**N:** 10 (Nodos)

Por tanto:

$$V(G) = 14 - 10 + 2$$

$$V(G) = 6$$

Determinar el conjunto básico de caminos linealmente independientes. Los caminos que deben ser probados dadas ciertas variables son 6. Estos caminos son los siguientes:

**Camino 1:** 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 - 10

**Camino 2:** 1 – 2 – 3 – 4 – 5 – 6 – 7 - 10

**Camino 3:** 1 – 2 – 3 – 4 – 5 – 6 - 10

**Camino 4:** 1 – 2 - 3 - 4 - 10

**Camino 5:** 1 – 2 – 3 - 10

**Camino 6:** 1 - 10

Preparar los casos de prueba para forzar la ejecución de cada camino. Esta última condición establece que, para la ejecución de ciertos caminos, se deben establecer las condiciones en las que al menos se ejecuta los nodos establecidos en el camino.

**Camino 1:** Se ejecuta cuando el inicio de registro de ventas, se valida los campos requeridos del sistema, se crea los datos del cliente, si todo está correctamente llenado, se crea la venta, se registra la venta, se busca la venta, agrega productos de venta, luego valida la venta, por ultimo finaliza el proceso de confirmar la venta.

**Camino 2:** Se ejecuta cuando el inicio de registro de ventas, se valida los campos requeridos del sistema si todo está correctamente llenado, se crea los datos del cliente, si todo está correctamente llenado, se crea los datos de la venta, se registra la venta, busca la venta, si no existe la venta entonces finaliza la venta.

**Camino 3:** Se ejecuta cuando el inicio de registro de ventas, se valida los campos requeridos del sistema si todo está correctamente llenado, se crea los datos del cliente, si todo está correctamente llenado se crea los datos de la venta, si existe un error en la venta se elimina la venta, entonces se finaliza el proceso de venta y la base de datos hace un rollback o reversión.

**Camino 4:** Se ejecuta cuando el inicio de registro de ventas, se valida los campos requeridos del sistema si todo está correctamente llenado, se crea los datos del cliente, si hay un error en los datos del cliente entonces se finaliza el proceso de ventas y no registra nada en nuestra base de datos.

**Camino 5:** Se ejecuta cuando el inicio de registro de ventas, se valida los campos requeridos del sistema, crea datos del cliente, si no llena los datos del cliente hay error entonces finaliza el proceso de ventas.

**Camino 6:** Se ejecuta cuando el inicio de registro de ventas, se valida los registros de formulario, si hay un campo sin datos, entonces se finaliza el proceso de ventas y devuelve error.

### 3.6.2 Pruebas de Caja Negra

Las pruebas de caja negra o también conocidas como pruebas de comportamiento del software.

#### Prueba de Autenticación

El usuario que quiera ingresar al sistema debe contar con un usuario y una contraseña, estos son asignados por un administrador del sistema, en este caso el administrador es el Gerente General de la empresa TESICON.

**Figura 56:** Autenticación Exitosa- Autenticación Fallida



Fuente: Elaboración propia



Una vez que el usuario realiza la autenticación con su nombre de usuario y contraseña el sistema le permite el ingreso a la página principal según el privilegio que se le haya asignado, si el usuario no corresponde el sistema no le permite el acceso.

Tabla 16:Valores de Entrada - Proceso de inicio del usuario

<b>Campo</b>	<b>Ingreso exitoso</b>	<b>Ingreso fallido</b>
Usuario	Nombre del usuario	Espacio vacío
Contraseña	Contraseña del usuario	Espacio vacío

Fuente: Elaboración propia

Se realiza las pruebas a la interfaz mostrada a continuación

Tabla 17: Prueba de Caja Negra Inicio de Usuario

<b>Entradas/Usuario</b>	<b>Contraseña</b>	<b>Salida</b>	<b>Resultado</b>
Correcto/incorrecto		Ingrese usuario y contraseña	Si el nombre está lleno y la contraseña está vacía el usuario debe completar el campo vacío.
	Correcto/incorrecto	Ingrese usuario y contraseña	Si la contraseña está llena y el nombre está vacía el usuario debe completar el campo vacío.
Correcto/incorrecto	Incorrecto	El nombre y contraseña no coincide	Si el nombre es correcto/incorrecto y la contraseña es incorrecto el usuario debe ingresar nuevamente los datos

---

	Correcto/incorrecto	El nombre y contraseña no coincide	Si la contraseña es correcto/incorrecto y el nombre es incorrecto el usuario debe ingresar nuevamente los datos
Incorrecto			
	Incorrecto	El nombre y contraseña no coincide	Si ambos campos son incorrectos el usuario deberá ingresar nuevamente los datos.
Incorrecto			
	Correcto	El usuario ingresa al sistema	Si ambos campos son correctos el usuario ingresa al sistema.
Correcto			

---

Fuente: Elaboración propia

Realizando esta prueba se ha podido constatar que el proceso de autenticación cumple con sus funciones correctamente.

## **CAPÍTULO IV**

# **CALIDAD Y SEGURIDAD**

## 4 CALIDAD Y SEGURIDAD

En este capítulo se destaca la aplicación de normas de desarrollo de software con el objetivo de que la aplicación obtenga la calidad y seguridad del sistema y este sea necesaria y suficiente para llegar a satisfacer las necesidades del usuario final.

### 4.1 Calidad de Software

El desarrollar un software de calidad tiene como objetivo cumplir y satisfacer las necesidades del cliente, existen diversas opciones, los estándares y modelos de evaluación y mejora de los procesos de software que están relacionados con la calidad, el presente proyecto usara la técnica ISO 9126.

#### 4.1.1 Usabilidad

La usabilidad consiste de un conjunto de atributos que permite evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, es decir realizar una serie de preguntas que permiten ver cuán sencillo, fácil de aprender y manejar es para los usuarios.

Es decir, facilidad de uso, esta métrica nos muestra el costo de aprender a manejar el producto, en la siguiente tabla se observa estos criterios en niveles de porcentajes a los que llego el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema.

Calculamos la siguiente ecuación:

$$FU = \frac{\left[ \frac{\sum x_i}{n} * 100 \right]}{n}$$

En la tabla se calcula  $x_i$  y  $\sum x_i$  utilizando las evaluaciones:

**Tabla 18:** Calculo de Usabilidad

Nº	Preguntas	Evaluación (xi)
1	¿El acceso al sistema es sencillo?	8
2	¿El sistema es agradable a la vista?	7
3	¿El sistema satisface los requerimientos de manejo de información?	8
4	¿Las salidas del sistema están de acuerdo a sus requerimientos?	7
5	¿Cómo considera el ingreso de datos del sistema?	8
6	¿Cómo considera los formularios que elabora el sistema?	7
7	¿El sistema facilita el trabajo que realiza?	7
<b>TOTAL y <math>\sum x_i</math></b>		<b>52</b>

Fuente: Elaboración Propia

Calculamos FU:

$$FU = \frac{\left[ \frac{52}{8} * 100 \right]}{8}$$

**FU = 84**

**Usabilidad = 84 %**

Por tanto, la facilidad de uso es de 84 %.

#### 4.1.2 Funcionabilidad

La funcionabilidad está orientada a la función del software y del proceso por el cual se desarrolla por esta razón que debemos medirla con el punto de función que cuantifica el tamaño y la complejidad del sistema en termino de función de usuario se define de la siguiente forma.

**Tabla 19:** Parámetros de Medición

Parámetros de Medición
1 Nro. De Entradas de Usuario
2 Nro. De Salida de Usuario
3 Nro. De Peticiones de Peticiones de Usuario
4 Nro. De Archivos
5 Nro. De Interfaces Externas

Fuente: Elaboración Propia

- **Entradas de Usuario:** Se refiere a cada entrada que proporciona datos al sistema, registro de roles, registro de usuario, registro de proveedores.
- **Salidas de Usuario:** Se refiere a cada salida que proporciona el sistema al usuario, entre estos están: permiso de acceso a menús, mostrar mensajes de error, reporte de ventas, etc.
- **Peticiones de Usuario:** Una petición se define como una entrada interactiva y esta genera una respuesta de software en forma de salida autenticación de usuario, listar menús, listar roles, listar usuarios editar roles, editar usuarios, editar proveedores, editar, productos.
- **Número de Archivos:** Se cuenta archivos maestros lógico, estos pueden ser: grupo lógico de datos o un archivo independiente
- **Interfaces Externas:** se cuenta las interfaces legibles por la máquina que se utiliza para transmitir información a otro sistema internet, intranet.

Para calcular el punto de función se utilizará la siguiente formula:

$$PF = \text{Cuenta Total} * (\text{Grado de confiabilidad} + \text{tasa de error} * \sum Fi)$$

Donde:

**PF:** Medida de la funcionalidad

**Cuenta Total:** Resultado del conteo de parámetros

**X:** Confiabilidad del proyecto, varía entre el 1% a 100%

**Min(Y):** Error mínimo aceptable a la complejidad, el margen de error es igual a 0.01

**Fi:** Valores de ajuste de la complejidad, donde i=1 a i=14

En la siguiente tabla se calcula el punto de función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de programación.

**Tabla 20:** Conteo de Parámetros PF

PARÁMETROS	CUENTA	FACTOR DE PONDERACIÓN	SUB TOTAL
Entradas de usuario	20	*4	80
Salidas de usuario	5	*5	25
Peticiones de usuario	19	*4	76
Número de archivos	8	*10	80
Número de interfaces	2	*7	14
<b>TOTAL</b>			<b>275</b>

Fuente: Elaboración Propia

Se realiza los valores de ajuste de complejidad de acuerdo a la siguiente tabla.

