

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

#### SISTEMA DE INFORMACIÓN DE EVALUACIÓN DOCENTE PARA LA CARRERA COMERCIO INTERNACIONAL

CASO: Universidad Pública de El Alto

Para optar al título de Licenciatura en Ingeniería de Sistemas

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante :** Univ. Gustavo Pérez Mamani

**Tutor Metodológico :** Ing. Marisol Arguedas Balladares

**Tutor Especialista :** Lic. Mario Torrez Cupiticono

**Tutor revisor :** Ing. Yolanda Escobar Mancilla

EI ALTO - BOLIVIA

2020

## **DEDICATORIA**

*Dedicarlo este trabajo con mucho cariño:*

*A Dios que me ha dado la vida y la fortaleza para concluir este proyecto de grado.*

*De la misma manera a mis estimados padres y hermanos por el gran esfuerzo, amor y confianza que depositaron en mí, por sus sabias enseñanzas y consejos en largo de mi carrera universitaria.*

## **AGRADECIMIENTO**

A Dios todo poderoso, por su sabiduría que me otorgo y me permitió finalizar el presente proyecto una de mis metas.

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

A la Universidad Pública de El Alto, (UPEA) por la oportunidad que me brindo al formarme profesionalmente en sus aulas.

A los docentes de la UPEA, en especial a la carrera de Ingeniería de Sistemas quienes con sus conocimientos y enseñanzas hicieron posible el desarrollo de nuestra formación profesional.

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 hacia mi persona.

A mi tutor especialista Lic. Mario Torrez Cupiticono por compartir su conocimiento, brindarme su orientación, sugerencias con paciencia y motivación durante el desarrollo del presente proyecto.

A mi tutor revisor Ing. Yolanda Escobar Mancilla por su disponibilidad de tiempo, por orientación y las observaciones brindadas por su amplio conocimiento en la realización de proyecto.

Por ultimo agradecer a la carrera Comercio Internacional, por la oportunidad de permitirme desarrollar el proyecto y llevarlo a cabo hasta su implementación.

## RESUMEN

*La ciencia y la tecnología van creciendo de manera acelerada y junto a ellas avanzan las ciencias informáticas y computacionales, las cuales se encuentran inmensas en nuestras vidas.*

*Se denomina aplicación Web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.*

*Se sabe que en la actualidad la sociedad está completamente informatizada, es un hecho que millones de personas están conectadas por medio de computadoras a la red. Todos los equipos no solo nos sirven para la búsqueda de información en internet si no también se las usan en entidades públicas o privadas para el manejo de información, control de personal, administración de actividades y toma de decisiones.*

*El presente proyecto tiene por objeto desarrollar un sistema de información de evaluación docente como herramienta para la carrera Comercio Internacional que proporcione datos oportunos y confiables para la toma de decisiones de la dirección de carrera de Comercio Internacional de la Universidad Pública de El Alto.*

*El sistema utilizará como gestor de base de datos MariaDB, lenguaje de programación PHP y de framework como codeigniter de la misma manera las herramientas para lado de front-end se utilizan javascript, JQuery, Ajax, Html5, Css3 y Bootstrap.*

*También podemos mencionar el uso de la metodología UWE es una propuesta basada en el proceso unificado y UML o también llamado perfil UML. UWE hace uso de notación UML pura y los tipos de diagramas UML en donde sea posible para el análisis y diseño de aplicaciones WEB. Cabe destacar para la métrica calidad de software se ha utilizado la ISO/IEC 9126, esta es un norma Internacional que fue publicada en 1992, la cual es usada para la evaluación de la calidad de software.*

*Finalmente se realiza el análisis de costos COCOMO y su modelo de estimación Post-Arquitectura.*

## ÍNDICE GENERAL

CAPITULO I MARCO PRELIMINAR .....	1
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES .....	2
1.2.1. Antecedentes Institucionales .....	2
1.2.2. Antecedentes Internacionales .....	3
1.2.3 Antecedentes Nacionales.....	4
1.3. PLANTEAMIENTO DE PROBLEMA .....	5
1.3.1. Problema Principal .....	5
1.3.2. Problemas Secundarios .....	5
1.4. OBJETIVOS.....	6
1.4.1. Objetivo General .....	6
1.4.2 Objetivo Específicos.....	6
1.5. JUSTIFICACIÓN .....	7
1.5.1. Justificación Técnica .....	7
1.5.2. Justificación Económica .....	7
1.5.3. Justificación Social .....	8
1.6. HERRAMIENTAS.....	8
1.6.1. Herramientas para desarrollo Web.....	8
1.7. METODOLOGÍA .....	9
1.8. LÍMITES Y ALCANCES.....	11
1.8.1. Limites.....	11
1.8.2. Alcances .....	11
1.9. APORTES.....	12
CAPITULO II MARCO TEÓRICO.....	13
2.1 DEFINICIÓN DEL SISTEMA.....	13
2.2 SISTEMA DE INFORMACIÓN .....	14
2.2.1 Características de un sistema de información .....	14
2.2.2 Componentes de un sistema de información.....	15
2.2.3 Actividad de sistema de información .....	15
2.3 DEFINICIÓN DE SOFTWARE .....	16

2.3.1 Tipos principales de software .....	16
2.4 EVALUACION .....	17
2.4.1 Concepto de evaluación .....	17
2.4.2 Dimensiones de la evaluación .....	18
2.4.3 Evaluación de desempeño docente por estudiantes universitario .....	18
2.4.4 Evaluación del desempeño docente .....	19
2.4.5 Modelos de evaluación de desempeño docente .....	20
2.4.5.1 Instrumento de evaluación de desempeño docente por estudiantes .....	21
2.5 WEB WWW (World Wide Web) .....	22
2.5.1 Ingeniería Web .....	23
2.5.1.1 Aplicaciones de la Web .....	23
2.5.1.2 Tipos de página Web .....	24
2.5.2 Arquitectura .....	26
2.5.2.1 Seguridad de aplicaciones Web .....	31
2.5.2.2 Arquitectura cliente – servidor .....	33
2.5.2.3 Cliente .....	33
2.5.2.4 Servidor .....	34
2.5.3 Estructura Web .....	34
2.6 HERRAMIENTAS DE DESARROLLO .....	35
2.6.1 Servidor apache .....	35
2.6.2 Gestor de base de datos .....	37
2.6.2.1 Gestor de base de datos MariaDB .....	38
2.6.3 Lenguaje de programación PHP7 .....	39
2.6.3.1 Framework CODEIGNITER .....	40
2.6.3.2 Definición de MVC .....	40
2.6.4 Herramienta de diseño .....	42
2.6.4.1 Definición de html5 .....	42
2.6.4.2 Definición de CSS3 .....	43
2.6.4.3 Definición de jQuery .....	44
2.6.4.4 Definición de javascript .....	44
2.6.4.5 Definición de AJAX .....	46

2.6.4.6 Framework bootstrap .....	46
2.7 INGENIERÍA WEB BASADA EN UML (UWE) .....	47
2.7.1 Características .....	48
2.7.2 Metodología de desarrollo UWE.....	48
2.7.3 Característica de la metodología UWE.....	49
2.7.4 Fase de la metodología UWE.....	50
2.7.4.1 Captura, análisis y especificación de requisitos.....	50
2.7.4.2 Diseño del sistema .....	50
2.7.4.3 Codificación del software.....	51
2.7.4.4 Pruebas.....	51
2.7.4.5 La Instalación o fase de Implementación.....	51
2.7.4.6 Mantenimiento.....	52
2.7.5 Modelo de la metodología UWE .....	52
2.7.5.1 Modelo de caso de uso .....	52
2.7.5.2 Modelo de contenido .....	55
2.7.5.3 Modelo de navegación .....	56
2.7.5.4 Modelo de estructura del proceso .....	58
2.7.5.5 Modelo de presentación .....	59
2.8 MÉTRICA DE CALIDAD DE SOFTWARE.....	61
2.8.1 ISO 9000 .....	63
2.8.2 Estándar ISO/IEC 9126.....	63
2.8.2.1 Métricas de calidad del modelo de ISO-9126 .....	64
2.8.2.1.1 Norma de calidad ISO-9126 .....	65
2.8.2.1.1.1 Funcionalidad .....	68
2.8.2.1.1.1.1 Punto función. ....	68
2.8.2.1.1.2 Confiabilidad .....	71
2.8.2.1.1.3 Usabilidad .....	72
2.8.2.1.1.4 Mantenibilidad .....	73
2.8.2.1.1.5 Portabilidad .....	74
2.9 ANALISIS DE COSTOS DE COCOMO .....	75
2.9.1 COCOMO .....	75

2.9.1.1 Modelos de estimación de costo con COCOMO .....	76
2.9.1.2 Atributos de coste.....	80
2.9.1.2.1 Modelo básico .....	80
2.9.1.2.2 Modelo intermedio.....	81
2.9.1.2.3 Modelo detallado .....	84
2.10 SEGURIDAD DE LA INFORMACIÓN .....	85
2.10.1 Definición de la seguridad informática .....	85
2.10.2 Estándar ISO/IEC 27000 .....	85
2.10.3 Estándar ISO/IEC 27002.....	86
2.10.3.1 Seguridad lógica .....	87
2.10.3.2 Seguridad física .....	87
2.10.3.3 Seguridad organizativa.....	88
2.10.4 caja negra y blanca .....	88
2.10.4.1 Pruebas (Diseño de Casos de Prueba) .....	88
2.10.4.2 Pruebas de Caja Negra.....	89
2.10.4.3 Pruebas de Caja Blanca.....	89
CAPITULO III MARCO APLICATIVO .....	90
3.1 INTRODUCCION .....	90
3.2 INGENIERÍA DE REQUERIMIENTOS .....	90
3.2.1 Requerimientos funcionales .....	91
3.2.2 Requerimientos no funcionales .....	92
3.3 FASE DE LA METODOLOGÍA UWE.....	92
3.3.1 Fase captura, análisis y especificación de requisitos. ....	92
3.3.2 Fase diseño del sistema.....	92
3.3.3 Fase codificación del software. ....	94
3.3.4 Fase pruebas. ....	94
3.3.5 Fase Instalación o fase de Implementación.....	98
3.3.6 Fase mantenimiento:.....	99
3.4 APLICACION DEL MODELO UWE .....	99
3.4.1 Modelo de casos de uso .....	99
3.4.1.1 Análisis de requerimientos .....	99

3.4.1.2 Diagrama de caso de uso comercial. ....	99
3.4.1.2.1 Modelo de caso de uso: General del Sistema .....	100
3.4.1.2.2 Diagrama de caso de uso: kardex .....	101
3.4.1.2.3 Diagrama de caso de uso: comisión evaluador .....	102
3.4.1.2.4 Diagrama de caso de uso: docente .....	105
3.4.1.2.5 Diagrama de caso de uso: estudiante .....	106
3.4.2 Modelo contenido .....	107
3.4.3 Modelo de navegación .....	109
3.4.3.1 Modelo de navegación: kardex.....	109
3.4.3.2 Modelo de navegación: comisión evaluador .....	110
3.4.3.3 Modelo de navegación: docente.....	110
3.4.3.4 Modelo de navegación: estudiante .....	111
3.4.4 Modelo de estructura de proceso. ....	111
3.4.5 Modelo de presentación .....	112
3.4.5.1 Modelo de presentación: kardex.....	112
3.4.5.2 Modelo de presentación: comisión evaluador.....	113
3.4.5.3 Modelo de presentación: docente.....	114
3.4.5.4 Modelo de presentación: estudiante .....	115
CAPITULO IV CALIDAD Y SEGURIDAD .....	117
4.1. INTRODUCCIÓN .....	117
4.2 MÉTRICAS DE CALIDAD DE SOFTWARE .....	117
4.2.1 Funcionalidad.....	117
4.2.2 Confiabilidad .....	122
4.2.3 Mantenibilidad .....	123
4.2.4 Usabilidad .....	124
4.2.5 Portabilidad .....	125
4.2.5.1 Portabilidad lado servidor .....	125
4.3 SEGURIDAD DEL SISTEMA .....	126
4.3 .1 ISO-27002.....	126
4.3.1.1 Seguridad lógica. ....	126
4.3.1.2 Seguridad física .....	126

4.3.1.3 Seguridad organizativa.....	126
4.4.1 Pruebas de Caja Negra .....	127
4.4.2 Prueba de Caja Blanca .....	130
CAPITULO V ANÁLISIS DE COSTO .....	133
5.1 INTRODUCCIÓN .....	133
5.2 MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE .....	133
5.2.1 Método de estimación de COCOMO .....	133
CAPITULO VI CONCLUSIONES Y RECOMENDACIONES.....	137
6.1 CONCLUSIONES .....	137
6.2 RECOMENDACIONES .....	138
BIBLIOGRAFIA .....	139
ANEXOS	
ANEXO A ARBOL DE PROBLEMAS	
ANEXO B ÁRBOL DE OBJETIVOS	
ANEXO C FORMULARIO DE EVALUCION DOCENTE- UPEA	
ANEXO D MANUAL DE USUARIO	
ANEXO E MANUAL TECNICO	
ANEXO F DOCUMENTOS DE AVAL	
ANEXO G ESTRUCTURA ORGANIZATIVA	

## ÍNDICE FIGURAS

### CAPITULO II MARCO TEORICO

FIGURA N° 2. 1 ACTIVIDAD DE SISTEMA DE INFORMACIÓN .....	15
FIGURA N° 2. 2 DEFINICIÓN DE SOFTWARE .....	16
FIGURA N° 2. 3 TIPOS PRINCIPALES DE SOFTWARE.....	17
FIGURA N° 2. 4 CONCEPTO DE EVALUACIÓN.....	18
FIGURA N° 2. 5 ARQUITECTURA EN DOS NIVELES .....	28
FIGURA N° 2. 6 ARQUITECTURA EN TRES NIVELES .....	29
FIGURA N° 2. 7 ARQUITECTURA CLÁSICA CLIENTE SERVIDOR .....	33
FIGURA N° 2. 8 ESTRUCTURA DE APLICACIÓN WEB.....	35
FIGURA N° 2. 9 SERVIDOR WEB .....	36
FIGURA N° 2. 10 DEFINICIÓN DE MVC.....	41
FIGURA N° 2. 11 DEFINICIÓN DE JQUERY.....	44
FIGURA N° 2. 12 LENGUAJE DE PROGRAMACIÓN JAVASCRIPT .....	45
FIGURA N° 2. 13 METODOLOGÍA DE DESARROLLO UWE.....	49
FIGURA N° 2. 14 DISEÑO DEL SISTEMA.....	51
FIGURA N° 2. 15 ESTEREOTIPOS DE DIAGRAMA DE CASO DE USO.....	53
FIGURA N° 2. 16 ACTOR.....	53
FIGURA N° 2. 17 CASO DE USO .....	53
FIGURA N° 2. 18 ASOCIACIÓN.....	54
FIGURA N° 2. 19 DEPENDENCIA O INSTANCIACIÓN .....	54
FIGURA N° 2. 20 GENERALIZACIÓN .....	54
FIGURA N° 2. 21 DIAGRAMA DE CASO DE USO EN LA METODOLOGÍA UWE.....	55
FIGURA N° 2. 22 DIAGRAMA DE CONTENIDO EN LA METODOLOGÍA UWE .....	56
FIGURA N° 2. 23 MODELO DE NAVEGACIÓN DE LA APLICACIÓN PORTAL MUSICAL .....	57
FIGURA N° 2. 24 MODELO DE ESTRUCTURA DEL PROCESO.....	58
FIGURA N° 2. 25 MODELO DE PRESENTACIÓN.....	61
FIGURA N° 2. 26 NORMA ISO 9000.....	63
FIGURA N° 2. 27 PUNTOS OBJETO.....	77

FIGURA N° 2. 28 DETALLES DE COEFICIENTES DE COCOMO LL.....	80
FIGURA N° 2. 29 ESTRUCTURA ISO 27000.....	86
CAPITULO III MARCO APLICATIVO	
FIGURA N° 3. 1 DISEÑO DEL SISTEMA LOGIN .....	93
FIGURA N° 3. 2 DISEÑO DEL SISTEMA.....	93
FIGURA N° 3. 3 CODIFICACIÓN DEL SOFTWARE .....	94
FIGURA N° 3. 4 PANEL DE INICIO DE SESIÓN .....	95
FIGURA N° 3. 5 PANEL DE KARDEX .....	95
FIGURA N° 3. 6 PANEL DE DOCENTE .....	96
FIGURA N° 3. 7 PANEL DE ESTUDIANTE.....	96
FIGURA N° 3. 8 LA INSTALACIÓN O FASE DE IMPLEMENTACIÓN .....	98
FIGURA N° 3. 9 MODELO DE CASO DE USO: GENERAL DEL SISTEMA .....	100
FIGURA N° 3. 10 DIAGRAMA DE CASO DE USO: KARDEX.....	101
FIGURA N° 3. 11 DIAGRAMA DE CASO DE USO: COMISIÓN EVALUADOR CONFIGURACIÓN ESTUDIANTE .....	102
FIGURA N° 3. 12 DIAGRAMA DE CASO DE USO: COMISIÓN EVALUADOR CONFIGURACIÓN DOCENTE .....	103
FIGURA N° 3. 13 DIAGRAMA DE CASO DE USO: DOCENTE .....	105
FIGURA N° 3. 14 DIAGRAMA DE CASO DE USO: ESTUDIANTE.....	106
FIGURA N° 3. 15 MODELO CONTENIDO.....	108
FIGURA N° 3. 16 MODELO DE NAVEGACIÓN: KARDEX.....	109
FIGURA N° 3. 17 MODELO DE NAVEGACIÓN: COMISIÓN EVALUADOR.....	110
FIGURA N° 3. 18 MODELO DE NAVEGACIÓN: DOCENTE.....	110
FIGURA N° 3. 19 MODELO DE NAVEGACIÓN: ESTUDIANTE .....	111
FIGURA N° 3. 20 MODELO DE ESTRUCTURA DE PROCESO DEL SISTEMA .....	111
FIGURA N° 3. 21 MODELO DE PRESENTACIÓN: PRINCIPAL LOGIN KARDEX.....	112
FIGURA N° 3. 22 MODELO DE PRESENTACIÓN: KARDEX.....	113
FIGURA N° 3. 23 MODELO DE PRESENTACIÓN: PRINCIPAL LOGIN COMISIÓN EVALUAR .....	113
FIGURA N° 3. 24 MODELO DE PRESENTACIÓN: COMISIÓN EVALUADOR .....	114
FIGURA N° 3. 25 MODELO DE PRESENTACIÓN: PRINCIPAL LOGIN DOCENTE....	114

FIGURA N° 3. 26 MODELO DE PRESENTACIÓN: DOCENTE.....	115
FIGURA N° 3. 27 MODELO DE PRESENTACIÓN: PRINCIPAL LOGIN ESTUDIANTE .....	115
FIGURA N° 3. 28 MODELO DE PRESENTACIÓN: ESTUDIANTE.....	116
CAPITULO IV CALIDAD Y SEGURIDAD	
FIGURA N° 4. 1 PROCESO DE LA CAJA NEGRA LOGIN .....	128
FIGURA N° 4. 2 PROCESO DE LA CAJA NEGRA KARDEX .....	128
FIGURA N° 4. 3 PROCESO DE LA CAJA NEGRA DOCENTE.....	129
FIGURA N° 4. 4 PROCESO DE LA CAJA NEGRA ESTUDIANTE.....	129
FIGURA N° 4. 5 PRUEBAS DE CAJA BLANCA PARA EL REGISTRO DE USUARIO EN EL SISTEMA.....	131
FIGURA N° 4. 6 PRUEBAS DE CAJA BLANCA PARA EL NUEVO REGISTRO .....	131
FIGURA N° 4. 7 PRUEBAS DE CAJA BLANCA PARA LA MODIFICACIÓN EL REGISTRO.....	132

## ÍNDICE TABLAS

### CAPITULO I MARCO PRELIMINAR

TABLA N° 1. 1 CALIDAD DE PRODUCTO – ISO 9126 .....	10
--	----

### CAPITULO II MARCO TEORICO

TABLA N° 2. 1 COMPARACIÓN DE ARQUITECTURAS .....	30
TABLA N° 2. 2 DESCRIPCIÓN DE ESTEREOTIPOS.....	57
TABLA N° 2. 3 DESCRIPCIÓN DE ESTEREOTIPOS MODELO DE PRESENTACIÓN ..	60
TABLA N° 2. 4 CÁLCULO DE PUNTO FUSIÓN NO AJUSTADO .....	69
TABLA N° 2. 5 CÁLCULO DE PUNTO FUSIÓN AJUSTADO .....	70
TABLA N° 2. 6 VALORES DE AJUSTE DE COMPLEJIDAD .....	70
TABLA N° 2. 7 MÉTRICA DE MADUREZ.....	71
TABLA N° 2. 8 MÉTRICA DE ENTENDIBILIDAD.....	73
TABLA N° 2. 9 AJUSTE DE PREGUNTAS .....	73
TABLA N° 2. 10 MÉTRICA DE CAMBIABILIDAD .....	74
TABLA N° 2. 11 CONSTANTES DE COSTES.....	80
TABLA N° 2. 12 COEFICIENTES DE COCOMO.....	81
TABLA N° 2. 13 ATRIBUTOS DE COSTE .....	83

### CAPITULO III MARCO APLICATIVO

TABLA N° 3. 1 CATEGORÍA DE LAS FUNCIONES .....	90
TABLA N° 3. 2 REQUERIMIENTOS FUNCIONALES .....	91
TABLA N° 3. 3 REQUERIMIENTOS NO FUNCIONALES .....	92
TABLA N° 3. 4 DIAGRAMA DE CASO DE USO: KARDEX.....	101
TABLA N° 3. 5 DIAGRAMA DE CASO DE USO: COMISIÓN EVALUADOR CONFIGURACIÓN ESTUDIANTE .....	102
TABLA N° 3. 6 DIAGRAMA DE CASO DE USO: COMISIÓN EVALUADOR.....	104
TABLA N° 3. 7 DIAGRAMA DE CASO DE USO: DOCENTE.....	105
TABLA N° 3. 8 DIAGRAMA DE CASO DE USO: ESTUDIANTE .....	106

### CAPITULO IV CALIDAD Y SEGURIDAD

TABLA N° 4. 1 NÚMERO DE ENTRADAS DE USUARIO.....	118
TABLA N° 4. 2 NÚMERO DE SALIDAS DE USUARIO .....	118

TABLA N° 4. 3 NÚMERO DE PETICIONES DE USUARIO .....	119
TABLA N° 4. 4 NÚMERO DE ARCHIVOS.....	119
TABLA N° 4. 5 NUMERO DE INTERFACES EXTERNAS .....	119
TABLA N° 4. 6 CALCULO DE PUNTO DE FUNCIÓN .....	120
TABLA N° 4. 7 VALORES DE AJUSTE DE COMPLEJIDAD .....	120
TABLA N° 4. 8 AJUSTE DE PREGUNTAS .....	124
CAPITULO V ANALISI DE COSTO	
TABLA N° 5. 1 APLICACIÓN DEL MODELO INTERMEDIO.....	134
TABLA N° 5. 2 ECUACIONES DEL MODELO COCOMO .....	134
TABLA N° 5. 3 CALCULO DE ATRIBUTOS FAE .....	135

## **CAPITULO I MARCO PRELIMINAR**

### **1.1. INTRODUCCIÓN**

En la actualidad el uso tecnologías de información y comunicación dentro de las diferentes instituciones públicas y privadas es muy importante puesto que permite optimizar procesos administrativos y las actualizaciones de datos son reflejadas en tiempo real. Esto se puede ver plasmado en el funcionamiento de sistemas información que apoyan en el registro, control y seguimiento de la información que maneja una institución, el uso de aplicaciones de esta naturaleza más, que incluso pueden llegar a determinar el triunfo o el fracaso de la institución. La creación y desarrollo de aplicaciones responde de manera muy efectiva los requerimientos de una institución y se constituye en ventajas significativas para la agilización de los procesos académicos y administrativos

Existen dos enfoques básicos sobre la evaluación docente: la formativa y la sumativa. La evaluación formativa se orienta al desarrollo personal y profesional, los resultado que emiten ofrecen información en aquellos aspectos en los que el docente debe mejorar o corregir y también en los que actúa en forma adecuada. La evaluación sumativa permite obtener resultados que suman un puntaje en la historia académica del docente con fines de mejora salarial o ascenso de categoría. Por tanto, la evaluación formativa es mucho más útil, porque implica al docente en la mejora y correcta marcha del proceso enseñanza aprendizaje del estudiante, superando deficiencias que les hace sentir mejor profesionalmente. Según (Molero & Ruiz, 2006).

El presente proyecto tiene por objeto desarrollar un sistema de información de evaluación docente para la carrera Comercio Internacional que proporcione datos oportunos y correctos para la toma de decisiones de la Dirección de Carrera y autoridades de la Universidad Pública de El Alto.

## **1.2. ANTECEDENTES**

### **1.2.1. Antecedentes Institucionales**

La carrera de Comercio Internacional de la Universidad Pública de El Alto fue creada sobre la base de la Resolución Vicerrectoral V.R. N° 15/2011 de fecha 27/04/2011, que el HCU dispone aprobar la creación e implementación de funcionamiento de la carrera bajo la Resolución N° 43/2011 de fecha el 27/04/2011, la misma ha sido ratificada en la XVI Conferencia Nacional Extraordinaria de Universidades llevada a cabo en la ciudad de Santa Cruz de la Sierra mediante Resolución Nro. 06/2012 de fecha 29 de febrero de 2012.

#### **❖ Misión**

Formar profesionales en Comercio Internacional con alto grado de calidad y conocimiento científico, con la capacidad de promover procesos de vinculación e intercambio comercial a nivel internacional en los sectores: productivo, social y gubernamental, diseñando políticas, estrategias y programas de comercialización y administración de operaciones de negocios internacionales, detectando oportunidades de comercio en mercados externos, basados en el desarrollo de proyectos sociales y económicos con incidencia nacional e internacional que aporten decididamente al mejoramiento de las condiciones de vida de la población de El Alto y del País.

#### **❖ Visión**

Ser una carrera líder en la formación de profesionales, al servicio del progreso de la ciudad de El Alto y del país, enfocada al desarrollo de sus actividades académicas, científicas, productivas y tecnológicas en el área del Comercio Internacional, y ser reconocida por la sociedad, alcanzando estándares de calidad en investigación en el marco del “Vivir Bien”, posicionando a la Carrera a nivel nacional e internacional con reconocido prestigio académico y liderazgo.

#### **❖ Filosofía, principios y valores**

La cultura y los valores fundamentales de la Carrera de Comercio Internacional son:  
La defensa de la primacía de la persona como valor principal.

- ❖ El compromiso y la defensa de los principios que inspiran la naturaleza y fines de la Universidad pública.
- ❖ La búsqueda de la excelencia en la docencia, en el proceso de formación, en la investigación, en la prestación de servicios y en la gestión institucional.
- ❖ El liderazgo en el conocimiento, contribuyendo a su creación, transmisión, actualización y aplicación.
- ❖ La sensibilidad y el compromiso con el medio ambiente, abogando por un desarrollo sostenible.
- ❖ La defensa y la promoción de los valores sociales e individuales que le son propios, tales como la libertad, fraternidad, justicia social, el pluralismo, el respeto de las ideas y el espíritu crítico, así como la búsqueda de la verdad, la convivencia pacífica y la defensa de los derechos humanos.
- ❖ Igualmente, la atención a la eliminación de la desigualdad social y la discriminación por razón de género y etnia
- ❖ El decidido apoyo a la causa de la paz social.
- ❖ La responsabilidad compartida por los distintos sectores de la comunidad universitaria en el gobierno, gestión y control del funcionamiento de la Universidad.
- ❖ La participación de los estudiantes en la vida universitaria, respetando su autonomía.
- ❖ La participación activa del estudiante en el proceso enseñanza-aprendizaje.
- ❖ El trabajo en equipo, como facilitador del proceso formativo, investigador y de gestión.
- ❖ La innovación y asunción de riesgos como medio de adaptación al cambio permanente.
- ❖ El carácter abierto y universal y el compromiso con la comunidad alteña y con su desarrollo humano, cultural, tecnológico y económico.

### **1.2.2. Antecedentes Internacionales**

- ❖ Según (Pablo A, 2005) “Automatización del proceso de evaluación docente para la escuela de formación de profesores de enseñanza media”; El sistema

fue realizado con el fin de implementar el programa informático que permita realizar el procesamiento de la información obtenida a través de los instrumentos diseñados para la evaluación del personal docente y administrativo de la Escuela de Formación de Profesores de Enseñanza Media, Universidad de San Carlos de Guatemala.

- ❖ Según (Eduardo P, 2011) “Sistema de evaluación docente para la carrera de ingeniería de sistemas computacionales”; nace de la necesidad imperiosa de optimizar la forma de evaluar al personal docente de la Carrera de Ingeniería en Sistemas Computacionales, como una forma de mejorar la calidad del servicio educativo que la misma ofrece, además de establecerse como un paso inicial para reunir los requerimientos necesarios a fin de obtener la acreditación universitaria, para tomar ciertas medidas en miras a obtener la acreditación universitaria, la misma que certifica la calidad educativa y el nivel de servicio que ofrece la institución.

