

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

“SISTEMA WEB DE GESTIÓN Y CONTROL DE ALMACENES”

**CASO: Unidad de servicios generales y mantenimiento (GAMEA)**

Para optar al título de Licenciatura en Ingeniería de Sistemas

**MENCION: INFORMATICA Y COMUNICACIONES**

**Postulante:** Heriberto Condori Cahuapaza  
**Tutor Metodológico:** Ing. Marisol Arguedas Balladares  
**Tutor Especialista:** Ing. David Limachi Callizaya  
**Tutor Revisor:** Ing. Yolanda Escobar Mancilla

LA PAZ – BOLIVIA

2020

## **DEDICATORIA**

*Dedicarlo este trabajo con mucho cariño:*

*A mi papito querido Emilio que me guía desde el cielo, a quien extraño mucho y que me enseñó el esfuerzo y trabajo tiene sus recompensas. A mi mamita querida Fermína por su apoyo, cariño y comprensión. A mi hermano Carlos por su compañía en los buenos momentos y su cuidado en momentos difíciles.*

## **AGRADECIMIENTOS**

Agradecer a todas aquellas personas quienes han tenido que ver en el desarrollo y conclusión de este proyecto.

Agradecer con mucho afecto a mis distinguidos tutores, a mi tutor metodológico Ing. Marisol Arguedas Balladares por su conocimiento, apoyo, confianza, tiempo, persistencia, paciencia y motivación que me brindo.

A mi tutor especialista Ing. David Limachi Callizaya por compartir su conocimiento, brindarme su orientación, sugerencias con paciencia y motivación durante el desarrollo del presente proyecto y sobre todo su amistad y compañerismo.

A mi tutor revisor ing. Yolanda Escobar Mancilla por su disponibilidad de tiempo, por orientación y las observaciones bridadas por su amplio conocimiento en la realización de proyectos.

Por ultimo a mis compañeros que gracias a su compañía, amistad y ánimos pude terminar una etapa más de formación académica.

## **RESUMEN**

*El presente proyecto se desarrolló para los almacenes de la unidad de servicios generales y mantenimiento, dependiente de la dirección administrativa del Gobierno Autónomo Municipal de El Alto, en donde se identificó problemas para el buen funcionamiento de las actividades y control de productos de los almacenes.*

*En los almacenes de la unidad de servicios generales, se tiene un gran y constante movimiento de información, debido a la cantidad y variedad de materiales, es difícil tener un dato actual de la cantidad de material que cuenta cada almacén y los reportes que solicitan los encargados de las solicitudes y por consiguiente el jefe de unidad.*

*Es así que se plantea desarrollar e implementar un sistema de gestión y control de almacenes, para que ayude en el control adecuado y en la optimización de tiempo para registro, control y reportes solicitados. Aplicando la metodología UWE, lenguaje de programación PHP, y gestor de base de datos MySQL, herramientas de frond-end javascript, Ajax, jquery, bootstrap, css3, entre otros.*

*Así mismo para definir la calidad del sistema se tomó en cuenta el ISO 9126.*

*Finalmente, el análisis de costos utilizado es el COMOMO II en donde se puede reflejar el costo del sistema mediante fórmulas ya establecidas*



## ÍNDICE

CAPITULO I MARCO PRELIMINAR .....	
1.1 INTRODUCCIÓN .....	1
1.2 ANTECEDENTES .....	2
1.3 PLANTEAMIENTO DEL PROBLEMA .....	3
1.3.1. Problema Principal .....	3
1.3.2. Problemas secundarios .....	4
1.4. OBJETIVOS .....	4
1.4.1 Objetivos Generales .....	4
1.4.2. Objetivos Específicos .....	4
1.5. JUSTIFICACIÓN .....	5
1.5.1 Justificación Técnica .....	5
1.5.2. Justificación Económica .....	5
1.5.3. Justificación Social .....	6
1.6. METODOLOGÍA .....	6
1.7. HERRAMIENTAS .....	7
1.8. LIMITES Y ALCANCES .....	9
1.8.1. Limites .....	9
1.8.2. Alcances .....	9
1.9. APORTES .....	10
CAPITULO II MARCO TEÓRICO .....	
2.1. Introducción .....	11
2.2. UNIDAD DE SERVICIOS GENERALES Y MANTENIMIENTO (GAMEA) .....	11
2.3 SISTEMA .....	14
2.4. Web .....	15

2.4.1 SISTEMA Web .....	16
2.5. GESTIÓN Y CONTROL DE ALMACÉN .....	16
2.6 MÉTODOS DE VALORACIÓN DE INVENTARIOS .....	19
2.6.1 Primero en entrar primero en salir- PEPS .....	19
2.7 INGENIERÍA DE SOFTWARE .....	20
2.7.1 Modelos de desarrollo de software .....	20
2.7.1.1 Metodologías tradicionales .....	24
2.7.1.2 Metodologías ágiles .....	24
2.8 METODOLOGÍA DE DESARROLLO UWE .....	26
2.8.1 Fases de la Metodología UWE .....	27
2.9 ARQUITECTURA DEL SOFTWARE .....	34
2.9.1 Patrón Modelo Vista Controlador .....	36
2.10 HERRAMIENTAS DE DESARROLLO .....	39
2.10.1 Gestor de base de datos .....	39
2.10.2 Lenguaje de programación .....	42
2.10.3 Framework .....	45
2.10.4 Herramientas de diseño .....	47
2.11 CALIDAD DEL SOFTWARE .....	49
2.11.1 Estándar ISO/IEC 9126 .....	51
2.12 ANÁLISIS DE COSTO DE SOFTWARE .....	63
2.12.1 COCOMO II .....	64
2.13 SEGURIDAD DE LA INFORMACIÓN .....	75
2.14 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE .....	77
2.14.1 técnica caja blanca .....	77
2.14.2 Técnica caja negra .....	78

CAPITULO III MARCO APLICATIVO .....	
3.1 INTRODUCCIÓN .....	80
3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL .....	80
3.3. CAPTURA, ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS .....	81
3.3.1 Obtención de requisitos.....	81
3.3.1.1 Definición de actores.....	83
3.3.1.2 Lista de requerimiento del sistema .....	84
3.3.1.2.1 Requerimientos funcionales .....	85
3.3.1.2.2 Requerimientos no Funcionales .....	85
3.3.2 Especificación de requerimientos para la aplicación. ....	86
3.4 ANÁLISIS DE REQUERIMIENTOS Y DISEÑO DEL SISTEMA.....	87
3.4.1 Modelo de caso de uso .....	87
3.4.2 Modelo de contenido. ....	96
3.4.3 Modelo De Navegación. ....	98
3.4.4 Modelo de Estructura de proceso.....	98
3.4.5 Modelo de Presentación.....	100
3.5 IMPLEMENTACIÓN DEL SISTEMA.....	101
CAPITULO V CALIDAD Y SEGURIDAD .....	
4.1 INTRODUCCIÓN .....	108
4.2 NORMA ISO/IEC 9126.....	108
4.2.1 Funcionalidad.....	108
4.2.2 Confiabilidad .....	115
4.2.3 Mantenibilidad .....	116
4.2.4 Usabilidad .....	117
4.2.5 Portabilidad .....	119



4.3 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE .....	120
4.3.1 Técnica de caja blanca.....	120
4.3.2 Técnica caja negra.....	123
4.3 SEGURIDAD.....	124
4.3.1 Seguridad Lógica. ....	124
4.3.2 Seguridad Física .....	124
4.3.3 Seguridad Organizativa.....	125
CAPITULO V ANÁLISIS DE COSTOS.....	
5.1. INTRODUCCIÓN .....	126
5.2. CALCULO DE COSTOS .....	126
CAPITULO VI CONCLUSIONES Y RECOMENDACIONES.....	
6.1 CONCLUSIONES .....	131
6.1.2 RECOMENDACIONES .....	131
Bibliografía .....	133
ANEXOS	
ANEXO A	
ARBOL DE PROBLEMA	
ANEXO B	
ARBOL DE OBJETIVOS	
ANEXO C	
ORGANIGRAMA DEL MINISTERIO	
ANEXO D	
MANUAL DE USUARIO	
ANEXO E	
AVALES	

## ÍNDICE DE FIGURAS

### CAPITULO II

Figura N° 2. 1 Grafica general de un sistema .....	15
Figura N° 2. 2 Método de cascada, método base de los otros tradicionales .....	24
Figura N° 2. 3 Ciclo de entrega general de las metodologías ágiles .....	25
Figura N° 2. 4 Ciclos de vida de las metodologías tradicionales y ágiles .....	25
Figura N° 2. 5 Aplicación de la Metodología UWE .....	28
Figura N° 2. 6 Elementos de un modelo de Caso de Uso .....	30
Figura N° 2. 7 Modelo de contenido UWE .....	31
Figura N° 2. 8 Modelo de navegación UWE .....	31
Figura N° 2. 9 Estereotipos e iconos para el modelo de navegación UWE .....	32
Figura N° 2. 10 Modelo de estructura de navegación UWE .....	33
Figura N° 2. 11 Modelo de presentación de UWE .....	34
Figura N° 2. 12 Que es MVC? .....	37
Figura N° 2. 13 Ciclo de ida de un MVC .....	37
Figura N° 2. 14 Evaluación ISO/IEC 9126 .....	53
Figura N° 2. 15 Característica de funcionalidad ISO/IEC 9126 .....	54
Figura N° 2. 16 Característica de confiabilidad ISO/IEC 9126 .....	55
Figura N° 2. 17 Característica de usabilidad ISO/IEC 9126 .....	56
Figura N° 2. 18 Característica de eficiencia ISO/IEC 9126 .....	57
Figura N° 2. 19 Característica de capacidad de mantenimiento ISO/IEC 9126 .....	58
Figura N° 2. 20 Característica de portabilidad ISO/IEC 9126 .....	59
Figura N° 2. 21 Calidad de uso de Software ISO/IEC 9126 .....	60
Figura N° 2. 22 Multiplicares del esfuerzo del modelo diseño temprano .....	68
Figura N° 2. 23 Puntos Objeto .....	70
Figura N° 2. 24 Detalles de coeficientes de COCOMO II .....	72
Figura N° 2. 25 Proceso caja blanca .....	78
Figura N° 2. 26 Proceso caja negra .....	79

### CAPITULO III

FIGURA N° 3. 1 Modelo de diagrama de caso de uso: General del Sistema .....	88
---	----

FIGURA N° 3. 2 Diagrama de caso de uso: Administrador/a principal.....	89
FIGURA N° 3. 3 Diagrama de caso de uso: Encargado. ....	92
Figura N° 3. 4 Diagrama de caso de uso: Jefe de unidad.....	94
Figura N° 3. 5 Diagrama de caso de uso: Supervisor. ....	95
Figura N° 3. 6 Modelo de contenido .....	97
Figura N° 3. 7 Modelo de navegación del sistema.....	98
Figura N° 3. 8 Modelo de Estructura de proceso .....	99
Figura N° 3. 9 Modelo de Estructura de proceso .....	99
Figura N° 3. 10 Modelo de Presentación: Administrador/a principal login.....	100
Figura N° 3. 11 Módulo de presentación: administrador principal .....	101
Figura N° 3. 12 Interfaz de Usuarios .....	102
Figura N° 3. 13 Administración de Usuarios .....	102
Figura N° 3. 14 Administración de funcionarios .....	103
Figura N° 3. 15 Administración proveedor .....	103
Figura N° 3. 16 Administración de productos .....	104
Figura N° 3. 17 Administración de colegios .....	104
Figura N° 3. 18 Reporte de acta inventario.....	105
Figura N° 3. 19 Reporte del inventario de los almacenes .....	105
Figura N° 3. 20 Reporte de kardex salida de material .....	106
Figura N° 3. 21 Reporte salida de material.....	106
Figura N° 3. 22 Kardex general .....	107
Figura N° 3. 23 Informe de salida de material .....	107

## ÍNDICE DE TABLAS

### Capítulo II

Tabla 2 1 Diferencia entre metodologías ágiles y tradicionales .....	26
Tabla 2 2 Comparación de bases de datos .....	41
Tabla 2 3 Lenguajes de programación .....	42
Tabla 2 4 Framework de PHP .....	46
Tabla 2 5 Características de Calidad – Modelo ISO/IEC 9126 .....	52
Tabla 2 6 Factor de Ponderación .....	61
Tabla 2 7 Factor de Escala .....	62
Tabla 2 8 Valoración Métrica Punto Función .....	63
Tabla 2 9 Productividad para el modulo Composición de Aplicación .....	66
Tabla 2 10 Atributos de Coste .....	74

### Capítulo III

Tabla 3. 1 Tareas de obtención de requisitos .....	82
Tabla 3. 2 Definición de actores .....	83
Tabla 3. 3 Lista de requerimiento del sistema .....	84
Tabla 3. 4 Requerimientos funcionales .....	85
Tabla 3. 5 Requerimientos no Funcionales .....	85
Tabla 3. 6 Requerimientos para la aplicación .....	86
Tabla 3. 7 Caso de uso: Administrador principal .....	90
Tabla 3. 8 Caso de uso: Encargado de almacén .....	93
Tabla 3. 9 Caso de uso: Jefe de unidad .....	94
Tabla 3. 10 Caso de uso: Supervisor .....	95

### Capítulo IV

Tabla 4. 1 Número de Entradas de Usuario .....	109
Tabla 4. 2 Número de Salidas Usuario .....	110
Tabla 4. 3 Número de Peticiones del Usuario .....	110
Tabla 4. 4 Numero de Archivos .....	111
Tabla 4. 5 Numero de Interfaces Externas .....	112
Tabla 4. 6 Factores de Ponderación .....	112
Tabla 4. 7 Valores de Ajuste de Complejidad .....	113

Tabla 4. 8 Ajuste de Preguntas .....	118
Tabla 4. 9 Ajuste de Preguntas .....	118
Capítulo V	
Tabla 5. 1 Aplicación del Modelo Intermedio .....	127
Tabla 5. 2 Ecuaciones del modelo COCOMO .....	127
Tabla 5. 3 Calculo de Atributos FAE .....	128



## 1.1 INTRODUCCIÓN

Actualmente las instituciones, organizaciones o empresas buscan mejorar el desarrollo y desempeño de sus actividades cotidianas, administrando de mejor manera sus recursos, como la mano de obra y la materia prima. La información resulta imprescindible puesto que llega a ser uno de los recursos principales, esté a su vez puede determinar el éxito o fracaso de una empresa o institución.

A causa de la evolución acelerada de las tecnologías de información y comunicación, y al surgimiento de nuevas herramientas de aplicación, las empresas o instituciones se ven en la necesidad de ingresar a este mercado tecnológico y de ser competitivos en él, por ende obtener de la tecnología informática el mayor beneficio es una tarea fundamental para mantenerse al nivel de la tecnología, obteniendo como resultado la aceleración y automatización de las operaciones manuales que se llevan a cabo para el cumplimiento de sus objetivos.

El GAMEA<sup>1</sup> cuenta con la dirección administrativa en cuya dependencia se encuentra la unidad de servicios generales y mantenimiento, donde se tiene almacenes en diferentes ambientes dependientes del municipio, donde actualmente las operaciones de gestión se las realiza de manera semiautomática y manuales en muchos casos, generando retrasos en el cumplimiento de los objetivos o metas y no se tiene la información de las cantidades existentes en los almacenes, y por ese motivo no se tiene información correcta y oportuna.

El presente proyecto de grado, sistema Web de gestión y control de almacenes, propone mejorar su actual proceso de control de información de almacenes, desarrollando un sistema que ayude a los procesos, entrada, salida de material, cuantificación de las mismas fecha de caducidad fecha de garantía fecha de obsolescencia, y respaldo de documentación que se maneja en almacén, que además nos permitan obtener la información de manera oportuna, veraz y actualizada al momento que se la requiera.

---

<sup>1</sup> Gobierno autónomo municipal de la ciudad El Alto, unidad de mantenimiento y emergencias

Ante esta situación, este proyecto propone una solución que permita a la unidad de servicios generales del GAMEA que a través de un sistema Web que facilite gestionar documentación y los materiales relacionada a los almacenes que tiene la unidad de servicios generales, y tener un control eficiente

## **1.2 ANTECEDENTES**

Haciendo referencia a trabajos realizados a nivel local, fruto de la investigación tanto en bibliotecas de la universidad pública de El Alto y distintas páginas Web, se cita los siguientes trabajos.

### **Nacionales**

- ❖ (Zamora, 2016) “Sistema de control de almacén y seguimiento para el programa de post alfabetización”, en su objetivo menciona la implementación del sistema para coadyuvar en el proceso control, registro y seguimiento de pedagogos, facilitadores y pacientes este sistema se la desarrollo en la dirección de educación del Gobierno Autónomo de la ciudad de El Alto, el sistema utilizo la metodología Web basada en UML y el método de ingeniería, realizado en la Universidad Pública de El Alto 2016
- ❖ (Callizaya, 2017) “Sistema de gestión de almacén e inventarios” caso: Gobierno Autónomo Municipal de Taraco, el objetivo que menciona es diseñar e implementar un sistema de información para la gestión de almacén e inventario, que coadyuve para una mejor administración en almacenes del gobierno municipal de taraco, la metodología que se usó en dicho proyecto es método de ingeniería, método científico, método de estimación de costos, el trabajo se realizó en la Universidad Pública de El Alto 2017.
- ❖ (Tapia, 2008) “Sistema de control de almacén para el área socio educativa de la fundación La Paz”, dicho proyecto tiene como objetivo implementar un sistema de control y optimización de almacén para el Área Socio educativa de la fundación La Paz, utilizo la metodología Web basada en UML y el método de ingeniería, este trabajo se realizó en la Universidad Mayor de San Andrés 2008.



## **1.3 PLANTEAMIENTO DEL PROBLEMA**

### **1.3.1. Problema Principal**

En los almacenes de la unidad de servicios generales y mantenimiento del GAMEA, existe una gran cantidad de movimiento de material que se requiere para hacer mantenimientos en unidades educativas y centros de salud dependientes del municipio plomería, albañilería y electricidad, la documentación que relacionada con solicitudes y entregas de las mismas.

Luego de realizar un análisis de problemas con la consecuente construcción del árbol de problemas. (ver anexo A) permitió identificar que existe la dificultades en los procesos de control de materiales y la documentación relacionada a los almacenes, se las realizan de forma semi-automática, la solicitud, el ingreso, el control de stock existente, ausencia de productos, productos vencidos, caducados todo esto repercute en un manejo NO confiable de la información, afectando a la entrega oportuna de material, control no eficiente, retrasos en la entrega de reportes, y la información solicitada por superiores en tiempo real.

En los almacenes de la unidad de servicios generales y mantenimiento de dependiente de la dirección administrativa del GAMEA falta un sistema Web de gestión y control de almacenes que integre la información respecto a las entradas, salidas, control de stock, control de fechas de vencimiento de los materiales que se encuentran en almacenes, esto causa que no se tenga información oportuna confiable y repercute en el desenvolvimiento óptimo del personal así mismo se tiene pérdidas económicas y finalmente evita que se tome decisiones correctas en base a las normas institucionales respecto a los materiales.

Después de haber analizado las distintas deficiencias se plantea la siguiente interrogante:

¿Cómo optimizar la gestión y control de almacenes y mejorar los procesos de entrada y salida de material de almacén en la Unidad de servicios generales y mantenimiento (GAMEA)?

### **1.3.2. Problemas secundarios**

- ❖ No se tiene la información completa respecto a los materiales que ingresan al almacén provocando pedidos innecesarios.
- ❖ No se tiene la información respecto a las salidas de materiales ocasionando que no se pueda cumplir muchas veces con las solicitudes por la inexistencia de los materiales.
- ❖ No se tiene el control del stock existente lo que ocasiona no poder tomar decisiones para el abastecimiento necesario.
- ❖ Demora en la presentación de informes debido a que los procesos de cuantificación de material son semiautomáticas, con información redundante y con errores.
- ❖ No se cuenta con estadísticas para realizar una planificación adecuada para el abastecimiento del almacén

### **1.4. OBJETIVOS**

El objetivo del presente proyecto nace como la respuesta a las necesidades de la institución, luego de un análisis de problemas y su relación causa-efecto se desarrolló un árbol de objetivos (ver anexo B)

#### **1.4.1 Objetivos Generales**

Desarrollar un sistema Web de gestión y control que coadyuve con la sistematización de información y el seguimiento de almacenes para la Unidad de Servicios Generales y Mantenimiento dependiente del GAMEA, que permita integrar los procesos de entradas, salidas, control de stock, fechas de vencimiento de los materiales y poder generar información en tiempo real cuando se lo requiera.

#### **1.4.2. Objetivos Específicos**

- ❖ Brindar una información completa respecto a los materiales que ingresan al almacén.

- ❖ Tener la información respecto a las salidas de materiales para que se pueda cumplir con las solicitudes de material.
- ❖ Tener un control del stock existente y poder tomar decisiones para el abastecimiento necesario.
- ❖ Reportar informes de manera oportuna y sin errores a las instancias que corresponden.
- ❖ Contar con estadísticas para realizar una planificación adecuada para el abastecimiento del almacén.
- ❖ Diseñar el sistema Web con una interfaz amigable al usuario, que le permitirá manipular de manera eficiente la información que se genera en los procesos de la Unidad.

## **1.5. JUSTIFICACIÓN**

### **1.5.1 Justificación Técnica**

El uso de las computadoras es de gran valor en las instituciones públicas y por ende los sistemas de información, en consecuencia, a esto es que la información que se maneja en las mismas fue creciendo haciendo que los sistemas de información sean un arma fundamental para la administración de la información.

Por consiguiente, el sistema que se realizará será de gran apoyo para el desarrollo de las actividades propias de esta unidad.

El desarrollo del siguiente proyecto se justifica, porque los requerimientos que se necesitan para el desarrollo y la implementación de la misma se adaptan a las características de los equipos con las que cuenta la unidad de servicios generales y mantenimiento.

### **1.5.2. Justificación Económica**

Viendo los requerimientos que necesita nuestro gestor de base de datos, los equipos de la unidad de servicios generales cuentan con estos requisitos y para el desarrollo del

presente proyecto se utilizó software libre<sup>2</sup>, los cuales no tienen costo y el código fuente se puede ser estudiado y modificado para ser conveniencia del sistema. La unidad de servicios generales no cuenta con un servidor Hosting aun, de momento el sistema funcionara de forma local, intranet, por esta razón no se menciona los costos de hosting o un dominio.

Con la implementación del sistema se minimizará los costos en la administración de almacenes, así como en los procesos realizados por el encargado de almacén, se tendrá un control eficiente en el registro de ingresos, salidas, control de stock y fechas de vencimiento de materiales evitando pérdidas de materiales y de tiempo lo que implica perdidas económicas.

### **1.5.3. Justificación Social**

La implementación de un sistema proporcionará a la unidad de servicios generales y mantenimiento una herramienta que colabore y facilite las actividades diarias del personal encargado del almacén (Responsable de almacén), mejorando el desempeño del personal (jefe de unidad, encargados de almacén y personal que interviene en los almacenes), mejorando el registro control de los ingresos y las salidas de materiales, beneficiando en gran parte a la gestión del GAMEA, por ende a la población de El Alto.

## **1.6. METODOLOGÍA**

### **Metodología UWE.**

Para el desarrollo del software se hará uso de la metodología UWE (UML-based Web Engineering, en español Web basada en UML), es una metodología que permite modelar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones, con una gran cantidad de definiciones, en el proceso de diseño. Procede de manera iterativa e incremental, coincidiendo con UM, incluyendo flujos de trabajo y puntos de control, en seis faces, (**Captura, análisis y especificación de requisitos, diseño del sistema,**

---

<sup>2</sup> **Software libre:** es todo software cuyo código fuente puede ser estudiado, modificado y utilizado libremente con cualquier fin.

**codificación del software, pruebas, la instalación o fase de implementación y el mantenimiento).** (Pressman, 2010)

### **Métricas de calidad**

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software. Los modelos de calidad para el software se describen así, calidad Interna, calidad externa y calidad de uso. (Andrade, 2015).

### **Método de estimación de costos de software**

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por ConstructiveCostModel (Modelo Constructivo de Coste), se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria. (Adriana Gómez, 2016).

COCOMO es una herramienta basada en la línea de código la cual le hace muy poderoso para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.

Con la realización de este trabajo ampliamos nuestro conocimiento acerca de la estimación de costos que es fundamental para un analista, administrador de proyecto. Según (Uncategorized, 2008).

## **1.7. HERRAMIENTAS**

FRAMEWORK PARA EL DESARROLLO WEB:

**JAVASCRIPT:** JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. (absurdum, 2016).

## **HTML5:**

Es un lenguaje markup (de hecho, las siglas de HTML significan Hyper Text Markup Lenguaje) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Según (pvn, 2013).

**BOOTSTRAP:** Es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales y se puede descargar de dos maneras, compilado o mediante el código fuente original. Según (Mark Otto, 2017).

**MySQL:** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. Según (Delgado, 2014).

**PHP:** Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. (Alvarez, 2017).

**AJAX:** Es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).

**JQUERY:** Es una librería de componentes para el framework jQuery que provee un conjunto de plugins, efectos visuales, es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto

simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web sin tener conocimientos del lenguaje. Según (Suárez, 2014).

## **1.8. LIMITES Y ALCANCES**

### **1.8.1. Limites**

El proyecto se limita al control de las actividades del almacén de la unidad de servicios generales y mantenimiento (ingresos, salidas, solicitudes y reporte de las mismas) que se genera al momento de realizar una solicitud de mantenimiento y respuesta a la misma. La implementación del sistema local, dentro del municipio, por la especificación de los requerimientos.

### **1.8.2. Alcances**

Para cumplir con las metas y requerimientos se realizara la implementación y el desarrollo de sistema orientado a la Web que contendrá los siguientes módulos:

❖ **Módulo de Administración de usuarios:**

En este módulo permite administrar al personal de almacenes de la unidad de servicios generales y asignar el nombre de usuario y contraseña para que accedan al sistema.

❖ **Módulo de Administración de proveedor:**

En este módulo muestra los distintos proveedores que suministran material a la unidad de servicios generales, donde se guardan los datos del proveedor y el tipo de material con la que cuenta.

❖ **Módulo de Administración de productos:**

En este módulo se muestra el tipo de material que se tiene en los almacenes, el lugar donde se encuentran estos y alerta sobre la poca existencia de un producto y la caducidad de las mismas

❖ **Módulo de Administración de colegios:**

En este módulo se muestra la frecuencia con la que solicita atención, mantenimiento, una unidad educativa, esto es para poder tener una atención equitativa a los 14 distritos de la ciudad de El Alto.

❖ **Módulo de Administración de Stock:**

En este módulo muestra las cantidades de material que se tiene en los almacenes, nos informa si se necesita solicitar algún tipo de material y nos da un reporte actual de las cantidades disponibles.

❖ **Módulo de reportes**

En este módulo muestra los reportes solicitados, ya sea de materiales existentes, salientes, de materiales solicitados por los colegios, listado de funcionarios con sus respectivos cargos, todo esto con la opción de impresión.

### **1.9. APORTES**

El desarrollo del sistema permite cubrir las necesidades para un mejor control y seguimiento de los materiales en almacenes de la Unidad de Servicios Generales implementando diferentes módulos que permitan acelerar la gestión de toda la información que se genera en los almacenes, haciendo que se emita resultados más rápidos y completos.



## **2.1. INTRODUCCIÓN**

En el presente capítulo se revisa las definiciones y conceptos empleadas para el desarrollo del presente proyecto.

## **2.2. UNIDAD DE SERVICIOS GENERALES Y MANTENIMIENTO (GAMEA)**

Mediante ley N° 728 del 6 de marzo de 1985, se crea la cuarta sección municipal de la provincia Murillo del departamento de La Paz, con su capital El Alto de La Paz, la misma que fue elevado a rango de ciudad mediante ley N° 1014, del 26 de septiembre de 1988, durante la presidencia del Dr. Víctor Paz Estensoro.

Se encuentra la urbe Alteña: ubicado en una meseta de superficie plana y ondulada, al pie de la cordillera de La Paz (Meseta del altiplano norte) y la Cordillera Oriental, al Noreste con la Sección capital de la provincia Murillo, el Este con el Municipio de La Paz, al sur con el Cantón Viacha, perteneciente a la provincia Ingavi, y al Oeste con el Cantón Laja, correspondiente a la segunda sección de la provincia Los Andes.

## **ÓRGANO EJECUTIVO DEL GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO**

### **Razón de ser de la unidad organizacional**

Generar políticas públicas para mejorar las condiciones de vida y elevar los niveles de bienestar social en el Municipio de El Alto, a través de una Gestión Municipal eficiente, responsable y transparente, respetando, preservando y fomentando valores culturales, sociales, familiares, derechos humanos y equidad de género, en estricto apego al ordenamiento jurídico vigente.

### **FUNCIONES Y ATRIBUCIONES**

- ❖ Cumplir las competencias exclusivas municipales establecidas en la Constitución Política del Estado.
- ❖ Ejercer las atribuciones y facultades establecidas por la Ley N° 031 de Autonomías y Descentralización “Andrés Ibáñez”, las normas que rigen el ámbito municipal y Ley N° 482 de Gobiernos Autónomos Municipales.
- ❖ Velar por la correcta administración de los recursos y bienes del GAMEA en el marco de la Ley N° 1178 de Administración y Control Gubernamentales.

- ❖ Promover mecanismos destinados a prevenir y procesar actos de corrupción cometidos por servidoras y servidores públicos, ex servidoras y ex – servidores públicos, personas naturales y jurídicas en el marco de la Ley N° 004 "Marcelo Quiroga Santa Cruz", Ley N° 974 Ley de Unidades de Transparencia y Lucha contra la Corrupción.
- ❖ Informar al Concejo Municipal sobre la ejecución de los planes, programas, proyectos y actividades desarrolladas por la administración municipal.
- ❖ Gestionar convenios de cooperación interinstitucional con instituciones nacionales e internacionales para el financiamiento de programas y proyectos de inversión.
- ❖ Establecer políticas que permitan la transparencia y la rendición de cuentas de la gestión municipal.
- ❖ Presentar información sobre el manejo de los recursos municipales ante las instancias que ejercen el control gubernamental.
- ❖ Otras funciones contenidas en disposiciones legales que rigen el ámbito municipal y autonómico. Disponible. (MOF, 2018) <sup>3</sup>

### **Extensión territorial**

La extensión territorial del municipio de El Alto, considerando el distrito municipal 13, es aproximadamente, de 387.56 Km<sup>2</sup> (38.756 Has), que representa el 7.58% de la superficie total de la provincia murillo.