Tabla 21: Valores de Ajuste de Complejidad

IMPORTANCIA	0	1	2	3	4	5
<b>ESCALAS</b>	<b>Sin importancia</b>	<b>Incremental</b>	<b>Moderado</b>	<b>Medio</b>	<b>Significativo</b>	<b>Escencial</b>
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2.- ¿Se requiere comunicación de datos?						X
3.- ¿Existe funciones de procesos distribuidos?					X	
4.- ¿Es critico el rendimiento?					X	
5.- ¿El sistema web será ejecutado el SO. Actual?						X
6.- ¿Se requiere una entrada interactiva para el sistema?						X
7.- ¿Facilidad Operativa?						X
8.- ¿Se actualiza los archivos de forma interactiva?						X
9.- ¿Son complejas las entradas, salidas, los archivos o las peticiones?					X	
10.- ¿Es complejo el procesamiento interno del sistema?					X	
11.- ¿Se ha diseñado el código para ser reutilizado?						X
12.- ¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X
13.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X
<b>CUENTA TOTAL</b>						$\sum (Fi) = 61$

Fuente: Elaboración Propia



- Calcularemos el punto de fusión PF, con la siguiente formula

$$PF = \text{Cuenta Total} * (\text{grado de confiabilidad} + \text{Tasa de Error} * \Sigma(Fi))$$

Remplazando datos de la ecuación tenemos:

$$PF = 275 * (0.65 + (0.01 * 61))$$

$$PF = 275 * (0.65 + 0.61)$$

$$PF = 275 * 1.26$$

$$\mathbf{PF \text{ obtenida} = 346.5}$$

- Si consideramos el máximo valor es decir 70 y lo dividimos con los puntos de función obtenida del ajuste de complejidad como al 100% de los factores se obtendría de la siguiente manera:

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \Sigma(Fi))$$

$$PF = 275 * (0.65 + (0.01 * 70))$$

$$PF = 275 * (0.65 + 0.70)$$

$$PF = 275 * 1.35$$

$$\mathbf{PF \text{ ideal} = 371.25}$$

Remplazando la ecuación de Funcionalidad será:

$$\text{Funcionalidad} = \frac{PF \text{ obtenida}}{PF \text{ ideal}} \times 100$$

$$\text{Funcionalidad} = \frac{346.5}{371.25} \times 100$$

$$\text{Funcionalidad} = 0.93 \times 100$$

$$\mathbf{\text{Funcionalidad} = 93.39 \%}$$

La funcionabilidad del sistema es de 93 % eso quiere decir que tiene una utilidad para la empresa, es decir que el sistema cumple con los requisitos funcionales de forma satisfactoria.

#### 4.1.3 Confiabilidad

Se define como la probabilidad de operación libre de fallas de un programa de computadoras en un entorno determinado y durante un tiempo específico.

Para medir la confiabilidad se tiene que medir entre fallas el cual se calculará con la siguiente ecuación:

$$F(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Donde:

**F(t)** = Confiabilidad del Sistema

**Funcionabilidad = 0.93**

$\lambda = 0.01$  (es decir 1 error en cada 6 ejecuciones)

**t = 12 meses**

Hallamos la confiabilidad del sistema:

$$F(12) = (0.93) * e^{-\frac{1}{6} * 12}$$

$$\mathbf{F(12) = 0.125}$$

Reemplazando en la fórmula de no hallar una falla se tiene:

$$P(T>t) = 1 - F(t)$$

$$P(T>t) = 1 - 0.125$$

$$\mathbf{P(T>t) = 0.875}$$

$$P(T>t) = 0.875 * 100$$

$$\mathbf{P(T>t) = 87.5 \%}$$

**Confiabilidad = 87.5 %**

Con este resultado podemos decir que la probabilidad de que el sistema no presente fallas es de 87.5 % durante los próximos 12 meses.

#### **4.1.4 Mantenibilidad**

La mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sean por la corrección de errores o por el incremento de funcionalidad.

Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software. Se determina la siguiente función (IMS):

$$IMS = \frac{Mt - (Fc + Fa + FE)}{Mt}$$

Donde:

**Mt:** Número de Módulos total de la versión actual

**Fc:** Número de Módulos en la versión actual que se han Cambiado.

**Fa:** Número de Módulos en la versión que se han Añadido.

**Fd:** Numero de Módulos en la versión anterior que se han borrado en la versión actual.

Calculando el IMS

$$IMS = \frac{7 - (1 + 0 + 0)}{7} = 0.85$$

Para la interpretación del resultado multiplicamos por 100 para sacar el porcentaje de la siguiente forma:

$$\text{Mantenibilidad} = 0.85 * 100$$

**Mantenibilidad = 85 %**

De acuerdo a los datos obtenidos se concluye que el sistema, tiene una Mantenibilidad de 85 % se encuentra en el rango satisfactorio.

#### 4.1.5 Portabilidad

La portabilidad se refiere al esfuerzo necesario para transferir el programa de un entorno ya sea de Hardware y/o software a otro, es una característica deseable de todo software, considerando los siguientes aspectos.

- **Capacidad de instalación:** Su instalación es de fácil realización, tomar en cuenta las versiones de los servicios de framework utilizados y la versión del servidor local xamp.
- **Capacidad de reemplazamiento:** Los reemplazos o modificación de códigos es de fácil acceso por la ubicación, ya que cuenta con la arquitectura de diseño que sugiere codigniter para el código fuente separándolo en tres grupos: modelo, vista y controlador, esta arquitectura aprovecha las ventajas de separación de código entre presentación, lógica y modelo de base de datos.
- **Adaptabilidad:** Puede ser adaptado según la necesidad del cliente en entornos específicos.
- **Facilidad de instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.

Entonces el sistema se ejecuta en todos los servidores web, ya que no necesita una instalación. Se comprobó en los distintos navegadores más usados, se le da una calificación de 97 % de Portabilidad.

**Portabilidad = 97 %**

#### 4.1.5.1 Resultados

Ahora veremos la calidad total de la aplicación, se tomará en cuenta los datos obtenidos con referencia las cuales se definieron anteriormente esto lo mostraremos en la siguiente tabla.

**Tabla 22:** Medición de Acuerdo al Resultado Obtenido

<b>CALIDAD GLOBAL DEL SOFTWARE</b>	
<b>Características</b>	<b>Resultado</b>
Usabilidad	84
Funcionabilidad	93.39
Confiabilidad	87.5
Mantenibilidad	85
Portabilidad	97
<b>Evaluación de Calidad</b>	<b>89.37</b>

Fuente: Elaboración Propia

El nivel de aceptación es satisfactorio, indica que los valores de preferencia se encuentran en el rango de 60% a 100%, lo que indica que es satisfactorio para el usuario al interactuar con el sistema.

**Evaluación de Calidad Total = 89.37 %**

## 4.2 SEGURIDAD DEL SISTEMA

### 4.2.1 Seguridad

Se define la seguridad de la información al conjunto de medidas técnicas y organizativas que permiten a la organización para asegurar la confidencialidad, disponibilidad e integridad del sistema.

Para la seguridad del sistema se considerará las siguientes precauciones:

#### **4.2.1.1 Seguridad a nivel usuario**

A nivel usuario se hará la validación de autenticación o verificación de un usuario para que tenga que ingresar con su contraseña que se asignó anteriormente.

#### **4.2.1.2 Seguridad a nivel aplicación**

- ✓ El sistema es capaz de controlar el acceso a partir de usuario y contraseña.
- ✓ Se aplica el MVC para tener seguridad en el código.
- ✓ Los backup deben ser almacenadas en distintos lugares, en áreas seguras y que solo se permita el acceso al personal autorizado.
- ✓ Los formularios serán validados, para evitar datos erróneos y esta sea almacenados en la base de datos.

#### **4.2.1.3 Seguridad a nivel Base de Datos**

A nivel de base de datos se aplica la seguridad de copias de seguridad, acceso y permiso a cuentas de usuario, mediante una contraseña y los backups.

## **CAPÍTULO V**

# **COSTOS DEL SOFTWARE**

## 5 COSTO DEL SOFTWARE

### 5.1 COCOMO II

Para determinar el Costo Total del proyecto se tomará en cuenta el costo de software, costo de implementación de la aplicación y elaboración del proyecto. Estos métodos de estimación de costos se engloban en el grupo de modelos algorítmicos que tratan de establecer una relación matemática entre el esfuerzo y tiempo de desarrollo.

Para nuestro caso utilizaremos el modelo de desarrollo intermedio que es el que realiza las estimaciones con precisión.

#### Estimación de costos

para la estimación de costos utilizaremos la siguiente ecuación:

$$E = a(KLDC)^b * ME, \text{ en personas mes}$$

$$T = c(E)^d, \text{ en meses}$$

$$P = \frac{E}{T}, \text{ en personas}$$

Donde:

**E:** Es el esfuerzo requerido por el proyecto, en personas/mes.

**T:** Es el tiempo requerido del proyecto, en meses.

**P:** Es el número de personas requeridas por el proyecto.

**a, b, c, d:** Son constantes con valores definidos en la figura 27 según cada modo de desarrollo.

**KLDC:** es la cantidad de líneas de código.

**ME:** Es el multiplicador de esfuerzo que depende de 15 atributos.

#### 5.1.1 Costo del desarrollo del software

Para determinar el costo de la aplicación se usa el modelo COCOMO II, la estimación de este sistema se desarrolló bajo las KLDC (Kilo Líneas de Código).



Para calcular el proyecto TESICON se implementó con 3500 líneas de código en lenguaje PHP.

Realizando conversiones tenemos lo siguiente:

$$\text{KLCD} = \text{LDC}/1000$$

$$\text{KLDC} = 3500/1000$$

$$\text{KLDC} = (3.500) \text{ KLDC}$$

La evaluación del sistema ha sido considerada en **3.500 KLDC**.

En este proyecto se utilizará el modo de desarrollo orgánico porque los programadores no son experimentados, además el proyecto no es muy complejo. Aplicaremos el modelo intermedio según la siguiente tabla que se muestra.

**Tabla 23:** Modo de Desarrollo COCOMO intermedio

Proyecto de Software	a	B	c	d
Orgánico	3.2	1.05	2.5	0.38
Semiacoplado	3	1.12	2.5	0.35
Empotrado	2.8	1.2	2.5	0.32

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

### 5.1.2 Costo de Implementación del Software

Para hallar los valores ME utiliza la siguiente tabla, y esta se obtiene mediante la multiplicación de los valores evaluados de los 15 atributos siguientes:

**Tabla 24:** Multiplicadores de Esfuerzo ME

MULTIPLICADORES DE ESFUERZO (ME)			VALORACIÓN				
			muy bajo	Bajo	Nominal	alto	muy alto
<b>Atributos Del Producto</b>							
1	RELY	Fiabilidad requerida del Software	0.75	0.88	1.00	1.15	1.4
2	DATA	Tamaño de la base de datos		0.94	1.00	1.08	1.16
3	CPLX	Complejidad del producto	0.70	0.85	1.00	1.15	1.3 1.65
<b>Atributos De La Computadora</b>							
4	TIME	Restricciones del tiempo de ejecución			1.00	1.11	1.3 1.66
5	STOR	Restricciones del almacenamiento principal			1.00	1.06	1.21 1.56
6	VIRT	Inestabilidad de la máquina virtual		0.87	1.00	1.15	1.3
7	TURN	Tiempo de respuesta del computador		0.87	1.00	1.07	1.15
<b>Atributos Del Personal</b>							
8	ACAP	Capacidad del analista	1.46	1.19	1.00	0.86	0.71
9	AEXP	Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82
10	PCAP	Experiencia de los programadores	1.42	1.17	1.00	0.86	0.70
11	VEXP	Experiencia de S. O. utilizado	1.21	1.10	1.00	0.90	
12	LEXP	Experiencia en el lenguaje de programación	1.14	1.07	1.00	0.95	
<b>Atributos Del Proyecto</b>							

13	MODP	Uso de prácticas de programación modernas	1.24	1.1	1.00	0.91	0.82
14	TOOL	Uso de herramientas software	1.24	1.1	1.00	0.91	0.83
15	SCED	Restricción en la duración del proyecto	1.23	1.08	1.00	1.04	1.1
<b>TOTAL</b>							<b>0.561</b>

Fuente: Elaboración Propia

El multiplicador de esfuerzo es:

**ME= 0.56**

### Calculando el esfuerzo

Aplicamos las ecuaciones de la formula esfuerzo.

$$E = a(KLDC)^b * ME, \quad \text{Personas Mes}$$

$$E = 3.2 * (3500)^{1.05} * 0.56 \quad (\text{Persona/Mes})$$

$$\mathbf{E = 9.1 \text{ (Persona/Mes)}}$$

### Calculando el tiempo de desarrollo

$$T = c(E)^d, \text{ en meses}$$

$$T = 2.5 * (9.1)^{0.38} \text{ en meses}$$

$$\mathbf{T = 5.78 \text{ en meses}}$$

### Calculando la productividad

$$P = \frac{E}{T}, \text{ en personas}$$

$$P = 9.1 / 5.78 \text{ en personas}$$

$$\mathbf{P = 1.57 \text{ en personas mes}}$$

### 5.1.3 Costo por Mes de Proyecto

Calculamos el costo por mes de la persona (Salario promedio \$300).