### **1.2.3 Antecedentes Nacionales**

- ❖ Según (Félix J, 2007) “Análisis de la carga horaria de los recursos humanos de docentes de la programación presupuestaria de la UMSA”; tiene por objetivo principal presentar una guía memoria de los Recursos Humanos de docentes, dar a conocer la determinación porcentual de una muestra que identifica a las restantes carreras y elaborar un software vinculado a la carga horaria de docentes para las diferentes gestiones, que esto sirva como una guía al departamento de planificación y coordinación, para verificar si es eficiente o no la distribución de la carga horaria, Universidad Mayor de San Andrés, La Paz-Bolivia.
- ❖ Según (Wendy P, 2017) El “Sistema de información administrativa y académica de programas de postgrado con la metodología ICONIX”; tiene por objetivo desarrollar un sistema informático en base a tecnología Web y la metodología ICONIX que facilite la organización y gestión de información administrativa y académica del área de posgrado del IIISP, Universidad Mayor de San Andrés, La Paz-Bolivia.

### **1.3. PLANTEAMIENTO DE PROBLEMA**

#### **1.3.1. Problema Principal**

Luego de realizar el análisis de los problemas, se ha diseñado el árbol de problemas (ver anexo A). Permitió identificar que existe la dificultad en el manejo de documentos y el proceso de evaluación docente que realiza periódicamente la carrera Comercio Internacional presenta demoras en la publicación de los resultados y reportes requeridos para la toma de decisiones debido a que el proceso se realiza de forma manual con apoyo de hojas electrónicas.

La falta de estrategia de comunicación interna es palpable desde el punto de vista de los estudiantes, de las autoridades, e incluso de los mismos docentes, contar con información sobre la permanencia de los docentes.

El planteamiento de la evaluación de la actividad docente universitaria como problema a resolver significa adentrarse en un contexto tremendamente complejo y arduo que requiere ser afrontado desde diversas perspectivas, sin olvidar que al evaluar se pretende encontrar una forma de medir una variable altamente abstracta y compleja, llamada “evaluación permanente del docente” como se denomina en la UPEA.

#### **1.3.2. Problemas Secundarios**

- ❖ El llenado de los datos personales de los estudiantes es realizado de forma manual en cada evaluación, lo que causa incomodidad en las aulas restando horas en clases y otras actividades al momento de la evaluación, debido a la falta de un sistema de información automatizado.
- ❖ Falta de información sistematizada de producción intelectual y vida universitaria de docentes activos.
- ❖ Poca organización por parte de la comisión de evaluación para la obtención de un resultado, debido al desfase del tiempo.
- ❖ Ausencia de base de datos actualizados de los docentes y estudiantes que afecta en la coordinación a los componentes de la comisión de evaluación docente.

- ❖ Falta de herramientas para la sistematización de los formularios de evaluación.
- ❖ No existe una correcta calificación de la documentación presentada de docentes debido a la falta de personal.

Por el cual surge la siguiente interrogante:

¿De qué manera el sistema de información de evaluación de docentes coadyuvaría en el proceso de evaluación docente de la carrera Comercio Internacional de la UPEA?

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Luego del análisis de problemas y su evaluación causa efecto se ha diseñado el árbol de objetivos (ver anexo B).

Desarrollar un sistema de información de evaluación docente, usando nuevas tecnologías de programación que permita optimizar el proceso de evaluación docente de la carrera Comercio Internacional de la Universidad Pública de El Alto en cada periodo académico.

### **1.4.2 Objetivo Específicos**

- ❖ Realizar la ingeniería de requerimientos funcionales y no funcionales del proceso de evaluación.
- ❖ Modelar la base datos acorde a los requerimientos realizados.
- ❖ Desarrollar el sistema con un interfaz amigable para el usuario y responsivo para los distintos dispositivos.
- ❖ Efectuar la evaluación de métricas de calidad software.
- ❖ Elaborar la documentación de uso y administración del sistema propuesto, que sirva de referencia a los usuarios del sistema.

## **1.5. JUSTIFICACIÓN**

### **1.5.1. Justificación Técnica**

El proyecto se justifica técnicamente porque para su desarrollo se utilizará tecnologías de última generación asimismo la carrera de Comercio Internacional dispone de equipos de computación en sus diferentes unidades administrativas.

Con la evolución constante de tecnologías de información, toda institución tiene la necesidad de mejorar sus servicios a través de la explotación de todos sus recursos tecnológicos. En este caso, al ser la carrera Comercio Internacional dependiente de la Universidad Pública de El Alto tiene la oportunidad de hacer uso sistemas informáticos e infraestructura tecnológica proporcionadas por la misma carrera.

Por esta razón la carrera de Comercio Internacional dispone de la tecnología adecuada para soportar este tipo de sistema.

Requerimientos mínimos (Servidor el cual aloja el sistema Web, instalación de red, conexión en internet, PC's mínimo sistema operativo Windows, navegador Web).

### **1.5.2. Justificación Económica**

La carrera será beneficiada con el desarrollo del nuevo sistema de información docente, además reducir horas de trabajo del personal de comisión de evaluación docente en el registro de todas las evaluaciones de estudiantes a los docentes que tiene la carrera. Esto ayudara a agilizar los procesos manuales, tener un mejor control y seguimiento del sistema de evaluación docente, lo que permitirá obtener información rápida, oportuna, confiable y fácil de comprender beneficiando a la carrera y a la comisión de evaluación docente.

El proyecto será desarrollado con herramientas de software libre<sup>1</sup> por lo cual este proyecto reducirá el costo de desarrollo.

---

<sup>1</sup> Software libre es la denominación del Software que respeta la Libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.

### **1.5.3. Justificación Social**

Con el desarrollo del proyecto se beneficiara a autoridades, estudiantes y docentes, puesto que agilizará el proceso de evaluación docente, como herramienta útil para cada uno de los usuarios citados.

La universidad es una institución de servicio público, debe estar orientada al beneficio social, y al realizar este tipo de evaluaciones orientadas a mejorar el nivel de servicio y la calidad educativa que se ofrece, los mayores beneficiados son la población estudiantil que se prepara en ella, mientras que a mediano y largo plazo el beneficiario de todo este proceso es el desarrollo económico y social del país en su conjunto.

## **1.6. HERRAMIENTAS.**

### **1.6.1. Herramientas para desarrollo Web.**

#### **❖ MODELO VISTA CONTROLADOR (MVC)**

Es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones Web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla. Además, que es un patrón orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no sigan este paradigma.

- a) Modelo: Representa la lógica de negocios. Es el encargado del acceso de forma directa a los datos actuando como “intermediario” con la base de datos.
- b) Vista: Es la encargada de mostrar la información al usuario de forma gráfica y “humanamente legible”.
- c) Controlador: Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que lo presente al usuario, de forma “humanamente legible”. Según (Álvarez, 2001).

- ❖ **XAMPP:** Es un paquete de software libre que consiste principalmente en el sistema de gestión de bases de datos MySQL. El servidor web Apache y los intérpretes para lenguaje de script PHP.
- ❖ **JAVASCRIPT:** Es un lenguaje de programación orientado a objetos, el cual se utiliza para acceder a objetos en aplicaciones, integrándolo en un navegador web permitiendo el desarrollo de interfaces de usuario. Según (Álvarez, 2001)
- ❖ **CSS3:** Son hojas de estilos creados para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML. Según (Álvarez, 2001)
- ❖ **PHP:** Es un lenguaje de código abierto muy adecuado para el desarrollo web que puede ser utilizado en HTML. Realiza proceso y manda una respuesta al navegador (cliente). El código es interpretado por un servidor web con el módulo procesador de PHP que genera la página web resultante. Según (Álvarez, 2001).
- ❖ **BOOTSTRAP:** Es un conjunto de herramientas de código abierto, para diseño de sitios y aplicaciones web, porque contiene plantillas de diseño de tipografías, formularios, botones, menús de navegación, cuadros y otros elementos de diseño basado en HTML y CSS.
- ❖ **MARIADB:** MariaDB es un sistema de base de datos que proviene de MySQL, pero con licencia GPL, desarrollado por Michael Widenius, fundador de MySQL y la comunidad de desarrolladores de software libre. Según (Sierra, Karim, 2018).

## 1.7. METODOLOGÍA

### ❖ Metodología UWE.

En el presente proyecto se empleará la Metodología UWE el cual es una metodología para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño. Este proceso, iterativo e incremental. Incluye flujos de trabajo y puntos de control, y sus frases coinciden con las propuestas en el proceso Unificado de Modelado. Según (Blanco, 2015).

❖ **Método de ciclo de vida.**

La metodología UWE emplea el proceso iterativo incremental, el cual consta de una secuencia de iteraciones, cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Según (Blanco, 2015).

❖ **Métricas de calidad.**

Para el desarrollo del proyecto se utiliza la ISO 9126 con el cual se hallaran la implementación para poder solucionarlos inmediatamente.

**Características:**

Capacidad del producto software para bajo condiciones especificadas.

TABLA Nº 1. 1 Calidad de Producto – ISO 9126

FUNCIONALIDAD	Proporcionar funciones que satisfacen necesidades declaradas e implícitas cuando se usa.
FIABILIDAD	Mantener un nivel especificado de prestaciones cuando se usa.
USABILIDAD	Ser entendido, aprendido, usado y ser atractivo para el usuario cuando se usa.
EFICIENCIA	proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados
MANTENIBILIDAD	Ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación a cambios en el entorno, requisitos o especificaciones funcionales
PORTABILIDAD	ser transferido de un entorno a otro

Según (Pereira, 2008).

### ❖ **Metodología de estimación de costos.**

Para estimar el costo que tendrá el sistema de información de evaluación docente se utilizara la metodología COCOMO, el cual procesa las líneas de código, las horas trabajadas, el nivel de conocimiento que tiene el desarrollador y la complejidad que tiene el sistema a ser desarrollado para la empresa. Según (Pereira, 2008).

## **1.8. LÍMITES Y ALCANCES**

### **1.8.1. Limites**

El sistema de información se limita al proceso de evaluación docente detallado en el estatuto orgánico de la Universidad Pública de El Alto.

El sistema planteado se limita a los requerimientos funcionales específicos del proceso de evaluación a docentes de la carrera Comercio Internacional.

### **1.8.2. Alcances**

Los alcances del presente proyecto abarcan los siguientes módulos.

- a) Módulo de acceso y sincronización del sistema. Se encarga del acceso de usuarios de manera individual tanto estudiante, docente y comisión evaluador.
- b) Módulo de presentación de resultados individual de cada materia. Se encargara de mostrar los resultados por pantalla vía web o por medio de una aplicación Android o otros dispositivos a usuarios (docentes, comisión evaluador y estudiantes) de manera personalizada por separado.
- c) Módulo de evaluación. Es el encargado de la configuración de la forma de evaluación individual basada en el formulario vice - rectorado en parámetros necesarios por materia.
- d) Módulo de presentación de resultados generales. Es el que muestra resultados de manera general debidamente organizada.
- e) Módulo de evaluación. En este módulo presenta el formulario de valuación a cada materia con su respectiva ponderación de puntuaciones.

## **1.9. APORTES**

El siguiente aporte del presente proyecto, para la carrera Comercio Internacional será la automatización de un sistema de evaluación a docentes, que desde antes hasta ahora se procedían de forma manual. Se dejará una estructura de procesos administrativos para el manejo de los programas de evaluación en base al estatuto de la UPEA.

Se creará un entorno de trabajo sistematizado que no sólo involucrará al personal administrativo, también permitirá el acceso a estudiantes, docentes y comisión evaluador.

Por tanto, la forma en la que se procese la información ya estará definida por el sistema y se mantendrán registros que servirán de antecedentes. De esta manera, sin importar que se cambie la dirección o personal de la carrera Comercio Internacional, se aportará a que el flujo de trabajo continúe con el paso del tiempo.

## CAPITULO II MARCO TEÓRICO

En este capítulo se expone los fundamentos teóricos que son necesarios para el Desarrollo del Sistema de evaluación de docentes en la carrera comercio internacional de la Universidad Pública de El Alto (UPEA).

### 2.1 DEFINICIÓN DEL SISTEMA

Para el presente proyecto se consultan a diferentes autores sobre la definición de sistemas.

“Sistema en general se define componentes conectados e interactivos, que tienen un propósito y una unidad total. Sistema de procedimiento de información es un sistema que transforma datos brutos en información organizada, significada y útil”. Según (Joyanes 1996).

Sistema basada en computadora, es un conjunto o disposición de elementos que están organizados alrededor de un objetivo predefinido, procesando información. Según (Pressman R. S., 2010).

Para conseguir el objetivo, un sistema basada en computadoras hace uso de varios elementos:

- ❖ **Software**, es suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenece a un sistema de cómputo.
- ❖ **Hardware**, son dispositivos electrónicos que proporcionan una función externa, del mundo real.
- ❖ **Personas**, usuarios y operadores del hardware y software.
- ❖ **Documentación manual**, formularios y otra información descriptiva que plasma el empleo y/o funcionamientos del sistema.
- ❖ **Procedimientos**, los pasos que definen el empleo específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

De los autores mencionados se concluye que, un sistema es un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares.

Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación.

## **2.2 SISTEMA DE INFORMACIÓN**

Según (James, 1992). Un sistema de información es un conjunto de componentes que interaccionan si para lograr un objetivos común. Los sistemas de información se clasifican en tres categorías:

- ❖ Sistema de procedimiento de transacciones.
- ❖ Sistema de información administrativa.
- ❖ Los sistemas para el soporte de decisiones.

Un sistema de información es el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas hasta sistemas de cómputo que generan reportes periódicos para varios usuarios. Según (Peralta, 2007).

Por lo que se define a un sistema de información como un conjunto de funciones o procedimiento interrelacionados que forman un todo, es decir, obtiene, procesa, almacena y distribuye informaciones para apoyar la toma de decisiones.

En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización.

### **2.2.1 Características de un sistema de información**

Un sistema de información se caracteriza principalmente por la eficiencia que procesa los datos en relación al área de acción. Los sistemas de información se alimentan de los procesos y herramientas de estadística, probabilidad, inteligencia de negocio, producción, marketing, entre otros para llegar a la mejor solución.

Un sistema de información se destaca por su diseño, facilidad de uso, flexibilidad, mantenimiento automático de los registros, apoyo en toma de decisiones críticas y mantener el anonimato en informaciones no relevantes. Según (Chen, 2008)

## 2.2.2 Componentes de un sistema de información

Los componentes que forman un sistema de comunicación son:

- ❖ La entrada: Por donde se alimentan los datos,
- ❖ El proceso: Uso de las herramientas de las áreas contempladas para relacionar, resumir o concluir,
- ❖ La salida: Refleja la producción de la información, y
- ❖ La retroalimentación: Los resultados obtenidos son ingresados y procesados nuevamente.

## 2.2.3 Actividad de sistema de información

Como acabamos de definirlo en la página anterior todo sistema de información es un conjunto de partes que se relacionan entre sí para lograr un objetivo que contribuirá a la toma de decisiones y esto es posible gracias a las diversas actividades que se llevan a cabo, lo cual estimula que la organización desarrolle sus procesos de una mejor manera, obteniendo así una eficiente labor. Los Sistemas de Información que realizan las diversas actividades principales haciendo uso de la computadora, lo cual permiten un mejor desempeño de los sistemas de información en las organizaciones públicas y/o privadas. Según (Resan, 2016).

FIGURA N° 2. 1 Actividad de sistema de información



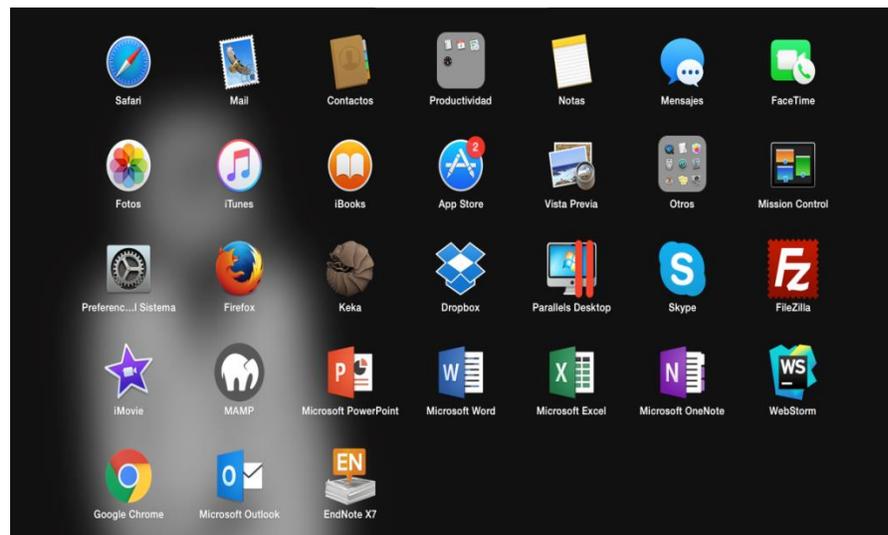
Fuente: (Resan, 2016).

## 2.3 DEFINICIÓN DE SOFTWARE

El término software es un vocablo inglés, que ha sido tomado por otros idiomas como el español para hacer referencia a determinados aplicativos en la informática. Este término designa al equipo lógico de una computadora, opuesto a los aspectos físicos de la misma.

El software está compuesto por un conjunto de programas que son diseñados para cumplir una determinada función dentro de un sistema, ya sean estos realizados por parte de los usuarios o por las mismas corporaciones dedicadas a la informática. Según (Resan, 2016).

FIGURA Nº 2. 2 Definición de software



Fuente: (Resan, 2016).

El concepto de software, como se mencionó, compone la parte lógica de un sistema de software, permitiéndole el funcionamiento. Esto quiere decir entonces que no solo los programas son y forman un software, sino que la información del usuario y los datos procesados integran el software, ya que forma parte de él todo componente intangible y no físico.

### 2.3.1 Tipos principales de software

Un software de aplicación está diseñado para la realización de una o más tareas a la vez.

FIGURA Nº 2. 3 Tipos principales de software



Fuente: (Estela, software, 2019)

- ❖ **Software de sistema:** Este grupo clasifica a los programas que dan al usuario la capacidad de relacionarse con el sistema, para entonces ejercer control por sobre el hardware. El software de sistema también se ofrece como soporte para otros programas. Ejemplos: sistemas operativos, servidores, etcétera.
- ❖ **Software de programación:** Programas directamente diseñados como herramientas que le permiten a un programador el desarrollo de programas informáticos. Influyen en su utilización diferentes técnicas utilizadas y lenguaje de programación específico. Ejemplos: compiladores, editores multimedia, etcétera.
- ❖ **Software de aplicación:** Programas diseñados para la realización de una o más tareas específicas a la vez, pudiendo ser automáticos o asistidos. Ejemplos: videojuegos, aplicaciones ofimáticas, etcétera. Según (Estela, software, 2019).

## 2.4 EVALUACION

### 2.4.1 Concepto de evaluación

La evaluación se conoce “como un proceso sistemático de recogida de datos, incorporado al sistema general de actuación educativa, que permite tener información valida y fiable para formar juicios de valor acerca de una situación.

Estos juicios a su vez utilizara en la toma de decisiones consecuente con objeto de mejorar la actividad educativa valorada.” Según (Casanova, 1923).

*FIGURA Nº 2. 4 Concepto de evaluación*



Fuente (Casanova, 1923).

“La evaluación es un proceso sistemático de recogida de información, implica un juicio de valor, orientado a la toma de decisiones.” Según (Tejada, 1999).

#### **2.4.2 Dimensiones de la evaluación**

La evaluación presenta múltiples dimensiones y variables que se deben tomar en cuenta al momento de evaluar. Por eso para llevar a cabo una evaluación, previamente se debe planificar de manera cuidadosa la acción en relación con la realidad que a evaluar. Por todo ello se hace imprescindible tomar en cuenta las dimensiones de la evaluación. Según (Luna, 2008).

#### **2.4.3 Evaluación de desempeño docente por estudiantes universitario**

Si bien es cierto que la evaluación de la docencia, por parte de los estudiantes y mediante el uso de instrumentos de evaluación, es un tema muy estudiado en todas las universidades del mundo. Está claro que la evaluación de la docencia en nuestras universidades se hace de algún modo, pues de otro modo carecería de sentido la premiación a docentes o el cambio de categoría docente, o la nueva contratación por el cese de actividades de algún docente. Lo que no queda claro en

algún documento oficial es el instrumento de evaluación y la metodología de aplicación, así como el uso que tiene la información. Según (Guzmán, 2010)

¿Por qué solo por los estudiantes?

En primer lugar, porque es la técnica de evaluación más usada por las universidades en el mundo y, al mismo tiempo, la más estudiada, existen más de dos mil trabajos publicados en relación a los cuestionarios de opinión docente. Según (Luna, 2008). Aunque mucho se critique la centralidad que tienen los cuestionarios, hay muchas experiencias que muestran que la construcción de una adecuada escala para las preguntas y otra para la respuesta proporciona resultados confiables. Según (González, 2010).

Para la evaluación del desempeño docente existen muchas alternativas metodológicas, entre las más usadas están los cuestionarios de opinión, la entrevista cualitativa, los grupos de discusión con estudiantes, entre otras.

#### **2.4.4 Evaluación del desempeño docente**

La evaluación de desempeño docente es un proceso que tiene por objetivo dar un juicio de valor sobre las cualidades del docente, o sea, su grado de excelencia en el comportamiento de rol de ocupante del cargo. A su vez el docente evaluado, conocerá cómo es su desempeño, cuáles son sus fortalezas y debilidades, qué proyección tiene en la carrera, institución y a nivel individual. Otro de los términos que está relacionado con este juicio de valor sobre el desempeño docente es el rendimiento académico por regularidad.

Las conclusiones a las que han arribado autores citados consisten en reconocer que evaluar es un proceso complejo y es necesario contar con la opinión de los estudiantes porque ellos se constituyen en informantes y destinatarios de la tarea docente, a la hora de responder los formularios o cuestionarios de evaluación.

La utilización de indicadores de evaluación y medición del desempeño docente es un factor importante de articulación entre los postulados de la transformación educativa, y la piedra angular del marco teórico.

Cuando hablamos del desempeño docente nos involucramos con conceptos de mejora de la enseñanza, con dosis de calidad en educación, en todo caso podríamos tocar un punto de la Implementación de un sistema de evaluación de desempeño a partir del concepto de mejoras prácticas laborales de docencia y los dueños de procesos tendrían que obtener logros de excelencia, a fin de estructurar la filosofía de la evaluación del desempeño, con procesos de mejora de la calidad.

No hay un sistema perfecto de evaluación del desempeño docente, pero se necesita herramientas para asegurar desempeños productivos del docente con calidad, sabiendo que el sistema que se crea académicamente, siempre se verá afectado por la presencia del factor humano es decir de una manera subjetiva. Siempre se generarán problemas de evaluación, pero no es posible para las instituciones obviar un sistema de evaluación del desempeño basado en estrategias, metas.

Por esto, la evaluación del desempeño docente se plantea, como razón de ser, proponer la de mejorar, sobre todo, a partir de un conjunto de datos e informes fundamentados en procesos de trabajos de investigación sobre el tema, con la máxima intervención posible de todos los involucrados, emitir un juicio valorativo sobre la amplitud, evolución, y complejidad de las variables que conforman el desempeño docente objeto del estudio.

#### **2.4.5 Modelos de evaluación de desempeño docente**

Según (Valdés, V., 2000, p. 6) existe la siguiente clasificación:

- ❖ Modelo centrado en el perfil ideal del Docente: Este modelo consiste en realizar el proceso de evaluación del desempeño docente, de acuerdo a su grado de concordancia según un perfil ideal del docente previamente elaborado.

- ❖ Modelo centrado en los resultados obtenidos: Este modelo del proceso de evaluación del desempeño docente, se fundamenta en la evaluación de los profesores mediante la comprobación del cumplimiento de los objetivos de aprendizaje alcanzados por los alumnos.
- ❖ Modelo centrado en el comportamiento del docente en el aula: Este modelo propone que la evaluación de la eficacia del desempeño docente, se debe realizar describiendo e identificando aquellos indicadores de la actividad del profesor, que se consideren relacionados directamente con los logros de los alumnos.
- ❖ Modelo de la práctica reflexiva: Este modelo de evaluación del desempeño docente, propone el desarrollo de un proceso evaluativo, con la finalidad de mejorar al personal docentes de modo integral, y para medir y controlar por motivos de promoción o despidos.

#### **2.4.5.1 Instrumento de evaluación de desempeño docente por estudiantes**

La evaluación del desempeño docente debe estar dirigida a las dimensiones del mismo, que tienen que ver con los roles propios de los educadores, entre las funciones de las dimensiones tenemos los indicadores para ver sus cualidades en ejercicio profesional.

Hay una generalización en educación superior, en donde los docentes son determinante en la mejora de la enseñanza. Por lo cual existe la necesidad de desarrollar sistemas de evaluación del desempeño docente que contribuyan a su profesionalización y, con ello, a la mejora de la formación profesional. Según La evaluación docente tiene valoraciones de puntajes de opinión de los estudiantes, es la estrategia de evaluación más utilizada en las facultades, universidades. Según (Jiménez, 2011).

En este contexto el uso generalizada de los cuestionarios, tiene cuestionantes por parte de los docentes y que son obstáculos fundamentales para validar su empleo; los docentes manifiestan una creciente incomodidad por su uso, en la toma de

decisiones administrativas; y se sobreestima o minimiza su utilidad. Según (Luna, 2008).

Actualmente, la evaluación de la docencia con base en los cuestionarios aplicado a los estudiantes, es reconocida como la estrategia de evaluación con mayor conocimiento en información y practico de respuesta en el contexto universitario. Por ello, se considera que los puntajes son útiles para los docentes, los estudiantes dentro del aula.

Entre los aspectos que los alumnos pueden evaluar, se detalla los siguientes: según Zambrano de conocimiento de la materia; claridad de las explicaciones; Responsabilidad al Trabajo; Capacidad Pedagógica; Estructuración de la Clase; Material Pedagógico Utilizado; Metodología de Enseñanza Adaptada; Motivación de los Alumnos en el Proceso de Enseñanza- Aprendizaje; Organización Pedagógica; Sistema de Evaluación adoptado en la Materia. En estos aspectos la opinión del alumno es indispensable. Según (Colima, 2012, p.15).

## **2.5 WEB WWW (World Wide Web)**

Una web es aquella que consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil como un smartphone o un monitor de computadora.

El formato de esta información es HTML y desde una web se puede entrar a otra con enlaces de hipertexto. Los recursos para la presentación de la información de una web pueden ser scripts, imágenes digitales, hojas en forma de cascada, entre otros. Para que una web pueda estar en línea y ser vista por cualquier persona, requiere de un hospedaje, cosa que puede hacerse en un servidor remoto o local. La composición de una página web consiste básicamente en información que puede presentarse con módulos multimedia y texto, lo cual se complementa con

hiperenlaces para conducir a otras secciones de dicha web o a otras webs. Según (Merino, 2013).

### **2.5.1 Ingeniería Web**

Es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

Se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

Es una disciplina que está relacionada con principios científicos de ingeniería y de gestión, con enfoques sistemáticos y disciplinados del éxito del desarrollo, empleo y mantenimiento de sistemas y aplicaciones basados en Web de alta calidad. Según (Pressman, 2005, p.317).

Por tanto podríamos decir que la ingeniería web son conocimientos científicos, métodos y metodologías de diseño y programación, sistemática, disciplinados y cuantificables al desarrollo eficiente aplicados y orientados a la web.

También es un área que abarca procesos, técnicas y modelos orientados a los entornos Web. Consiste en la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones web de alta calidad.

#### **2.5.1.1 Aplicaciones de la Web**

El término aplicación Web ha ido evolucionando desde un pequeño sitio Web a una robusta y entera aplicación. Ahora ya no es raro entender por una aplicación Web a cientos de usuarios simultáneos, distribuidos alrededor del mundo, todos conectados

a la vez si se requiere. Este término varía de acuerdo a las personas, algunos creen que una aplicación Web es cualquier página, otros consideran cualquier sistema que use un servidor Web. Aquí, una aplicación Web, será un servidor Web en el cual los usuarios introducen datos que afectan la condición del negocio. La diferencia entre una aplicación Web y un sitio Web radica en su uso. Una aplicación Web implementa la lógica del negocio, y usa cambios de estados del negocio. Así, las aplicaciones Web son sistemas de información donde una gran cantidad de datos volátiles, altamente estructurados, son consultados, procesados y actualizados mediante navegadores. Según (López, 2005, p.5).

Es un sistema de información donde una gran cantidad de datos volátiles, altamente estructurados, van a ser consultados, procesados y analizados mediante navegadores. Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuarios.

Por tanto podríamos decir que Una aplicación web es una herramienta de software que se accede y utiliza a través de un navegador web. Es una nueva estrategia de programación.

#### **2.5.1.2 Tipos de página Web**

El desarrollo de diferentes tipos de aplicaciones web ha cambiado significativamente nuestras vidas y es que ahora basta con tener conexión a Internet para hacer la compra, encontrar transporte inmediatamente o encontrar aquella información que necesitamos y que nos interesa.

Hoy en día hay infinitas Webs a nuestra disposición. Algunas de ellas son de entretenimiento, otras sirven para comprar o vender un producto o servicio, otras para intercambiar algo entre usuarios individuales y otras simplemente nos hacen la vida más sencilla. Pero vamos a ver cuáles son los tipos de aplicaciones web que existen para poder comprender cómo funcionan desde dentro.

