

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

#### SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORÍA Y CAPACITACIÓN CASO: “NUAM” (NUEVO AMANECER)

Para optar al título de Licenciatura en Ingeniería de Sistemas

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante:** Univ. Raquel Apaza Cupana

**Tutor Metodológico:** M. Sc. Ing. Enrique Flores Baltazar

**Tutor Especialista:** Ing. Ramiro Kantuta Limachi

**Tutor Revisor:** Lic. Norman Gudy Cardeña Pinto

**EL ALTO – BOLIVIA**

**2020**

*Dedicatoria*

*A mis queridos padres Víctor Apaza y Valeriana Cupana que me brindaron su apoyo incondicional.*

*A mi hija Jazmín por darme el valor de seguir luchando en la vida, con su constante apoyo y amor.*

*Al ser supremo que guía mis pasos, a mi entera familia y todas las personas que se les lleva en el corazón por su comprensión incondicional que me brindaron en todo este proceso.*

*Univ. Raquel Apaza Cupana*

## AGRADECIMIENTOS

*A Dios por haberme permitido llegar hasta esta etapa de mi vida.*

*Agradecer a todos los que tuvieron que ver en el desarrollo y conclusión de este proyecto de grado.*

*A mi tutor metodológico M. Sc. Ing. Enrique Flores Baltazar por su conocimiento, apoyo, confianza, tiempo, paciencia y motivación constante para la culminación del presente proyecto.*

*A mi tutor especialista Ing. Ramiro Kantuta Limachi por su conocimiento y orientación durante el desarrollo del presente proyecto.*

*A mi tutor revisor Lic. Norman Gudy Cardeña Pinto por su disponibilidad, su acertada orientación y sugerencias durante el desarrollo del presente proyecto.*

*A la institución por otorgarme la oportunidad y confianza de desarrollar el sistema.*

*A la Universidad Pública de El Alto, por acogerme en sus aulas durante todos los años de estudio, Así también a la carrera de Ingeniería de Sistemas por ser parte de ella.*

*¡¡Muchas Gracias a todos!!*

## **RESUMEN**

En la actualidad existen instituciones que cada día generan mucha información, en el caso la institución de consultoría y capacitación "NUAM" cada día genera informes sobre pagos de mensualidades e inscripciones, esta institución desarrolla consultoría y capacitación en distintas áreas donde se aplica de forma practica la enseñanza y apoyo a alumnos en todas las edades para su desarrollo educativo.

El presente proyecto tiene como finalidad facilitar el proceso de inscripciones y control de pago de mensualidades mediante la implementación de un Sistema de control, el cual permitirá la automatización de dichos procesos.

En la parte introductoria se muestra los antecedentes y actividades que realiza la institución. También se muestra el análisis de los problemas y los objetivos propuestos. Para el desarrollo del proyecto se aplicó la metodología de desarrollo UWE para el modelado del diseño que es comprensible y entendible para el desarrollo del sistema.

El Sistema es un producto de calidad de acuerdo a la metodología de evaluación que se hizo conforme a las métricas de calidad ISO 9126.

Finalmente se concluye que los objetivos planteados fueron alcanzados y que el sistema cumple con los requerimientos establecidos por el usuario.

**Palabra clave: Sistema, control, seguimiento, metodología UWE, métricas de calidad ISO 9216.**

## **ABSTRACT**

Currently there are institutions that generate a lot of information every day, in case the institution consultant and training "NUAM" generates daily reports on monthly payments and registrations, this institution developed consultant and training in varied subject, where apply in a practice way to support students of all levels for educational development.

This project aims to facilitate the process of registration and control of monthly payments by implementing a system control, which will allow the automation of these processes.

In the introductory part of the background and activities of the institution shown. The analysis of problems and objectives are also shown.

For the project development methodology UWE for modeling design that it was applied.

The system is a product of quality because is applied according to the methodology of evaluation quality metrics ISO 9126.

Finally it can be concluded that the objectives were achieved and that the system satisfy the requirements set by the user.

**Key Word: System, control, followup, UWE methodology, quality metrics ISO 9126.**

## ÍNDICE

<b>1. MARCO PRELIMINAR.....</b>	<b>1</b>
<b>1.1. Introducción .....</b>	<b>1</b>
<b>1.2. Antecedentes .....</b>	<b>1</b>
1.2.1. Antecedentes Institucionales.....	1
1.2.2. Antecedentes Académicos.....	2
<b>1.3. Planteamiento Del Problema.....</b>	<b>3</b>
1.3.1. Problema Principal.....	3
1.3.2. Problemas Secundarios.....	4
<b>1.4. Objetivos .....</b>	<b>4</b>
1.4.1. Objetivo General .....	4
1.4.2. Objetivos Específicos.....	5
<b>1.5. Justificaciones.....</b>	<b>5</b>
1.5.1. Justificación Técnica.....	5
1.5.2. Justificación Económica.....	5
1.5.3. Justificación Social .....	6
<b>1.6. Metodología.....</b>	<b>6</b>
1.6.1. UWE (Ingeniería Web basada en UML).....	6
1.6.2. Métricas de Calidad.....	7
1.6.3. Ingeniería de Costos.....	7
<b>1.7. Herramientas.....</b>	<b>7</b>
<b>1.8. Límites y Alcance.....</b>	<b>9</b>
1.8.1. Límites.....	9

1.8.2.	Alcances .....	10
<b>1.9.</b>	<b>Aportes. ....</b>	<b>10</b>
<b>2.</b>	<b>MARCO TEÓRICO .....</b>	<b>12</b>
<b>2.1.</b>	<b>Introducción .....</b>	<b>12</b>
<b>2.2.</b>	<b>Conceptos Básicos. ....</b>	<b>12</b>
2.2.1.	Técnica. ....	12
2.2.2.	Datos.....	12
2.2.3.	Sistema.....	12
2.2.4.	Control. ....	13
2.2.5.	Control Administrativo.....	14
2.2.6.	Seguimiento.....	14
2.2.7.	Sistema Administrativo. ....	14
2.2.8.	Sistema de información. ....	15
2.2.9.	Interfaz.....	15
<b>2.3.</b>	<b>Ingeniería web.....</b>	<b>15</b>
2.3.1.	Sistema web .....	15
2.3.2.	Arquitectura modelo vista controlador .....	16
2.3.3.	Características.....	17
2.3.4.	Ventajas.....	18
2.3.5.	Desventajas. ....	18
<b>2.4.</b>	<b>Ingeniería del software. ....</b>	<b>19</b>
2.4.1.	Proceso del software.....	20

2.4.2.	Modelo del proceso del software.....	20
2.4.3.	Modelo del proceso incremental. ....	21
<b>2.5.</b>	<b>Metodología UWE ( Uml-based Web Engineering) .....</b>	<b>22</b>
2.5.1.	Fases de la Metodología UWE.....	23
2.5.2.	Modelos UWE.....	25
2.5.3.	Ventajas y Desventajas. ....	29
<b>2.6.</b>	<b>Método de Prueba del Software.....</b>	<b>30</b>
2.6.1.	Definición caso de prueba.....	30
2.6.2.	Prueba de Caja Blanca.....	30
2.6.3.	Prueba de Caja Negra. ....	31
<b>2.7.</b>	<b>Métricas de Calidad de Software. ....</b>	<b>31</b>
2.7.1.	Definición de Calidad. ....	31
2.7.2.	Métricas de Calidad Establecida por el Estándar ISO 9126.....	32
2.7.3.	Fórmulas de Evaluación de Calidad.....	32
2.7.4.	Confiabilidad.....	36
2.7.5.	Usabilidad. ....	38
2.7.6.	Mantenimiento.....	40
2.7.7.	Portabilidad.....	41
2.7.8.	Eficiencia.....	41
<b>2.8.</b>	<b>MÉTODO DE ESTIMACIÓN DE COSTOS DEL SOFTWARE. ....</b>	<b>42</b>
2.8.1.	Definición del modelo.....	42
2.8.2.	Ecuación del Modelo.....	43



<b>2.9.</b>	<b>Herramientas de desarrollo.....</b>	<b>46</b>
2.9.1.	Lenguajes De Programación. ....	46
2.9.2.	Sistema Gestor de Base De Datos. ....	50
2.9.3.	Servidor Web Apache. ....	52
<b>3.</b>	<b>MARCO APLICATIVO.....</b>	<b>55</b>
<b>3.1.</b>	<b>Introducción .....</b>	<b>55</b>
<b>3.2.</b>	<b>Esquema del Organigrama.....</b>	<b>55</b>
<b>3.3.</b>	<b>Análisis y Diseño de Requerimientos. ....</b>	<b>56</b>
3.3.1.	Descripción general del escenario .....	56
3.3.2.	Requerimientos Funcionales. ....	58
3.3.3.	Requerimientos no funcionales .....	59
<b>3.4.</b>	<b>Desarrollo Metodología UWE.....</b>	<b>60</b>
3.4.1.	Modelo de requerimientos.....	60
3.4.2.	Modelo de contenido .....	66
3.4.3.	Modelo de navegación .....	67
3.4.4.	Modelo de presentación.....	68
<b>3.5.</b>	<b>Implementación del sistema .....</b>	<b>69</b>
3.5.1.	Base de Datos .....	69
3.5.2.	Modelo de la Base de Datos .....	69
3.5.3.	Especificaciones de Requerimiento para la Aplicación .....	70
3.5.4.	Módulo de Navegación .....	71
<b>3.6.</b>	<b>Aplicación de la Métrica de Calidad ISO 9126.....</b>	<b>89</b>
3.6.1.	NORMA ISO 9126 .....	89

3.7.	<b>Estimación de Costos del Sistema .....</b>	<b>99</b>
3.7.1.	Cocomo II .....	99
3.7.2.	Método de Estimación.....	99
<b>4.</b>	<b><i>PRUEBAS Y RESULTADOS.....</i></b>	<b><i>106</i></b>
4.1.	Introducción .....	106
4.2.	Pruebas de Factibilidad del Sistema .....	106
4.3.	Pruebas de Caja Negra.....	106
4.4.	Niveles de Prueba .....	107
4.4.1.	Pruebas Unitarias .....	107
4.4.2.	Pruebas de Integración.....	107
4.4.3.	Pruebas de Software.....	107
4.4.4.	Pruebas de Aceptación.....	107
4.5.	Resultados.....	108
<b>5.</b>	<b><i>CONCLUSIONES Y RECOMENDACIONES.....</i></b>	<b><i>110</i></b>
5.1.	Conclusiones.....	110
5.2.	Recomendaciones .....	111

## ÍNDICE DE FIGURAS

<b><i>CAPITULO II</i></b>	<b><i>MARCO TEORICO</i></b>	<b><i>Pág.</i></b>
<b><i>Figura 2.1 Gráfica General del Sistema</i></b>	_____	<b>13</b>
<b><i>Figura 2.2 Modelo Vista Controlador</i></b>	_____	<b>17</b>
<b><i>Figura 2.3 Grafico de Modelo Incremental.</i></b>	_____	<b>22</b>
<b><i>Figura 2.4 UML- Based Web Engineering</i></b>	_____	<b>22</b>
<b><i>Figura 2.5 Modelo de casos de uso</i></b>	_____	<b>25</b>
<b><i>Figura 2.6 Modelo de Actividades.</i></b>	_____	<b>26</b>
<b><i>Figura 2.7 Modelo de Contenido</i></b>	_____	<b>27</b>
<b><i>Figura 2.8 Modelo de Navegación.</i></b>	_____	<b>28</b>

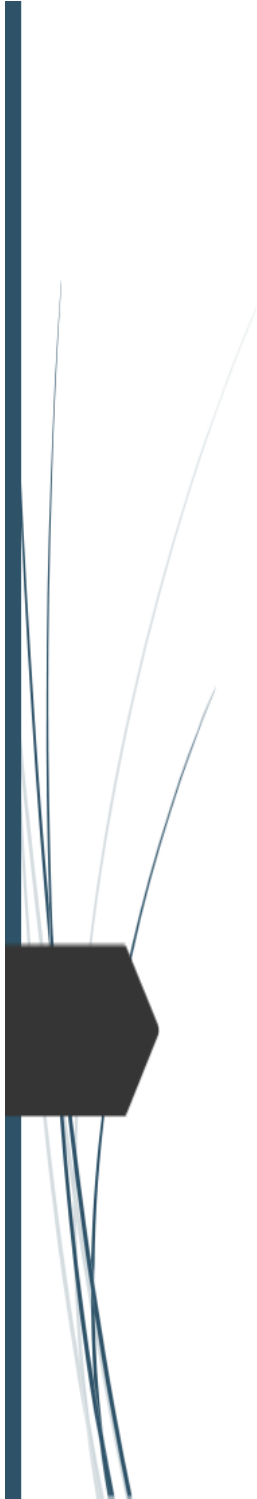
<b>Figura 2.9 Modelo Vista Controlador</b>	29
<b>Figura 2.10 JavaScript</b>	47
<b>Figura 2.11 jQuery</b>	48
<b>Figura 2.12 Bootstrap</b>	49
<b>Figura 2.13 Yii Framework</b>	50
<b>Figura 2.14 MySQL</b>	51
<b>Figura 2.15 Servidor Web Apache</b>	52
<b>CAPITULO III</b>	
<b>MARCO APLICATIVO</b>	<b>Pág</b>
<b>Figura 3.1 Organigrama de la consultoría y Capacitación “NUAM”</b>	56
<b>Figura 3.2 Diagrama de caso de uso general</b>	62
<b>Figura 3.3 Modelo de Contenido</b>	67
<b>Figura 3.4 Modelo de Navegación</b>	68
<b>Figura 3.5 Modelo de Presentación</b>	69
<b>Figura 3.6 Modelo de la Base de Datos</b>	70
<b>Figura 3.7 Interfaz de inicio</b>	72
<b>Figura 3.8 Código Interfaz de Inicio</b>	72
<b>Figura 3.9 Ingreso login</b>	73
<b>Figura 3.10 Código Ingreso Login</b>	73
<b>Figura 3.11 Módulo Coordinador</b>	74
<b>Figura 3.12 Código Modulo Coordinador</b>	74
<b>Figura 3.13 Adicionar Nuevo Coordinador</b>	75
<b>Figura 3.14 Código Adicionar Nuevo Coordinador</b>	75
<b>Figura 3.15 Módulo Docente</b>	76
<b>Figura 3.16 Código Modulo Docente</b>	76
<b>Figura 3.17 Adiciona Docente</b>	77
<b>Figura 3.18 Código Adiciona Docente</b>	77
<b>Figura 3.19 Módulo Estudiante</b>	78
<b>Figura 3.20 Código Modulo Estudiante</b>	78
<b>Figura 3.21 Adiciona Estudiante</b>	79
<b>Figura 3.22 Código Adiciona Estudiante</b>	79
<b>Figura 3.23 Módulo Cursos</b>	80

<b>Figura 3.24 Código Módulo Cursos</b>	80
<b>Figura 3.25 Adiciona Cursos</b>	81
<b>Figura 3.26 Código Adiciona Cursos</b>	81
<b>Figura 3.27 Módulo Inscripción</b>	82
<b>Figura 3.28 Código Módulo Inscripción</b>	82
<b>Figura 3.29 Adiciona Inscripción</b>	83
<b>Figura 3.30 Código Adiciona Inscripción</b>	83
<b>Figura 3.31 Módulo Mensualidades</b>	84
<b>Figura 3.32 Código Módulo Mensualidades</b>	84
<b>Figura 3.33 Adiciona Mensualidad</b>	85
<b>Figura 3.34 Código Adiciona Mensualidad</b>	85
<b>Figura 3.35 Módulo Administrar Pagos</b>	86
<b>Figura 3.36 Código Módulo Administrar Pagos</b>	86
<b>Figura 3.37 Adiciona Administrar Pagos</b>	87
<b>Figura 3.38 Código Adiciona Administrar Pagos</b>	87
<b>Figura 3.39 Modulo Historial de Reportes</b>	88
<b>Figura 3.40 Código Módulo Historial de Reportes</b>	88

## ÍNDICE DE TABLA

<b>CAPITULO II</b>	<b>MARCO TEORICO</b>	<b>Pág.</b>
<b>Tabla 2.1</b>	<b>Ecuación de la Funcionalidad</b>	33
<b>Tabla 2.2</b>	<b>Factor de Ponderación</b>	34
<b>Tabla 2.3</b>	<b>Variable para el Cálculo de la Funcionalidad.</b>	34
<b>Tabla 2.4</b>	<b>Ajuste de la Complejidad</b>	35
<b>Tabla 2.5</b>	<b>Ecuación de la Confiabilidad</b>	37
<b>Tabla 2.6</b>	<b>Ecuación de la Usabilidad</b>	38
<b>Tabla 2.7</b>	<b>Valoración de las Preguntas</b>	39
<b>Tabla 2.8</b>	<b>Preguntas para Determinar la Usabilidad</b>	39
<b>Tabla 2.9</b>	<b>Ecuación de la Mantenibilidad.</b>	40
<b>Tabla 2.10</b>	<b>Detalle de Coeficiente de COCOMO II</b>	44
<b>Tabla 3.1</b>	<b>Requerimientos Funcionales</b>	59

<b>Tabla 3.2</b>	<i>Requerimientos No Funcionales</i>	59
<b>Tabla 3.3</b>	<i>Descripción de caso de uso “Administrador del Sistema”</i>	62
<b>Tabla 3.4</b>	<i>Descripción de Caso de Uso “Reportes”</i>	63
<b>Tabla 3.5</b>	<i>Descripción de Caso de Uso “Asignación de Materia”</i>	63
<b>Tabla 3.6</b>	<i>Descripción de Caso de Uso “Registrar Estudiante”</i>	64
<b>Tabla 3.7</b>	<i>Descripción de Caso de Uso “Emisión de Recibo”</i>	64
<b>Tabla 3.8</b>	<i>Descripción de Caso de Uso “Inscripción”</i>	64
<b>Tabla 3.9</b>	<i>Descripción de Caso de Uso “Retiro y Adición”</i>	65
<b>Tabla 3.10</b>	<i>Descripción de Caso de Uso “Consultas”</i>	65
<b>Tabla 3.11</b>	<i>Descripción de Caso de Uso “Acceso al Sistema”</i>	66
<b>Tabla 3.12</b>	<i>Requerimientos para la Aplicación</i>	71
<b>Tabla 3.13</b>	<i>Cálculo de Parámetros de PF</i>	91
<b>Tabla 3.14</b>	<i>Parámetros de Medición</i>	91
<b>Tabla 3.15</b>	<i>Valoración de las Preguntas</i>	95
<b>Tabla 3.16</b>	<i>Parámetros de Medición de la Usabilidad</i>	96
<b>Tabla 3.17</b>	<i>Resultados de la Norma ISO 9126</i>	98
<b>Tabla 3.18</b>	<i>Coeficiente de COCOMO II</i>	101
<b>Tabla 3.19</b>	<i>Ecuaciones del Modelo COCOMO II</i>	102
<b>Tabla 3.20</b>	<i>Cálculo de los atributos de la FAE</i>	103



# **CAPITULO I**

## **MARCO**

### **PRELIMINAR**

## **1. MARCO PRELIMINAR**

### **1.1. Introducción**

En la actualidad un sistema de información constituye la base para el desarrollo de cada empresa o institución. Es el soporte principal del trabajo de los directivos, ya que permite coordinar el trabajo dentro y fuera de la organización con el fin de tomar decisiones oportunas y efectivas.

Los sistemas sirven para el adecuado control de flujo de información para la institución es la más necesaria. Ya que en Bolivia la implementación de estos sistemas avanza de a poco por lo que algunas instituciones trabajan de manera manual. En el cual el presente trabajo permite desarrollar un SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORÍA Y CAPACITACIÓN. Con el fin de agilizar los procesos manuales para así obtener una información oportuna y confiable.

En la actualidad la Consultoría y Capacitación “NUAM” no cuenta con un sistema de información digital que realice los reportes de las capacitaciones, así como el registro de los estudiantes en sus distintas materias. De esta manera la institución hará frente a la necesidad de sofisticar su servicio e incrementar su eficiencia para hacer frente al ámbito de las distintas capacitaciones.

La metodología que se utiliza es UWE (UML-BASED WEB ENGINEERING) puesto que en algún momento los cambios podrán llegar a ser recurrentes, así se debe analizar, implementar, codificar y pasar el test cada vez que sea necesario, permitiendo la construcción del Sistema.

Las Herramientas utilizadas son el lenguaje de programación JavaScript, el gestor de base de datos MySQL, servidor web Apache, y un Yii Framework bajo la plataforma Windows.

### **1.2. Antecedentes**

#### **1.2.1. Antecedentes Institucionales.**

En la Consultoría y Capacitación “NUAM”, todos los registros de estudiantes, docentes, administrativos y el pago de mensualidades, se realiza de forma

manual y el uso de Microsoft Word y Microsoft Excel, por lo cual se necesita una buena organización de la información de toda la institución.

Por lo general una buena administración constituye un pilar fundamental para que ayude a la toma de decisiones de manera más segura y confiable.

### **Ubicación:**

La consultoría y capacitación “NUAM” se encuentra ubicado en la Ciudad de El Alto La Paz, zona Villa Adela, Plan 36 Av. Junín N.º 1 Edif. (Don Tino)

#### **1.2.2. Antecedentes Académicos.**

En la consultoría y capacitación “NUAM” no existe ningún antecedente de algún proyecto similar al que se está planteando de un “SISTEMA”, sin embargo, existen otros proyectos similares implementados en otras instituciones.

### **DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORÍA Y CAPACITACIÓN**

#### **Internacional**

- [Darwin Alberto Coronel Ailla, el año 2010]. Análisis, diseño e implementación de un portal web para administrar la información de los proyectos de organizaciones.

La aplicación es una herramienta de apoyo para tomar decisiones con respecto a la canalización de los diferentes recursos asignados a las organizaciones y poder efectuar proyectos más eficaces y eficientes.

- [Fanny Maritza, Ana Rosa Guamaní Vizuete, el año 2010]. Diseño construcción e implementación de un portal web para el control y seguimiento académico de los estudiantes del System centros de Formación.

Permite solucionar los problemas de los diferentes departamentos ya que se lleva el control de cada estudiante en cuanto al registro de cada matricula, calificaciones, justificaciones, asistencia y pagos pendientes.

#### **Nacional**

- [Yenes Mónica Mendoza Mamani el año 2008]. Sistema Académico Vía Web Escuela Superior de Administración de Empresas (ESAE).



Tiene por objetivo brindar una serie de servicios como: consulta de calificaciones, materias inscritas, horarios registro de estudiantes y otros. Para su implementación se usó PHP como lenguaje de programación y como gestor de base de datos MYSQL.

- [Edwin Rubén Cachaga Villegas el año 2015]. Sistema Web de Gestión Académica y Repositorio Virtual, de la Universidad Mayor de San Andrés (UMSA).

Tiene por objetivo es ayudar a la comunidad educativa detallando las materias de Gestión escolar y proceso de notas, registro pedagógico en línea.

Para su implementación se aplicó la metodología UWE para el modelado de sistemas Web para el desarrollo del software, las herramientas que usa es Framework php Codeigniter, JavaScript y MySQL para la base de datos.

## **Local**

- [Elaborado por Víctor Hugo Calcina López el año 2007]. Sistema de Seguimiento Académico Carrera de Arquitectura Universidad Pública de el Alto (UPEA).

El sistema tiene por objetivo facilitar el trabajo del personal y universitarios, haciendo que los primeros mejoren la administración que se genera durante las actividades, especialmente de inscripción y las ventajas en cuanto a información.