**Costo Mes = Persona \* Salario promedio entre programadores**

Remplazando tenemos:

$$\text{Costo Mes} = 2 * 300$$

$$\text{Costo Mes} = 600 \$$$

Costo total del proyecto

**Costo Total = Costo mes \* Tiempo**

$$\text{Costo Total} = \text{Costo Mes} * T$$

$$\text{Costo Total} = 600 * 5.7$$

$$\text{Costo Total} = 3.420 \$us$$

En conclusión, se necesita 2 personas para desarrollar el proyecto en un tiempo de 5 meses y 7 días el cual su costo será de 3.420 \$us esto equivale en bolivianos **23.598 Bs.**

### 5.1.4 Costo de Implementación de la aplicación

Debido a que la aplicación está desarrollada con software libre y las mismas son utilizadas por muchas empresas, el sistema será utilizada por la empresa.

### 5.1.5 Costo de Elaboración del Proyecto

El proyecto comprende la siguiente serie de gastos que se realizan a lo largo de las diferentes fases. Costo elaboración del proyecto con los requisitos para su elaboración.

**Tabla 25:**Costo de Elaboración

<b>Detalle</b>	<b>Importe en Bs.</b>
Análisis y diseño de la aplicación	1000
Material de escritorio	300
Investigación en internet	800
Otro	200
<b>Total</b>	<b>2300</b>

Fuente: Elaboración propia

### 5.1.6 Costo total del Software

El costo del software se lo obtiene de la sumatoria del costo de la aplicación desarrollada, costo de elaboración del proyecto.

**Tabla 26:** Resultados Finales

<b>Detalle</b>	<b>Importe en Bs.</b>
Costo de Desarrollo	23598
Costo de Elaboración	2300
<b>Total</b>	<b>25898</b>

Fuente: Elaboración propia

Costo total del proyecto es de **25.898 bolivianos** o su equivalencia en dólares es 3.737 \$us.

## **CAPÍTULO VI**

# **CONCLUSIONES Y RECOMENDACIONES**

## **6 CONCLUSIONES Y RECOMENDACIONES**

En culminación del presente proyecto de grado, se llegaron a las siguientes conclusiones y recomendaciones que se citan a continuación:

### **6.1 CONCLUSIONES**

Se concluye con los objetivos planteados y estos han sido alcanzados satisfactoriamente, de manera que, una vez finalizado el desarrollo del Sistema de Información para el Registro y Control de Compras, Ventas e Inventarios de la empresa TESICON se produjo un producto de calidad que cumple con los requerimientos de la empresa.

- ✓ Se logró automatizar la información de todos los movimientos de la empresa TESICON.
- ✓ Se optimizó la provisión de abastecimiento de productos.
- ✓ Se logró automatizar el registro de producto ordenando por categoría y marca.
- ✓ Se logró mejorar la atención a los clientes y proveedores ya que sus datos están guardados y la búsqueda es rápida.
- ✓ Se optimizó el movimiento de ventas y pedidos.
- ✓ Se logró generar reportes confiables por día, por fechas e inventario.

### **6.2 RECOMENDACIONES**

A partir del presente trabajo se propone las siguientes recomendaciones con el fin de buscar el mejoramiento del sistema:

- ✓ Se recomienda a los usuarios que interactúan con el sistema, que los datos de ingreso al sistema sean de conocimiento estrictamente confidencial o personal, debe tener una contraseña segura para evitar usuarios no

autorizados al sistema, debe realizar el cambio de contraseña periódicamente por seguridad.

- ✓ Se recomienda realizar copias de seguridad de la base de datos periódicamente.
- ✓ Se recomienda a la empresa implementar, utilizar y administrar el sistema de acuerdo a las instrucciones brindadas.
- ✓ Capacitar a nuevos administradores para que puedan realizar sus operaciones del sistema y así poder administrarlos correctamente.
- ✓ En caso de que se quiera la ampliación y creación de nuevos módulos, se recomienda revisar la documentación antes de tomar decisiones, debido a que el sistema presenta elementos reutilizables que podrían ser utilizados en los nuevos módulos.



## BIBLIOGRAFÍA

- Grupo Global. (27 de agosto de 2020). *Enlaces rápidos a Postgresql*. Obtenido de <https://www.postgresql.org/about/>
- Aparicio Gil, C. (06 de Mayo de 2012). <https://www.eoi.es/blogs/cesaraparicio/2012/05/06/el-modelo-cocomo-para-estimar-costes-en-un-proyecto-de-software/>. Obtenido de el-modelo-cocomo-para-estimar-costes-en-un-proyecto-de-software.
- Arias, A. (2017). *Aprende Programacion WEb con PHP y Mysql*. España: IT Campus Academy.
- Aspiazu Gutierrez, A. A. (2016). *Sistema de gestión de exámenes clínicos y enfermedades de la altura*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA".
- Beck, K. (2002). *Metodologías de Desarrollo de Software de Programación Extrema (eXtreme Programming o XP)*. Estados Unidos.
- Bustamante, D., & Rodríguez, J. (2014). Metodologia Actual Metodologia XP. (*Proyecto de Grado*). Universidad Nacional Experimental de los Llanos Occidentales Ezequiel Zamora, Barinas.
- Callisaya, W. (2017). *SOFTWARE DE GESTION Y CONTROL DE INVENTARIOS*. La Paz.
- Chavez, K. (29 de 11 de 2012). *Ingenieria Web*. Recuperado el 20 de 04 de 2019, de Conceptos Web: Chavez, K. (29 de Noviembre de 2012). Ingeniería Web. Recuperar<http://kevinchavez93.blogspot.com/2012/11/ingenieria-Web.html>
- Chiri, C. (2009). *Sistema de Entradas y Salidas e Inventario Caso: BOLITAL S.R.L.* Universidad Mayor de San Andres, La Paz - Bolivia, La Paz.
- Echeverry , L., & Delgado, L. (2007). *Caso Practico de la Metodologia Agil al Desarrollo de Software*. Universidad Tecnologica de Pereira, Pereira.
- EcuRed. (2011). *Metodologias de desarrollo de Software*. Recuperado el 2019, de [https://www.ecured.cu/Metodologias\\_de\\_desarrollo\\_de\\_Software](https://www.ecured.cu/Metodologias_de_desarrollo_de_Software)
- Ferrer, J., & Robles, G. (2002). Programacion Extrema y Software Libre. *proyecto de Grado*. Universidad Rey Juan Carlos , y Universidad Politecnica de Madrid, Madrid.

- Garita Gonzalez, G., & Lizano Madriz, F. (31 de Enero de 2018). Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO. *UNICIENCIA*.
- Guajardo, P. (04 de 12 de 2020). *Bootstrap: Guía para principiantes de qué es, por qué y cómo usarlo*. Obtenido de <https://rockcontent.com/es/blog/bootstrap/>
- Hernandez, U. (22 de 02 de 2015). <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>. Obtenido de */mvc-model-view-controller*.
- Huanca Cantuta, S. P. (2015). *Sistema web de control de pagos, citas e historiales clínicos "Caso: Clínica Dental Lavadent"*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz.
- ISO 9126. (19 de Junio de 2016). Obtenido de INFORMATICA: <https://sites.google.com/site/informaticamcprats/iso-9126>
- Jozkowicz. (2008). "Nuevas Técnicas de Desarrollo de Software en Ingeniería Telemática. *Reglas y Practicas en Extreme Programming*. Doctorado de Ingeniería Telemática de la Universidad de Vigo, España.
- Larman, C. (1999). *UML y Patrones*. Mexico: Camara Nacional de la Industria Editorial Mexicana N 1524.
- Laudon, K., & Laudon, J. (2012). *Sistemas de Información Gerencial*. Mexico. Obtenido de <https://juanantonioleonlopez.files.wordpress.com/2017/08/sistemas-de-informacic3b3n-gerencial-12va-edicic3b3n-kenneth-c-laudon.pdf>
- Lerdorf, R. (01 de 08 de 2009). *php*. Recuperado el 27 de 08 de 2019, de [php: https://www.php.net/manual/es/intro-what-is.php](https://www.php.net/manual/es/intro-what-is.php)
- Lopez, N. (2010). *Diseño de un Sistema de Control Interno de Inventarios en la ferreteria "LOZADA"*. Ecuador.
- Martin, G. (30 de 05 de 2020). *QUE ES XAMPP* . Obtenido de <https://www.apachefriends.org/es/about.html>
- Mendoza, M. (2016). *SSISTEMA WEB DE CONTROL DE INVENTARIOS, MANUFACTURACIÓN Y PRODUCTO FINAL PARA LA EMPRESA INDUSTRIAL COMERCIAL DE ALIMENTOS INCADEX S.R.L*. La Paz.