### **Misión y Visión**

La visión del Gobierno Municipal de El Alto es la siguiente:

***“institución ágil y moderna orientada a la consecución del Suma Qamaña de los Alteños”***

La misión plantea una institución con capacidad de dar soluciones apropiadas y oportunas a los problemas sociales y económicos de la población del municipio, facilitando el bienestar de los Alteños traducido en el “Vivir Bien”

La Misión del Gobierno Municipal de El Alto es la de:

***“Promover el desarrollo participativo del municipio a través de procesos de integración social y económica”.***

La razón de ser Gobierno Municipal de El Alto es la búsqueda del desarrollo participativo que supone el compromiso de los actores sociales e institucionales, quienes se constituyen en generadores de estrategias de desarrollo mediante propuestas integrales, es decir, articulando aspectos sociales y económicos. Según, (GAMEA, 2010)

ORGANIGRAMA ÓRGANO EJECUTIVO DEL GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO (organigrama ver anexo C)

Dentro del órgano ejecutivo se tiene secretarías, direcciones y unidades:

❖ **Secretaría Municipal de Administración y Finanzas**

Llevar adelante la gestión administrativa y financiera del GAMEA, en el marco de la normativa vigente, mediante la definición e implementación de políticas, bajo parámetros de transparencia, eficiencia, eficacia y economicidad. (MOF, 2018)

❖ **Dirección Administrativa**

Administrar, supervisar y coordinar con las unidades operativas dependientes de la Dirección Administrativa para el cumplimiento de la normativa vigente, reglamentos, procedimientos, instructivos, para su aplicación en el GAMEA.

❖ **Unidad de Servicios Generales y Mantenimiento**

Atender el mantenimiento y refacción de los bienes muebles e inmuebles del Gobierno Autónomo Municipal de El Alto, gestionar la provisión y coordinar el pago de los servicios básicos de agua, luz, gas, telefonía fija y móvil, alquileres y fotocopiado, y la dotación de insumos para mantenimiento preventivo y correctivo

La unidad de servicios generales y mantenimiento cuenta con almacenes en donde se acopia el material para atender el mantenimiento de las unidades solicitantes, en donde hay en responsable y 3 encargados de estos almacenes.

Dentro de la unidad de Servicios Generales y Mantenimiento también se tiene un orden jerárquico, a la cabeza está el jefe de unidad.

Unidad de servicios generales y mantenimiento

**Jefe de unidad:** coordina con la dirección de educación y entidades solicitantes de algún tipo de refacción o intervención para el mantenimiento de las mismas.

**Inspectores:** tiene la labor de verificar el trabajo de los funcionarios

❖ **Funcionarios**

**Responsable de almacén:** coordina la entrada y salida de materiales.

- ❖ Encarados de almacén: realizan la entrega de material.
- ❖ Personal: organización de los materiales en almacenes.

**Supervisores:** verifica las cantidades requeridas por los colegios

- ❖ Jefe de cuadrilla: controla el trabajo de los obreros
- ❖ Obreros  
(Ver anexo C)

## **2.3 SISTEMA**

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software).

El sistema tiene interacción con el ambiente, del cual recibe entradas y al cual se le devuelven salidas. El ambiente también puede ser una amenaza para el sistema.

“Un sistema es módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. El concepto se utiliza tanto para definir a un conjunto de conceptos como a objetos reales dotados de organización”. (Porto J. P., 2018)

Figura N° 2. 1 Grafica general de un sistema



**Fuente:** (Porto J. P., 2018)

#### 2.4. Web

Es un vocablo inglés que significa “red”, “telaraña” o “malla”. El concepto se utiliza en el ámbito tecnológico para nombrar a una red informática y, en general, a Internet este término, de todas formas, tiene varios usos. Además de nombrar a Internet en general, la palabra Web puede hacer mención a una página Web, un sitio Web o hasta un servidor Web.

Es importante establecer que este término además forma parte de lo que se conoce como World Wide Web que es la red informática que se emplea en todo el mundo. A finales de la década de los años 80 fue cuando nació aquella que permite que hoy cualquier ciudadano, a través de una conexión a Internet junto a un navegador y un ordenador, pueda acceder desde cualquier rincón del mundo a la web que desea. Lo puede hacer para informarse, para entretenerse o simplemente por simple curiosidad.

Una página web es un documento que incluye un archivo HTML con texto, imágenes, videos, animaciones Flash, etc. Al conjunto de páginas web que suelen formar parte del mismo dominio o subdominio de Internet se lo conoce como sitio web. Dentro del sitio web, todas las páginas guardan alguna relación entre sí y están vinculadas mediante vínculos (también conocidos como enlaces, hipervínculos, hiperenlaces o links). Según, (Porto J. P., 2013)

### **2.4.1 SISTEMA Web**

Se denomina así a un conjunto de elementos orientados a internet, ordenadamente relacionadas entre sí que aporta a la organización en su funcionamiento, dando información necesaria para el cumplimiento de sus fines u objetivos, obteniéndola con procesar, almacenar información dinámicamente, son sistemas informáticos complejos, como los programas que antes teníamos en la computadora, pero para internet, es decir, que se codifican en lenguajes soportados por los navegadores web y se alojan en un servidor en Internet. Por ejemplo, un sistema para llevar la administración de una clínica, al que se accede mediante www o una red privada local, es una aplicación web. Las aplicaciones web siempre están en internet, pero pueden manejarse mediante intranets y extranets, depende la seguridad y privacidad requerida por el cliente.

Los Sistemas Web tienen un gran impacto ahorrador, optimizando los procesos de su empresa, y/o mejorar y facilitar la relación con sus clientes, con seguridad y privacidad. Estando accesible por Internet se acoplan a empresas con oficinas regionales, o con clientes en diferentes países. Estas aplicaciones se rigen por los mismos principios de usabilidad como las páginas web.

Los Sistemas Web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Sabemos la importancia de la relación Calidad / Precio, por eso ofrecemos precios razonables con la Mejor Calidad de Desarrollo en el Mercado. Contamos con un desarrollo dividido en fases: Análisis, Desarrollo, Implementación, Puesta en Marcha del Sistema, Capacitación y Soporte, con informes constantes del avance del proyecto para que pueda supervisar usted mismo el desarrollo del Proyecto. (Zegarra, 2015)

### **2.5. GESTIÓN Y CONTROL DE ALMACÉN**

**GESTIÓN:** El concepto de gestión hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo

cualquiera. Administrar, por otra parte, abarca las ideas de gobernar, disponer dirigir, ordenar u organizar una determinada cosa o situación.

La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio.

Partiendo de dichas acepciones podríamos utilizar como frase que dejara patente las mismas, la siguiente: Pedro ha sido ascendido dentro de su empresa como reconocimiento a la magnífica gestión que ha realizado al frente del área en el que se encontraba trabajando.

Importante es subrayar que la gestión, que tiene como objetivo primordial el conseguir aumentar los resultados óptimos de una industria o compañía, depende fundamentalmente de cuatro pilares básicos gracias a los cuales puede conseguir que se cumplan las metas marcadas. Según (Porto J. P., 2013)

**CONTROL:** Control puede ser el dominio sobre algo o alguien, una forma de fiscalización, un mecanismo para regular algo manual o sistémicamente o un examen para probar los conocimientos de los alumnos sobre alguna materia.

En administración, control es un mecanismo del proceso administrativo creado para verificar que los protocolos y objetivos de una empresa, departamento o producto cumplen con las normas y las reglas fijadas. El control tiene como objetivo evitar irregularidades y corregir aquello que frena la productividad y eficiencia del sistema como, por ejemplo, los mecanismos de control de calidad.

Significa comprobación, inspección, fiscalización o intervención. También puede hacer referencia al dominio, mando y preponderancia, o a la regulación sobre un sistema.

El control, por otra parte, es la oficina, el despacho o la dependencia donde se controla. Por eso puede hablarse de puesto de control. Según (Porto J. P., 2013)

**ALMACÉN:** Es un espacio, recinto, edificio, o instalación donde se suele guardar la mercancía, pero al mismo tiempo puede hacer otras funciones, como por ejemplo el acondicionamiento de productos determinados, hacer recambios (tanto para el

mantenimiento como para la existencia técnica) y otros. Etimológicamente diríamos que el término almacén viene derivado del árabe (almaizan) y es una casa o edificio donde se guardan géneros de cualquier clase. Por tanto, un almacén fundamentalmente se encarga de guardar el stock, pero no debemos de confundir los términos. La gestión del stock no será la misma que la gestión del almacén. La primera se encarga de aprovisionar para un buen nivel de servicio mientras que la segunda intenta realizar las operaciones de almacenamiento (algunas veces también de preparación y producción) con los mínimos recursos propios del almacén (como son el espacio, la maquinaria y el Personal). De esta forma, para la gestión del almacén, la gestión del stock se convertirá en proveedora de servicios logísticos de almacenaje y preparación. Según (Molina, 2007).

### **Clasificación de los almacenes**

Los criterios para clasificar los almacenes pueden ser varios y, por tanto, se pueden considerar varias clases de almacenes:

- ❖ Almacenes principales o centrales.
- ❖ Almacenes subsidiarios o periféricos.
- ❖ Depósitos y almacenes móviles.

Los almacenes de distribución antes mencionados también pueden presentar su propia clasificación, que podemos dividir en:

- ❖ Almacenes de PLANTA. Contienen productos terminados en espera de ser distribuidos. Suelen hallarse situados dentro del recinto de la fábrica, constituyendo el primer escalón del sistema logístico.
- ❖ Almacenes de CAMPO. Dentro del sistema logístico se encuentran en diferentes niveles: regionales, provinciales, locales, etc. Tienen por misión el mantenimiento de los stocks del sistema logístico.
- ❖ Almacenes de TRÁNSITO o PLATAFORMAS. Son creados fundamentalmente para atender a las necesidades de transporte, compensan los costes de almacenamiento con mayores volúmenes transportados. Actualmente este tipo de almacén está teniendo mucha aceptación entre los operadores logísticos y las empresas productoras.



- ❖ Almacenes TEMPORALES o DEPÓSITOS. Son los dedicados, casi siempre, a los productos perecederos. Si tenemos en cuenta la naturaleza de los productos almacenados, podemos distinguir almacenes de materias primas, de productos semi elaborados, de productos terminados. de piezas de recambio, de materiales auxiliares y de archivos de información. Según, (Medrano, 2012)

## **2.6 MÉTODOS DE VALORACIÓN DE INVENTARIOS**

Los métodos de valoración o métodos de valuación de inventarios son técnicas utilizadas con el objetivo de seleccionar y aplicar una base específica para evaluar los inventarios en términos monetarios. La valuación de inventarios es un proceso vital cuando los precios unitarios de adquisición han sido diferentes.

Existen numerosas técnicas de valoración de inventarios, sin embargo las comúnmente utilizadas por las organizaciones en la actualidad (dada su utilidad) son:

- ❖ Identificación Específica
- ❖ Primeros en Entrar Primeros en Salir – PEPS
- ❖ Últimos en Entrar Primeros en Salir – UEPS
- ❖ Costo promedio constante o Promedio Ponderado.

### **2.6.1 Primero en entrar primero en salir- PEPS**

Comúnmente conocido como FIFO (First In, First Out), este método de valoración de inventarios se basa en la interpretación lógica del movimiento de las unidades en el sistema de inventario, por ende el costo de las últimas compras es el costo de las existencias, en el mismo orden en que ingresaron al almacén. (Lopez, 2019)

**El método FIFO tiene varias ventajas a nivel logístico que no debes despreciar:**

- ❖ Es muy recomendable cuando se trata de productos perecederos, ya que usando este método evitarás que se estropee tu mercancía antes de haberla vendido.
- ❖ No sólo es útil para productos perecederos, si no que con cualquier tipo de mercancía evitarás que quede obsoleta.

- ❖ Es muy útil cuando el precio de la mercancía que manejas tiende a bajar o a mantenerse con el tiempo.
- ❖ No sólo es aplicable a la hora de realizar el inventario, sino que también puedes utilizarlo en la cadena de suministro, el transporte y el almacenamiento de todos los productos de tu empresa.
- ❖ La complejidad en la gestión del inventario de tus productos aumenta en función de la naturaleza o el volumen de los mismos. Informatizar tu almacén y elegir el método más apropiado para la gestión de tus mercancías son herramientas que te permitirán maximizar tus beneficios y minimizar tus pérdidas de stock. (Lopez, 2019).

## **2.7 INGENIERÍA DE SOFTWARE**

La Ingeniería de Software es una disciplina que ayuda a producir el software desde la etapa de requerimientos hasta el mantenimiento del mismo, para desarrollar software fiable y que funciones de manera eficiente. (Pressman, 2001)

Según el IEEE (Institute of Electrical and Electronics Engineer) define a la Ingeniería del Software como: la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software

### **2.7.1 Modelos de desarrollo de software**

Los distintos modelos para el desarrollo de software son:

#### **❖ Modelo en cascada**

Este es el modelo en el cual se ordenan rigurosamente las etapas del desarrollo del software, de esto se obtiene que el inicio de una etapa de desarrollo deba de esperar el fin de la etapa anterior. De esto se obtiene que cualquier error detectado lleve al rediseño del área de código afectado, lo cual aumenta de costo el desarrollo del proyecto.

#### **❖ Modelo de prototipos**

Pertenece a los modelos evolutivos, en el cual el prototipo debe de ser construido rápidamente y con la utilización escasa de recursos. El prototipo es construido para mostrárselo al cliente, obtener críticas y retroalimentación, con lo cual se obtendrán los

requisitos específicos para la aplicación a partir de las metas graficas que son mostradas. Las etapas de este modelo son:

1. Plan rápido
2. Modelo
3. Construcción del prototipo
4. Entrega y retroalimentación
5. Comunicación
6. Entrega del desarrollo final

Entre sus ventajas se encuentra que, es apto para el cliente que conoce a grandes rasgos el objetivo del software y a su vez, al equipo de desarrollo le ofrece una mejor visibilidad de la interacción del cliente con el software y el ambiente en el cual debe de realizarse.

#### ❖ **Modelo en espiral**

Es el modelo en el cual las actividades se desarrollan en espiral, estas actividades se realizan conforme se van seleccionando de acuerdo al análisis de riesgo. En cada iteración en este modelo, se deberán de tomar en cuenta los objetivos, las alternativas que se deberán de tomar de acuerdo a las características, estas son experiencia personal, requisitos a cumplir, las formas de gestión del sistema, entre otros. Este modelo tiene dos formas en las cuales se debe de planificar el proyecto, la forma angular, la cual indica únicamente el avance del software dentro del proyecto y la forma radial, la cual indica el aumento del costo dado que cada iteración conlleva más tiempo de desarrollo.

#### ❖ **Desarrollo por etapas**

Es similar al modelo por etapas, sin embargo se diferencia en que al momento de la crítica o bien retro alimentación por parte del usuario final, no se obtendrán completamente las características del software. Estas se irán descubriendo en el proceso del avance del software, mediante la creación de las diferentes versiones del código. En este modelo, se distinguen las siguientes fases:

1. Especificación conceptual

2. Análisis de requisitos
3. Diseño inicial
4. Codificación y depuración.

#### ❖ **Desarrollo iterativo y creciente**

Resuelve los problemas encontrados en el modelo en cascada, en cual ofrece entornos de trabajo con técnicas para su correcta utilización. Este tipo de modelo es esencial para el método de programación extrema. Este tipo de programación consiste en la realización de programas de manera incremental, la cual sirve para obtener ventaja de lo que se ha realizado a lo largo del proyecto. En este se entran a varias iteraciones con las cuales se obtendrá el software final y a su vez, se le agregaran nuevas funcionalidades a cada etapa. Se puede dividir en los siguientes procesos:

##### 1. Etapa de inicialización

La meta de esta etapa es la construcción de un producto en el cual se pueda obtener retroalimentación de parte del usuario final.

##### 2. Etapa de iteración

Consiste en el análisis, rediseño e implementación del producto de las iteraciones anteriores.

##### 3. Lista de control del proyecto.

Son las tareas que se crean que describen las partes que conforman el proyecto, son implementadas y rediseñadas en cada etapa de iteración del producto.

#### ❖ **RAD o desarrollo de aplicaciones rápidas**

Como su nombre lo indica permite la construcción rápida de sistemas utilizables. Está compuesto por un grupo reducido de personas incluyendo desarrolladores y testers del sistema. También se debe de hacer énfasis al desarrollo de la aplicación cumpliendo correctamente las funcionalidades principales, dejando a un lado a las implementaciones secundarias. Este modelo toma principalmente en cuenta las características de usabilidad, utilidad y rapidez de la ejecución de la aplicación.

### ❖ **Desarrollo concurrente**

Se conoce como ingeniería concurrente y es utilizado en su mayoría para aplicaciones cliente servidor, en el cual se describen los múltiples procesos que ocurren simultáneamente en la aplicación. Una de las características de este proceso es que está orientado a las necesidades del usuario, las decisiones de la gestión y los resultados de las revisiones. Las ventajas que se pueden mencionar es que está orientado a grupos de trabajo independientes, proporcionando una visión exacta de lo que se lleva desarrollado del proyecto. Las desventajas se tiene que se necesitan de grupos de trabajo y de las condiciones necesarias para su implementación.

### ❖ **Proceso unificado**

Este proceso se distingue por la utilización de casos de uso, el cual está centrado en la arquitectura y la utilización de iteraciones incrementales. Este es un marco de trabajo extensible, el cual puede ser implementado hacia otros proyectos de distintas organizaciones. Este proceso es utilizado para evitar problemas legales con el método RUB dado que este otro método es una marca registrada de IBM. En cada iteración, se busca el avance e iteración en determinadas áreas, con la cual se obtendrán resultados en los cuales se puede constatar el tiempo que se ha dedicado a las distintas áreas en el desarrollo de software. Sus principales características son: es iterativo e incremental, dirigido por casos de uso, centrado en la arquitectura y enfocado en los riesgos

### ❖ **RUP o proceso unificado relacional**

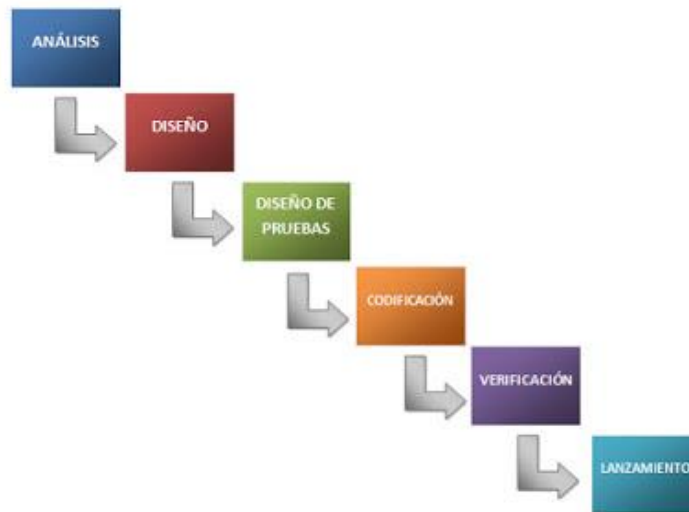
Es propiedad de IBM desarrollado por la empresa Rational Software, es conocido como RUP y llamado Proceso Unificado Rational, por el nombre de la empresa. Es uno de los modelos más utilizado en el análisis, diseño e implementación de la documentación de sistemas orientados a objetos. Los principios básicos de este método son: adaptar el proceso, equilibrar prioridades, demostrar valor iterativamente, colaboración entre equipos, elevar el nivel de abstracción y enfocarse a la calidad. (Valdez J. L., 2013)

### 2.7.1.1 Metodologías tradicionales

Las metodologías tradicionales o denominadas a veces como metodologías pesadas. Buscan siempre llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto al pie de la letra, definido todo esto, en la fase inicial del desarrollo del proyecto.

Otra de las características importantes dentro de estas tradicionales, es el alto costo que significaría realizar un cambio a nuestro proyecto y la falta de flexibilidad en proyectos donde el entorno es cambiante. Las metodologías tradicionales centran su atención en la documentación, planificación y procesos. (Camacho, 2016)

**Figura Nº 2. 2 Método de cascada, método base de los otros tradicionales**



**Fuente:** (Camacho, 2016)

### 2.7.1.2 Metodologías ágiles

Las metodologías ágiles nacen como respuesta a los problemas que se presentan en las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Las metodologías están basadas en adaptabilidad de los procesos de desarrollo. Las metodologías ágiles ven más importante

la capacidad de adaptarse a los cambios que el seguir un plan estricto de desarrollo. (Camacho, 2016)

**Figura N° 2. 3 Ciclo de entrega general de las metodologías ágiles**



**Fuente:** (Camacho, 2016)

❖ **Diferencias entre las metodologías ágiles y tradicionales.**

**Figura N° 2. 4 Ciclos de vida de las metodologías tradicionales y ágiles**



**Fuente:** (Camacho, 2016)

*Tabla 2 1 Diferencia entre metodologías ágiles y tradicionales*

<b>METODOLOGÍAS ÁGILES</b>	<b>METODOLOGÍAS TRADICIONALES</b>
Están preparadas para cambios durante el proyecto	Son poco flexibles a los cambios
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte activa en el proceso de desarrollo	El cliente interactúa con el equipo solo mediante reuniones de entregas
Grupos pequeños, 10 integrantes o menos y trabajando en el mismo sitio en el cual todos tienen conocimiento sobre todo el proceso de desarrollo	Grupos grandes y posiblemente distribuidos donde a cada integrante se le asignan tareas específicas
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

**Fuente:** (Camacho, 2016)

## **2.8 METODOLOGÍA DE DESARROLLO UWE**

Es una metodología que permite modelar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones, con una gran cantidad de definiciones, en el proceso de diseño de sistemas. Procede de manera iterativa e incremental, coincidiendo con UML<sup>3</sup> incluyendo flujos de trabajo y puntos de control. Según, (Pressman, 2010)

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

---

<sup>3</sup> UML lenguaje de modelo unificado



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. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. (Pressman, 2010)

### **2.8.1 Fases de la Metodología UWE**

#### **1) Captura, análisis y especificación de requisitos.**

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

#### **2) Diseño del sistema.**

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

#### **3) Codificación del software.**

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

#### **4) Pruebas**

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

#### **5) La Instalación o Fase de Implementación.**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Implementación y Lanzamiento.

En la implementación de la Pagina Web es recomendable utilizar estándares (HTML, XHTML...) para asegurar la futura compatibilidad y escalabilidad del sitio. Una vez implementada la página web y aprobada su funcionalidad se procede al lanzamiento del sitio.

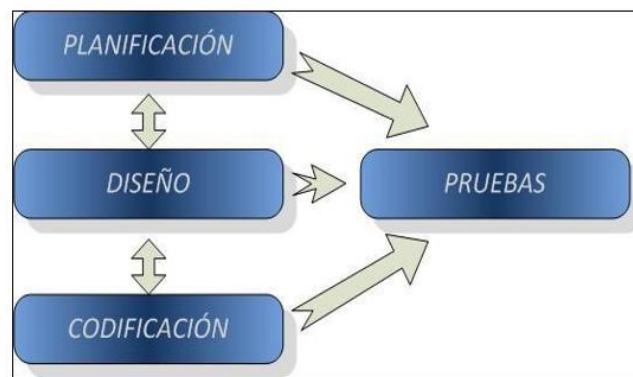
## 6) El Mantenimiento.

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

### ❖ Mantenimiento y Seguimiento.

Una vez puesta la Pagina Web a Disposición de los usuarios hay que ir cambiando datos y mantener este sitio actualizado, ya que esta página no puede permanecer estática. Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes, opiniones de los usuarios, el comportamiento y uso del sitio. Según, (Pressman, 2010)

**Figura Nº 2. 5 Aplicación de la Metodología UWE**



**Fuente:** (Pressman, 2010)

### ❖ UWE basado en UML

La ingeniería Web basada en UML (UWE) fue presentada por Nora Koch en el 2000. Esta metodología utiliza un paradigma orientado a objetos, y está orientada al usuario.

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. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

#### ❖ **Actividades del Modelado de UWE**

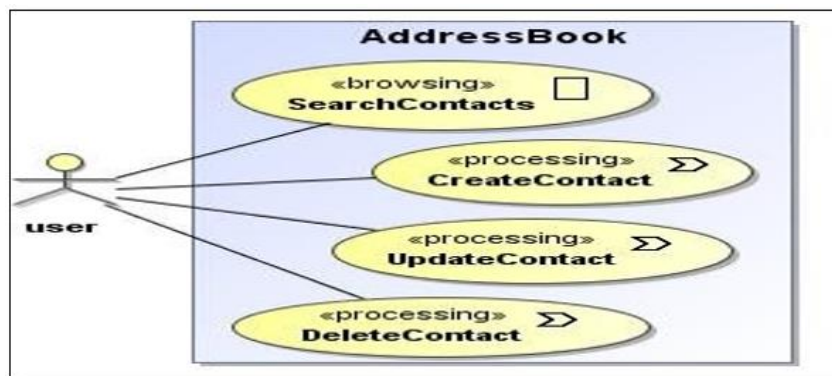
Las actividades base de modelado de UWE son el análisis de requerimientos el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de escenarios web.

- ❖ Análisis de Requerimientos
- ❖ Modelo Conceptual
- ❖ Modelo de Navegación
- ❖ Modelo de Presentación

## ❖ Modelo de Casos de Uso

Para describir los requerimientos funcionales de una aplicación se puede usar un modelo de caso de uso. Este describe un trozo de comportamiento de la aplicación sin revelar su estructura interna. Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe. Los componentes primarios de un modelo de casos de uso (case-use model) son los casos de uso (use cases), los actores y el sistema modelado. Según, (Pressman, 2010)

**Figura Nº 2. 6 Elementos de un modelo de Caso de Uso**



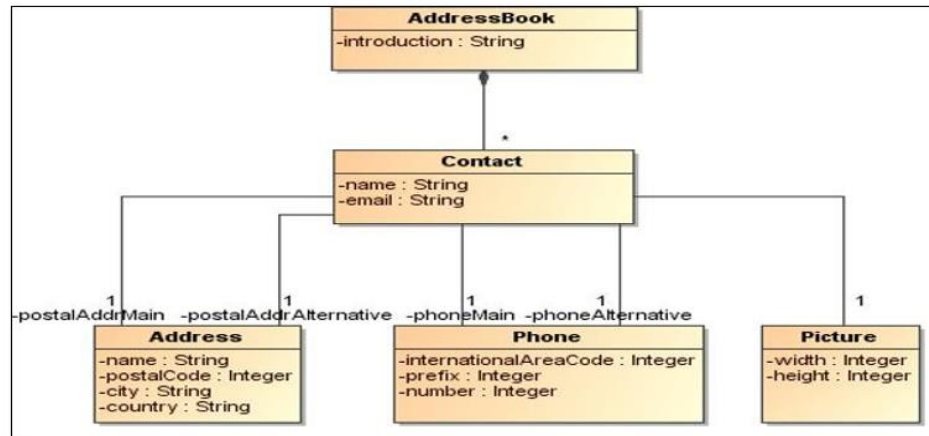
**Fuente: (Pressman, 2010)**

## ❖ Modelo Conceptual

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir define la estructura de los datos que se encuentran alojados en el sitio web.

Su objetivo es construir un modelo conceptual del dominio de la aplicación considerando los requisitos reflejados en los casos de uso. Da como resultado un diagrama de clases de dominio. (Pressman, 2010)

**Figura Nº 2. 7 Modelo de contenido UWE**



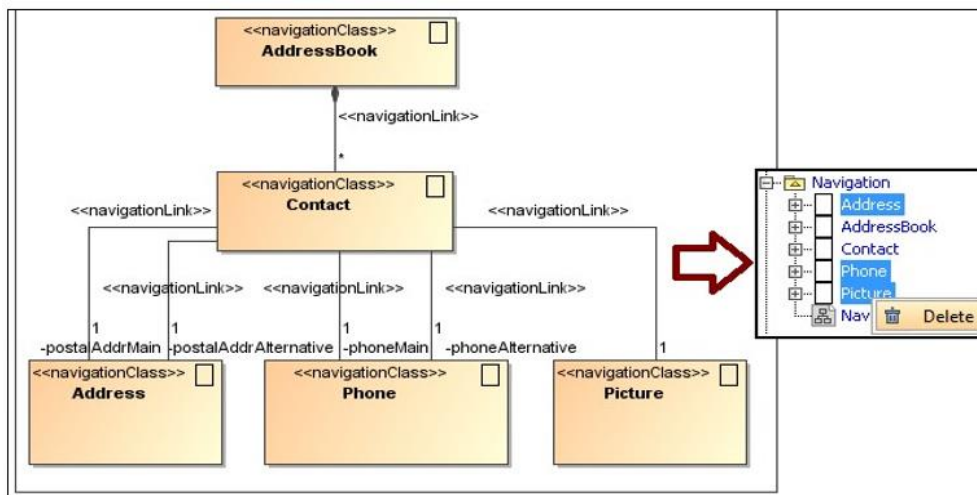
Fuente: (Pressman, 2010)

❖ **Modelo de Navegación**

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links).

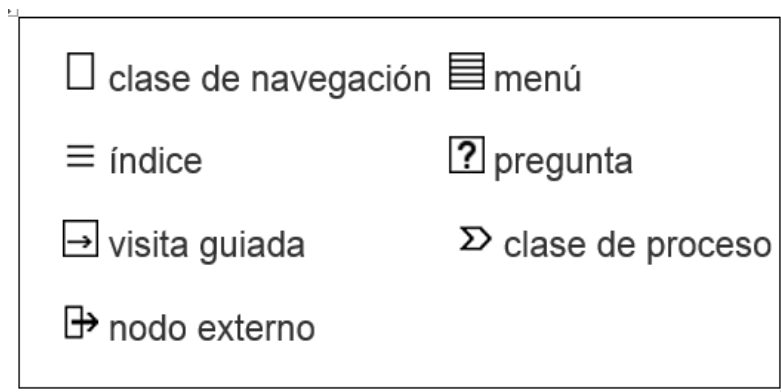
Se obtienen el modelo de espacio de navegación y modelo de estructura de navegación, que muestra cómo navegar a través del espacio de navegación. Se obtienen diagramas de clases que representan estos modelos. Según, (Pressman, 2010)

**Figura Nº 2. 8 Modelo de navegación UWE**



Fuente: (Pressman, 2010)

**Figura Nº 2. 9 Estereotipos e iconos para el modelo de navegación UWE**



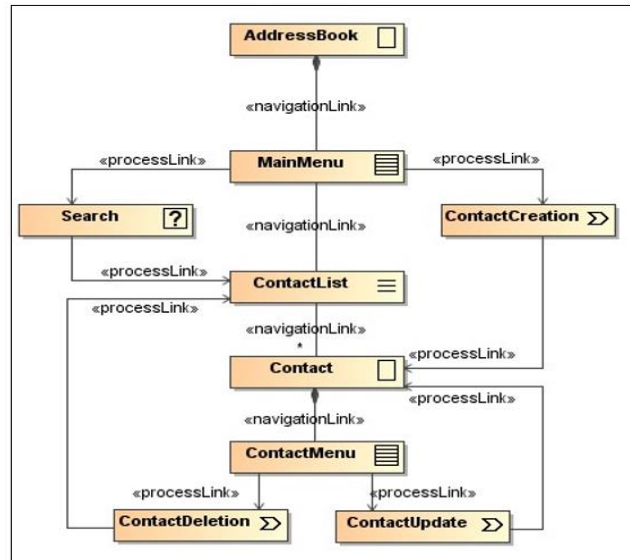
**Fuente:** (Pressman, 2010)

❖ **Modelo estructura de navegación**

Este refinamiento consiste en mejorar el modelo d espacio de navegación añadiendo estructuras de acceso como: índices, guías de ruta, consultas y menús.