- ❖ **Página Web estática:** son las más sencillas y no suelen estar sometidas a muchos cambios porque no resulta sencillo hacerlos. Están desarrolladas con código HTML y CSS y también pueden mostrar en algunas partes banners o vídeos, entre otras opciones. La razón que explica que no estén sometidas a muchas variaciones es que para hacerlos hay que descargar el HTML, modificarlo y subirlo de nuevo al servidor.
- ❖ **Página Web dinámica:** Las aplicaciones web dinámicas son mucho más complejas que las anteriores si hablamos desde un punto de vista técnico, y la información y el contenido se actualizan cada vez que los usuarios acceden a la web. Existen muchos lenguajes de programación para aplicaciones web dinámicas, pero los más comunes son PHP y JavaScript. Además de cambiar la información y el contenido, también se puede modificar el diseño de la web.
- ❖ **Tienda virtual o comercio electrónico:** son conocidas como e-commerce y están pensadas para vender productos. Son más complejas porque tienen que incluir el método de pago y estar sincronizadas con el stock de la compañía y con la logística. Este tipo de aplicaciones web son muy comunes.
- ❖ **Portal Web app:** este tipo de aplicaciones web incluyen diferentes categorías y secciones. Pueden tener chats, foros o buscador, entre otras opciones. Infojobs es un portal web app.
- ❖ **Aplicación Web animada:** estaban relacionadas con la tecnología FLASH, aunque hoy en día lo están con CSS y SVG. Permiten presentar contenido con efectos animados.
- ❖ **Aplicación Web con Gestor de contenido:** son las aplicaciones más comunes para todas aquellas webs que actualizan su contenido constantemente. Tienen un gestor de contenidos (CMS) gracias al cual se añaden, modifican y actualizan los contenidos y algunos ejemplos de CMS son WordPress, Drupal y Joomla, entre otros. Las páginas de periódicos digitales o los blogs, ya sean personales o corporativos, son buenos ejemplos de aplicaciones web con gestor de contenido digital.

## 2.5.2 Arquitectura

La arquitectura cliente-servidor consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta. Según (Myriam Yadira, 2011).

### Características

Un sistema Cliente/Servidor es un Sistema de Información distribuido basado en las siguientes características:

- ❖ Servicio: unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- ❖ Recursos compartidos: Muchos clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- ❖ Protocolos asimétricos: Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.
- ❖ Transparencia de localización física de los servidores y clientes: El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- ❖ Independencia de la plataforma HW y SW que se emplee.

- ❖ Sistemas débilmente acoplados. Interacción basada en envío de mensajes.
- ❖ Encapsulamiento de servicios. Los detalles de la implementación de un servicio son transparentes al cliente.
- ❖ Escalabilidad horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).

## Ventajas

- ❖ **Centralización del control:** Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- ❖ **Escalabilidad:** Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- ❖ **Fácil mantenimiento:** Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio. Esta independencia de los cambios también se conoce como encapsulación.
- ❖ **Tecnología:** Existen suficientemente desarrolladas, diseñadas para el paradigma que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.
- ❖ **Manejo de Capas:** Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos.

## Desventajas

- ❖ Dificultad de la programación en comparación con modelo de dos capas ya que se comunican más dispositivos para realizar las peticiones de los clientes; además el incremento del tráfico en la red debido al flujo de los datos.

## Seguridad de Modelo Cliente /Servidor

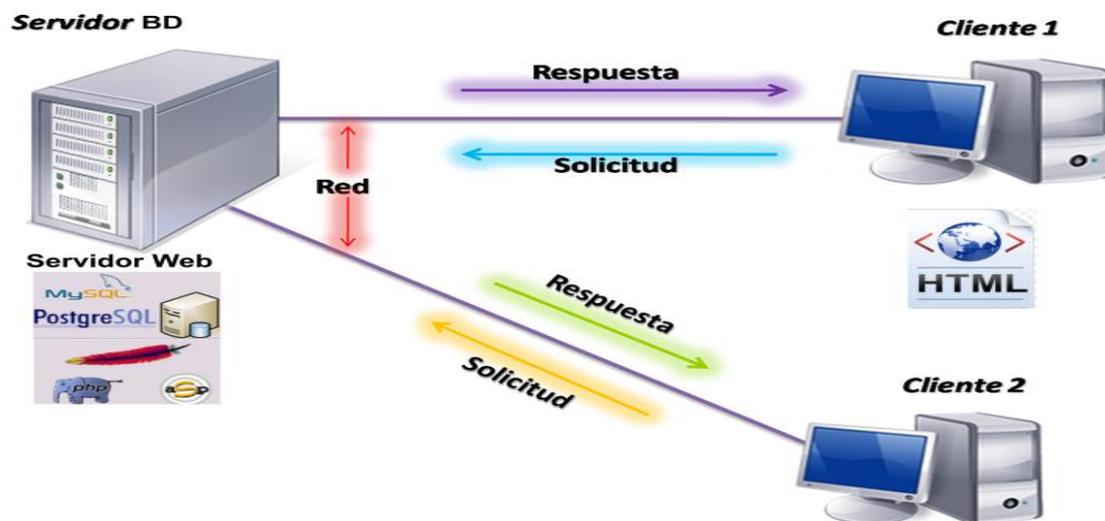
- ❖ Es de gran importancia por el valor intrínseco para los usuarios. Tiene tres componentes:
- ❖ Confidencialidad.- Protección contra individuos no autorizados.
- ❖ Integridad.- Protección contra la alteración o corrupción.
- ❖ Disponibilidad.- Protección contra la interferencia con los procedimientos de acceso a los recursos.

## Arquitectura de Dos y Tres Niveles Cliente /Servidor

- ❖ Arquitectura Dos Niveles

La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

FIGURA Nº 2. 5 Arquitectura en dos niveles



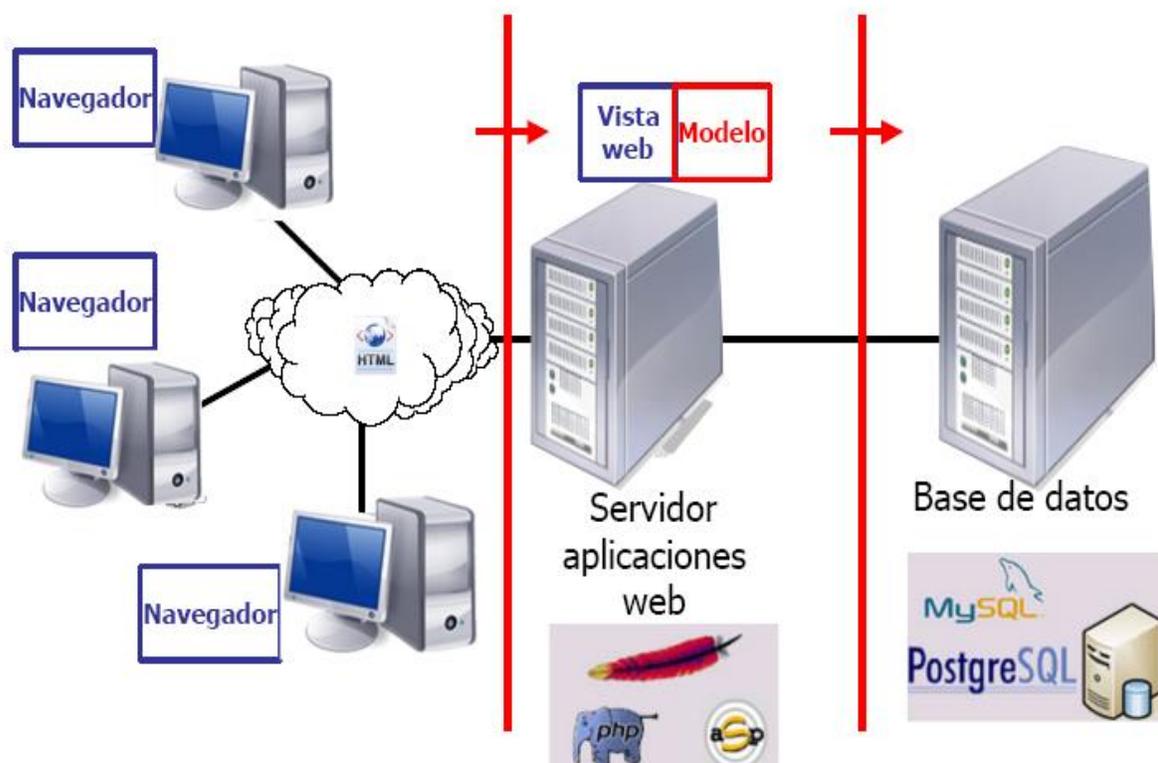
Fuente: (Myriam Yadira, 2011)

- ❖ Arquitectura Tres Niveles

En la arquitectura en tres niveles, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

- ❖ Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación
- ❖ El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- ❖ El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

FIGURA N° 2. 6 Arquitectura en tres niveles



**Fuente:** (Myriam Yadira, 2011)

El uso masivo del término arquitectura en tres niveles también denota las siguientes arquitecturas:

- ❖ Aplicación compartida entre un cliente, un software intermedio y un servidor empresarial.
- ❖ Aplicación compartida entre un cliente, un servidor de aplicaciones y un servidor de base de datos empresarial.

### Comparación entre Ambos Tipos de Arquitecturas

La arquitectura en dos niveles es, por lo tanto, una arquitectura cliente/servidor en la que el servidor es polivalente, es decir, puede responder directamente a todas las solicitudes de recursos del cliente.

Sin embargo, en la arquitectura en 3 niveles, las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo: servidor web/servidor de bases de datos). La arquitectura en 3 niveles permite:

- ❖ Un mayor grado de flexibilidad.
- ❖ Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- ❖ Mejor rendimiento, ya que las tareas se comparten entre servidores.

TABLA N° 2. 1 Comparación de Arquitecturas

Dos Niveles	Tres Niveles
<ul style="list-style-type: none"> <li>❖ Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Cuando se requiera poco procesamiento de datos en la organización.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y la capa de la base de datos.</li> <li>❖ Cuando se requiera mucho procesamiento de datos en la aplicación.</li> <li>❖ En aplicaciones donde la funcionalidad este en constante cambio.</li> </ul>
<ul style="list-style-type: none"> <li>❖ Cuando se tiene una base de datos</li> </ul>	<ul style="list-style-type: none"> <li>❖ Cuando los procesos no están relativamente muy relacionados con los</li> </ul>

<p>centralizada en un solo servidor.</p> <ul style="list-style-type: none"> <li>❖ Cuando la base de datos es relativamente estática.</li> <li>❖ Cuando se requiere un mantenimiento mínimo.</li> </ul>	<p>datos.</p> <ul style="list-style-type: none"> <li>❖ Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.</li> <li>❖ Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.</li> <li>❖ Está muy adecuada para utilizarla con la tecnología orientada a objetos.</li> </ul>
--	---

**Fuente:** (Myriam Yadira, 2011)

### 2.5.2.1 Seguridad de aplicaciones Web

La seguridad de las aplicaciones Web es un tema crítico y complejo para los desarrolladores Web. Ya que requiere realizar estudios y tener entendimiento sobre los puntos de vulnerabilidad en cada aplicación Web.

Se puede clasificar los puntos más importantes:

#### **Seguridad en el servidor**

- ❖ Servidor Web
- ❖ Servidor de Base de Datos
- ❖ Lenguaje de programación

#### **Seguridad en la Aplicación**

- ❖ Control de Acceso
- ❖ Validación de datos de entrada
- ❖ Pruebas de código

#### **Seguridad en la Comunicación**

- ❖ Protocolos de seguridad
- ❖ Certificados de Comunicación

## Copias de Seguridad

- ❖ Backup de la Base de Datos
- ❖ Backup del control de Acceso

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo (Pressman R. , 2005)

A continuación se muestra las vulnerabilidades al lado del cliente

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

Por lo tanto, se puede aplicar las siguientes medidas

**Cortafuegos:** Mecanismo de filtrado combinación de hardware y software para examinando cada paquete de información entrante para garantizar que no sea un dato sospechoso

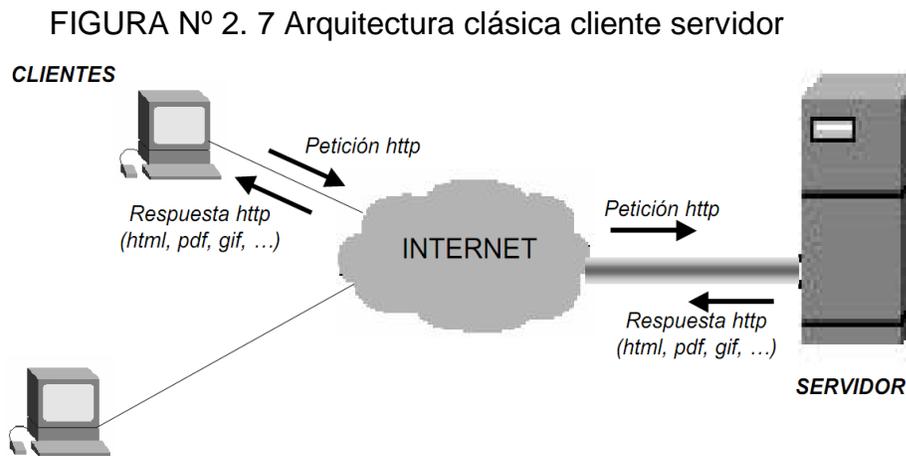
**Uso de Autenticación:** Una contraseña o clave es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso. A aquellos que desean acceder a la información se les solicita una clave; si conocen o no conocen la contraseña, se concede o se niega el acceso a la información según sea el caso.

**Modelo de Criptografía:** La encriptación está asociada con la transformación de un mensaje inteligente a una forma no inteligente con la ayuda de una clave secreta antes de que sea colocada en un medio inseguro.

**Algoritmo básico del MD5:** El mensaje se divide en bloques de 512 bits, añadiendo bits si es necesario al último bloque (de 1 a 448, por una parte, más 64 fijos de un valor  $k \bmod 264$ ). Con cada bloque de 512 se realiza 64 iteraciones. En cada iteración la función MD5 se alimenta con el bloque correspondiente y con el resultado

de la iteración anterior, un vector de cuatro elementos de 32 bits cada uno (con valores predeterminados en la primera iteración). La salida de cada iteración se convierte en el nuevo vector (128 bits).

### 2.5.2.2 Arquitectura cliente – servidor



**Fuente:**( Lewis M. Soncco, 2008)

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

### 2.5.2.3 Cliente

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN<sup>2</sup> o WAN. La

---

<sup>2</sup>LAN son las siglas de **Local Area Network**, Red de área local. Una LAN es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada.

ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente. (Limachi, 2016, p.27).

#### **2.5.2.4 Servidor**

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LAN o WAN, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc. Según (Limachi, 2016, p.27).

#### **2.5.3 Estructura Web**

Es una aplicación web es proporcionada por un servidor web y utilizada por usuarios que se conectan desde cualquier punto vía cliente web. La arquitectura de un Sitio Web tiene tres componentes principales:

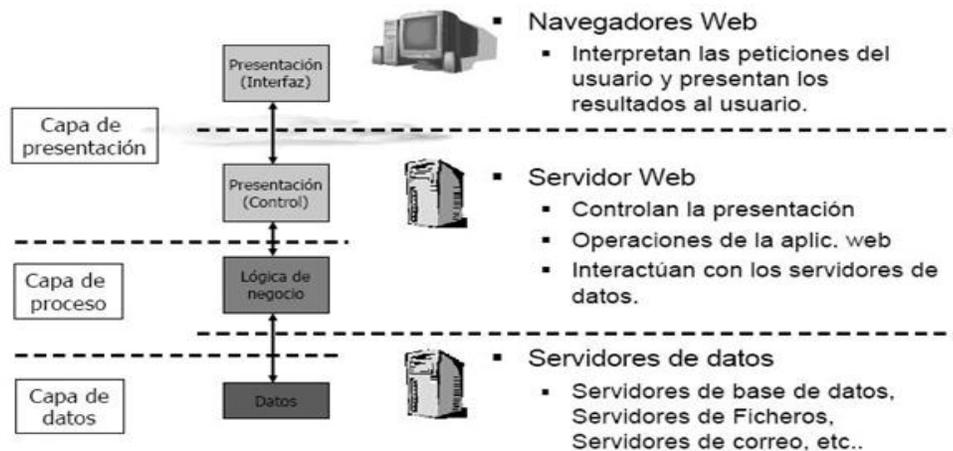
- ❖ Un servidor Web
- ❖ Una conexión de red
- ❖ Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

---

**WAN** es la sigla de Wide Area Network (“**Red** de Área Amplia”). El concepto se utiliza para nombrar a la **red** de computadoras que se extiende en una gran franja de territorio, ya sea a través de una ciudad, un país o, incluso, a nivel mundial.

FIGURA N° 2. 8 Estructura de aplicación web



**Fuente:** (Limachi, 2016, p.27).

Aunque existen muchas posibilidades para estructurar una aplicación web, todas se pueden resumir en tres capas principales. Interfaz gráfica, Controlador o servidor y Base de Datos.

## 2.6 HERRAMIENTAS DE DESARROLLO

### 2.6.1 Servidor apache

Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Su popularidad es tal que, actualmente, cerca del 50% de las páginas web de todo el mundo se ejecutan en un servidor de este tipo.

Aunque se le conoce así, su nombre completo es Apache HTTP Server, y sus responsables tienen también un nombre similar: Apache Software Foundation.

Esta es la firma responsable de todo el código que da forma a este software para servidores que cualquiera puede utilizar sin necesidad de pagar, como también modificar a su total antojo al ser completamente abierto.

En este proyecto compartimos y explicamos de forma detallada cómo instalar en una computadora de escritorio o Laptop el servidor Apache.

Los parámetros que son necesario modificar para que funcione correctamente en su forma elemental y posteriormente agregarle módulos y otras funcionalidades, de acuerdo a las necesidades y propósitos de quien lo vaya a utilizar.

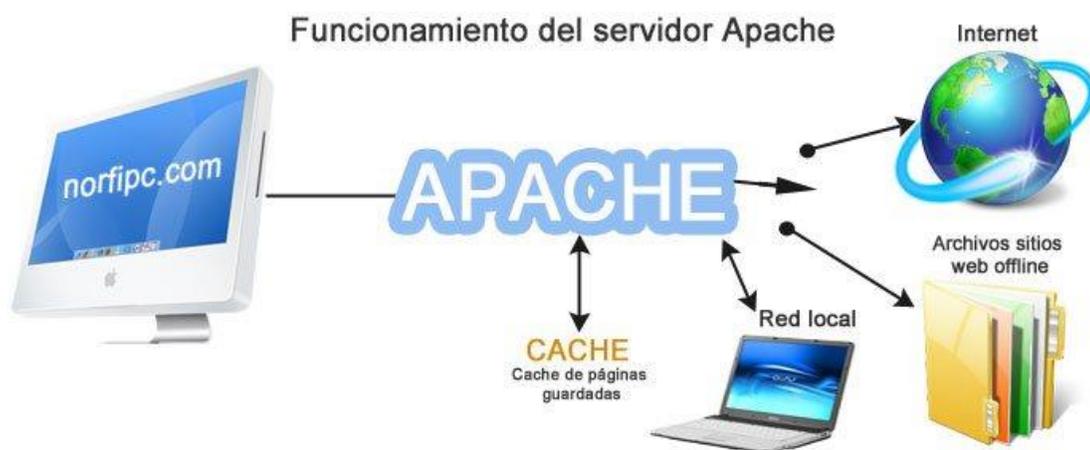
Como hacer funcionar Apache de forma similar a un servidor de internet y cargar las paginas en nuestro navegador offline, como si estuviéramos conectados a la red.

### **Definición de servidor web.**

Un servidor web como su nombre lo indica, es un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización. Existen varios tipos de servidores web, Apache es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows.

Al instalarlo en nuestra PC dispondremos de un servidor completo, con todos los requisitos para ejecutarlo de forma local. Según (Bernal., 2019).

FIGURA N° 2. 9 Servidor web



Fuente (Bernal., 2019)

## 2.6.2 Gestor de 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. Según (Estela, Base de datos, 2019)

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.

El nombre de un campo generalmente identifica la información almacenada en el campo. Los cuales pueden llamarse Nombre, Dirección o Número telefónico. Cada campo tiene un tipo que identifica la clase de información que puede almacenar: números, fechas, caracteres alfanuméricos y otros. Como cada campo contiene un tipo específico de datos, usted puede realizar cálculos y otras operaciones con la información guardada en ellos. También puede sumar los números de dos campos.

Puede comparar la fecha de un campo con la de otro. Puede mostrar el nombre de una persona (almacenado en un campo) después de su apellido (almacenado en otro campo) para construir la primera línea de una etiqueta de correo.

El conjunto de registros que utilizan los mismos campos conforma una tabla. Una base de datos puede contener muchas tablas. La siguiente imagen muestra cómo se relacionan estos conceptos.

### 2.6.2.1 Gestor de base de datos MariaDB

Sistema de base de datos que proviene de MySQL, pero con licencia GPL, desarrollado por Michael Widenius, fundador de MySQL y la comunidad de desarrolladores de software libre.

Como hemos comentado anteriormente, MariaDB se trata de un “sustituto” de MySQL que corre bajo una licencia GPL. Por lo tanto, incorpora todas las mejoras de la versión de MySQL en la que se encuentre y además incluye diferentes mejoras:

Lo primero que tiene que quedar totalmente claro es que al ser MariaDB compatible con MySQL, la migración a MariaDB es simple y directa, no hay que adaptar el código ni nada.

#### **Ventajas de MariaDB.**

Nuevos motores de almacenamiento, para la mayoría de usuarios lo interesante es, que viene a reemplazar a MyISAM y también tenemos XtraDB que reemplaza a InnoDB. Los nuevos motores de almacenamiento son:

- ❖ Aria: Un motor de almacenamiento a prueba de fallos basado en MyISAM.
- ❖ XtraDB: El reemplazo del motor InnoDB basado en el plug-in de InnoDB.
- ❖ PBXT: Un motor de almacenamiento transaccional con una gran cantidad de nuevas y bonitas características.
- ❖ FederatedX: El reemplazo del motor Federated.

Mejoras de velocidad sobre todo en consultas complejas cuando se usa el motor de almacenamiento Aria, ya que Aria cachea los datos de tablas temporales en memoria, lo que supone un rendimiento frente al uso del disco duro (que es lo que emplea MyISAM).

Se añaden nuevas tablas de sistema (INFORMATION\_SCHEMA) para almacenar estadísticas que nos pueden ayudar a optimizar las bases de datos.

El sistema para manejar las conexiones se ha mejorado, ya que implementa el sistema pool-of-threads de MySQL 6.0 con el que podemos tener más de 200.000 conexiones a MariaDB.

En general se han hecho muchas modificaciones para mejorar el rendimiento, velocidad e incluso implementar características nuevas.

### **Desventajas de MariaDB.**

La única desventaja es la propia de tener que desinstalar MySQL e instalar MariaDB, pero la migración es muy sencilla. Según (Ramos, 2018).

### **2.6.3 Lenguaje de programación PHP7**

PHP son las 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 (Alvarez, 2001).

### **VENTAJAS Y DESVENTAJAS.**

#### **Ventajas:**

- ❖ Es un lenguaje multiplataforma.
- ❖ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ❖ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado

HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- ❖ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con varios gestores de base de datos.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).

### **Desventaja:**

Como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución. Según (MARTINEZ, 2011).

#### **2.6.3.1 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.

#### **2.6.3.2 Definición de MVC**

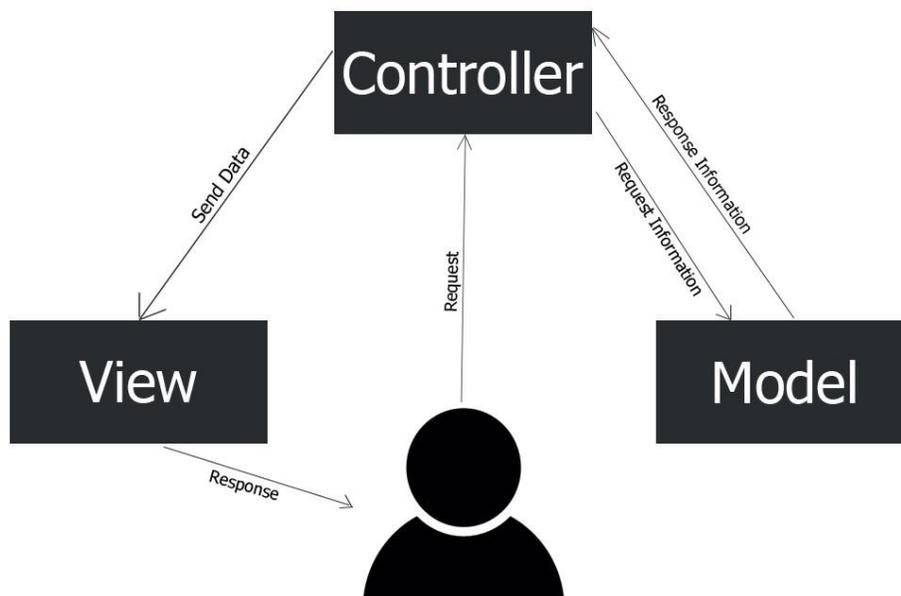
El MVC o Modelo Vista Controlador es un patrón de arquitectura de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir por un lado define los componentes para la representación de la información y por otro lado la interacción del usuario. Según (Fontán, 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. 10 Definición de MVC

## Model-View-Controller



Fuente (Fontán, 2012)

### Algunas 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.

## **2.6.4 Herramienta de diseño**

### **2.6.4.1 Definición de 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.

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 para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.
- ❖ CSS3: Nos ofrece una nueva gran variedad de opciones para hacer diseños más sofisticados.

Según (Martin, 2019)

#### **2.6.4.2 Definición de 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 (Jiménez, 2019).

### Funcionamiento CSS3

Vamos a ver en un pequeño esquema qué es lo que sucede desde que se solicita una página web hasta que se le aplica el estilo. El proceso es el siguiente:

#### 2.6.4.3 Definición de 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, 2018)

Jquery es un Framework de Javascript que nos permite la creación de aplicaciones complejas del lado del cliente.

FIGURA N° 2. 11 Definición de Jquery



Fuente (Porto, 2018).

#### 2.6.4.4 Definición de 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 (González, 2016).

FIGURA N° 2. 12 Lenguaje de programación Javascript



Fuente (González, 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

#### **2.6.4.5 Definición de AJAX**

AJAX son las siglas de Asynchronous JavaScript And XML, (Javascript asíncrono y XML). No es en sí un lenguaje de programación, sino una nueva técnica que combina varios lenguajes de programación.

La ventaja de ajax respecto a otros lenguajes de programación web es la asincronía. Esto consiste en que cuando queremos intercambiar datos con el servidor (por ejemplo enviar o comprobar un formulario, consultar una base de datos, etc), la página no se queda parada esperando la respuesta, sino que se pueden seguir ejecutando acciones mientras tanto.

Con ajax podemos crear páginas interactivas. En éstas solicitamos datos al servidor, los cuales podemos tener guardados en otras páginas o en bases de datos. El servidor devuelve los datos, los cuales se cargan en la misma página y en segundo plano. Lo de "segundo plano" significa que mientras esperamos que se reciban los datos la página no se queda parada, y el usuario o la programación de la página puede seguir haciendo otras cosas.

Para poder entender este manual debemos tener conocimientos de HTML, CSS, y Javascript. Es conveniente también tener nociones de XML y acceso al DOM. Para las últimas páginas necesitamos también tener conocimientos de PHP, ya que es el lenguaje que se emplea para procesar los datos en el servidor. Según (Alsina, 2018)

#### **2.6.4.6 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 con 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 (Ruiz, 2014).

## **2.7 INGENIERÍA WEB BASADA EN UML (UWE)**

UWE surgió a finales de la década de los noventa con la idea de encontrar un estándar para construir modelos de análisis y diseño para sistemas basados en la Web. El objetivo fue usar un lenguaje común o por lo menos definir un modelo a partir de los existentes.

UWE es una metodología basado en el proceso Unificado y UML para el desarrollo de aplicaciones Web. Según (Maximilians, 2016).

### 2.7.1 Características

La metodología UWE define vistas especiales representadas gráficamente por diagramas UML, tales como el análisis de requerimientos, diseño conceptual modelo de navegación y el modelo de presentación. UWE no limita el número de diagramas posibles de una aplicación.

- ❖ Es una metodología orientada a objetos, iterativa e incremental basada en UML.
- ❖ Se basa también en el proceso de desarrollo del software unificado.
- ❖ Proporciona un diseño sistemático y una generación semiautomática en las aplicaciones web a través de un Framework de publicación XML.
- ❖ UWE define su propio perfil UML en el cual se definen todos los elementos necesarios para modelar los diferentes aspectos de una aplicación web que son la presentación y la navegación entre otros.
- ❖ En esta metodología se proponen 2 tipos de diagramas para el modelado de la navegación que son: el modelo de espacio, el cual se definen todos los caminos navegacionales, es decir todas aquellas asociaciones de navegación directa entre los distintos objetos de la aplicación más bien conocida como clases de navegación, y el segundo modelo de estructura de navegación el cual se define la estructura de acceso que son utilizadas en la navegación, es decir todo aquello referente a menús, índices y demás.

### 2.7.2 Metodología de desarrollo UWE

Metodologías de desarrollo de Software. Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada

proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas. Según (Maximilians, 2016).

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.