- Navarro, A. (2 de 11 de 2013). *redalyc.org*. Obtenido de Revision de Metodologías Ágiles para el Desarrollo de Software: <https://www.redalyc.org/articulo.oa?id=496250736004>
- Nieves G., C., Ucan P., J., & Menendez D., V. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio*. Obtenido de ResearchGate.net: [https://www.researchgate.net/profile/Juan\\_Ucan\\_Pech/publication/280580830\\_UWE\\_en\\_Sistema\\_de\\_Recomendacion\\_de\\_Objetos\\_de\\_Aprendizaje\\_Aplicando\\_Ingenieria\\_Web\\_Un\\_Metodo\\_en\\_Caso\\_de\\_Estudio/links/55bc295408aec0e5f44198ca/UWE-en-Sistema-de-Recomendacion-de-Obj](https://www.researchgate.net/profile/Juan_Ucan_Pech/publication/280580830_UWE_en_Sistema_de_Recomendacion_de_Objetos_de_Aprendizaje_Aplicando_Ingenieria_Web_Un_Metodo_en_Caso_de_Estudio/links/55bc295408aec0e5f44198ca/UWE-en-Sistema-de-Recomendacion-de-Obj)
- Olsina, L. A. (2000). *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web*. La Plata,: Facultad de Ciencias Exactas de la UNLP, Argentina.
- Orellana Funes, J. (2016). *Contabilidad Intermedia*. Bolivia: Sabiduría y Cultura.
- Pech, J. P., Guerrero, G., & Menéndez, V. H. (s.f de 06 de 2014). *Un metodo en caso de estudio*. Recuperado el 20 de 04 de 2019, de Ingeniería WEB: [https://www.researchgate.net/publication/280580830\\_UWE\\_en\\_Sistema\\_de\\_Recomendacion\\_de\\_Objetos\\_de\\_Aprendizaje\\_Aplicando\\_Ingenieria\\_Web\\_Un\\_Metodo\\_en\\_Caso\\_de\\_Estudio](https://www.researchgate.net/publication/280580830_UWE_en_Sistema_de_Recomendacion_de_Objetos_de_Aprendizaje_Aplicando_Ingenieria_Web_Un_Metodo_en_Caso_de_Estudio)
- Pressman, R. (2010). *Ingeniería del software Un enfoque práctico*. McGraw-Hill.
- Ramos, J. L. (2006). *Sistema de control de inventarios para laboratorios CRESPAL S.A. Regional Sucre*. La Paz.
- Schiaffarino, A. (12 de marzo de 2019). *Modelo cliente servidor*. Obtenido de [blog.infranetworking: https://blog.infranetworking.com/modelo-cliente-servidor/](https://blog.infranetworking.com/modelo-cliente-servidor/)
- Sommerville, I. (2005). *Ingeniería del Software (Séptima edición)*. Madrid, España: Pearson Educación S.A.
- Vergara, J. (03 de 11 de 2016). *Que es Codeigniter y cuales son algunas de sus ventajas*. Obtenido de <https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/>

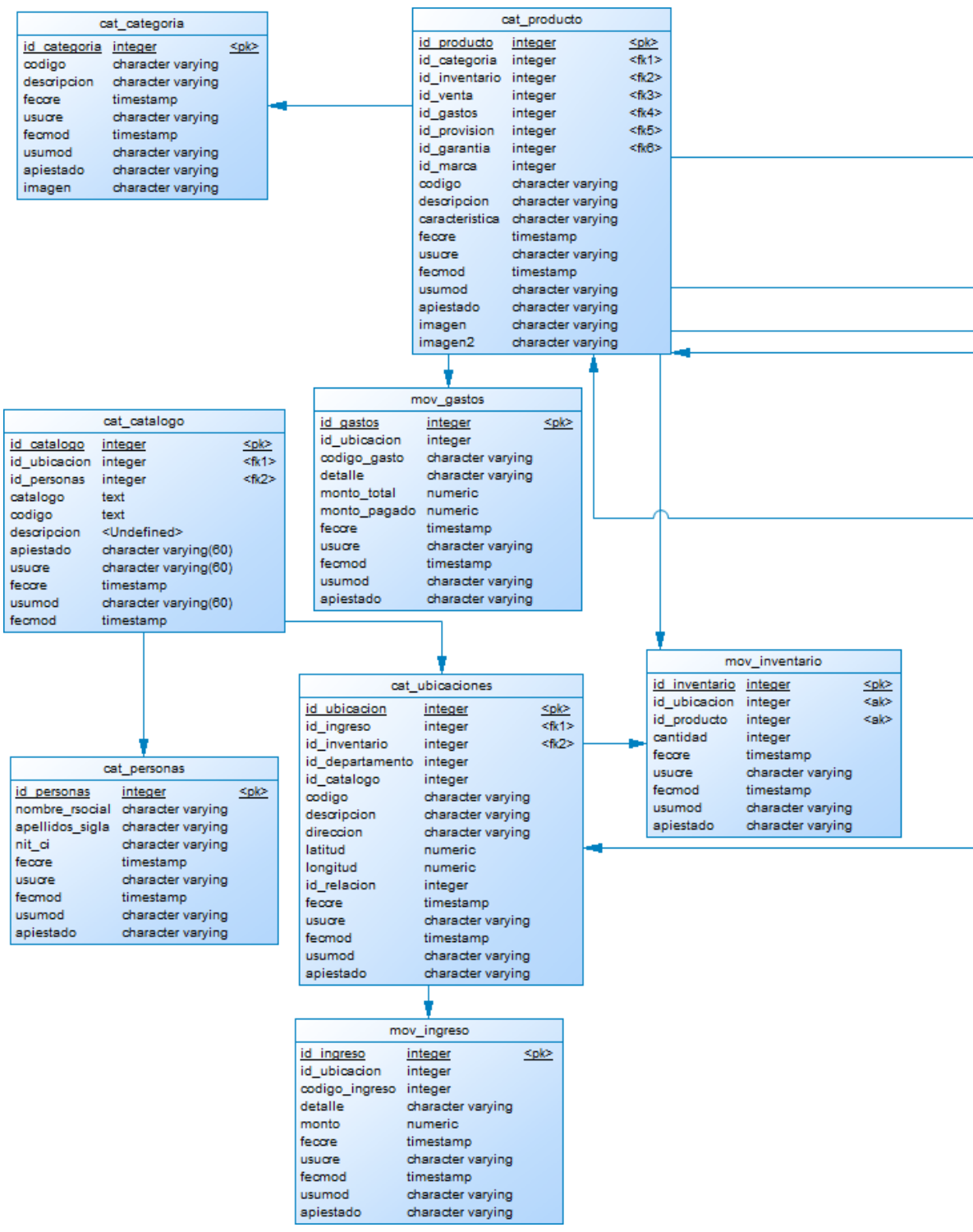
Villagómez, C. (08 de marzo de 2017). *Ciclo de vida del 'software'*. Obtenido de CCM. Obtenido de Ciclo de vida del 'software'. Obtenido de CCM:: <https://es.ccm.net/contents/223-ciclo-de-vida-del-software>

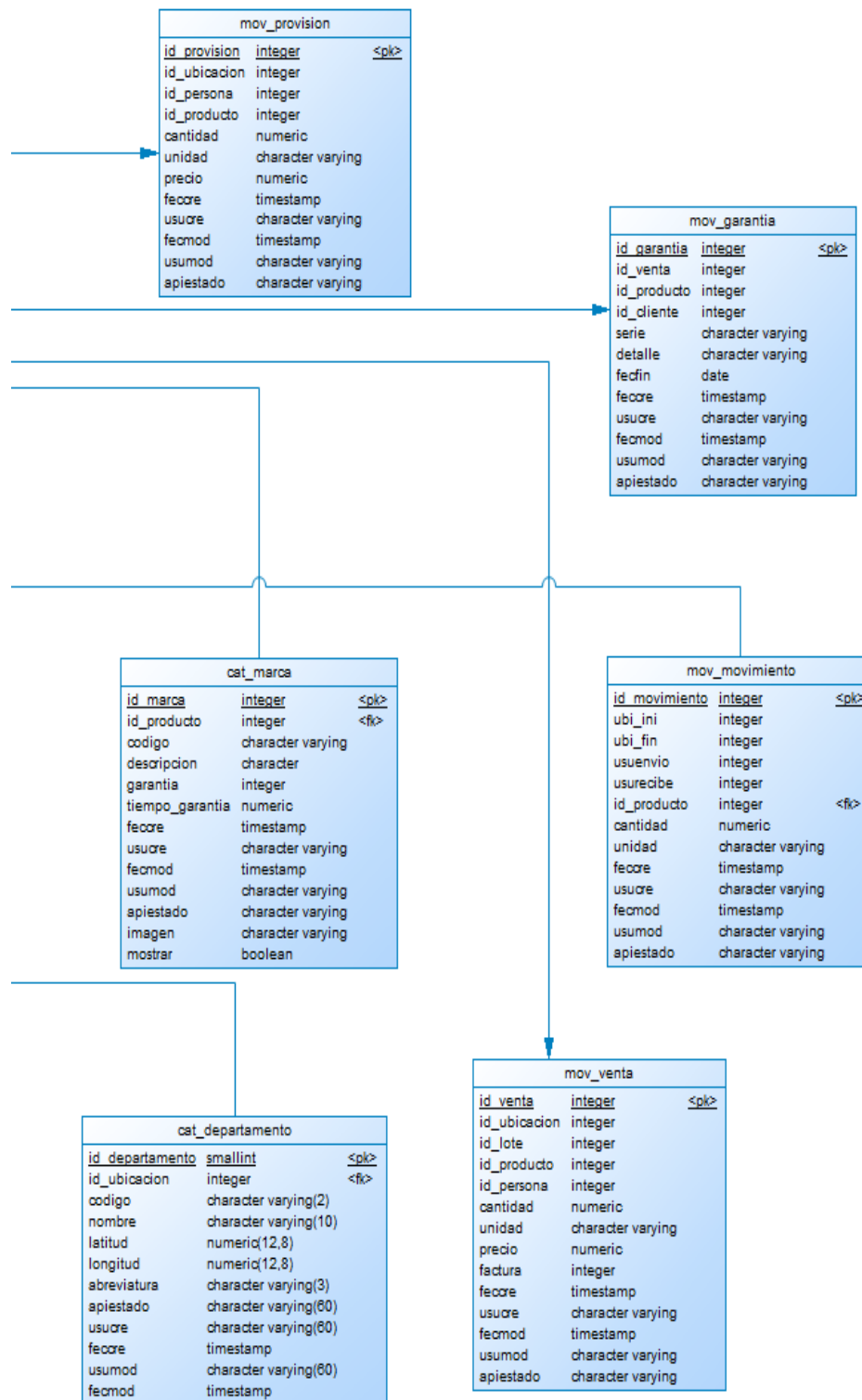
Von Bertalanffy, L. (2001). *Teoría General de los Sistemas*. Mexico. Recuperado el 2020, de [https://cienciasyparadigmas.files.wordpress.com/2012/06/teoria-general-de-los-sistemas-\\_fundamentos-desarrollo-aplicacionesludwig-von-bertalanffy.pdf](https://cienciasyparadigmas.files.wordpress.com/2012/06/teoria-general-de-los-sistemas-_fundamentos-desarrollo-aplicacionesludwig-von-bertalanffy.pdf)

Web Engineering Group. (10 de Agosto de 2016). *Tutorial sobre UWE*. Obtenido de UWE – UML-based Web Engineering: <http://uwe.pst.ifi.lmu.de/teachingTutorialSpanish.html>

# **ANEXOS**

**ANEXO A**  
**MODELO DE BASE DE DATOS**







## **ANEXO B**

# **REQUERIMIENTO DE HARDWARE Y SOFTWARE**

**Requerimientos de Hardware:** Entre los requerimientos de hardware que la empresa TESICON debe tener en cuenta es el siguiente:

<b>Requerimientos mínimos de hardware</b>	
Procesador	Dual Core, Core i3 de 2.8 GHz. superior
Memoria Ram	1 Gb
Disco Duro	500 Gb
Tarjeta de video	1 Gb
Monitor	15 pulgadas
Lectora	DVD
Teclado	Si
Mouse	Si

**Requerimiento de Software:** Entre los requerimientos mínimos de software, para la implementación del sistema se detalla lo siguiente:

<b>Requerimientos mínimos de Software</b>	
Sistema Operativo	Windows
Lenguaje de programación PHP	PHP versión 7
Sistema de gestor de Base de Datos	Postgresql
Servidor Web	Xampp (Apache)
Navegador	Firefox, Chrome, Opera(actuales)
Antivirus	Avast/Norton o similares

**ANEXO C**

**MANUAL DE USUARIO**

**TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON**



***TESICON***  
Tecnología en Sistemas de Control

## **MANUAL DE USUARIO**

**SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE  
COMPRAS VENTAS E INVENTARIOS**

**Elaborado por: Eulogia Mamani Chuca  
LA PAZ – BOLIVIA**

## INTRODUCCIÓN

En el presente manual se explicará lo necesario para poder utilizar el Sistema de Información para el Registro de Compras Ventas e Inventarios a nivel administrador.