Las clases estereotipadas para estas estructuras son: <<index>> (especifica acceso directo a todas las instancias del destino proporcionando una lista de todos los elementos desde donde se puede seleccionar para la navegación en la aplicación web), <<guided tour>> (proporciona acceso secuencial a instancias del nodo destino), <<query>> (representa la posibilidad de buscar por instancias en el nodo destino), <<menu>> (son utilizados para estructurar la salida de enlaces desde un nodo). Según, (Pressman, 2010)

**Figura Nº 2. 10 Modelo de estructura de navegación UWE**



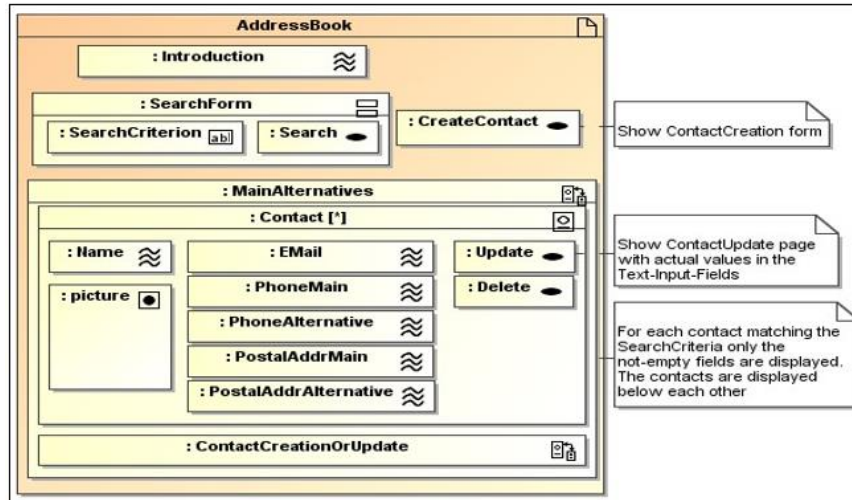
**Fuente:** (Pressman, 2010)

#### ❖ **Modelo de Presentación**

Basándose en el modelo anterior y en información adicional recopilada durante el análisis de requerimientos se elabora el modelo de presentación, cuyo propósito es el diseño abstracto de interfaces de usuario.

Se obtienen el modelo de espacio de navegación y modelo de estructura de navegación, que muestra cómo navegar a través del espacio de navegación. Se obtienen diagramas de clases que representan estos modelos. Según, (Pressman, 2010)

**Figura N° 2. 11 Modelo de presentación de UWE**



**Fuente: (Pressman, 2010)**

## 2.9 ARQUITECTURA DEL SOFTWARE

Debe ser muy común, escuchar en la actualidad acerca de la Arquitectura de Software y sobre el rol que cumplen los Arquitectos en la implementación de soluciones, pero conocemos de forma detallada ¿qué es la Arquitectura de Software?

En las próximas líneas estoy tratando de brindar un panorama resumido, pero a la vez detallado de la Arquitectura de Software.

Definamos primero, que el diseño de la Arquitectura de un Software es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes forman el software, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada, cumpliendo con los criterios previamente establecidos; como seguridad, disponibilidad, eficiencia o usabilidad.

Durante el diseño de la arquitectura se tratan tópicos que puedan provocar un impacto importante en el éxito o fracaso de nuestro software. Son esenciales realizar las siguientes interrogantes para cubrir este punto:



- ❖ ¿En qué entorno se desplegará nuestro software?
- ❖ ¿Cómo se pondrá en producción nuestro software?
- ❖ ¿Cómo utilizarán los usuarios nuestro software?
- ❖ ¿Existen requisitos adicionales que el software debe cumplir? (Por ejemplo: seguridad, rendimiento, concurrencia, configuración, disponibilidad, entre otros)
- ❖ ¿Cuáles serían los cambios sobre la arquitectura propuesta, que impactarían al software durante o después de desplegarse?

Para diseñar la arquitectura de un software es de vital importancia tomar en cuenta los intereses de los distintos agentes que participan. Estos, son los usuarios del software, el propio software y los objetivos del negocio. Cada uno de ellos establece requisitos y restricciones que deben tomarse en cuenta para el diseño de la arquitectura, los que en algún momento podrían entrar en conflicto.

Para los usuarios es importante que el software responda a la interacción de una forma fluida, mientras que para los objetivos del negocio es importante que el software cueste poco. Los usuarios pueden querer que se implemente primero una funcionalidad útil para su trabajo del día a día, mientras que el software puede tener prioridad en que se implemente la funcionalidad que permita definir su estructura.

He aquí, que el trabajo del arquitecto es delinear los escenarios y requisitos de calidad importantes para cada agente así como los puntos clave que debe cumplir y las acciones o circunstancias que no deben ocurrir.

El objetivo final de la arquitectura es identificar los requisitos que producen un impacto en la estructura del software y reducir los riesgos asociados con la construcción del mismo. La arquitectura debe soportar los cambios futuros del software, del hardware y de funcionalidad demandada por los clientes (que ocurren muy a menudo). Del mismo modo, es responsabilidad del arquitecto analizar el impacto de sus decisiones de diseño y establecer un compromiso entre los diferentes requisitos de calidad así como entre los compromisos necesarios para satisfacer a los usuarios, al software y los objetivos del negocio.

Finalmente, resumamos que la Arquitectura de Software debería poseer las siguientes capacidades:

- Mostrar la estructura del software, pero ocultando los detalles.
- Concebir y diseñar todos los casos de uso.
- Satisfacer en la medida de lo posible los intereses de los agentes.
- Ocuparse de los requisitos funcionales y de calidad.
- Determinar el tipo de software a desarrollar.
- Determinar los estilos arquitecturales que se usarán.
- Tratar las principales cuestiones transversales.

Espero haber cubierto inicialmente puntos específicos, que nos permitan entender un poco más acerca de lo que es y significa la Arquitectura de Software en la actualidad. Según, (Sifuentes, 2018)

### 2.9.1 Patrón Modelo Vista Controlador

El MVC o Modelo Vista Controlador es un patrón de arquitectura de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir por un lado define los componentes para la representación de la información y por otro lado la interacción del usuario. Según, (Fontan, 2012)

- ❖ **Modelo:** se trata de la capa que trabaja con los datos, por lo que tiene los mecanismos para acceder a la información y también actualizar su estado. Estos datos estarán habitualmente en una base de datos.
- ❖ **Vista:** esta capa contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, es decir, el código que permitirá renderizar los estados de nuestra aplicación en HTML.
- ❖ **Controlador:** contiene el código necesario para responder a las acciones que solicita la aplicación, como por ejemplo realizar una compra o visualizar un elemento. Esta capa sirve de enlace entre la vista y el modelo.

**Figura N° 2. 12 Que es MVC?**

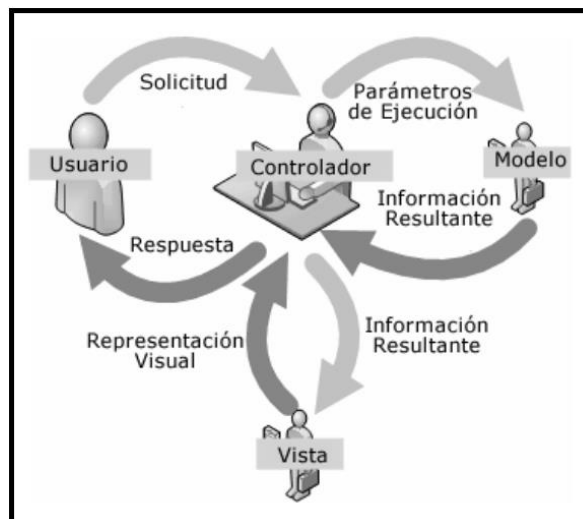


**Fuente: (Fontan, 2012)**

❖ **Ciclo de vida**

El ciclo de vida de MVC es normalmente representado por 3 capas presentadas anteriormente y el cliente (también conocido como usuario). El siguiente diagrama representa el ciclo de vida de manera sencilla. (Alvarez, 2014)

**Figura N° 2. 13 Ciclo de vida de un MVC**



**Fuente: (Alvarez, 2014)**

El primer paso empieza en el ciclo de vida empieza cuando el usuario hace una solicitud al controlador con información sobre lo que el usuario desea realizar. Entonces el Controlador decide a quien debe delegar la tarea y es aquí donde el Modelo empieza su

trabajo. En esta etapa, el Modelo se encarga de realizar operaciones sobre el información que maneja para cumplir con lo que le solicita el controlador. Una vez que termina su labor, le regresa al Controlador la información resultante de sus operaciones, el cual a su vez redirige a la vista. La Vista se encarga de transformar los datos en información visualmente entendible para el usuario. Finalmente, la representación gráfica es la transmitida de regreso al Controlador y este se encarga de transmitirla al usuario. El ciclo puede empezar nuevamente si el usuario así lo requiere. (Alvarez, 2014)

**Ventajas del MVC**, las principales ventajas de hacer uso del patrón MVC son:

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Permite el escalamiento de la aplicación en caso de ser requerido.

**Desventajas de MVC son:**

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos.

Cabe mencionar que la comparación de ventajas y desventajas de MVC puede ser un tema muy subjetivo y se puede presentar como tema de debate, sin embargo se tomó la decisión usando principalmente los puntos mencionados anteriormente ya que en

términos generales la balanza se inclina a favor del MVC en vez de en su contra. (Alvarez, 2014)

## **2.10 HERRAMIENTAS DE DESARROLLO**

### **❖ Servidor Apache:**

Los servidores web son uno de los principales pilares de Internet tal como lo conocemos hoy, son mediante ellos que despachan todas las páginas web y material multimedia que vemos día a día, también forman parte de la infraestructura de las aplicaciones móviles y servicios cloud.

Cuando en un navegador escribimos una dirección web nuestro navegador genera una petición HTTP a un servidor remoto que mediante un servicio llamado servidor web responde a esa petición con la página web que pretendemos visualizar, para entender mejor este complejo proceso puedes leer sobre cómo funciona un servidor web, donde lo explicamos con lujo de detalles. Según, (Leon, 2019)

Existen varios tipos de servidores web en el mercado y entre ellos está Apache, uno de los servidores más antiguos y si bien en los últimos años han surgido alternativas más modernas y eficientes que han ganado terreno al día de hoy Apache sigue uno servidor de referencia.

En un comienzo eran solo un grupo de parches para el servidor web NCSA HTTPd al que posteriormente sustituyó por completo al reescribir todo el código original del que ya no queda prácticamente nada dentro del código de Apache.

En definitiva, Apache es un servidor HTTP que permite servir contenido a las peticiones que vienen desde los clientes web (navegadores).

El servidor Apache nació a mediados de los años noventa y es sus mejores años alcanzo una cuota de mercado del 70% de las web, siendo el primer servidor que alojó más de 100 millones de sitios web. Según, (Leon, 2019)

### **2.10.1 Gestor de base de datos**

## ❖ **Base de datos**

El concepto de base, procedente del latín basis, tiene múltiples usos. El término puede utilizarse con referencia al sostén o fundamento de algo. Un dato, por otra parte, es una información concreta, un testimonio, una prueba o una documentación.

Con estas definiciones ya podemos detallar los alcances de la noción de base de datos, que se emplea con mucha frecuencia en el ámbito de la informática. Se conoce como base de datos al conjunto de informaciones que está organizado y estructurado de un modo específico para que su contenido pueda ser tratado y analizado de manera rápida y sencilla.

Las bases de datos, por lo tanto, presentan datos estructurados de acuerdo a diferentes parámetros. Al disponer la información de una cierta forma, el usuario puede encontrar aquello que busca con facilidad, a diferencia de lo que le sucedería si todos los datos estuvieran mezclados y sin ningún tipo de orden. Según, (Estela, 2019)

## ❖ **MySQL.**

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL.

MySQL es un sistema de administración de bases de datos

Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

MySQL es un sistema de administración de bases de datos relacionales

Una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando

se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales.

❖ **MySQL es Open Source**

Open Source significa que la persona que quiera puede usar y modificar MySQL. Cualquiera puede descargar el software de MySQL de Internet y usarlo sin pagar por ello. Inclusive, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si uno está incómodo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión de MySQL con una licencia comercial. Para mayor información, ver la página oficial de MySQL en la cual se proporciona mayor información acerca de los tipos de licencias. (Eduardo, 2007)

*Tabla 2 2 Comparación de bases de datos*

<b>SGBD</b>	<b>Características</b>	<b>Ventajas</b>	<b>Desventajas</b>
DB2	Es propietario Es propietario IBM Integra XML de forma nativa Es relacional Arquitectura similar a Oracle El SQL es muy potente Tiene muchos años	Multiplataforma-elimina tareas rutinarias, esto permite menor uso de recursos hardware Tiene una versión gratuita llamada DB2 Express-C Escalable Estable	No es tan robusto como Oracle Puede ser caro
MySQL	Propietaria y publica Portabilidad	Fácil de aprender y utilizar multiplataforma Código abierto Fácil configuración Veloz a realizar operaciones	El soporte para disparadores es muy básico
Oracle	Propietaria Portable Compatible Alto rendimiento	DBMS popular Oracle ofrece porte técnico Permite la gestión de múltiples bases de datos	Una mala configuración ofrece resultados desfavorables

Postgre SQL	Incluye herencia entre las tablas Incorpora estructuras de arrays	Ahora en costos Instalación limitada Estabilidad Gran capacidad de almacenamiento	Lento en inserciones y actualizaciones Ofrece soporte en línea
SQLite	Dominio publico DBMS relacional Algunos lenguajes de programación lo incluyen en sus módulos o bibliotecas	Multiplataforma -muchos lenguajes de programación tiene soporte o módulos para sqlite Pequeño tamaño	Su límite es de 2 terabytes su base de datos

**Fuente:** (Eduardo, 2007)

## 2.10.2 Lenguaje de programación

### ❖ PHP

Siglas en inglés de “Hypertext Pre-Processor” que al traducirlo al español pierde un poco el sentido, mejor lo analizamos y encontramos que significa “Lenguaje de Programación Interpretado”. Este lenguaje es al que le debemos la visualización de contenido dinámico en las páginas web. Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Según, (Alvares, 2001)

*Tabla 2 3 Lenguajes de programación*

LENGUAJE	CARACTERÍSTICAS	FORTALEZAS	DEBILIDADES
PHP	Utilizado para generar páginas web dinámicas Se ejecuta en el servidor	Su sintaxis es muy similar a otros lenguajes Fácil	Necesita un servidor para funcionar



	<p>Los usuarios no pueden ver el código PHP únicamente reciben en sus navegadores código HTML</p> <p>Las páginas que genera son visibles para prácticamente cualquier navegador y computadora o dispositivos móviles que pueda interpretar el HTML.</p> <p>Versiones reciente permiten la POO</p> <p>Lenguaje de alto nivel</p>	<p>Es un lenguaje muy popular tiene una comunidad muy grande</p> <p>Rápido</p> <p>Multiplataforma</p> <p>Maneja base de datos</p> <p>Bastante documentado</p> <p>Libre y gratuito.</p> <p>Varias funciones</p> <p>Tiene muchos frameworks que facilitan el desarrollo en este lenguaje.</p>	<p>La POO es deficiente para aplicaciones grandes</p> <p>Todo el trabajo se realiza el en servidor y mucha información o solicitudes pueden ser ineficiente.</p>
RUBY	<p>Orientado a objetos</p> <p>Lenguaje de alto nivel</p> <p>Sintaxis similar a Python y Perl</p> <p>Lenguaje para la creación de aplicaciones de escritorio y aplicaciones web.</p>	<p>Diferencia entre mayúsculas y minúsculas</p> <p>Maneja excepciones</p> <p>Puede cargar librerías si el sistema operativo lo permite</p> <p>Multiplataforma</p> <p>Portátil</p>	<p>es relativamente nuevo y no cuenta con mucha documentación en comparación con otros lenguajes de programación</p>
Python	<p>Permite la creación de todo tipo de programas incluso sitios web</p> <p>no requiere de compilación es un código interpretado</p>	<p>libre y código fuente abierto</p> <p>lenguaje de propósito general</p> <p>multiplataforma</p> <p>orientado a objetos</p> <p>portable</p>	<p>los lenguajes interpretados suelen ser relativamente lentos</p>
JavaScript	<p>es un lenguaje interpretado</p> <p>es similar a java</p> <p>es orientado a objetos</p>	<p>los script tiene capacidad limitada por razones de seguridad</p> <p>se ejecuta del lado del cliente</p> <p>lenguaje de scripting seguro y fiable</p>	<p>No soporta herencias</p> <p>Código visible por cualquier usuario</p> <p>El código debe ser descargado completamente</p>
C++	<p>Orientado a objetos</p> <p>Rápido</p>	<p>Ideal para sistemas robustos</p> <p>IDEs de desarrollo son DEV C++, BORLAND C, TURBO C</p>	<p>No soporta creación de aplicaciones web</p> <p>Complejo visualmente</p>

Fuente: (Alvares, 2001)

## ❖ JavaScript

JavaScript consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones en el lado del cliente, estando embebido en el código fuente de la página web

Técnicamente, constituye un dialecto del estándar ECMAScript, propuesto por la entidad internacional de estándares de información y comunicación ECMA International y diseñado inicialmente por Netscape y, posteriormente, por la Fundación Mozilla. También constituye un estándar ISO.

Si bien recibe en su sintaxis y forma de trabajo una mayor herencia del lenguaje C, también adopta nombres y convenciones que encontramos en Java, aunque se quedan en eso, en parecidos “cosméticos”.

JavaScript permite, en una página web, crear elementos como cuadros de diálogo, recoger información entrada por el usuario y mandarla al servidor para ser procesada. Con el tiempo, JavaScript ha evolucionado, incluyendo funcionalidades de otros lenguajes más avanzados, como clases para programación orientada a objetos, o módulos para la reutilización de código fuente. Según (Gonzales G. A., 2016)

Pero pese a estos cambios y actualizaciones, JavaScript mantiene la compatibilidad hacia atrás:

Es un lenguaje estructurado e imperativo, tomando como modelo la estructura de programación del lenguaje C

Tipos dinámicos, de forma que cuando definimos el valor de la variable, se define el tipo de datos

Orientado a objetos, con funciones que son consideradas como objetos

Los objetos y métodos mediante los que el lenguaje puede interactuar con el usuario se encuentran limitados por el entorno de ejecución como, por ejemplo, el browser

De igual forma que Perl, es compatible con expresiones regulares, (Gonzales G. A., 2016)

## ❖ jQuery

Se denomina jQuery a una librería o biblioteca de JavaScript que facilita la programación en este lenguaje. Por lo general se emplea para añadir elementos interactivos a una página web sin necesidad de tener que programar demasiado.

En el caso de jQuery, la biblioteca es de código abierto (todos pueden contribuir al desarrollo ya que el acceso al código fuente es libre) y está basada en el lenguaje JavaScript. Lo que hace es ofrecer diversas funcionalidades que, de otro modo, exigirían desarrollar más código, por lo tanto ayudan a ahorrar tiempo. Según, (Porto J. P., 2018)

### 2.10.3 Framework

## ❖ CodeIgniter

CodeIgniter es un framework para el desarrollo de aplicaciones en PHP, 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. Según, (Colectiva, 2015)

### **Ventajas de usar el framework CodeIgniter:**

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

*Tabla 2 4 Framework de PHP*

Nombre	Descripción	Ventajas
Kumbia PHP	Es un framework para aplicaciones Web libre escrito en PHP5 basado en las prácticas de desarrollo web como DRY y el principio KISS para software comercial y educativo	Hace la mayor parte del trabajo y se ocupa de los detalles. Manteniibilidad de la aplicación. Curva de aprendizaje minima. Su estructura de directorio favorece la proactividad.
Code Igniter	Es un fremework de aplicación Web para ayudar a desarrollar programas en php, el objetivo de la aplicación es ayudar a los desarrolladores de los proyectos de código, a desarrollar más rápido que escribir código desde cero.	Ofrece un marco con una pequeña huella. Tiene exelente rendimiento Documentación completa Aplica compatibilidad con alojamiento estándar. Casi utiliza cero configuraciones.
Zend framework	Robusta implementación en el MVC, da al usuario una fácil manipulación del código creando para reutilizarlo	Código abierto, orientado a objetos, modelo visto controlador, gran rendimiento.
Yii	Un poco más fácil de aprender que otros framework, cuenta con foro y IRC que proporciona ayuda al usuario,	Software libre, patrón de diseño modelo vista controlador, integración JQuery
Laravel	Además de manejar MVC también cuenta con el uso de mapeo objeto-relacional.	Administrador de extensiones, completa documentación, contiene ORM.
Symfony	Fácil de instalar y usar en la mayoría de plataformas, su extensión permite una mejor integración de bibliotecas,	Contiene una gran cantidad de plugins, integración con AJAX
Cake PHP	Facilita al usuario la interacción con la base de datos. Contiene componente de seguridad	Compatible con PHP 4 y 5, validación integrada, sintetización de datos.

**Fuente: (Alvares, 2001)**

### ❖ Framework Bootstrap

Bootstrap es conjunto conceptos, prácticas y criterios (framework) desarrollado por Mark Otto y Jacob Thornton dentro de Twitter con la intención de estandarizar el conjunto de herramientas que utilizaban todos los involucrados en el desarrollo del front-end.

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta

automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

De esta manera crearon un conjunto de librerías JavaScript y CSS que toda la compañía debía usar evitando que las partes desarrolladas por un equipo no pudiesen ser mantenidas por otros.

Bootstrap nos ayuda a maquetar un sitio web con rapidez y, sobretodo, ayudándonos a que el diseño sea correcto y usable tanto en dispositivos convencionales como en los táctiles (responsive web design). Para hacerlo, nos ofrece una serie de estilos CSS y librerías JavaScript que nos ayudarán de una manera rápida a desarrollar nuestro sitio web y sobretodo es recomendable para el desarrollo de prototipos y tener un tiempo de respuesta realmente bueno.

La base de Bootstrap es un diseño de rejilla dividido en 12 columnas con un ancho total de 940 píxeles. De esta manera, la maquetación se simplifica haciendo que solamente se deban encajar correctamente nuestras secciones dentro de esta parrilla y Bootstrap se encargará de que la visualización sea correcta en todos los dispositivos. Nos ahorramos, todo el trabajo de revisar con varios navegadores, el desarrollo de CSS específicos para móviles, etc. Según, (Gonzales C. R., 2014)

#### **2.10.4 Herramientas de diseño**

##### **❖ HTML5**

HTML5 (HyperText Markup Language) es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML5 podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5.

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- ❖ Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.

- ❖ Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.

Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta página referencia numerosos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función.

- ❖ Semántica: Permite describir con mayor precisión cuál es su contenido.
- ❖ Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.
- ❖ Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.
- ❖ Multimedia: Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- ❖ Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- ❖ Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- ❖ Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo. Según, (Martin, 2019)

### ❖ **CSS3**

CSS es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML.

CSS3 es una tecnología que ha tenido una evolución en el tiempo, que actualmente se encuentra en su versión 3, como su propio nombre indica.

Sus siglas corresponden a “Cascading Style Sheets”, que tiene el siguiente significado:

Cascading, que significa que los estilos que aplicamos a los elementos de una página web se propagan a los elementos que contiene, se propagan en cascada.

Style, porque mediante CSS lo que hacemos es aplicar estilos visuales a los distintos elementos de nuestra página web.

Sheets, que significa hojas, porque los estilos de una página web se añaden en ficheros aparte, en ficheros con la extensión .css de manera general.

CSS ha ido evolucionando desde hace 25 años a la versión actual, que es CSS3, pero su función es la misma, sirve para dar estilos visuales a las páginas web. Según, (Jimenez, 2019)

## 2.11 CALIDAD DEL SOFTWARE

"La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario". (IEEE, Std. 610-1990). "Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario". Según, (Pressman, 2002)

- ❖ **CALIDAD:** La calidad es un término que actualmente se encuentra en multitud de contextos y con el que se busca despertar en quien lo escucha una sensación positiva, transmitiendo la idea de que algo es mejor. La palabra calidad tiene muchos significados. La calidad de un producto o servicio es la percepción que el cliente tiene del mismo, es una fijación mental del consumidor que asume conformidad con un producto o servicio determinado, que solo permanece hasta el punto de necesitar nuevas especificaciones. La calidad es la capacidad de un producto o servicio para satisfacer las necesidades del cliente o usuario.
  
- ❖ **MÉTRICA:** Históricamente se habló de métrica en referencia a los sistemas que existían para escribir versos diferenciados en base al número de sílabas que

contenía cada verso, así como en referencia al estudio y “medición” de la cantidad de sílabas y estrofas que contenían los versos.

En informática, el término métrica hace referencia a la medición del software en base a parámetros predeterminados, como puede ser el número de líneas de código de que consta o el volumen de documentación asociada. A veces en vez de hablar de métrica se usa el término “Indicadores” del software. Algunos ingenieros lo usan como sinónimos mientras que otros les atribuyen significados distintos.

Algunas métricas o indicadores pueden ser:

- a. ***Índice de productividad = tamaño / esfuerzo = líneas de código generado / horas trabajadas.***
- b. ***Tasa de defectos = defectos / tamaño = número de errores / líneas de código generadas.*** Según, (Pressman, 2002)

**Métricas de calidad del software:** Los sistemas de métricas de calidad del software tradicionales se han centrado fundamentalmente en las métricas de procesos, de productos y de recursos.

Los sistemas de métricas hoy en día son los usados en los perfiles o aplicaciones para probar las aplicaciones. Este tipo de aplicaciones usan sistemas de métricas en tiempo de ejecución para medir tiempos, buscar cuellos de botella en las aplicaciones, medir capacidades máximas, etcétera.

Así, las métricas tratan de servir de medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento.

Los tres objetivos fundamentales de la medición son (Fenton y Pfleeger, 1997):

- ❖ Entender qué ocurre durante el desarrollo y el mantenimiento.
- ❖ Controlar qué es lo que ocurre en nuestros proyectos.
- ❖ Mejorar nuestros procesos y nuestros productos.



## ❖ ISO 9000

ISO 9000 consiste en una serie de normas inherentes a la calidad y a la administración continua de la calidad, que se aplica en las organizaciones, cualquiera sea su naturaleza, que están dedicadas a la producción de bienes y de servicios.

A grandes rasgos la norma ISO 9000 se propone: estandarizar la actividad del personal documentando su tarea; trabajar en orden a satisfacer las necesidades de los clientes; monitorear y medir los procesos involucrados en todos los niveles; evitar caer en los re procesos; promover la eficiencia para conseguir objetivos propuestos; promover que los productos mejoren constantemente, entre otros. (Ucha, 2012)

### **2.11.1 Estándar ISO/IEC 9126**

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado "Informationtechnology-Software productevaluation-Qualitycharacteristics and guidelinesfortheir use"; o también conocido como ISO 9126 (o ISO/IEC 9126).

Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad.

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así:

## ❖ Métricas de Calidad del Modelo de ISO-9126

El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software (Pressman, 2002).

Este estándar es una simplificación del Modelo de McCall (Losavio, 2003), e identifica seis características básicas de calidad que pueden estar presentes en cualquier producto de software. El estándar provee una descomposición de las características en sub características, que se muestran en la Tabla

*Tabla 2 5 Características de Calidad – Modelo ISO/IEC 9126*

CARACTERÍSTICAS	SUB CARACTERÍSTICAS	
Funcionalidad	<ul style="list-style-type: none"> <li>○ Adecuación</li> <li>○ Exactitud</li> <li>○ Interoperatividad</li> </ul>	<ul style="list-style-type: none"> <li>○ Seguridad</li> <li>○ Conformidad</li> </ul>
Fiabilidad	<ul style="list-style-type: none"> <li>○ Madurez</li> <li>○ Tolerancia a fallas</li> </ul>	<ul style="list-style-type: none"> <li>○ Recuperabilidad</li> <li>○ Conformidad</li> </ul>
Usabilidad	<ul style="list-style-type: none"> <li>○ Comprensibilidad</li> <li>○ Facilidad de aprendizaje</li> <li>○ Operabilidad</li> </ul>	<ul style="list-style-type: none"> <li>○ Atracción</li> <li>○ Conformidad</li> </ul>
Eficiencia	<ul style="list-style-type: none"> <li>○ Comportamiento temporal</li> <li>○ Utilización de recursos</li> </ul>	<ul style="list-style-type: none"> <li>○ Conformidad</li> </ul>
Mantenibilidad	<ul style="list-style-type: none"> <li>○ Analizabilidad</li> <li>○ Cambiabilidad</li> <li>○ Estabilidad</li> </ul>	<ul style="list-style-type: none"> <li>○ Capacidad de prueba</li> <li>○ Conformidad</li> </ul>
Portabilidad	<ul style="list-style-type: none"> <li>○ Adaptabilidad</li> <li>○ Facilidad de instalación</li> <li>○ Coexistencia</li> </ul>	<ul style="list-style-type: none"> <li>○ Remplazabilidad</li> <li>○ Conformidad</li> </ul>

**Fuente: (Pressman, 2002)**

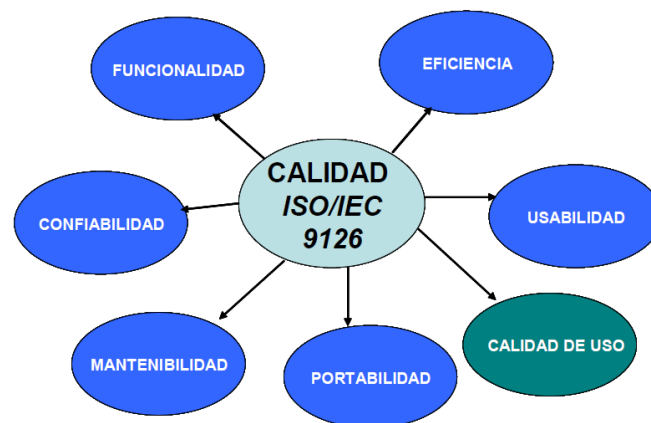
Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo

tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así: según, (Pressman, 2002)

#### ❖ Norma de Evaluación ISO/IEC 9126

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en varios indicadores; estas se pueden medir por métrica interna o externa.