Para su implementación se aplicó PHP como lenguaje de programación y MYSQL como gestor de base de datos.

## **1.3. Planteamiento Del Problema**

### **1.3.1. Problema Principal**

El problema central de la institución radica en registros manuales y consiguientemente inadecuados, que se realiza en las áreas y materias de consultoría y capacitación, inscripción de estudiantes, registros de tutores, pago de mensualidades poco eficientes lo cual genera falencias en los procesos de

administración lo que no coadyuva al logro de los objetivos de la institución” NUAM” (NUEVO AMANECER).

### **1.3.2. Problemas Secundarios**

Los problemas secundarios ligados a la problemática principal son los siguientes:

- Proceso de inscripción de forma manual y convencional existiendo dificultades al momento de la búsqueda de un alumno inscrito.
- El incremento de la documentación escrita, dificulta el acceso a la información de inscripciones y pagos de mensualidades, de forma precisa y oportuna.
- El registro de pagos de mensualidades y matrículas es de forma manual, el cual implicaría la posible pérdida de registros.
- Registros inexactos de los pagos de mensualidades e inscripción de los alumnos, asignación de tutores.
- No cuentan con una página donde centralice la información de cursos habilitados.
- La publicación de los requisitos para la inscripción de estudiantes o cualquier información sobre la institución son volantes, lo cual da un aspecto ambiguo a la institución por la falta de un sistema automatizado.

¿Cómo podría optimizar el sistema de control administrativo de registros de inscripción, control de pago de mensualidades, además del seguimiento de cada consultoría y capacitación?

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

“Diseñar un sistema de control administrativo, que permita el seguimiento de las inscripciones, pagos de mensualidades, asignación de pedagogos y registro de materias para consultoría y capacitación, brindándole así un medio de registros y reportes oportunos a cerca de la información que brinda “NUAM” (NUEVO AMANECER), con lo cual coadyuvará al logro de los objetivos de la institución.

### **1.4.2. Objetivos Específicos.**

- Analizar los requerimientos necesarios para el desarrollo del sistema.
- Diseñar el sistema de control administrativo bajo requerimiento de la institución.
- Centralizar la información en una base de datos para optimizar el manejo de la información de inscripciones.
- Registrar los pagos de mensualidades y matrícula de manera oportuna.
- Disminuir el tiempo del proceso de pago de matrícula, mensualidades para una mejor atención al cliente.
- Generar reportes con los requerimientos necesarios del usuario para un mejor control de la información de pagos de mensualidades, inscripción de alumnos y asignación de tutores.
- Coadyuvar la modernización de la institución “NUAM” NUEVO AMANECER con un sistema de control administrativo diseñada para el acceso a la información dando valor local, nacional al conocimiento de sus distintos servicios.

## **1.5. Justificaciones**

### **1.5.1. Justificación Técnica.**

La implementación del sistema cuenta con todos los equipos de hardware y software necesarios el cual permite llegar al objetivo planteado anteriormente. Los avances tecnológicos han logrado precisión, velocidad, confiabilidad y muchas otras ventajas en todos los campos laborales como ser educación, salud, seguridad y demás, es de esta manera recurriendo a las innovaciones tecnológicas como base fundamental la metodología UWE es una metodología completamente basada en UML (Unified Modeling Language), que cubre por completo el ciclo de vida de desarrollo de aplicaciones web.

### **1.5.2. Justificación Económica**

La institución tiene los recursos en hardware y software con el que cuenta actualmente, son suficientes para el desarrollo y la implementación del sistema gracias a la decisión de emplear herramientas económicamente factibles.

En cuanto a las herramientas que se aplicara en el Sistema se implementara los lenguajes libres para el desarrollo Web y base de datos. Para ello no tendrá ningún costo en cuanto a licencias comerciales u otras ya que son de manera gratuita.

### **1.5.3. Justificación Social**

La calidad de servicio que brinda el sistema es tener un constante control y comunicación a todas las personas que requieran de una consultoría o capacitación, con lo cual los beneficiarios directos será la administración porque al ser implementado se incrementa el grado de eficiencia del usuario, así obtener información oportuna cuando así lo requieran, llevando un control eficaz.

Los beneficiarios indirectos son la población que requiera de una consultoría o capacitación, quienes podrán acceder a consultar sobre los servicios que se brinda, horarios, áreas disponibles y otros servicios que requieran, gozando de la buena administración, para la toma de decisiones y mantener el prestigio.

## **1.6. Metodología.**

### **1.6.1. UWE (Ingeniería Web basada en UML).**

La Ingeniería Web basada en UML es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado. Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos, Corporación Universitaria Centro Superior (2017).

Las etapas de UWE cubre todo el ciclo de este tipo de aplicaciones, además su atención en aplicaciones personalizadas o adaptativas.

- Análisis de Requerimiento
- Diseño del Sistema

- Codificación del Software
- Pruebas
- Instalación y Fase de Implementación
- Mantenimiento.

### **1.6.2. Métricas de Calidad.**

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad (Pressman, 1998).

### **1.6.3. Ingeniería de Costos.**

La ingeniería de costos es la práctica ingenieril enfocada a la gestión de proyectos con un enfoque particular en costos; esta incluye actividades como la estimación y control, pronóstico, evaluación de inversiones y análisis de riesgos.<sup>1</sup> El presupuesto de ingeniería de costos planifica y monitorea proyectos de inversión. También se busca el equilibrio entre el balance de costos, calidad y requisiciones de tiempo (AACEE, 2015).

## **1.7. Herramientas**

Las herramientas a ser utilizados en el desarrollo de software:

- **JavaScript.**

JavaScript es un lenguaje que puede ser utilizado para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

JavaScript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Java por su parte tiene como principal característica ser un lenguaje

independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, (maestros de la Web, 2007).

- **Servidor Web Apache.**

El servidor HTTP Apache es un servidor web de software libre desarrollado por la Apache Software Foundation (ASF). El producto obtenido de este proyecto es un servidor de código fuente completo, descargable y gratuito.

Apache es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores voluntarios de que dispone.

Es también un servidor estable, eficiente, extensible y multiplataforma.

Estable: es una consecuencia de su probada robustez que impide caídas o cambios en el servidor inesperados.

Flexible y eficiente: es capaz de trabajar con el estándar HTTP/1.1 (RFC2616) y con la mayor parte de las

extensiones web que existen en la actualidad, como son los módulos PHP, SSL, CGI, SSI, proxy.

Extensible: dispone de gran cantidad de módulos que amplían su funcionalidad.

Multiplataforma ya que está disponible para diferentes plataformas como GNU/Linux, Windows, MacOS, (Elvira Mifsuf Talón).

- **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 el tamaño de una PC, una Tablet y otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive desing” o diseño adaptable (creado originalmente por los desarrolladores de Twitter en 2011).

- **MySQL.**

Es un sistema de gestión de base de datos relacional (RDBMS) desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base de datos Open Source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y esta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

- **Yii Framework.**

Es un Framework de PHP de alto rendimiento, basado en componentes para desarrollar aplicaciones web modernas en poco tiempo. El nombre Yii significa "simple y evolutivo" en chino. También se puede considerar como el acrónimo de Yes It Is (que en inglés significa Sí, lo es).

Yii es un Framework genérico de programación web, lo que significa que se puede utilizar para desarrollar todo tipo de aplicaciones web en PHP. Debido a su arquitectura basada en componentes y a su sofisticada compatibilidad de caché, es especialmente apropiado para el desarrollo de aplicaciones de gran envergadura, como páginas web, foros, sistemas de gestión de contenidos (CMS), proyectos de comercio electrónico, servicios web compatibles con la arquitectura REST. (Desarrollado por Yii Software LLC, 2008)

## **1.8. Límites y Alcance**

### **1.8.1. Límites**

El presente sistema de control administrativo de seguimiento para la consultoría y capacitación NUEVO AMANECER se limitará a:

- El Sistema no emitirá la impresión de una factura.
- No controlará la asistencia de estudiantes por las políticas de la institución.
- No efectuará el control de asistencia del personal.
- El sistema no realizará el proceso de notas académicas.

### **1.8.2. Alcances**

Para la implementación del sistema de información administrativa para seguimiento se automatizarán los siguientes procesos, los cuales serán los alcances del sistema.

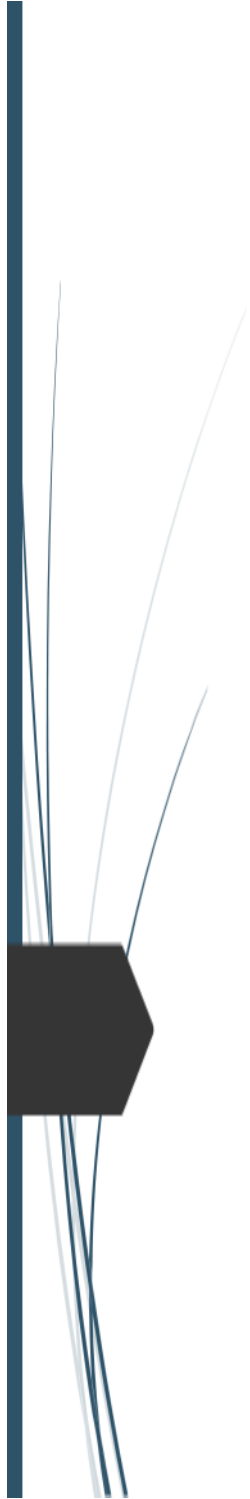
- Módulo de administración de usuarios.
- Módulo de inscripciones de estudiantes.
- Módulo de administración de tutores y/o responsables de área.
- Módulo de registro de pago de mensualidades y matrícula.
- Módulo de generación de reportes.

### **1.9. Aportes.**

El sistema desarrollado para la institución “NUAM” (NUEVO AMANECER), brinda un aporte óptimo de control y manejo de la información.

- Fácil acceso a la información que se generará en las inscripciones y pagos de mensualidades.
- Generación de reportes para un mejor control de pago de mensualidades.





# **CAPITULO II**

## **MARCO**

### **TEÓRICO**

## **2. MARCO TEÓRICO**

### **2.1. Introducción**

En este capítulo se describe los conceptos y teorías fundamentales que se consideren para la realización del presente proyecto, especificando las descripciones técnicas y metodológicas de desarrollo, viendo las funcionalidades, procesos, técnicas que permitan realizar el análisis de diseño e implementación del sistema.

### **2.2. Conceptos Básicos.**

#### **2.2.1. Técnica.**

Es el conjunto de instrumentos y medios a través de lo cual se efectúa el método y solo se aplica a una ciencia. La diferencia entre método y técnica es que el método es el conjunto de pasos y etapas que debe cumplir una investigación y este se aplica a varias ciencias mientras que técnica es el conjunto de instrumentos en el cual se efectúa el método, las técnicas son el mecanismo más eficaz para detectar los errores en una etapa temprana del proceso de software, (Pressman, 2010)

#### **2.2.2. Datos.**

El objetivo de todo sistema es el manejo de la información o datos. en una entidad estos datos pueden ser las cifras de pagos mensuales o las calificaciones de una clase, un dato es una expresión general que describe los objetos con los cuales se opera en los sistemas, (Luis Joyanes Aquilar, 2003).

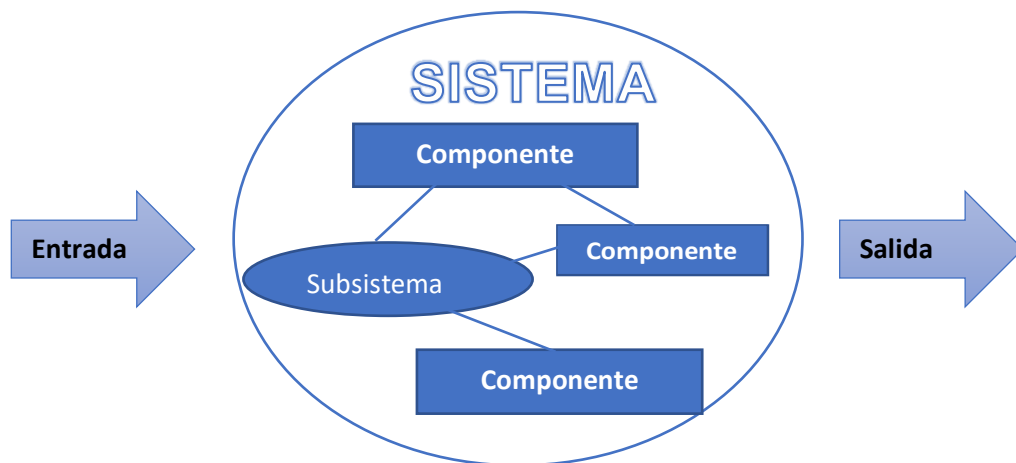
#### **2.2.3. Sistema.**

Un sistema es modelo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. El concepto se utiliza tanto para

definir a un conjunto de conceptos como a objetos reales dotados de organización. Este tipo de sistemas denominan al conjunto de hardware y software con el soporte humano que forma parte de una organización. (Julián Pérez Porto, 2008).

Conjunto de elementos interrelacionados, interdependientes, interactuantes, combinados y coordinados como un todo organizado en pos de un objetivo común, cuyo resultado (output) es mayor que el de la sumatoria que se obtendría si las unidades actuaran independientemente. Es decir que la sinergia del conjunto genera un resultado superior al de la suma de acciones individuales. (Ludwig von Bertalanffy, 1968).

Por tanto, se menciona que un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Esta definición se aplica a todos los tipos de sistemas de información. (Kendal, 1997)



**Figura 2.1** Gráfica General del Sistema

**Fuente:** (Leandro Alegsa, 2018)

#### **2.2.4. Control.**

Control es el proceso por el cual los procesos se aseguran que la obtención y el empleo de los recursos se efectúen en forma efectiva y eficiente, en el logro de

los objetivos de la organización. El proceso de control tiende a ser rítmico, sigue una pauta y es recurrente mes tras mes y año tras año.

“El proceso para determinar lo que se está llevando a cabo, evaluando y, si es necesario, aplicando medidas correctivas, de manera que la ejecución se desarrolle de acuerdo con lo planeado. “(Terry, 1984)

#### **2.2.5. Control Administrativo.**

El control administrativo se considera como un sistema de realimentación similar al que opera en el termostato de los hogares.

Este sistema coloca la función de control en una perspectiva más compleja y realista que si se la considera sólo como cuestión de establecer estándares, medir el desempeño y corregir las desviaciones.

Los gerentes miden el desempeño real, comparan esta medición contra los estándares e identifican y analizan las desviaciones, y entonces, para hacer las correcciones necesarias, deben desarrollar un programa de acción correctiva e instrumentarlo para llegar al desempeño deseado. (Harold Koontz, 2012)

#### **2.2.6. Seguimiento.**

“Seguimiento a la evaluación continua de la ejecución completa de los proyectos en relación con el programa acordado (durante), y su relación con la utilización de insumos, infraestructura y servicios.

El seguimiento proporciona información constante a los administradores y otros interesados, retroalimentando la ejecución del proyecto como tal. Permite identificar dificultades potenciales y adelantarse a las posibles complicaciones en tiempo y falta de recursos. “(Thumm, 1998).

#### **2.2.7. Sistema Administrativo.**

Conjunto de componentes que interactúan entre sí y se encuentran interrelacionados recibe el nombre de sistema. Administrativo, por su parte, es aquello vinculado a la administración (el acto de administrar: organizar o gestionar recursos).

Un sistema administrativo es una red o un esquema de procesos cuya finalidad es favorecer el cumplimiento de los objetivos de una organización. El sistema

apunta a que los recursos de la organización en cuestión sean administrados de forma eficiente. (Julián Pérez Porto y Ana Gardey, 2014).

### **2.2.8. Sistema de información.**

Un sistema de información es un conjunto de elementos relacionados entre sí, que se encarga de procesar manual y/o automáticamente datos, en función de determinados objetivos.

Estos elementos constituyentes de un sistema de información son:

- Personas o recurso humano: las personas que utilizan el sistema, siendo parte de éstos.
- Datos e información: de entrada, de salida y almacenada.
- Actividades o técnicas de trabajo.
- Recursos materiales: por ejemplo, computadoras, dispositivos, etc.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. (Diccionario de informática y Tecnología, 2017).

### **2.2.9. Interfaz.**

Esta noción sirve para señalar a la conexión que se da de manera física y a nivel de utilidad entre dispositivos o sistemas.

Se conoce como interfaz de usuario al medio que permite a una persona comunicarse con una máquina. La interfaz, en este caso, está compuesta por los puntos de contacto entre un usuario y el equipo. (Julián Pérez Porto y Ana Gardey, 2011).

## **2.3. Ingeniería web**

### **2.3.1. Sistema web**

Un sistema web o WebApp, es una aplicación que no depende de una plataforma, sino que está desarrollada para ser implementada en un servidor web con características y capacidades mayores a una página web.

Señala que las características de una WebApp son las siguientes:

- Inmediatez. Las aplicaciones basadas en Web tienen una inmediatez que no se encuentra en otros tipos de software. Es decir, el tiempo que se tarda

en comercializar un sitio Web completo puede ser cuestión de días o semanas. Los desarrolladores deberán utilizar los métodos de planificación, análisis, diseño, implementación y comprobación que se hayan adaptado a planificaciones apretadas en tiempo para el desarrollo de WebApps.

- Seguridad. Dado que las WebApps están disponibles a través del acceso por red, es difícil, si no imposible, limitar la población de usuarios finales que pueden acceder a la aplicación. Con objeto de proteger el contenido confidencial y de proporcionar formas seguras de transmisión de datos, deberán implementarse fuertes medidas de seguridad en toda la infraestructura que apoya una WebApp y dentro de la misma aplicación.
- Estética. Una parte innegable del atractivo de una WebApp es su apariencia e interacción. Cuando se ha diseñado una aplicación con el fin de comercializarse p vender productos o ideas, la estética puede tener mucho que ver con el éxito del diseño técnico. Disponible en: R. Pressman, 2010

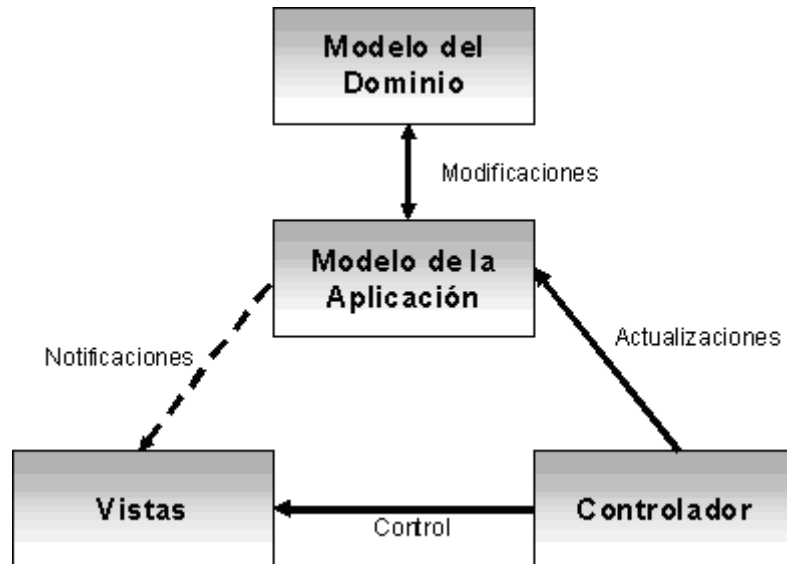
### **2.3.2. Arquitectura modelo vista controlador**

El Model-View-Controller o Modelo–Vista–Controlador (MVC) es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista, o a cualquier parte del sistema puedan ser hechas con un mínimo impacto en el componente del modelo de datos o en los otros componentes del sistema. Este patrón cumple perfectamente el cometido de modularizar un sistema.

El Modelo Vista Controlador (MVC) Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones.

Fue descrito por primera vez en 1979 por Trygve Reenskaug, entonces trabajando en Smalltalk en laboratorios de investigación de Xerox. Para los años 70 seguidamente en los años 80 Jim Althoff y otros implementaron una versión de MVC para la biblioteca de clases de Smalltalk-80. Solo más tarde, en 1988, MVC se expresó como un concepto general en un artículo sobre Smalltalk-80. (Nilsen Espitia, 2016)



**Figura 2.2 Modelo Vista Controlador**

**Fuente:** (Pantoja)

### 2.3.3. Características

Las tres principales características del patrón MVC son:

- **Modelo:** El Modelo es el nivel en el cual se manejan y controla y gestiona los datos (convertidos en información) del programa, el modelo puede interpretarse como independiente del resto puesto que la única relación de este nivel con el controlador y la vista, se basa en responder a las solicitudes de información de la vista que llegan al modelo por medio del controlador.
- **Vista:** La vista es la interfaz dirigida al usuario final, como ser una página web que debe ser desplegada por el navegador, mostrando la información

que obtiene e interpreta del modelo. Cabe destacar que pueden existir diferentes vistas relacionadas a un solo modelo.

- **Controlador:** El controlador se encarga de la interpretación de las solicitudes del usuario, se ocupa de recibir peticiones, las que envía al modelo, para que la información resultante pueda ser enviada e interpretada por la vista para el usuario final. Es decir, como se nombre lo indica se encarga de controlar el flujo de peticiones que involucran las actividades respectivas del modelo y la vista.

#### **2.3.4. Ventajas**

Desarrollar una aplicación siguiendo este patrón de diseño tiene muchas ventajas:

- La aplicación está implementada modularmente.
- Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, sólo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

#### **2.3.5. Desventajas.**

El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa, que el tiempo de desarrollo de una aplicación que no lo implementa, ya que MVC requiere que el programador



implemente una mayor cantidad de clases que en un entorno de desarrollo común no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es muchísimo más mantenible, extensible y modificable que una aplicación que no lo implementa.

MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación.

Esta arquitectura inicial debe incluir, por lo menos: un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.

MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma

#### **2.4. Ingeniería del software.**

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases clave:

- Disciplina de la ingeniería. Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.
- Todos los aspectos de producción de software. La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y

el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en Web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico. (SOMMERVILLE, 2005)

#### **2.4.1. Proceso del software.**

Un proceso del software es un conjunto de actividades que conducen a la creación de un producto software. Estas actividades pueden consistir en el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C. Sin embargo, cada vez más, se desarrolla nuevo software ampliando y modificando los sistemas existentes y configurando e integrando software comercial o componentes del sistema.

#### **2.4.2. Modelo del proceso del software.**

Un modelo del proceso del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona sólo información parcial sobre ese proceso.

En esta sección, se introducen varios modelos de proceso muy generales (algunas veces llamados paradigmas de proceso) y se presentan desde una perspectiva arquitectónica, vemos el marco de trabajo del proceso, pero no los detalles de actividades específicas.

Estos modelos generales no son descripciones definitivas de los procesos del software. Más bien, son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. (SOMMERVILLE, 2005)

### 2.4.3. Modelo del proceso incremental.

El modelo incremental combina elementos de los flujos de proceso lineal y paralelo, el modelo incremental aplica secuencias lineales en forma escalonada a medida que avanza el calendario de actividades. Cada secuencia lineal produce “incrementos” de software susceptibles de entregarse de manera parecida a los incrementos producidos en un flujo de proceso evolutivo.