## INGRESO AL SISTEMA

Para utilizar el sistema de control de Inventarios, se debe acceder al sistema a través de estos navegadores de internet.

Mozilla Firefox.



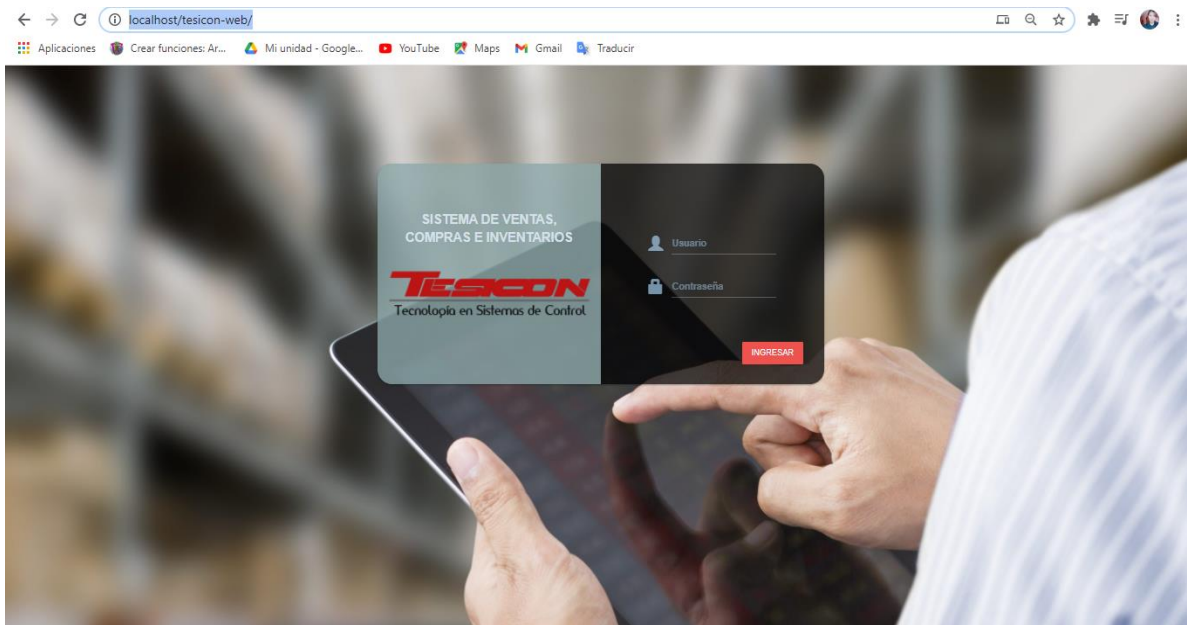
Google Chrome.



Las mismas deben ser de versiones actuales, ya que el sistema requerirá características actuales.

Para ingresar al sistema se debe digitar la dirección <http://localhost/tesicon-web/> al dar click en la dirección , se podrá apreciar la página de inicio del sistema tal como se muestra en la siguiente figura.

### Pantalla de Inicio



### Descripción de la ventana de Página de Inicio

## Ingresar Usuario

### Pantalla de Login; Ingresar Usuario



## Contraseña

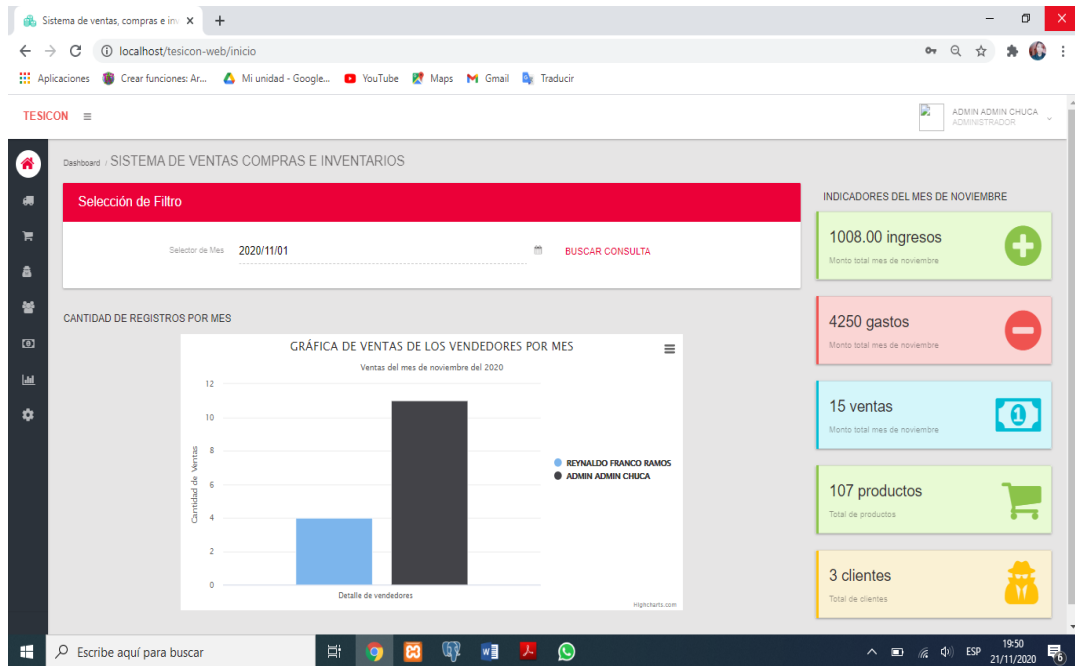
### Pantalla de Login; Ingresar Contraseña



**Nota:** En caso de que no cuente con un usuario y contraseña de acceso al sistema, debe contactarse con el usuario administrador de dicha entidad.

Si cuenta con un usuario y contraseña correcta debe ingresar, y la página cambiará y mostrará los datos del usuario que tiene iniciada a sesión, tal como se muestra en la figura.

## Página de Inicio del Sistema



## Página de Administración

Una vez iniciado el sistema, se presenta la pantalla del módulo de administrador. Se puede realizar la Búsqueda de acuerdo al mes que requiera buscar la cantidad de Registros por mes.

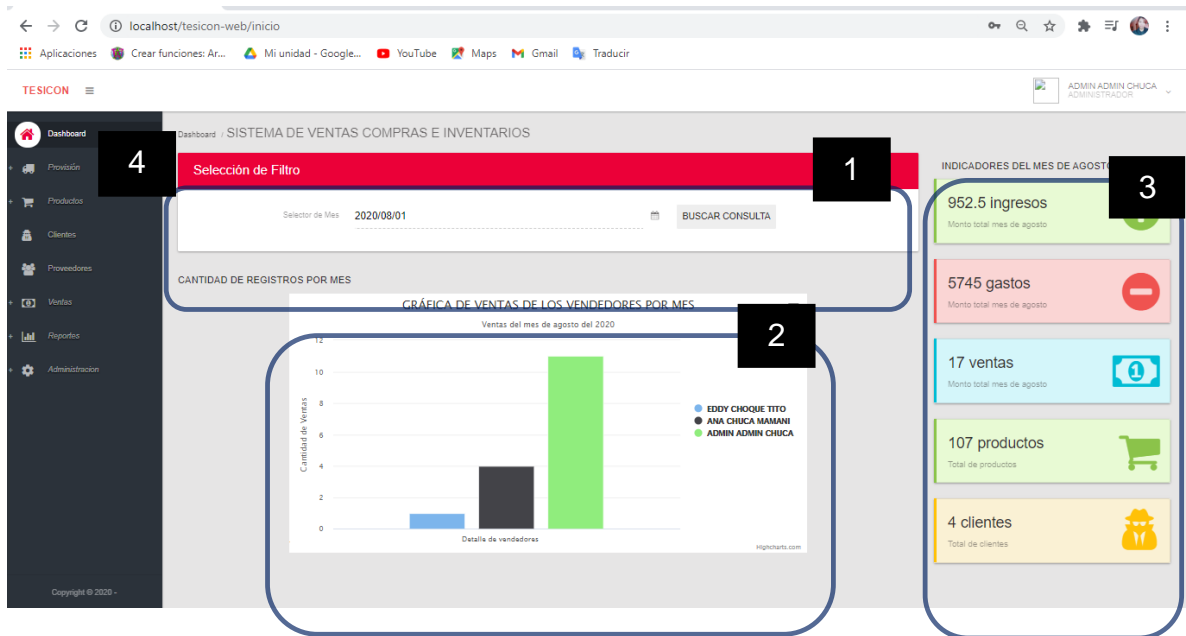
Grafica de Venta de los Vendedores por mes.

Indicadores del mes se puede ver el registro de Clientes, cuantos existe Productos. Ventas Realizadas, Gastos realizados.

Se muestra los módulos para cada usuario con distinto rol.

En la siguiente figura se muestra la pantalla de inicio del lado izquierdo todas las operaciones que realiza el sistema, según se el tipo de rol que se designe a cada usuario, restringiendo a su acceso a usuarios con un rol distinto al administrador

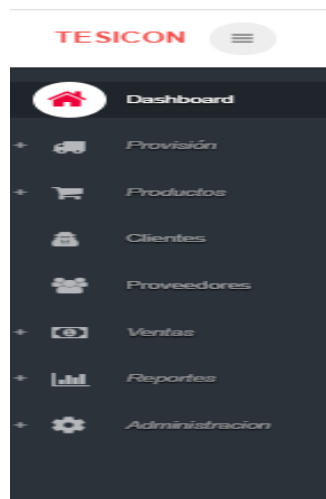
## Datos del Registro



### Descripción de los menús

Se muestra el menú de todas las operaciones que realiza el sistema, según el rol de usuario.