FIGURA N° 2. 13 Metodología de desarrollo UWE



Fuente: (Maximilians, 2016).

### **2.7.3 Característica de la metodología UWE**

UML es una técnica de modelamiento propietaria, es la aceptación de UML en el proceso de desarrollo de software, la flexibilidad para la definición de un lenguaje de modelamiento específico en el dominio WEB, también llamado perfil UML.

UWE hace uso de notación UML pura y los tipos de diagramas UML en donde sea posible para el análisis y diseño de aplicaciones WEB.

## **2.7.4 Fase de la metodología UWE**

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas. Las fases o etapas a utilizar son:

- ❖ Captura, análisis y especificación de requisitos.
- ❖ Diseño del sistema.
- ❖ Codificación del software.
- ❖ Pruebas.
- ❖ La Instalación o fase de Implementación.
- ❖ Mantenimiento.

### **2.7.4.1 Captura, análisis y especificación de requisitos**

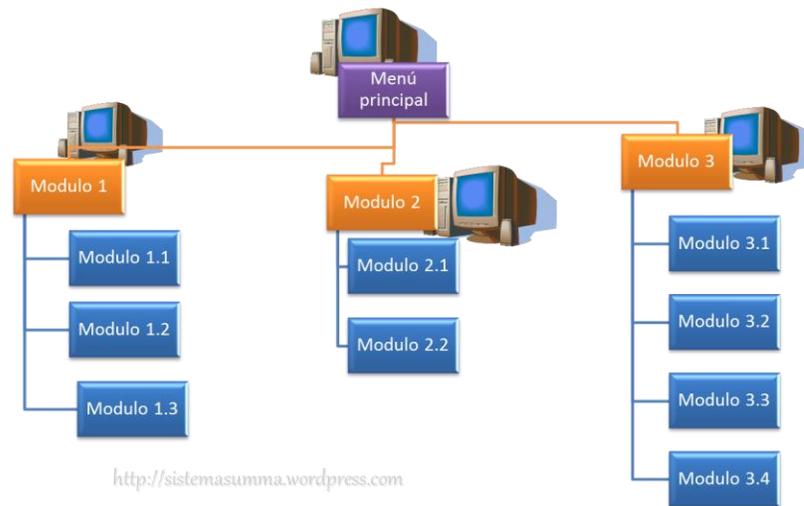
En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web. Según (Esquen, 2015)

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

### **2.7.4.2 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.

FIGURA Nº 2. 14 Diseño del sistema



Fuente (Esquen, 2015)

#### 2.7.4.3 Codificación del software

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

#### 2.7.4.4 Pruebas

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

#### 2.7.4.5 La Instalación o fase de Implementación

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

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones. Según (Esquen, 2015).

#### 2.7.4.6 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. Según (Esquen, 2015).

#### 2.7.5 Modelo de la metodología UWE

El modelo que propone UWE está compuesto por sub-modelos.

- ❖ **Modelo de caso de uso:** Modelo para capturar los requisitos del sistema.
- ❖ **Modelo de contenido:** Es un modelo conceptual para el desarrollo del contenido.
- ❖ **Modelo de navegación:** En el cual se encuentra la presentación del sistema y el modelo de flujo.
- ❖ **Modelo de estructura del proceso:** Incluye el modelo de la interfaz de usuario y el modelo de ciclo de vida del objeto.
- ❖ **Modelo de presentación:** En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además, distinguen entre las fases de captura, definición y validación de requisitos.

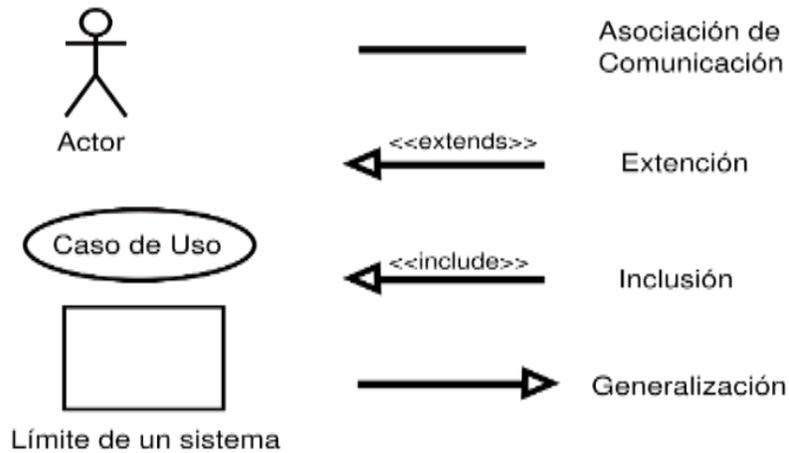
##### 2.7.5.1 Modelo de caso de uso

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 deberían realizarse para llevar a cabo algún proceso.

Los personajes o entidades que participaran en un caso de uso se denominan actores UWE provee diferentes estereotipos. Un caso de uso especifica el comportamiento de un sistema o una parte del mismo, y es una descripción de un conjunto de secuenciales de acciones, donde cada secuencia representa la interacción de los elementos externos del sistema (actores) con el propio sistema.

Un diagrama de casos de uso consta de los siguientes elementos:

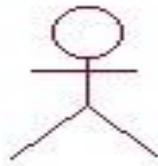
FIGURA Nº 2. 15 Estereotipos de Diagrama de Caso de Uso



Fuente: (Soto, 2008)

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

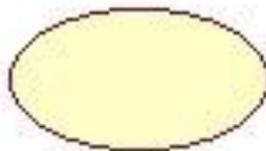
FIGURA Nº 2. 16 Actor



Fuente: (Soto, 2008)

Un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

FIGURA Nº 2. 17 Caso De Uso



Fuente: (Soto, 2008)

Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

FIGURA N° 2. 18 Asociación



**Fuente:** (Soto, 2008)

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

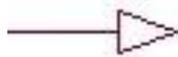
FIGURA N° 2. 19 Dependencia o Instanciación



**Fuente:** (Soto, 2008)

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

FIGURA N° 2. 20 Generalización



**Fuente:** (Soto, 2008)

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).

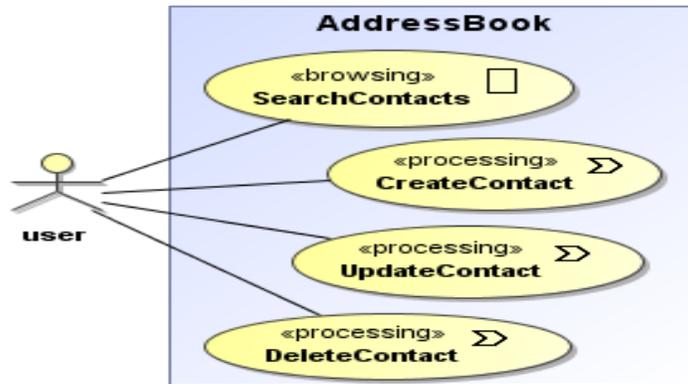
Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

**EXTENDS:** Se recomienda utilizar cuando un caso de uso es similar a otro (características).

**USES:** Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de usar o heredar.

FIGURA N° 2. 21 Diagrama de Caso de Uso en la Metodología UWE



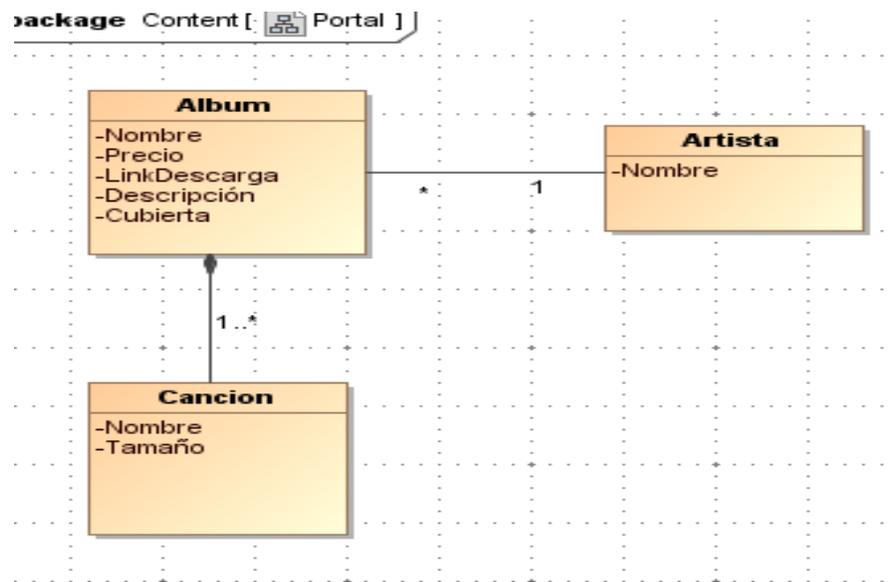
Fuente: (Soto, 2008).

### 2.7.5.2 Modelo de contenido

Dentro de este ciclo se modela el universo de la aplicación, creando para ello el modelo de dominio.

**Diagrama de contenido.** Este es un diagrama UML normal de clases, por ello debemos pensar en las clases que son necesarias. Los diagramas de clases se describen las estructuras estáticas de un sistema.

FIGURA N° 2. 22 Diagrama de Contenido en la Metodología UWE



Fuente: (Engineering, 2012)

Modelo de contenido define el contenido de los datos de la aplicación.

### 2.7.5.3 Modelo de navegación

Este diseño especifica que objetos pueden ser visitados a través de la aplicación web.

En el proceso de construir el modelo de espacio de navegación el desarrollador toma decisiones cruciales de diseño, tales como qué vista del modelo conceptual es necesaria para la aplicación y cuáles serán los caminos de navegación requeridos para el aseguramiento de la funcionalidad. Los elementos utilizados para este modelo son las clases de navegación y las asociaciones de navegación, que expresan la navegación directa.

La navegación está fuertemente simplificada y se destina principalmente para demostrar el uso de los elementos del modelo para lo cual se utiliza estereotipos e iconos las que se describen en la siguiente Tabla.

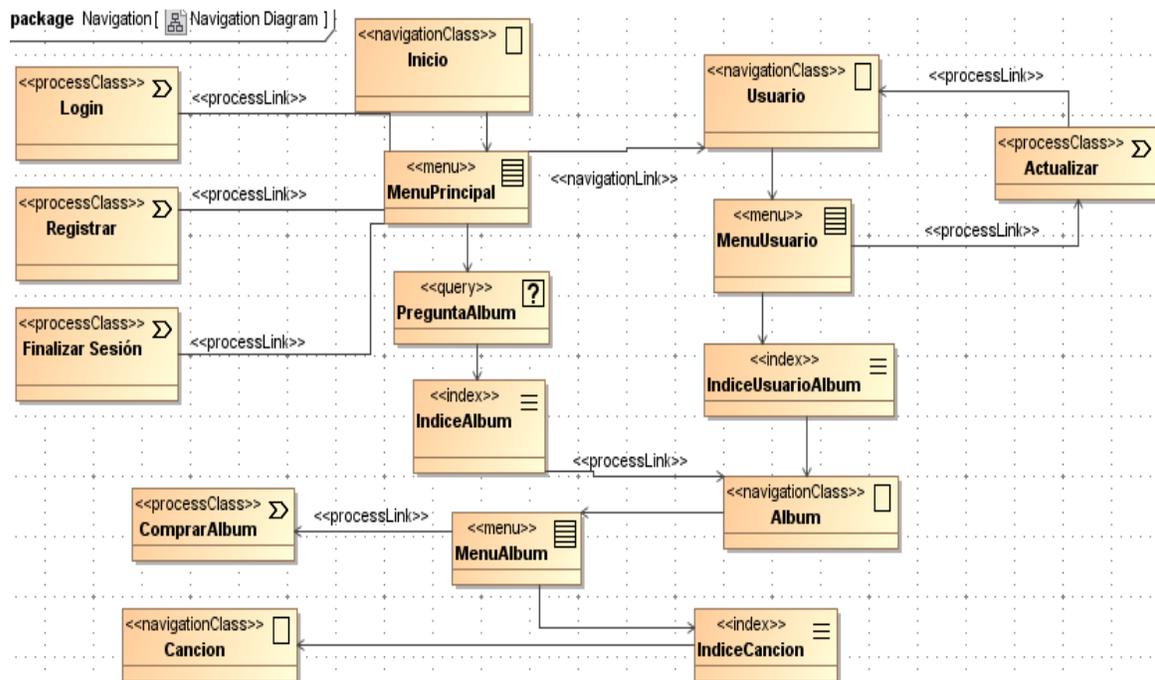
TABLA N° 2. 2 Descripción de Estereotipos

Icono	Descripción	Icono	Descripción
	Clase de Navegación		Menú
	Índice		Pregunta
	Visita Guiada		Clase de Proceso
	Nodo Externo		

Fuente: Elaboración Propia Basado en (KOCH, 2000)

Descripción de Estereotipos

FIGURA N° 2. 23 Modelo de Navegación de la Aplicación Portal Musical



Fuente: (Engineering, 2012).

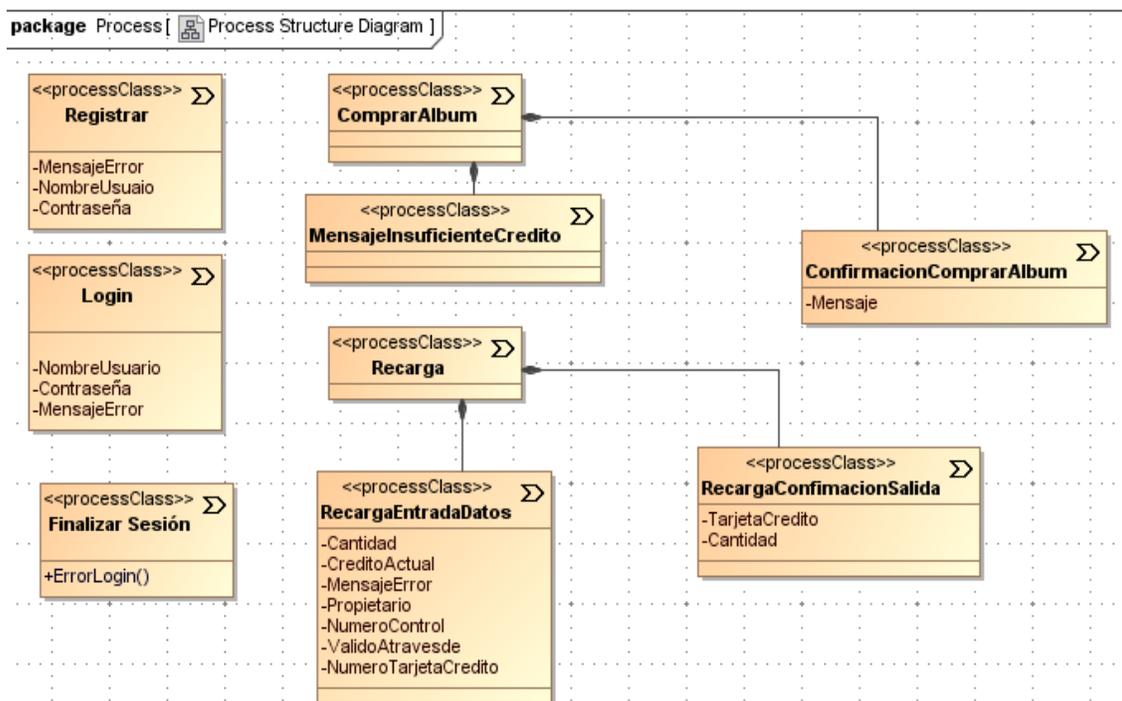
En este diagrama podemos ver como se relacionan las clases entre ellas, llegando a un entendimiento superior que un simple diagrama UML con la extensión que UWE

aporta a este estándar, esta extensión es la que explicamos en el modelo de navegación de la introducción de este documento con los estereotipos correspondientes.

#### 2.7.5.4 Modelo de estructura del proceso

El modelo de proceso del ejemplo se compone de las clases de procesos que se integran en el modelo de navegación, las clases de proceso adicionales para manejar la entrada del usuario y las actividades que definen el comportamiento de los procesos. El modelo de proceso se muestra la relación de la clase de proceso principal y clases de procesos, que se han asociado el flujo del proceso.

FIGURA N° 2. 24 Modelo de estructura del proceso



Fuente: (Engineering, 2012)

#### MODELO DE ESTRUCTURA DEL PROCESO:

En el cual se incluye las relaciones entre clases de proceso, se crea un diagrama de clases donde cada una se presente con un estereotipo de clases de proceso. Se conjuntan estas clases y se asocian a una superclase que representa el

proceso, se agrega algunas clases si es necesario para denotar algunas interacciones u operaciones en común.

#### **MODELO DE FLUJO DEL PROCESO:**

La conducta de un proceso es representando mediante un diagrama de actividades UML, describiendo el flujo de una clase de proceso, lo que sucede cuando un usuario navegación hacia una clase de proceso.

#### **2.7.5.5 Modelo de presentación**

La presentación se describe en función de distintos modelos estándares UML. El Modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenecen a una página web. Podemos usar un Diagrama de Presentación con el fin de proveer esta información la metodología UWE presenta diferentes estereotipos.

Las propiedades pueden anidarse, por ejemplo, cada contacto cubre diferentes textos y botones. Ello significa, que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados en la página.

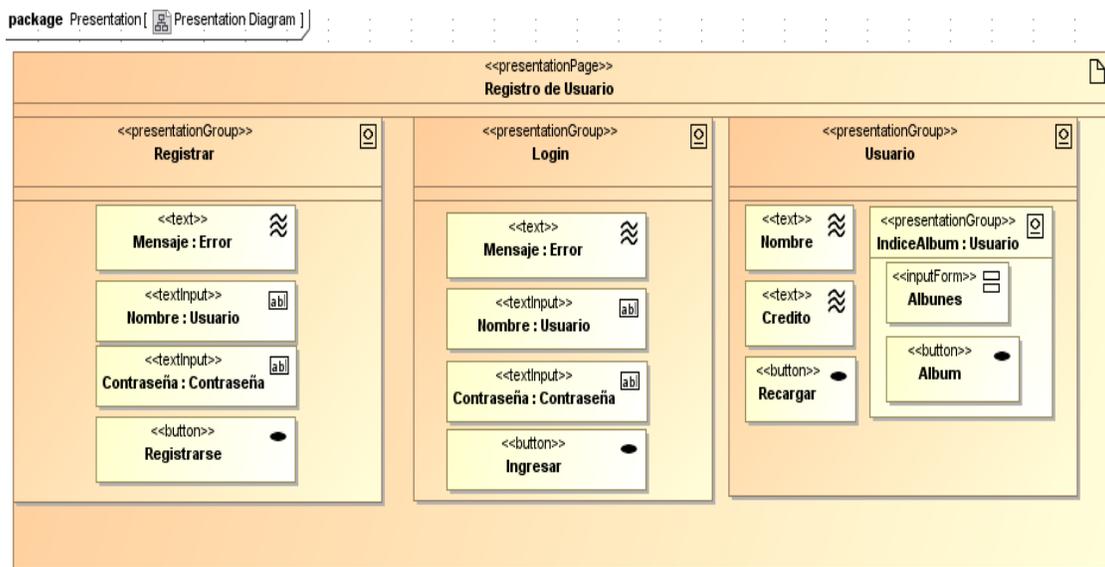
Para el modelo de presentación se utiliza los siguientes Estereotipos e iconos como se detalla en el siguiente cuadro:

TABLA N° 2. 3 Descripción de Estereotipos Modelo de presentación

Icono	Descripción	Icono	Descripción
	Grupo de Presentación		Página de Presentación
	Texto		Entrada de Texto
	Ancla		FileUpload
	Boton		Imagen
	Formulario		Componente de Cliente
	Alternativas de Presentación		Selección

Fuente: (KOCH, 2000)

FIGURA N° 2. 25 Modelo de presentación



Fuente: (Engineering, 2012)

El modelo de presentación del ejemplo se muestra como un diagrama de estructura compuesta UML. En este tipo de diagrama, las propiedades que están contenidos por la composición se muestran como rectángulos que están contenidos en la figura de la clase que contiene.

Los siguientes diagramas y los estereotipos son solamente representados por sus iconos. En MagicDraw se puede configurar la visualización de ambos: nombres e iconos de los estereotipos. Por ello, los estereotipos presentan alternativas y es usado nuevamente para evitar los múltiples modelos de todo el contenido de ambos formularios de ingreso de datos.

## 2.8 MÉTRICA DE CALIDAD DE SOFTWARE

- ✓ **Calidad de 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, 1998).

- ✓ **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. Según (Pressman, 1998).

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.

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:

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

### 2.8.1 ISO 9000

**Según** (Engineering, 2012). La serie de Normas ISO 9000 son un conjunto de enunciados, los cuales especifican que elementos deben integrar el Sistema de Gestión de la Calidad de una Organización y como deben funcionar en conjunto estos elementos para asegurar la calidad de los bienes y servicios que produce la Organización. Al hablar de Organización nos estamos refiriendo a una Empresa, Compañía o cualquier Estructura Organizada que genere o comercialice productos o servicios de algún tipo.

FIGURA Nº 2. 26 NORMA ISO 9000



Fuente (Engineering, 2012).

### 2.8.2 Estándar ISO/IEC 9126

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado "Information technology-Software

product evaluation-Quality characteristics and guidelines for their 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 auditoría de software.

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos siete 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. Según (Engineering, 2012).

#### **2.8.2.1 Métricas de calidad del modelo de ISO-9126**

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

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

La calidad de software es medible y varía de un sistema o programa a otro. Un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad mientras que un software para ser explotado durante un largo, necesita ser confiable, mantenible y flexible para disminuir los costos.

La calidad de software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es impredecible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

Características propias del software aquellas que tú quieres controlar y asegurar, el software es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla. El software puede tener errores, incidencias pero no son similares a lo que cualquier equipo de carácter físico. Según (Engineering, 2012).

### **2.8.2.1.1 Norma de calidad ISO-9126**

#### **¿QUE ES LA CALIDAD?**

Conjunto de propiedades y características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas.

#### **¿QUE ES LA 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.

#### **Definición del ISO 9000**

Un Conjunto de normas internacionales que definen requisitos para los llamados sistemas de gestión de calidad. ISO 9000 tiene reconocimiento internacional y se aplica a todas las industrias, incluso al desarrollo de tecnologías de la información.

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en término de seis atributos cada una de las cuales se detalla a través de un conjunto de sub-atributos que permiten analizar y profundizar en la evaluación de calidad de productos de software. A continuación, se explican estos atributos.

**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.

❖ **ADECUACIÓN:** Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.

- ❖ **EXACTITUD:** Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- ❖ **INTEROPERABILIDAD:** Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados.
- ❖ **SEGURIDAD:** Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos. Cumplimiento funcional. Según (Engineering, 2012).

**CONFIABILIDAD:** El conjunto de atributos que soporta la capacidad del software para mantener su nivel de rendimiento bajo condiciones establecidas por un periodo de tiempo establecido.

- ❖ **MADUREZ:** Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.
- ❖ **RECUPERABILIDAD:** Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectados en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- ❖ **TOLERANCIA A FALLOS:** Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.
- ❖ **CUMPLIMIENTO DE FIABILIDAD:** La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.

**USABILIDAD:** Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- ❖ **APRENDIZAJE:** Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- ❖ **COMPRENSIÓN:** Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- ❖ **OPERATIVIDAD:** Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.

- ❖ **EFICIENCIA:** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas.
- ❖ **COMPORTAMIENTO EN EL TIEMPO:** Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.
- ❖ **COMPORTAMIENTO DE RECURSOS:** Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

**MANTENIBILIDAD:** Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

- ❖ **ESTABILIDAD:** Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.
- ❖ **FACILIDAD DE ANÁLISIS:** Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
- ❖ **FACILIDAD DE CAMBIO:** Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
- ❖ **FACILIDAD DE PRUEBAS:** Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.

**PORTABILIDAD:** Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.

- ❖ **CAPACIDAD DE INSTALACIÓN:** Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
- ❖ **CAPACIDAD DE REEMPLAZAMIENTO:** Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado. Según (Engineering, 2012).

### 2.8.2.1.1.1 Funcionalidad

La métrica orientada a la función, utiliza una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la funcionalidad no se puede medir directamente es necesario derivar mediante otras medidas directas como el punto función.

#### 2.8.2.1.1.1.1 Punto función.

La métrica de punto de función (PF), se puede usar como medio para predecir el tamaño de un sistema que se va a obtener de un modelo de análisis. Los puntos de función se obtienen utilizando una relación empírica basada en medidas cuantitativas del dominio de información de software y valoración subjetiva de la complejidad del software.

Para poder determinar la funcionalidad del sistema se debe determinar cinco características del dominio de información.

- ❖ **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, estas aplicaciones pueden ser: insertar, actualizar, borrar datos del sistema.
- ❖ **NÚMERO DE SALIDAS DE USUARIO:** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, datos en pantalla, mensaje de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.
- ❖ **NÚMERO DE ARCHIVOS:** Se cuenta cada archivo maestro lógico (esto es un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

- ❖ **NÚMERO DE INTERFACES EXTERNAS:** Se cuentan todas las interfaces legibles por la maquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

TABLA N° 2. 4 Cálculo de Punto Fusión No ajustado

Parámetros de Medición	Cuentas	Factor de Ponderación			
		Simple	Medio	Complejo	Total
No de Entradas de Usuario	x	3	4	6	x
No de Salidas de Usuario	x	4	5	7	x
No de Peticiones de Usuario	x	3	4	6	x
No de Archivos en Operación	x	7	10	15	x
No de Interface Externos	x	5	7	10	x

**Fuente:** (Pressman, 2006).

El punto función se puede calcular mediante la siguiente ecuación:

$$PF = Cuenta\ Total * (X + Y * \sum F_i)$$

Dónde:

Cuenta Total: Es la suma de todas las entradas obtenidas en No de Entradas, No de Salidas, No de Archivos y No de Interfaces Externas.

X: Nivel de confiabilidad del sistema es de (0,65).

Y: Nivel de error igual a (0,01).

F<sub>i</sub> (i=1 a 14): Son los valores de ajustes de complejidad según las respuestas a las preguntas destacados en la siguiente tabla.

TABLA N° 2. 5 Cálculo de Punto Fusión ajustado

No	Factores de Complejidad	Valor
1	¿Requiere el sistema copias de seguridad y de recuperación fiable?	x
2	¿Se requiere comunicación de datos?	x
3	¿Existen funciones de procesamiento distribuido?	x
4	¿Es crítico el rendimiento?	x
5	¿Se ejecutan el sistema en un entorno operativo existente y fuertemente utilizado?	x
6	¿Requiere el sistema entrada de datos interactiva?	x
7	Facilidad Operativa.	x
8	¿Se actualizan los archivos maestros de forma interactiva?	x
9	¿Son complejas as entradas, las salidas, los archivos o las peticiones?	x
10	¿Es complejo el procesamiento interno?	x
11	¿Se ha diseñado el código para ser reutilizable?	x
12	Facilidad de Instalar.	x
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferente organización?	x
14	Facilidad de cambio.	x
total		

Fuente: (Pressman, 2006).

TABLA N° 2. 6 Valores de Ajuste de Complejidad

Sin Influencia	Incidental	Moderada	Medio	Significativa	Esencial
0	1	2	3	4	5

Fuente: (Pressman, 2006).

Cada una de las preguntas anteriores es respondida usando una escala con rangos desde 0 (lo importante o aplicable) hasta 5 (absolutamente esencial). Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas de los dominios de información se determina empíricamente.

El indicador de funcionalidad se lo encuentra calculando la siguiente relación.

$$\% \text{Funcionalidad} = \text{\$PF} = \frac{\text{PFCalculada}}{\text{PFMaxima}} * 100\%$$

### 2.8.2.1.1.2 Confiabilidad

La confiabilidad es la cantidad tiempo que el software está disponible para su uso, es decir, la cantidad de tiempo que el sistema en funcionamiento dentro de la institución y esté libre de fallas; se puede considerar que mayor el número de fallas, menor confiabilidad, pero a menor número de fallas mayor será la confiabilidad.

Por lo tanto, para poder medir la confiabilidad del sistema, se tomó la siguiente fórmula que calcula la confiabilidad del sistema.

Probabilidad de hallar una falla:  $P(T \leq t) = F(t)$

Probabilidad de no hallar una falla:  $P(T > t) = 1 - F(t)$

Con:  $F(t) = F_c * (e^{-\lambda / 7 * 12})$

Dónde:

$F_c = 0.87$  Funcionalidad del sistema

$\lambda = 1$  tasa de fallos dentro de un mes

También utilizando la métrica de madurez de la siguiente tabla.

TABLA N° 2. 7 Métrica de madurez

Nombre	Suficiencia de pruebas
Propósito	Cuántos de los casos de prueba necesarios están cubiertos por el plan de pruebas
Método de aplicación	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.

Formula	$x = a/b$ a = número de casos de prueba en el plan b = número de casos de prueba requeridos
Interpretación	$0 \leq x$ Entre "x" sea mayor, mejor la suficiencia.
Tipo de escala	Absoluta
Tipo de medida	$x = \text{count}/\text{count}$ a = count b = count
Fuente de medición:	A provienen del plan de pruebas B proviene de la especificación de requisitos.
ISO/IEC 12207 SCP	Aseguramiento de Calidad Resolución de problemas Verificación
Audiencia:	Desarrolladores, Mantenedores.

**Fuente:** (Pressman, 2006)

### 2.8.2.1.1.3 Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que debería invertir el usuario para utilizar el sistema.