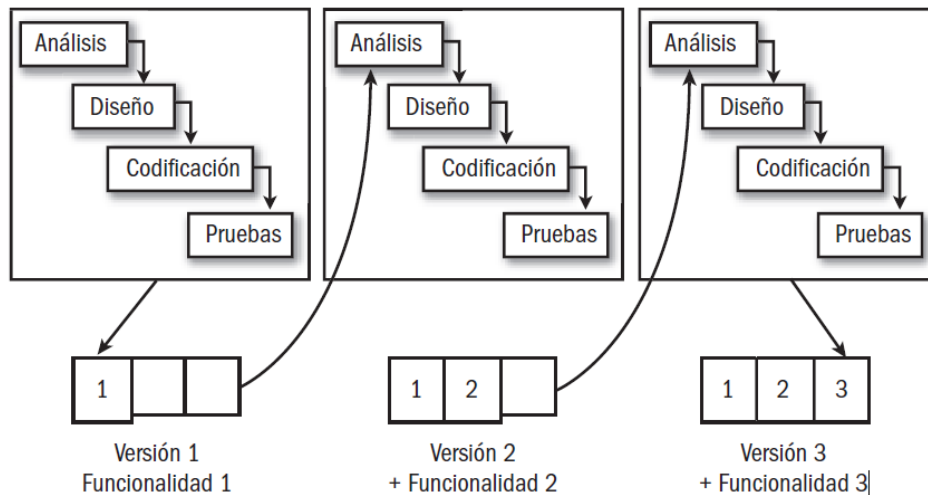
Cuando se utiliza un modelo incremental, es frecuente que el primer incremento sea el producto fundamental. Es decir, se abordan los requerimientos básicos, pero no se proporcionan muchas características suplementarias. El cliente usa el producto fundamental. Como resultado del uso y/o evaluación. (Pressman, 2010). El modelo del ciclo de vida incremental nos genera algunos beneficios tales como:

- Construir un sistema pequeño siempre es menos riesgoso que construir un sistema grande.
- Como desarrollamos independientemente las funcionalidades, es más fácil relevar los requerimientos del usuario.
- Si se detecta un error grave, solo desecharnos la última iteración.
- No es necesario disponer de los requerimientos de todas las funcionalidades en el comienzo del proyecto y además facilita la labor del desarrollo con la conocida filosofía de divide y vencerás.

Este modelo está orientado a cierto tipo de usuario o cliente. Podremos utilizar este modelo para casi cualquier proyecto, pero será verdaderamente útil cuando el usuario necesite entregas rápidas.

El modelo de proceso incremental se divide en cuatro etapas:

- Análisis
- Diseño
- Codificación
- Pruebas



**Figura 2.3** Grafico de Modelo Incremental.

Fuente: (Pressman, 2010)

## 2.5. Metodología UWE ( Uml-based Web Engineering)

La propuesta de Ingeniería Web basada en UML es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.



**Figura 2.4** UML- Based Web Engineering

Fuente: (LMU, 2016)

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

Otras características relevantes del proceso y método, son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

UWE, es un proceso interactivo e incremental, que se complementa con los procesos o fases de la metodología UML y comparten ciertos elementos de los diagramas que se utilizan en el modelo del diseño; en otras palabras se puede decir que es una extensión que apoya al diseño basado en la web que cubre la navegación, presentación, proceso de negocio y los aspectos de adaptación, de esta forma describe la metodología UWE el Instituto de Informática, Ludwig Maximilians University Múnich (2012), así también define las siguientes fases y diagramas.

El modelo que propone UWE está compuesto por etapas o submodelos:

- Modelo de Casos de Uso
- Modelo de Contenido
- Modelo de Usuario
- Modelo de estructura
- Modelo Abstracto
- Modelo de Adaptación
- Modelo de flujo de presentación.
- Modelo de ciclo de vida del objeto.

### **2.5.1. Fases de la Metodología UWE.**

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

- **Captura, análisis y especificación de requisitos:** En simples palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

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

- **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.
- **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.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **La Instalación o Fase de Implementación:** Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

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

- **El Mantenimiento:** Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de

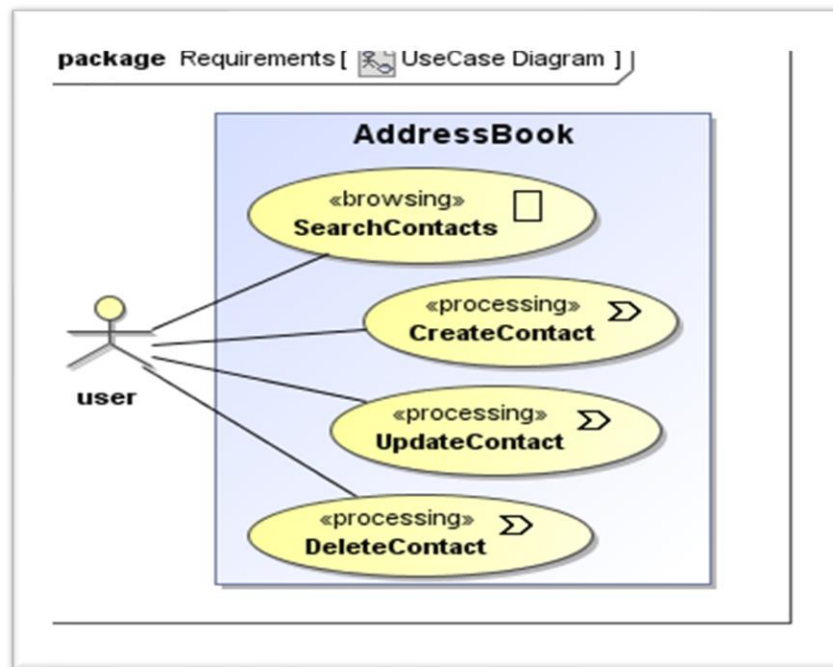
errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (QUIROGA, 2015)

### 2.5.2. Modelos UWE.

En esta sección se explicarán los modelos para cada uno de los aspectos web que cubre la metodología UWE.

- **Modelo de requisitos.**
  - a) **Modelo de casos de uso.**

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



**Figura 2.5** Modelo de casos de uso

Fuente: (LMU, 2016)

## b) Modelo de Actividades.

Durante ingeniería de requisitos es usual determinar qué datos son representados donde y cuando. Para modelar grupos de presentación en UWE son usados el estereotipo «display Action», mientras que los dos pines de acción estereotipados «interaction Pin» y «display Pin» son usados para modelar la entrada y la salida de datos.

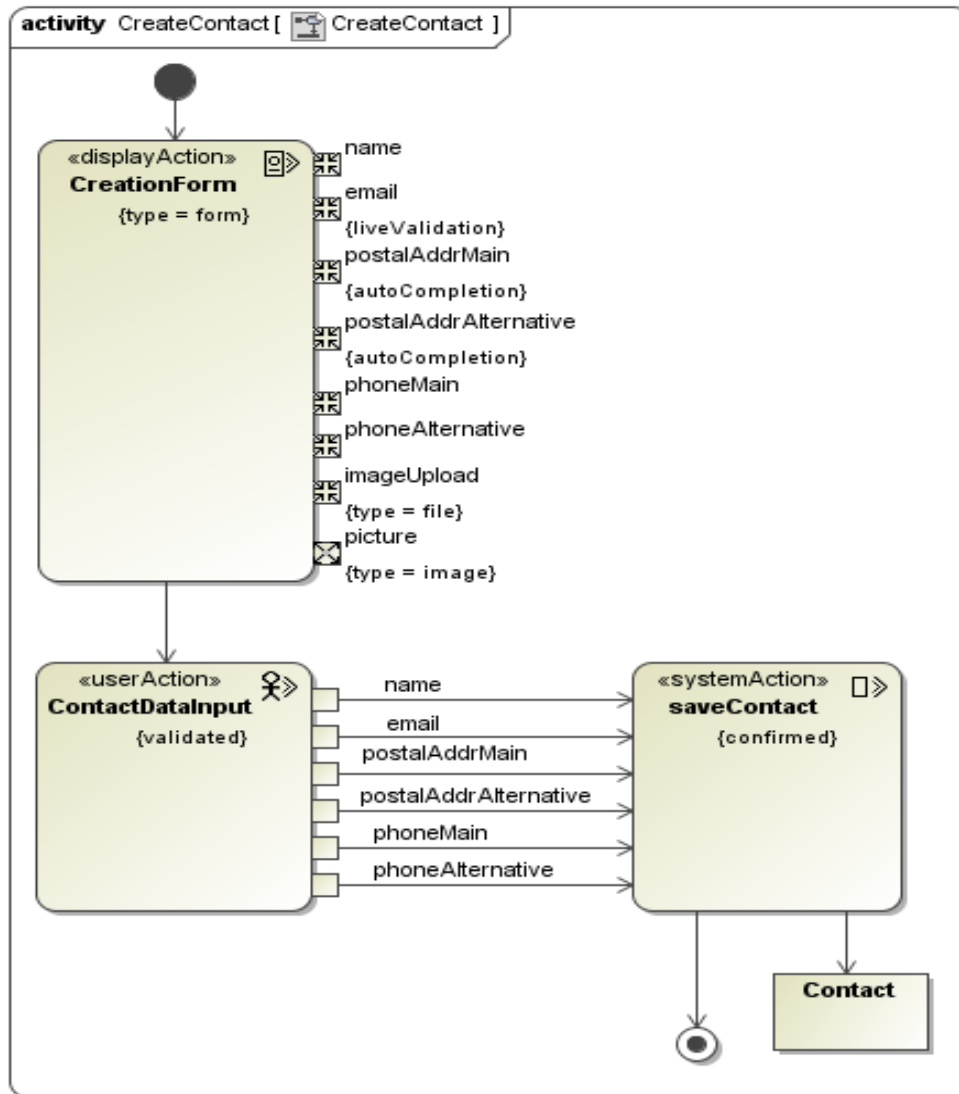


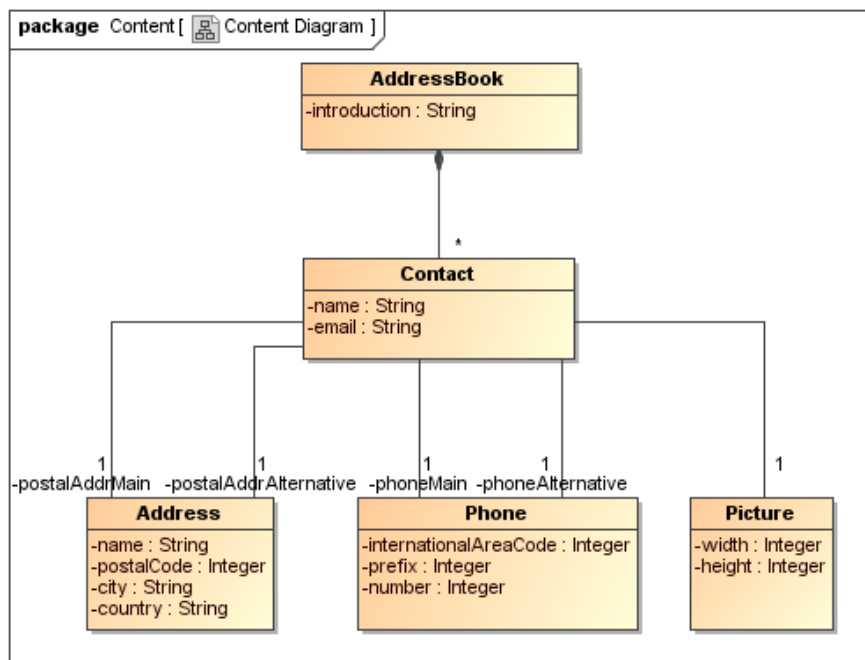
Figura 2.6 Modelo de Actividades.

Fuente: (LMU, 2016)



- **Modelo de contenido.**

Este es un diagrama UML normal de clases, por ello debemos pensar en las clases que son necesarias para nuestro ejemplo. Primero queremos disponer de una clase agenda ("AddressBook") conteniendo un conjunto de contactos. Cada contacto debe contener un nombre, y puede contener una dirección de correo, dos números de teléfono y dos direcciones postales. El nombre y la dirección de correo son Strings, el teléfono y la dirección postal son clases que representan más información.

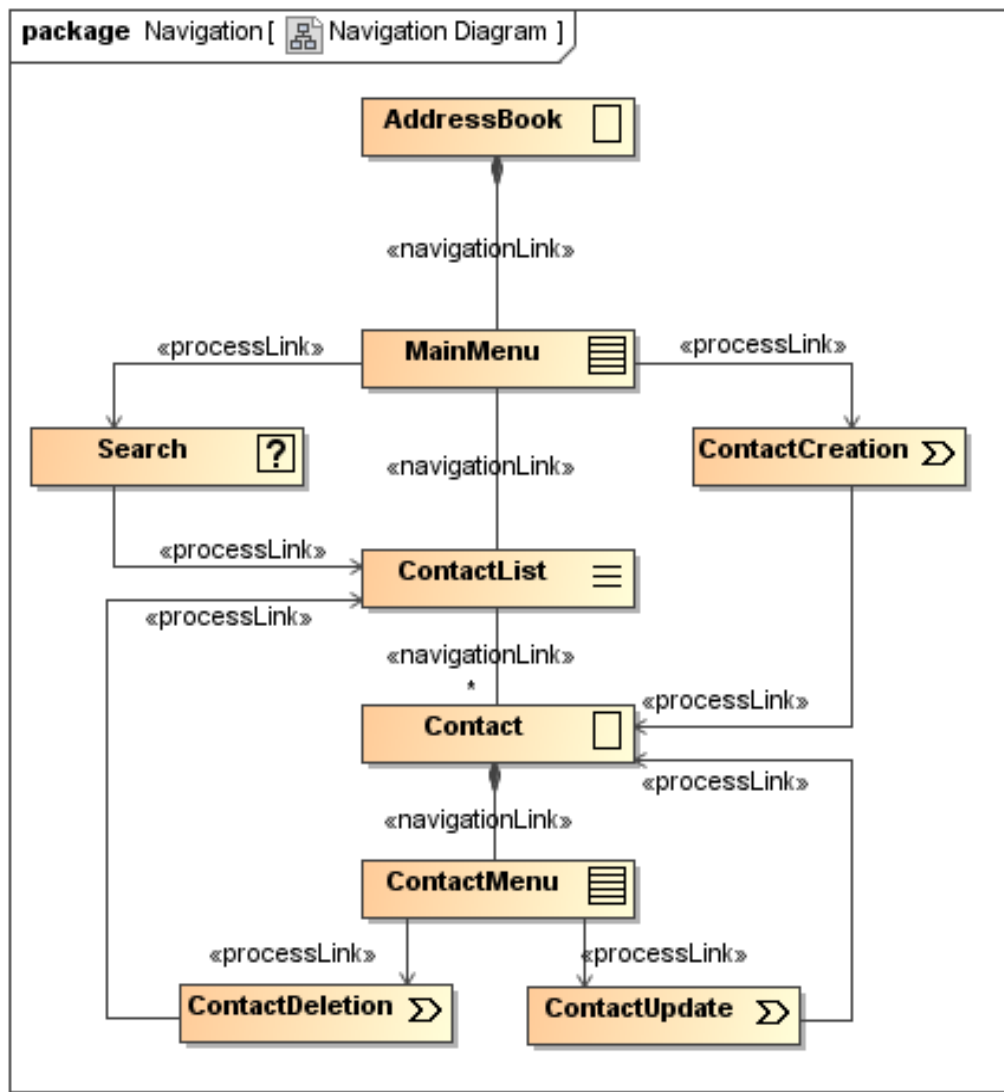


**Figura 2.7 Modelo de Contenido**

**Fuente:** (LMU, 2016)

- **Modelo de navegación.**

UWE provee diferentes estereotipos, los que presentaremos mediante nuestro ejemplo. La forma más simple de obtener un Diagrama de Navegación básico es utilizando la Transformación Content to Navigation. En este caso obtenemos para nuestro ejemplo un diagrama que contiene más nodos de los necesarios.



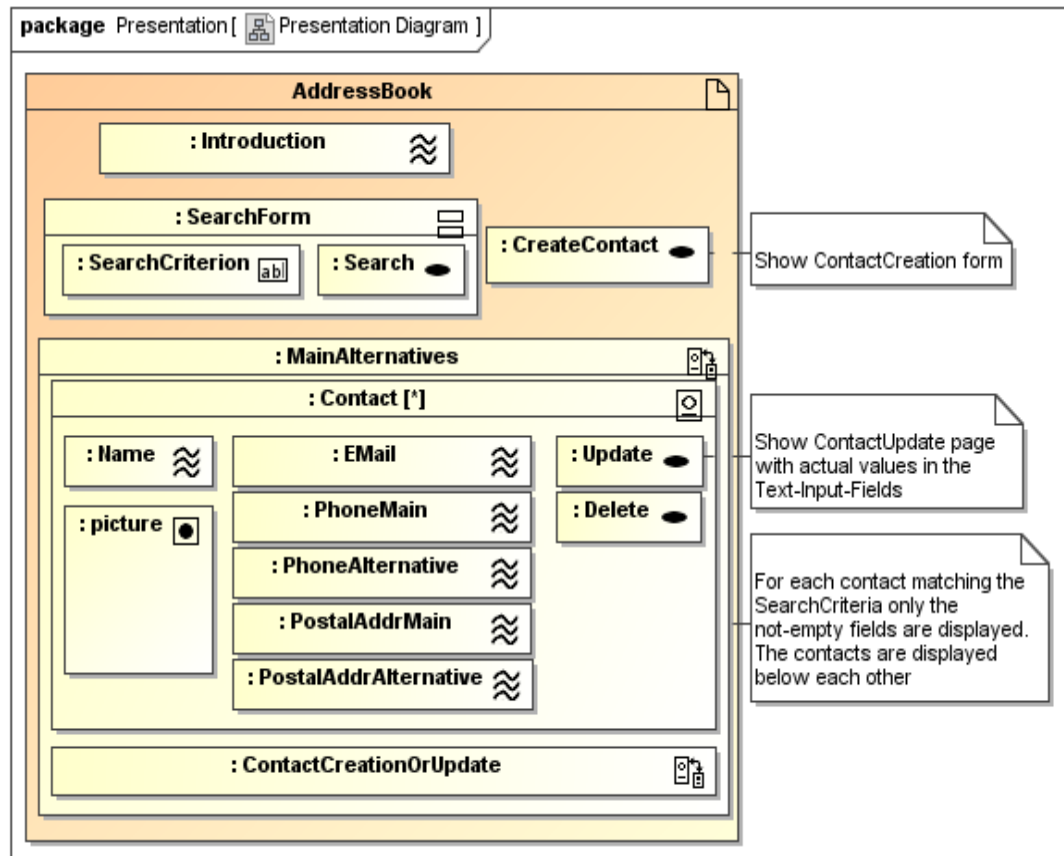
**Figura 2.8** Modelo de Navegación.

**Fuente:** (LMU, 2016)

- **Modelo de presentación.**

Diagrama de Presentación agrega una «presentationPage» class y agrega las propiedades con los estereotipos de UWE en ellos para expresar, que el elemento está ubicado en una página web. Las propiedades pueden anidarse, cada contacto («presentationGroup»-property) 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.



**Figura 2.9** Modelo Vista Controlador

**Fuente:** (LMU, 2016)

### 2.5.3. Ventajas y Desventajas.

Las principales razones para el uso de los mecanismos de extensión de UML en lugar de una técnica de modelado de propiedad es la aceptación del UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio Web: el llamado perfil UML, y amplio apoyo de modelado visual por herramientas CASE UML existentes.

UWE utiliza "puro" notación UML y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones Web, es decir, sin las extensiones de cualquier tipo. Por las características Web, como nodos y enlaces

de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre la navegación, la presentación, los procesos de negocio y los aspectos de adaptación.

## **2.6. Método de Prueba del Software.**

La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos.

### **2.6.1. Definición caso de prueba.**

un caso de prueba es: el flujo, o sea la serie de pasos a ejecutar sobre el sistema, los datos de entrada, ya sean entradas proporcionadas por el usuario al momento de ejecutar, o el propio estado de la aplicación, y por último las salidas esperadas, el oráculo, lo que define si el resultado fue positivo o negativo.

El “caso de prueba” es la personalidad más famosa en el mundo del testing (Rodríguez, 2014).

Ésta debe incluir varios elementos en su descripción, y entre los que queremos destacar se encuentran:

- Flujo: secuencia de pasos a ejecutar
- Datos entrada
- Estado inicial
- Valor de respuesta esperado
- Estado final esperado

### **2.6.2. Prueba de Caja Blanca.**

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

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

### **2.6.3. Prueba de Caja Negra.**

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca (Pressman, 2010).

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

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

## **2.7. Métricas de Calidad de Software.**

### **2.7.1. Definición de Calidad.**

Los desarrolladores de software más experimentados estarán de acuerdo en que obtener software de alta calidad es una meta importante. Pero, ¿cómo se define la calidad del software?

En el sentido más general se define como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan (Pressman, 2010).

### **2.7.2. Métricas de Calidad Establecida por el Estándar ISO 9126.**

La ISO 9126 es un estándar internacional para la evaluación del software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software. Los requisitos del software constituyen el funcionamiento para medir la calidad. (Pressman, 1998).

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad:

- Funcionalidad.
- Confiabilidad.
- Usabilidad.
- Eficiencia.
- Facilidad de recibir mantenimiento.
- Portabilidad.

Igual que otros factores de la calidad del software estudiados en las subsecciones anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema. (Pressman,2010).

### **2.7.3. Fórmulas de Evaluación de Calidad.**

#### **2.7.3.1. Funcionalidad.**

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

La funcionalidad se divide en 5 criterios:

**Adaptabilidad:** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

**Exactitud:** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

**Interoperabilidad:** La capacidad del software de interactuar con uno o más sistemas específicos.

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

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

Para la medición del sistema, se determina las siguientes ecuaciones:

**Tabla 2.1**

*Ecuación de la Funcionalidad*

VARIABLES	ECUACIÓN
Punto de función	$PF = \text{Cuenta Total} \times (0.65 + 0.01 \times \sum Fi)$ <p>Donde:            Cuenta Total. Es la sumatoria del producto del factor de ponderación y valores de los parámetros.  <math>\sum Fi</math>: Es la sumatoria de los valores de ajuste de la complejidad.</p>
Punto de función Máximo	$PF_{max} = \text{Cuenta Total} \times (0.65 + 0.01 \times \sum Fi)$ <p>Donde:            Considerando el máximo valor de ajuste de complejidad  <math>\sum Fi = 70</math></p>
Funcionalidad	$\text{Funcionalidad} = \frac{P F}{P F_{max}}$

Fuente: (Evaluación de Software, 2017)

**Métricas Basado en la Función**

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

**Tabla 2.2**

*Factor de Ponderación*

PARÁMETROS DE MEDIDA		FACTORES DE PONDERACIÓN		
		SIMPLE	MEDIO	COMPLEJO
1	Número de Entrada de Usuario	3	4	6
2	Número de Salida de Usuario	4	5	7
3	Número de Peticiones de Usuario	3	4	6
4	Número de Archivos	7	10	15
5	Número de Interfaces Externas	5	7	10

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

**Tabla 2.3**

*Variable para el Cálculo de la Funcionalidad.*

CARACTERÍSTICAS	DESCRIPCIÓN
<b>Número de entrada de usuario</b>	Se cuenta cada entrada del usuario que proporcione al software diferentes datos para la aplicación. Las entradas deben ser distintas de las peticiones.