### Menú donde el Usuario Realiza Consultas



Menu registro de Usuarios: en el menu registro de usuarios estan los sub menus: Registro de Usuario y Asignacion de Rol.

En la siguiente figura se hara el registro de Usuario llenando todos los campos



## Registro de Usuario

Registro de Personal

Nombre

Apellido Paterno

Apellido Materno

Nro. Carnet de Identidad

Expedido

Género

Fecha de Nacimiento

Dirección

Teléfono

Correo Institucional

Seleccione Ubicación de Trabajo

Cargo

MODIFICAR USUARIO REGISTRAR USUARIO

En la siguiente figura se muestra los datos ingresados de Registro de Personal, Detallando todos los usuarios Registrados

## Pantalla Detalle de Usuarios Registrados

Detalle de Usuarios

Mostrar 10 filas por página

Nº	Login	Nombre Completo	Carnet de Identidad	Correo Institucional	Teléfono	Ubicación de Trabajo	Acción
1	admin.admin	ADMIN-ADMIN-CHUCA	4927950 LPZ	eylnchuca@gmail.com	89795345	TIENDA	
2	ana.chuca	ANA-CHUCA-MAMANI	2180975 LPZ	marachuca@gmail.com	7320980	ALMACEN	
3	eddy.choque	EDDY-CHOQUE-TITO	8050388 LPZ	eddy.heberson.ing@gmail.com	74009001	TIENDA	
4	reynaldo.franco	REYNALDO-FRANCO-RAMOS	72579337 LPZ	reys@gmail.com	72081143	TIENDA	
5	santos.lima	SANTOS-LIMA-HUANCA	2020202 CBB	svuente@gmail.com	2131654	ALMACEN	

Mostrando página 1 de 1

Después de haber llenado datos del personal de la empresa se asignará un rol, realizando primero la búsqueda del usuario elegido.

## Buscar Usuario

Asignación de Rol

Usuario

Seleccione Usuario

Q

Asignamos un rol, esto lo realiza el administrador.

## Asignación de Rol

Administración | Usuarios

PERSONAL TESICON

Registro de Usuario

Asignación de Rol

### Asignación de Rol

Usuario  
EDDY CHOQUE TITO

Nombre: EDDY      Apellido Paterno: CHOQUE      Apellido Materno: TITO

Carnet de Identidad: 6050358 LPZ      Login Usuario: eddy.choque      Permiso:

ADMINISTRADOR  
ENCARGADO DE VENTAS  
ENCARGADO DE ALMACENES

### Detalle de Usuarios

Mostrar: 10 filas por página

ID	Login	Nombre Completo	Carnet de Identidad	Permiso	Estado	Accion
1	admin.admin	ADMINADMIN CHUCA	4927850	ADMINISTRADOR	ELABORADO	
2	eddy.choque	EDDY CHOQUE TITO	6050358	ENCARGADO DE VENTAS	ELABORADO	

## Modulo Ventas

Ingresamos al sistema con el rol de encargado de ventas

## Pantalla principal Encargado de Ventas



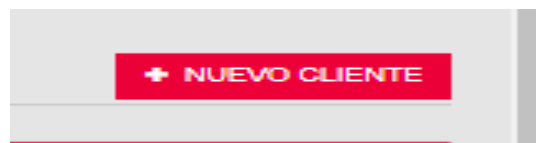
Para realizar una venta primero debemos registrar al Cliente

### Registrar Cliente



Para registrar un Nuevo cliente debemos hacer click en el botón como se muestra en la figura.

### Agregar Nuevo Cliente



Llenamos los datos Nombre, Apellido y Carnet de Identidad y hacemos click en registrar Usuario.

## Registrar Cliente

Registrar Cliente

Nombres  
MARISOL

Apellidos  
ARGUEDAS

Carnet de Identidad (CI)  
98765433

MODIFICAR CLIENTE REGISTRAR CLIENTE

En el Menu de Ventas realizamos Pedido luego buscamos el producto , aparecera una ventana indicando el Stock que hay en inventario tanto en Almacen como en la Tienda tambien desplegara el costo del producto, el usuario debera solicitar la cantidad que nesecite y posteriormente aparecera el costo total del producto luego hacemos click en Agregar Producto a pedido.

## Pantalla Formulario de Venta y Stock en inventario y Agregar Pedido

TESICON

REYNALDO FRANCO RAMOS  
Administrador de Ventas

Ventas / Pedidos

Selección de Filtro

Productos  
(9) DISCO DURO WESTERN BLANCO 1TB - HDD-WESTWHITE1TB

Cantidad en Almacen: 5

Cantidad en Tienda: 9

Precio ref. por unidad (Bs.): 315.00

Cantidad:

Precio Total:

AGREGAR PRODUCTO A PEDIDO

Mostrar: 10 filas por página

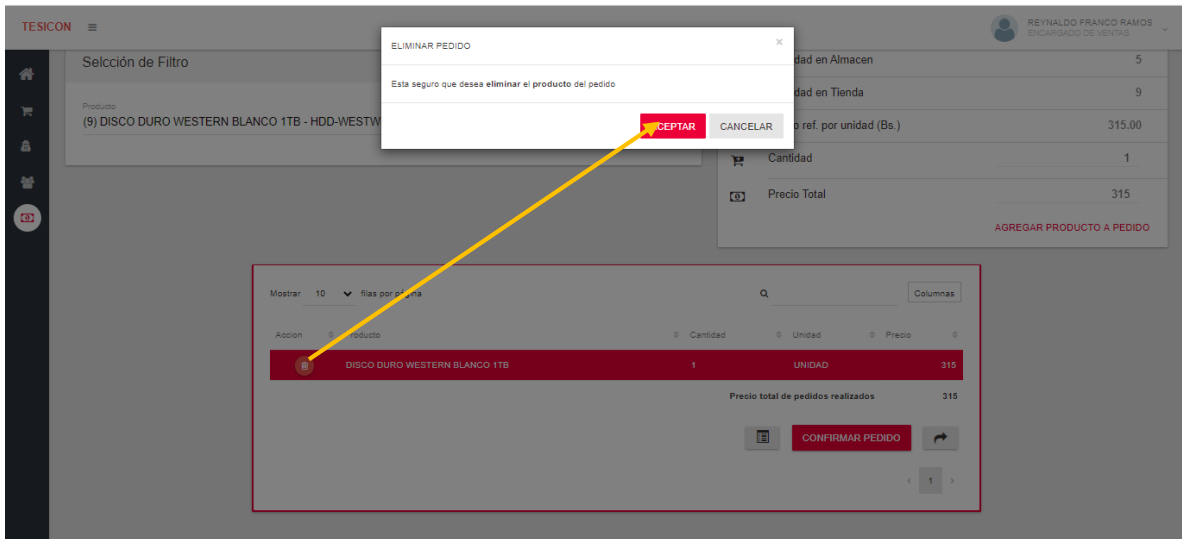
Acción	Producto	Cantidad	Unidad	Precio
	DISCO DURO WESTERN BLANCO 1TB	1	UNIDAD	315

Precio total de pedidos realizados: 315

CONFIRMAR PEDIDO

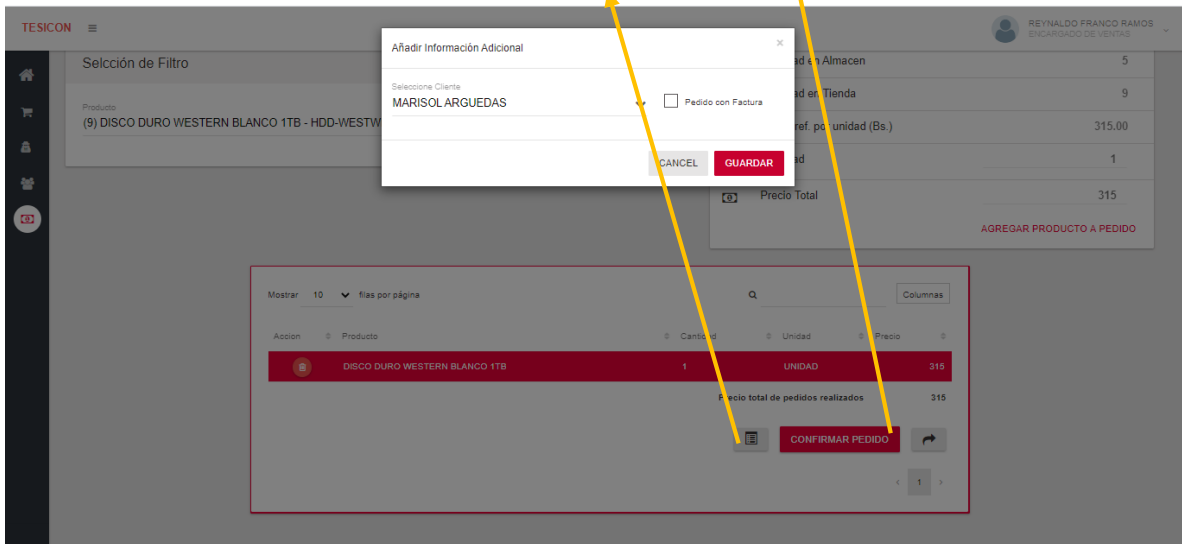
Si la Venta no se realiza se debe Eliminar pedido click en aceptar, si procediera a la venta del producto click en Cancelar.

## Eliminar Pedido de Venta



Agregamos el cliente click en Guardar y por ultimo realizamos un click en Confirmar pedido.

## Agregar Cliente y Confirmar Pedido



En el siguiente menu confirmamos Venta, lo que nos mostrara el numero de Pedidos realizados el nombre del Cliente, el tipo de Producto, y por ultimo nos detallara por quien fue vendido.

## Confirmar Venta del Producto

TESISICON

REYNALDO FRANCO RAMOS  
ENCARGADO DE VENTAS

Venta / Confirmación de Venta

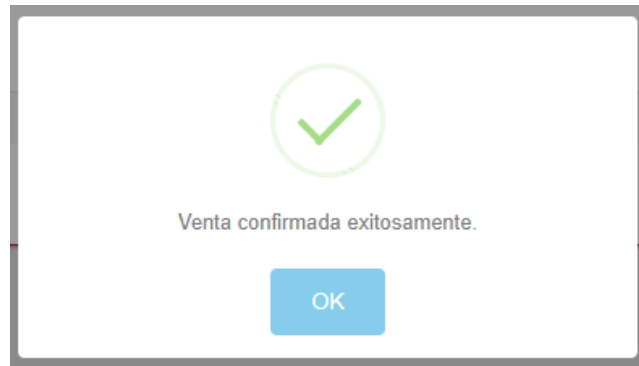
Detalle de pedidos realizados

Mostrar: 10 filas por página

N°	Cliente	Productos	Vendido por	Acción
Pedido 1	MARISOL ARGUEDAS	DISCO DURO WESTERN BLANCO 1TB	reynaldo franco	OK

Confirmar Venta

Ventana confirmada exitosamente realizamos click en ok  
Venta Confirmada Exitosamente



## Modulo Proveedores

En este módulo se registrará los datos del registro de Proveedores insertando los datos de Razón Social o nombre, sigla, y número de Nit. Hacemos Click en el Botón Nuevo Proveedor.