- ❖ **COMPRESIBILIDAD:** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- ❖ **FACILIDAD DE APRENDIZAJE:** Establece atributos del software relativos al esfuerzo que los usuarios deben para aprender a usar la aplicación.
- ❖ **OPERABILIDAD:** Agrupa los conceptos que evalúan la operación y el control del sistema.

Para realizar el cálculo de usabilidad del sistema, aplicamos la siguiente Tabla:

TABLA N° 2. 8 Métrica de Entendibilidad

PREGUNTAS	Ponderación %
¿Puede Utilizar con facilidad el sistema?	x
¿Puede Controlar operaciones que el sistema solicita?	x
¿Las Respuestas del sistema son complicadas?	x
¿El Sistema permitió la retroalimentación de información?	x
¿El sistema cuenta con interface agradable a la vista?	x
¿La respuesta del sistema es satisfactoria?	x
¿Le parece complicada las funciones del sistema?	x
¿Se hace difícil o dificultoso aprender a manejar el sistema?	x
¿Los resultados que proporciona el sistema facilitan el trabajo?	x
¿Durante el uso del sistema se produjo errores?	x
USABILIDAD	x

Fuente: (Pressman, 2006)

TABLA N° 2. 9 Ajuste de Preguntas

SI	NO
5	0

Fuente: (Pressman, 2006)

#### 2.8.2.1.1.4 Mantenibilidad

Es la facilidad con que una modificación puede ser realizada en el sistema. Las modificaciones puedes incluir correcciones, mejoras o adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos. Para este fin, Pressman nos sugiere el Índice de Madurez de Software (IMS) para determinar la

estabilidad de un producto software. Dicha IMS es calculada por la siguiente ecuación:

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

Dónde:

Mt: Número de módulos de la versión actual.

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

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

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

TABLA N° 2. 10 Métrica de Cambiabilidad

Nombre:	Registrabilidad de cambios
Propósito:	¿Se registran adecuadamente los cambios a la especificación y los módulos con comentarios en el código?
Método de Ampliación:	Registrar la proporción de información sobre cambios a los módulos.
Medición, fórmula:	X = A/B A = número de cambios a funciones o módulos que tienen comentarios confirmados. B = total de funciones o módulos modificados.
Interpretación:	0 <= X <= 1 Entre más cercano a 1, más registrable. 0 indica un control de cambios deficiente o pocos cambios y alta estabilidad
Tipo de Escala:	absoluta

**Fuente:** (Pressman, 2006)

### 2.8.2.1.1.5 Portabilidad

La portabilidad se define como la característica que posee un software para ejecutar en diferentes plataformas, el código fuente del software es capaz de

reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad es menor la dependencia del software con respecto a la plataforma.

La ISO 9126 se basa en que el objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso del software por parte de los usuarios y es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

## **2.9 ANALISIS DE COSTOS DE COCOMO**

### **2.9.1 COCOMO**

Según (Mercado, 2018). Modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (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.

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

- ❖ 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.

### **2.9.1.1 Modelos de estimación de costo con COCOMO**

En la estimación del tamaño de software COCOMO 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:

- ❖ Determinar Cantidad de Objetos: Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
- ❖ Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil)
- ❖ 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.
- ❖ Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

FIGURA Nº 2. 27 Puntos Objeto

<b>Para Pantallas</b>			
<b>Cantidad de Vistas Contenidas</b>	<b>Cantidad y fuente de las tablas de datos</b>		
	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
<b>Para Reportes</b>			
<b>Cantidad de Vistas Contenidas</b>	<b>Cantidad y fuente de las tablas de datos</b>		
	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 (Mercado, 2018).

## 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.

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

Según (Durán, 2016). 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, 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 (Constructive Cost Model) 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.

FIGURA N° 2. 28 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>				
	Media	7					
	Baja	5					
			Modo (Tipo de Proyecto)	a	b		
			Orgánico	3.2	1.05		
			Semiacoplado	3.0	1.12		
			Empotrado	2.8	1.20		
			$E = a * (KLDC)^b * FAE$ $D = 2.5 * (E)^{0.38}$				

Fuente (Mercado, 2018).

### 2.9.1.2 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:

#### 2.9.1.2.1 Modelo básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes.

TABLA N° 2. 11 Constantes de Costes

Modo	a	b	c	d
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semilibre</b>	3.00	1.12	2.50	0.35
<b>Rígido</b>	3.60	1.20	2.50	0.32

Fuente: (COCOMO, 2013)

Estos valores son para las fórmulas:

- ✓ Personas necesarias por mes para llevar adelante el proyecto (**MM**) =  $a \cdot (Kl^b)$
- ✓ Tiempo de desarrollo del proyecto (**TDEV**) =  $c \cdot (MM^d)$
- ✓ Personas necesarias para realizar el proyecto (**CosteH**) =  $MM/TDEV$
- ✓ Costo total del proyecto (**CosteM**) =  $CosteH \cdot \text{Salario medio entre los programadores y analistas.}$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

### 2.9.1.2.2 Modelo intermedio

En este modelo se introducen 15 atributos de coste para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula se muestran en la siguiente tabla.

TABLA N° 2. 12 Coeficientes de COCOMO

PROYECTO SOFTWARE	A	b
Orgánico	3,2	1,05
Semi-Libre	3,0	1,12
Rígido	2,8	1,20

**Fuente:** (COCOMO, 2013)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los

modos orgánico y rígido han cambiado, para mantener el equilibrio al rededor del SemiLibre con respecto al efecto multiplicador de los atributos de coste.

### **Ecuaciones Nominales De Coste.**

Para cada modo de desarrollo, los 15 atributos del coste intervienen como multiplicadores en el coste nominal,  $K_n$ , para producir el coste ajustado.

Las ecuaciones nominales de coste para el modelo intermedio son

Modo orgánico  $K_n = 3.2 Sk1.05$

Modo semiencajado  $K_n = 3.0 Sk1.12$

Modo empotrado  $K_n = 2.8 Sk1.20$

### **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:

### **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.

### **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.

## 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.

## 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 N° 2. 13 Atributos de Coste

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
<b>Atributos de software</b>						
<b>Fiabilidad</b>	0,75	0,88	1,00	1,15	1,40	
<b>Tamaño de Base de datos</b>		0,94	1,00	1,08	1,16	
<b>Complejidad</b>	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos de hardware</b>						
<b>Restricciones de tiempo de ejecución</b>			1,00	1,11	1,30	1,66
<b>Restricciones de memoria virtual</b>			1,00	1,06	1,21	1,56
<b>Volatilidad de la máquina virtual</b>		0,87	1,00	1,15	1,30	
<b>Tiempo de respuesta</b>		0,87	1,00	1,07	1,15	
<b>Atributos de personal</b>						

<b>Capacidad de análisis</b>	1,46	1,19	1,00	0,86	0,71	
<b>Experiencia en la aplicación</b>	1,29	1,13	1,00	0,91	0,82	
<b>Calidad de los programadores</b>	1,42	1,17	1,00	0,86	0,70	
<b>Experiencia en la máquina virtual</b>	1,21	1,10	1,00	0,90		
<b>Experiencia en el lenguaje</b>	1,14	1,07	1,00	0,95		
<b>Atributos del proyecto</b>						
<b>Técnicas actualizadas de programación</b>	1,24	1,10	1,00	0,91	0,82	
<b>Utilización de herramientas de software</b>	1,24	1,10	1,00	0,91	0,83	
<b>Restricciones de tiempo de desarrollo</b>	1,22	1,08	1,00	1,04	1,10	

**Fuente:** (COCOMO, 2013)

### 2.9.1.2.3 Modelo detallado

Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo. Presenta principalmente dos mejoras respecto al modelo básico e Intermedio.

Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones.

Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.

Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

Para nuestro caso el modelo intermedio será el que usaremos, dado que realiza las estimaciones con bastante precisión.

## **2.10 SEGURIDAD DE LA INFORMACIÓN**

### **2.10.1 Definición de la seguridad informática**

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 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).

### **2.10.2 Estándar ISO/IEC 27000**

Según (Wikipedia, 2015). 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.

FIGURA N° 2. 29 Estructura ISO 27000



Fuente (Wikipedia, 2015).

### 2.10.3 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, 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. Según (Pandini, 2005).

### **2.10.3.1 Seguridad lógica**

Se debe crear un documento sobre la política de seguridad de la información de la empresa, que debe contener los conceptos de seguridad de la información, una estructura para establecer los objetivos y las formas de control, el compromiso de la dirección con la política, entre tantos otros factores.

### **2.10.3.2 Seguridad física**

Los equipos e instalaciones de procesamiento de información crítica o sensible deben mantenerse en áreas seguras, con niveles y controles de acceso apropiados, incluyendo protección contra amenazas físicas y ambientales.

### 2.10.3.3 Seguridad organizativa

Según (GARCÍA, 2017). La organización de la seguridad de la información se puede dar de dos formas: organización interna y organización con respecto a terceros.

- ❖ **Organización interna:** se tiene como objetivo manejar la seguridad de la información dentro de la organización.
- ❖ **Organización con respecto a terceros:** La organización en materia de seguridad de la información debe también considerarse respecto a terceros. El objetivo de esto es mantener la seguridad de la información y los medios de procesamiento de información de la organización que son ingresados, procesados, comunicados a, o manejados por, grupos externos. Para ello se debe comenzar por la identificación de los riesgos relacionados con los grupos externos. Se debe estudiar cómo a raíz de procesos comerciales que involucran a grupos externos se les puede estar otorgando acceso que afecte la seguridad. Esto se puede dar tanto con clientes o con proveedores. Se debe tener especial cuidado respecto a los contratos que se hagan con terceros, para no afectar la seguridad de la información.

### 2.10.4 caja negra y blanca

Es poner en producción el producto en servidores que cumplan con los requerimientos especificados, además de realizar las pruebas necesarias sobre la nueva infraestructura para el acceso de los usuarios. Según [Sommerville, 2005].

#### 2.10.4.1 Pruebas (Diseño de Casos de Prueba)

Para la realización de estas es necesario la creación de casos de prueba especificando la forma de probar el sistema como un todo, [Sommerville, 2005]. Esto incluye:

- ❖ Realizar pruebas de instalación en la plataforma de producción definida.
- ❖ Pruebas de configuración.
- ❖ Pruebas negativas, encontrar debilidades del sistema.
- ❖ Pruebas de tensión o de estrés al no existir recursos suficientes.

#### **2.10.4.2 Pruebas de Caja Negra**

También denominadas pruebas de comportamientos, se centran en los requisitos funcionales del sistema. Esta prueba permite al ingeniero obtener conjunto de condiciones de entrada que ejerciten los requisitos funcionales del sistema. Según [Sommerville, 2005].

Este tipo de pruebas intentan encontrar errores de tipo:

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores de estructuras de datos.
- ❖ Errores de rendimiento.
- ❖ Errores de inicio y fin.

#### **2.10.4.3 Pruebas de Caja Blanca**

Denominada también prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control de diseño para obtenerlos. Según [Sommerville, 2005].

Con estas pruebas se pretenden:

- ❖ Garantizar que se ejecuta al menos una vez todos los caminos independientes de cada módulo.
- ❖ Ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
- ❖ Ejecuten todos los bucles en sus límites.

## CAPITULO III MARCO APLICATIVO

### 3.1 INTRODUCCION

En el presente capitulo se aplicará la metodología UWE para el análisis correspondiente del proyecto en sus 6 fases principales descritos en marco teórico, como también se dará a conocer sobre el desarrollo de modulo del sistema de evaluación en base al análisis de metodología.

En este capítulo III se aplica la metodología, las normas y técnicas que se mencionaron en el anterior capítulo II de desarrollo del sistema de evaluación de docentes en la carrera Comercio Internacional de la Universidad Pública de El Alto.

La carrera se encuentra en el bloque D en el quinto piso, actualmente cuenta con más de 1500 estudiantes inscrito en esta gestión de 2019 y 80 docentes.

### 3.2 INGENIERÍA DE REQUERIMIENTOS

Ingeniería de requerimientos es el conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución.

Las Funciones que un sistema debe realizar se clasifican en tres categorías como se detallan en la tabla.

TABLA N° 3. 1 Categoría de las funciones

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

**Fuente:** (Elaboración propia)

### 3.2.1 Requerimientos funcionales

Los requerimientos funcionales para el modelado del sistema de evaluación de docentes en la carrera Comercio Internacional de la Universidad Pública de El Alto, se detallan a continuación en la tabla.

TABLA N° 3. 2 Requerimientos Funcionales

Ref.	Función	Categoría
R.1.1	Acceder al Sistema por tipos de Usuario(Administrador, docente y estudiante)	Evidente
R.1.2	Registro de Usuario del Sistema	Evidente
R.1.3	Registro de estudiantes	Evidente
R.1.4	Registro de docentes	Evidente
R.1.5	Asignar comité de evaluador	Evidente
R.1.6	Habilitar fecha de evaluación docente	Evidente
R.1.7	Configuración de formulario de evaluación a docente	Evidente
R.1.8	Realizar informe de evaluación docente.	Evidente
R.1.9	Inicio y Cierre de Gestión del sistema	Oculto
R.1.10	Administración de cargos de los usuarios del sistema	Oculto
R.1.11	Migrar datos de estudiante en base Excel.	Evidente
R.1.12	Realizar un informe la cantidad de docente y su asignatura.	Evidente

**Fuente:** (Elaboración Propia)

### 3.2.2 Requerimientos no funcionales

TABLA N° 3. 3 Requerimientos No Funcionales

Ref.	Función	Categoría
R.1.1	El sistema debe visualizarse y funcionar correctamente en cualquier Navegador, Internet Explorer, Opera, Mozilla, Chrome.	Evidente
R.1.2	El sistema no debe tardar más de diez segundos en mostrar los resultados de una búsqueda. Si se supera este plazo, el sistema detiene la búsqueda y muestra los resultados encontrados.	Superflua
R 1.3	El sistema debe tener seguridad en el acceso a la información del sistema.	Evidente

**Fuente:** (Elaboración Propia).

### 3.3 FASE DE LA METODOLOGÍA UWE.

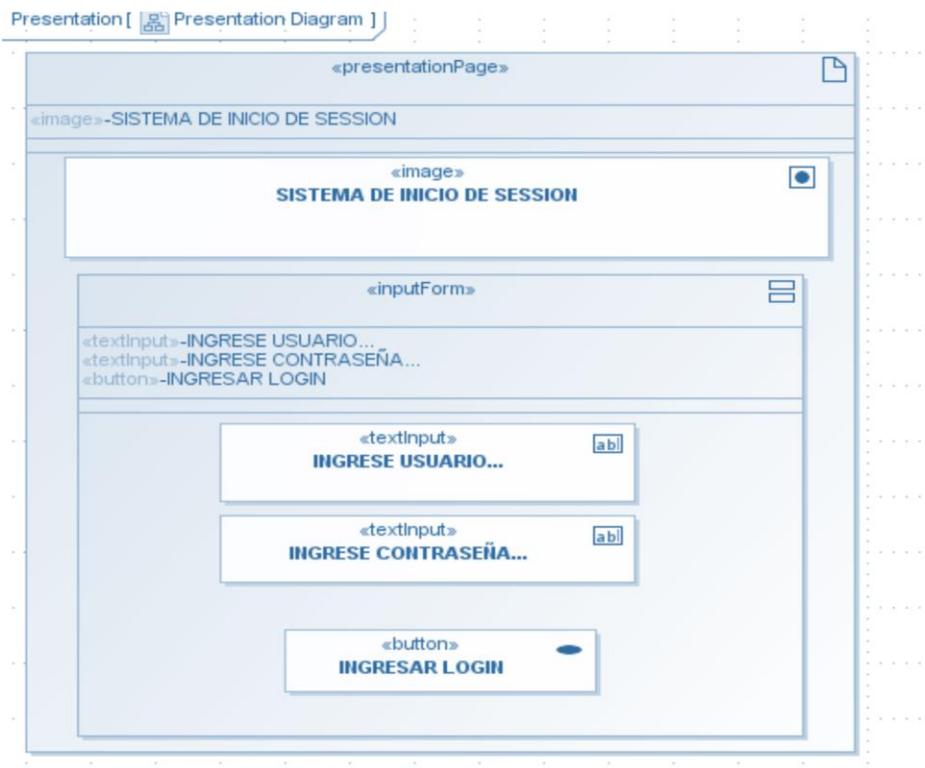
#### 3.3.1 Fase captura, análisis y especificación de requisitos.

En este análisis y especificaciones de requerimientos sobre el sistema de evaluación de docentes, ya mencionadas anteriormente en el índice 3.2.1 obtención de requisitos, sobre los requerimientos específicos para el desarrollo de este presente proyecto.

#### 3.3.2 Fase 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, la estructura que debe darse a la aplicación web.

FIGURA N° 3. 1 Diseño del sistema login



Fuente (Elaboración propia).

FIGURA N° 3. 2 Diseño del sistema

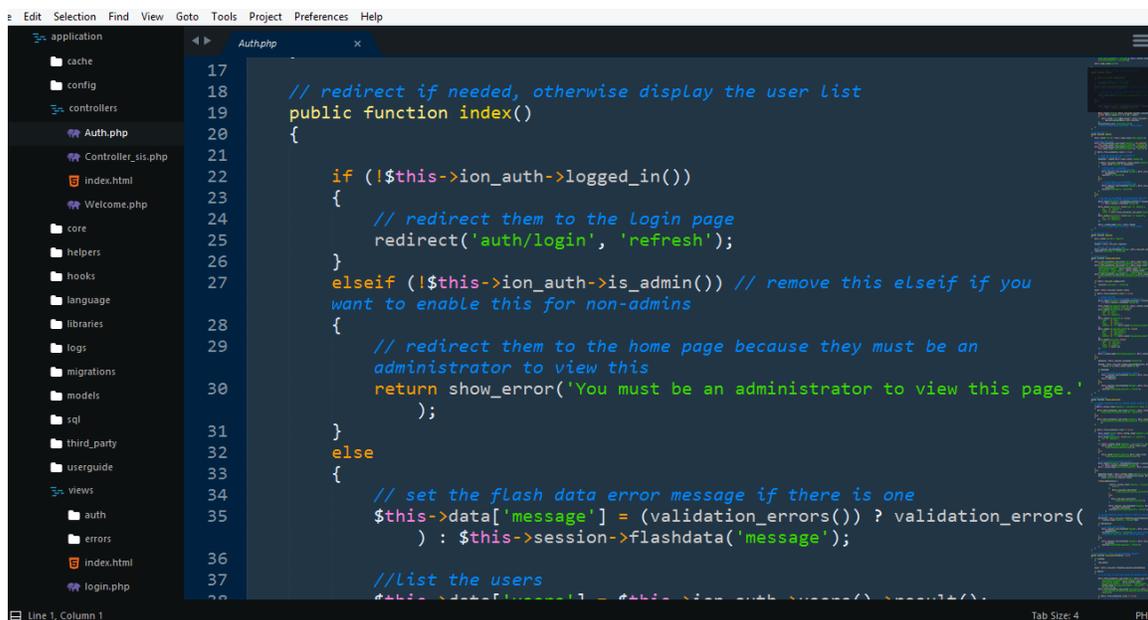


Fuente (Elaboración propia).

### 3.3.3 Fase codificación del software.

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación PHP y el manejo de framework de codeigniter, diseño de front-End CSS3, Ajax, JQuery, Javascript, Html5, Bootstrap y otros.

FIGURA N° 3. 3 Codificación del software



The image shows a screenshot of a code editor with a dark theme. The left sidebar displays a file explorer for an 'application' directory, listing folders like 'cache', 'config', 'controllers', and 'views', along with files like 'Auth.php', 'Controller\_sis.php', 'index.html', and 'Welcome.php'. The main editor window is open to 'Auth.php' and shows the following PHP code:

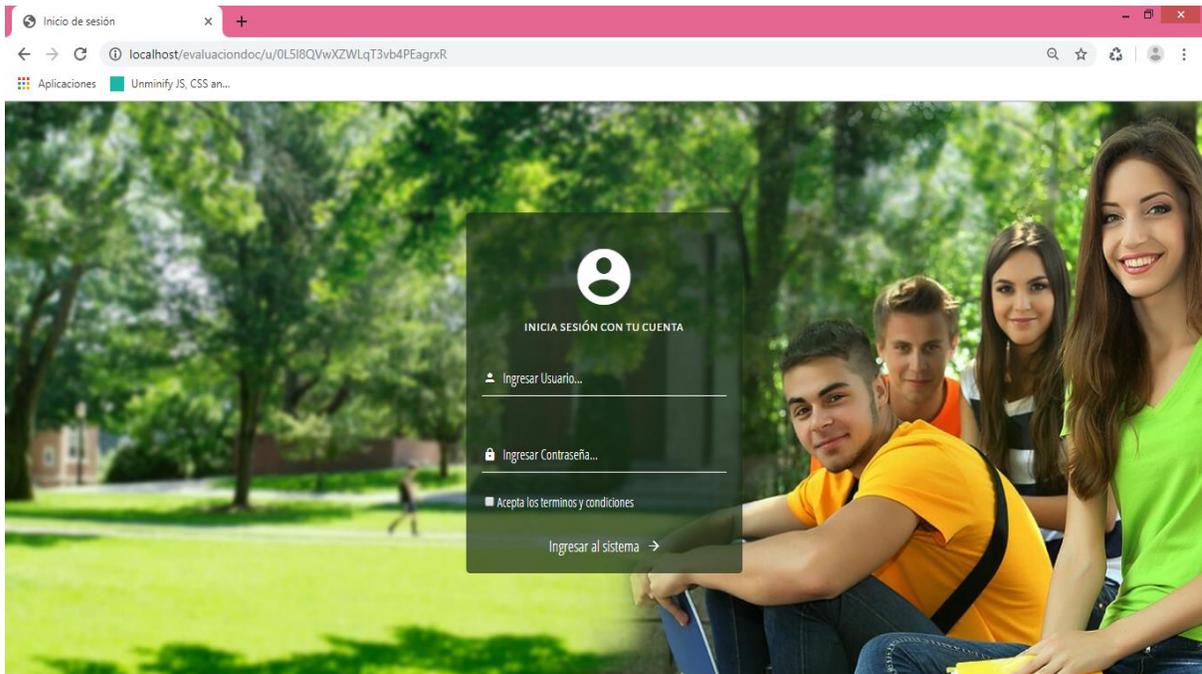
```
17 // redirect if needed, otherwise display the user list
18 public function index()
19 {
20
21     if (!$this->ion_auth->logged_in())
22     {
23         // redirect them to the login page
24         redirect('auth/login', 'refresh');
25     }
26     elseif (!$this->ion_auth->is_admin()) // remove this elseif if you
27     // want to enable this for non-admins
28     {
29         // redirect them to the home page because they must be an
30         // administrator to view this
31         return show_error('You must be an administrator to view this page.'
32         );
33     }
34     else
35     {
36         // set the flash data error message if there is one
37         $this->data['message'] = (validation_errors()) ? validation_errors(
38         ) : $this->session->flashdata('message');
39     }
40     //List the users
41     $this->data['users'] = $this->ion_auth->users();
42 }
```

Fuente: (Elaboración propia).

### 3.3.4 Fase pruebas.

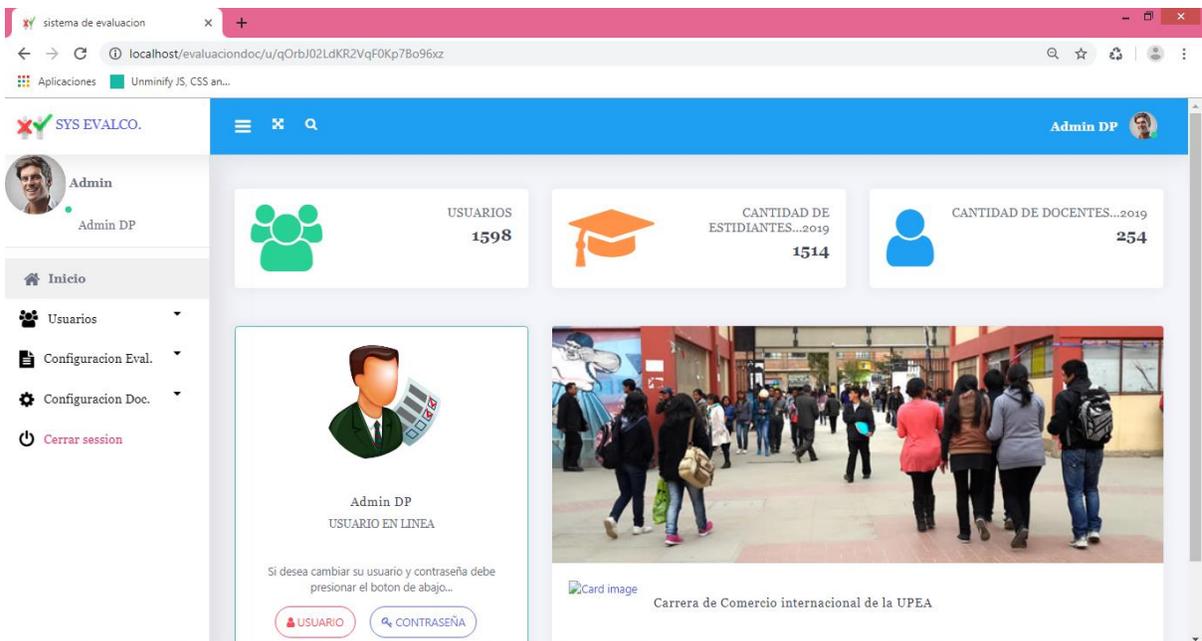
Las pruebas se utilizan para asegurar el correcto funcionamiento de los distintos módulos, los formularios y secciones de código del sistema de evaluación de docentes.

FIGURA N° 3. 4 Panel de inicio de sesión



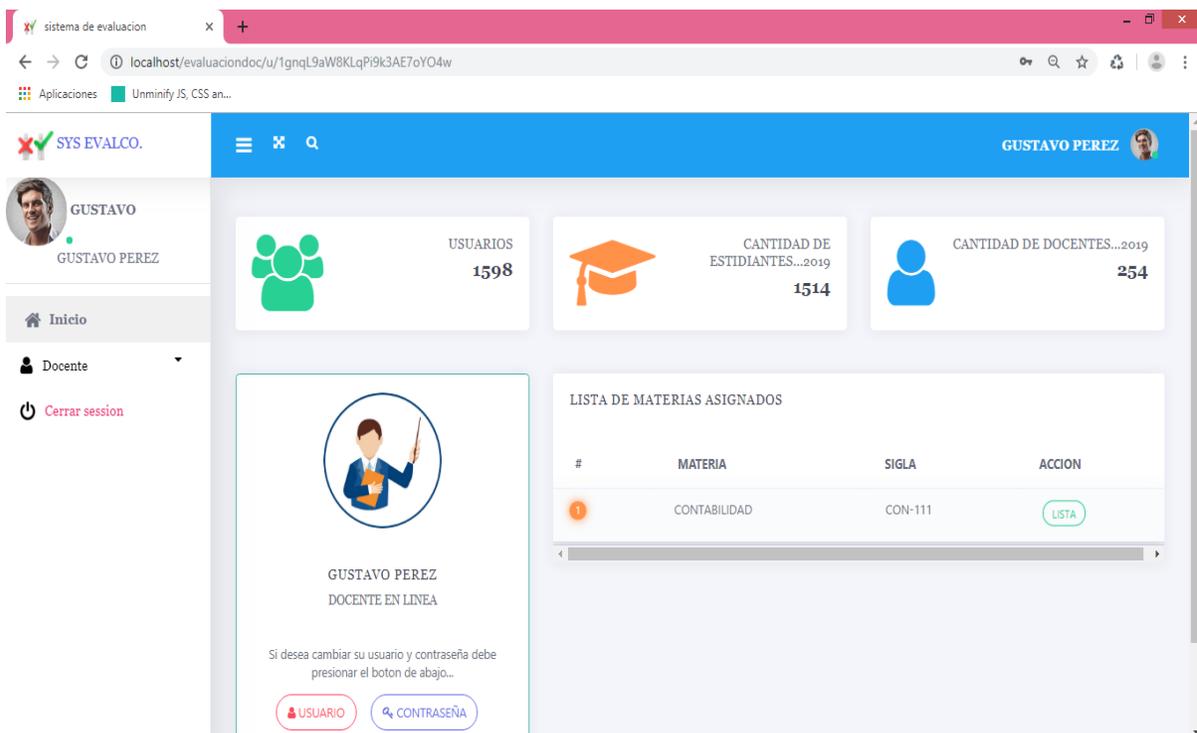
Fuente: (Elaboración propia).