<b>Número de Salida del usuario</b>	Se encuentra cada salida que proporciona al usuario informático orientada a la aplicación. En este contexto se refieren a informes, pantallas, mensajes de error.
<b>Número de peticiones al usuario</b>	Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
<b>Número de archivo</b>	se cuenta cada archivo maestro lógico es decir una agrupación lógica de datos que puede ser una parte en gran base de datos o un archivo independiente.
<b>Número de interfaces externas</b>	Se cuenta todas las interfaces legibles por la máquina.

Fuente: (Pressman Roger S. 2010)

Posteriormente determina los valores de ajuste de complejidad, para ello se debe analizar las preguntas que se muestra en la siguiente Figura de acuerdo de Factor de escala.

**Tabla 2.4**

*Ajuste de la Complejidad*

Importancia	0%	20%	40%	60%	80%	100%	Fi
Escala							
	No	Influencia	Moderado	Medio	Significativo	Esencial	
Nº	0	1	2	3	4	5	
1	¿Requiere el sistema copia de seguridad y de recuperación fiable?						
2	¿Requiere comunicación de datos?						
3	¿Existen funciones de procesamientos distribuido?						

- 
- 4 ¿El rendimiento es crítico?
  - 5 ¿Será ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?
  - 6 ¿Requiere el sistema entrada de datos interactivo?
  - 7 ¿Requiere la entrada de datos interactivo que las transiciones de entradas se lleven a cabo sobre múltiples o variados operaciones?
  - 8 ¿Se actualizan los archivos maestros en forma interactiva?
  - 9 ¿Son complejos las entradas, las salidas, los archivos o peticiones?
  - 10 ¿Es complejo el procesamiento interno?
  - 11 ¿Se ha diseñado el código para ser reutilizables?
  - 12 ¿Están incluidos en el diseño la conversación y la instalación?
  - 13 ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
  - 14 ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?

TOTAL

---

Fuente: (Pressman R., 2010)

#### **2.7.4. Confiabilidad.**

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

La confiabilidad se divide en 4 criterios:

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

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

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

**Conformidad de la fiabilidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

Para calcular la fiabilidad del sistema se toma en cuenta el periodo de tiempo en el que se ejecuta y se obtiene muestras con la siguiente ecuación:

**Tabla 2.5**

*Ecuación de la Confiabilidad*

<b>VARIABLE</b>	<b>ECUACIÓN</b>
<b>Confiabilidad</b>	$F(t) = f * e^{-\mu * t}$ Donde: $f$ : Funcionalidad del sistema $\mu$ : Es la probabilidad de error que puede tener el sistema. $t$ : Tiempo que dura la gestión en el sistema.
<b>Probabilidad de operación</b>	$P(T \leq t) = F(t)$ Probabilidad de fallas (el termino en el cual el sistema trabaja sin fallas) $P(T \leq t) = 1 - F(t)$ Probabilidad de trabajo sin fallas (Tiempo en el cual no falla el sistema)

**Fuente: (Elaboración Propia basado en modelos de métricas de calidad, Jesús A. Guzmán, 2016)**

### 2.7.5. Usabilidad.

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

La usabilidad se divide en 4 criterios:

**Entendimiento:** La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

**Aprendizaje:** La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

**Operabilidad:** La manera como el software permite al usuario operarlo y controlarlo.

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

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

**Tabla 2.6**

*Ecuación de la Usabilidad*

VARIABLE	ECUACIÓN
Usabilidad	$FU = [(\sum Xi/n) * 100]$ Donde: Xi: Es la sumatoria de valores n: Es el número de preguntas

**Fuente: (Elaboración Propia basado en modelos de métricas de calidad, Jesús A. Guzmán, 2016)**

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

**Tabla 2.7**

*Valoración de las Preguntas*

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

**Fuente: (Elaboración Propia basado en modelos de métricas de calidad, Jesús A. Guzmán, 2016)**

**Tabla 2.8**

*Preguntas para Determinar la Usabilidad*

<b>NRO</b>	<b>PREGUNTAS</b>	<b>SI</b>	<b>NO</b>	<b>EVALUACIÓN</b>
1	¿Puede utilizar con facilidad del sistema?			X
2	¿Puede controlar operaciones que el sistema solicite?			X
3	¿El sistema permite la retroalimentación de información?			X
4	¿El sistema cuenta con interfaz agradable a la vista?			X
5	¿La respuesta del sistema es satisfactoria?			X
6	¿Le parece complicadas las funciones del sistema?			X
7	¿Los resultados que proporciona el sistema facilitan el trabajo?			X
8	¿Durante el uso del sistema se produjo errores?			X
Total				X

**Fuente: (Modelos de métricas de calidad, Jesús A. Guzmán, 2016)**

### 2.7.6. Mantenimiento.

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

El mantenimiento se divide en 5 criterios:

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

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

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

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

**Conformidad de facilidad de mantenimiento:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

**Tabla 2.9**

*Ecuación de la Mantenibilidad.*

VARIABLE	ECUACIÓN
<b>Mantenibilidad</b>	$IMS =  Mt - (Fa + Fc + Fd)  / Mt$ <p>Donde: Mt: Número de métodos de la versión actual Fc: Número de módulos en la versión actual que se han modificado. Fa: Número de módulos en la versión actual que se han añadido. Fd: Número de módulos de la anterior versión que se han borrado en la versión actual.</p>

**Fuente: (Modelos de métricas de calidad, Jesús A. Guzmán, 2016)**

### 2.7.7. Portabilidad.

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

La portabilidad se divide en 5 criterios:

**Adaptabilidad:** Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

**Facilidad de instalación:** La facilidad del software para ser instalado en un entorno específico o por el usuario final.

**Coexistencia:** La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.

**Reemplazabilidad:** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

**Conformidad de portabilidad:** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

### 2.7.8. Eficiencia.

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

La eficiencia se divide en 3 criterios:

**Comportamiento de tiempos:** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y

esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

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

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

## **2.8. MÉTODO DE ESTIMACIÓN DE COSTOS DEL SOFTWARE.**

Una de las tareas de mayor importancia en la administración de proyectos de software es la estimación de costos. Si bien es una de las primeras actividades, inmediatamente posterior al establecimiento de los requerimientos, se ejecuta regularmente a medida que el proyecto progresa con el fin de ajustar la precisión en la estimación.

En los 90, las técnicas de desarrollo de software cambiaron dramáticamente, surgieron la necesidad de reusar software existente, la construcción de sistemas usando librerías, etc. Estos cambios comenzaron a generar problemas en la aplicación del modelo COCOMO. La solución fue reinventar el modelo. Después de algunos años y de un esfuerzo combinado de USC-CSE (University of Southern California- Center For Software Engineering), IRUS at UC Irvine y organizaciones privadas, aparece COCOMO II. Las incorporaciones a este modelo lo reforzaron e hicieron apto para ser aplicado en proyectos vinculados a tecnologías como orientación a objetos, desarrollo incremental, composición de aplicación, y reingeniería. COCOMO II consta de tres modelos, cada uno de los cuales ofrece una precisión acorde a cada etapa de desarrollo del proyecto. Enunciados en orden creciente de fidelidad son, modelo de Composición de Aplicación, Diseño Temprano y Post Arquitectura.

### **2.8.1. Definición del modelo**

Los objetivos principales para construir el modelo COCOMO II son:



- 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.

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

Se puede afirmar que para las aplicaciones desarrolladas por usuarios finales no se justifica la utilización de un modelo de estimación de costos. Estas aplicaciones normalmente se construyen en poco tiempo, por lo tanto, requieren solamente una estimación basada en actividades.

### **2.8.2. Ecuación del Modelo.**

COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

a)  $E = a(KLDC)^b$ ; Persona – Mes

b)  $D = c(E)^d$ ; Meses

c)  $P = E/D$ ; Personas

#### **Dónde:**

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada submodelo.

KLDC: Cantidad de líneas de código distribuidas en miles.

A la vez cada modelo se subdivide en modos, los mismos son: [Calero, 2010]

**Modo orgánico:** Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).

**Modo semi – libre o semi acoplado:** Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

**Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

**Tabla 2.10**

*Detalle de Coeficiente de COCOMO II*

PROYECTO DEL SOFTWARE	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semi – Acoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.33

Fuente: (Pressman, 2010)

### **Puntos de Función No Ajustados**

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).

$$\mathbf{FP = UFP \times TCF}$$

**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 Interfase (Interfases): 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.

## 2.9. Herramientas de desarrollo.

### 2.9.1. Lenguajes De Programación.

- **JavaScript.**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y, por tanto, las páginas web comenzaban a incluir formularios complejos.

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.



**Figura 2.10 JavaScript**

Fuente: (Introduccion JavaScript, 2020)

- **jQuery.**

jQuery es un Framework JavaScript, pero quizás muchos de los lectores se preguntarán qué es un Framework. Pues es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar (Alvarez, 2015).

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript.



**Figura 2.11** jQuery

Fuente: (Web)

- **Bootstrap.**

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end, con la cual todo lo que tiene que ver con la presentación de la misma y que se logra mediante el buen uso de los lenguajes de programación con HTML + CSS (Estilos) + Javascript. Con el HTML lo que se hace es la estructura de nuestras páginas web, con el CSS se le da diferentes formas, colores, bordes y demás estilos a cada parte de nuestros sitios y Javascript es un lenguaje de programación que se ejecuta del lado del cliente y sirve para darle efectos visuales y funcionales a la página.

Bootstrap, originalmente llamado Blueprint de Twitter, fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un marco de trabajo (framework) para fomentar la consistencia entre las herramientas internas.

Es compatible con todos los navegadores habituales, su arquitectura está basada en LESS. Es de código abierto publicado en 2011 con licencia Apache. Los beneficios que ofrece Bootstrap son:

**Responsive web desing:** Diseño de páginas web para que el usuario las visualice perfectamente en un amplio rango de dispositivos (Navegador en el PC, tableta, Smartphone)

**Bootstrap 3:** Mobile first, se ha diseñado para adaptarse desde el principio a tamaños de pantalla típicos de móviles. Sistema grid fluido que escala a 12 columnas al aumentar el tamaño de viewport. (Pavón, 2014).



**Figura 2.12 Bootstrap**

Fuente: (Web)

- **Yii Framework.**

Yii es un framework de PHP de alto rendimiento, basado en componentes para desarrollar aplicaciones web modernas en poco tiempo. El nombre Yii significa "simple y evolutivo" en chino. También se puede considerar como el acrónimo de Yes It Is (que en inglés significa Sí, lo es).

Una de las principales razones por las que se eligió utilizar este framework para el proyecto es su estructura modelo-vista-controlador, que proporciona una clara organización garantizando un alto rendimiento y una gran escalabilidad.

Esta estructura tiene como objetivo separar la lógica de negocio de las consideraciones de la interfaz de usuario a fin de que los desarrolladores puedan modificar cada parte más fácilmente sin afectar a otra. El modelo representa la información (los datos) y las reglas de negocio; la vista contiene elementos de la interfaz de usuario como textos, formularios de entrada; y el controlador administra la comunicación entre la vista y el modelo.

- Yii implementa el patrón de diseño MVC (Modelo-Vista-Controlador) y promueve la organización de código basada en este patrón.
- Yii consiste en escribir el código de manera simple y elegante, sin sobre diseñar nunca por el hecho de seguir un patrón de diseño determinado.
- Yii es un framework completo (full stack) que provee muchas características probadas y listas para usar, como los constructores de consultas y la clase Active Record para las bases de datos relacionales y NoSQL, la compatibilidad con la arquitectura REST para desarrollar API, la compatibilidad de caché en varios niveles.
- Yii es extremadamente extensible, se puede personalizar o reemplazar prácticamente cualquier pieza de código de base, como se puede también aprovechar su sólida arquitectura de extensiones para utilizar o desarrollar extensiones distribuibles.



**Figura 2.13** Yii Framework

Fuente: (Web)

### **2.9.2. Sistema Gestor de Base De Datos.**

MySQL es la base de datos de código abierto más popular del mercado. Gracias a su rendimiento probado, a su fiabilidad y a su facilidad de uso, MySQL se ha convertido en la base de datos líder elegida para las aplicaciones basadas en web y utilizada por propiedades web de perfil alto, como Facebook, Twitter, YouTube y los cinco sitios web principales. Además, es una elección muy popular como base de datos integrada, distribuida por miles de ISV y OEM.



La base de datos al completo es diseñada y gestionada mediante el gestor de bases de datos que ofrece MySQL, pero las distintas consultas que deba realizar el cliente serán llevadas a cabo por el servicio web.

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones.



*Figura 2.14 MySQL*

Fuente: (Web)

- **Características de MySQL.**

El estándar MySQL constituye la base de datos de software libre más popular del mercado. Es desarrollada, distribuida y costeadada por el grupo de empresas MySQL AB, el cual fue fundado por los creadores de MySQL con el objetivo de perpetuar esta base de datos.

- a) MySQL es un sistema de gestión de base de datos que ofrece los mecanismos para añadir, acceder y procesar los distintos datos almacenados en una base de datos.
- b) MySQL ofrece una base de datos relacional en lenguaje SQL, la cual almacena los datos en tablas de datos separadas almacenadas todas ellas en un mismo espacio de almacenamiento. Este hecho ofrece unas buenas características de velocidad y flexibilidad.

- c) MySQL es software libre, lo cual significa que cualquiera puede hacer uso del código fuente que constituye a MySQL para usarlo libremente o incluso modificarlo sin restricciones.
- d) Los motivos por los que MySQL se ha hecho tan popular, además de lo expuesto anteriormente, residen en el hecho de que supone un Servidor de bases de datos cuyas características de velocidad, fiabilidad y facilidad de uso son extremadamente atractivas y competitivas respecto del resto de soluciones existentes en el mercado.
- e) El software de MySQL ofrece un modelo de cliente/servidor consistente en un servidor SQL multihilo que es capaz de soportar diferentes clientes, librerías, herramientas administrativas y APIs.

### **2.9.3. Servidor Web Apache.**

El servidor HTTP Apache es un servidor web de software libre desarrollado por la Apache Software Foundation (ASF). El producto obtenido de este proyecto es un servidor de código fuente completo, descargable y gratuito.

Apache es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores que dispone.

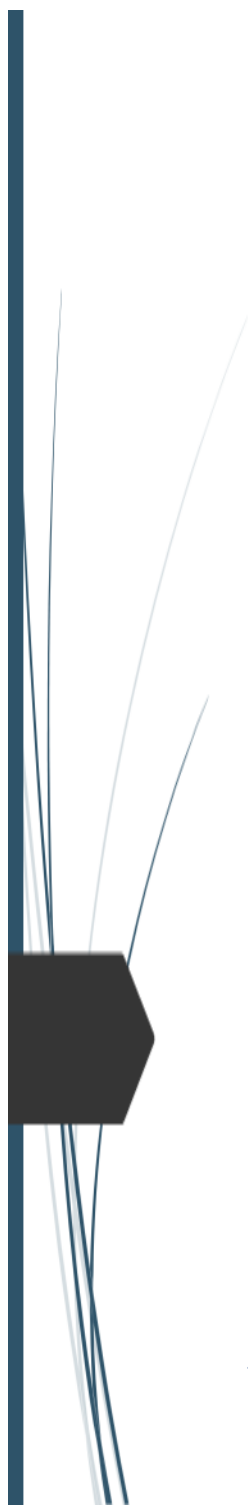


**Figura 2.15 Servidor Web Apache**

Fuente: (Web)

Es también un servidor estable, eficiente, extensible y multiplataforma.

- Estable: es una consecuencia de su probada robustez que impide caídas o cambios en el servidor inesperados.
- Flexible y eficiente: es capaz de trabajar con el estándar HTTP/1.1 (RFC2616) y con la mayor parte de las extensiones web que existen en la actualidad, como son los módulos PHP, SSL, CGI, SSI, proxy.
- Extensible: dispone de gran cantidad de módulos que amplían su funcionalidad.
- Multiplataforma: está disponible para diferentes plataformas como GNU/Linux, Windows, MacOS.



# **CAPITULO III**

## **MARCO**

### **APLICATIVO**

### **3. MARCO APLICATIVO**

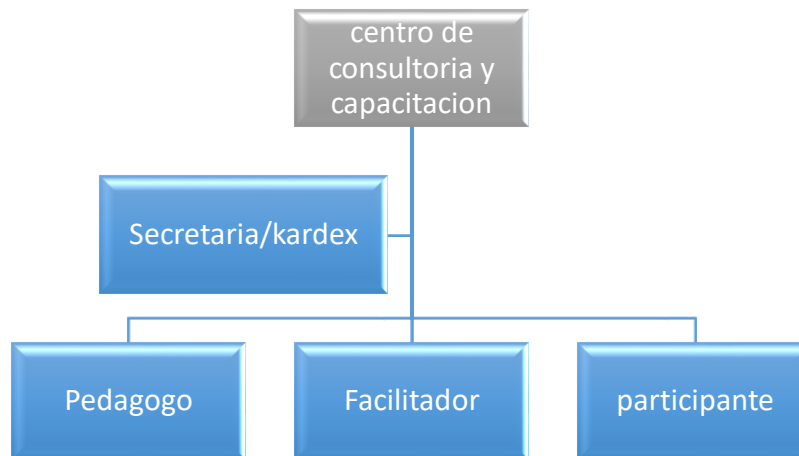
#### **3.1. Introducción**

En este capítulo, se realiza una breve descripción de la organización de la institución, y la función de usuarios para conocer el funcionamiento de cada proceso, para la aplicación de la metodología UWE y la construcción de la base de datos, su implementación aplicando las métricas de calidad y costos del software, el resultado principal es la construcción del sistema de control administrativo y seguimiento para consultoría y capacitación “NUAM”.

#### **3.2. Esquema del Organigrama.**

En esta primera etapa del desarrollo del software detectamos procesos elaborados manualmente en el manejo del control de pago de mensualidades, seguimiento del proceso de inscripción, programación de procesos de registro de participantes, así como el registro de puntos de capacitación de los mismos, procesos que ocasiona trabajos monótonos y repetitivos en el personal encargado (secretaria), haciendo que este tome más tiempo de lo necesario en estas actividades y provocando molestias en los estudiantes por las demoras en estos procesos.

La institución de consultoría y capacitación “NUAM” (Nuevo Amanecer) cuenta con la siguiente estructura organizacional.



**Figura 3.1 Organigrama de la consultoría y Capacitación “NUAM”**  
**Fuente: (Elaboración Propia)**

### **3.3. Análisis y Diseño de Requerimientos.**

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Los requerimientos reflejan y muestran las necesidades de los usuarios de un sistema, de esta manera pueda ayudar a resolver algún problema en el control de instituciones que requieran un sistema para su obtener resultados óptimos, hacer un pedido o encontrar información. El proceso de encontrar, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos.

Un requerimiento es simplemente una declaración abstracta de un servicio que se debe proporcionar el sistema. En el otra parte, es una definición detallada y formal de una función del sistema.

#### **3.3.1. Descripción general del escenario**

Para describir el escenario se presenta un análisis de los requerimientos identificados en las visitas al lugar de trabajo, en conversaciones, entrevistas y observación directa de los problemas. Haciendo un análisis de la especificación de requerimientos del software con el uso del estándar IEEE-830 se ha podido

identificar los requisitos de la institución para el desarrollo del producto software y la disponibilidad de los mismos.

#### ***3.3.1.1. Análisis de punto de partida.***

Para el punto de partida del diseño, es precisamente el análisis de la situación actual de los procesos que se realizan en la administración de Kardex, específicamente en el área de control de inscripción y seguimiento en la administración de su propia información, haciendo uso de procedimientos ya establecidos de acuerdo a sus necesidades.

La asignación de facilitadores de área de capacitación de la institución NUAM se registra de manera manual no teniendo un claro control de la inscripción y pago de mensualidades que ingresa a la institución, haciendo difícil el control y la realización de reportes de la existencia de distintas áreas en el mismo.

El proceso de registro es muy moroso ya que en un día de trabajo secretaria solo atiende a determinado número de estudiantes y facilitadores provocando en el peor de los casos 2 días para registro entre inscripciones y control de pagos.

Trabajos innecesarios por parte de secretaria al transcribir datos de estudiantes que son archivados en carpetas de manera manual.

Con base en las razones anotadas es necesario recalcar la importancia de contar con un sistema de información basado en la Web que proporcione soluciones a los problemas mencionados, utilizando de manera eficiente los procedimientos de registro, control de inscripciones y pagos administrativos y seguimiento para consultoría y capacitación.

#### ***3.3.1.2. Análisis de requisitos.***

Para el análisis de requisito con las características del proyecto planteado, constituyen los parámetros fundamentales para la identificación de requerimientos, sobre los mismos se debe hacer un análisis y estudio pertinente antes de elaborar el diseño de la aplicación.

Los requisitos globales identificados son las características con las que debe contar la aplicación para proporcionar un manejo fácil de la información para los usuarios que son de administración, secretaria y facilitadores. Estos requisitos globales hacen referencia a los siguientes puntos de interés:

- Entrega de información actualizada a dirección.
- Administración de la información de fácil acceso, y que la interfaz de usuario proporcione suficiente referencia en la presentación, respecto a los contenidos del sistema.
- Que las interfaces destinadas a la actualización de la información, sean sencillas y de fácil manejo de manera complementaria.
- Pueda ser visitado desde cualquier parte del mundo.
- Proporcione seguridad en el acceso, con autenticación, a la página personal.
- Usuario de Kardex, con la concesión de permiso de registrar, insertar datos, modificar, ver vista y ver reportes.
- Que la navegación del sistema sea sencilla y que el tiempo de resultados del software sea rápido.
- Que el acceso al área restringida sea con autenticación de usuario.

### **3.3.2. Requerimientos Funcionales.**

La especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente, se describe las interacciones entre el sistema y su entorno. Lo que significa que todos los servicios solicitados por el usuario deben estar definidos como se muestra en el siguiente recuadro.



**Tabla 3.1**

*Requerimientos Funcionales*

<b>N.º</b>	<b>REQUERIMIENTOS FUNCIONALES</b>
1	El acceso al sistema a través de un usuario y password para los distintos niveles de jerarquía según la institución.
2	La administración del sistema a cargo de director y encargado de Kardex.
3	El despliegue de los módulos en base a la necesidad del sistema.
4	El registro de los estudiantes y docente facilitador al sistema.
5	El despliegue de los registros de estudiante y docente facilitador mediante sus datos personales y materia a la que desea cursar.
6	La asignación de materia y área para cada estudiante y docente facilitador
7	Los reportes de la asignación de estudiantes y docente facilitador de la materia que está cursando.
8	Registro de inscripción de datos del estudiante
9	Reporte de la boleta de inscripción del estudiante con datos de la materia que cursa.
10	Historial del registro de estudiantes y docente facilitador que realizaron cursos en las distintas áreas.
12	Registro y reporte de pago de mensualidades del estudiante.
13	Reporte de estudiantes con mora en mensualidades.
14	Reportes de estudiantes y docentes actualizados.

Fuente: (Elaboración Propia)

**3.3.3. Requerimientos no funcionales**

Los requerimientos no funcionales, como su nombre lo sugieren, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, que no se relacionan de forma directa con el comportamiento funcional del sistema.

**Tabla 3.2**

*Requerimientos No Funcionales*

<b>N.º</b>	<b>REQUERIMIENTO NO FUNCIONAL</b>
1	El sistema de funcionar correctamente en cualquier ordenador que tenga disponible la conexión a internet desde los navegadores.