También tenemos la opción de Editar y Eliminar el Registro de Proveedor.

## Registro de Proveedores

Proveedor / Proveedores

Listado de Proveedores

+ NUEVO PROVEEDOR

Registrar Proveedor

Razón social: DAHUATECHNOLOGY

Sigla: DAHUA

NIT: 34567898765

MODIFICAR PROVEEDOR REGISTRAR PROVEEDOR

N°	Razón Social	Sigla	NIT	Estado	Acción
1	CASA SEGURA	CASA SEGURA	1234565015	ELABORADO	[+][-][X]
2	ALIBABA	ALIBABA	2345412017	ELABORADO	[+][-][X]
3	XTREME - COMERCIAL	XTREME - COMERCIAL	10858789430188	ELABORADO	[+][-][X]

## Módulo de Productos

Para realizar el nuevo producto debemos hacer click en el botón nueva categoría, luego registramos el nombre de la nueva categoría y Código, seleccionamos una imagen y para finalizar hacemos click en botón Registrar Categoría

## Nuevo Categoría

Productos / Categorías

Listado de Categorías

+ NUEVA CATEGORÍA

Registrar Categoría

Categoría: CAMARAS HD 1080P10MP

Codigo: 05-1

SELECCIONAR IMAGEN

MODIFICAR CATEGORÍA REGISTRAR CATEGORÍA

N°	Codigo	Categoría	Estado	Acción
----	--------	-----------	--------	--------

Desplazamos el registro de **+ NUEVA MARCA** llenamos los datos de tipo de marca código y garantía posteriormente seleccionamos imagen de la marca y por ultimo hacemos click en Registrar Marca.

## Registro de Marcas

The screenshot shows the 'Registro de Marcas' form in the TESICON system. The form is titled 'Registrar Marca' and is set against a red header. It contains the following fields and options:

- Marca:** ALHUA
- Código:** 016
- Garantía:** Radio buttons for 'Sin Garantía' (selected) and 'Con Garantía en Bolivia'.
- Tiempo de Garantía:** A field for entering the duration in years.
- Image Selection:** A red-bordered box containing the 'ALHUA TECHNOLOGY' logo, with a 'SELECCIONAR IMAGEN' button below it.
- Buttons:** 'MODIFICAR MARCA' and 'REGISTRAR MARCA' buttons are located at the bottom right of the form.

Desplazamos el registro de **+ NUEVO PRODUCTO** llenamos los datos categoría, seleccionamos la marca, el tipo de producto y su código. Luego hacemos click en el botón **REGISTRAR PRODUCTO**

## Registro de Producto

The screenshot shows the 'Registro de Producto' form in the TESICON system. The form is titled 'Registrar Producto' and is set against a red header. It contains the following fields and options:

- Selección de Categoría:** CAMARAS HD 720P 1MP
- Selección de Marca:** HIKVISIÓN
- Producto:** KIT DE CAMARAS XVR : 4 CAMA
- Código:** XVR2006
- Características:** KIT XVR: 4 CÁMARAS METÁLICAS, BALAS 2MP IR 20 MT. 1 XVR DE 4 CANALES DE 1080P. 4 CABLES BNC DE 18 MT. 1 POWER SPLITER DE 5 CANALES. 1 FUENTE DE 12V 5A. DISCO DURO DE 1TB. ROUTER INALAMBRICO
- Image Selection:** A red-bordered box containing an image of a security camera kit, with a 'SELECCIONAR IMAGEN' button below it.
- Buttons:** 'MODIFICAR PRODUCTO' and 'REGISTRAR PRODUCTO' buttons are located at the bottom right of the form.



También se podrá editar y eliminar el producto como también se podrá visualizar la imagen con los siguientes iconos.



### Mostrar Lista de Productos

Categoría	Marca	Código	Producto	Características	Estado	Acción
OTROS	SIN MARCA	ADP-L2DA	FUENTE DE PODER 12VDC	FUENTE DE PODER 12VDC X 2A 11 CANAL DE SALIDA 12VDC X 2A 10PORTS CÁMARA CONTROL DE ACCESO CHARRIS ENTRADA 1219220 V AC	ELABORADO	
OTROS	SIN MARCA	BSD-3011	Contacto magnético para proteger contornos	Contacto magnético para proteger contornos metálicos, por su forma ideal para adosarse en piso, dimensiones 110X14mm	ELABORADO	
CÁMARA HD ZOOM 18MP	LONGSE	LEB24THC1008	Cámara Bala Exterior 18MP HDTVI	Cámara Bala Exterior 18MP HDTVI, 1/4" Sensor CMOS OV, ZOOM180X, Dia/Niche, 0.01 Lux@1.2, Control OSD, IR LED 14x 18PCS, IR 20M, IP66, Lente fijo 3.6mmF2.0, Salida de Video TVCV/AV/HD/Analog	ELABORADO	
CÁMARA HD ZOOM 18MP	LONGSE	LEB24THC1008	Cámara Bala Metálica 18MP HDTVI	Cámara Bala Metálica 18MP HDTVI, 1/4" Sensor CMOS OV, ZOOM180X, Dia/Niche, 0.01 Lux@1.2, Control OSD, IR LED 14x 18PCS, IR 20M, IP66, Lente fijo 3.6mmF2.0, Salida de Video TVCV/AV/HD/Analog	ELABORADO	
CÁMARA DE 1.3 MP	DAHUA	PRD-01	PRODUCTO	SIN DESCRIPCIÓN	ELABORADO	
KITS DE 4 CÁMARA	LONGSE	WFD20MP10100	KIT WFI	KIT WFI, 4 Cámaras IP ZOOM Helicon, WFI hasta 100 metros en 4dirección, Metálica, IP66, IR 20M, Video noturno, ONVIF, 1 NVR WFI, 1 80TA hasta 1TB, 4 RJ45 10/100 Mbps, Network PoE 1 HDR, 1 VGA, 2 USB Backup/Mouse, 4 Power Adapter 12V, Disco 8cm de 1TB	ELABORADO	
HERRAMIENTAS	PROTIO	MW1-3	MANTILLO DE MADERA	HERRAMIENTA DE PERFORACION UTILIZADA PARA GOLPEAR DIRECTAMENTE	ELABORADO	
ACCESORIOS	LONGSE	PTM800-E	Video Bala HD 1 Canal Power TVCV/AV/HD 18MP -2MP	Video Bala HD 1 Canal Power TVCV/AV/HD 18MP -2MP	ELABORADO	
CÁMARA HD 180P 2MP	ZKTECO	LPC24THC200V	Cámara Dome Metálica 2.1MP HDTVI	Cámara Dome Metálica 2.1MP HDTVI, 1/2.7" Sensor CMOS OV, 180P/90X, Fibra PLGUT, Dia/Niche, Luminancia 0.01 Lux@1.2, ONVR, DVR, LUTO, Control OSD, IR LED 10x 24PCS, IR 20M, IP66, Lente fijo 2.8mm, Salida de Video TVCV/AV/HD/Analog	ELABORADO	
CREBENCIALES	HOBSTON	456453	CREBENCIALES	DURABILIDAD Y 50'12	ELABORADO	

### Módulo de Proveedor

Seleccionamos el proveedor que realizara el abastecimiento el producto que se desea adquirir, la cantidad y el precio por unidad, y por ultimo seleccionamos el destino de la provisión en almacén o en tienda y por ultimo click en **AGREGAR PRODUCTO(S) A LA COMPRA**

### Registro de Abastecimiento

Registro de Abastecimiento

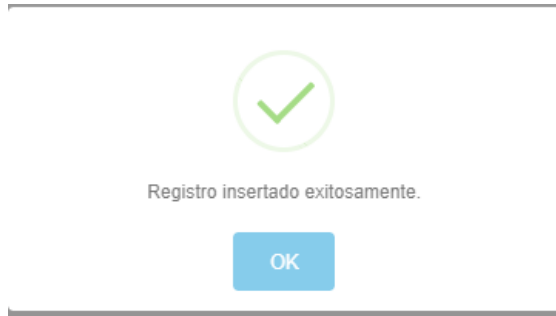
Selección Proveedor	Selección Producto	Cantidad
CENTER INFORMATIC CENTINF x v	(T8-A-5) DISCO DURO WESTERN BLANCO 1TB-HDD- x v	25


---

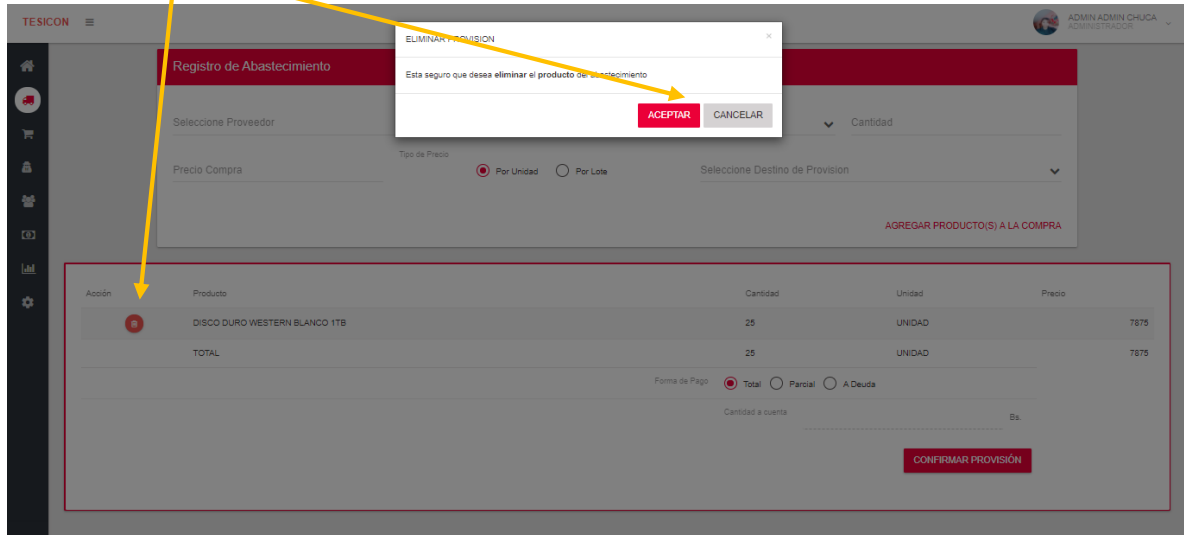
Precio Compra	Tipo de Precio	Selección Destino de Provisión
315	<input checked="" type="radio"/> Por Unidad <input type="radio"/> Por Lote	ALMACEN - ALMACEN x v

AGREGAR PRODUCTO(S) A LA COMPRA

## Registro insertado exitosamente



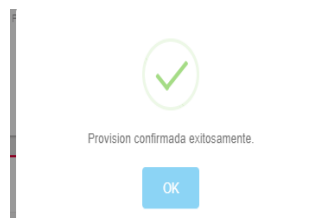
Cuando  no deseamos el producto realizamos la accion eliminar y click en aceptar



Y por último click en el botón

**CONFIRMAR PROVISIÓN**

## Provisión Confirmada exitosamente



## Módulo Reportes

**Sub menu reporte de Venta por Día** nos muestra de que fecha se desea buscar y si el reporte será de tienda o del almacén, también se detallará que usuario vendedor, el producto, la cantidad y el costo del producto. Por último el usuario podrá ver en formato pdf.