**Figura Nº 2. 14 Evaluación ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

#### ❖ Evaluación Interna, externa y Calidad de Uso ISO/IEC 9126

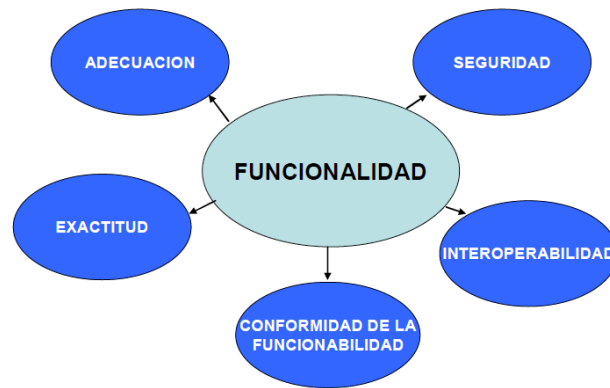
Las definiciones se dan para cada característica y sub-característica de calidad del software que influye en la calidad. Para cada característica y sub-característica, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y sub-características se pueden medir externamente por la capacidad del sistema que contiene el software. (Ardila, 2013)

### 1. FUNCIONALIDAD

Funcionalidad 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 muestra la característica de Funcionalidad y las sub-características que cubre:

**Figura N° 2. 15 Característica de funcionalidad ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

❖ **Característica de funcionalidad.**

La funcionalidad se divide en 5 criterios:

**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.

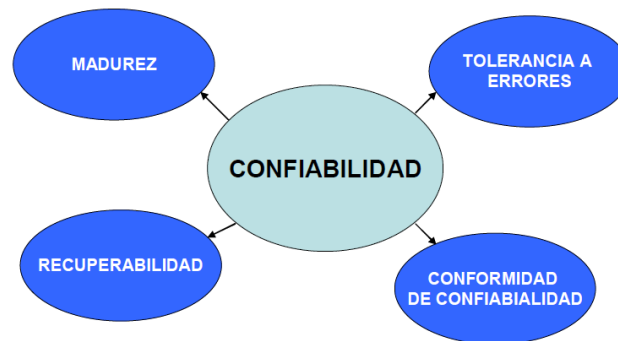
**Seguridad:** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

**Conformidad de la funcionalidad:** La capacidad del software de cumplir los estándares referentes a la funcionalidad. (Ardila, 2013)

## 1. CONFIABILIDAD.

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. En este caso a la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida.

**Figura Nº 2. 16 Característica de confiabilidad ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

### ❖ Característica de Confiabilidad

La confiabilidad se divide en 4 criterios:

**Madurez:** La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.

**Tolerancia a errores:** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

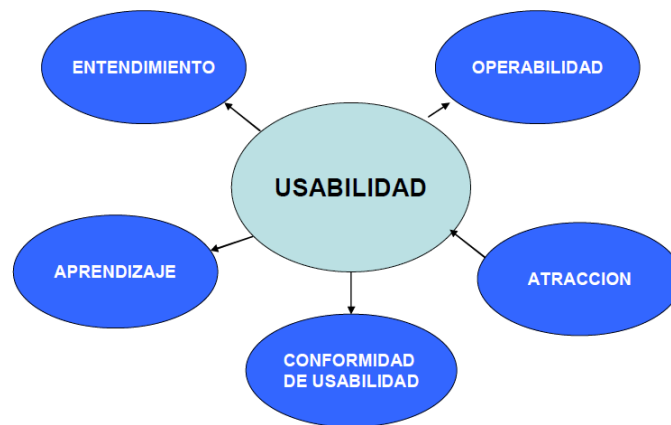
**Recuperabilidad:** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

**Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad. (Ardila, 2013)

## 4. 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.

**Figura Nº 2. 17 Característica de usabilidad ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

### ❖ Característica de Usabilidad

La usabilidad se divide en 5 criterios.

**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.

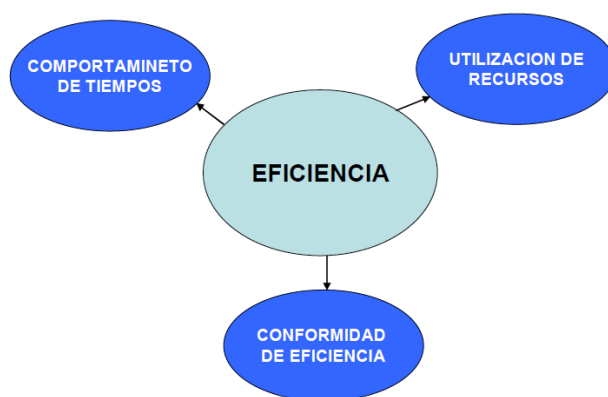
**Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

Conformidad de uso: La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad. (Ardila, 2013)

#### 4. EFICIENCIA

La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

**Figura N° 2. 18 Característica de eficiencia ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

##### ❖ **Característica de Eficiencia**

La eficiencia se divide en 3 criterios:

**Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

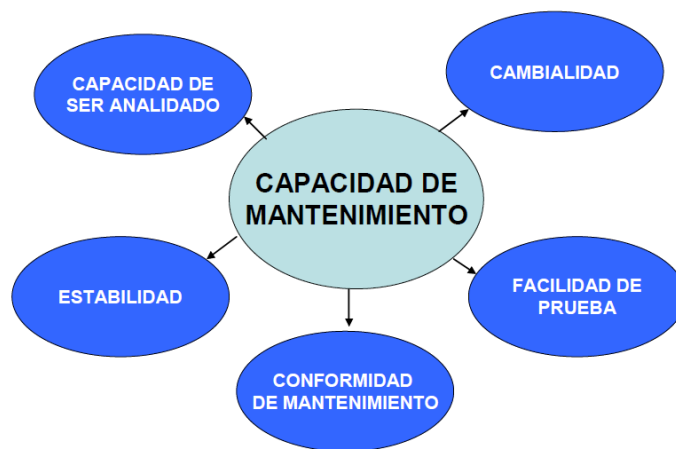
**Utilización de recursos:** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

**Conformidad de eficiencia:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia. (Ardila, 2013)

## 5. CAPACIDAD DE MANTENIMIENTO

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.

**Figura N° 2. 19 Característica de capacidad de mantenimiento ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

### ❖ Característica de Mantenimiento

El mantenimiento se divide en 5 criterios:

**Capacidad de ser analizado:** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

**Cambiabilidad:** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

**Estabilidad:** La forma como el software evita efectos inesperados para modificaciones del mismo.



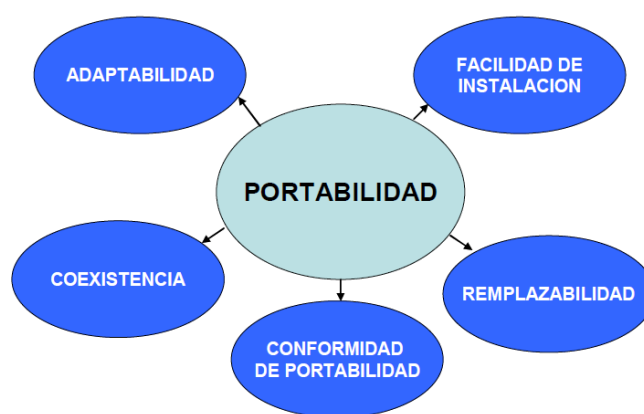
**Facilidad de prueba:** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

**Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento. (Ardila, 2013)

## 6. PORTABILIDAD

La capacidad que tiene el software para ser trasladado de un entorno a otro.

**Figura N° 2. 20 Característica de portabilidad ISO/IEC 9126**



**Fuente: (Ardila, 2013).**

### ❖ Característica de portabilidad

La usabilidad se divide en 5 criterios:

**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 que tiene el software para cumplir con los estándares relacionados a la portabilidad. (Ardila, 2013)

### ❖ CALIDAD EN USO

Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud.

La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.

**Figura Nº 2. 21 Calidad de uso de Software ISO/IEC 9126**



**Fuente: (Ardila, 2013)**

### ❖ Característica Calidad de uso

La calidad de uso se divide en 4 criterios:

**Eficacia:** La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.

**Productividad:** La forma como el software permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto específico de uso. Para una empresa es muy importante que el software no afecte a la productividad del empleado

**Seguridad:** Se refiere al que el Software no tenga niveles de riesgo para causar daño a las personas, instituciones, software, propiedad intelectual o entorno. Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (Incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.

**Satisfacción:** La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo.

❖ **Métricas basadas en la función**

La métrica punto función (PF) se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. PF se deriva empleando una relación empírica basada en medidas contables del dominio de la información del software y las evaluaciones de la complejidad de este.

Sea los valores de dominio de la información según la Tabla

*Tabla 2 6 Factor de Ponderación*

Parámetros de medición		Cuenta	Factores de ponderación			Total
			Simple	Medio	Complejo	
1	Nro. de entradas	x	3	4	6	=
2	Usuario	x	3	4	6	=
3	Nro. de Peticiones de usuario	x	3	4	6	=
4	Nro. de Archivos	x	7	10	15	=
5	Nro. de Interfaces Externas	x	5	7	10	=
Cuenta Total						

**Fuente: (Pressman, 2002)**

Número de entradas de usuario: Se cuenta cada entrada del usuario que proporcione al software diferentes datos aplicados a la aplicación. Las entradas deben ser distinguidas de las peticiones que se contabilizan por separado.

Número de Salidas del usuario: Se encuentra cada salida que proporciona al usuario información orientada a la aplicación. En este contexto se refieren a informes, pantallas, mensajes de error.

Número de peticiones al usuario: Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva se cuenta cada petición por separado.

Numero de archivos: Se cuenta cada archivo maestro lógico es decir una agrupación lógica de datos que puede ser una parte en gran base de datos o un archivo independiente.

Numero de interfaces externas: Se cuentan todas las interfaces legibles por la maquina por ejemplo: archivo de datos, en cintas o discos que son utilizados para transmitir información a otro sistema.

A cada conteo se le asocia un valor de complejidad, no obstante la determinación de la complejidad es un poco subjetiva.

Para Calcular los puntos función se usa la siguiente ecuación.

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

Cuenta Total es la suma de todas las entradas de PF obtenidas de la tabla anterior Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la tabla.

*Tabla 2 7 Factor de Escala*

0	1	2	3	4	5
Sin influencia	Incidental	Moderno	Medio	Significativo	Esencial

**Fuente: (Pressman, 2002)**

*Tabla 2 8 Valoración Métrica Punto Función*

-----	Preguntas	F1
1	¿Requiere el sistema copia de seguridad y recuperación?	
2	¿Requiere comunicación de datos?	
3	¿Existen funciones de procesos distribuidos?	
4	¿El rendimiento es crítico?	
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?	
6	¿Entrada de datos EN LÍNEA?	
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se llevan a cabo sobre múltiples pantallas o variadas opciones?	
8	¿Se actualizan los archivos maestros de forma interactiva?	
9	¿Son complejas de las entradas de salidas de archivos?	
10	¿Lógica del proceso Complejo?	
11	¿Se diseña el código para ser reutilizable?	
12	¿Están incluidas en el diseño conversiones de instalación?	
13	¿Instalaciones Múltiples?	
14	¿Facilidad de Cambios?	

**(Pressman, 2002)**

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

## **2.12 ANÁLISIS DE COSTO DE SOFTWARE**

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismos, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente.

La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo incidiendo en la gestión empresarial.

En la actualidad existen un conjunto de métricas que no se utilizan, y que pueden ser aplicables a cualquier tipo de proyecto de software para calcular el costo de los mismos.

Esta investigación propone el diseño de un conjunto de métricas para calcular el costo en el proceso de desarrollo de software.

Hoy en día existen diversas herramientas y metodologías que nos permiten estimar costos y existen muchos factores que afectan las estimaciones de costo como:

- ❖ Incertidumbre en los requerimientos.
- ❖ Términos contractuales rígidos.
- ❖ Salud financiera (ganar licitaciones sacrificando costo y tiempo).
- ❖ Falta de experiencia con “X” tecnología. Según: (Roque, 2014)

### **2.12.1 COCOMO II**

Modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por ConstructiveCostModel (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.

#### **❖ CARACTERÍSTICAS**

- ❖ Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
- ❖ Representa el más extenso modelo empírico para la estimación de software.
- ❖ Existen herramientas automáticas que estiman costos basados en COCOMO como ser: COCOMO 81.

#### **❖ Objetivos para la construcción de COCOMO II**

- ❖ Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- ❖ Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.

- ❖ Implementar una herramienta de software que soportara el modelo.
- ❖ Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

#### ❖ **Modelos de COCOMO II**

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

1. **Modelo de composición de aplicación.** Utilizado durante las primeras etapas de la Ingeniería del software, donde el prototipado de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
2. **Modelo de fase de diseño previo.** Utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
3. **Modelo de fase posterior a la arquitectura.** Utilizado durante la construcción del software. (Mercado, 2018)

#### ❖ **Estimación del Esfuerzo**

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona fulltime, requeridos para desarrollar el proyecto. (Gomez, 2002)

#### ❖ **Modelo Composición de Aplicación**

La fórmula propuesta en este modelo es la siguiente:

$$PM = NOP / PROD$$

Donde:

**NOP** (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$$\text{NOP} = \text{OP} \times (100 - \% \text{reuso}) / 100$$

**OP** (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto

**%reuso**: Porcentaje de reuso que se espera lograr en el proyecto

**PROD**: Es la productividad promedio determinada a partir del análisis de datos de proyectos en [Banker 1994], mostrada en

*Tabla 2 9 Productividad para el modulo Composición de Aplicación*

Experiencia y capacidad de los desarrolladores	Muy bajo	Bajo	Normal	Alto	Muy alto
Madurez y Capacidad del ICASE	Muy bajo	Bajo	Normal	Alto	Muy alto
<b>PROD</b>	4	7	13	25	50

**Fuente: (Gomez, 2002)**

### ❖ Modelo Diseño Temprano

Este modelo se usa en las etapas tempranas de un proyecto de software, cuando se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o detalles específicos del proceso a utilizar. Este modelo podría emplearse tanto en productos desarrollados en sectores de Generadores de Aplicación, Sistemas Integrados o Infraestructura.

El modelo de Diseño Temprano ajusta el esfuerzo nominal usando siete factores de costo. La fórmula para el cálculo del esfuerzo es la siguiente: (Gomez, 2002)

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^7 EM_i$$



$$PM_{nominal} = A \times (KSLOC)^B$$

$$B = 1.01 + 0.01 \times \sum_{j=1}^5 W_j$$

Donde:

- ❖ **PM<sub>Estimado</sub>**, es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.
- ❖ **KSLOC** es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.
- ❖ **A** es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (**A=2.94**).
- ❖ **B** es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y diseconomías de escala producidas cuando un proyecto de software incrementa su tamaño.
- ❖ **EM<sub>i</sub>** corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers). Cada factor se puede clasificar en seis niveles diferentes que expresan el impacto del multiplicador sobre el esfuerzo de desarrollo. Esta escala varía desde un nivel Extra Bajo hasta un nivel Extra Alto. Cada nivel tiene un peso asociado. El peso promedio o nominal es 1.0. Si el factor provoca un efecto nocivo en el esfuerzo de un proyecto, el valor del multiplicador correspondiente será mayor que 1.0, caso contrario el multiplicador será inferior a 1.0. (Gomez, 2002)

Clasificados en categorías, los 7 Multiplicadores de Esfuerzo son:

- ❖ Del Producto
  - RCPX: Confiabilidad y Complejidad del producto
  - RUSE: Reusabilidad Requerida
- ❖ De la Plataforma

- PDIF: Dificultad de la Plataforma
- ❖ Del Personal
  - PERS: Aptitud del Personal
  - PREX: Experiencia del Personal
- ❖ Del Proyecto
  - FCIL: Facilidades
  - SCED: Cronograma de Desarrollo Requerido

**Figura N° 2. 22 Multiplicares del esfuerzo del modelo diseño temprano.**

	XLO	VLO	LO	NOM	HI	VHI	XHI
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
PDIF	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX
USR1	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX
USR2	XXXX	1.00	1.00	1.00	1.00	1.00	XXXX

**Fuente:** (Gomez, 2002)

❖ **Modelo Post-Arquitectura**

Es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software incluidos en las áreas de Sistemas Integrados, Infraestructura y Generadores de Aplicaciones.

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente

el producto de software bajo desarrollo. La fórmula para el cálculo del esfuerzo es la siguiente: (Gomez, 2002)

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^7 EM_i$$

#### ❖ **Modelos de Estimación de costo con COCOMO II.**

En la estimación del tamaño de software COCOMO II utiliza tres técnicas:

- ❖ Puntos Objeto,
- ❖ Puntos Función No Ajustados y
- ❖ Líneas de Código Fuente.

Además, se emplean otros parámetros relativos al tamaño que contemplan aspectos tales como: reusó, reingeniería, conversión y mantenimiento.

**Puntos objeto:** El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

1. Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
2. Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil)
3. Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
4. Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

**Figura N° 2. 23 Puntos Objeto**

Para Pantallas			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 ( < 2 servidor < 3 cliente)	Total < 8 ( < 2 - 3 servidor < 3 - 5 cliente)	Total 8 + ( > 3 servidor < 5 cliente)
< 3	Simple	Simple	Media
3 - 7	Simple	Media	Difícil
> 8	Media	Difícil	Difícil
Para Reportes			
Cantidad de Vistas Contenidas	Cantidad y fuente de las tablas de datos		
	Total < 4 ( < 2 servidor < 3 cliente)	Total < 8 ( < 2 - 3 servidor < 3-5 cliente)	Total 8 + ( > 3 servidor < 5 cliente)
0 o 1	Simple	Simple	Media
2 o 3	Simple	Media	Difícil
4 +	Media	Difícil	Difícil

**Fuente:** (Gil, 2012)

**Puntos Función:**

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura.

Los puntos función están basados en información disponible en las etapas tempranas del ciclo de vida del desarrollo de software.

COCOMO II considera solamente UFP (Puntos Función no ajustados).

$$FP = UFP \times TCF$$

Donde UFP: Puntos Función no Ajustados

TCF: Factor de Complejidad Técnica

Para calcular los UFP, se deben identificar los siguientes elementos:

- ❖ **Entradas Externas (Inputs):** Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.

- ❖ **Salidas Externas (Outputs):** Salida de datos de usuario o de control que deja el límite del sistema de software.
- ❖ **Archivo Lógicos Internos (Archivos):** Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.
- ❖ **Archivos Externos de Interface (Interfaces):** Archivos transferidos o compartidos entre sistemas de software.
- ❖ **Solicitudes Externas (Queries):** Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.

Una vez identificados los elementos se clasifican de acuerdo al grado de complejidad en: bajo, promedio o alto. Se asigna un peso a cada ítem según el tipo y el grado de complejidad correspondiente. Finalmente, los UFP son calculados sumando los pesos de todos los ítems identificados. (Gil, 2012)

#### ❖ **Líneas de Código Fuente (SLOC):**

El objetivo es medir la cantidad de trabajo intelectual puesto en el desarrollo de un programa. Definir una línea de código es difícil debido a que existen diferencias conceptuales cuando se cuentan sentencias ejecutables y de declaraciones de datos en lenguajes diferentes.

A los efectos de COCOMO II, se eliminan las categorías de software que consumen poco esfuerzo. Así no están incluidas librerías de soporte, sistemas operativos, librerías comerciales, etc., ni tampoco el código generado con generadores de código fuente.

#### ❖ **Conversión de Puntos Función a Líneas de Código Fuente (SLOC):**

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a líneas de código fuente considerando el lenguaje de implementación.

Por otro lado, el modelo COCOMO (COConstructiveCOstMOdel) Calcula esfuerzo y coste en función del tamaño del programa (LDC). COCOMO está definido para tres tipos de proyectos de Software:

1. Modo orgánico: Proyectos pequeños y sencillos, con equipos de experiencia en la aplicación y requisitos poco rígidos.
2. Modo semi-acoplado: Proyectos intermedios (más complejos), con equipos que poseen variados niveles de experiencia y requisitos más rígidos.
3. Modo empotrado: Proyectos que deben ser desarrollados en un conjunto de Hardware, Software y restricciones muy grandes. (Gil, 2012)

**Figura N° 2. 24 Detalles de coeficientes de COCOMO II**

PARA PUNTOS DE FUNCIÓN:			COCOMO BASICO																
DOMINIO	COMPLEJIDAD	PESO	Modo (Tipo de Proyecto)	a	b	c	d												
Salidas	Alta	7	Orgánico	2.4	1.05	2.5	0.38												
	Media	5																	
	Baja	4																	
Entradas	Alta	6	Semiacoplado	3.0	1.12	2.5	0.35												
	Media	4																	
	Baja	3																	
Consultas	Alta	7	Empotrado	3.6	1.20	2.5	0.32												
	Media	5																	
	Baja	4																	
Archivo Interno	Alta	15	$E = a * (KLDC)^b$ $D = c * (E)^d$																
	Media	10																	
	Baja	7																	
Archivo Externo / Interfaces	Alta	10	<b>COCOMO INTERMEDIO</b> <table border="1"> <thead> <tr> <th>Modo (Tipo de Proyecto)</th> <th>a</th> <th>b</th> </tr> </thead> <tbody> <tr> <td>Orgánico</td> <td>3.2</td> <td>1.05</td> </tr> <tr> <td>Semiacoplado</td> <td>3.0</td> <td>1.12</td> </tr> <tr> <td>Empotrado</td> <td>2.8</td> <td>1.20</td> </tr> </tbody> </table>					Modo (Tipo de Proyecto)	a	b	Orgánico	3.2	1.05	Semiacoplado	3.0	1.12	Empotrado	2.8	1.20
	Modo (Tipo de Proyecto)	a						b											
	Orgánico	3.2	1.05																
Semiacoplado	3.0	1.12																	
Empotrado	2.8	1.20																	
Media	7																		
Baja	5																		

$$E = a * (KLDC)^b * FAE \quad D = 2.5 * (E)^{0.38}$$

Fuente: (Gil, 2012)

#### ❖ Atributos de Coste

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula el significado de los atributos es el siguiente, según su tipo de:

a. SOFTWARE

- ❖ RELY: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).
- ❖ DATA: tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$ , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- ❖ CPLX: representa la complejidad del producto.

b. HARDWARE

- ❖ TIME: limitaciones en el porcentaje del uso de la CPU.
- ❖ STOR: limitaciones en el porcentaje del uso de la memoria.
- ❖ VIRT: volatilidad de la máquina virtual.
- ❖ TURN: tiempo de respuesta requerido.

c. PERSONAL

- ❖ ACAP: calificación de los analistas.
- ❖ AEXP: experiencia del personal en aplicaciones similares.
- ❖ PCAP: calificación de los programadores.
- ❖ VEXP: experiencia del personal en la máquina virtual.
- ❖ LEXP: experiencia en el lenguaje de programación a usar.

d. PROYECTO

- ❖ MODP: uso de prácticas modernas de programación.
- ❖ TOOL: uso de herramientas de desarrollo de software.
- ❖ SCED: limitaciones en el cumplimiento de la planificación El valor de cada atributo, de acuerdo a su calificación, se muestra en la siguiente tabla

Tabla 2 10 Atributos de Coste

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0.75	0.88	1.00	1.15	1.40	
Tamaño de Base de datos		0.94	1.00	1.08	1.16	
Complejidad	0.70	0.85	1.00	1.15	1.30	1.65
Atributos de Hardware						
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de memoria virtual			1.00	1.06	1.21	1.56
Volatilidad de memoria virtual		0.87	1.00	1.15	1.30	
Tiempo de respuesta		0.87	1.00	1.07	1.15	
Atributos de personal						
Capacidad de análisis	1.46	1.19	1.00	0.86	0.71	
Experiencia en la aplicación	1.29	1.13	1.00	0.91	0.82	
Calidad de los programadores	1.42	1.17	1.00	0.80	0.70	
Experiencia en la máquina virtual	1.21	1.10	1.00	0.90		
Experiencia en el lenguaje	1.14	1.07	1.00	0.95		
Atributos de proyecto						
Técnicas actualizadas de programación	1.24	1.10	1.00	0.91	0.82	
Utilización de herramientas de software	1.24	1.10	1.00	0.91	0.83	
Restricciones de tiempo de desarrollo	1.24	1.08	1.00	1.04	1.10	

Fuente: (Gómez, 2015)



## 2.13 SEGURIDAD DE LA INFORMACIÓN

La seguridad informática, también conocida como ciberseguridad o seguridad de tecnología de la información, es el área relacionada con la informática y la telemática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y, especialmente, la información contenida en una computadora o circulante a través de las redes de computadoras. Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información.

La ciberseguridad comprende software (bases de datos, metadatos, archivos), hardware, redes de computadoras y todo lo que la organización valore y signifique un riesgo si esta información confidencial llega a manos de otras personas, convirtiéndose, por ejemplo, en información privilegiada.

La definición de seguridad de la información no debe ser confundida con la de (seguridad informática), ya que esta última solo se encarga de la seguridad en el medio informático, pero la información puede encontrarse en diferentes medios o formas, y no solo en medios informáticos.

La seguridad informática también se refiere a la práctica de defender las computadoras y los servidores, los dispositivos móviles, los sistemas electrónicos, las redes y los datos de ataques maliciosos.

En resumen, la seguridad en un ambiente de red es la habilidad de identificar y eliminar vulnerabilidades. Una definición general de seguridad debe también poner atención a la necesidad de salvaguardar la ventaja organizacional, incluyendo información y equipos físicos, tales como los mismos computadores. Nadie a cargo de seguridad debe determinar quién y cuándo puede tomar acciones apropiadas sobre un ítem en específico. Cuando se trata de la seguridad de una compañía, lo que es apropiado varía de organización en organización. Independientemente, cualquier compañía con una red debe tener una política de seguridad que se dirija a la conveniencia y la coordinación. Según (Wikipedia, 2015)

### ❖ **Estándar ISO/IEC 27000**

El estándar ISO 27000 apunta a exigir niveles concretos y adecuados de seguridad informática, niveles necesarios para las empresas que compiten a través del comercio electrónico y que por lo tanto tienen que exponer sus infraestructuras de información.

Al aplicar en una empresa o institución un estándar como ISO 27000 el trabajo sobre la norma debe ser continuo, pues ISO año tras año publica nuevas modificaciones con el fin de asegurar una correcta gestión de la información de las organizaciones. Modificaciones orientadas a cubrir todas las posibles brechas de seguridad informática, que puedan llevar a cualquier tipo de riesgo la información de una organización. La certificación ISO 27000 será una obligación de cualquier organización que desee exigir niveles concretos y adecuados de seguridad informática. (Pandini W., 2005)

### ❖ **Estándar ISO/IEC 27002**

Es un estándar para la seguridad de la información publicado por la Organización Internacional de Normalización y la Comisión Electrotécnica Internacional. La versión más reciente es la ISO/IEC 27002:2013.

Directrices del estándar.

ISO/IEC 27002 proporciona recomendaciones de las mejores prácticas en la gestión de la seguridad de la información a todos los interesados y responsables en iniciar, implantar o mantener sistemas de gestión de la seguridad de la información. La seguridad de la información se define en el estándar como "la preservación de la confidencialidad (asegurando que sólo quienes estén autorizados pueden acceder a la información), integridad (asegurando que la información y sus métodos de proceso son exactos y completos) y disponibilidad (asegurando que los usuarios autorizados tienen acceso a la información y a sus activos asociados cuando lo requieran)". Según (Wikipedia- ISO/IEC 27002).

En este grupo se encuentra la ISO/IEC 27002, norma internacional que establece el código de mejores prácticas para apoyar la implantación del Sistema de Gestión de Seguridad de la Información (SGSI) en las organizaciones.

A través del suministro de una guía completa de implementación, esa norma describe cómo se pueden establecer los controles. Dichos controles, a su vez, deben ser elegidos en base a una evaluación de riesgos de los activos más importantes de la empresa de tecnología.

Objetivos: El principal objetivo de la ISO 27002 es establecer directrices y principios generales para iniciar, implementar, mantener y mejorar la gestión de la seguridad de la información en una organización. Esto también incluye la selección, implementación y administración de controles, teniendo en cuenta los entornos de riesgo encontrados en la empresa. Según, (Pandini W., 2005)

## **2.14 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE**

La mayoría del software puede probarse de dos maneras diferentes. Conociendo el funcionamiento interno, podemos probar que todos los módulos encajan unos con otros, es decir, desde una visión interna. Estas pruebas son pruebas de caja blanca.

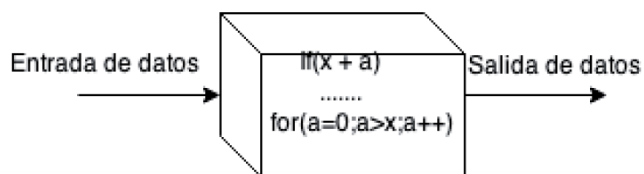
Al conocer las funciones específicas del producto se pueden llevar a cabo pruebas que demuestren que estas funciones son operativas y la búsqueda de errores en dichas funciones. Estas pruebas se realizan desde una visión externa, mediante las pruebas de caja negra.

Estas dos técnicas nos ayudan a definir los casos de prueba para tener mayor probabilidad de errores ahorrando esfuerzo y tiempo

### **2.14.1 técnica caja blanca**

La técnica de caja blanca, a veces definida como prueba de “caja de cristal” o “caja transparente”, es una técnica de diseño de casos de prueba que usa la estructura de control para obtener los casos de pruebas. (Peño, 2015)

**Figura N° 2. 25 Proceso caja blanca**



**Fuente: (Peño, 2015)**

Dentro de esta estructura podemos encontrar la estructura de un componente de software como puede ser sentencias de decisiones, caminos distintos del código, la estructura de una página Web.