FIGURA N° 3. 5 Panel de kardex



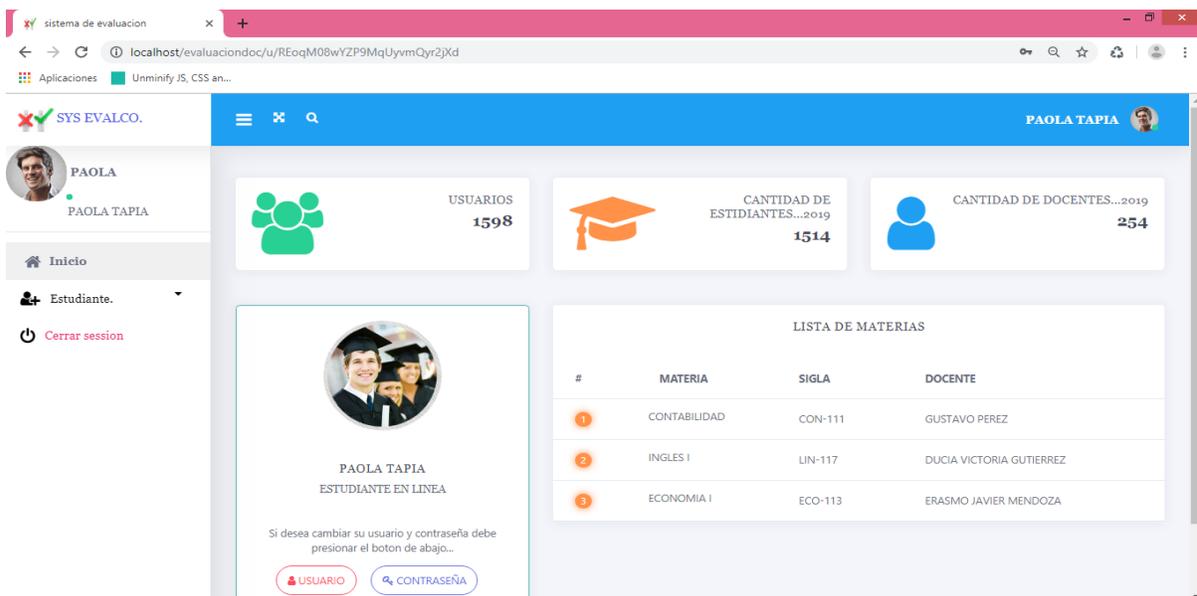
Fuente: (Elaboración propia).

FIGURA N° 3. 6 Panel de docente



Fuente: (Elaboración propia).

FIGURA N° 3. 7 Panel de estudiante



Fuente: (Elaboración propia).

localhost/evaluaciondoc/u/zybW2KEyzf3XhjAH2mVnKaNBd

**SYS EVALCO.** Admin DP

Admin DP  
Admin DP

Inicio

Usuarios

Configuración Eval.

Configuración Doc.

- » Opción Eval.
- » Pregunta Eval
- » Formulario Eval. Doc.
- » Administración Doc.
- » Docente asignatura

Cerrar sesión

APellidos : PEREZ MAMANI  
FECHA NAC. : 1974-05-14

FECHA : 26-10-2019

### INFORME DE EVALUACION

DESCRIPCION	PONDERACION	CANTIDAD EST	ESTU
preparacion pedagogica (didactica y metodologia)	ponderacion max. 20 pts.	0	0
Dominio de tema y /o materia	Ponderacion max. 20 pts.	0	0
Cumplimiento del plan de trabajo	ponderacion max. 10 pts.	0	0
Puntualidad en los horarios	ponderacion max. 10 pts.	0	0
Objetividad en la evaluacion y calificaciones (en base a criterio academico)	ponderacion max. 10 pts.	0	0

### INFORME FINAL

#	DESCRIPCION	PONDERACION	CANTIDAD
1	preparacion pedagogica (didactica y metodologia)	ponderacion max. 20 pts.	0
2	Dominio de tema y /o materia	Ponderacion max. 20 pts.	0
3	Cumplimiento del plan de trabajo	ponderacion max. 10 pts.	0
4	Puntualidad en los horarios	ponderacion max. 10 pts.	0
5	Objetividad en la evaluacion y calificaciones (en base a criterio academico)	ponderacion max. 10 pts.	0
TOTAL (70%) :			0

Reporte de informe de evaluación al docente.

localhost/evaluaciondoc/u/p5qbKod7RIGMIRbHGYWLkWDYd

**Universidad Pública de El Alto**  
Creada por Ley N° 2115 del 01 de Septiembre de 2009 y Anterior por Ley N° 2756 del 12 de Noviembre de 2003

### INFORME DE EVALUACION DOCENTE

**DATOS DEL DOCENTE**

CARRERA : COMERCIO INTERNACIONAL  
ITEM NOMBRAMIENTO DOC. : CONSULTOR EN COMERCIO INTERNACIONAL  
ASIGNATURA : CONTABILIDAD  
CATEGORIA DOC. : INVITADO ANTRIGO  
SIGLA-ASIGNATURA : CON-III  
NOMBRE DEL DOC. : GUSTAVO PEREZ MAMANI  
PARALELO : IN-C  
SEDE : VILLA ESPERANZA

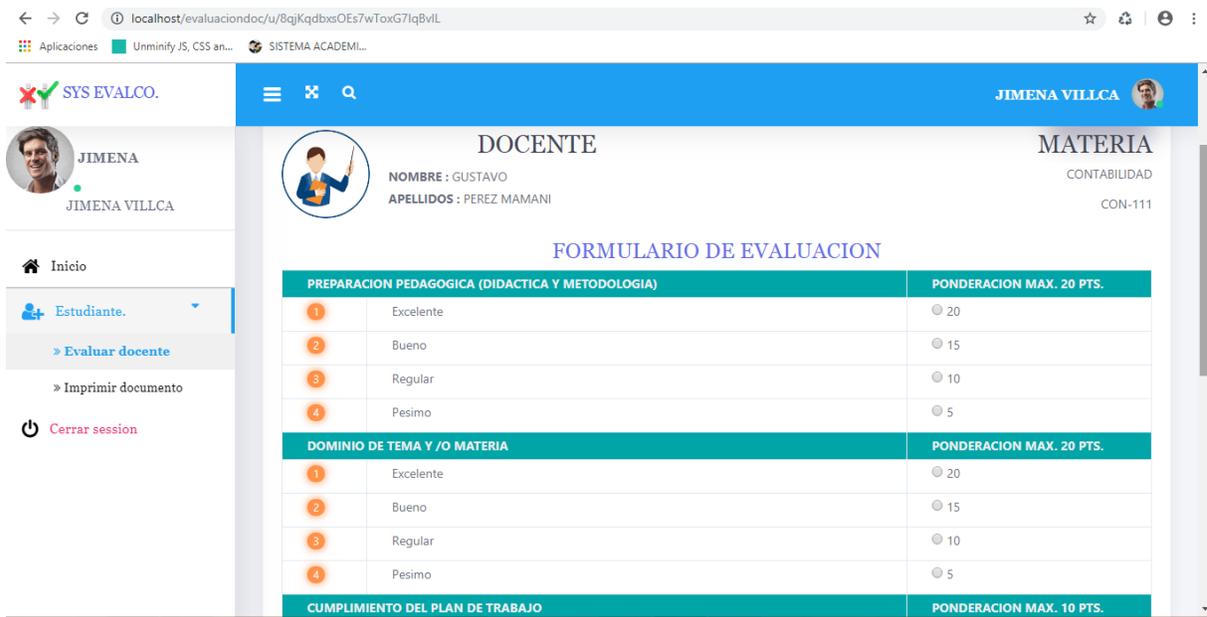
### INFORME DE EVALUACION

DESCRIPCION	PONDERACION	CANTIDAD EST.	CANTIDAD EVAL.
preparacion pedagogica (didactica y metodologia)	ponderacion max. 20 pts.	20	1
Dominio de tema y /o materia	Ponderacion max. 20 pts.	20	1
Cumplimiento del plan de trabajo	ponderacion max. 10 pts.	10	1
Puntualidad en los horarios	ponderacion max. 10 pts.	10	1
Objetividad en la evaluacion y calificaciones (en base a criterio academico)	ponderacion max. 10 pts.	10	1

### INFORME FINAL DE EVALUACION

#	DESCRIPCION	PONDERACION	CANTIDAD
1	preparacion pedagogica (didactica y metodologia)	ponderacion max. 20 pts.	20
2	Dominio de tema y /o materia	Ponderacion max. 20 pts.	20
3	Cumplimiento del plan de trabajo	ponderacion max. 10 pts.	7
4	Puntualidad en los horarios	ponderacion max. 10 pts.	10
5	Objetividad en la evaluacion y calificaciones (en base a criterio academico)	ponderacion max. 10 pts.	10
TOTAL (70%) :			67

El presente panel es representación del formulario de evaluación al docente con diferentes parámetros.



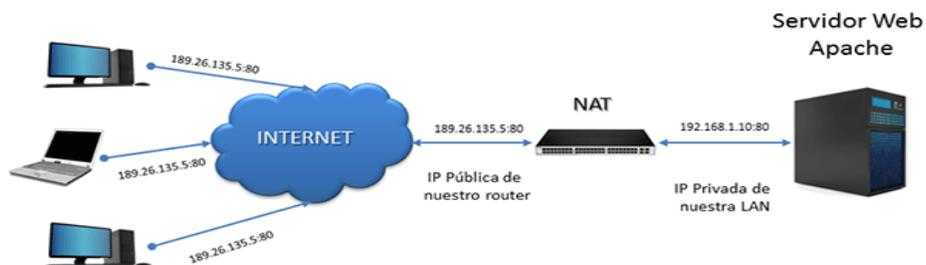
Fuente: (Elaboración propia).

### 3.3.5 Fase Instalación o fase de Implementación.

Para esta fase de implementación del sistema desarrollado son transferidos apropiadamente a un servidor donde se alojaran el sistema por el cual donde se solicitara un dominio para el acceso al sistema, todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, al criterio del usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

FIGURA Nº 3. 8 La Instalación o fase de Implementación



Fuente (Busó, 2015)

### **3.3.6 Fase 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.

El mantenimiento consta la implementación de algunos módulos o mejora de ciertos interfaces grafica del sistema al criterio del usuario.

## **3.4 APLICACION DEL MODELO UWE**

### **3.4.1 Modelo de casos de uso**

En el presente numeral se plasmara el análisis de requerimientos del sistema mediante el diseño de casos de uso mismos expresados en el comportamiento del sistema frente a las acciones de los actores del mismo, funcionalidades del sistema y además elementos que permiten la abstracción del problema.

#### **3.4.1.1 Análisis de requerimientos**

En este punto se plasma el análisis de requerimiento del sistema mediante el diseño de diagrama de caso de uso comercial el cual describe el comportamiento del sistema de evaluación de docentes en la carrera Comercio Internacional de la Universidad Pública de El Alto.

El diagrama de caso de uso permite que describa el comportamiento del sistema frente a las acciones de los diferentes actores, así como las funcionalidades del sistema.

#### **3.4.1.2 Diagrama de caso de uso comercial.**

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 diagramas de casos de uso a continuación.

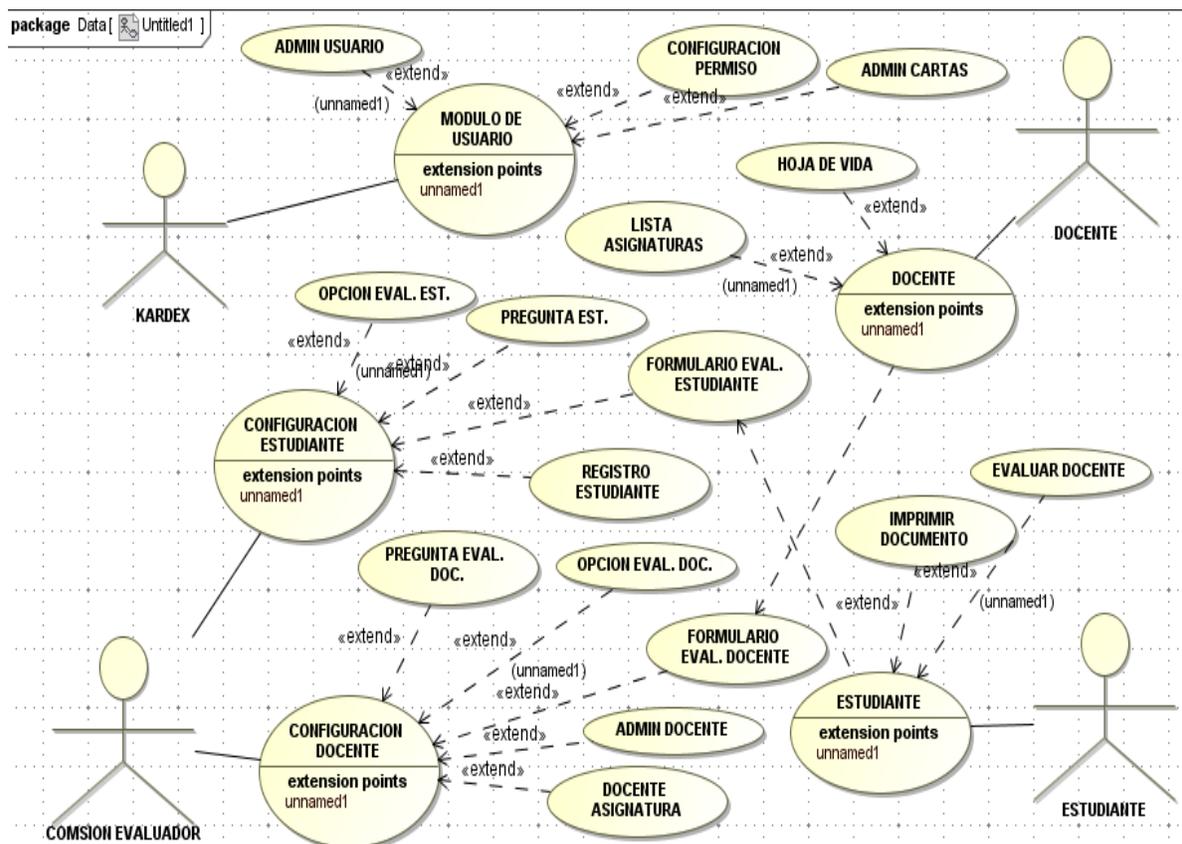
- ❖ Modelo de caso de uso: General del Sistema
- ❖ Diagrama de caso de uso: kardex

- ❖ Diagrama de caso de uso: comisión evaluador
- ❖ Diagrama de caso de uso: docente.
- ❖ Diagrama de caso de uso: estudiante

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

### 3.4.1.2.1 Modelo de caso de uso: General del Sistema

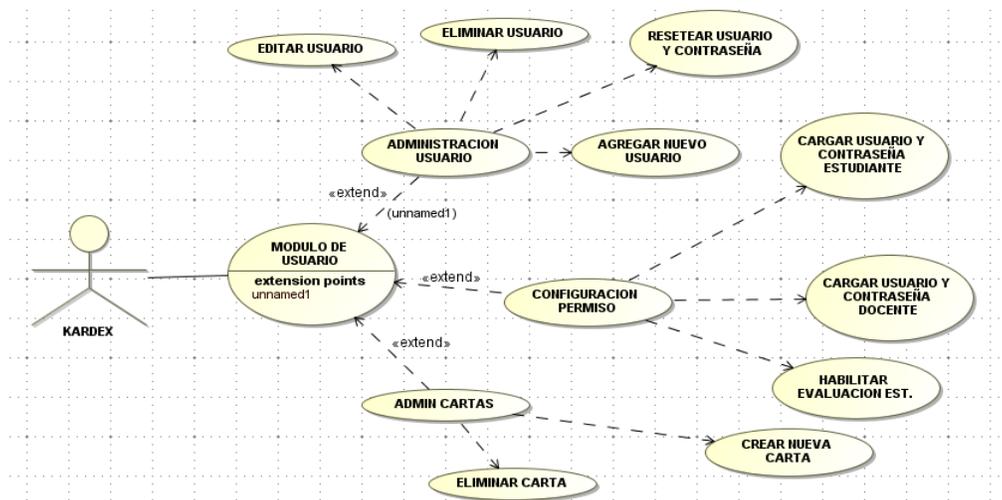
FIGURA Nº 3. 9 Modelo de caso de uso: General del Sistema



Fuente (Elaboración propia).

### 3.4.1.2.2 Diagrama de caso de uso: kardex

FIGURA N° 3. 10 Diagrama de caso de uso: kardex.



Fuente (Elaboración propia).

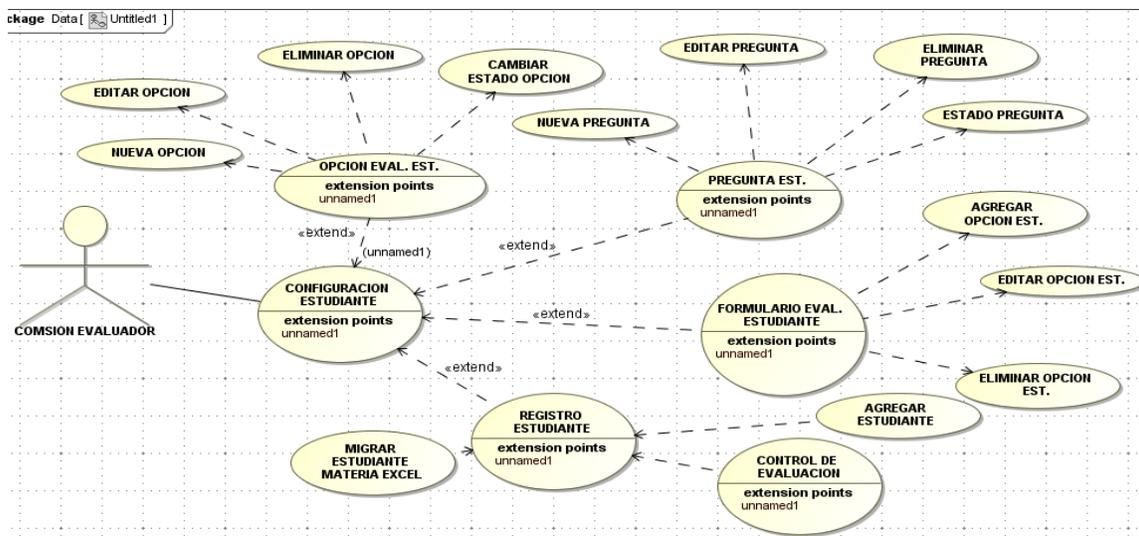
TABLA N° 3. 4 Diagrama de caso de uso: kardex

Caso de uso: kardex	
Actores	Administrador de kardex.
Tipo	Primario esencial.
Descripción	<p>El administrador de kardex tiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administrar usuario, crear un nuevo usuario, editar usuario, resetear usuario, contraseña del usuario, realizar búsqueda de datos en tiempo real y por ultimo eliminar usuario.</li> <li>❖ Configuración de permisos en este módulo el kardista tiene las opciones de cargar usuario y contraseña de los estudiantes, cargar usuario y contraseña de los docentes y por ultimo habilitar fecha de evaluación a docentes.</li> <li>❖ Módulo de administración de cartas nos permite realizar un nuevo registro de carta y eliminar registro</li> </ul>

Fuente (Elaboración propia).

### 3.4.1.2.3 Diagrama de caso de uso: comisión evaluador

FIGURA N° 3. 11 Diagrama de caso de uso: comisión evaluador configuración estudiante



Fuente (Elaboración propia).

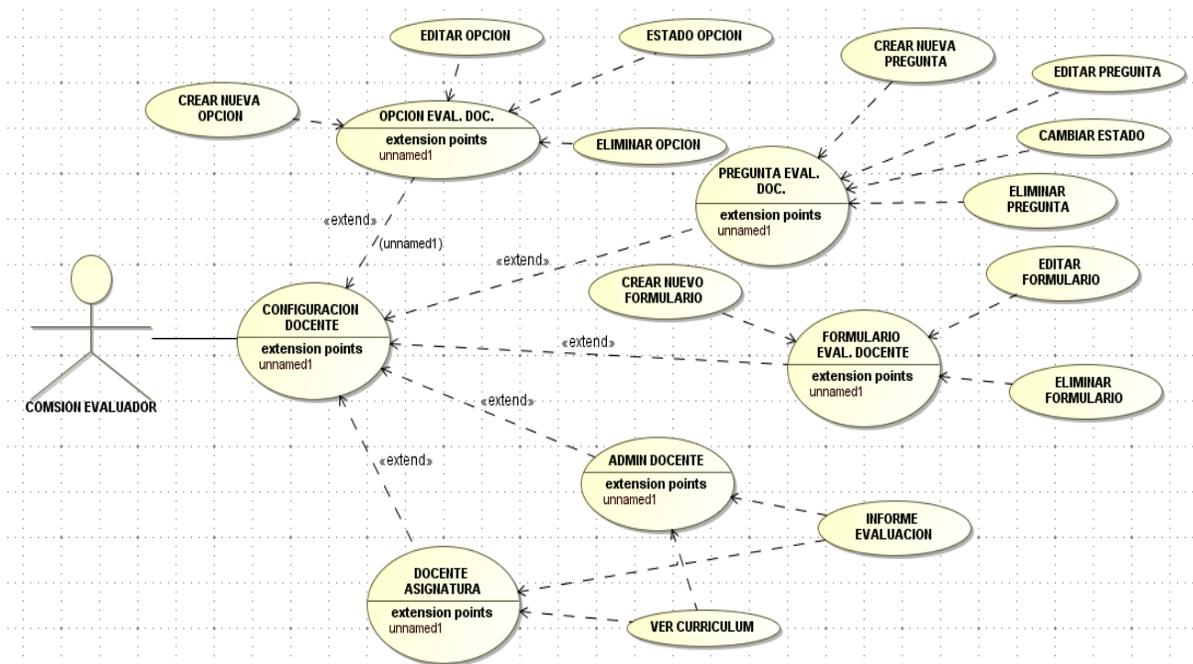
TABLA N° 3. 5 Diagrama de caso de uso: comisión evaluador configuración estudiante

Caso de uso: comisión evaluador configuraciones estudiante	
Actores	Comisión evaluador configuración estudiante.
Tipo	Primario esencial.
Descripción	<p>El comisión evaluador módulo de configuración estudiante tiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administrar opciones, crear una nueva opción de pregunta, editar opción, cambiar estado y por ultimo eliminar opción.</li> <li>❖ Módulo de administrar opciones ponderación de pregunta, crear una nueva opción de pregunta, editar opción, cambiar estado y por ultimo eliminar opción.</li> <li>❖ Módulo de administrar configuración de formulario</li> </ul>

	<p>de evaluación al docente, crear una nueva configuración de formulario, editar configuración de formulario y por ultimo eliminar configuración de formulario.</p> <ul style="list-style-type: none"> <li>❖ Módulo de registro de estudiante, en este módulo nos permite migrar estudiante del periodo en base formato Excel que se puede descargar en este panel o en forma única ingresando a la asignatura del docente en base manual, de esa manera podremos también monitorear la evaluación al docente.</li> </ul>
--	---

Fuente (Elaboración propia).

FIGURA N° 3. 12 Diagrama de caso de uso: comisión evaluador configuración docente



Fuente (Elaboración propia).

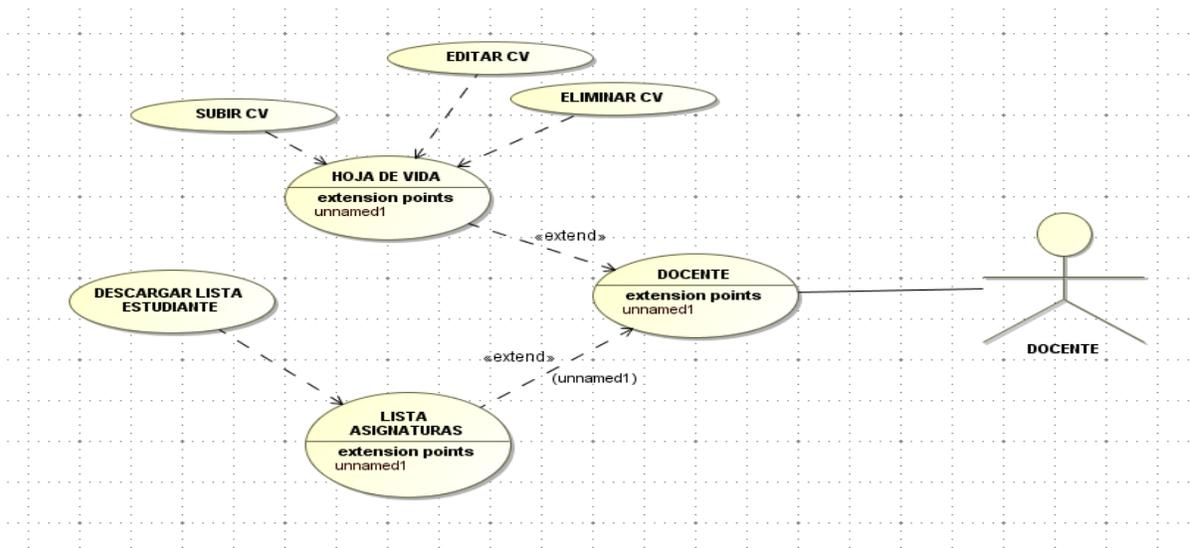
TABLA N° 3. 6 Diagrama de caso de uso: comisión evaluador

<b>Caso de uso: comisión evaluador configuración docente</b>	
Actores	Comisión evaluador configuración docente.
Tipo	Primario esencial.
Descripción	<p>El comisión evaluador módulo de configuración docente tiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>❖ Módulo de administrar opciones docente, crear una nueva opción de pregunta, editar opción, cambiar estado y por ultimo eliminar opción.</li> <li>❖ Módulo de administrar opciones ponderación de pregunta docente, crear una nueva opción de pregunta, editar opción, cambiar estado y por ultimo eliminar opción.</li> <li>❖ Módulo de administrar configuración de formulario de curriculum del docente, crear una nueva configuración de formulario, editar configuración de formulario y por ultimo eliminar configuración de formulario.</li> <li>❖ Módulo de administrar de docente y asignación docente nos permite brindar un informe final de evaluación al docente de los estudiante de la misma manera poder ver el curriculum de los libros realizados.</li> </ul>

Fuente (Elaboración propia).

### 3.4.1.2.4 Diagrama de caso de uso: docente

FIGURA N° 3. 13 Diagrama de caso de uso: docente



Fuente (Elaboración propia).

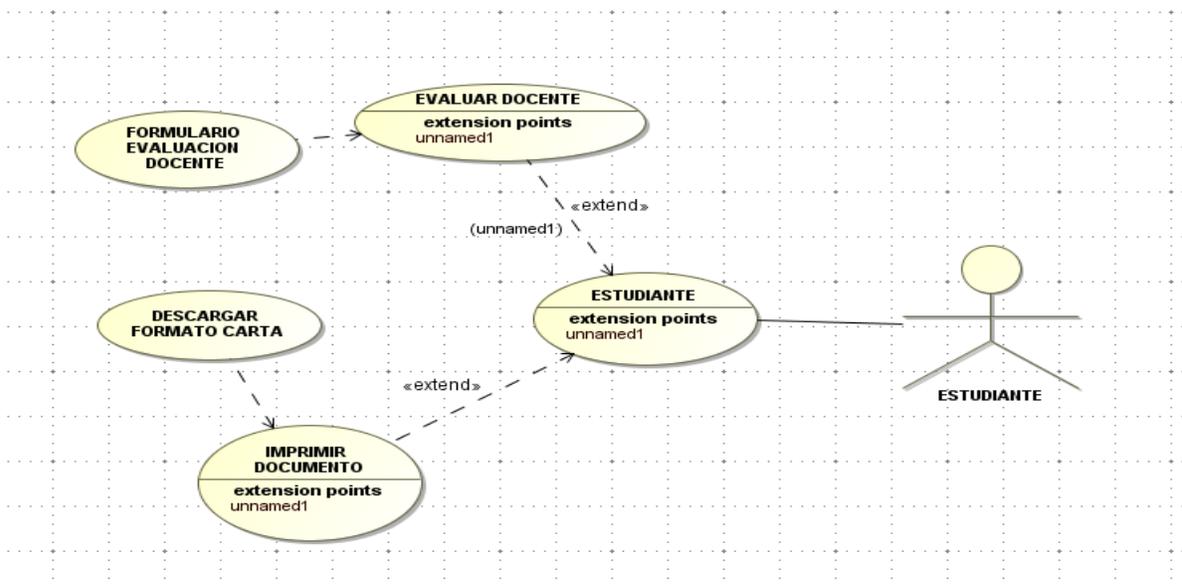
TABLA N° 3. 7 Diagrama de caso de uso: docente

Caso de uso: docente	
Actores	Usuario docente.
Tipo	Primario esencial.
Descripción	<p>El Usuario docente tiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>❖ Módulo de hoja de vida donde puede, subir un nuevo libre con su depósito legal que ha ya realizado, editar y por ultimo eliminar.</li> <li>❖ En este módulo nos lista las asignaturas asignadas y poder descargar la lista de los estudiantes en formato Excel.</li> </ul>

Fuente (Elaboración propia).

### 3.4.1.2.5 Diagrama de caso de uso: estudiante

FIGURA N° 3. 14 Diagrama de caso de uso: estudiante



Fuente (Elaboración propia).

TABLA N° 3. 8 Diagrama de caso de uso: estudiante

Caso de uso: estudiante	
Actores	Usuario estudiante
Tipo	Primario esencial.
Descripción	<p>El usuario estudiante tiene las siguientes opciones:</p> <ul style="list-style-type: none"> <li>❖ Módulo de evaluar estudiante nos lista las asignaturas asignadas y su respectivo nombre de docente y sigla de asignatura y nos enlaza a un formulario de evaluación de docente.</li> <li>❖ Módulo de carta solo nos permite descargar un formato de carta para cualquier tipo de trámite que realiza en la carrera.</li> </ul>

Fuente (Elaboración propia).