- 
- 2 El ingreso al sistema debe ser para un uso sencillo en su navegación para facilitar el manejo a la administración.
  - 3 El acceso debe ser restringido mediante autenticación de usuario y contraseña.
  - 4 Se debe evitar que el personal no autorizado pueda acceder a datos privados del sistema.
  - 5 El URL del sistema debe ser encriptado para que o pueda ser visible para posibles accesos no autorizados.
- 

Fuente: (Elaboración Propia)

### **3.4. Desarrollo Metodología UWE**

En la aplicación de la metodología UWE se plasma el análisis del modelo de requerimientos del sistema para realizar el diseño de los casos de uso, diseño conceptual, diseño de navegación, modelo de contenido y modelo de presentación en la cual se describe el comportamiento del sistema de esta manera conocer las acciones de los actores y el funcionamiento del sistema.

#### **3.4.1. Modelo de requerimientos.**

Una de las primeras actividades para la construcción de aplicaciones web es la identificación de los requisitos, en la metodología UWE el modelo de requerimientos involucra desarrollar el modelado de casos de uso con UML.

##### **3.4.1.1. Casos de Uso**

El análisis del caso de uso muestra el sistema y el exterior, generalmente, los actores están fuera del sistema, mientras que los casos de uso están dentro de él.

Un diagrama de casos de uso, muestra la relación entre los actores y los casos de uso del sistema, los actores representan a los usuarios y otros sistemas que interaccionan con el sistema Los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta desde un actor.

##### **➤ Identificación de actores**

**DIRECTOR ADMINISTRATIVO:** La dirección de la gestión académica ha de ser cuidadosamente manejada ya que requiere la revisión y actualización de sus procesos en forma permanente. El director tiene la mayor responsabilidad en la complejidad dinámica de la consultoría.

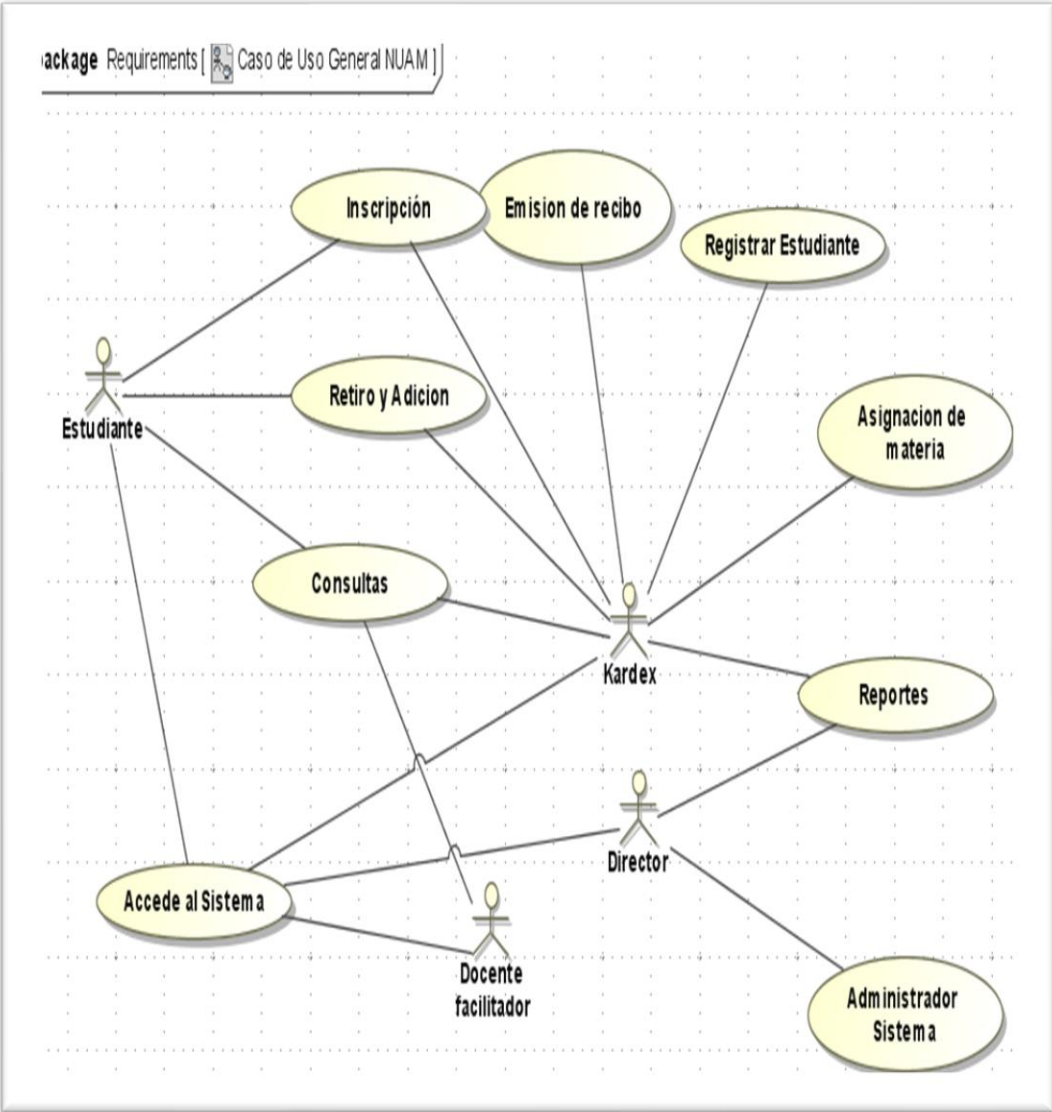
**RESPONSABLE DE ARCHIVO KARDEX:** Organizar y controlar los archivos y Kardex para la fluidez de los diferentes requerimientos de trámite de inscripción, certificación e información.

Por lo que se registra y archiva todos los datos de las actas oficiales de cada gestión académica. Garantizando su seguridad, acceso y uso apropiado de todas las consultorías académicas.

**FACILITADOR:** En este rol el Docente facilitador es un mediador entre los alumnos y el contexto, su papel es orientar e incentivar a los Estudiantes para que desarrollen competencias, con capacidades para interiorizar los diferentes elementos que intervienen en el proceso educativo; el Docente como mediador facilita la interacción para que el grupo participe en Actividades de análisis y síntesis sustentadas en una acción reflexiva sobre lo realizado y lo que se puede realizar.

**ESTUDIANTE:** Es asistir, participar puntualmente a clases y cumpliendo sus Actividades Pedagógicas para una capacitación óptima.

➤ **Diagramas de caso de uso general**



**Figura 3.2** Diagrama de caso de uso general

Fuente: (Elaboración propia)

➤ **Descripción de caso de uso.**

**Tabla 3.3**

*Descripción de caso de uso “Administrador del Sistema”*

Caso de Uso	Administrador del Sistema
-------------	---------------------------

Actor	Director
Tipo	Primario
Descripción	El sistema es manejado por el director de la Institución en la cual registrar, modificar y hasta eliminar algún dato de ser necesario. Como también tiene la autoridad de ver todos los reportes que se realizaron de cada gestión académica.

**Fuente: (Elaboración Propia)**

**Tabla 3.4**

*Descripción de Caso de Uso “Reportes”*

<b>Caso de Uso</b>	<b>Reportes</b>
Actor	Director / Kardex
Tipo	Primario
Descripción	El sistema es manejado por el director y Kardex de manera que el sistema genera reportes en PDF y EXCEL de los registros de estudiantes, docentes facilitadores, y las asignaciones de materias.

**Fuente: (Elaboración Propia)**

**Tabla 3.5**

*Descripción de Caso de Uso “Asignación de Materia”*

<b>Caso de Uso</b>	<b>Asignación de Materia</b>
Actor	Kardex
Tipo	Primario
Descripción	El sistema es manejado por el encargado de Kardex, el cual asigna la materia llenado datos del tipo de área que corresponde.

**Fuente: (Elaboración Propia)**

**Tabla 3.6**

*Descripción de Caso de Uso “Registrar Estudiante”*

<b>Caso de Uso</b>	<b>Registrar Estudiante</b>
Actor	Kardex
Tipo	Primario
Descripción	El sistema es manejado por el encargado de Kardex, el cual registra al estudiante con sus datos personales de los estudiantes el cual esta enlazado con la base de datos para que tiene registrado para futuras inscripciones en las distintas áreas.

**Fuente: (Elaboración Propia)**

**Tabla 3.7**

*Descripción de Caso de Uso “Emisión de Recibo”*

<b>Caso de Uso</b>	<b>Emisión de recibo</b>
Actor	Kardex
Tipo	Primario
Descripción	El sistema es manejado por el encargado de Kardex, el cual asigna un recibo con los datos del estudiante y el tipo de área que corresponde.

**Fuente: (Elaboración Propia)**

**Tabla 3.8**

*Descripción de Caso de Uso “Inscripción”*

<b>Caso de Uso</b>	<b>Inscripción</b>
Actor	Kardex / Estudiante

Tipo	Primario
Descripción	El sistema realiza una pre inscripción mediante la web a cualquier hora y asigna el cupo de la materia que un estudiante realiza.

**Fuente: (Elaboración Propia)**

**Tabla 3.9**

*Descripción de Caso de Uso “Retiro y Adición”*

<b>Caso de Uso</b>	<b>Retiro y Adición</b>
Actor	Kardex / Estudiante
Tipo	Primario
Descripción	El sistema realiza el retiro y adición de materia una vez que el estudiante tiene su contraseña mediante la web a cualquier hora la materia que un estudiante deja de cursar o adicionar.

**Fuente: (Elaboración Propia)**

**Tabla 3.10**

*Descripción de Caso de Uso “Consultas”*

<b>Caso de Uso</b>	<b>Consultas</b>
Actor	Kardex / Estudiante / Docente
Tipo	Primario
Descripción	El sistema realiza las consultas de las materias disponibles, información de la institución, a cualquier hora que un estudiante, docente o Kardex realice su consulta.

**Fuente: (Elaboración Propia)**

**Tabla 3.11**

*Descripción de Caso de Uso "Acceso al Sistema"*

<b>Caso de Uso</b>	<b>Acceso al Sistema</b>
Actor	Kardex / Estudiante / Docente / Administrador
Tipo	Primario
Descripción	El Acceso al sistema se realiza con un usuario y password que se le asigno al usuario y super usuario.

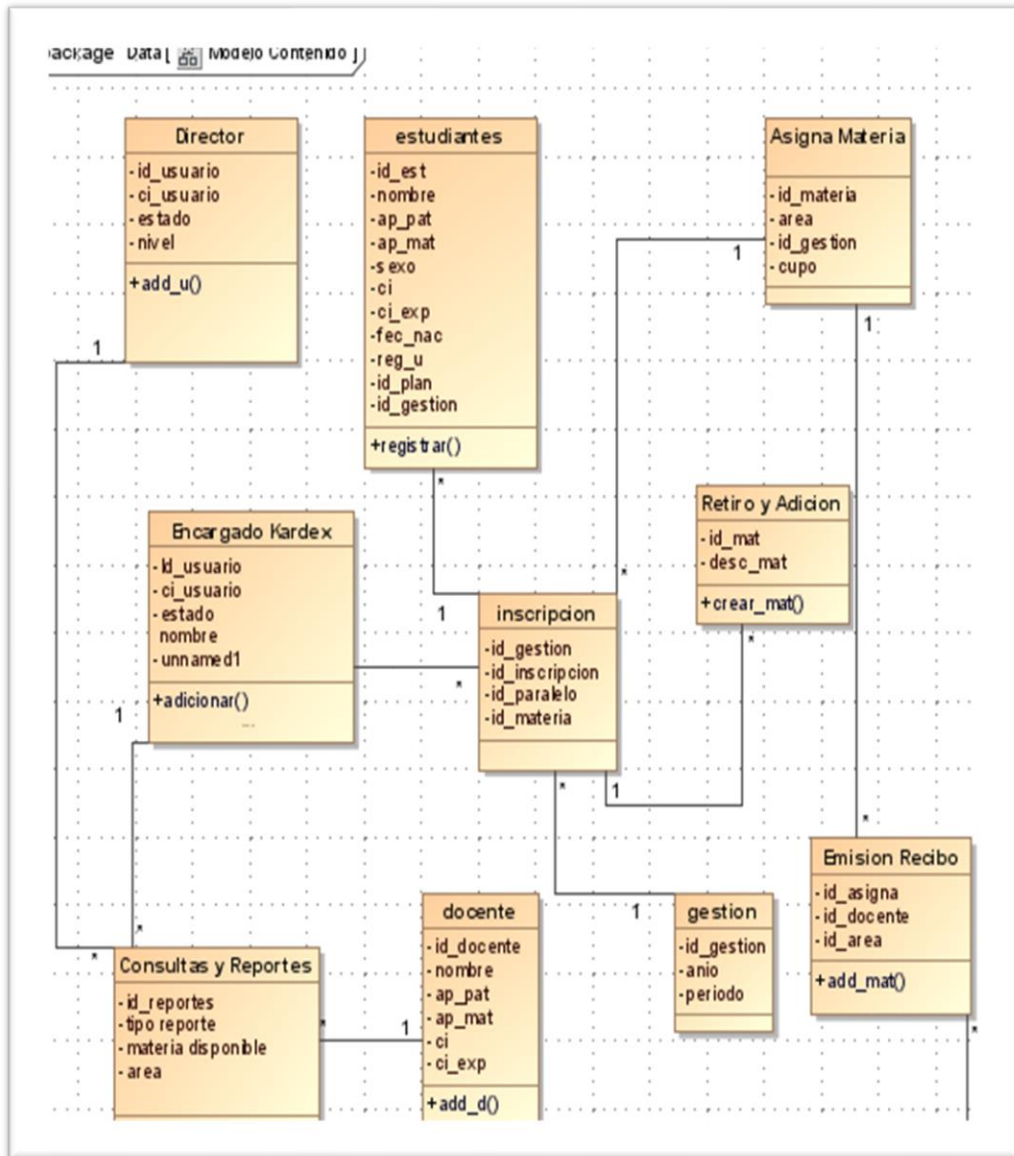
**Fuente: (Elaboración Propia)**

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

El objetivo del modelo de contenido es proporcionar, mostrar las especificaciones de la información relevante para la aplicación de la metodología.

UML proporciona un diagrama de clase, donde la cual aporta una visión estática o de estructura para el diseño del sistema, mostrando la naturaleza dinámica de comunicación entre ellas, que son los iconos utilizados para representar clases e interfaces.



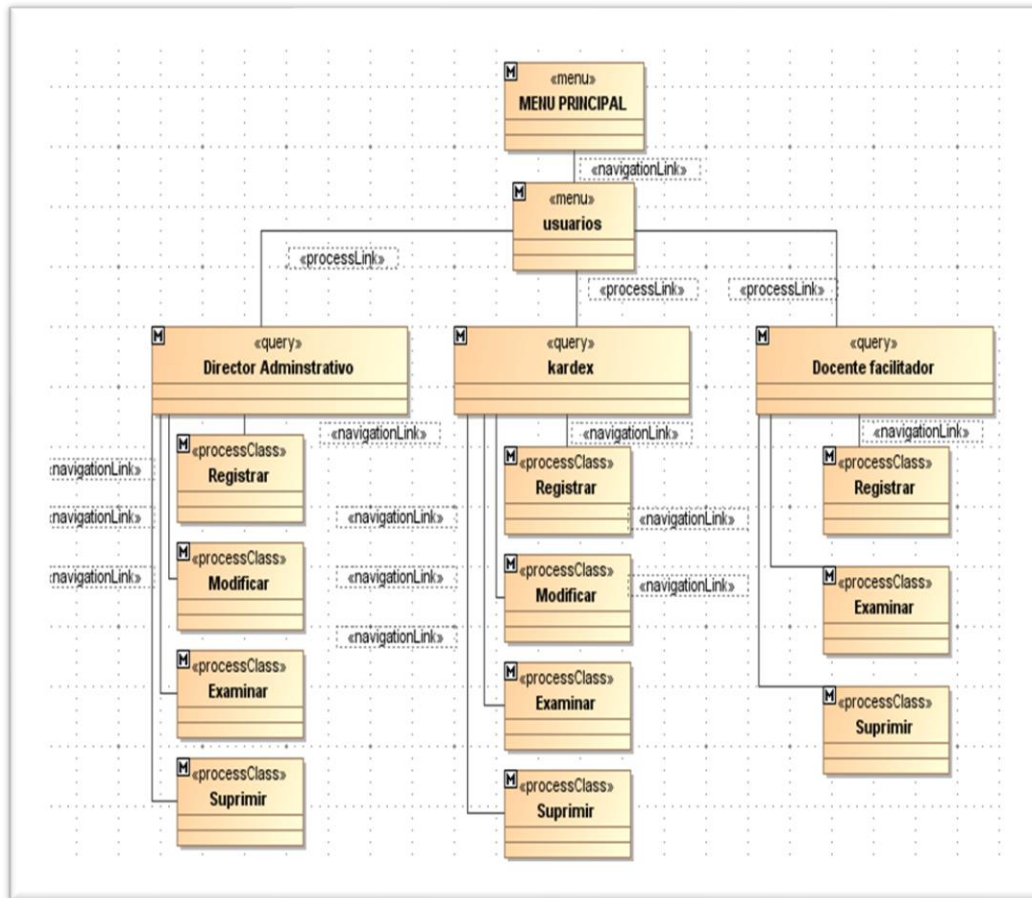


**Figura 3.3 Modelo de Contenido**

Fuente: (Elaboración Propia)

### 3.4.3. Modelo de navegación

Este diagrama se modela en base al análisis de los requisitos y el modelo de contenido el modelo requiere de un diagrama de navegación que contenga nodos y enlaces para conocer como están las paginas enlazadas en el sistema.

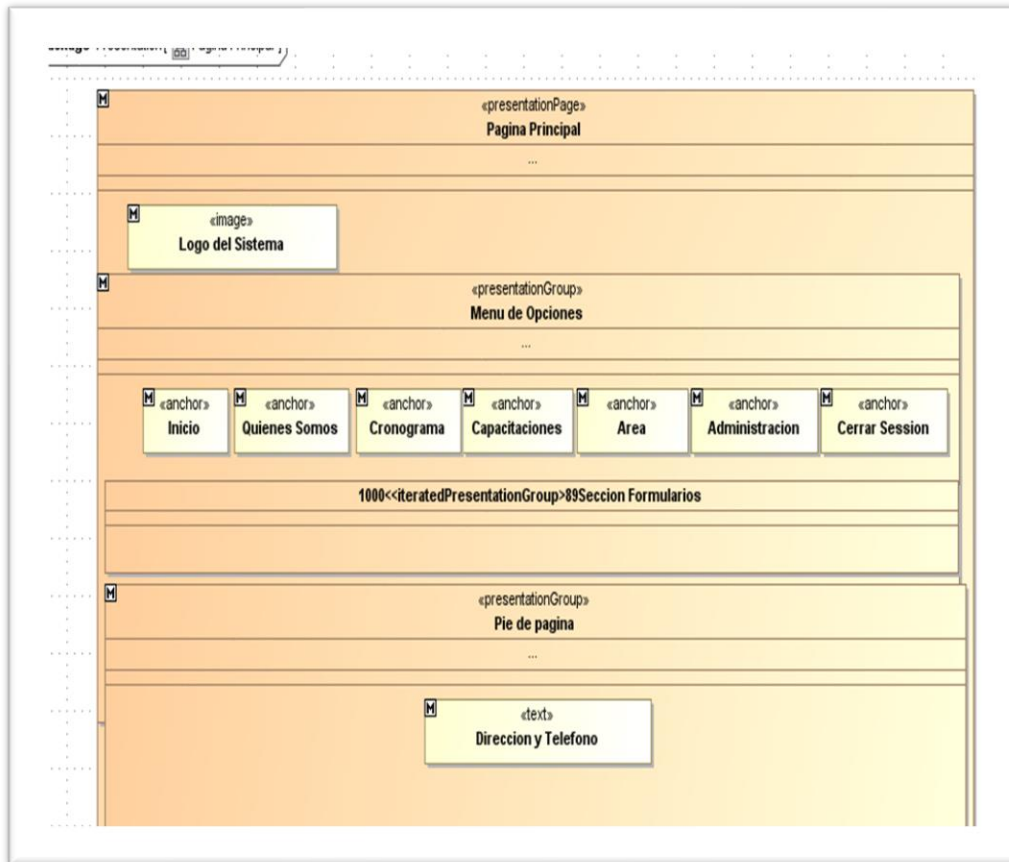


**Figura 3.4 Modelo de Navegación**

Fuente: (Elaboración Propia)

### 3.4.4. Modelo de presentación

El modelo de presentación tiene una visión general de la interfaz de usuario, describe la estructura de la interfaz que pueden contener texto, imágenes, enlaces, formularios utilizadas para representar la forma de navegación.



**Figura 3.5 Modelo de Presentación**

Fuente: (Elaboración Propia)

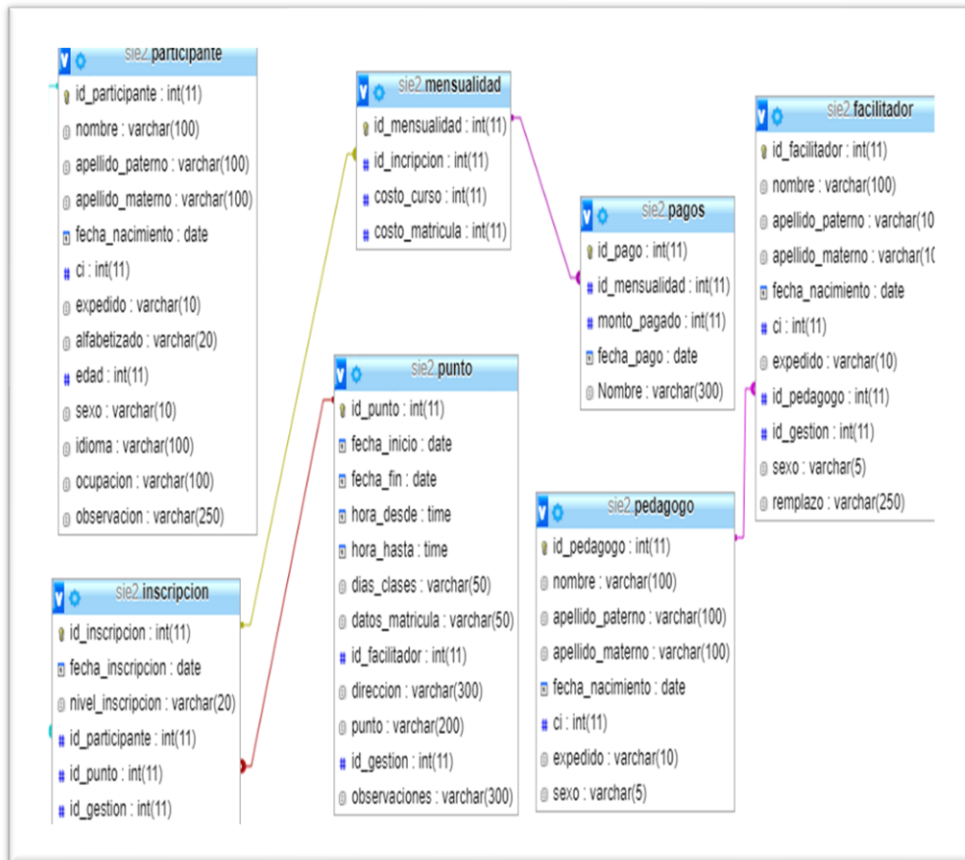
### 3.5. Implementación del sistema

#### 3.5.1. Base de Datos

Con los requerimientos del sistema ya establecidos y revisados para la institución se estructura las tablas relacionales de la base de datos, a continuación, el modelo de la Base de Datos.