Los métodos de caja blanca aportan los siguientes puntos:

- ❖ Garantizan que todas las rutas del código se revisan al menos una vez.
  - ❖ Revisan la condición lógica
  - ❖ Revisan estructuras de datos
  
  - ❖ **Pruebas de ruta básica**
- Este tipo de pruebas se basa en diseñar un caso de prueba por cada camino independiente del programa. Con esta técnica se intenta garantizar que se prueben todos los caminos de ejecución del programa al menos una vez.
- ❖ **Pruebas de ciclos de bucles**

Este tipo de pruebas se centra exclusivamente en la validación de las construcciones de bucles, en donde se pueden definir cuatro tipos de bucles simples concatenados, anidados y no estructurados. (Peño, 2015)

### **2.14.2 Técnica caja negra**

Las técnicas de diseño de caja negra, también llamadas pruebas de comportamiento, son las que utilizan el análisis de la especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del programa para diseñar los casos de prueba y, a

diferencia de las pruebas de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba.

**Figura N° 2. 26 Proceso caja negra**



**Fuente: (Peño, 2015)**

Con los métodos de caja negra se intenta encontrar los errores

- ❖ Funciones incorrectas o faltantes
- ❖ Errores de inicialización y terminación
- ❖ Errores de interfaz

Errores en las estructuras

### **3.1 INTRODUCCIÓN**

El propósito de capítulo es poner en práctica lo mencionado en los anteriores capítulos, que se analiza, desarrolla la metodología UWE, que satisfaga las necesidades que se tiene en los almacenes de la unidad de servicios generales y mantenimiento, de la base de datos del GAMEA

### **3.2 ANÁLISIS DE LA SITUACIÓN ACTUAL**

#### **❖ Solicitud de material**

La solicitud de material e intervención en el mantenimiento de un centro de salud o unidad educativa llega al responsable de la unidad, quien hace un cronograma de inspección con los supervisores y jefes de cuadrilla, quienes se encargan de la intervención y mantenimiento de las instituciones.

#### **❖ Inspección y cuantificación del material**

El supervisor junto a un personal especializado realizan la inspección y la real cuantificación de los materiales a requerir para el arreglo y/o mantenimiento de los desperfectos o faltas que tiene una institución

Cabe aclarar que los desperfectos el mantenimiento que se realiza es debido que los daños que ocasionan los estudiantes o en algunos casos es la antigüedad o por daños por el clima.

#### **❖ Dotación del material**

Una vez cuantificado el material se solicita las cantidades al encargado de almacenes y en esta parte hay falencias en la coordinación, porque en muchos casos no se tiene el material solicitado y entonces lo que se habla con los que solicitaron el mantenimiento (justa escolar, directores de unidades educativas o encargados de centros de salud) no se puede cumplir por falta de material y entra la susceptibilidad de los solicitantes.

#### **❖ Descripción del escenario**

Se toma como punto de inicio para el diseño el análisis de la situación de las actividades de los almacenes que tiene la unidad de servicios generales y mantenimiento.

La principal observación que se hace al responsable y a los encargados de almacenes, es que no tienen los datos precisos del material que aún se tiene en almacenes de manera oportuna, a su vez hay falencias en la entrega de las mismas y el tiempo de demora para entregar es considerable, porque el registro se lo hace de manera manual. Dicha observación se la hace para poder coordinar con las entidades solicitantes y poderles decir si se puede realizar la intervención o aun no, en base a los materiales con las que se cuenta, es decir la intervención de mantenimiento en la mayoría de los casos se las realiza con material dependiendo al área a intervenir.

### **3.3. CAPTURA, ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS**

En esta etapa del desarrollo del software detectamos procesos elaborados manualmente tanto el registro y los informes que se elaboran del estado y cantidades de materiales del almacén de la unidad de servicios generales y mantenimiento. Todo este proceso ocasiona trabajos monótonos en el personal encargado (responsable de almacenes), haciendo que este tome más tiempo de lo necesario en estas actividades y provocando demoras en los reportes de la cantidad de material que cuenta los almacenes.

#### **3.3.1 Obtención de requisitos**

La descripción general del escenario además de las características inherentes al proyecto planteado, constituyen los parámetros fundamentales para la identificación de requerimientos, sobre los mismos se debe hacer un análisis y estudio pertinente antes de elaborar el diseño de la aplicación.

Los requisitos globales identificados son las características con las que debe contar la aplicación para proporcionar un manejo fácil de la información para los usuarios, en este caso el jefe de unidad, el responsable y los encargados de almacén. Estos requisitos globales hacen referencia a los siguientes puntos de interés.

- ❖ Entrega de información actualizada
- ❖ Administración de la información de fácil acceso, y que la interfaz de usuario proporcione suficiente en la presentación respecto a los contenidos del sistema.
- ❖ Que la interfaz destinada a la actualización de la información, sea sencilla y de fácil manejo para los usuarios.

De manera complementaria a los requerimientos globales, se requiere que el sistema a ser desarrollado:

- ❖ Puede ser consultado desde cualquier lugar
- ❖ Proporcione seguridad en el acceso, con autenticación a la página personal.

De manera funcional el sistema debe encontrarse organizada a través de roles de administración.

- ❖ Administración principal, que se encargue de administrar los permisos a los demás usuarios.
- ❖ Encargado de almacenes, quien podrá ingresar al sistema y tendrá permisos de ingresar datos del ingreso y salida de los materiales, hacer informes y reportes, actualizar datos de cantidades y estados de los materiales.
- ❖ Jefe de unidad tendrá permisos de ingresar datos del ingreso de los materiales, introducir datos de fechas de compra y adquisición de materiales, actualizar datos de cantidades de los materiales.
- ❖ Responsable de almacenes, quien podrá ingresar al sistema y tendrá permisos de ingresar datos de salida de los materiales, hacer informes y reportes, actualizar datos de cantidades y estados de los materiales.
- ❖ Supervisores, quien podrá ver las cantidades de material disponible en almacenes

*Tabla 3. 1 Tareas de obtención de requisitos*

<i>TAREAS REALIZADAS</i>	<i>CARACTERÍSTICAS</i>
Entrevistas	Se realizan entrevistas con los siguientes personajes <ul style="list-style-type: none"> <li>❖ Jefe de unidad</li> <li>❖ Encargado de almacenes</li> </ul>
Observaciones	En los almacenes de la unidad de servicios generales y mantenimiento (GAMEA), se pudo observar algunas dificultades que retrasan las entregas, no tienen información actualizada, demoran en entregar reportes, información poco confiable entre otras, esto debido al proceso semi-automatico que se tiene en almacenes



Documentación	Se pudo recabar gran parte de la información necesaria para desarrollar el sistema.
---------------	---

**Fuente: Elaboración propia**

### 3.3.1.1 Definición de actores

Las identificaciones de los actores, nos permite conocer a los personajes involucradas en el proceso del sistema de gestión y control de almacenes.

*Tabla 3. 2 Definición de actores*

ACTORES	DESCRIPCIONES
Jefe de unidad Servicios generales y mantenimiento	Tiene las siguientes funciones dentro de la unidad <ul style="list-style-type: none"> <li>○ Le llegan las solicitudes de mantenimiento y autoriza el uso de material</li> <li>○ Coordina las intervenciones con los supervisores.</li> </ul>
Responsable de almacenes	Tiene las siguientes funciones dentro la unidad <ul style="list-style-type: none"> <li>○ Recibe materiales</li> <li>○ Hace la solicitud de material</li> <li>○ Cuantifica y distribuye los materiales a los almacenes.</li> <li>○ Realiza reportes cuando se lo requieran</li> <li>○ Realiza informes mensuales del estado de los almacenes</li> <li>○ Hace modificaciones si se lo requiere</li> </ul>
Encargado de almacenes	Tiene las siguientes funciones dentro la unidad <ul style="list-style-type: none"> <li>○ Realiza reportes cuando se lo requieran</li> <li>○ Hace modificaciones si se lo requiere</li> <li>○ Ingresa datos de materiales en el sistema</li> <li>○ Distribuye material del almacén</li> <li>○ Coordina con los supervisores</li> </ul>
Supervisores (servicios generales y mantenimiento)	Tiene las siguientes funciones dentro la unidad <ul style="list-style-type: none"> <li>○ Ingresa al sistema para consultar</li> <li>○ Consulta cantidad de materiales de almacenes</li> <li>○ Coordina la cantidad de material que se utilizara para la intervención</li> </ul>

**Fuente: Elaboración propia**

### Administrador (Responsable de Almacenes)

Gestión de usuarios el administrador basado en la lista de encargados de almacén y los demás usuarios del sistema, esto para que cada usuario tenga diferentes privilegios.

- ❖ Usuario
- ❖ Password
- ❖ Nivel

### **Jefe de unidad**

Gestiona la administración de la unidad de servicios generales y mantenimiento así mismo atiende diferentes requerimientos y solicitudes de mantenimiento que necesitan las unidades educativas y los centros de salud, y también autoriza la adquisición de materiales de acuerdo a la cantidad requerida

### **Encargados de almacén**

Tienen la función principal de distribuir material y entregar reportes de las cantidades que se tiene en los almacenes de acuerdo a cada área de intervención. La función en el sistema es consultar, registrar ingresos y salidas, y realizar informes cuando se lo requiera.

### **Supervisores**

La función de los supervisores, son coordinar con las entidades solicitantes en cuanto al tiempo, el personal y el material que se requiere para la atención a la solicitud. El rol en el sistema es el de consulta y de acuerdo al material disponible es autorizado por el jefe de unidad.

### **3.3.1.2 Lista de requerimiento del sistema**

La obtención correcta del requerimiento que se puede llegar a describir con claridad, en forma consistente, el comportamiento del sistema. Las funciones que deben realizarse se clasifican en tres categorías como se detallan en la siguiente tabla.

*Tabla 3. 3 Lista de requerimiento del sistema*

<b>FUNCIONES</b>	<b>SIGNIFICADO</b>
Evidente	Alerta de confirmación, el usuario debe saber que acción se ha realizado.
Oculto	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica en varios módulos donde la información es guardada en un mecanismo persistente de almacenamiento oculto.
Superflua	Esta función es opcional en el cual no repercuten la forma significativa en el costo ni en otras funciones.

**Fuente** (Elaboración propia).

### 3.3.1.2.1 Requerimientos funcionales

Los requisitos funcionales en la siguiente tabla muestran las características que necesita el sistema a partir de la información obtenida como parte de la tarea de obtención de requisitos, se detallan a continuación en la siguiente tabla

*Tabla 3. 4 Requerimientos funcionales*

Ref.	Función	Categoría
R.1	Control de acceso seguro y diferenciado por tipo de usuarios o rol.	Evidente
R.2	Gestión de Usuario del Sistema.	Evidente
R.3	Desplegar vistas y menús de acuerdo al cargo de cada usuario.	Oculto
R.4	Determinar el estado de cada usuario activado.	Evidente
R.5	Registros de personal de almacén	Evidente
R.6	Registrar materiales de almacenes	Evidente
R.7	Registrar proveedores	Evidente
R.8	Listar por cargo y unidad a la que pertenece	Evidente
R.9	Listar materiales existentes.	Evidente
R.10	Registrar unidades educativas.	Evidente
R.11	Registrar productos salientes	Evidente
R.12	Generar reportes	Evidente
R.10	Búsqueda de materiales.	Evidente

**Fuente** (elaboración propia).

### 3.3.1.2.2 Requerimientos no Funcionales

El requerimiento no funcional muestra su comportamiento en la siguiente tabla.

*Tabla 3. 5 Requerimientos no Funcionales*

REF.	FUNCIÓN	CATEGORÍA
R.1	El sistema debe visualizarse y funcionar correctamente en cualquier Navegador, Internet Explorer, Mozilla Firefox, Chrome.	Evidente
R.2	El sistema no debe tardar más de diez segundos en mostrar los resultados de una búsqueda.	Evidente
R.3	El sistema debe tener seguridad en el acceso a la información del sistema bajo los roles del usuario.	Evidente

**Fuente:** (Elaboración Propia).

### 3.3.2 Especificación de requerimientos para la aplicación.

A continuación se presenta un análisis de los requerimientos identificados en el lugar de trabajo, con entrevistas y la observación directa de los problemas. Haciendo un análisis de las especificaciones de los requerimientos del software con el uso del estándar IEEE-830 se ha podido identificar los requisitos mismos.

Al proponer una aplicación web se debe tener en cuenta la utilización de herramientas propias de la www, que permita por defecto el servicio de aplicaciones distribuidas basadas en el modelo cliente/servidor, además del requerimiento de los siguientes componentes de software, tanto para el diseño, desarrollo e implementación de la aplicación.

*Tabla 3. 6 Requerimientos para la aplicación*

Tipo	Nombre	Fase
Lenguaje de programación	PHP.	Etapa de desarrollo de la aplicación
Gestor de base de datos	MySQL	Etapa de implementación de la aplicación
Framework	Bootstrap Codegniter	Etapa de funcionamiento de la aplicación
Servidor de aplicaciones web	Apache	Funcionamiento de la aplicación
Entorno de desarrollo	PHP	Etapa de implementación en la etapa de desarrollo
Diseño grafico	HTML5 Css3	Etapa de la implementación en el desarrollo de la aplicación

**Fuente:** (Elaboración propia)

## **3.4 ANÁLISIS DE REQUERIMIENTOS Y DISEÑO DEL SISTEMA**

### **3.4.1 Modelo de caso de uso**

Dentro de este ciclo se realiza un análisis del sistema construyendo para ello diagramas de casos de uso en una descripción de los pasos o las actividades que realiza acabo los proceso del sistema.

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

A continuación, se hace el modelamiento donde se puede apreciar cómo interactúan los actores sobre los casos de uso.

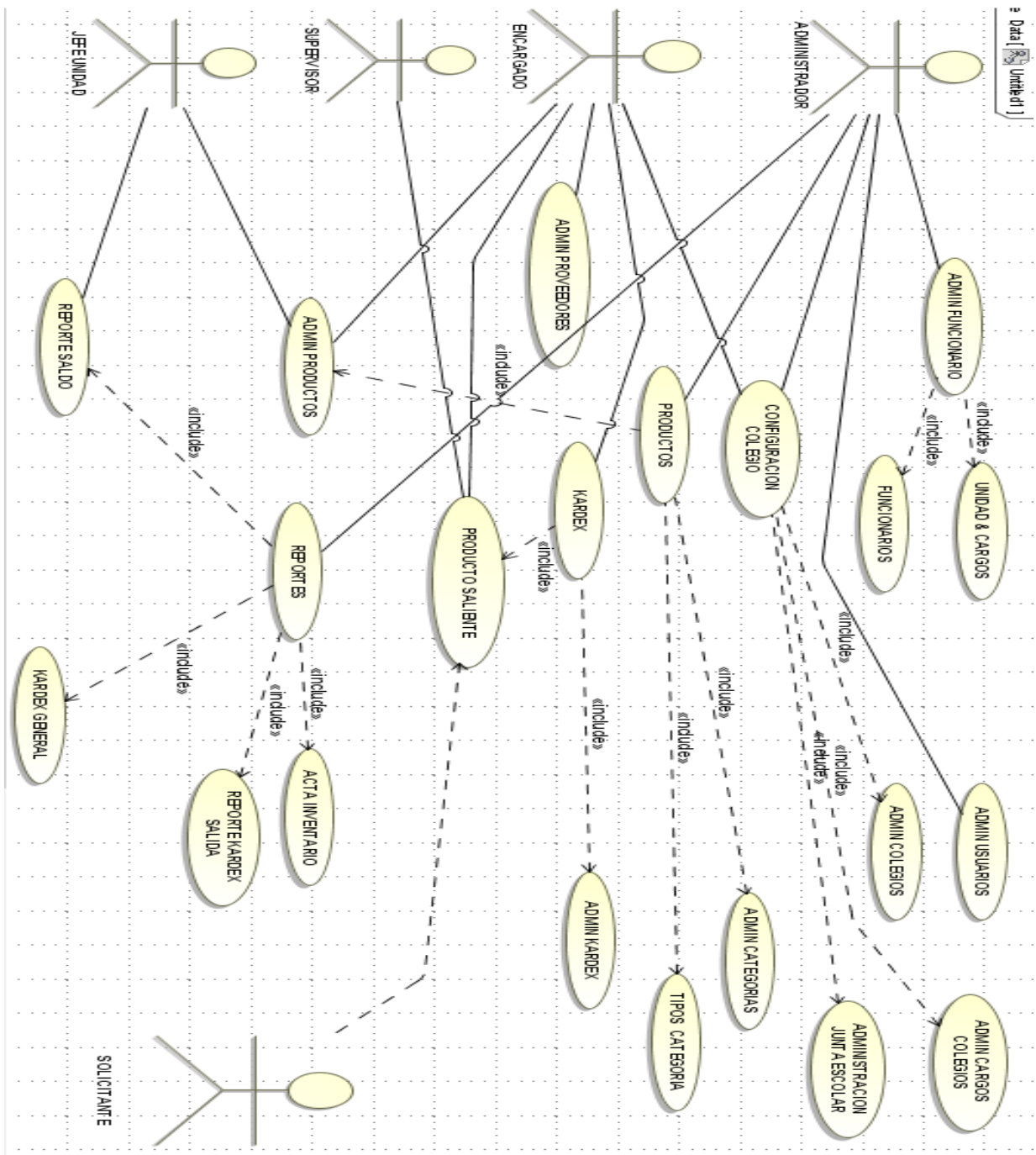
Realizaremos una presentación previa de los actores en diagramas de casos de uso a continuación.

- ❖ Modelo de diagrama de caso de uso: General del Sistema
- ❖ Diagrama de caso de uso: Administrador/a principal.
- ❖ Diagrama de caso de uso: Encargado/a.
- ❖ Diagrama de caso de uso: Jefe de unidad.
- ❖ Diagrama de caso de uso: Supervisor.

#### **a) Modelo de diagrama de caso de uso: General del Sistema**

En este modelo presentaremos el modelamiento de diagrama de caso de uso general como interactúan los distintos actores con distintos módulos del sistema.

FIGURA N° 3. 1 Modelo de diagrama de caso de uso: General del Sistema

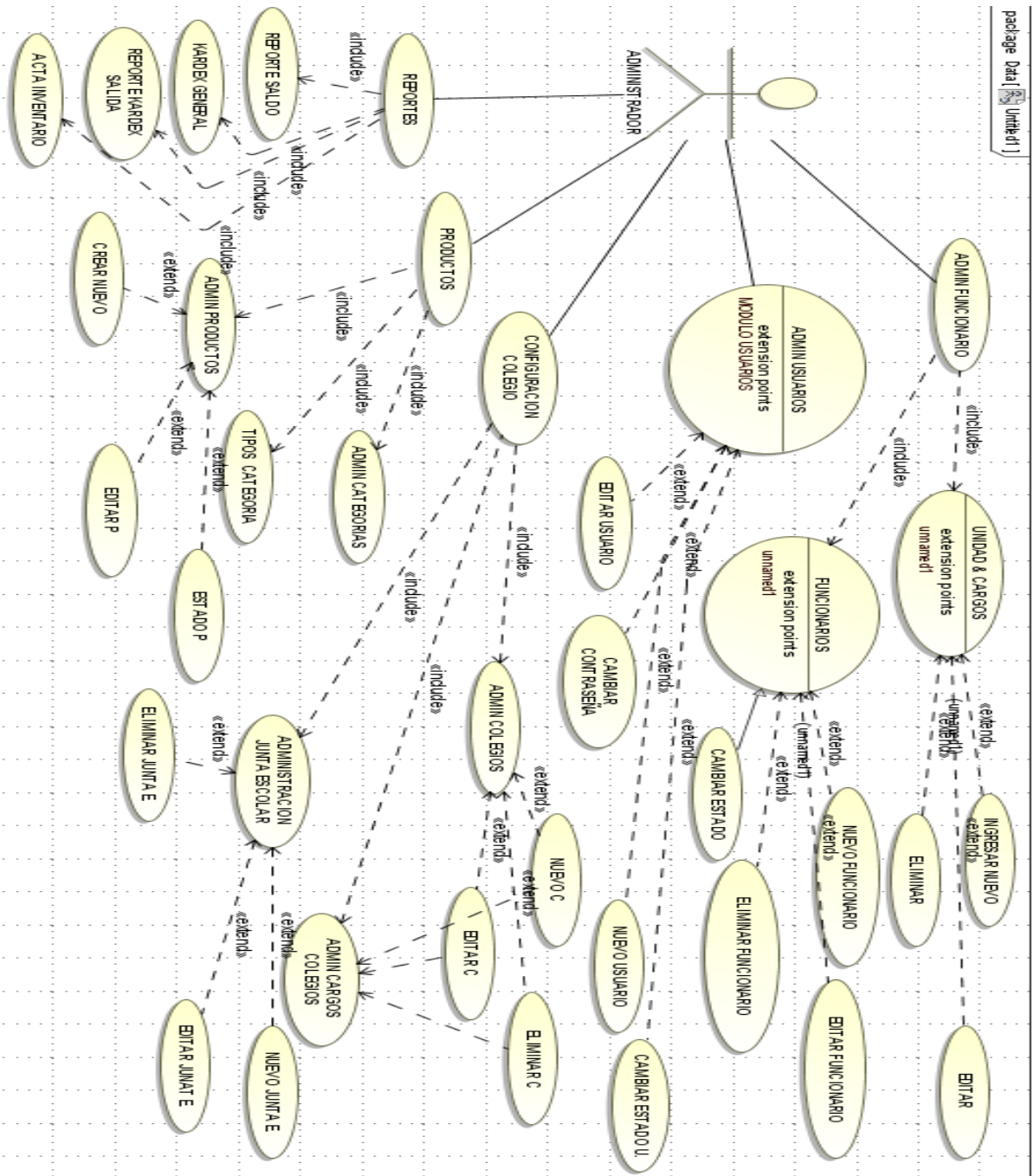


Fuente (Elaboración propia).

**b) Diagrama de caso de uso: Administrador/a principal.**

En este módulo presentaremos los movimientos y privilegios que tiene el administrador principal, y todas las relaciones que tiene con los otros módulos.

FIGURA N° 3. 2 Diagrama de caso de uso: Administrador/a principal.



Fuente (Elaboración propia).

Tabla 3. 7 Caso de uso: Administrador principal

<b>Caso de uso: Administrador principal</b>	
<b>Actores</b>	Administrador principal.
<b>Tipo</b>	Primario esencial.
<b>Descripción</b>	<p>El administrador principal tiene las siguientes opciones de modulo como ser:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administración unidad lo cual lo permite mostrar todo los unidad de la misma manera podrá registrar un nuevo unidad, editar unidad y eliminar unidad.</li> <li>❖ Módulo de administración cargos de la misma manera el administrador podrá visualizar todo los cargos con su respectiva unidad, también podrá realizar nuevo registro de cargo, editar cargo, eliminar y por ultimo cambiar estado.</li> <li>❖ Módulo de administración de funcionarios, en este módulo el administrador podrá registrar nuevo funcionario, editar funcionario, eliminar funcionario y por ultimo cambiar de estado.</li> <li>❖ Módulo de administración de usuario, en este módulo el usuario podrá registra un nuevo usuario, modificar usuario, cambiar de estado y cambiar o resetear su usuario.</li> <li>❖ Módulo de administración nombre unidad educativas, en este módulo nos permite registrar todo los colegios, editar colegios, cambiar estado, eliminar. De la misma manera existe un sub modulo donde nos permite registrar todo los cargos de junta escolar.</li> <li>❖ Módulo de administración de junta escolar, en este módulo nos permite mostrar las junta escolares con su respectivo colegio y cargos. Por ultimo cuenta con la administración de nuevo registro, editar, cambiar estado y eliminar.</li> <li>❖ Módulo de administración de productos, en este módulo el usuario de la misma manera podrá realizar un nuevo registro de producto con su respectivo proveedor, editar, cambiar de estado.</li> <li>❖ Módulo de administración de reportes, en este módulo el usuario cuenta con el permiso de poder realizar un acta de inventario,</li> </ul>



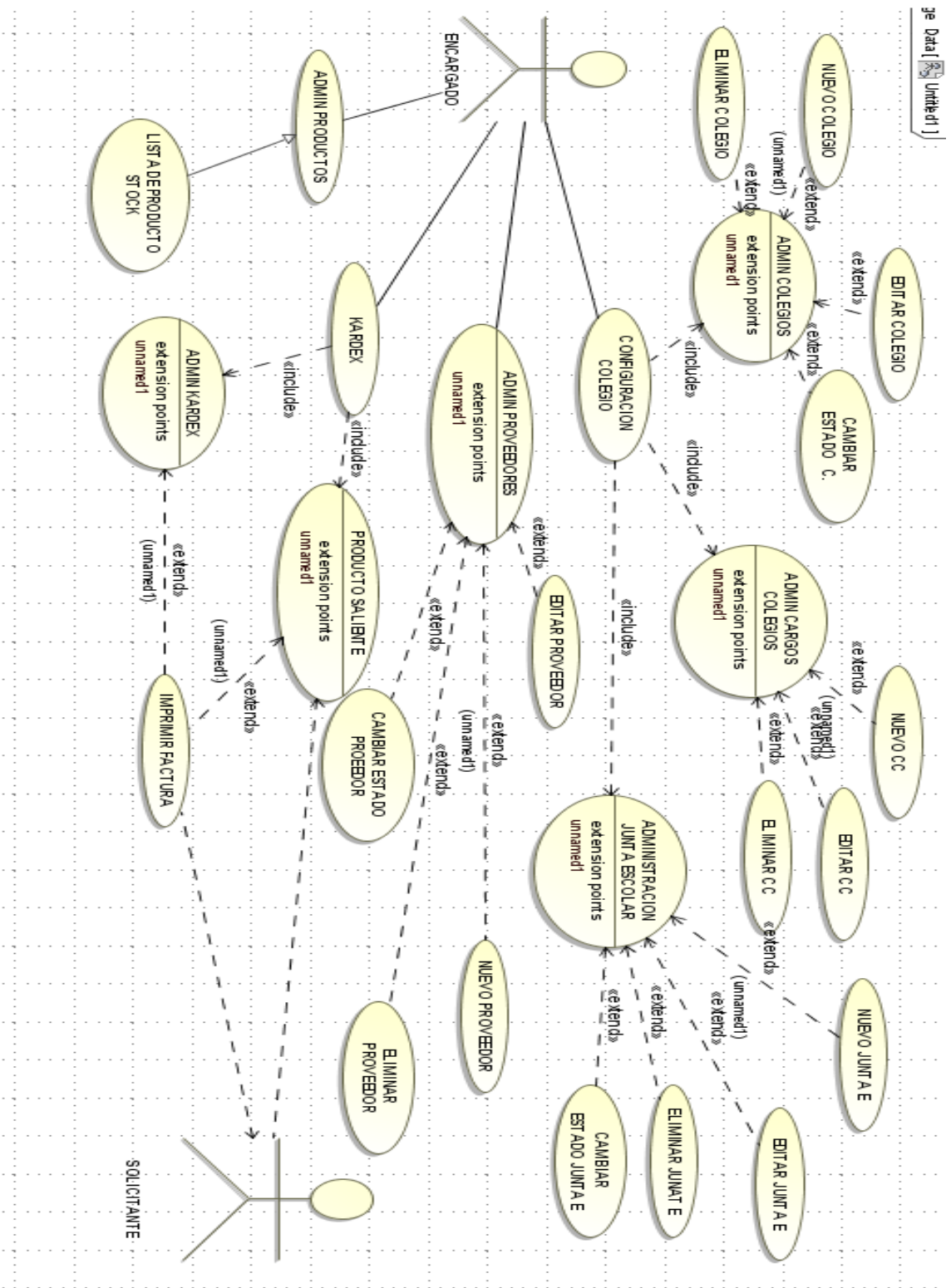
	<p>reporte de kardex de productos salientes, kardex general y reportes de saldos de productos.</p> <ul style="list-style-type: none"><li>❖ Módulo de administración de inventario en este módulo el usuario cuenta el privilegio de poder realizar un inventario de todos los productos.</li></ul>
--	--

Fuente (Elaboración propia).

### **c) Diagrama de caso de uso: Encargado de almacén.**

En este módulo presentaremos los movimientos, privilegios y restricciones de acceso que tiene el encargado de almacén y todas las relaciones que tiene con los otros módulos.

FIGURA N° 3. 3 Diagrama de caso de uso: Encargado.



Fuente: (Elaboración propia).

Tabla 3. 8 Caso de uso: Encargado de almacén

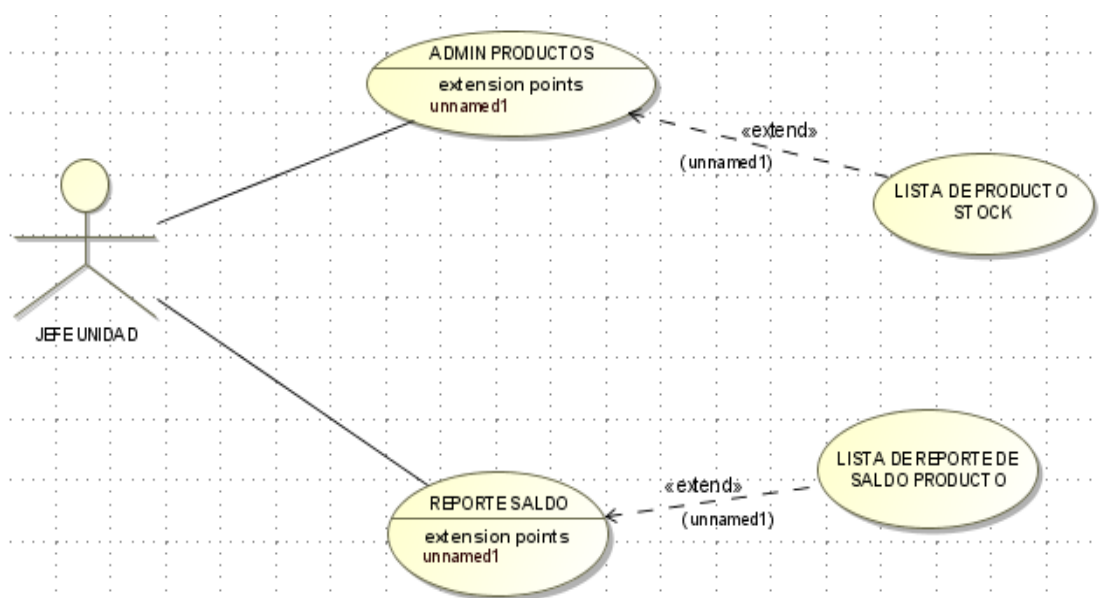
Caso de uso: Encargado.	
<b>Actores</b>	Encargado.
<b>Tipo</b>	Primario esencial.
<b>Descripción</b>	<p>El encargado de almacenes tiene las siguientes opciones de modulo como ser:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administración nombre unidad educativas, en este módulo nos permite registrar todo los colegios, editar colegios, cambiar estado, eliminar. De la misma manera existe un sub modulo donde nos permite registrar todo los cargos de junta escolar.</li> <li>❖ Módulo de administración de junta escolar, en este módulo nos permite mostrar las junta escolares con su respectivo colegio y cargos. Por ultimo cuenta con la administración de nuevo registro, editar, cambiar estado y eliminar.</li> <li>❖ Módulo de administración de productos, lista de productos stock.</li> <li>❖ Módulo de administración de proveedores, este módulo nos permite listar todo los proveedores, crear un nuevo proveedor, editar, cambiar estado y eliminar proveedor.</li> <li>❖ Módulo de administración de kardex asignar producto al solicitante e imprimir factura.</li> </ul>

Fuente (Elaboración propia).