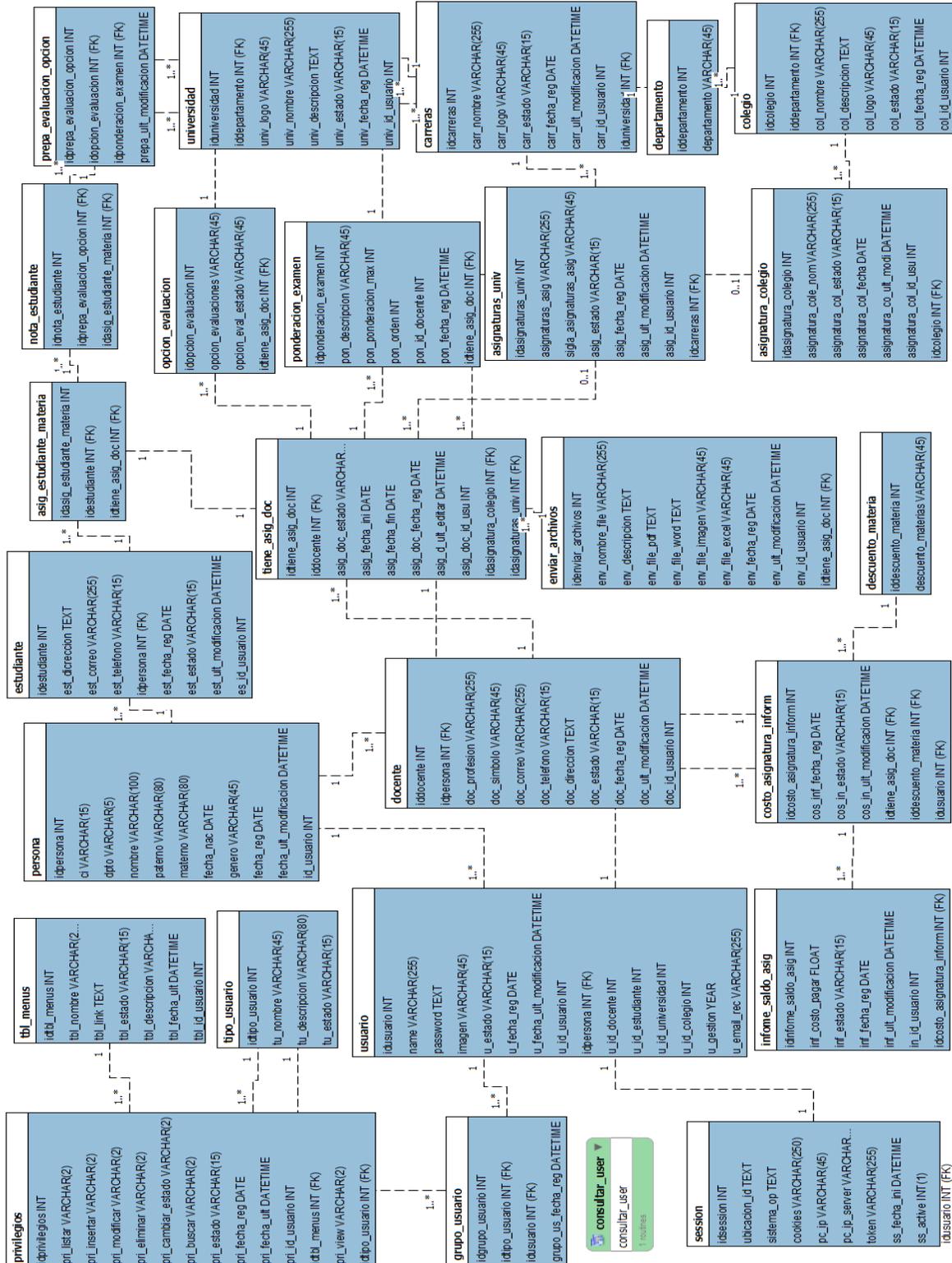
### **3.4.2 Modelo contenido**

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 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. Esto se representa mediante un diagrama de clases UML como se muestra en la Figura.

FIGURA N° 3. 15 Modelo Contenido



Fuente (Elaboración propia).

### 3.4.3 Modelo de navegación

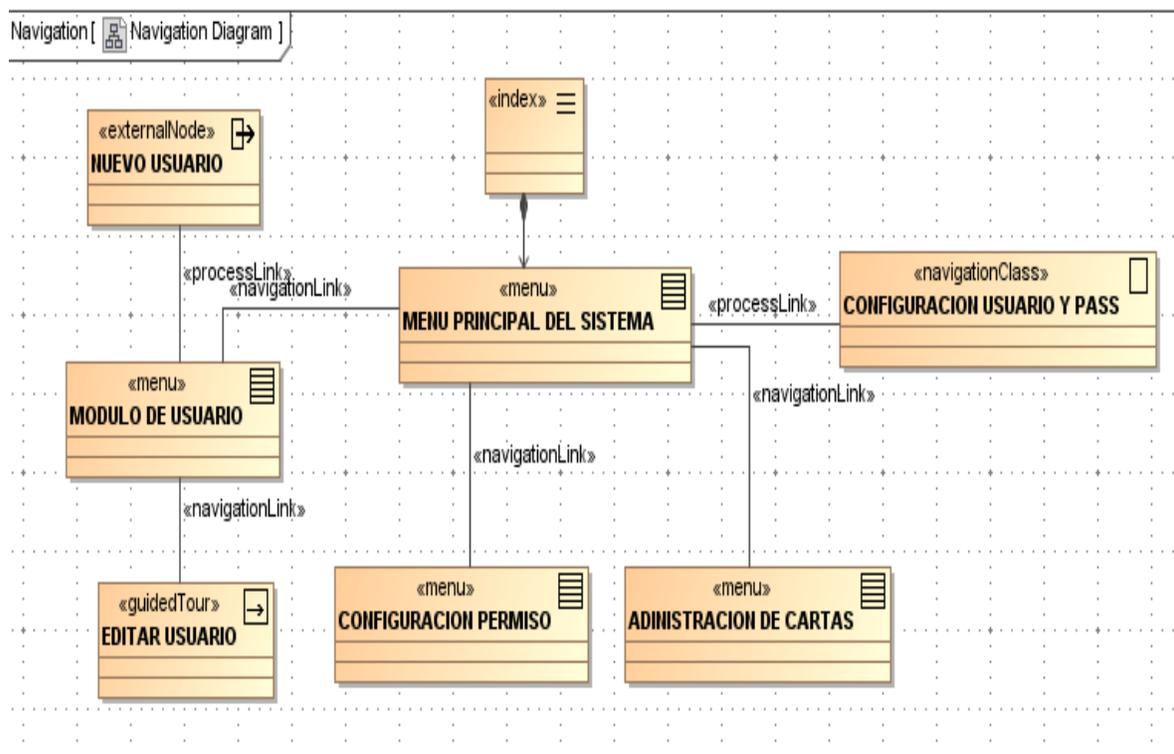
A continuación, se realiza el modelamiento de modelo de navegación, donde se puede apreciar cómo interactúan los usuarios en el modelo de navegación en cada clase.

- ❖ Modelo De Navegación: kardex.
- ❖ Modelo De Navegación: comisión evaluador.
- ❖ Modelo De Navegación: docente.
- ❖ Modelo De Navegación: estudiante.

Estos tienen por cometido 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 se muestra en la Figura.

#### 3.4.3.1 Modelo de navegación: kardex

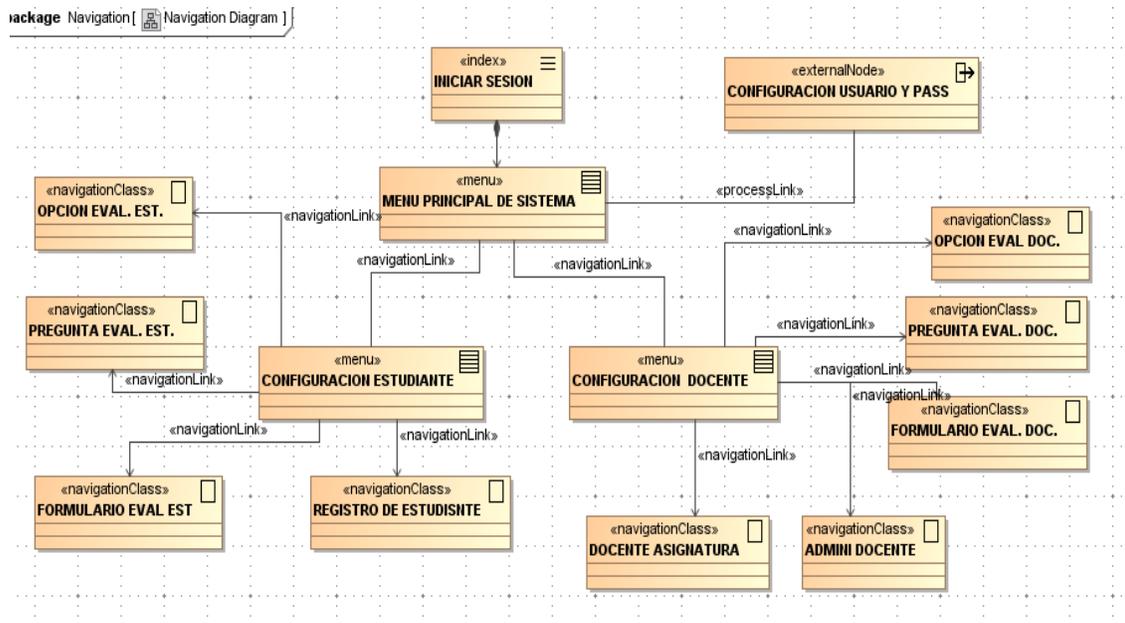
FIGURA N° 3. 16 Modelo De Navegación: kardex.



Fuente (Elaboración propia).

### 3.4.3.2 Modelo de navegación: comisión evaluador

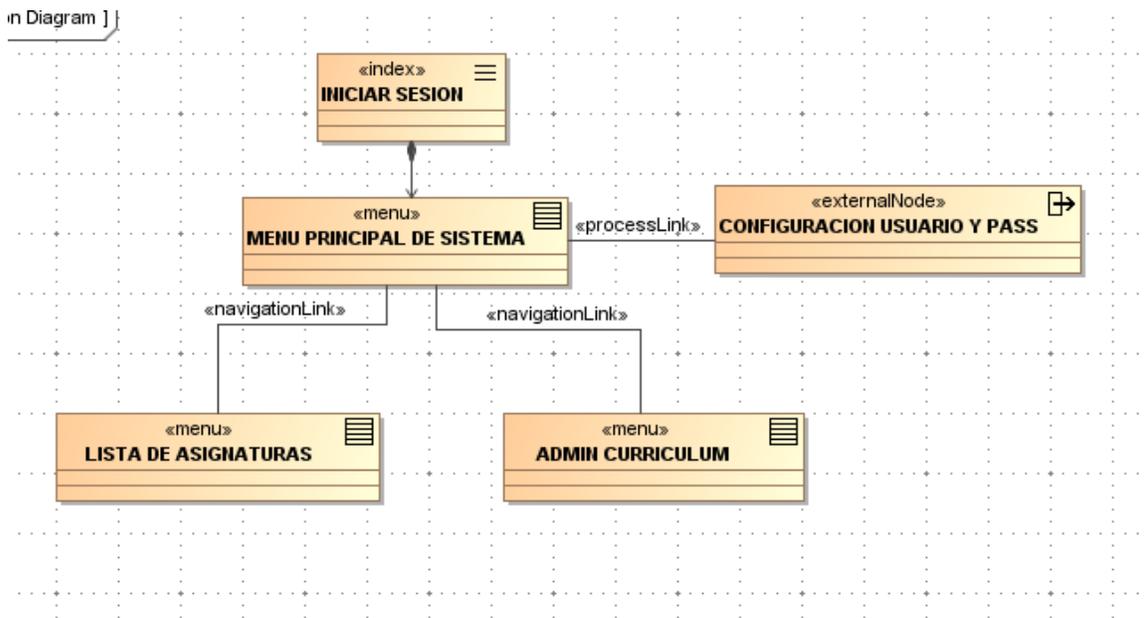
FIGURA N° 3. 17 Modelo De Navegación: comisión evaluador.



Fuente (Elaboración propia).

### 3.4.3.3 Modelo de navegación: docente

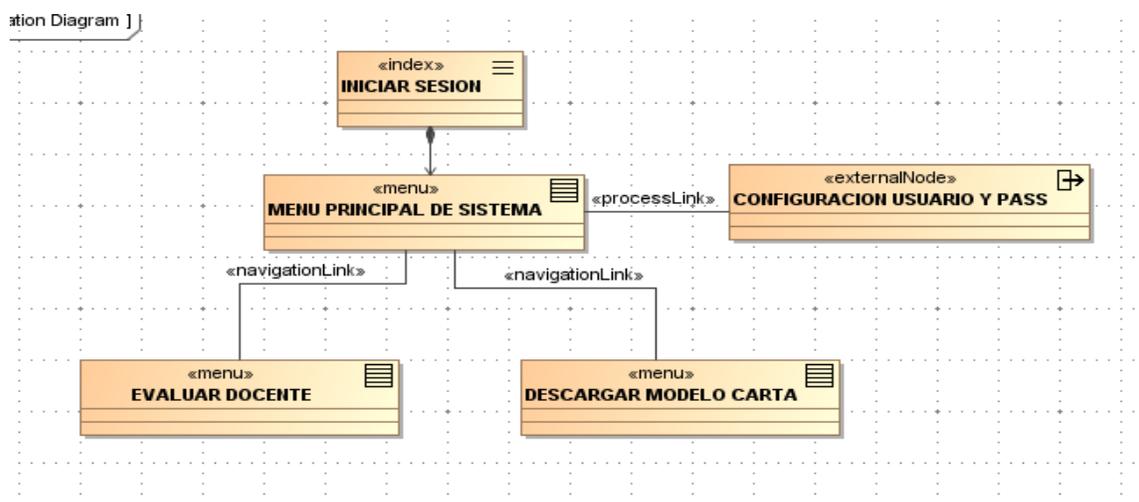
FIGURA N° 3. 18 Modelo De Navegación: docente.



Fuente (Elaboración propia).

### 3.4.3.4 Modelo de navegación: estudiante

FIGURA N° 3. 19 Modelo De Navegación: estudiante

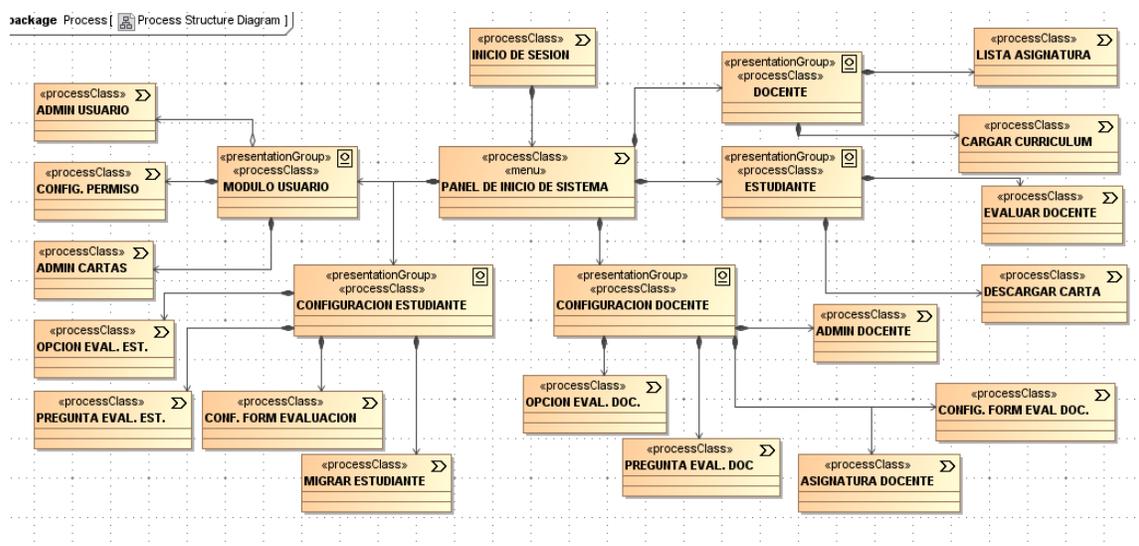


Fuente (Elaboración propia).

### 3.4.4 Modelo de estructura de proceso.

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

FIGURA N° 3. 20 Modelo de estructura de proceso del sistema



Fuente (Elaboración propia).

### 3.4.5 Modelo de presentación

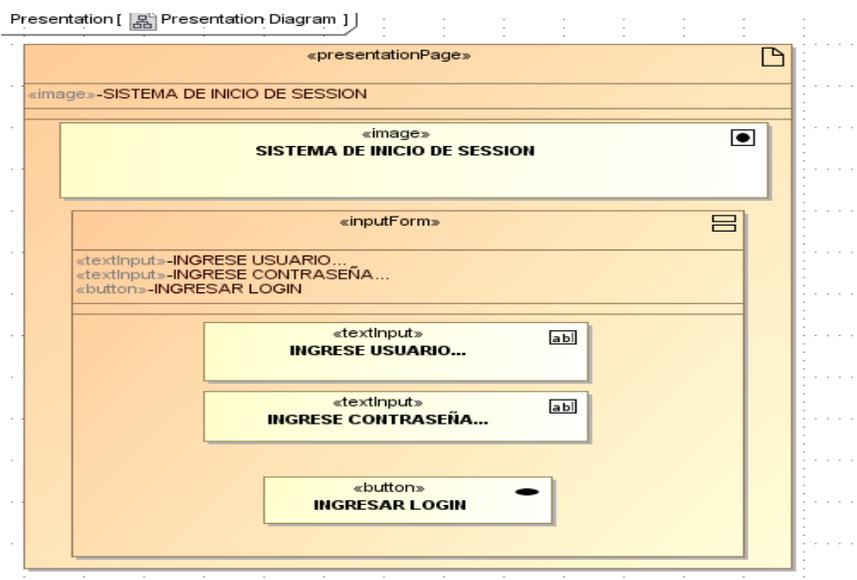
A continuación se muestran los modelos de presentación del sistema de evaluación de docentes en la carrera Comercio Internacional de la Universidad Pública de El Alto, según la metodología UWE que propone para la construcción de las páginas en forma de bosquejos derivadas donde los usuarios podrán acceder al sistema mostrando los menús correspondientes según el tipo de usuario.

- ❖ Modelo de Presentación: kardex.
- ❖ Modelo de Presentación: comisión evaluador.
- ❖ Modelo de Presentación: docente.
- ❖ Modelo de Presentación: estudiante.

El objetivo es ilustrar los vínculos lógicos de modelo de presentación entre clases. Las propiedades pueden anidarse, por ejemplo, cada contacto cubre diferentes textos y botones. Ello significa, acciones que realizan el sistema que para cada acción realizada correspondiente dirección de enlaces y los correspondientes de distintos campos que se visualizan.

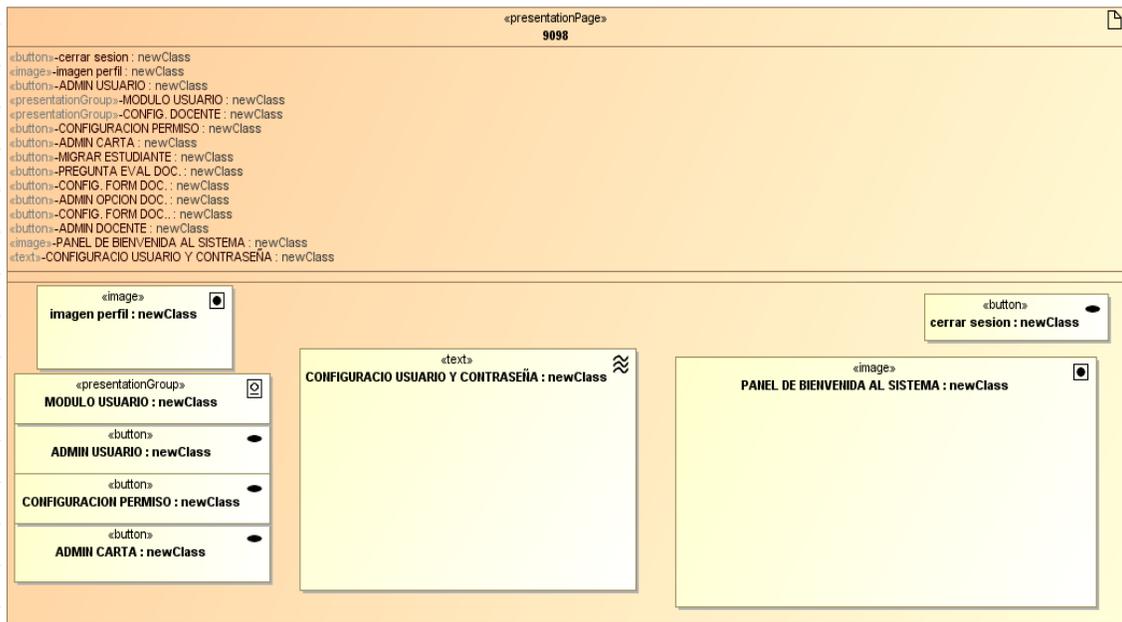
#### 3.4.5.1 Modelo de presentación: kardex

FIGURA N° 3. 21 Modelo de Presentación: Principal login kardex



Fuente (Elaboración propia).

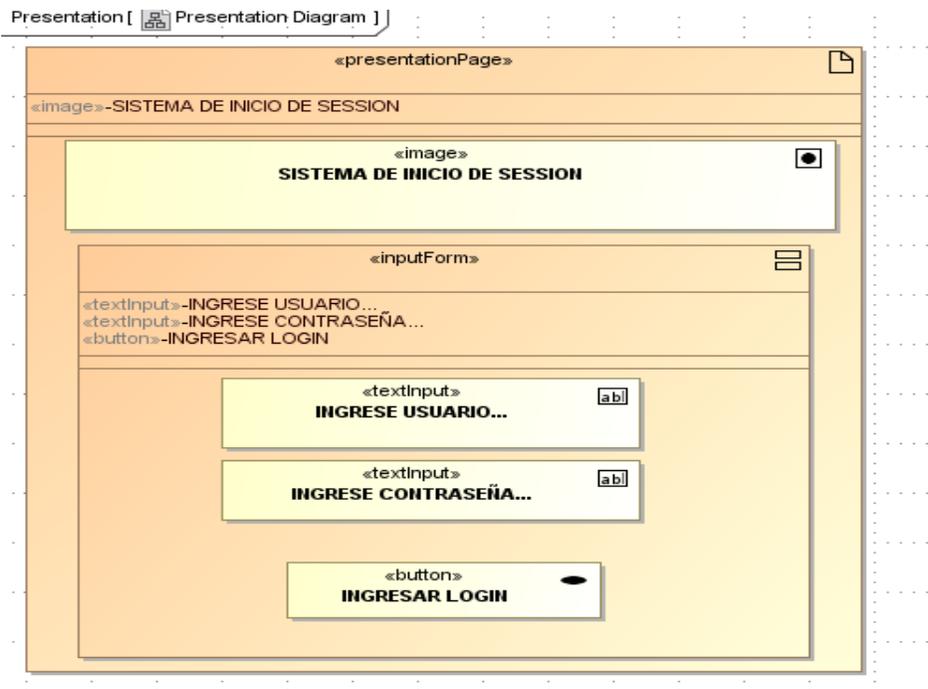
FIGURA N° 3. 22 Modelo de Presentación: kardex



Fuente (Elaboración propia).

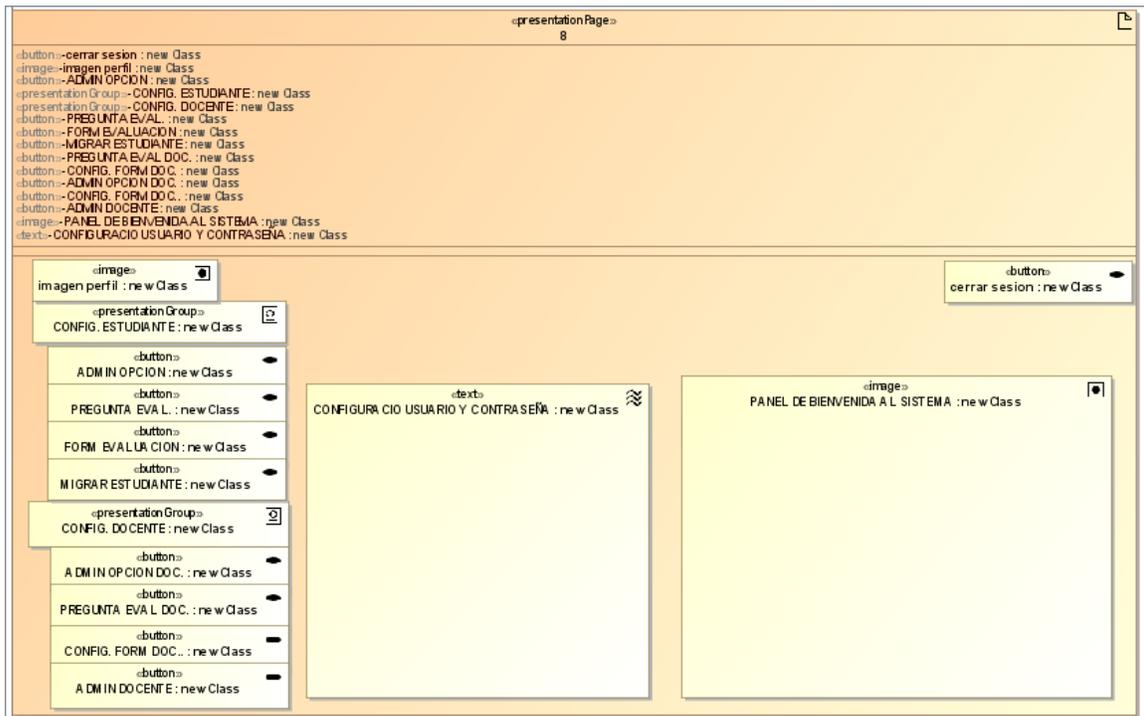
### 3.4.5.2 Modelo de presentación: comisión evaluador

FIGURA N° 3. 23 Modelo de Presentación: Principal login comisión evaluar



Fuente (Elaboración propia).

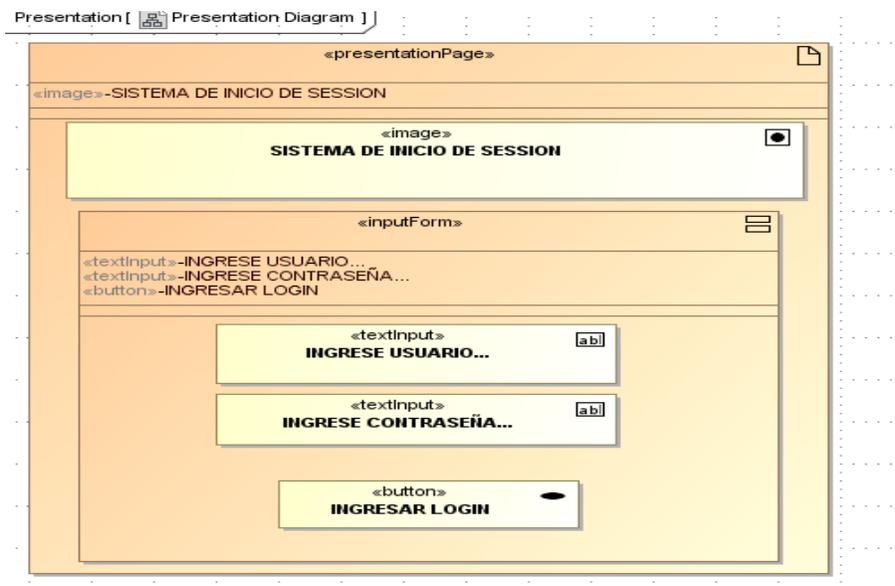
FIGURA N° 3. 24 Modelo de Presentación: comisión evaluador



Fuente (Elaboración propia).

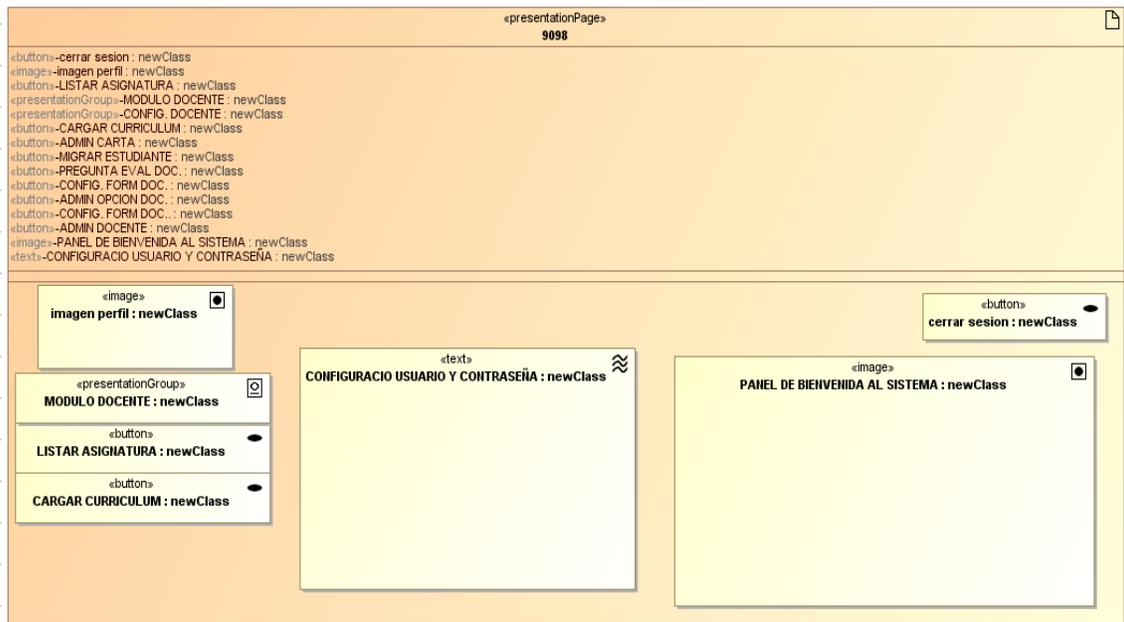
### 3.4.5.3 Modelo de presentación: docente

FIGURA N° 3. 25 Modelo de Presentación: Principal login docente



Fuente (Elaboración propia).