### Menú Reporte de Venta por Día

The screenshot displays the 'Reporte de Venta por Día' interface. On the left is a dark sidebar with navigation items: Dashboard, Proveedor, Productos, Clientes, Proveedores, Ventas, Reportes (selected), and Administracion. The 'Reportes' section is expanded to show 'Reporte de Ventas', 'Reporte de Ventas General', and 'Reporte de Inventarios'. The main content area is titled 'Reportes / Ventas' and contains a 'Selección de Filtro' box. This box has a 'Fecha' field set to '2020/11/25' and a 'Ubicación de Trabajo' dropdown set to 'TIENDA'. A yellow arrow points to a search icon in the filter box. Below the filter is a table with the following data:

N°	VENDEDOR	PRODUCTO	CANTIDAD	PRECIO	COSTO	UTILIDAD BRUTA	FECHA
1	reytalo franco	KIT.XVR: 4 Cámaras metálicas	1	215	200.00	15.00	2020-11-25
2	admin.admin	DISCO DURO WESTERN BLANCO 1TB	1	330	315.00	15.00	2020-11-25
3		TOTAL				30.00	2020-11-25

At the top of the main content area, there is a header with the TESICON logo, 'EMPRESA TESICON', and 'REPORTES DE VENTAS POR DÍA'. The user information 'Usuario: admin.admin' and 'Fecha: 2020-11-25 17:38:46' is displayed in the top right corner of the main content area.

Sub menú Reporte de Venta en general se mostrará por fecha inicio y fecha fin también se podrá buscar por ubicación Tienda o Almacén.

Reportes / Ventas General

Selección de Filtro por Fechas

Desde: 2020/09/01 Hasta: 2020/11/25

Ubicación de Trabajo

TIENDA

ALMACEN

TIENDA

EMPRESA TESICON  
TECNOLOGÍA EN SISTEMAS DE CONTROL

Reporte de Ventas General

Usuario: admin.admin  
Fecha de Reporte: 2020-11-25  
Desde: 2020/09/01  
Hasta: 2020/11/25

Nº	VENDEDOR	PRODUCTO	CANTIDAD	PRECIO	COSTO	UTILIDAD BRUTA	FECHA
1	reyinaldo franco	TECLADO DELUX MULTIMEDIA CONECTOR USB	1	26	5.00	20.00	2020-11-20
2	admin.admin	Grabador XVR de 15CH-4IN-1	1	170	150.00	20.00	2020-11-20
3	reyinaldo franco	NIT XVR: 4 Cámaras metálicas	1	215	200.00	15.00	2020-11-25
4	reyinaldo franco	TECLADO DELUX MULTIMEDIA CONECTOR USB	1	26	5.00	20.00	2020-11-20
5	reyinaldo franco	TECLADO DELUX MULTIMEDIA CONECTOR USB	1	26	5.00	20.00	2020-11-20
6	admin.admin	DISCO DURO WESTERN BLANCO 1TB	1	330	315.00	15.00	2020-11-25
7	reyinaldo franco	DISCO DURO WESTERN BLANCO 1TB	1	315	315.00	0.00	2020-11-22
8	admin.admin	MOUSE DELUX CONECTOR USB	2	140	100.00	40.00	2020-11-19

Reporte de Inventarios nos mostrara si queremos buscar por ubicación por tienda, por almacén.

### Reporte de Inventarios

Reportes / Inventarios

Selección de Filtro

Ubicación de Trabajo

TIENDA

ALMACEN

TIENDA

EMPRESA TESICON  
TECNOLOGÍA EN SISTEMAS DE CONTROL

Reporte de Inventarios

Usuario: admin.admin  
Fecha: 2020-11-25 17:49:08

Nº	CODIGO	PRODUCTO	PRECIO	UBICACIÓN	CANTIDAD
1	232702-12	Fuente, Entrada: Ac 110-240v, Salida: Dc 12v 2a	60	Tienda	5
2	410014-2	Cable Hdmi Engomado De 2 Mt Dts Full Hd Alta Velocidad	35	Tienda	5
3	410014-5	Cable Hdmi Engomado De 5 Mt Dts Full Hd Alta Velocidad	65	Tienda	8
4	410020-1.5	Cable Hdmi Nylon Trenzado De 1.5 Mt Dts Full Hd Alta Velocidad	27	Tienda	7
5	410020-1.8	Cable Hdmi Nylon Trenzado De 1.8 Mt Dts Full Hd Alta Velocidad	30	Tienda	0
6	410020-3	Cable Hdmi Nylon Trenzado De 3 Mt Dts Full Hd Alta Velocidad	40	Tienda	5
7	5V MINI UPS	Mini Ups 5vdc 1a Para Biométricos	240	Tienda	5
8	ABS-UTPHD	Video Balun Hd 1 Canal Pasivo TwiCvIAHD 1mp - 2mp	35	Tienda	5
9	ADR-12VZA	Fuente De Poder 12vdc	60	Tienda	5
10	BRACKET	Bracket De Metal Largo 18cm	40	Tienda	5

Esto se podrá mostrar en formato pdf

# Reporte de Inventarios en Formato PDF

1 / 4

## EMPRESA TESICON

Usuario: admin.admin  
Fecha: 2020-11-25 17:49:19

### REPORTE DE INVENTARIOS

N°	Codigo	Producto	Precio	Ubicación	Cantidad
1	232702-12	Fuente, Entrada: Ac 110-240v ; Salida: De 12v 2a	60	Tienda	5
2	410014-2	Cable Hdmi Engomado De 2 Mt Dis Full Hd Alta Velocidad	35	Tienda	5
3	410014-5	Cable Hdmi Engomado De 5 Mt Dis Full Hd Alta Velocidad	65	Tienda	6
4	410020-1.5	Cable Hdmi Nylon Trenzado De 1.5 Mt Dis Full Hd Alta Velocidad	27	Tienda	7
5	410020-1.8	Cable Hdmi Nylon Trenzado De 1.8 Mt Dis Full Hd Alta Velocidad	30	Tienda	0
6	410020-3	Cable Hdmi Nylon Trenzado De 3 Mt Dis Full Hd Alta Velocidad	40	Tienda	5
7	SV MINI UPS	Mini Ups Svdc 1a Para Biométricos	240	Tienda	5
8	ABS-UTPHD	Video Balun Hd 1 Canal Pasivo Tv/Cvi/Ahd Imp - 2mp	35	Tienda	5
9	ADP-12V2A	Fuente De Poder 12vdc	60	Tienda	5
10	BRACKET	Bracket De Metal Largo 18cm	40	Tienda	5
11	BS-2021WH	Contacto Magnético Para Blindex Color Blanco	17	Tienda	5
12	BSD-3011	Contacto Magnético Para Proteger Cortinas	180	Tienda	5
13	CAJA 8X10X3	Caja Metálica Para Paneles Paradox	129	Tienda	5
14	CAJA PVC	Caja Sobrepuestas Con Tapa Ciega	11	Tienda	5

# DOCUMENTACIÓN

El Alto, 23 de noviembre de 2020

Señor  
**Ing. David Carlos Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado “**SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS CASO: TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON**”, elaborado por la universitaria: **MAMANI CHUCA EULOGIA**, con cedula de identidad: **4927650 L.P.** y registro universitario: **20025302**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



**Ing. Marisol Arguedas Balladares**  
**TUTOR METODOLÓGICO**

El Alto, 23 de noviembre de 2020

Señor  
**Ing. David Carlos Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado “**SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS CASO: TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON**”, elaborado por la universitaria: **MAMANI CHUCA EULOGIA**, con cedula de identidad: **4927650 L.P.** y registro universitario: **20025302**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



Ing. Sergio Ramiro Rojas Saire  
TUTOR REVISOR



El Alto, 23 de noviembre de 2020

Señor  
**Ing. David Carlos Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

**REF.: AVAL DE CONFORMIDAD**

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado “**SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS CASO: TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON**”, elaborado por la universitaria: **MAMANI CHUCA EULOGIA**, con cedula de identidad: **4927650 L.P.** y registro universitario: **20025302**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



Ing. Santos Aurelio Limachi Huanca  
TUTOR ESPECIALISTA

El Alto, 20 de noviembre de 2020

Señor  
**Ing. David Carlos Mamani Quispe**  
**DIRECTOR**  
**CARRERA INGENIERÍA DE SISTEMAS – UPEA**

Presente. -

## CERTIFICA

Mediante la presente tengo a bien comunicarle mi conformidad con el proyecto de grado que lleva como título **“SISTEMA DE INFORMACIÓN PARA EL REGISTRO Y CONTROL DE COMPRAS, VENTAS E INVENTARIOS CASO: TECNOLOGÍA EN SISTEMAS DE CONTROL - TESICON”**, desarrollado por la Univ. EULOGIA MAMANI CHUCA, con C.I.: 4927650 L.P., que ha concluido satisfactoriamente el desarrollo del sistema, cumpliendo los requerimientos de la empresa.

En este sentido, corresponde emitir el Aval de Conformidad del mismo, para su defensa pública y evaluación correspondiente.

Sin otro particular, reciba usted saludos cordiales.

Atentamente.

  
**Reynaldo Franco Ramos**  
GERENTE GENERAL  
**TESICON**  
Tecnología en Sistemas de Control



Cc/archivo

[www.tesiconbolivia.com](http://www.tesiconbolivia.com)

Plaza Alonso de Mendoza, Edif. Santa Anita, Piso 2 Of. 202

☎ 2 457838

📞 72575237

✉ [tesicon@live.com](mailto:tesicon@live.com)