#### **d) Diagrama de caso de uso: Jefe de unidad.**

En este módulo presentaremos los movimientos, privilegios y restricciones de acceso que tiene el jefe de unidad que también tiene acceso al sistema y todas las relaciones que tiene con los otros módulos.

**Figura N° 3. 4 Diagrama de caso de uso: Jefe de unidad.**



**Fuente** (Elaboración propia).

*Tabla 3. 9 Caso de uso: Jefe de unidad*

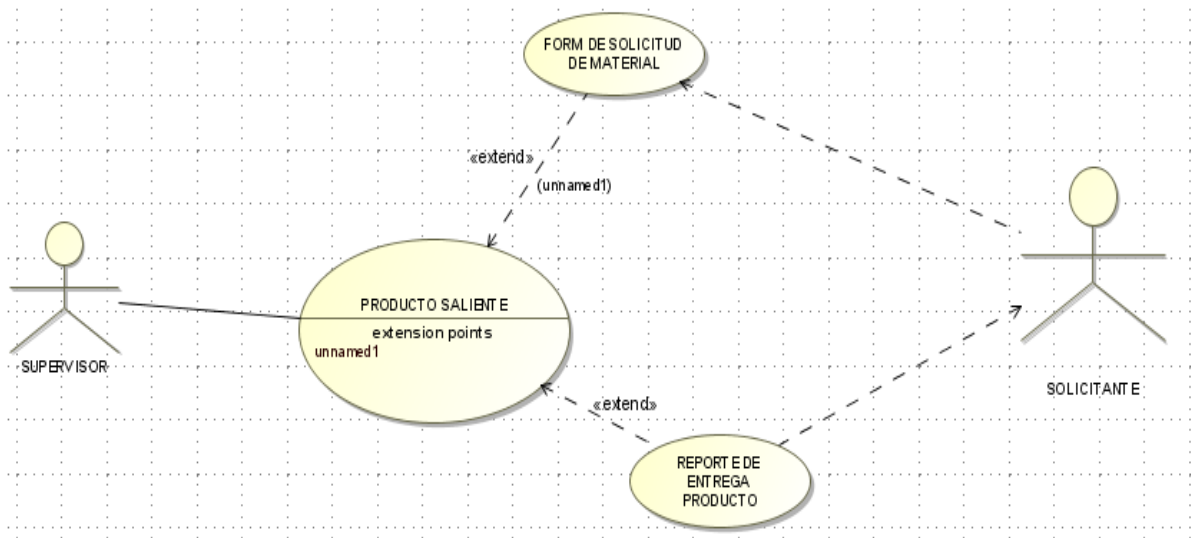
Caso de uso: Jefe de unidad	
<b>Actores</b>	Jefe de unidad
<b>Tipo</b>	Primario esencial.
<b>Descripción</b>	<p>El administrador principal tiene las siguientes opciones de modulo como ser:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administración de productos, lista de productos stock.</li> <li>❖ Lista de reporte de saldo de productos en un formato pdf de una fecha inicial y una fecha final.</li> <li>❖ Módulo de administración de proveedores, este módulo nos permite listar todo los proveedores, crear un nuevo proveedor, editar, cambiar estado y eliminar proveedor.</li> <li>❖ Módulo de administración de kardex asignar producto al solicitante e imprimir factura.</li> </ul>

**Fuente:** (Elaboración propia).

**e) Diagrama de caso de uso: Supervisor.**

En este módulo presentaremos los movimientos, privilegios y restricciones de acceso que tiene cada supervisor que también tiene acceso al sistema y todas las relaciones que tiene con los otros módulos.

**Figura N° 3. 5 Diagrama de caso de uso: Supervisor.**



**Fuente (Elaboración propia).**

*Tabla 3. 10 Caso de uso: Supervisor*

Caso de uso: Supervisor	
<b>Actores</b>	Supervisor
<b>Tipo</b>	Primario esencial.
<b>Descripción</b>	<p>En este módulo el supervisor solo puede realizar:</p> <ul style="list-style-type: none"> <li>❖ Módulo de kardex entrega de material al solicitante y poder imprimir el reporte de entrega de material en detalle.</li> </ul>

**Fuente (Elaboración propia).**

### **3.4.2 Modelo de contenido.**

El modelo de contenido para aplicaciones Web en UWE, utiliza modelos estándares UML para modelar estructuras de clases, asociaciones y paquetes. El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojados en el sistema el modelo de contenido contiene la información relevante almacenada en el sistema como se estructura como se relaciona.

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación Web.



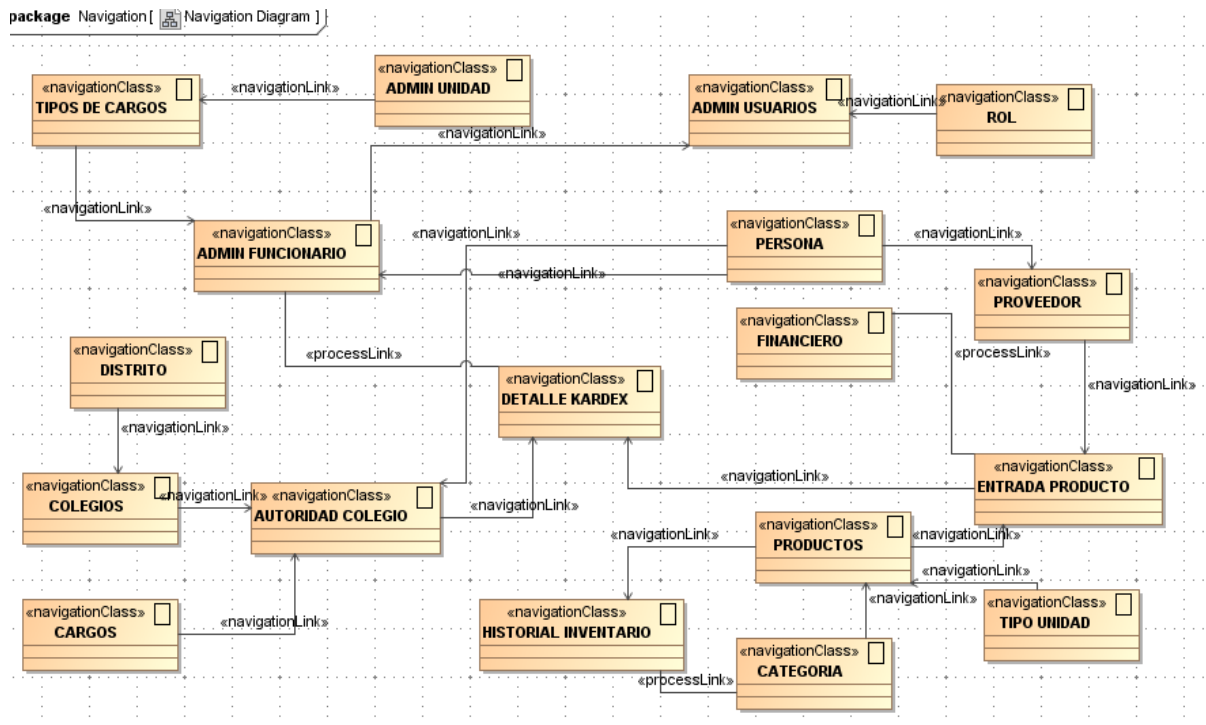
El modelo muestra la relación entre las entidades del sistema, pudiendo destacar la importancia de la entidad de usuario en el sistema

### 3.4.3 Modelo De Navegación.

A continuación, se realiza el modelamiento de modelo de navegación, donde se puede apreciar cómo interactúan los usuarios en el modelo de navegación en cada clase.

Estos tienen por objetivo de ilustrar los vínculos lógicos y de navegación entre clases. El modelo de clase navegación y las asociaciones de navegación general del proyecto, que expresa la navegación directa que se muestra en las siguientes modelos de navegación.

FIGURA N° 3. 7 Modelo de navegación del sistema



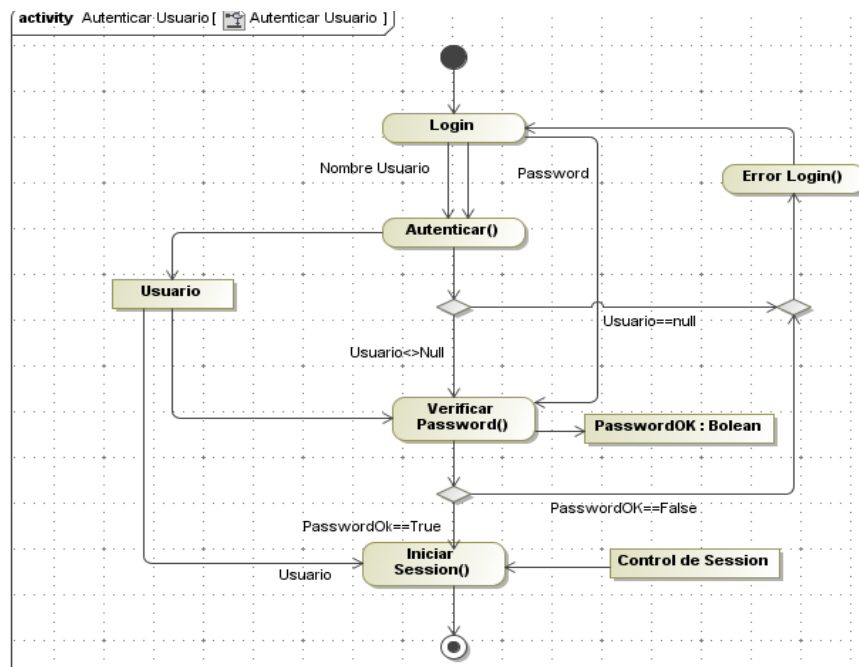
Fuente: (Elaboración propia).

### 3.4.4 Modelo de Estructura de proceso.

Este modelo de estructura de navegación permite ilustrar los vínculos lógicos y de navegación entre clases, menús, índices y clases de proceso para tener una mejor comprensión.

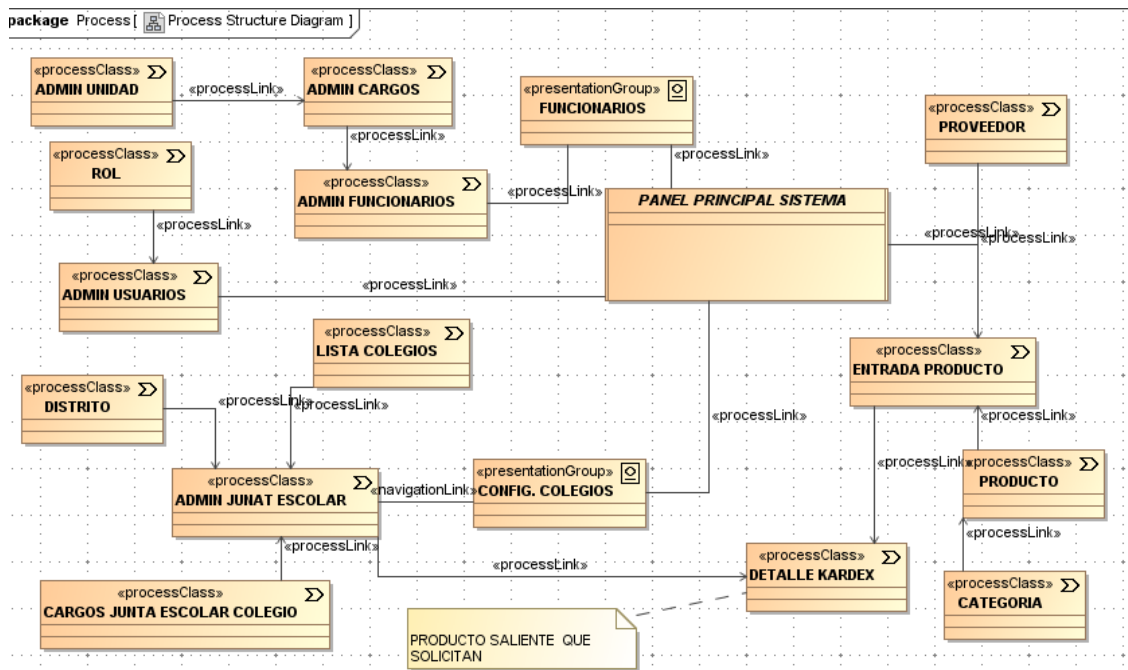


**Figura N° 3. 8 Modelo de Estructura de proceso**



Fuente: (Elaboración propia).

**Figura N° 3. 9 Modelo de Estructura de proceso**



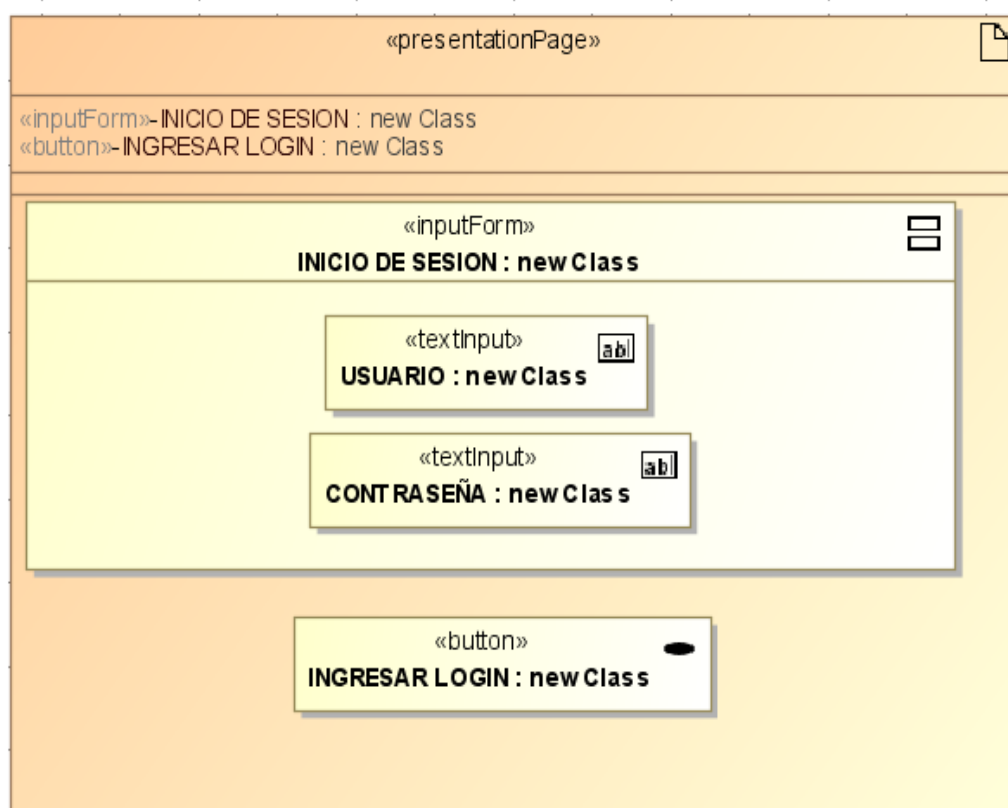
Fuente: (Elaboración propia).

### 3.4.5 Modelo de Presentación

A continuación se muestran los modelos de presentación del sistema, según UWE propone para la construcción de las páginas en forma de bosquejos derivadas desde la Figuras donde se muestra como los usuarios podrán acceder al sistema mostrando los menús correspondientes.

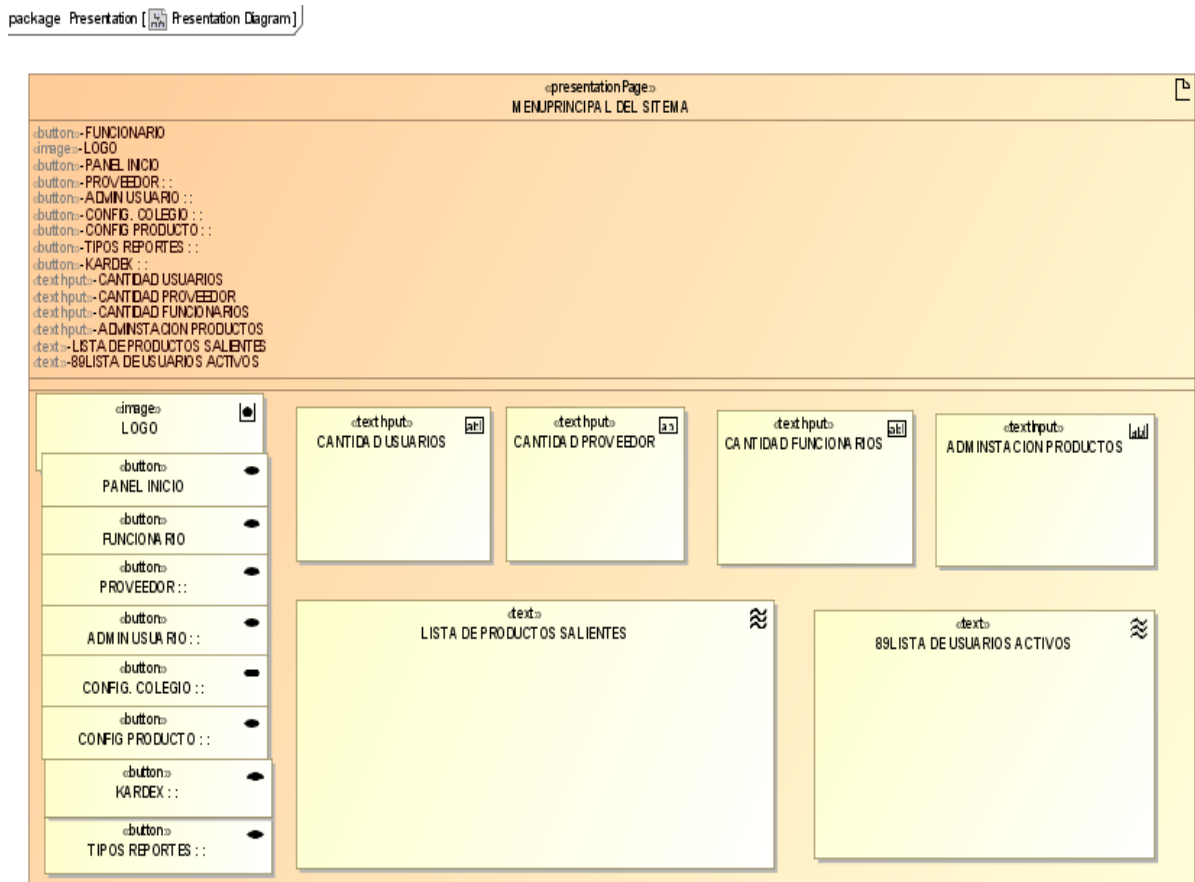
### 3 Modelo de Presentación: Administrador/a principal.

**Figura Nº 3. 10 Modelo de Presentación: Administrador/a principal login**



**Fuente:** (Elaboración propia).

**Figura N° 3. 11 Módulo de presentación: administrador principal**

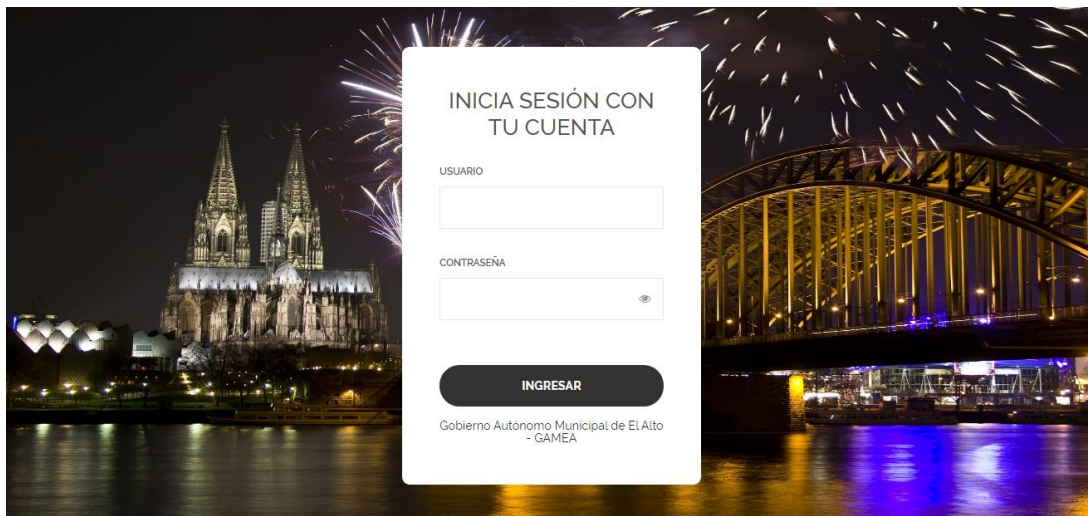


Fuente: (Elaboración propia).

### 3.5 IMPLEMENTACIÓN DEL SISTEMA

En esta fase de implementación consiste en mostrar la el desarrollo de la presentación de las interfaces del sistema y sus elementos construidos a partir del diseño del Modelo de Presentación UWE. Como se muestra en las siguientes Figuras.

**Figura N° 3. 12 Interfaz de Usuarios**



Fuente: (Elaboración propia).

**Figura N° 3. 13 Administración de Usuarios**

ADMINISTRACION DE USUARIOS

Mostrar 10 registros

Buscar: ...

#	CARNET	NOMBRE & APELLIDO	TIPO DE USUARIO	ESTADO	Action
1	4844706 LP	JUAN MALDONADO	ADMINISTRADOR/A	ACTIVO	ACCION
2	11112221 LP	LUIS HERNAN MONTENEGRO	ENCARGADO ALMACEN	ACTIVO	ACCION
3	11112222 PD	HERNAN MERCADO	JEFE DE UNIDAD	ACTIVO	ACCION
4	11112223 LP	MONIKA COPITICONA	SUPERVISOR	ACTIVO	ACCION

Fuente: (Elaboración Propia)

**Figura N° 3. 14 Administración de funcionarios**

BIENVENIDO

MENU > ADMIN FUNCIONARIOS

ADMINISTRACION DE FUNCIONARIOS

+ NUEVO FUNCIONARIO

LISTA DE FUNCIONARIO

Mostrar 10 registros

Buscar: ...

CARNET	NOMBRE & APELLIDO	CELULAR	CORREO	UNIDAD	TIPO CARGO	ESTADO	FECHA
11112233 LP	MARITHA MORALES MERCADO	734534534	mar@gmail.com	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES	ACTIVO	2019-06-03
11112224 PD	MARITHA TICONA TARQUI	23423423	marth@gmail.com	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN	ACTIVO	2019-06-03
11112223 LP	MONIKA COPITICONA MENDOZA	12312323	mkr@gmail.com	UNIDAD DE REGISTRO ADMISIONES	JEFE DE UNIDAD REG ADMI	ACTIVO	2019-04-26
11112222 PD	HERNAN MERCADO	12312323	her@gmail.com	UNIDAD RECURSO	JEFE DE UNIDAD	ACTIVO	2019-04-26

Fuente: (Elaboración Propia)

**Figura N° 3. 15 Administración proveedor**

GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO

MENU > ADMIN PROVEEDORES

ADMINISTRACION DE PROVEEDORES

+ NUEVO PROVEEDORES

Mostrar 10 registros

Buscar: ...

CARNET	NOMBRE & APELLIDO	CELULAR	CORREO	NIT	ESTADO	IMAGEN
11112225 PT	MARIO MERCADO MAMANI	562423423	mkr@gmail.com	5424	ACTIVO	
23123213 LP	LUIS FERNANDO MONTENEGRO MONTECINO	12312323	fer@gmail.com	5423	ACTIVO	
11112224 PD	MARITHA TICONA TARQUI	23423423	marth@gmail.com	11231233	ACTIVO	

Mostrar registros del 1 al 3 de un total de 3 registros

« < 1 > »

Fuente: (Elaboración Propia)

**Figura N° 3. 16 Administración de productos**

The screenshot displays the 'Productos' management interface. At the top, there are buttons for 'AGREGAR PRODUCTOS +' and 'IMPRIMIR RESPALDO DE ENTRADA DE PRODUCTOS'. Below these, a search bar and a 'Mostrar 10 registros' dropdown are visible. The main table lists the following products:

#	Producto Ingresado	Proveedor de Productos	Cantidades Actuales	Estado	Acciones
1	BOLSA DE CEMENTO VIACHA	Marithza Ticona	82	ACTIVO	[Edit] [Delete]
2	FOCO DE 220W	Marithza Ticona	85	ACTIVO	[Edit] [Delete]
3	BOLSA DE CEMENTO CAMBA	Marithza Ticona	87	ACTIVO	[Edit] [Delete]
4	FOCO DE 160W	Marithza Ticona	140	ACTIVO	[Edit] [Delete]
5	BOLSA DE ESTUCO	Marithza Ticona	177	ACTIVO	[Edit] [Delete]
6	FOCO DE 120W	Marithza Ticona	195	ACTIVO	[Edit] [Delete]
7	Ladrillo 6 Huecos	Marithza Ticona	10000	ACTIVO	[Edit] [Delete]

Fuente: (Elaboración Propia)

**Figura N° 3. 17 Administración de colegios**

The screenshot displays the 'ADMINISTRACION DE JUNTA ESCOLAR' interface. It features a '+ NUEVO REGISTRO' button, a search bar, and a 'Mostrar 10 registros' dropdown. The table contains the following records:

ACCION	CARNET	NOMBRE & APELLIDO	CARGO	DISTRITO	COLEGIO	ESTADO
[ACCION]	11112226 BN	LUIS CARLOS MAMANI MERCADO	PRESIDENTE DE JUNTA ESCOLAR	DISTRITO 1	COLEGIO MARCELO QUENTA ORTIZ	INACTIVO
[ACCION]	11112226 BN	LUIS CARLOS MAMANI MERCADO	PRESIDENTE DE JUNTA ESCOLAR	DISTRITO 1	COLEGIO MARCELO QUENTA ORTIZ	ACTIVO

Mostrar registros del 1 al 2 de un total de 2 registros

Fuente: (Elaboración Propia)

**Figura N° 3. 18 Reporte de acta inventario**

**SICAL** JUAN MALDONADO ADMINISTRADORA

REPORTES

REPORTES SEMANA REPORTES MENSUAL REPORTES ANUAL

REPORTES SEMANAL

FECHA INICIO: 09/07/2019 FECHA FINAL: 12/11/2019 ACCION: BUSCAR

IMPRIMIR

1) MATERIAL DE CONSTRUCCION

#	DESCRIPCION	N° FACTURA	CANTIDAD	VALOR UNITARIO	IMPORTE TOTAL
1	BOLSA DE CEMENTO PERU	2222	47	45	2115
2	ladrillo18 huecos	3333	100	0.5	50
TOTAL :					2165 Bs.

2) MATERIAL DE ELECTRECIDAD

Fuente: (Elaboración Propia)

**Figura N° 3. 19 Reporte del inventario de los almacenes**

**GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO**

G.A.M.E.A. INVENTARIO DE ALMACENES (EXPRESADO EN BOLIVIANOS)

FECHA DE INVENTARIO: 9 de julio de 2019 AL 12 de noviembre de 2019

1 MATERIAL DE CONSTRUCCION

#	DESCRIPCION	N° FACTURA	CANTIDAD	VALOR UNITARIO	IMPORTE TOTAL
1	BOLSA DE CEMENTO PERU	2222	47	45	2115
2	ladrillo18 huecos	3333	100	0.5	50
TOTAL :::					2165 Bs.

2 MATERIAL DE ELECTRECIDAD

#	DESCRIPCION	N° FACTURA	CANTIDAD	VALOR UNITARIO	IMPORTE TOTAL
1	FOCO DE 220W	2222	100	20	2000
TOTAL :::					2000 Bs.

SALDO TOTAL A LA FECHA

MATERIAL DE CONSTRUCCION	2165
MATERIAL DE ELECTRECIDAD	2000
TOTAL :::	4165 Bs.