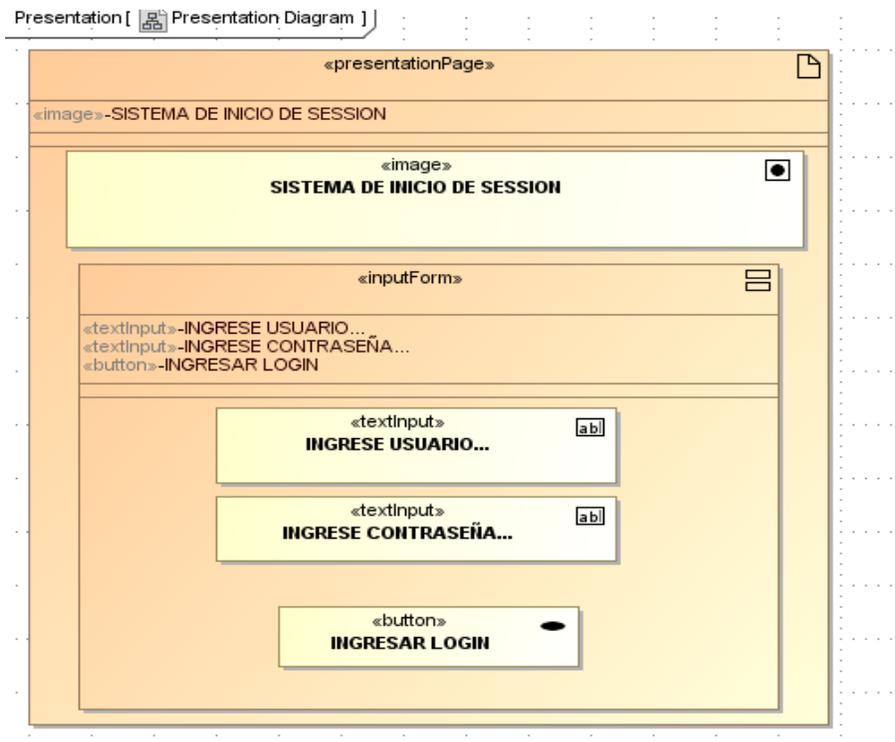
FIGURA N° 3. 26 Modelo de Presentación: docente



Fuente (Elaboración propia).

### 3.4.5.4 Modelo de presentación: estudiante

FIGURA N° 3. 27 Modelo de Presentación: Principal login estudiante



Fuente (Elaboración propia).

FIGURA N° 3. 28 Modelo de Presentación: estudiante



Fuente (Elaboración propia).

## **CAPITULO IV CALIDAD Y SEGURIDAD**

### **4.1. INTRODUCCIÓN**

En este capítulo presentaremos la calidad del sistema y la seguridad que contempla, orientado a la transición de las pruebas que se realizan en el sistema.

### **4.2 MÉTRICAS DE CALIDAD DE SOFTWARE**

En este acápite se verá el desarrollo de la medición de la calidad del software mediante la métrica iso-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.

Se puede decir que el software tiene calidad si cumple o excede las expectativas del usuario en cuanto a:

- ❖ Funcionalidad (que sirva un propósito).
- ❖ Ejecución (que sea práctico).
- ❖ Confiabilidad (que haga lo que debe).
- ❖ Disponibilidad (que funcione bajo cualquier circunstancia).

Apoyo, a un costo menor o igual al que el usuario está dispuesto a pagar.

#### **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. Determina las cinco características del dominio de información, teniendo en cuenta su cantidad y se definen de la siguiente forma:

- ❖ Número de Entradas de Usuario
- ❖ Número de Salidas de Usuario
- ❖ Numero de petición de Usuario
- ❖ Numero de archivos
- ❖ Numero de Interfaces externas

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 N° 4. 1 Número de Entradas de Usuario

Entradas de Usuario		
1	Módulo del Kardex	10
2	Módulo del encargado de comisión	32
3	Módulo del Docente	4
4	Módulo del Estudiante	3
Total		49

**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 informe no se cuentan de forma separada.

TABLA N° 4. 2 Número de Salidas de Usuario

Salidas de Usuario		
1	Módulo del Kardex	8
2	Módulo del encargado de comisión	25
3	Módulo del Docente	3
4	Módulo del Estudiante	2
Total		38

**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 N° 4. 3 Número de Peticiones de usuario

Número de petición de Usuario		
1	Módulo del Kardex	11
2	Módulo del encargado de comisión	37
3	Módulo del Docente	4
4	Módulo del Estudiante	3
Total		55

**Fuente:** (Elaboración Propia)

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

TABLA N° 4. 4 Número de Archivos

Número de Archivos		
1	Módulo del Kardex	16
2	Módulo del encargado de comisión	23
3	Módulo del Docente	5
4	Módulo del Estudiante	3
Total		47

**Fuente:** (Elaboración Propia)

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

TABLA N° 4. 5 Numero de interfaces Externas

Número de petición de Usuario		
1	Internet	1
Total		1

**Fuente:** (Elaboración Propia)

Para calcular los puntos de función se tiene:

TABLA N° 4. 6 Calculo de punto de función

Parámetros de medida		Cuenta	Factores de Ponderación			Total
			Simple	Medio	Complejo	
1	Nro. de Entradas de Usuario	49	3	4	6	147
2	Nro. de Salidas de Usuario	38	4	5	7	152
3	Nro. de Peticiones de Usuario	55	3	4	6	165
4	Nro. de Archivos	47	7	10	15	329
5	Nro. de Interfaces Externas	1	5	7	10	5
Total						798

**Fuente:** (Elaboración Propia)

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

TABLA N° 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		4
4	¿El rendimiento es crítico?				X			3
5	¿Será 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 sobre múltiples pantallas o variadas opciones?						X	5
8	¿Se actualizan los archivos maestros de forma interactiva?						X	5
9	¿Son complejas de las entradas de salidas de archivos?				X			3
10	¿Lógica del proceso Interno Compleja?			X				2
11	¿Se diseña el código para ser reutilizable?						X	5
12	¿Están incluidas en el diseño conversiones de instalación?		X					1
13	¿Instalaciones Múltiples?		X					1
14	¿Facilidad de Cambios?						X	5
Factor de Complejidad Total (FCT)								54

**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

$(0.65+0.01*\sum Fi)$ : 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:

$$PF = 798 * [0.65 + 0.01 * 54]$$

**PF = 949.6** Si calculamos al 100% el nivel de confianza consideramos la sumatoria de **Fi=70** como el máximo valor de ajuste de complejidad se tiene:

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

$$PF \text{ max} = 798 * [0.65 + 0.01 * 70]$$

$$PF = 1077.3$$

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

$$\text{Funcionalidad} = \frac{949.6}{1077.3} = 0,88$$

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

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 pros balísticos.

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).

$P(T \leq t) = F(t)$  (2) probabilidad de fallas

$P(T \leq t) = 1 - F(t)$  (3) 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.

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

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

$F(t) = 0,84$

La confiabilidad del sistema es del 84% 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 que la carrera tenga y los reglamentos que está regida por la misma.

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 formula.

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

**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

$$IMS = \frac{[4 - (0 + 1 + 0)]}{4}$$

$$IMS = 0,75$$

Con lo que podemos decir que el nuevo sistema tiene una estabilidad de 75% que es la facilidad de mantenimiento el 25% 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.

TABLA N° 4. 8 Ajuste de Preguntas

PREGUNTAS	Respuestas		Ponderación %
	SI	NO	
¿Puede Utilizar con facilidad el sistema?	5	1	83%
¿Puede Controlar operaciones que el sistema solicita?	5	1	83%
¿Las Respuestas del sistema son complicadas?	1	5	83%
¿El Sistema permitió la retroalimentación de información?	6	0	100%
¿El sistema cuenta con interface agradable a	6	0	100%

la vista?			
¿La respuesta del sistema es satisfactoria?	5	1	83%
¿Le parece complicada las funciones del sistema?	1	5	83%
¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	5	83%
¿Los resultados que proporciona el sistema facilitan el trabajo?	6	0	100%
¿Durante el uso del sistema se produjo errores?	1	5	83%
<b>USABILIDAD</b>			<b>88%</b>

**Fuente:** (Elaboración Propia)

Existe un 88% 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.

El sistema se ejecuta en diferentes plataformas, es decir la portabilidad se refiere exclusivamente a la propiedad que posee un software que le permite ser ejecutado en diferentes plataformas y/o sistemas operativos.

##### **4.2.5.1 Portabilidad lado servidor**

A nivel sistema de software. El sistema porta bajo los siguientes sistemas operativos de la familia Microsoft, 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 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.

La facilidad de llevar el sistema de un entorno a otro sin ningún problema.

### **4.3 SEGURIDAD DEL SISTEMA**

#### **4.3 .1 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.

##### **4.3.1.1 Seguridad lógica.**

Gestión de comunicación y operaciones.

- ❖ 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.

##### **4.3.1.2 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.

##### **4.3.1.3 Seguridad organizativa**

En él 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.
- ❖ Encriptación Sha1 en las contraseñas.
- ❖ Manejo de roles y tipos usuario en el gestor de Base de Datos.

#### 4.4.1 Pruebas de Caja Negra

En la evaluación fundamental del sistema se observaron aspectos de funcionalidad, operativa, aceptación de entradas, resultado, etc.

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.

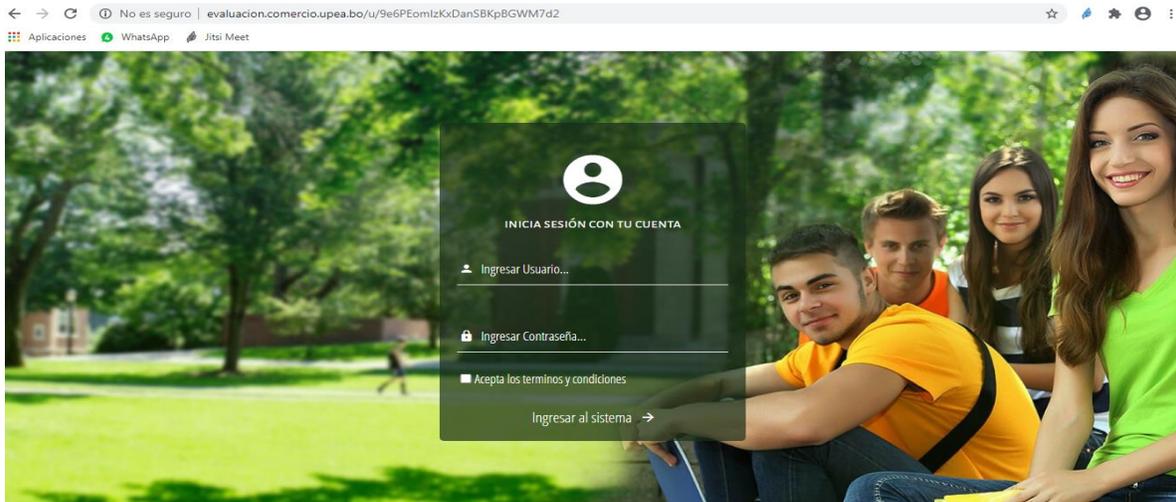
Registro de Usuario Datos de Ingreso:

Se ingresa un usuario y contraseña, proporcionado por el administrador del sistema.

**Prueba:** Se espera que el sistema realice las respectivas validaciones para el ingreso de los actores que participan el sistema.

**Resultado:** El sistema no permite el ingreso de los actores que no estén registrado en el sistema.

FIGURA N° 4. 1 proceso de la caja negra login



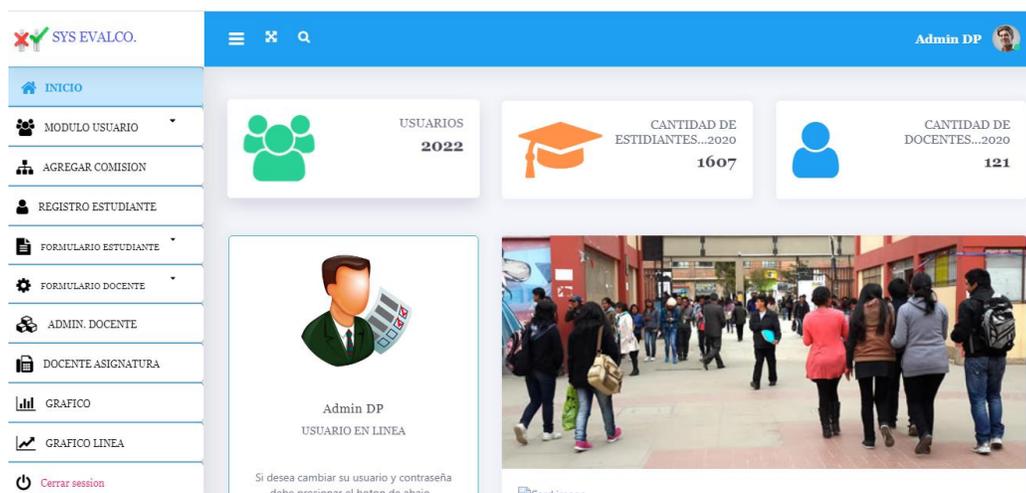
Fuente (Elaboración propia).

**Datos de Ingreso:** Ingresas al módulo del sistema KARDEX.

**Prueba:** Se espera que el sistema despliegue la información en base a su rol o tipo de usuario asignado.

**Resultado:** El sistema despliega enlaces de todos los menús asignados los actores del sistema.

FIGURA N° 4. 2 proceso de la caja negra KARDEX



Fuente (Elaboración propia).

**Datos de Ingreso:** Ingresa al módulo del sistema DOCENTE.

FIGURA N° 4. 3 proceso de la caja negra DOCENTE

The screenshot shows the user interface for a teacher. The top navigation bar includes the logo 'SYS EVALCO.' and the user's name 'UBALDO APAZA'. The main content area is divided into a sidebar and a main panel. The sidebar contains navigation options: 'INICIO', 'Docente', 'ASIGNATURA', 'HOJA DE VIDA', and 'Cerrar session'. The main panel displays the user's profile and a table of assigned subjects.

#	MATERIA	SIGLA	ACCION
1	LEGISLACION COMERCIAL	DER-112	LISTA
2	HISTORIA Y CULTURA ANDINA	SOC-118	LISTA
3	LEGISLACION COMERCIAL	DER-112	LISTA
4	DERECHO ADUANERO	DER-132	LISTA

Fuente (Elaboración propia).

**Datos de Ingreso:** Ingresa al módulo del sistema ESTUDINATE.

FIGURA N° 4. 4 proceso de la caja negra ESTUDIANTE

The screenshot shows the user interface for a student. The top navigation bar includes the logo 'SYS EVALCO.' and the user's name 'MILTON BLANCO'. The main content area is divided into a sidebar and a main panel. The sidebar contains navigation options: 'INICIO', 'Estudiante', '» EVALUACION DOCENTE', '» IMPRIMIR DOCUMENTO', and 'Cerrar session'. The main panel displays the user's profile and a table of assigned subjects.

#	MATERIA	SIGLA	DOC
1	ADMINISTRACION GENERAL	ADM-115	NAT SERI CON
2	HISTORIA Y CULTURA ANDINA	SOC-118	REM ZALI
3	LEGISLACION COMERCIAL	DER-112	UBA FIDE
4	ECONOMIA I	ECO-113	JAVI GUT
5	CONTABILIDAD	CON-111	GUS PERI

Fuente (Elaboración propia).

#### 4.4.2 Prueba de Caja Blanca

Para las pruebas de caja blanca se proporcionara una medición cuantitativa de la complejidad lógica del programa.

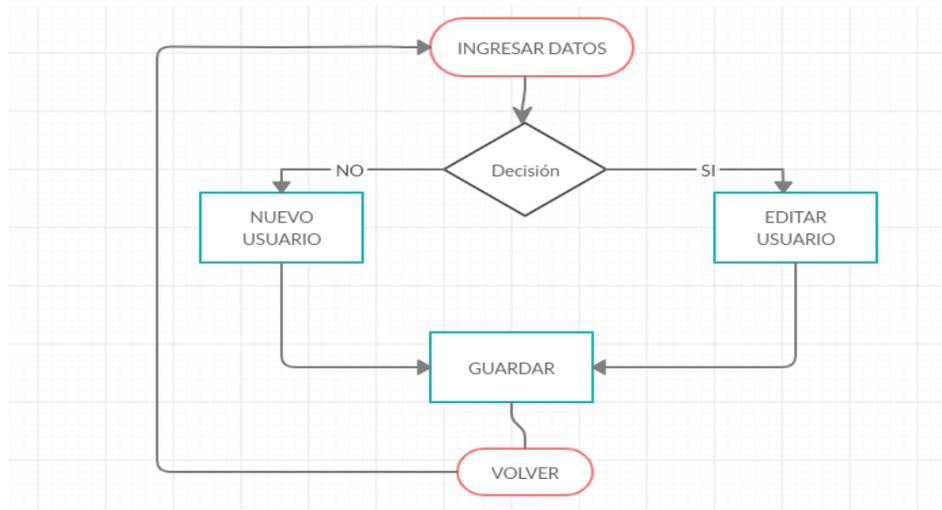
La Complejidad ciclomática de un grado de flujo  $V(G)$  establece el número de caminos independientes:

- ❖ 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.

#### Gestión de módulo de usuario

Se vio necesario realizar pruebas de caja blanca para el módulo de administración de usuario del sistema debido al flujo grama que presentaba como se puede mostrar en la figura.

FIGURA N° 4. 5 Pruebas de Caja blanca para el registro de usuario en el sistema



Fuente (Elaboración propia).

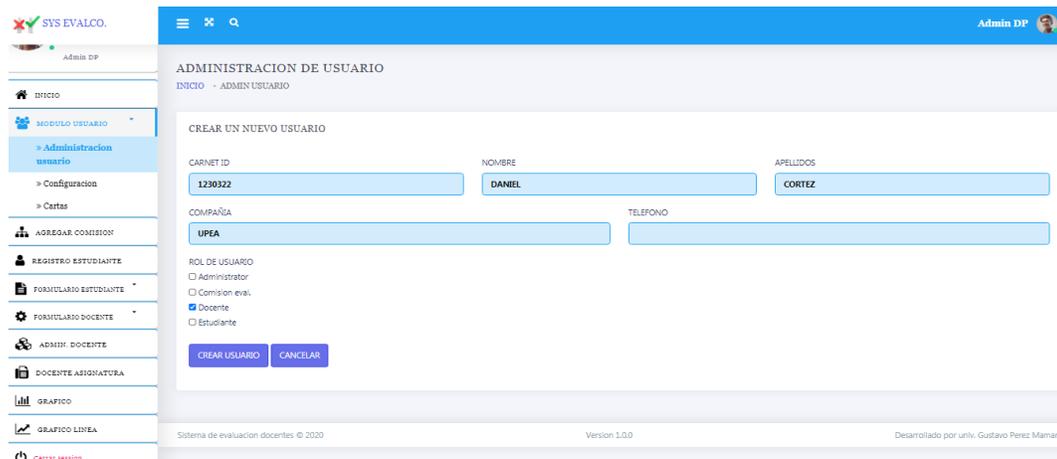
Para ilustrar el flujo grama se puede evidenciar el procedimiento y de administración de usuario.

FIGURA N° 4. 6 Pruebas de Caja blanca para el nuevo registro

The screenshot shows a web application interface for user management. The top navigation bar is blue and contains the logo 'SYS EVALCO.', a search icon, and the user name 'Admin DP'. The main content area is titled 'ADMINISTRACION DE USUARIO' and 'INICIO - ADMIN USUARIO'. Below the title, there is a section 'CREAR UN NUEVO USUARIO' with several input fields: 'CARNET ID.', 'APELLIDOS', 'APELLIDOS', 'COMPañIA', 'TELEFONO', 'USUARIO', 'CONTRASEÑA', and 'CONFIRMAR CONTRASEÑA'. There are also radio buttons for 'ROL DE USUARIO' with options: 'Administrator', 'Comision eval.', 'Docente', and 'Estudiante'. At the bottom of the form, there are two buttons: 'CREAR USUARIO' and 'CANCELAR'. The left sidebar contains a menu with items like 'Inicio', 'Modulo Usuario', 'Administracion usuario', 'Configuracion', 'Cartas', 'Agregar Comision', 'Registro Estudiante', 'Formulario Estudiante', 'Formulario Docente', 'Admin. Docente', 'Docente Asignatura', 'Grafico', and 'Grafico Linea'. The bottom of the sidebar has a 'Cerrar Sesión' button.

Fuente (Elaboración propia).

FIGURA N° 4. 7 Pruebas de Caja blanca para la modificación el registro



Fuente (Elaboración propia).

Muestra la complejidad ciclomatica 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.

TABLA N° 4.9 Casos de Prueba

CAMINO 1	CAMINO 2
1. Ingresar datos en el formulario de Administración de usuario. 2. Vuelve a ingresar datos para un nuevo registro de usuario.	1. Ingresar datos en el formulario de Administración de usuario. 2. Verifica y valida si existen datos incorrecto en el formulario. 3. Guarda información del funcionario en el sistema. 4. Vuelve a ingresar datos para un nuevo registro de usuario.

Elaboración (propia)

## **CAPITULO V ANÁLISIS DE COSTO**

### **5.1 INTRODUCCIÓN**

El presente capítulo, se presenta el análisis de costo del sistema en base al modelo de estimación de costo COCOMO.

### **5.2 MÉTODO DE ESTIMACIÓN DE COSTO DE SOFTWARE**

Existe distintos métodos para la estimación de costo de desarrollo de software, estos métodos no son otra cosa que establece una relación matemática entre el esfuerzo y el tiempo de desarrollo.

#### **5.2.1 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 7250 Líneas de Código en el lenguaje PHP, aplicado con la herramienta contador de líneas de código SLOC.

Aplicando Conversiones se tiene.

$$\mathbf{KLCD= (LCD)/1000}$$

$$\mathbf{KLCD= 7250 /1000}$$

$$\mathbf{KLDC= 7,25 KLDC}$$

Es un modelo Intermedio y sistema orgánico la evaluación del sistema ha sido considerada bajo las 7250 KLDC los coeficientes que se usarán serán los valores que se detallan en la siguiente tabla.

TABLA N° 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 N° 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 Nº 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	
Total			<b>0,72</b>			

Fuente: (Elaboración Propia)

Por tanto, nuestro Factor de ajuste será

$$FAE=0,72$$

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

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

$$E = 3,2 * 7,25^{1,05} * 0,72$$

$$E = 18 \text{ (Personas Mes)}$$

### **Cálculo del Tiempo**

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

$$T = 2,5 * 18^{0,38}$$

$$T = 7.49 \text{ (Meses)}$$

$$T = 7 \text{ meses}$$

### **Calculo de la Productividad**

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

$$PR = \frac{7250}{18} \text{ (Meses)}$$

$$PR = 402.77 \text{ (LCD/personas Mes)}$$

Calculo del personal requerido

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

$$P = \frac{18}{7.49} \text{ (Personas)}$$

$$P = 2.40 \text{ (Personas) equivalente a 2 personas}$$

### **Costo Total del Proyecto**

$$\text{(Coste Mes)} = P * \text{Salario medio entre los programadores y analistas}$$

### **Costo Persona Mes 350\$**

$$\text{Coste Mes} = 2 * 350 = 700 \$us$$

$$\text{Costo Total} = 700 * 7 = 4900 \$us$$

En resumen, se requiere 2 personas estimando un trabajo de 7 meses y con costo total de 4900\$us Equivalente en bolivianos a 35104 Bs.

## **CAPITULO VI CONCLUSIONES Y RECOMENDACIONES**

En este capítulo se dará las conclusiones y recomendaciones referentes al presente Proyecto de Grado.

### **6.1 CONCLUSIONES**

Se concluyó con los objetivos planteados en el presente proyecto, desarrollar un sistema de evaluación de docente, usando nuevas tecnologías de programación para optimizar el proceso de evaluación docente de la carrera Comercio Internacional de la Universidad Pública de El Alto en cada periodo académico que se conllevan.

- ❖ El sistema desarrollo de evaluación de docente, cuenta con las políticas de seguridad utilizando roles de usuarios, activándose la sesión al ingresar al sistema.
- ❖ Cuentas con los reportes necesarios para una toma de decisiones en la información correspondiente.
- ❖ El sistema se construyó con los siguientes módulos que a continuación lo mencionamos.
  - a) Módulo de administración kardex.
  - b) Módulo de administración comisión evaluador.
  - c) Módulo de administración de docente.
  - d) Módulo de administración de estudiante.
- ❖ Aplicando con éxito las normas de calidad, paradigma de Modelo Vista Controlador con PHP y las herramientas de programación para que tenga alta usabilidad y funcionalidad.

## 6.2 RECOMENDACIONES

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

- ❖ 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 encargado de sistema.

## BIBLIOGRAFIA

- 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>
- Álvarez, M. (09 de 09 de 2001). *Definición de PHP*. Recuperado el 09 de 11 de 2018, de Definición de PHP: <https://www.desarrolloweb.com/articulos/392.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>
- 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/>
- Blanco, S. D. (21 de 11 de 2015). *Metodología UWE*. Obtenido de Metodología UWE:  
<https://metodologiauwe.wordpress.comSistema>.
- 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 CodeIgniter: <http://blog.nubecolectiva.com/url-amigables-con-codeigniter/>
- Dianelys, V. R. (25 de noviembre de 2014). *Estimación de costos de desarrollo de software*. Recuperado el 12 de Julio de 2019, de Estimación de costos de desarrollo de software: <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
- Durán, J. (14 de marzo de 2016). *CLOC la herramienta para contar líneas de código*. Recuperado el 08 de Septiembre de 2019, de CLOC la herramienta para

contar líneas de código: <https://www.somosbinarios.es/cloc-herramienta-contar-lineas-codigo/>

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/>

Estela, M. (13 de Febrero de 2019). *software*. Recuperado el 10 de Julio de 2019, de

¿Qué es software?: <https://concepto.de/software/>

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>

GARCÍA, D. (07 de Septiembre de 2017). *ISO 27001 e ISO 27002 para la Gestión de*

*Riesgos en la Seguridad de la Información*. Recuperado el 10 de 09 de 2019,

de ISO 27001 e ISO 27002 para la Gestión de Riesgos en la Seguridad de la

Información: <https://www.ealde.es/gestion-de-riesgos-seguridad-de-la-informacion/>

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 Ingeniería del Software:

<http://ingsoftaihs.blogspot.com/>

- Jiménez, J. D. (20 de Enero de 2019). *Qué significa CCS3*. Recuperado el 15 de Julio de 2019, de Qué significa CCS3: <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.
- 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*. Recuperado el 09 de Agosto de 2019, de UWE – UML-based Web Engineering: <http://uwe.pst.ifi.lmu.de/teachingTutorialPresentationSpanish.html>
- Mercado, J. (27 de Agosto de 2018). *Técnicas de estimación de software*. Recuperado el 11 de Junio de 2019, de Técnicas de estimación de software: <http://www.pmoinformatica.com/2018/08/tecnicas-estimacion-software.html>
- Merino, J. P. (21 de diciembre de 2013). *Definición de Web*. Recuperado el 12 de Junio de 2019, de Definición de Web: <https://definicion.de/web/>
- Midena, J. (22 de Octubre de 2013). *COCOMO*. Recuperado el 12 de Diciembre de 2013, de COCOMO: <http://es.wikipedia.org/wiki/COCOMO>
- Myriam Yadira, Y. C. (22 de Enero de 2011). *Taller Cliente Servidor*. Recuperado el 14 de Diciembre de 2011, de Taller Cliente Servidor: <http://basesii.wikispaces.com/file/view/Caracter%C3%ADsticas+de+la+arquitectura+Cliente.pdf>
- 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: <https://www.isotools.org/2019/06/11/iso-27002-la-importancia-de-las-buenas-practicas-en-los-sistemas-de-seguridad-de-la-informacion/>

Porto, J. P. (30 de Diciembre de 2018). *definicion de JQuery*. Recuperado el 15 de julio de 2019, de definicion de JQuery: <https://definicion.de/jquery/>

PRESSMAN, R. (2002). *Ingeniería de Software. Un Enfoque Práctico. Quinta Edición* (QUITA EDICIÓN ed.). MEXICO: MC GRAW HILL.

Pressman, R. (2005). *Ingeniería del Software*. Recuperado el 12 de Agosto de 2019

Pressman, R. S. (2010). *Ingeniería del software un Enfoque Practico* (7ma. 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)