#### 3.5.2. Modelo de la Base de Datos

Una vez revisado los requerimientos para el sistema (Caso de Uso), las entidades encontradas para el sistema de Consultoría y Capacitación. Entonces la estructura relacional de la Base de Datos generada para el sistema se puede observar en la figura.



**Figura 3.6** Modelo de la Base de Datos

**Fuente:** (Elaboracion Propia)

### 3.5.3. Especificaciones de Requerimiento para la Aplicación

A continuación se presenta un análisis de los requerimientos para la aplicación del sistema como propósito de resolver los problemas inherentes para la validación, especificación para los requisitos del sistema, de tal manera gestionar un diseño que satisfaga los requisitos del usuario.

**Tabla 3.12**

*Requerimientos para la Aplicación*

<b>TIPO</b>	<b>NOMBRE</b>	<b>FACE</b>
Lenguaje de programación	PHP 5.0	Etapa de desarrollo de la aplicación
Gestor de base de datos	MySQL	Etapa de implementación de la aplicación
Servidor de aplicaciones web	Apache versión 2.0	Etapa de funcionamiento de la aplicación
Framework	Yii Framework Bootstrap jQuery	Funcionamiento de la aplicación

**Fuente: (Elaboración Propia)**

#### **3.5.4. Módulo de Navegación**

Para el diseño del sistema de la interfaz de toma en cuenta una interfaz amigable, visible que reconozca con el nombre de la institución que permite seleccionar acciones fácilmente, además de mostrar las necesidades y requerimientos del sistema. A continuación, se mostrarán las pantallas relevantes del sistema



**Figura 3.7** Interfaz de inicio

**Fuente:** (Elaboración Propia)

```

<center><h1>Centro de Capacitacion NUAM</h1>
<h4>Educando para un futuro mejor<br> Centro Nuevo Amanecer</h4></center>

<br />
<div class="row-fluid">
  <ul class="thumbnails center">
    <li class="span3">
      <div class="thumbnail">
        <center><h3>Actividades en Agenda</h3>

        <div class="round_background r-grey-light">
          
          </div></center>

          <center><p>Esta opcion muestra las actividades programadas como inspecciones, control de actividades
            etc...</p></center>
          </div>
        </li>
        <li class="span3">
          <div class="thumbnail">
            <center> <h3>Inscripciones</h3>

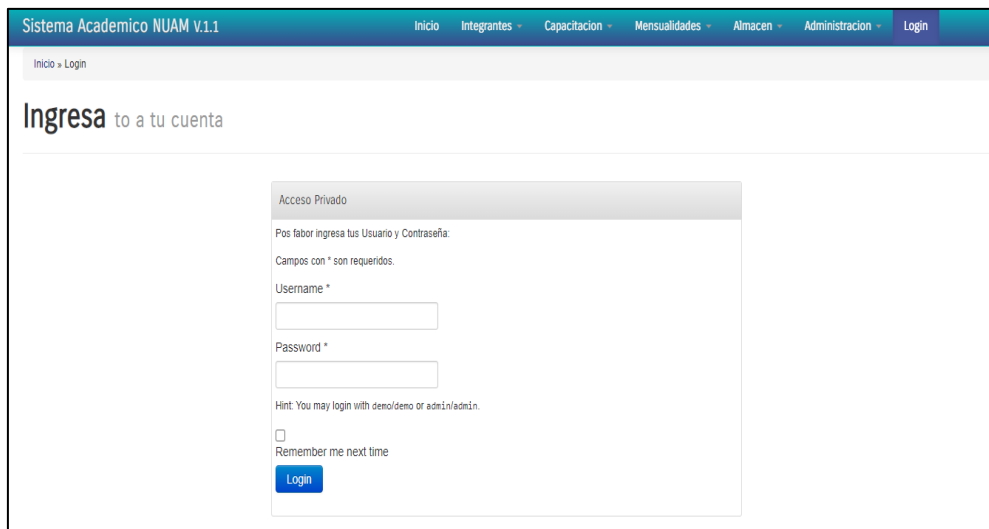
            <div class="round_background r-yellow">
              
              </div></center>

              <center><p>Realiza la inscripciones de la institucion en el area segun requerida</p></center>
            </div>
          </li>
          <li class="span3">
            <div class="thumbnail">

```

**Figura 3.8** Código Interfaz de Inicio

**Fuente:** (Elaboración Propia)



**Figura 3.9** Ingreso login

Fuente: (Elaboración Propia)

```

<p>Por favor ingrese su Usuario y Contraseña:</p>

<div class="form">
<?php $form->$this->beginWidget('CActiveForm', array(
    'id'=>'login-form',
    'enableClientValidation'=>true,
    'clientOptions'=>array(
        'validateOnSubmit'=>true,
    ),
)); >>

<p class="note">Campos con <span class="required">*</span> son requeridos.</p>

<div class="row">
    <?php echo $form->labelEx($model,'username'); ?>
    <?php echo $form->textField($model,'username'); ?>
    <?php echo $form->error($model,'username'); ?>
</div>

<div class="row">
    <?php echo $form->labelEx($model,'password'); ?>
    <?php echo $form->passwordField($model,'password'); ?>
    <?php echo $form->error($model,'password'); ?>
</div>

<div class="row rememberMe">
    <?php echo $form->checkbox($model,'rememberMe'); ?>
    <?php echo $form->label($model,'rememberMe'); ?>
    <?php echo $form->error($model,'rememberMe'); ?>
</div>

```

**Figura 3.10** Código Ingreso Login

Fuente: (Elaboración Propia)

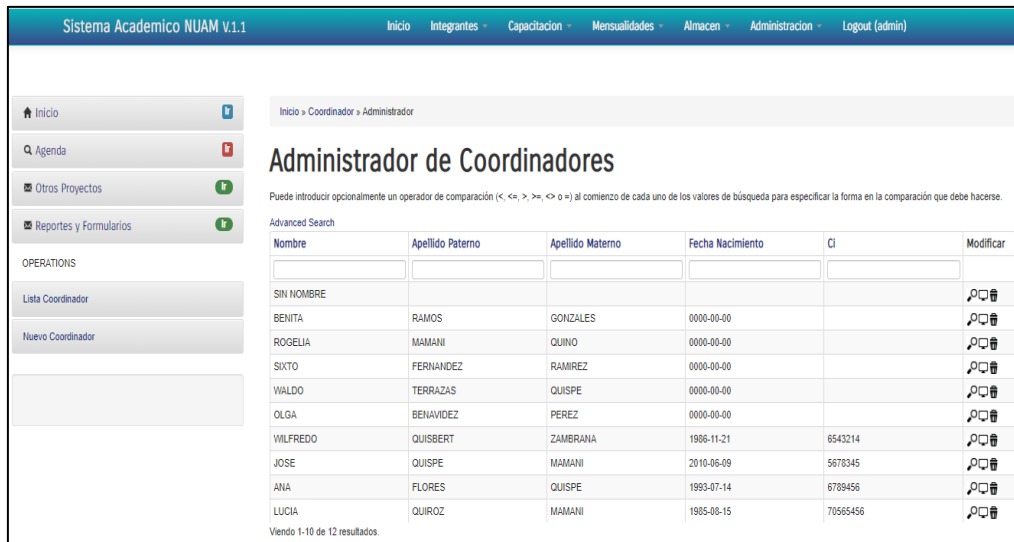


Figura 3.11 Módulo Coordinador

Fuente: (Elaboración Propia)

```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'pedagogo-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'template'=>"{items}{summary}{pager}",
    'filter'=>$model,
    'columns'=>array(
        'nombre',
        'apellido_paterno',
        'apellido_materno',
        'fecha_nacimiento',
        'ci',
        /*
        'expedido',
        */
        array(
            'class'=>'CButtonColumn',
            'header'=>'Modificar', //Titulo de la columna
            'template'=>'{view}{update}{delete}', //Habilitar los tres botones o uno o dos

            'viewButtonUrl'=>'Yii::app()->createUrl("pedagogo/view",array("id"=>$data->id_pedagogo))', //
            Accion del boton hacia donde va direccionado
            'updateButtonUrl'=>'Yii::app()->createUrl("pedagogo/update",array("id"=>$data->id_pedagogo))',//
            Accion del boton hacia donde va direccionado
            'deleteButtonUrl'=>'Yii::app()->createUrl("pedagogo/delete",array("id"=>$data->id_pedagogo))',//
            Accion del boton hacia donde va direccionado
            'deleteConfirmation'=>'Estas seguro de eliminar el elemento',// mensaje que lanza antes de borrar
            un elemento
            'viewButtonImageUrl'=>'Yii::app()->baseUrl.'/images/1.png',// imagen que aparece como icono en el
            boton
            'updateButtonImageUrl'=>'Yii::app()->baseUrl.'/images/2.png',// imagen que aparece como icono en el
        )
    )
);

```

Figura 3.12 Código Modulo Coordinador

Fuente: (Elaboración Propia)



Sistema Academico NUAM v.1.1 Inicio Integrantes Capacitacion Mersualidades Almacen Administracion Logout (admin)

Inicio » Coordinador » Adicionar

## Insertar Nuevo Pedagogo

Campos con \* Son requeridos.

Nombre \*

Apellido Paterno \*

Apellido Materno

Fecha Nacimiento

Ci

Expedido

**Figura 3.13** Adicionar Nuevo Coordinador

**Fuente:** (Elaboración Propia)

```

1 <?php
2 /* @var $this PedagogoController */
3 /* @var $model Pedagogo */
4
5 $this->breadcrumbs=array(
6     'Coordinador'=>array('index'),
7     'Adicionar',
8 );
9
10 $this->menu=array(
11     array('label'=>'Lista Coordinador', 'url'=>array('index')),
12     array('label'=>'Administrador Coordinador', 'url'=>array('admin')),
13 );
14 ?>
15
16 <h1>Insertar Nuevo Pedagogo</h1>
17
18 <?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.14** Código Adicionar Nuevo Coordinador

**Fuente:** (Elaboración Propia)



Figura 3.15 Módulo Docente

Fuente: (Elaboración Propia)

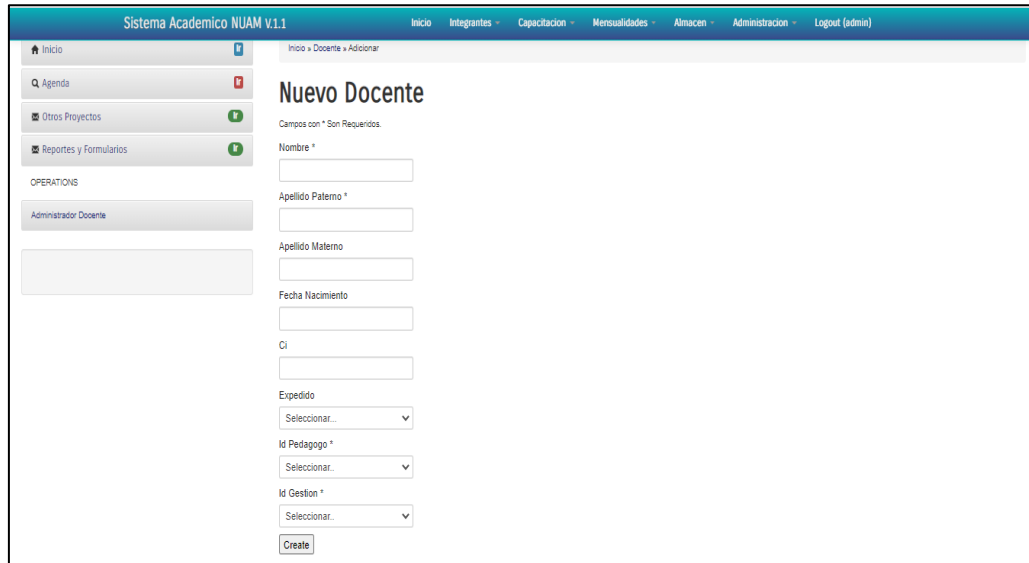
```

$this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'facilitador-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'template'=>"{items}{summary}{pager}",
    'filter'=>$model,
    'columns'=>array(
        'nombre',
        'apellido_paterno',
        'apellido_materno',
        // 'fecha_nacimiento',
        'ci',
        'id_gestion',
        // 'id_pedagogo',
        array(
            'name'=>'id_pedagogo',
            'header'=>'Nombre Coordinador',
            'value'=>'Pedagogo::model()->getNombreC($data->id_pedagogo)',
            'filter'=>CHtml::listData(Pedagogo::model()->findAll(), 'id_pedagogo', 'nombreCompleto'),
        ),
        /*
        'expedido',
        */
        array(
            'class'=>'CButtonColumn',
            'header'=>'Modificar', //Titulo de la columna
            'template'=>'{view}{update}{delete}', //Habilitar los tres botones o uno o dos
            'viewButtonUrl'=>'Yii::app()->createUrl("facilitador/view",array("id"=>$data->id_facilitador))', //
    )
);

```

Figura 3.16 Código Modulo Docente

Fuente: (Elaboración Propia)



**Figura 3.17** Adiciona Docente

**Fuente:** (Elaboración Propia)

```

index.php — themes\...site x login.php x admin.php — pedagogo x create.php — pedagogo x admin.php — facilitador x cr
<?php
/* @var $this FacilitadorController */
/* @var $model Facilitador */

$this->breadcrumbs=array(
    'Docente'=>array('index'),
    'Adicionar',
);

$this->menu=array(
    array('label'=>'Administrador Docente', 'url'=>array('admin')),
);
?>

<h1>Nuevo Docente</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.18** Código Adiciona Docente

**Fuente:** (Elaboración Propia)

Sistema Academico NUAM v.1.1 Inicio Integrantes Capacitacion Mensualidades Almacenes Administracion Logout (admin)

Inicio

Agenda

Otros Proyectos

Reportes y Formularios

OPERATIONS

Nuevo Participante

Inicio » Participantes » Manage

### Administrador de Participantes

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <= o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search




Nombre	Apellido Paterno	Apellido Materno	CI	Edad	Sexo	Inscrito	Modificar
NATIVIDAD	CALLISAVA	DE COPA	2217322	58	F	Codinas = 3	 
ELSA	CASI	TIPO DE SARZURI	2480168	48	F	Sin Inscritoon	 
MARGARITA	PEREZ	JIMENEZ	2609916	50	F	Sin Inscritoon	 
JUSTA	CANAVIRI	HUALLPA	2590168	48	F	Sin Inscritoon	 
NELLY SANTUSA	TORREZ	DE ARUQUIPA	2187931	59	F	Sin Inscritoon	 
TERESA	QUISPE	CALLIZAVA	4765149	50	F	Sin Inscritoon	 
PAULINA	SARZURI	BERNA	2427808	53	F	Sin Inscritoon	 
SALOME	TARQUI	FABIO	4247390	39	F	Sin Inscritoon	 
JUANA		MAYTA	5986359	37	F	Sin Inscritoon	 

Figura 3.19 Módulo Estudiante

Fuente: (Elaboración Propia)

```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'participante-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'template'=>"{items}{summary}{pager}",
    'filter'=>$model,
    'columns'=>array(
        // id participante',
        'nombre',
        'apellido_paterno',
        'apellido_materno',
        // 'fecha_nacimiento',
        'ci',
        'edad',
        'sexo',

        array(
            'name'=>'insc',
            'header'=>'Inscrito',
            'value'=>'Participante::model()->getVerificaI($data->id_participante)',
        ),
        array(
            'class'=>'CButtonColumn',
            'header'=>'Modificar', //Titulo de la columna
            'template'=>'{view}{update}{delete}', //Habilitar los tres botones o uno o dos

            'viewButtonUrl'=>'Yii::app()->createUrl("participante/view",array("id"=>$data->id_participante))',
            //Accion del boton hacia donde va direccionado
            'updateButtonUrl'=>'Yii::app()->createUrl("participante/update",array("id"=>$data->id_participante))',
            //Accion del boton hacia donde va direccionado
            'deleteButtonUrl'=>'Yii::app()->createUrl("participante/delete",array("id"=>$data->id_participante))',
            //Accion del boton hacia donde va direccionado
            'deleteConfirmation'=>'Estas seguro de eliminar el elemento',// mensaje que lanza antes de borrar
        )
    )
);

```

Figura 3.20 Código Modulo Estudiante

Fuente: (Elaboración Propia)

The screenshot shows a web application interface for adding a new participant. The page title is "Nuevo Participante". The form includes the following fields and controls:

- Nombre \*
- Apellido Paterno \*
- Apellido Materno
- Fecha Nacimiento
- CI
- Especialidad (dropdown menu)
- Alfabetizado (radio buttons for SI / NO)
- Sexo (radio buttons for Masculino / Femenino)
- Idioma (dropdown menu)
- Ocupacion (dropdown menu)
- Observacion (text area)
- Guardar button

**Figura 3.21** Adiciona Estudiante

**Fuente:** (Elaboración Propia)

```

<?php
/* @var $this ParticipanteController */
/* @var $model Participante */

$this->breadcrumbs=array(
    'Participantes'=>array('index'),
    'Create',
);

$this->menu=array(
    array('label'=>'Listar Participante', 'url'=>array('index')),
    array('label'=>'Administrar Participante', 'url'=>array('admin')),
);
?>

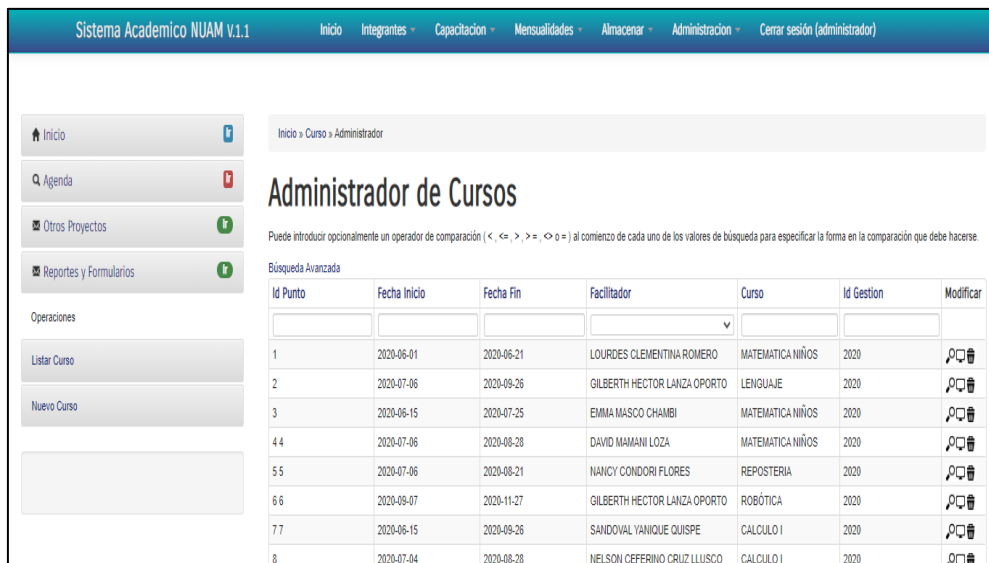
<h1>Nuevo Participante</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.22** Código Adiciona Estudiante

**Fuente:** (Elaboración Propia)



**Figura 3.23** Módulo Cursos

**Fuente:** (Elaboración Propia)

```

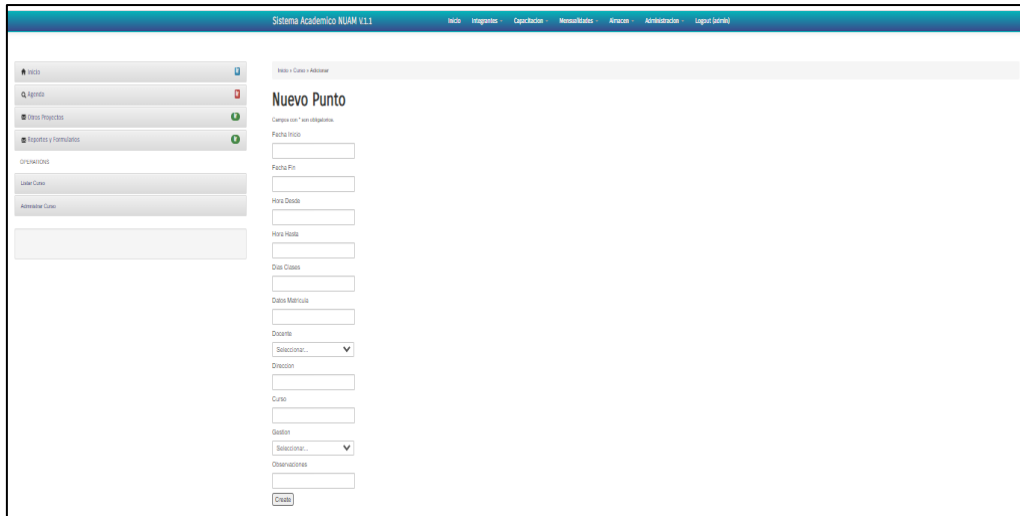
<?php echo CHtml::link('Advanced Search','#',array('class'=>'search-button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search',array(
    'model'=>$model,
)); ?>
</div><!-- search-form -->

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'punto-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'template'=>"{items}{summary}{pager}",
    'filter'=>$model,
    'columns'=>array(
        'id_punto',
        'fecha_inicio',
        'fecha_fin',
        array(
            'name'=>'id_facilitador',
            'header'=>'Facilitador',
            'value'=>'Facilitador::model()->getNombreC($data->id_facilitador)',
            'filter'=>CHtml::listData(Facilitador::model()->findAll(array('order'=>'nombre')), 'id_facilitador', 'nombreCompleto')
        ),
    ),
    array(
        'name'=>'punto',
        'header'=>'Curso',
    )
);

```

**Figura 3.24** Código Módulo Cursos

**Fuente:** (Elaboración Propia)



**Figura 3.25** Adiciona Cursos

**Fuente:** (Elaboración Propia)

```
2 /* @var $this PuntoController */
3 /* @var $model Punto */
4
5 $this->breadcrumbs=array(
6     'Curso'=>array('index'),
7     'Adicionar',
8 );
9
10 $this->menu=array(
11     array('label'=>'Listar Curso', 'url'=>array('index')),
12     array('label'=>'Administrar Curso', 'url'=>array('admin')),
13 );
14 ?>
15
16 <h1>Nuevo Punto</h1>
17
18 <?php $this->renderPartial('_form', array('model'=>$model)); ?>
```

**Figura 3.26** Código Adiciona Cursos

**Fuente:** (Elaboración Propia)

Sistema Academico NUAM v1.1 Inicio Integrantes - Capacitación - Mensualidades - Almacenar - Administración - Cerrar sesión (administrador)

Inicio » Inscripciones » Gestionar

## Inscripciones Administrativas

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <= o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Búsqueda Avanzada











Id Inscripción	Participante	Fecha Inscripción	Nivel Inscripción	Punto Asignado	Id Gestion	Modificar inscripción
1	BENIGNA YUJRA DE ALVAREZ	2020-02-06	ROBÓTICA	ROBÓTICA	2020	 
2	BETTY APAZA NAWA	2020-02-06	ROBÓTICA	ROBÓTICA	2020	 
3	JORGE GONZALES ALVARO	2020-02-06	CALCULO I	CALCULO I	2020	 
44	RAQUEL APAZA CUPANA	2020-02-06	CALCULO I	CALCULO I	2020	 
55	BETTY APAZA NAWA	2020-06-12	REPOSTERIA	REPOSTERIA	2020	 
66	BERTHA LAYME RAMIREZ DE CONDORI	2020-09-12	LENGUAJE	LENGUAJE	2020	 

Figura 3.27 Módulo Inscripción

Fuente: (Elaboración Propia)