Fuente: (Elaboración Propia)

**Figura N° 3. 20 Reporte de kardex salida de material**

#	CARNET	NOMBRE & APELLIDO	UNIDAD	CARGO	CANTIDAD	FECHA
1	11112233 LP	MARITZA MORALES MERCADO	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES	5	2019-11-12 09:15:51
2	4844709 LP	JUAN MALDONADO MENDOZA	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES	1	2019-06-03 05:24:29
3	11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN	3	2019-06-03 05:15:19
4	11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN	4	2019-06-02 10:13:56
5	11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN	3	2019-06-02 19:59:10
6	11112223 LP	MONIKA COPITICONA MENDOZA	UNIDAD DE REGISTRO ADMISIONES	JEFE DE UNIDAD REG ADMI	2	2019-06-02 19:55:07
7	11112223 LP	MONIKA COPITICONA MENDOZA	UNIDAD DE REGISTRO ADMISIONES	JEFE DE UNIDAD REG ADMI	2	2019-06-02 15:22:36
8	11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN	4	2019-06-02 15:21:54
9	4844709 LP	JUAN MALDONADO MENDOZA	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES	1	2019-06-02 13:35:02

Fuente: (Elaboración Propia)

**Figura N° 3. 21 Reporte salida de material**

**GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO**  
G.A.M.E.A. INVENTARIO DE ALMACENES  
SALIDA DE MATERIA KARDEX EN DETALLE

CARNET	NOMBRE & AP.	UNIDAD	CARGO			
11112233 LP	MARITZA MORALES MERCADO	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES			
#	TIPO DE PRODUCTO	NOMBRE DE PRODUCTO	FECHA	CANTIDAD	COSTO U	COSTO
1	MATERIAL DE CONSTRUCCION	BOLSA DE CEMENTO PARI	2019-11-12 09:00:01	47	45	2115 Bs.
2	MATERIAL DE ELECTRICIDAD	FOCO DE 120W	2019-11-12 09:00:01	100	20	2000 Bs.
3	MATERIAL DE CONSTRUCCION	Salvo Bolt 8mm	2019-11-12 09:00:01	100	0.3	30 Bs.
<b>TOTAL : 4165 Bs.</b>						

CARNET	NOMBRE & AP.	UNIDAD	CARGO			
4844709 LP	JUAN MALDONADO MENDOZA	UNIDAD BIENES Y SERVICIOS	RESPONSABLE DE ALMACENES			
#	TIPO DE PRODUCTO	NOMBRE DE PRODUCTO	FECHA	CANTIDAD	COSTO U	COSTO
1	MATERIAL DE CONSTRUCCION	BOLSA DE CEMENTO CASBA	2019-06-03 05:24:29	10	46	460 Bs.
<b>TOTAL : 460 Bs.</b>						

CARNET	NOMBRE & AP.	UNIDAD	CARGO			
11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN			
#	TIPO DE PRODUCTO	NOMBRE DE PRODUCTO	FECHA	CANTIDAD	COSTO U	COSTO
1	MATERIAL DE CONSTRUCCION	BOLSA DE CEMENTO CASBA	2019-06-03 05:15:19	30	60	2000 Bs.
2	MATERIAL DE CONSTRUCCION	BOLSA DE CEMENTO PARI	2019-06-03 05:15:19	10	40	400 Bs.
3	MATERIAL DE ELECTRICIDAD	FOCO DE 120W	2019-06-03 05:15:19	10	10	100 Bs.
<b>TOTAL : 3250 Bs.</b>						

CARNET	NOMBRE & AP.	UNIDAD	CARGO			
11112221 LP	LUIS HERNAN MONTENEGRO LANDAVETA	UNIDAD RECURSO HUMANO	ENCARGADO ALMACEN			
#	TIPO DE PRODUCTO	NOMBRE DE PRODUCTO	FECHA	CANTIDAD	COSTO U	COSTO
1	MATERIAL DE CONSTRUCCION	BOLSA DE CEMENTO VASCA	2019-06-02 19:59:10	1	10	100 Bs.
<b>TOTAL : 100 Bs.</b>						

Fuente: (Elaboración Propia)



**Figura N° 3. 22 Kardex general**

The screenshot shows the SICAL system interface. On the left is a navigation menu with options like 'INICIO', 'ADMIN FUNCIONARIO', 'ADMIN USUARIO', 'CONFIG. COLEGIOS', 'PRODUCTO', 'REPORTES', 'ACTA INVENTARIO', 'REPORTE KARDEX SALIDA', 'KARDEX GENERAL', 'REPORTES DE SALDO', 'INVENTARIO PRODUCTO', and 'Database Backup'. The main area displays a report for 'BOLSA DE CEMENTO VIACHA' with filters for 'CATEGORIA DE PRODUCTO' (MATERIAL DE CONSTRUCCION), 'TIPO DE PRODUCTO' (BOLSA DE CEMENTO VIACHA), 'FECHA INICIO' (01/05/2019), and 'FECHA FINAL' (12/11/2019). A table shows the inventory movements, and an 'IMPRIMIR' button is visible.

#	FECHA	N° FACTURA	DETALLE	U. DE MEDIDA	FISICO			PRECIO UNITARIO	VALORADO		
					ENTRADA	SALIDA	SALDO		ENTRADA	SALIDA	SALDO
1	2019-06-01	2222	BOLSA DE CEMENTO VIACHA	Unidades	200	xxxxxxxxxx	200	50 Bs.	10000 Bs.	xxxxxxxxxx	10000 Bs.
1	2019-06-02	2222	ENCARGADO ALMACEN	Unidades	xxxxxxxxxx	100	100	50 Bs.	xxxxxxxxxx	5000 Bs.	5000 Bs.
2	2019-06-03	2222	JEFE DE UNIDAD REG ADMI	Unidades	xxxxxxxxxx	15	85	50 Bs.	xxxxxxxxxx	750 Bs.	4250 Bs.
3	2019-06-02	2222	ENCARGADO ALMACEN	Unidades	xxxxxxxxxx	3	82	50 Bs.	xxxxxxxxxx	150 Bs.	4100 Bs.
	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	118	82	xxxxxxxxxx	xxxxxxxxxx	5900 Bs.	4100 Bs.

Fuente: (Elaboración Propia)

**Figura N° 3. 23 Informe de salida de material**

The screenshot shows a report from the 'GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO G.A.M.E.A.' titled 'INVENTARIO DE KARDEX GENERAL'. It includes a table with the following data:

Datos del Inventario Kardex General												
CATEGORIA DE PRODUCTO		DESCRIPCIÓN DE PRODUCTO			UNIDAD MEDIDA	PERIODO	FECHA I.	FECHA F.				
MATERIAL DE CONSTRUCCION		BOLSA DE CEMENTO VIACHA			Unidades	2019	2019-05-01	2019-11-12				
#	FECHA	N° FACTURA	DETALLE	U. MEDIDA	FISICO			PRECIO U.	VALORADO			
					ENTRADA	SALIDA	SALDO		ENTRADA	SALIDA	SALDO	
1	2019-06-01	2222	BOLSA DE CEMENTO VIACHA	Unidades	200	xxxxxxxxxx	200	50 Bs.	10000 Bs.	xxxxxxxxxx	10000 Bs.	
1	2019-06-02	2222	ENCARGADO ALMACEN	Unidades	xxxxxxxxxx	100	100	50 Bs.	xxxxxxxxxx	5000 Bs.	5000 Bs.	
2	2019-06-03	2222	JEFE DE UNIDAD REG ADMI	Unidades	xxxxxxxxxx	15	85	50 Bs.	xxxxxxxxxx	750 Bs.	4250 Bs.	
3	2019-06-02	2222	ENCARGADO ALMACEN	Unidades	xxxxxxxxxx	3	82	50 Bs.	xxxxxxxxxx	150 Bs.	4100 Bs.	
	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	xxxxxxxxxx	118	82	xxxxxxxxxx	xxxxxxxxxx	5900 Bs.	4100 Bs.	

Fuente: (Elaboración Propia)

## 4.1 INTRODUCCIÓN

En este índice se verá el desarrollo de la medición de la calidad del software mediante la métrica ISO/IEC 9126 lo cual es un estándar internacional para la evaluación de software, que establece que cualquier componente de la calidad pueda ser descrito por las características de Funcionalidad, Confiabilidad, Mantenibilidad, Usabilidad y Portabilidad.

## 4.2 NORMA ISO/IEC 9126

### 4.2.1 Funcionalidad

La funcionalidad no se puede medir directamente por esta razón corresponde derivar medidas directas como es el punto función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario. Para la funcionalidad o medición del sistema, se debe determinar las siguientes características:

La métrica que se utiliza para medir la calidad del software será una métrica llamada punto función. La métrica punto función se centra en la funcionalidad o utilidad del programa, los puntos de función se calculan realizando una serie de actividades comenzando por determinar los siguientes números.

- ❖ **Número de entradas de usuarios.** Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados al sistema.
- ❖ **Número de salidas de usuarios.** Estas se refieren a informes, mensajes de error, es decir salidas que por Proporcionen al usuario información orientada al sistema.
- ❖ **Número de peticiones de usuario.** Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
- ❖ **Numero de archivos.** Se cuenta cada archivo maestro lógico.
- ❖ **Numero de interfaces externas.** Se cuenta todas las interfaces legibles por el ordenador que se solicitadas para transmitir información a otro sistema.

**Número de Entradas de usuario**, se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

*Tabla 4. 1 Número de Entradas de Usuario*

<b>Entradas de Usuario</b>		
1	Módulo de administración de material	6
2	Módulo de administración de usuario	4
3	Módulo de lista de funcionarios	4
4	Módulo de administración proveedores	3
5	Módulo de categorías de material	3
6	Módulo administración unidad y cargo	3
7	Módulo administración tipo de unidades educativas	4
8	Módulo administración kardex	4
9	Módulo productos salientes.	4
10	Módulo reportes	4
11	Módulo kardex general	6
<b>Total</b>		<b>45</b>

**Fuente:** (Elaboración Propia)

**Número de Salidas Usuario**, se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un reporte no se cuentan de forma separada.

*Tabla 4. 2 Número de Salidas Usuario*

<b>Salidas de Usuario</b>		
1	Módulo de administración de material	8
2	Módulo de administración de usuario	6
3	Módulo de lista de funcionarios	8
4	Módulo de administración proveedores	6
5	Módulo de categorías de material	3
6	Módulo administración unidad y cargo	4
7	Módulo administración tipo de unidades educativas	4
8	Módulo administración kardex	4
9	Módulo productos salientes.	4
10	Módulo reportes	4
11	Módulo kardex general	8
<b>Total</b>		<b>59</b>

**Fuente:** (Elaboración Propia)

**Número de Peticiones de usuario**, una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separada.

*Tabla 4. 3 Número de Peticiones del Usuario*

<b>Peticiones del Usuario</b>		
1	Módulo de administración de material	9
2	Módulo de administración de usuario	7

3	Módulo de lista de funcionarios	7
4	Módulo de administración proveedores	7
5	Módulo de categorías de material	4
6	Módulo administración unidad y cargo	6
7	Módulo administración tipo de unidades educativas	4
8	Módulo administración kardex	4
9	Módulo productos salientes.	4
10	Módulo reportes	4
11	Módulo kardex general	7
<b>Total</b>		<b>63</b>

**Fuente:** (Elaboración Propia)

**Número de Archivos**, se Cuenta archivo maestro Lógico

*Tabla 4. 4 Numero de Archivos*

<b>Numero de Archivos</b>		
1	Módulo de administración de material	8
2	Módulo de administración de usuario	4
3	Módulo de lista de funcionarios	6
4	Módulo de administración proveedores	5
5	Módulo de categorías de material	5
6	Módulo administración unidad y cargo	3
7	Módulo administración tipo de unidades educativas	3

8	Módulo administración kardex	5
9	Módulo productos salientes.	2
10	Módulo reportes	2
11	Módulo kardex general	3
<b>Total</b>		<b>46</b>

**Fuente:** (Elaboración Propia)

**Numero de interfaces Externas**, se cuenta todas las interfaces legibles por la máquina.

*Tabla 4. 5 Numero de Interfaces Externas*

<b>Interfaces Externas</b>		
1	Internet	1
<b>Total</b>		<b>1</b>

**Fuente:** (Elaboración Propia)

Para realizar el cálculo de la cuenta Total con factores de ponderación se debe realizar con la siguiente Tabla.

*Tabla 4. 6 Factores de Ponderación*

Parámetros de medida		Cuenta	Factores de Ponderación			Total
			Simple	Medio	Complejo	
1	Nro. de Entradas de Usuario	45	3	4	6	180
2	Nro. de Salidas de Usuario	59	4	5	7	295
3	Nro. de Peticiones de Usuario	63	3	4	6	252

4	Nro. de Archivos	46	7	10	15	460
5	Nro. de Interfaces Externas	1	5	7	10	7
Total						1194

**Fuente:** (Elaboración Propia)

Valores de ajuste de complejidad según las respuestas a las siguientes preguntas que se muestra en la siguiente tabla.

*Tabla 4. 7 Valores de Ajuste de Complejidad*

Nro	Factor de Complejidad	Sin Influencia	Incidental	Moderada	Medio	Significativa	Esencial	Fi
		0	1	2	3	4	5	
1	¿Requiere el sistema copia de seguridad y recuperación?						X	5
2	¿Requiere comunicación de datos?						X	5
3	¿Existen funciones de procesos distribuidos?				X			3
4	¿El rendimiento es crítico?				X			3
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?						X	5
6	¿Entrada de datos EN LINEA?						X	5
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se llevan a cabo?						X	5

	sobre múltiples pantallas o variadas opciones?							
8	¿Se actualizan los archivos maestros de forma interactiva?						X	5
9	¿Son complejas de las entradas de salidas de archivos?			X				2
10	¿Lógica del proceso Interno Compleja?					X		3
11	¿Se diseña el código para ser reutilizable?						X	5
12	¿Están incluidas en el diseño conversiones de instalación?			X				2
13	¿Instalaciones Múltiples?		X					1
14	¿Facilidad de Cambios?						X	5
Factor de Complejidad Total (FCT)								55

**Fuente:** (Elaboración Propia)

Para calcular los puntos función (PF), utilizaremos la relación siguiente

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

Dónde:

Cuenta Total: Nivel de complejidad del sistema con respecto al usuario

**(0.65+0.01\*∑Fi):** Ajuste de complejidad según el dominio de la información.

**0.01:** Factor de conversión, es decir un error de 1%

**0.65:** Valor mínimo de ajuste

Calculando el punto función de según la ecuación:



$$PF = 1194 * [0.65 + 0.01 * 55]$$

$$PF = 1432.8$$

Si calculamos al 100% el nivel de confianza consideramos que la sumatoria de  $F_i=70$  valdría como el máximo valor de ajuste de complejidad se tiene:

$$PF \text{ max} = \text{Cuenta Total} * [1 + 0.01 * \sum F_i]$$

$$PF \text{ max} = 1194 * [0.65 + 0.01 * 70]$$

$$PF = 1611.9$$

Con los máximos valores de ajuste de complejidad se tiene que la funcionalidad real es:

$$\text{Funcionalidad} = \frac{1432.8}{1611.9} = 0,88$$

$$\text{Funcionalidad} = 0,88 * 100 = 88\%$$

Entonces la funcionalidad del sistema es un 88% esto quiere decir que el sistema tiene un 88% que funcione sin riesgos de fallo y operatividad constante y 12% de colapso de sistema.

#### 4.2.2 Confiabilidad

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico.

Para calcular confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras

$$F(t) = f * e^{(-\mu * t)}$$

En el inicio de ejecución  $t_0=0$  lo que significa el tiempo inicial en el cual dará inicio el funcionamiento del sistema.

$$F(0) = f * e^{(-\mu * t_0)}$$

Se observa el trabajo del sistema hasta que produce una falla en el instante T, el cual se aproxima a una variable aleatoria continua.

Como se aproxima a variables aleatorias continuas, la confiabilidad a ser obtenida en términos probabilísticos.

Entonces el término en el cual el sistema trabaja sin falla está dado por la ecuación (2) y tiempo en el cual no falla el sistema está dado por ecuación (3).

$$P(T \leq t) = F(t) \quad (2) \text{ probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \quad (3) \text{ Probabilidad de trabajo sin falla}$$

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla

Conociendo la funcionalidad del 88% del sistema calculamos para el periodo establecido.

$$P(T \leq t) = 1 - F(t)$$

$$F(t) = 1 - 0,88 + e^{(-\frac{1}{10} + 20)}$$

$$F(t) = 1 - 0,129$$

$$F(t) = 0,87$$

La confiabilidad del sistema es del 87% en un periodo de 20 días como tiempo de prueba.

#### 4.2.3 Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos.

El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente fórmula.

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

Mt= Numero de módulos la Versión Actual

Fa= Numero de Módulos en la versión actual que se han añadido

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

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

Calculado el IMS

$$IMS = [11 - (1 + 0 + 0)] / 11$$

$$IMS = 0,90$$

Con lo que podemos decir que el nuevo sistema tiene una estabilidad de 90% que es la facilidad de mantenimiento el 10% restante es el margen de error correspondiente a los cambios y modificaciones efectuados desde el prototipo de la versión actual

Puesto que es un sistema diseñado con los requerimientos actuales con el tiempo surgirán nuevos requerimientos los cuales cambiara el valor índice de madurez del software.

Mantenimiento Correctivo, el sistema presenta diseño modular y es por eso que tolera variaciones en su corrección.

Mantenimiento Adaptativo. Se realiza cuando en la organización se produce algún cambio haciendo que el sistema sufra modificaciones.

El sistema por su programación modular permitirá fácilmente hacer modificaciones en sus módulos o integrar nuevos al sistema.

#### **4.2.4 Usabilidad**

La usabilidad es lo mismo decir facilidad de uso, esta métrica nos muestra el costo de aprender a manejar el producto, lo cual se calcula con la siguiente formula:

$$FU = [(Sum(xi)/n) * 100]$$

Donde:

**Xi:** es la sumatoria de valores

**n:** es el número de preguntas

Para responder a las preguntas se debe considerar la siguiente tabla:

*Tabla 4. 8 Ajuste de Preguntas*

<b>ESCALA</b>	<b>VALOR</b>
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

**Fuente:** (Elaboración Propia)

*Tabla 4. 9 Ajuste de Preguntas*

<b>PREGUNTAS</b>	<b>Respuestas</b>		<b>Ponderación %</b>
	<b>SI</b>	<b>NO</b>	
<b>¿Puede Utilizar con facilidad el sistema?</b>	4	1	80%
<b>¿Puede Controlar operaciones que el sistema solicita?</b>	4	1	80%
<b>¿Las Respuestas del sistema son complicadas?</b>	1	4	80%
<b>¿El Sistema permitió la retroalimentación de información?</b>	5	0	100%
<b>¿El sistema cuenta con interface agradable a la vista?</b>	5	0	100%
<b>¿La respuesta del sistema es satisfactoria?</b>	4	1	80%
<b>¿Le parecen complicada las funciones del sistema?</b>	1	4	80%
<b>¿Se hace difícil o dificultoso aprender a manejar el sistema?</b>	1	4	80%

¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	100%
¿Durante el uso del sistema se produjo errores?	1	4	80%
USABILIDAD			86%

**Fuente:** (Elaboración Propia)

Existe un 86% de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

#### **4.2.5 Portabilidad**

Para la portabilidad del sistema se tomará en cuenta dos aspectos como ser a nivel aplicación, nivel hardware la portabilidad la dividimos en dos secciones portabilidad del lado del servidor y portabilidad del lado del cliente.

##### **❖ Portabilidad lado servidor**

A nivel sistema de software. El sistema porta bajo los siguientes sistemas operativos de la familia Microsoft, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 y todo el sistema operativo libre de Linux.

A nivel de base de datos se utiliza base de datos creada en MySQL, la portabilidad se muestra que la base de datos puede ser migrada SQL server, Oracle y Postgres.

A nivel Hardware el sistema porta bajo las siguientes características mínimas de hardware microprocesador Pentium IV RAM de 1GB Como mínimo espacio en el disco duro 50 Gb como mínimo, monitor SVGA.

Por tratarse de un sistema Web, el proyecto presenta una gran portabilidad, ya que los requerimientos para la implementación y ejecución del software son mínimos, por ello a criterio propio se brinda un donde demuestra el porcentaje de portabilidad.

### 4.3 PRUEBAS DE FUNCIONALIDAD DEL SOFTWARE

La mayoría del software puede probarse de dos maneras diferentes. Conociendo el funcionamiento interno, podemos probar que todos los módulos encajan unos con otros, es decir, desde una visión interna. Estas pruebas son pruebas de caja blanca.

Al conocer las funciones específicas del producto se pueden llevar a cabo pruebas que demuestren que estas funciones son operativas y la búsqueda de errores en dichas funciones. Estas pruebas se realizan desde una visión externa, mediante las pruebas de caja negra.

#### 4.3.1 Técnica de caja blanca

Se proporcionara una medición cuantitativa de la complejidad lógica del sistema.

La complejidad ciclica de un grado de flujo

$V(G)$  establece el número de caminos independientes:

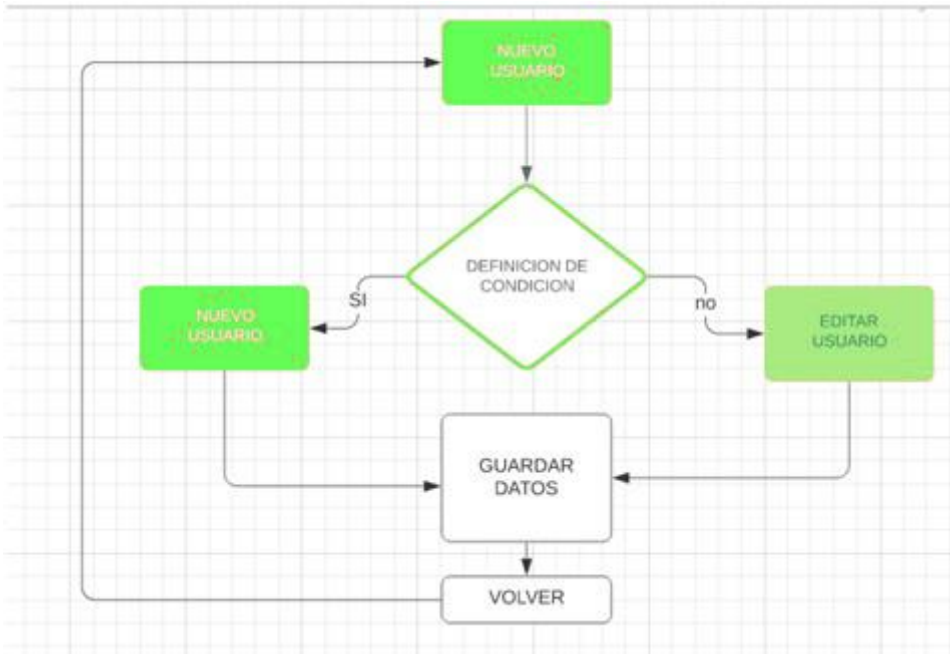
- ❖ El número de regiones del grafico de flujo.
- ❖  $V(G) = A - N + 2$ , donde  $A$  es el número de aristas y  $N$  es el número de nodos.
- ❖  $V(G) = P + 1$ , donde  $P$  es el número de nodos predicado.

#### Módulo de Usuario

Debido al flujo gramal que presenta el módulo de administrador de usuario, se ve la necesidad de realizar la prueba de caja blanca

Pruebas de Caja blanca para el registro de usuario en el sistema.

Figura 4.1 Creación de nuevo usuario flujograma



Fuente (elaboración propia).

Para ilustrar el flujo grama se puede evidenciar el procedimiento en la figura anterior del módulo de administración de usuario.

Figura 4.2 Formulario de creación de usuario

INGRESAR UN NUEVO USUARIO

Para registrar un nuevo usuario, primeramente debe ingresar el número de carnet de identidad...

NRO CARNET # 345235	EXPEDIDO [dropdown]	NOMBRE Ingresar nombre...
PATERNO Ingresar paterno...	MATERNO Ingresar materno...	TELEFONO # Ingresar celular...
FECHA REGISTRO 16/07/2020	UNIDAD PERTENECIENTE [dropdown]	CARGO DE FUNCIONARIO [dropdown]
TIPO USUARIO [dropdown]	Imagen USUARIO [Seleccionar archivo]   Ningún archi... seleccionado	
NOMBRE USUARIO Ingresar usuario...	CONTRASEÑA NUEVA nuevo password...	REPETIR CONTRASEÑA repetir password...

[CANCELAR] [GUARDAR USUARIO]

Fuente: (Elaboración propia).

Figura 4.3 Listar usuario

#	CARNET	NOMBRE & APELLIDO	TIPO DE USUARIO	ESTADO	Action
1	4844706 LP	HERIBERTO CONDORI	ADMINISTRADORA	ACTIVO	ACCION
2	7889798 BN	JUAN VARGAS	ADMINISTRADORA	INACTIVO	ACCION
3	11112221 LP	LUIS FERNANDO MONTENEGRO	ENCARGADO ALMACEN	INACTIVO	ACCION
4	6340953 LP	RUDDY DE LA CRUZ	ENCARGADO ALMACEN	ACTIVO	ACCION
5	686946893 LP	JOSE LUIS CHOQUE	ENCARGADO ALMACEN	ACTIVO	ACCION
6	111222 LP	IVAN VILLCA	ENCARGADO ALMACEN	ACTIVO	ACCION
7	11112222 PD	HERNAN MERCADO	JEFE DE UNIDAD	INACTIVO	ACCION
8	333444 LP	CARLOS CATARI	JEFE DE UNIDAD	ACTIVO	ACCION
9	11112223 LP	MONIKA COPITICONA	SUPERVISOR	ACTIVO	ACCION
10	548548 LP	JAIMF NINA	SUPERVISOR	ACTIVO	ACCION

Fuente: (elaboración propia).

Se muestra la complejidad ciclica de un módulo de administración de usuario por lo que reemplazando el número de predicados en la segunda fórmula se obtiene:

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

Este valor determina dos caminos independientes, dos casos de prueba, para el módulo de administración de usuario. Por lo tanto se realizara las pruebas necesarias para cada uno de los caminos como se muestra en la tabla.

Tabla 4. 10 Valor de determina dos casos de prueba

CAMINO 1	CAMINO 2
<ul style="list-style-type: none"> <li>❖ Ingresar al módulo de administración de usuario.</li> <li>❖ Vuelve a ingresar al botón de nuevo usuario para el nuevo registro.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Ingresar datos en el formulario de administración de usuario.</li> <li>❖ Verifica y valida si existen usuario activo incorrecto en el formulario.</li> <li>❖ Guarda información del usuario en el sistema.</li> <li>❖ Listar todos los usuarios.</li> </ul>

Fuente: (elaboración propia).



### 4.3.2 Técnica caja negra

Se observa aspectos de funcionalidad, operativa en el proceso realizado en la evaluación fundamental del sistema

Se procede a realizar una evaluación de acuerdo a los módulos para cotejar los resultados que entrega el sistema ya sea mediante reportes o interfaces de resultados.

#### Registro de Usuario

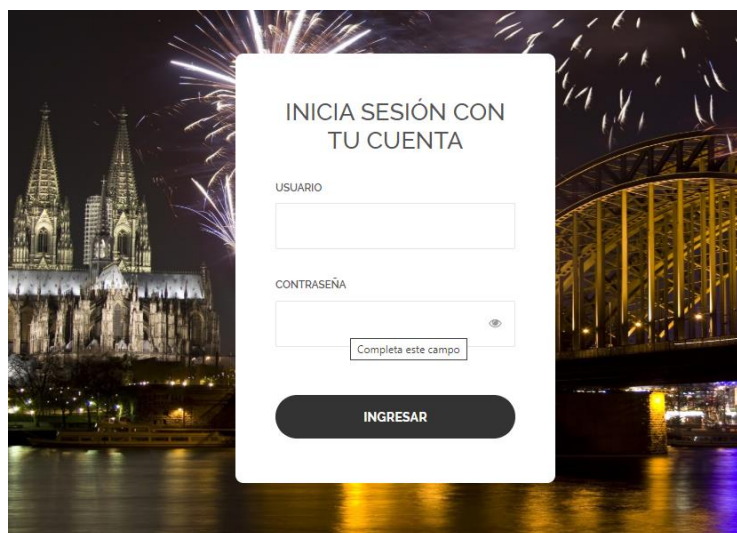
Datos de Ingreso: Se ingresa un usuario y una contraseña asignada por el administrador del sistema en momento otorgado asignación de cargo.

Prueba: Se espera que el sistema realice las respectivas validaciones para el ingreso de funcionarios.

Resultado: El sistema no permite el ingreso al personal no autorizado por el administrador del sistema.

Inicio de sesión al sistema

*Figura 4.4 Inicio de sesión al sistema*



Fuente: (Elaboración propia).

Datos de Ingreso: se ingresa al módulo de inicio.

Prueba: Se espera que el sistema despliegue la información del sistema.

Resultado: El sistema despliega enlaces de todo el menú correspondiente a su privilegio o rol asignado.

Figura 4.5 Interfaz de inicio del sistema



Fuente: (Elaboración propia).

## 4.3 SEGURIDAD

### ISO-27002

La ISO-27002 evalúa y rectifica la implementación mediante el cumplimiento de normas, así como la mejor forma de controles que permitan reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento de la institución en la información, para lo cual se toma los siguientes tipos de seguridad.

#### 4.3.1 Seguridad Lógica.

- ❖ Los respaldos (Back-up) de la base de datos (BD).
- ❖ Los personales involucrados en el proceso del sistema deberán cambiar el Password del sistema periódicamente una vez al mes.
- ❖ Recomendamos al personal involucrado en el sistema cada vez que ingrese al sistema en recomendable que cierre sesión.

#### 4.3.2 Seguridad Física

Seguridad física del entorno de alojamiento de sistema.

Se recomienda los back-up 2 a 3 copias en distintos discos de almacenamiento.

Las copias de la base de datos (BD) deberá ser protegido en un área donde solo el personal autorizado tenga el acceso.

### **4.3.3 Seguridad Organizativa**

En la seguridad del sistema se consideran las siguientes precauciones

- ❖ Autenticación de usuarios
- ❖ Manejo de tipos de usuario en la aplicación
- ❖ Se aplica el paradigma de Modelo Vista Controlador para tener mayor seguridad en el código
- ❖ Manejo de sesiones
- ❖ Manejo de roles y tipos usuario en el gestor de Base de Datos

## **5.1. INTRODUCCIÓN**

Existe distintos métodos para la estimación de costo de desarrollo de software, estos métodos establecen una relación matemática entre el esfuerzo y el tiempo de desarrollo.

## **5.2. CALCULO DE COSTOS**

### **Análisis de costos del valor insumido**

El análisis de costos del proyecto desde el punto de vista del valor de costo insumido, se ha considerado los siguientes parámetros.

Investigación, obtención de fuentes de información de manera online y física caracterizadas por el costo erogado en la obtención de información.

Las herramientas utilizadas para la implementación de la solución de software son programas servicios, open source y gratuitos, previstos por terceros. Software cuyo costo de licencia es gratuito.

En cuanto a costos de infraestructura se ha considerado el uso de equipo excepcionalmente para el sistema, siendo este costo despreciable, ya que la institución cuenta con propios servicios.

También consideramos los costos de recursos humanos por hora para la realización de modelo teórico y el desarrollo del sistema. Las horas se calculan considerando 4 horas laborales por día, sin analizar el detalle de sub actividades que incluyen costos por hora del tipo de actividad realizado, enmarcando el valor de los costos en los horarios que perciben el personal de la unidad.

Por tanto, si bien el proyecto no posee apoyo económico de terceros, la inversión principal será de tiempo, en el desarrollo del módulo teórico y sistema. Igualmente los resultados esperados del proyecto y sus posibles aplicaciones futuras presentan un justo balance a esta inversión.

### **Método de estimación de COCOMO**

La estimación de costos del sistema ha sido desarrollada bajo la KLDC (kilo-líneas de código) como se detalla a continuación.

Para calcular el esfuerzo, necesitaremos hallar la variable KLDC (kilo-líneas de código)

Este proyecto se implementa 7897 Líneas de Código en el lenguaje PHP, aplicado con la herramienta contador de líneas de código SLOC.