```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'inscripcion-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'template'=>"{items}{summary}{pager}",
    'filter'=>$model,
    'columns'=>array(
        'id_inscripcion',
        array(
            'name'=>'id_participante',
            'header'=>'Participante',
            'value'=>'Participante::model()->getNombreC($data->id_participante)',
            'filter'=>'CHtml::listData(Participante::model()->findAll(array('order'=>'nombre'),'id_participante','nombreCompleto'))',
        ),
        'fecha_inscripcion',
        'nivel_inscripcion',

        // 'id_punto',
        array(
            'name'=>'id_punto',
            'header'=>'Punto Asignado',
            'value'=>'$data->idPunto->punto',
            'filter'=>'CHtml::listData(Punto::model()->findAll(array('order'=>'punto'),'id_punto','punto'))',
        ),
        'id_gestion',
        array(
            'class'=>'CButtonColumn',
            'header'=>'Modificar Inscripción', //Título de la columna
            'template'=>'{view}{update}{delete}', //Habilitar los tres botones o uno o dos
            'viewButtonImageUrl'=>'Yii::app()->baseUrl.'/images/1.png', // imagen que aparece como icono en el
            button
    )
);

```

Figura 3.28 Código Módulo Inscripción

Fuente: (Elaboración Propia)



Sistema Academico NUAM v.1.1 Inicio Integrantes Capacitación Mensualidades Almacenar Administración Cerrar sesión (administrador)

Inicio » Inscripciones » Crear

## Nueva Inscripción

Los campos con \* son obligatorios.

Fecha Inscripción

Nivel Inscripción

Participante

Curso

Gestión

**Figura 3.29** Adiciona Inscripción

**Fuente:** (Elaboración Propia)

```

<?php
/* @var $this InscripcionController */
/* @var $model Inscripcion */

$this->breadcrumbs=array(
    'Inscripciones'=>array('index'),
    'Create',
);

$this->menu=array(
    array('label'=>'Listar Inscripcion', 'url'=>array('index')),
    array('label'=>'Administrador Inscripcion', 'url'=>array('admin')),
);
?>

<h1>Nueva Inscripcion</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.30** Código Adiciona Inscripción

**Fuente:** (Elaboración Propia)



Figura 3.31 Módulo Mensualidades

Fuente: (Elaboración Propia)

```

<?php echo CHtml::link('Advanced Search', '#', array('class' => 'search-button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search', array(
    'model' => $model,
)); ?>
</div><!-- search-form -->

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id' => 'mensualidad-grid',
    'dataProvider' => $model->search(),
    'htmlOptions' => array('class' => 'table table-striped table-bordered table-condensed'),
    'filter' => $model,
    'columns' => array(
        // 'id mensualidad',
        array(
            'name' => 'id mensualidad',
            'header' => 'Codigo Mensualidad',
        ),
        // 'id inscripcion',
        array(
            'name' => 'id inscripcion',
            'header' => 'Curso y Participante',
            'value' => 'Inscripcion::model()->getNombreCursos($data->id_inscripcion)',
            // 'value' => 'Participante::model()->getNombreC($data->id_participante)',
            'filter' => CHtml::listData(mensualidad::model()->findAll(), 'id_inscripcion', 'nombresCursos'),
        ),
        'costo_curso',
        'costo_matricula',
        array(
            'class' => 'CButtonColumn',

```

Figura 3.32 Código Módulo Mensualidades

Fuente: (Elaboración Propia)



**Figura 3.33** Adiciona Mensualidad

**Fuente:** (Elaboración Propia)

```

<?php
/* @var $this MensualidadController */
/* @var $model Mensualidad */

$this->breadcrumbs=array(
    'Mensualidades'=>array('index'),
    'Adicionar',
);

$this->menu=array(
    array('label'=>'Listar Mensualidad', 'url'=>array('index')),
    array('label'=>'Administrar Mensualidad', 'url'=>array('admin')),
);
?>

<h1>Adicionar Mensualidad</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.34** Código Adiciona Mensualidad

**Fuente:** (Elaboración Propia)

Sistema Academico NUAM v.1.1 Inicio Integranets Capacitacion Mensualidades Almacen Administracion Logout (admin)

Inicio Mensualidades Administrador

## Administrar Pagos

Puede introducir opcionalmente un operador de comparación (< <= > >= <> <= >=) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search  
Vendo 1-5 de 5 resultados

Id Mensualidad	Curso y Participante	Curso	Participante	Costo Curso	Costo Matrícula	Realizar Pago
1	ROBOTICA - BENIGNA YUJRA DE ALVAREZ	ROBOTICA	BENIGNA YUJRA DE ALVAREZ	250	100	
2	ROBOTICA - BETTY APAZA NAVIA	ROBOTICA	BETTY APAZA NAVIA	50	100	
3	CALCULO I - JORGE GONZALES ALVARO	CALCULO I	JORGE GONZALES ALVARO	150	50	
4	CALCULO I - RAQUEL APAZA CUPANA	CALCULO I	RAQUEL APAZA CUPANA	200	100	
5	LENGUAJE - BERTHA LAYME RAMIREZ DE CONDORI	LENGUAJE	BERTHA LAYME RAMIREZ DE CONDORI	100	100	

Figura 3.35 Módulo Administrar Pagos

Fuente: (Elaboración Propia)

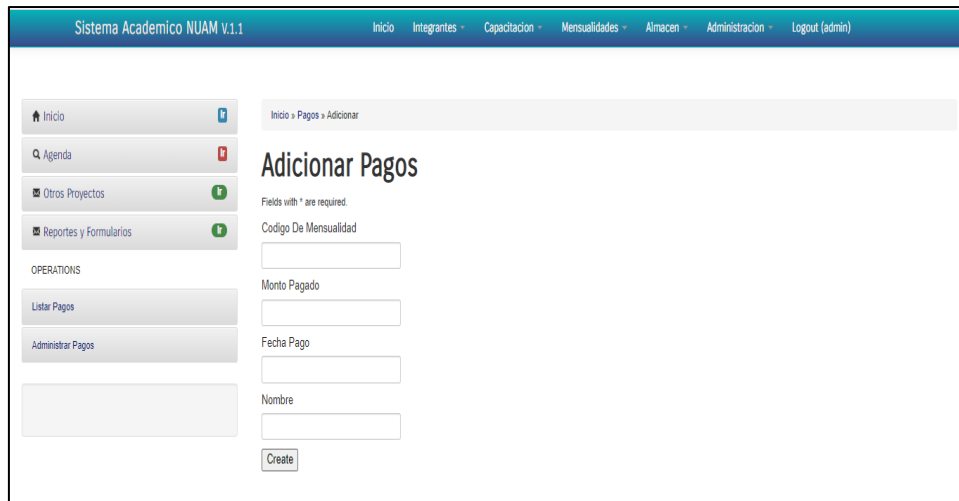
```

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'mensualidad-grid',
    'dataProvider'=>$model->search(),
    'htmlOptions'=>array('class'=>'table table-striped table-bordered table-condensed'),
    'filter'=>$model,
    'columns'=>array(
        'id_mensualidad',
        //id_inscripcion,
        array(
            'name'=>'id_inscripcion',
            'header'=>'Curso y Participante',
            'value'=>'Inscripcion::model()->getNombreCursos($data->id_inscripcion)',
            //value'=>'Participante::model()->getNombreC($data->id_participante)',
            'filter'=>CHtml::listData(mensualidad::model()->findAll(), 'id_inscripcion', 'nombresCursos'),
        ),
        array(
            //name'=>'id_inscripcion',
            'header'=>'Curso',
            'value'=>'Inscripcion::model()->getNombreCursos($data->id_inscripcion)',
            //value'=>'Participante::model()->getNombreC($data->id_participante)',
            //filter'=>CHtml::listData(inscripcion::model()->findAll(), 'id_inscripcion', 'nombreCurso'),
        ),
        array(
            //name'=>'id_inscripcion',
            'header'=>'Participante',
            'value'=>'Inscripcion::model()->getNombreParticipante($data->id_inscripcion)',
            //value'=>'$data->idPunto->punto',
            //filter'=>CHtml::listData(inscripcion::model()->findAll(), 'id_inscripcion', 'nombreCurso'),
        ),
    ),
);

```

Figura 3.36 Código Módulo Administrar Pagos

Fuente: (Elaboración Propia)



**Figura 3.37** Adiciona Administrar Pagos

**Fuente:** (Elaboración Propia)

```

<?php
/* @var $this PagosController */
/* @var $model Pagos */

$this->breadcrumbs=array(
    'Pagos'=>array('index'),
    'Adicionar',
);

$this->menu=array(
    array('label'=>'Listar Pagos', 'url'=>array('index')),
    array('label'=>'Administrar Pagos', 'url'=>array('admin')),
);
?>

<h1>Adicionar Pagos</h1>

<?php $this->renderPartial('_form', array('model'=>$model)); ?>

```

**Figura 3.38** Código Adiciona Administrar Pagos

**Fuente:** (Elaboración Propia)



**Figura 3.39** Modulo Historial de Reportes

**Fuente:** (Elaboración Propia)

```

    $f1='<center><h4>' . $f1 . '</h4></center>';

    $p1='Detalle de puntos..';
    $i=CHtml::link(CHtml::image('images/5.png', 'Lov'), array('reportes/puntos'));
    $j=CHtml::link(CHtml::image('images/4.png', 'Lov'), array('reportes/Rpuntos'));
    $k=CHtml::link(CHtml::image('images/6.png', 'Lov'), array('reportes/Epuntos'));
    $p1=$p1.'&nbsp;&nbsp;&nbsp;'.$k.'&nbsp;&nbsp;&nbsp;'.$i.'&nbsp;&nbsp;&nbsp;'.$j;

    $p1='<center><h4>' . $p1 . '</h4></center>';