Aplicando Conversiones se tiene

$$KLCD = (LCD) / 1000$$

$$KLCD = 7897 / 1000$$

$$KLDC = 7,9 \text{ KLDC}$$

Es un modelo Intermedio y sistema orgánico la evaluación del sistema ha sido considerada bajo las 7897 KLDC los coeficientes que se usarán serán los valores que se detallan en la siguiente tabla.

*Tabla 5. 1 Aplicación del Modelo Intermedio*

PROYECTO SOFTWARE	A	B	C	D
<b>Orgánico</b>	3,2	1,05	2,5	0,38
<b>Semiacoplado</b>	3,0	1,12	2,5	0,35
<b>Empotrado</b>	2,8	1,20	2,5	0,32

**Fuente:**(S. Pressman, 2010)

A continuación, presentaremos las ecuaciones que nos permitirán el costo total de software:

*Tabla 5. 2 Ecuaciones del modelo COCOMO*

VARIABLE	ECUACIÓN	TIPO/UNIDAD
Esfuerzo requerido por el proyecto	$E = a \times (KLDC)^b \times FAE$	<b>Personas/mes</b>
Tiempo requerido por el proyecto	$T = c \times (E)^n$	<b>mese</b>
Número de personas requeridas para el proyecto	$NP = \frac{E}{T}$	<b>personas</b>
Costo total	$CT = \text{Sueldo Mes} \times NP \times T$	<b>\$us</b>

**Fuente:**(S. Prentice-Hall, 1981)

Para hallar los valores de FAE, se utilizará la tabla de atributos multiplicadores

*Tabla 5. 3 Calculo de Atributos FAE*

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Atributos de software</b>						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos de hardware</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos de personal</b>						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
<b>Atributos del proyecto</b>						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	
<b>Total</b>			<b>0,72</b>			

**Fuente:** (Elaboración Propia)

Por tanto, nuestro Factor de ajuste será

$$FAE=0,72$$

Aplicando y remplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE \text{ (Personas/Mes)}$$

$$E = 3,2 * 7,9^{1,05} * 0,72$$

$$E = 20 \text{ (Personas Mes)}$$

### **Cálculo del Tiempo**

$$T = c * Esfuerzo^d \text{ (Meses)}$$

$$T = 2,5 * 20^{0,38}$$

$$T = 7.80 \text{ (Meses)}$$

### **Calculo de la Productividad**

$$PR = \frac{LCD}{Esfuerzo} \text{ (Meses)}$$

$$PR = \frac{7897}{20} \text{ (Meses)}$$

$$PR = 394.8 \text{ (LCD/personas Mes)}$$

Calculo del personal requerido

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

$$P = \frac{20}{7.80} \text{ (Personas)}$$

$$P = 2.56 \text{ (Personas) equivalente a 3 personas}$$

### **Costo Total del Proyecto**

(Coste Mes) = P\* Salario medio entre los programadores y analistas

**Costo Persona Mes 350\$**

$$\text{Coste Mes} = 3 * 350 = 1050 \$us$$

$$\text{Costo Total} = 1050 * 8 = 8400 \$us$$

En resumen, se requiere 3 personas estimando un trabajo de 8 meses y con costo total de 8400\$us Equivalente en bolivianos a 57960 Bs.



## **6.1 CONCLUSIONES**

Se concluyó con los objetivos planteados en el presente proyecto, desarrollando e implementación de un sistema web de gestión y control de almacenes de la Unidad de Servicios generales y mantenimiento dependiente del GAMEA, que coadyuve con la administración y la optimización de la información, para los almacenes que se tiene en dicha unidad.

El sistema web de gestión y control de almacenes fue desarrollado de manera satisfactoria haciendo el uso de la metodología UWE con el cual se realizó, diseño y desarrollo para su implantación.

- ❖ Se tiene una mejor administración de la información de los almacenes con las que cuenta esta unidad.
- ❖ El sistema implementado cuenta con las normas de seguridad
- ❖ Cuentas con los reportes necesarios actuales y esta opción ayuda a la hora de toma de decisiones en cuanto a atención de solicitudes que se tiene.
- ❖ Aplicando con éxito las normas de calidad, paradigma de Modelo Vista Controlador con PHP5 y las herramientas de programación para que tenga alta usabilidad y funcionalidad.
- ❖ Se diseñó una base de datos acorde a los requerimientos de los usuarios.
- ❖ Se realizó una prueba de calidad del sistema con las herramientas adecuadas y necesarias.

### **6.1.2 RECOMENDACIONES**

En base a las políticas de seguridad propuesta y las observaciones realizadas durante las pruebas se elabora las siguientes recomendaciones.

- ❖ Se sugiere implementar el sistema a las diferentes unidades y direcciones que tiene el gobierno autónomo municipal de El Alto
- ❖ Es recomendable que se pueda implementar este sistema a nivel nacional con las modificaciones de acuerdo a los requerimientos solicitados.
- ❖ Es necesario realizar copias de seguridad de la base de datos

- ❖ Asignar un administrador de desarrollo de software, que se encargue en el mantenimiento respectivo del sistema, que actualice a los nuevos requerimientos.
- ❖ Se debe tener sumo cuidado respecto a las claves de acceso.

Las actualizaciones deben ser coordinadas con el o los responsables de almacén.

## Bibliografía

- Alsina, G. G. (25 de noviembre de 2018). *Definición de AJAX*. Recuperado el 15 de Julio de 2019, de Definición de AJAX:  
<https://www.definicionabc.com/tecnologia/ajax.php>
- Alvarez, M. A. (09 de mayo de 2001). *Qué es PHP*. Recuperado el 15 de Julio de 2019, de Qué es PHP: <https://desarrolloweb.com/articulos/392.php>
- Alvarez, M. A. (02 de enero de 2014). *Desarrolloweb.com*. Recuperado el 12 de marzo de 2019, de Que es MVC: <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Bernal., G. (16 de Junio de 2019). *Descripción completa del servidor web Apache*. Recuperado el 15 de Julio de 2019, de <https://www.hostinger.es/tutoriales/que-es-apache/>
- Camacho, F. (5 de Abril de 2016). *Diferencias metodologicas*. Recuperado el 2/ de marzo de 2019, de diferencias metodologicas:  
[http://diferenciasmetodologias.blogspot.com/2016/04/las-diferencias-entre-las-metodologias\\_5.html](http://diferenciasmetodologias.blogspot.com/2016/04/las-diferencias-entre-las-metodologias_5.html)
- Chavez, K. (29 de Noviembre de 2012). *Ingeniería Web*. Recuperado el 08 de Diciembre de 2013, de <http://kevinchavez93.blogspot.com/2012/11/ingenieria-web.html>
- COCOMO. (Octubre de 2013). Recuperado el 12 de Diciembre de 2013, de <http://es.wikipedia.org/wiki/COCOMO>
- Colectiva, N. (9 de TUTORIALESABRIL de 2015). *CodeIgniter*. Obtenido de <http://blog.nubecolectiva.com/url-amigables-con-codeigniter/>
- Dianelys, V. R. (25 de noviembre de 2014). Recuperado el 12 de Julio de 2019, de <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
- Engineering, I. f. (07 de 09 de 2012). *UWE - UML - BASED WEB ENGINEERING*. Recuperado el 18 de 07 de 2013, de <http://uwe.pst.ifi.lmu.de/>

- Esquen, L. A. (16 de Diciembre de 2015). *Mantenimiento de software*. Recuperado el 10 de Agosto de 2019, de Mantenimiento de software:  
<https://www.slideshare.net/leyneradanabadesquen/mantenimiento-desoftwarev6-abadesquenleyner>
- Estela, M. (10 de Enero de 2019). *Base de datos*. Recuperado el 10 de Junio de 2019, de Base de datos: <https://concepto.de/base-de-datos/>
- Fontán, M. (25 de octubre de 2012). *CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones web*. Recuperado el 10 de Julio de 2019, de ¿Qué es MVC?: <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- Gomez, A. (23 de febrero de 2002). *Cocoma*. Recuperado el 18 de junio de 2019, de Un modelo de estimacion de proyectos de software:  
<http://sunset.usc.edu/COCOMOII/cocoma.html>
- González, G. A. (28 de noviembre de 2016). *Definición de JavaScript*. Recuperado el 15 de Julio de 2019, de JavaScript:  
<https://www.definicionabc.com/tecnologia/javascript.php>
- Hernández Salguera , Á. I. (4 de Diciembre de 2012). *Ingeniería del Software*. Recuperado el 10 de Octubre de 2013, de <http://ingsoftaihs.blogspot.com/>
- Jiménez, J. D. (20 de Enero de 2019). Recuperado el 15 de Julio de 2019, de <https://openwebinars.net/blog/que-es-css3/>
- KOCH, N. (2000). *Software Engineering for Adaptive Hypermedia Systems. Referece Modeling Techniques and Development Process*. Germany: Ludeing.Maximilians-Universität München.
- Lopez, B. S. (16 de julio de 2019). *Ingeniería Industrial*. Recuperado el 12 de junio de 2020, de <https://www.ingenieriaindustrialonline.com/gestion-de-inventarios/metodos-de-valoracion-de-inventarios/>
- Martin, A. (24 de Abril de 2019). *HTML5*. Recuperado el 15 de Julio de 2019, de HTML5: <https://developer.mozilla.org/es/docs/HTML/HTML5>

- MARTINEZ, K. (4 de abril de 2011). *TODO SOBRE PHP*. Recuperado el 12 de Julio de 2019, de VENTAJAS Y DESVENTAJAS:  
<http://klarimartinezbenjumea.blogspot.com/2011/04/ventajas-y-desventajas.html>
- Maximilians, L. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/teachingTutorialPresentationSpanish.html>
- Mercado, J. (27 de Agosto de 2018). Recuperado el 11 de Junio de 2019, de <http://www.pmoinformatica.com/2018/08/tecnicas-estimacion-software.html>
- MOF. (2018). Manual de organizacion y funciones. En G. M. Alto, *Manual de organizacion y funciones* (pág. 232). El Alto: GAMEA.
- Pandini, W. (30 de Diciembre de 2005). *seguridad de la información*. Recuperado el 10 de Junio de 2019, de ISO 27002: Buenas prácticas para gestión de la seguridad de la información.
- Peño, J. M. (12 de marzo de 2015). *Pruebas de software Fundamentos y tecnicas*. Recuperado el 16 de diciembre de 2019, de Pruebas de software Fundamentos y tecnicas: <http://www.sstqb.es/ficheros/sstqb.fives95-a69acf.pdf>
- Porto, J. P. (30 de Diciembre de 2018). Recuperado el 15 de julio de 2019, de <https://definicion.de/jquery/>
- PRESSMAN, R. (2002). *Ingeniería de Software. Un Enfoque Práctico. Quinta Edición* (QUITA EDICIÓN ed.). MEXICO: MC GRAW HILL.
- Ramos, P. (12 de 01 de 2018). *MariaDB*. Recuperado el 10 de Junio de 2019, de MariaDB: <https://styde.net/mariadb-vs-mysql/>
- ruiz, a. (20 de septiembre de 2014). *Bootstrap*. Recuperado el 15 de Julio de 2019, de ¿Qué es Bootstrap?:  
<https://www.mastermarketingdigital.com/everriculum/2014/09/20/que-es-bootstrap/>
- Soto, A. M. (2008). *Diagramas de Casos de Uso*. Obtenido de <https://es.scribd.com/doc/82603040/Casos-de-Uso-explicacion-y-ejemplos>

Wikipedia. (5 de noviembre de 2015). *Seguridad informática*. Recuperado el 19 de Junio de 2019, de [https://es.wikipedia.org/wiki/Seguridad\\_inform%C3%A1tica](https://es.wikipedia.org/wiki/Seguridad_inform%C3%A1tica)

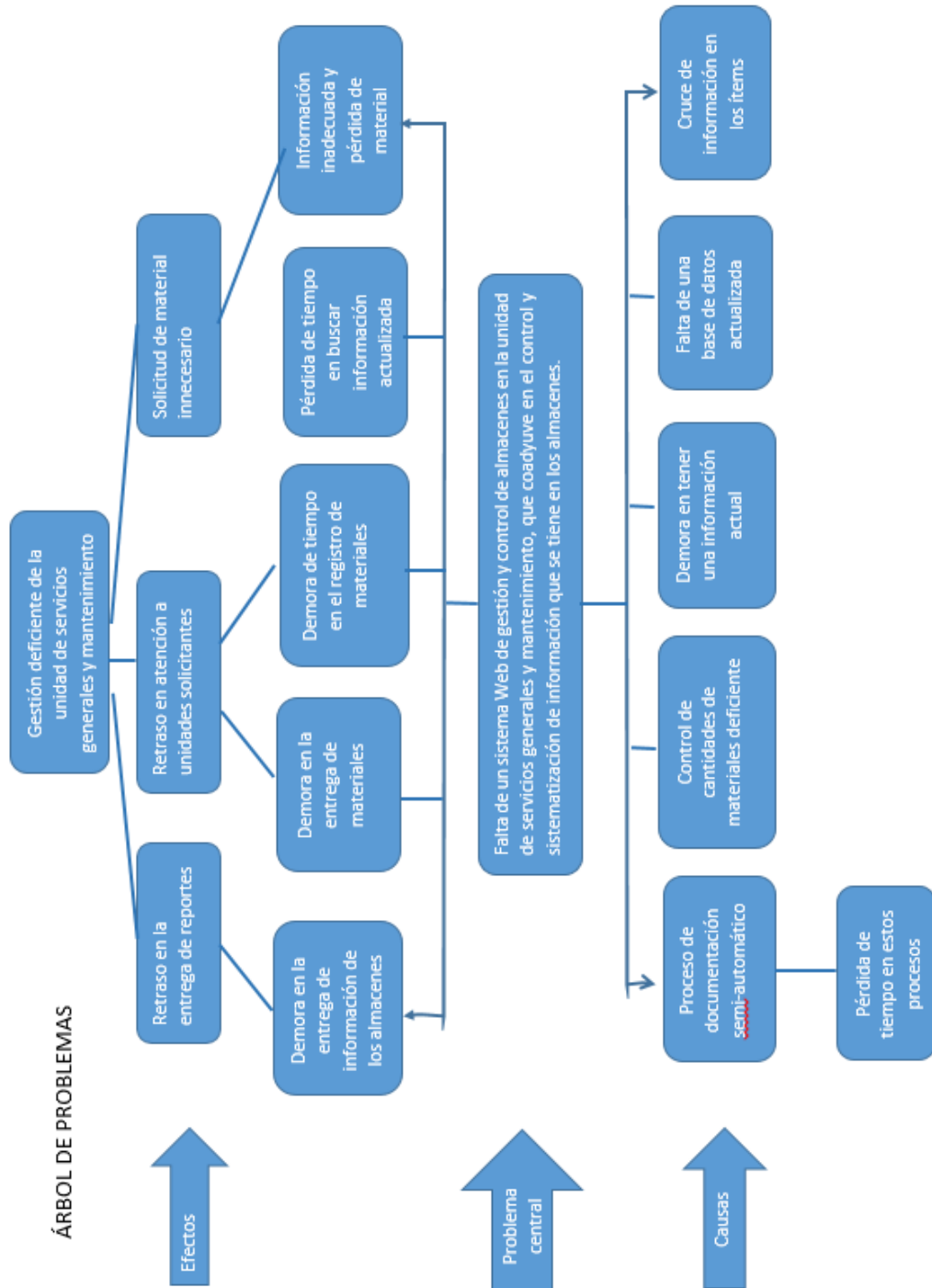
Zegarra, J. C. (25 de junio de 2015). *Web System*. Recuperado el 5 de julio de 2019, de Web System: <https://websystemperu.com/sistemas-web>

# ANEXOS

# **ANEXO A**

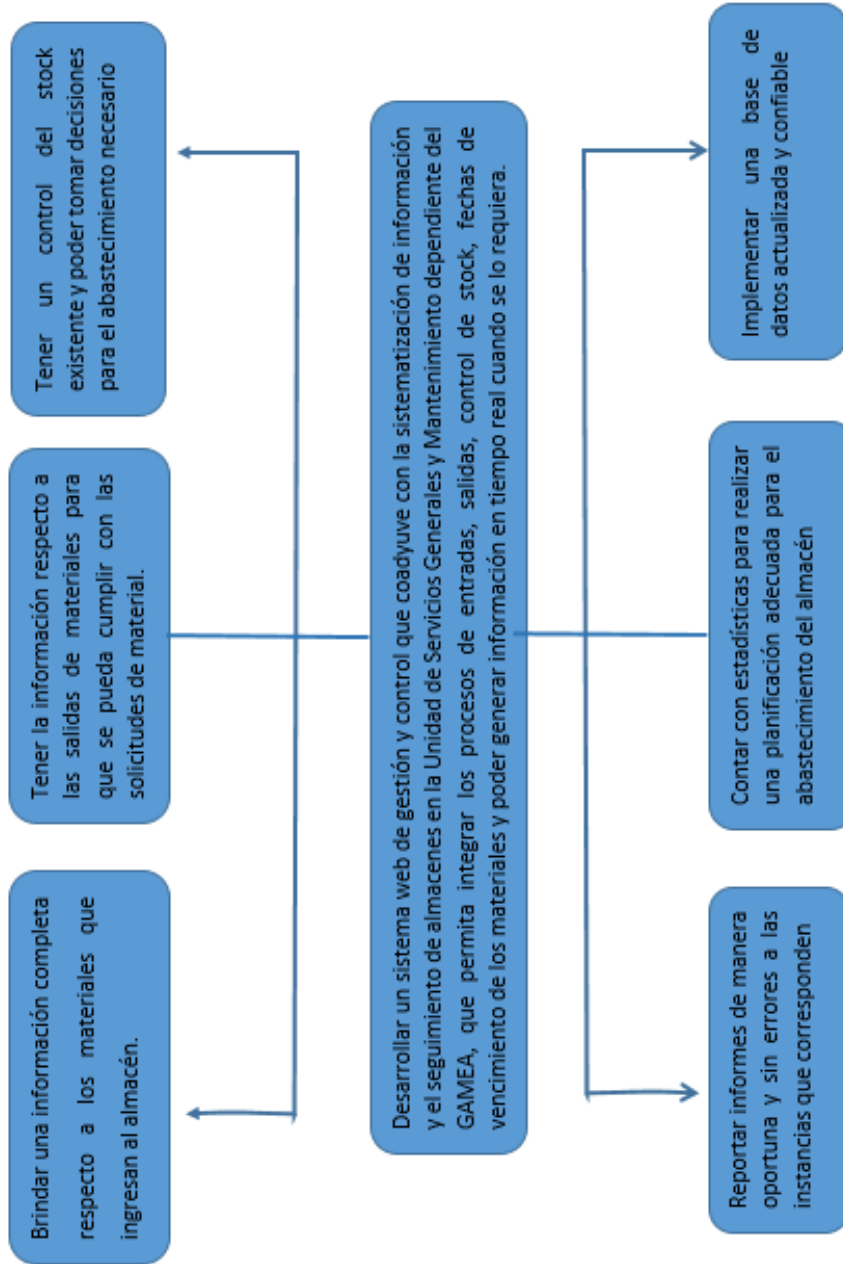


# ÁRBOL DE PROBLEMAS



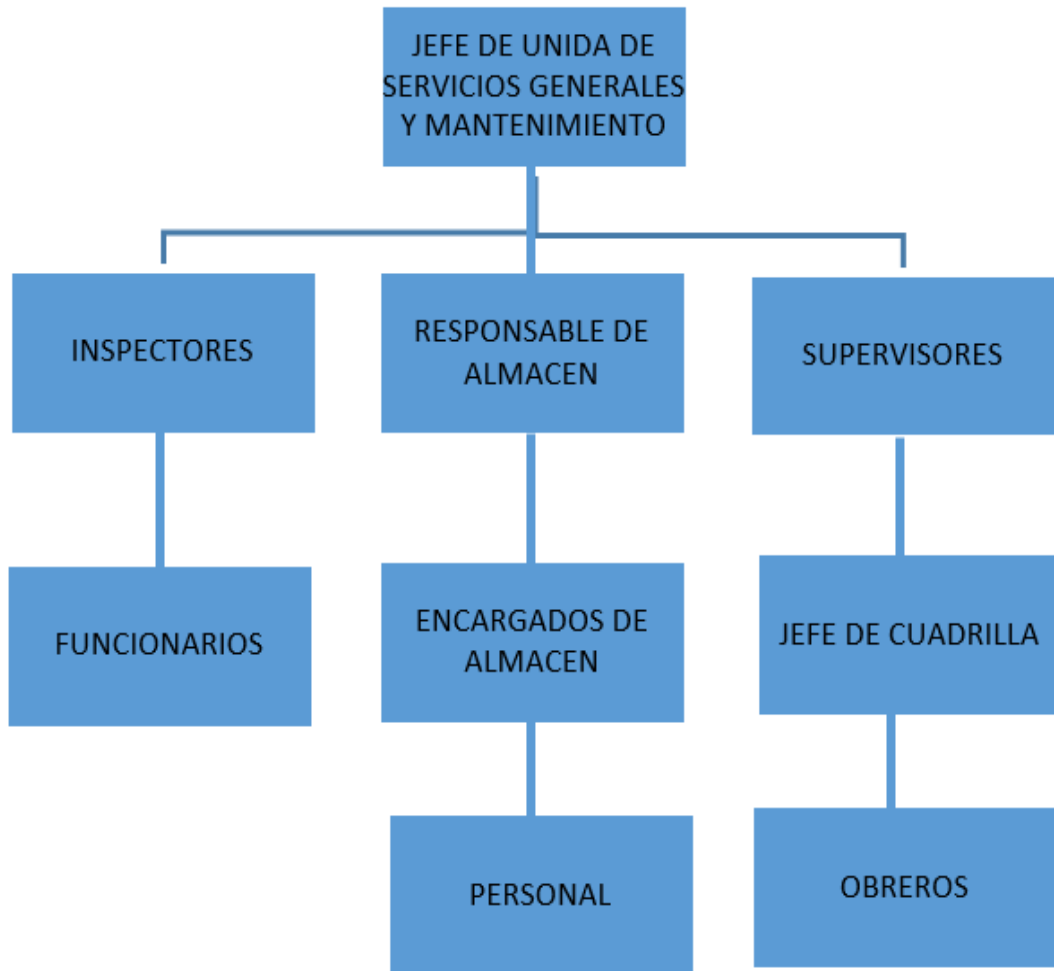
# **ANEXO B**

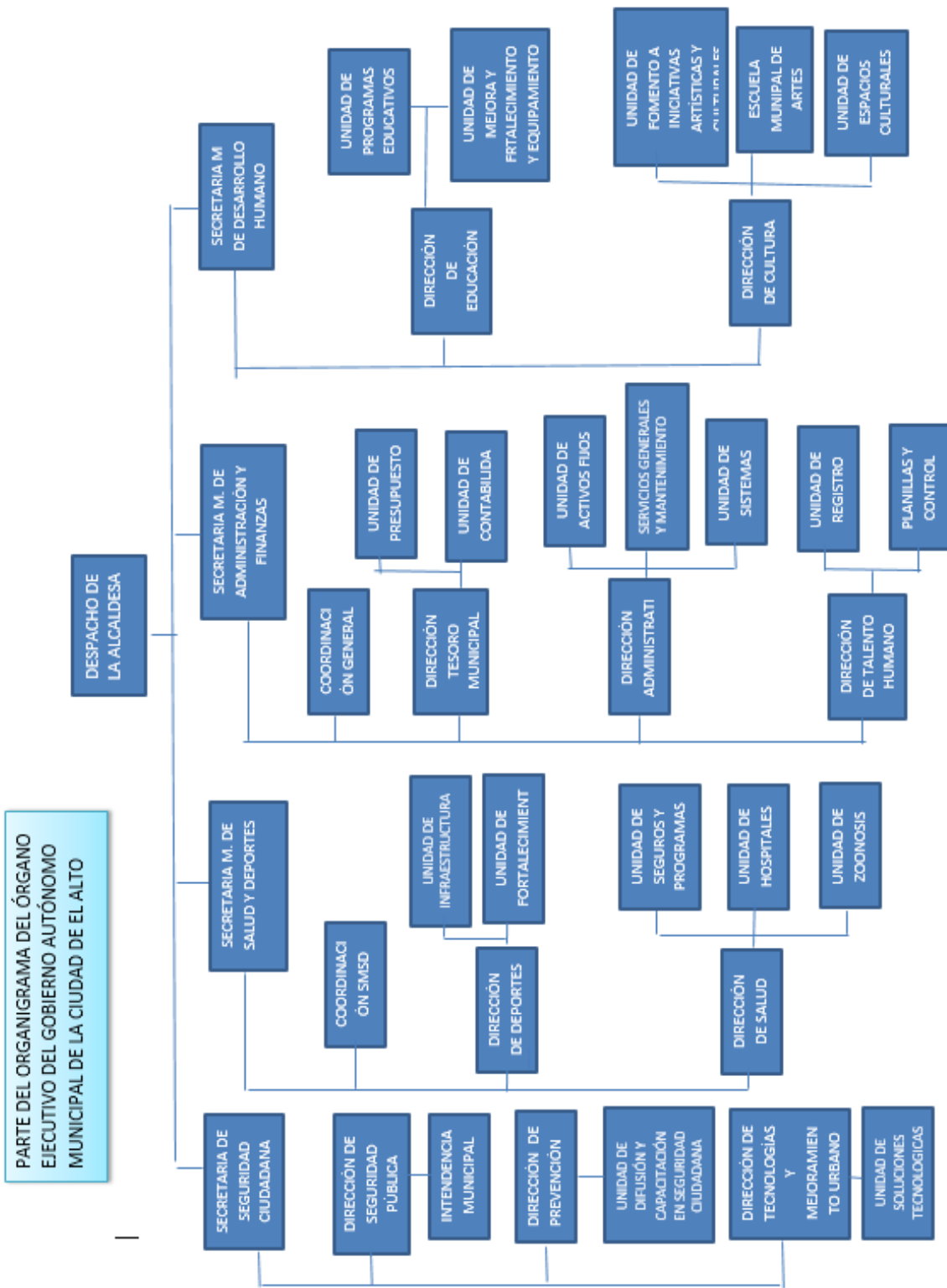
## ÁRBOL DE OBJETIVOS



# **ANEXO C**

## ORGANIGRAMA SERVICIOS GENERALES





# **ANEXO D**

# **ANEXO E**





## GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO

El Alto, 03 de marzo de 2020



Señora:

Ing. Marisol Arguedas Balladares  
Docente Taller de licenciatura II  
Ingeniería de sistemas UPEA  
Presente:

### REF. AVAL DE CONFORMIDAD.

Distinguida Ingeniera.


Mediante la presente comunico mi conformidad por el proyecto titulado **“SISTEMA WEB DE GESTIÓN Y CONTROL DE ALMACENES” CASO: UNIDAD DE SERVICIOS GENERALES Y MANTENIMIENTO GAMEA**. Que propuso el señor **Heriberto Condori Cahuapaza** con CI **6014525 LP** para su defensa pública y posterior evaluación de la materia Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

El presente proyecto ha cumplido con los requerimientos de los almacenes de la unidad de servicios generales y mantenimiento y manifestar el interés con el que fue elaborado para su aplicación a favor del personal encargado de los almacenes.

Con este motivo saludo a usted con las consideraciones del caso.

Atentamente

  
  
Cesar Montaña Guíñez  
JEFE UNIDAD DE SERVICIOS  
GENERALES Y MANTENIMIENTO  
Dirección Administrativa  
GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO

  
Ruddy A. De La Cruz Quispe

El Alto, 10 de julio de 2020

Señor:

Ing. David Carlos Mamani Quispe

**DIRECTOR DE CARRERA**

**INGENIERIA DE SISTEMAS U.P.E.A.**

Presente:

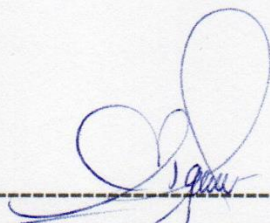
**REF. AVAL DE CONFORMIDAD.**

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad sobre el proyecto de grado titulado "**SISTEMA WEB DE GESTIÓN Y CONTROL DE ALMACENES**" caso: **Unidad de Servicios Generales y Mantenimiento GAMEA**. Propuesta por el universitario Heriberto Condori Cahuapaza con CI 6014525 LP para su defensa pública y posterior aprobación de la materia Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular reciba saludos cordiales.

Atentamente



---

Ing. Marisol Arguedas Balladares  
Tutor Metodológico

El Alto, 22 de Junio de 2020

Señora:

Ing. Marisol Arguedas Balladares

Tutor metodológico

Presente:

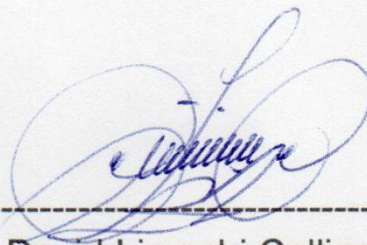
**REF. AVAL DE CONFORMIDAD.**

Distinguida Ingeniera.

Mediante la presente tengo a bien comunicarle mi conformidad por la propuesta de proyecto de grado titulado "SISTEMA WEB DE GESTIÓN Y CONTROL DE ALMACENES" caso: Unidad de Servicios Generales y Mantenimiento GAMEA. Propuesta por el universitario Heriberto Condori Cahuapaza con CI 6014525 LP para su defensa pública y posterior evaluación de la materia Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular reciba saludos cordiales.

Atentamente



---

Ing. David Limachi Callizaya  
Tutor especialista

El Alto, 06 de julio de 2020

Señora:  
Ing. Marisol Arguedas Balladares  
Tutor Metodológico  
Presente:

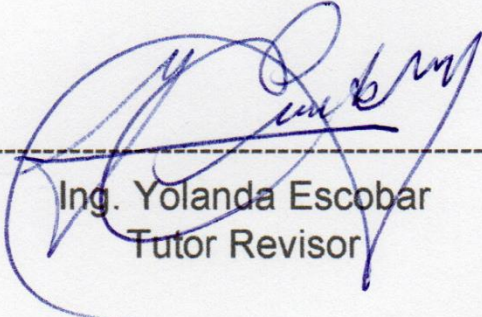
**REF. AVAL DE CONFORMIDAD.**

Distinguida Ingeniera.

Mediante la presente tengo a bien comunicarle mi conformidad por el proyecto de grado desarrollado "**SISTEMA DE GESTIÓN Y CONTROL DE ALMACENES**" caso: **Unidad de Servicios Generales y Mantenimiento GAMEA**. Propuesta por el universitario **Heriberto Condori Cahuapaza** con CI **6014525** LP para su defensa pública y posterior evaluación de la materia Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular reciba saludos cordiales.

Atentamente



---

Ing. Yolanda Escobar  
Tutor Revisor