    $this->widget('zii.widgets.jui.CJuiTabs', array(
        'tabs'=>array(
            // 'Reportes'=>'...',
            'Estudiante Detalle'=>$f1,
            'Docentes'=>$f,
            // 'Facilitadores'=>array('content'=>'hola', 'id'=>'tab3'),
            'Cursos'=>$p1,
        ),
        // additional javascript options for the tabs plugin
        'options'=>array(
            'collapsible'=>true,
        ),
    ),

```

**Figura 3.40** Código Módulo Historial de Reportes

**Fuente:** (Elaboración Propia)

### 3.6. Aplicación de la Métrica de Calidad ISO 9126

Para valorar la calidad de los productos de software que se desarrolla con la información proporcionada, medir la calidad de un software determina una de las tareas más complicadas que se presenta en el desarrollo de un sistema. Pero gracias a esta necesidad se fueron creando diferentes formas de medición de las mismas, que se adecuan a los datos referentes a la calidad para el resultado de los objetivos del proyecto.

Para el siguiente proyecto implementado se usa el modelo de calidad de la ISO 9126, donde se medirá los aspectos como la funcionalidad, fiabilidad, usabilidad y mantenibilidad.

#### 3.6.1. NORMA ISO 9126

La norma ISO 9126 (International Standard Organization – Organización Internacional de Normalización) nos ayudara a medir la calidad del sistema siguiendo los siguientes criterios planteados.

##### 3.6.1.1. *Funcionalidad.*

La funcionalidad es el grado en que el sistema satisface las necesidades que indican los siguientes sub atributos: estabilidad, exactitud, interoperabilidad, cumplimientos de seguridad.

El punto de función se toma en cuenta cinco características, el dominio de información, como son números de entrada, salida, condiciones, archivos e interfaz externa. Luego se realiza el cálculo de punto de función hallando la suma de estas características, parámetros de medición y el factor de ponderación también llamado punto medio de ponderación.

**Numero de entrada de usuarios:** Se cuenta cada entrada de usuario que proporciona datos al sistema.

**Numero de salidas de usuario:** Se refiere cada salida que proporciona el sistema al usuario. Entre estos pueden ser informes, reportes y pantallas de advertencia, notificaciones y errores.

**Número de peticiones al usuario:** Una petición está definida como una entrada

interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.

**Numero de archivos:** Se toma en cuenta cada archivo, estos pueden ser grupos lógicos de datos (tablas de base de datos).

**Numero de interfaces externas:** Se cuentan todas las interfaces legibles por la máquina.

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

$$PF = \text{Cuenta Total} \times (X + \text{Min } Y \sum Fi)$$

**Donde:**

PF: Medida de funcionalidad Cuenta Total = es el resultado del conteo de parámetros

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

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

Fi: Son los valores de ajustes de la complejidad, donde (i=1 a i=4)

A continuación, se muestra en la tabla el cálculo de los factores de ponderación donde se obtiene la cuenta total de parámetros de medición.



**Tabla 3.13**

*Cálculo de Parámetros de PF*

PARAMETROS DE MEDICIÓN	CUENTA	FATOR DE PESO			TOTAL
		Simple	Medio	Complejo	
Número de entradas de Usuario	20	3	4	6	80
Número de salida de usuario	15	4	5	7	75
Número de petición de usuario	10	3	4	6	40
Número de archivo	8	7	10	15	80
Número de Interfaces externos	0	5	7	10	0
<b>Cuenta Total</b>					<b>275</b>

**Fuente: (Elaboración Propia)**

Para la determinar los valores de ajuste de complejidad se indica según corresponda a las siguientes preguntas de la siguiente tabla:

**Tabla 3.14**

*Parámetros de Medición*

Importancia	0%	20%	40%	60%	80%	100%	Fi
<b>Escala</b>							
	No	Influencia	Moderado	Medio	Significativo	Esencial	
<b>Nº</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
<b>Factor/Preguntas</b>							

1	¿Requiere el sistema copia de seguridad y de recuperación fiable?		X	5
2	¿Requiere comunicación de datos?		X	5
3	¿Existen funciones de procesamientos distribuido?	X		3
4	¿El rendimiento es crítico?	X		3
5	¿Sera ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?		X	5
6	¿Requiere el sistema entrada de datos interactivo?		X	4
7	¿Requiere la entrada de datos interactivo que las transiciones de entradas se llevan a cabo sobre múltiples o variados operaciones?	X		3
8	¿Se actualizan los archivos maestros en forma interactiva?		X	5
9	¿Son complejos las entradas, las salidas, los archivos o peticiones?		X	4
10	¿Es complejo el procesamiento interno?		X	4
11	¿Se ha diseñado el código para ser reutilizables?	X		3

12	¿Están incluidos en el diseño la conversación y la instalación?	X	3
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	X	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?	X	5
<b>TOTAL</b>			<b>56</b>

**Fuente: (Elaboración Propia)**

Calculando el punto de función mediante la siguiente ecuación:

$$PF = Cuenta\ Total * (0,65 + 0,1 * \Sigma Fi)$$

**Donde:**

**Cuenta total:** es la suma del producto del factor de ponderación y valores de los parámetros.

$\Sigma Fi$ : es la sumatoria de los valores de ajuste de la complejidad.

**Calculando:**

$$PF = 275 * (0,65 + 0.01 * 56)$$

$$PF = 275 * (36,96)$$

$$PF = 319,64$$

Consideramos el máximo valor de complejidad  $\Sigma Fi=70$  calculamos al 100% el nivel de confianza de la siguiente manera:

$$PFmax = Cuenta\ Total * (0.65 + 0.01 * \Sigma Fi)$$

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

$$PFmax = 371.25$$

Con los valores ya obtenidos se encuentra la funcionalidad.

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

$$\text{Funcionalidad} = \frac{319,64}{371,25}$$

$$\text{Funcionalidad} = 0.860 * 100$$

$$\text{Funcionalidad} = 86 \%$$

Por tanto, la funcionalidad del sistema es representada en un 86 % lo cual indica que el sistema puede funcionar sin riesgos de fallar, y un 14 % aproximadamente de colapso del sistema, lo cual indica que cumple con los requisitos funcionales de manera satisfactoria.

### **3.6.1.2. Confiabilidad.**

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

Donde se encuentra:

$P(T \leq t) = F(t)$  Probabilidad de fallas (el termino en el cual sistema trabaja sin fallas)

$P(T \leq t) = 1 - F(t)$  Probabilidad de trabajo sin fallas (Tiempo en el cual no falla el sistema)

**Donde:**

$f$ : Funcionalidad del sistema.

$\mu$ : Es la probabilidad de error que puede tener el sistema.

$t$ : Tiempo de duración de gestión en el sistema.

Para lo que consideramos un periodo de 10 días como tiempo de prueba donde se define que cada 10 ejecuciones se presenta una falla de aproximación.

Calculando:

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

$$\text{Confiabilidad} = 0,860 * e^{(-0.01*10)}$$

$$\text{Confiabilidad} = 0.778 * 100$$

$$\text{Confiabilidad} = 77.8 \%$$

Reemplazando en las fórmulas de probabilidades:

$$P(T \leq t) = F(t) \quad \rightarrow P(T \leq t) = 0.778 = 77.8 \%$$

$$P(T \leq t) = 1 - F(t) \quad \rightarrow P(T \leq t) = 1 - 0.778$$

$$P(T \leq t) = 0, 222 = 22,2 \%$$

Por lo tanto, la confiabilidad del sistema en éxito es de 77.8 % en un tiempo de 10 días como tiempo de prueba y un 22.2 % de probabilidad de fallo.

### 3.6.1.3. Usabilidad

Para conocer el sistema satisface los requerimientos del usuario, se espera que el sistema sea de fácil entendimiento. En la norma ISO 9126, no se ve afectada por la funcionalidad y eficiencia ya que la usabilidad está determinada por los usuarios.

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

$$FU = [(\sum Xi/n) * 100]$$

**Donde:**

*Xi*: Es la sumatoria de valores

*n*: Es el número de preguntas

Para medir la ponderación de la usabilidad se tiene la siguiente tabla.

**Tabla 3.15**

*Valoración de las Preguntas*

ESCALA	VALOR
Muy Bueno	5
Bueno	4
Regular	3
Malo	2

---

**Pésimo**                      **1**

---

**Fuente: (Elaboración Propia)**

**Tabla 3.16**

*Parámetros de Medición de la Usabilidad*

<b>NRO</b>	<b>PREGUNTAS</b>	<b>Ponderación</b>
<b>1</b>	¿Puede utilizar con facilidad del sistema?	<b>5</b>
<b>2</b>	¿Puede controlar operaciones que el sistema solicite?	<b>5</b>
<b>3</b>	¿El sistema permite la retroalimentación de información?	<b>4</b>
<b>4</b>	¿El sistema cuenta con interfaz agradable a la vista?	<b>5</b>
<b>5</b>	¿La respuesta del sistema es satisfactoria?	<b>5</b>
<b>6</b>	¿Le parece complicadas las funciones del sistema?	<b>3</b>
<b>7</b>	¿Los resultados que proporciona el sistema facilitan el trabajo?	<b>4</b>
<b>8</b>	¿Durante el uso del sistema se produjo errores?	<b>3</b>
<b>X</b>		<b>34</b>

**Fuente: (Elaboración Propia)**

La usabilidad se calcula mediante la siguiente fórmula.

$$\text{Usabilidad} = [\text{Valor } n \times 100] \div 5$$

$$\text{Usabilidad} = [34 / 8 \times 100] \div 5$$

$$\text{Usabilidad} = 85\%$$

De esta manera, la usabilidad del sistema es de un 85% lo que indica que 9 de cada 10 usuarios pueden usar el sistema con facilidad.

#### **3.6.1.4. Mantenibilidad**

La mantenibilidad del sistema debe ser capaz de ser modificado a nivel funcional, con las posibilidades de hacer mejoras y nuevos requerimientos según los usuarios.

Por lo tanto, se determina la siguiente ecuación para el índice de madurez del software.

$$\text{MANTENIBILIDAD} = [Mt - (Fa + Fi + Fd)] / Mt$$

La mantenibilidad se determina mediante:

Mt = Número de módulos en la versión actual

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

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

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

Los valores para el sistema son los siguientes:

**Mt = 5; Fi = 1; Fa = 0; Fd = 0**

Por tanto:

$$\text{Mantenibilidad} = [5 - (0+1+0)] / 5$$

$$\text{Mantenibilidad} = 0,8 * 100$$

$$\text{Mantenibilidad} = 80\%$$

Por tanto, la mantenibilidad del sistema tiene un índice del 80% lo que indica que el sistema puede adaptarse a nuevos cambios y modificaciones

#### **3.6.1.5. Portabilidad**

La portabilidad del sistema es ejecutable desde cualquier plataforma debido a su diseño adaptable para ser trasladado de un sistema a otro.

Para calcular la portabilidad usamos la siguiente fórmula.

**Portabilidad = 1 - (Número de días para implementar el sistema / Número de días para portar el sistema)**

$$\text{Portabilidad} = 1 - (1 / 5)$$

$$\text{Portabilidad} = 0,8 \times 100$$

$$\text{Portabilidad} = 80\%$$

Por tanto, la portabilidad es de un 80%, que significa que el sistema es fácil de transportar, es decir puede ser llevado de un lugar a otro sin mucho esfuerzo.

Para sacar los resultados obtenidos de los porcentajes se detalla es siguiente cuadro:

**Tabla 3.17**

*Resultados de la Norma ISO 9126*

<b>CRITERIOS EVALUADOS</b>	<b>RESULTADOS %</b>
Funcionalidad	86
Confiabilidad	78
Usabilidad	85
Mantenibilidad	80
Portabilidad	80
<b>Evaluación de la Calidad Final</b>	<b>82</b>

**Fuente: (Elaboración Propia)**

Como conclusión de los resultados obtenidos de los porcentajes de cada uno de los atributos clave del software se evalúa, que el sistema es aceptable en un 82 % lo cual indica la factibilidad del sistema.



### 3.7. Estimación de Costos del Sistema

Para el método de estimación de costos del desarrollo de software, se establece una relación matemática entre lo que es el esfuerzo y el tiempo de desarrollo.

Para poder realizar el cálculo de V.A.N. se hará uso del modelo COCOMO II que es una herramienta que nos ayudará a estimar el costo del Sistema basado en el tamaño del mismo y utilizando otras características que se conocerá más adelante. Después de realizar los cálculos necesarios para la obtención de los resultados esperados estaremos en la capacidad de afirmar si el proyecto es viable, redituable y comprobar que es buena opción invertir en el proyecto.

#### 3.7.1. Cocomo II

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

#### 3.7.2. Método de Estimación

El Coste del sistema lo plantearemos en tres partes: Desarrollo del Software, Elaboración del Proyecto, Total del Software.

COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

- a) **Persona – Mes.**

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

- b) **Meses.**

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

- c) **Personas.**

$$P = E/T$$

**Dónde:**

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada submodelo.

KLDC: Cantidad de líneas de código distribuidas en miles.

**3.7.2.1. Costo de Software**

Para hallar el costo del software desarrollado bajo KLDC (Kilo-Líneas de código) se hará uso del modelo constructivo de costos orientado a los Puntos de Función.

**Por tanto:** se usa el cálculo del Factor de Ajuste

$$\text{Factor de Ajuste} = (0,65 + 0,01 \times \text{Factor de complejidad})$$

$$\text{Factor de Ajuste} = (0,65 + 0,01 \times 56)$$

$$\text{Factor de Ajuste} = 1,21$$

Para el cálculo de los puntos función (PF) se basa en la fórmula que se detalla a continuación:

$$PF = \text{Cuenta Total} \times \text{Factor de Ajuste}$$

$$PF = 275 \times 1,21$$

$$PF = 332.75$$

**Tabla 3.18**

*Conversión de Puntos de Función a KLDC*

Lenguaje	Nivel	Factor LDC/PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

**Fuente: (Pressman, 2002)**

Para hallar reemplazamos en la siguiente ecuación:

$$LDC = PF \times Factor \ LDC / PF$$

$$LDC = 332.75 \times 29$$

$$LDC = 9649.75$$

Para el cálculo del costo del sistema bajo KLDC se detalla a continuación:

$$KLDC = LD / 1000$$

$$KLDC = 9649.75 / 1000$$

$$KLDC = 9.64$$

**Tabla 3.19**

*Coeficiente de COCOMO II*

<b>PROYECTO DEL SOFTWARE</b>	<b>A</b>	<b>b</b>	<b>C</b>	<b>D</b>
Orgánico	2.40	1.05	2.50	0.38
Semi-Acoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.33

**Fuente: (Pressman, 2010)**

Ecuaciones para calcular el costo del Software:

**Tabla 3.20**

*Ecuaciones del Modelo COCOMO II*

VARIABLE	ECUACION	TIPO/UNIDAD
Esfuerzo requerido por el Proyecto	$E = a * (KLDC)^b * FAE$	Personas/Mes
Tiempo requerido por el Proyecto	$T = c * (E)^d$	Meses
Número de personas requeridos para el proyecto	$NP = \frac{E}{T}$	Personas
Costo Total	$CT = SueldoMes * NP * T$	Bs.

**Fuente** (Prentice- Haa, 1981)

Para hallar los valores del factor de ajuste del esfuerzo se utiliza la siguiente tabla:

Tabla 3.21

*Cálculo de los atributos de la FAE*

<b>ATRIBUTOS QUE AFECTAN AL COSTE</b>	<b>muy Bajo</b>	<b>Bajo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muy alto</b>	<b>Extra alto</b>
<b>Atributos del Software</b>						
Fiabilidad del software	0.75	0.88	1.00	1.15	1.40	
Tamaño base de datos		0.94	1.00	1.08	1.16	
Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
<b>Atributos del Hardware</b>						
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de memoria			1.00	1.06	1.21	1.56
Volatilidad de 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	
Capacidad de programadores	1.42	1.17	1.00	0.86	0.70	
Experiencia de S.O. usado	1.21	1.10	1.00	0.90		
Experiencia en el lenguaje de programación	1.14	1.07	1.00	0.95		
<b>Atributos del Proyecto</b>						
Uso de técnicas actuales de programación	1.24	1.10	1.00	0.91	0.82	
Uso de herramientas de software	1.24	1.10	1.00	0.91	0.83	
Restricciones de tiempo de desarrollo	1.23	1.08	1.00	1.04	1.10	

Fuente: (Elaboración Propia)

Hallamos la FAE.

$$FAE = (1.15 * 1.08 * 1 * 1 * 1 * 1 * 1.07 * 0.86 * 1 * 1 * 0.90 * 1 * 0.91 * 0.91 * 1)$$

$$FAE = 0.85$$

Para el presente proyecto de software tomaremos en cuenta como tamaño de complejidad en intermedio que es el orgánico, será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, con el cálculo de los atributos de la FAE.

Cálculo de esfuerzo:

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

$$E = 2.4 (9.64)^{1,05} * 0.85$$

$$E = 22.02 \text{ Personas/Mes}$$

Cálculo del tiempo:

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

$$T = 2,5 * (22.02)^{0,35}$$

$$T = 7.37 \text{ Equivalente a 7 Meses}$$

Cálculo para obtener el número de personas promedio:

$$NP = E/T$$

$$NP = 22.02 / 7.37$$

$$NP = 2.98 \text{ Equivale a 3 personas}$$


Calculando el costo total:

$$CT = \text{Sueldo mes} * NP * T$$

$$CT = 2.122 * 3 * 7$$

$$CT = 44562 \text{ Bs.}$$

Por lo tanto, se requiere un estimado 3 personas un trabajo de 7 meses para el desarrollo de sistema con un costo total de 44562 Bs.



# **CAPITULO IV**

# **PRUEBAS Y**

# **RESULTADOS**

## 4. PRUEBAS Y RESULTADOS

### 4.1. Introducción

En este capítulo, se realiza una breve descripción de las pruebas y resultados a la consultoría y capacitación “NUAM” que validan a este proyecto como el resultado al problema planteado en el capítulo 1, de manera que las pruebas y resultados muestran que el sistema fue factible y oportuno con su implementación, viendo una eficacia con resultados óptimos y rápidos a comparación del control de registros y reportes en gestiones pasadas.

### 4.2. Pruebas de Factibilidad del Sistema

### 4.3. Pruebas de Caja Negra.

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. (Pressman, 2010) Pág. 423



*Figura 4.1 Prueba de Caja Negra*

Fuente: (Elaboración Propia)



La prueba de la caja negra intenta descubrir errores en las siguientes categorías:

Funciones incorrecta o ausente.

Errores de interfaz.

Errores en la estructura de datos o acceso a base de datos externas.

Errores de rendimiento.

Errores en la inicialización y terminación.

#### **4.4. Niveles de Prueba**

Para cada nivel de prueba se asegura que los resultados cumplen con la verificación y validación del software.

##### **4.4.1. Pruebas Unitarias**

Para las pruebas unitarias se realizó el correcto funcionamiento de cada módulo que están desarrolladas al momento de verificarlas y ponerlo en producción.

##### **4.4.2. Pruebas de Integración**

En la prueba de integración se ocupan de probar las interfaces y las interacciones en los módulos, mostrando el correcto funcionamiento al realizar pruebas de todo el sistema.

##### **4.4.3. Pruebas de Software**

En las pruebas del sistema se comprueba que los componentes y la integración cumple con los requisitos especificados. Una vez ya integrada los módulos procedemos a la prueba del software con datos reales. (Francisco J. Garcia, 2020)

##### **4.4.4. Pruebas de Aceptación**

Las que se realizaron se encuentra bajo responsabilidad del programador, las pruebas de aceptación son responsabilidad del cliente o Usuario. Usualmente estas pruebas se realizan cuando el sistema está en marcha con datos reales y así poder satisfacer las expectativas del usuario final.

#### **4.5. Resultados**

Realizar la ejecución de las pruebas planteadas, ayudo y evidencio que los resultados de las pruebas que se realizaron al momento de insertar los datos y verificar los reportes, cumplen y satisfacen los requerimientos del usuario satisfactoriamente , por tanto, se evaluó que, a diferencia de la forma de trabajo antiguo de la institución al realizar registros o reportes manuales era lo que causaba demora y extravíos, de tal manera que, el sistema evita la demora y extravíos de reportes además de ser más rápido y eficiente.



**CAPITULO V**  
**CONCLUSIONES**  
**Y**  
**RECOMENDACIONES**

## **5. CONCLUSIONES Y RECOMENDACIONES**

En la culminación del presente proyecto de grado, se llegó a las siguientes conclusiones que se detalla a continuación.

### **5.1. Conclusiones.**

Habiendo realizado un estudio de los objetivos planteados se realizó un SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO en el cual el presente proyecto logró centralizar los procesos de los registros y reportes de información, de esta manera realizar un control de estudiantes, facilitadores, cursos, inscripciones y pago de mensualidades, que coadyuva en una mejor administración y control de información de la institución.

- ✓ Se logro centralizar toda la información mediante una base de datos conforme a las necesidades de la institución.
- ✓ Mediante la metodología de desarrollo UWE se pudo cumplir con el desarrollo del sistema de tal manera cumplir con los requerimientos del usuario final.
- ✓ Se logró mejorar el proceso de inscripciones de manera que ahora el sistema reduce el tiempo de inscripción de un alumno y se tiene un mejor control de los pagos para reducir.
- ✓ Se logro el control de pagos de mensualidades y matrícula de forma precisa y oportuna.
- ✓ Se logró generar reportes de listado de estudiantes inscritos, docentes y cursos realizados en la institución.
- ✓ Se logro mostrar el detalle de pagos de mensualidad de los estudiantes inscritos de tal manera que se tiene un mejor control.

Cumpliendo con los objetivos específicos se llega a la conclusión que el desarrollo e implementación del sistema cumple con el aporte tecnológico, por lo que se redujo el tiempo en la búsqueda y control de registros pertenecientes a la institución de CONSULTORIA Y CAPACITACION “NUAM” (NUEVO AMANECER).

## 5.2. Recomendaciones

Durante la evaluación y transcurso del desarrollo del software, se vio la necesidad de realizar reportes mensuales y anuales del pago de las mensualidades y control de inventario de materiales de apoyo, lo cual no está planteado en los objetivos porque la institución no solicitó, por consiguiente, se plantea las siguientes recomendaciones que son de importancia para la institución “NUAM” (NUEVO AMANECER).

- ✓ Se recomienda añadir un nuevo módulo de reportes, mensuales y anuales del costo total de las inscripciones y mensualidades.
- ✓ Se recomienda añadir un nuevo módulo de control de inventarios para hacer un mejor control del uso de sus materiales de apoyo.
- ✓ Para posteriores versiones del sistema se recomienda que se desarrollen sistemas de información integrados en diferentes áreas así de esta manera centralizar la información referente a cada área y también el presente proyecto sirva como una base para futuras actualizaciones.

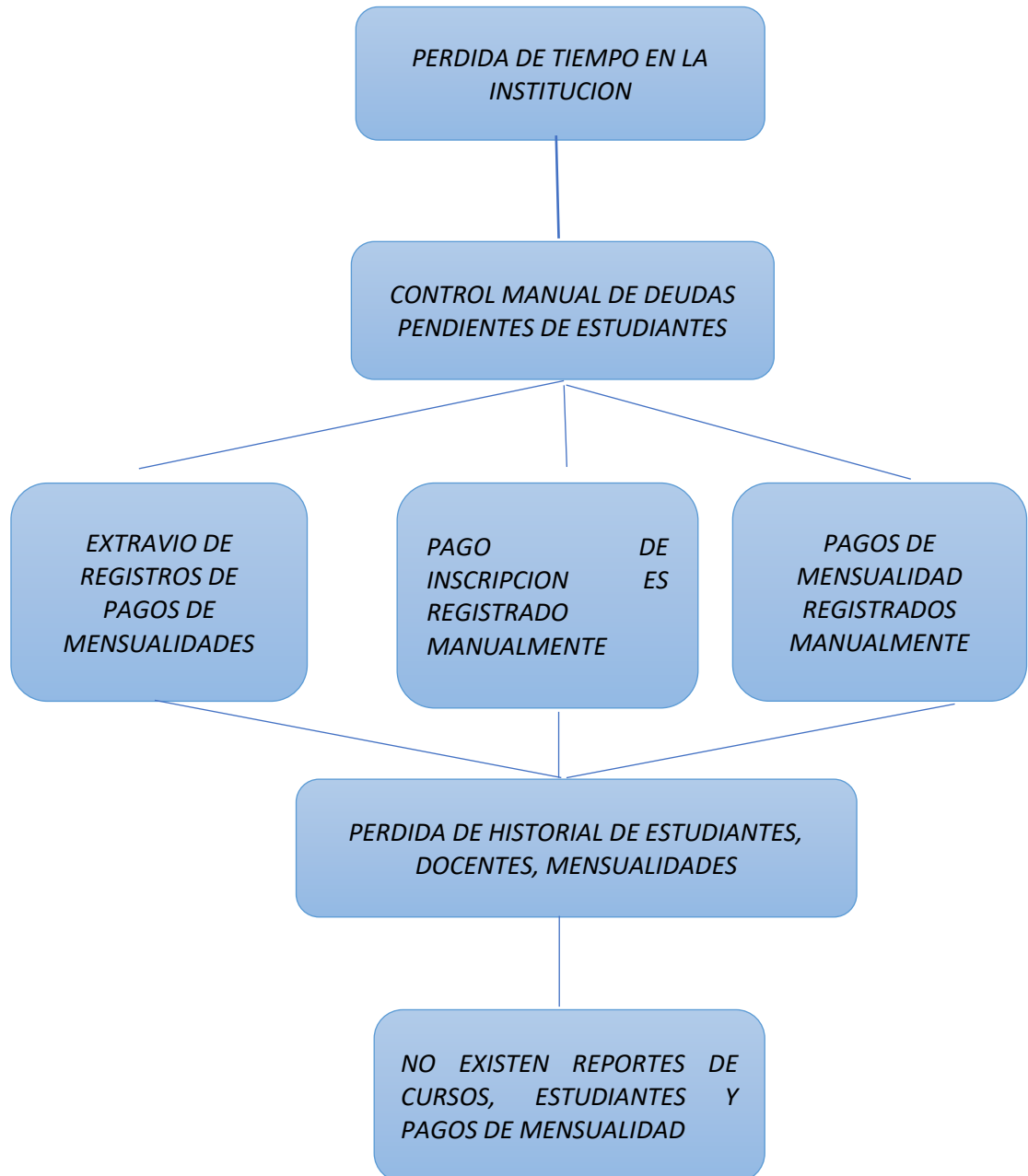
## BIBLIOGRAFÍA

- Alvarez, M. A. (s.f.). *Manual de jQuery*. Obtenido de Desarrolloweb.com:  
<http://desarrolloweb.com/manuales/manual-jquery.html>
- Aristegui, J. L. (2010). Los Casos d Prueba en la Prueba de Software. *Grupo Ingesoft, Chile*.
- beck, k. (2002). *metodologias XP*. mexico.
- Bertalanffy, L. v. (1989). *Teoria general de los Sistemas Fundamentos, desarrollo, aplicaciones*. Mexico: George Braziler.
- CastilloSánchez, A. (2011). *Diseño y Desarrollo de base de Datos en MySQL y Aplicación Web en PHP con servidor central APACHE*. Universidad Politecnica de Valencia.
- Duarte, G. (s.f.). *Guia Basica de Yii Framework*.
- Eshill. (2020). *Yiiframework*. Obtenido de <https://www.yiiframework.com/>
- Francisco J. Garcia, A. G. (26 de Febrero de 2020). *Ingenieria de Software I*. Obtenido de Departamento de Informática Universidad de Salamanca.
- Gauchat, J. D. (2012). *El Gran Libro de HTML5, CSS3 y Javascript*. Barcelona: Marcombo S.A.
- Harold Koontz, H. W. (2012). *Administración una Perspectiva Global y Empresarial*. Mexico: Karen Estrada.
- Introduccion JavaScript. (2020). *Escuela JavaScript*.
- Kabir, M. J. (2002). *La Biblia de Servidor Apache "*. Madrid: Anaya Multimedia.
- Kendall, K. y. (2011). *Analisis y Diseño de Sistemas*. México: Pearson Educación.
- Leandro Alegsa. (17 de Febrero de 2018). Obtenido de <http://www.alegsa.com.ar>

- LMU. (18 de junio de 2016). *UWE- Ingeniería web basada en UML*. Obtenido de <https://uwe.pst.ifi.lmu.de/>
- Lozano, J. A. (2016). *Maestría en Ciencias de la Información y Comunicaciones*. Bogotá. *Modelo Vista Controlado*. (s.f.).
- Moreno, A. (s.f.). Estimación de Proyectos Software. 85 - 138.
- Nilsen Espitia, O. A. (2016). *Modelo Vista Controlador*.
- Pantoja, E. B. (s.f.). El Patron de Diseño Modelo Vista Controlador (MVC). *Desarrollador de Software Jalasoft*.
- Peño, J. M. (2015). *Pruebas de Software, Fundamentos y Técnicas*.
- Pérez, J. E. (2008). *Introducción a JavaScript*. Obtenido de <http://www.librosweb.es/javascript>
- Pressman, R. S. (2010). *Ingeniería de Software*.
- Rodríguez, F. T. (2014). *Introducción a las pruebas de Sistemas de Información*.
- Sommerville, I. (2005). *Ingeniería del Software*.
- Sum, C. E. (2012). Automatización de la Codificación del Patrón Modelo Vista Controlador en Proyectos Orientados a la Web. *Red de Revistas Científicas de America Latina*, 239-250.
- Talón, E. M. (s.f.). Apache. *Curso mentor Apache*, 19.
- Terry, G. R. (1984). *Principios de Administración*. Mexico: Cia. Editorial Continental, S. A.

## ANEXO A

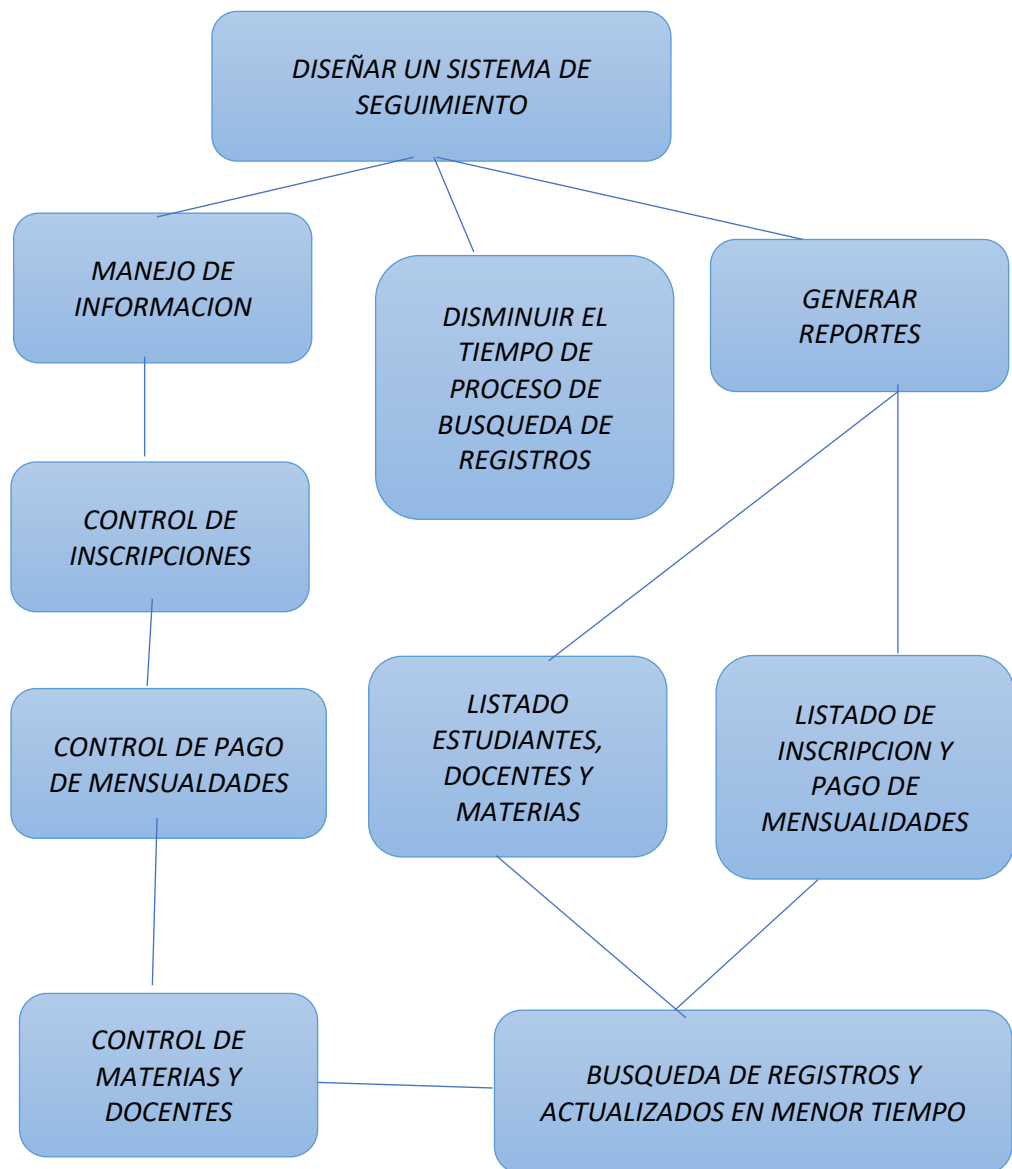
# ÁRBOL DE PROBLEMAS





## ANEXO B

### ÁRBOL DE OBJETIVOS



## **ANEXO C**

---

### **AVALES DE CONFORMIDAD**



centro de capacitacion  
**NUAM**

A : Ing. David Carlos Mamani Quispe  
DIRECTOR DE LA CARRERA INGENIERIA DE SISTEMAS  
UNIVERSIDAD PÚBLICA DE EL ALTO – UPEA

VIA: Ing Enrique Flores Baltazar  
DOCENTE DE TALLER DE LICENCIATURA II

DE: Marisol Sonco Poma  
COORDINADORA ADMINISTRATIVA CONSULTORIA Y CAPACITACION  
"NUAM" (NUEVO AMANECER)

El Alto, julio de 2020

Presente. -

Ref.: AVAL DE CONFORMIDAD

De mi mayor Consideración:

Por intermedio de la misma, reciba usted un cordial saludo, el motivo es hacerle conocer a su distinguida autoridad, se evidencia que la universitaria Raquel Apaza Cupana con cedula de Identidad 7045129 LP. Realizo y cumplió con los requerimientos de la institución para su proyecto de grado titulado "SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORIA Y CAPACITACION" CASO: "NUAM" (NUEVO AMANECER). El proyecto desarrollado será de gran ayuda y aporte para nuestra institución con lo cual damos paso libre para su defensa pública.

Sin otra particularidad, me despido con las consideraciones más atentas.

Atentamente,

  
Marisol Sonco Poma  
C.I. 9245612 LP.



OF.CENTRAL: 7/VILLA ADELA

PLAN 36 CALLE H N° 1

TELF: 22836483 – CEL: 78872391

---

El Alto, 07 de julio de 2020

Señor:

**Ing. Enrique Flores Baltazar**

**TUTOR METODOLOGICO TALLER II**

Presente. -

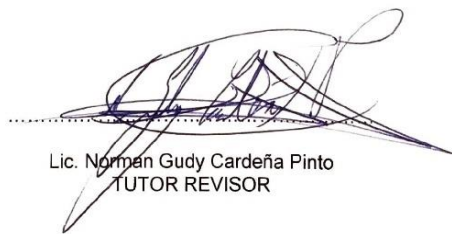
**Ref.: AVAL DE CONFORMIDAD**

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del Proyecto de Grado titulado **"SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORIA Y CAPACITACION"** caso: **"NUAM" (NUEVO AMANECER)**. Que propone la postulante universitaria: **RAQUEL APAZA CUPANA** con CI: **7045129 LP**. Para su defensa publica y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo con el reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



Lic. Norman Gudy Cardeña Pinto  
TUTOR REVISOR

## AVAL DE CONFORMIDAD

La Paz - El Alto, 07 de julio de 2020

Señor:

**Ing. Enrique Flores Baltazar**

**TUTOR METODOLÓGICO TALLER II**

Presente. -

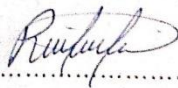
**Ref.: AVAL DE CONFORMIDAD**

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del Proyecto de Grado titulado **"SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORIA Y CAPACITACION"** caso: **"NUAM" (NUEVO AMANECER)**. Que propone la postulante universitaria: **RAQUEL APAZA CUPANA** con CI: **7045129 LP**. Para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



Ing. Ramiro Kantuta Limachi  
TUTOR ESPECIALISTA

**AVAL DE CONFORMIDAD**

La Paz - El Alto, julio de 2020

Señor:

Ing. David Carlos Mamani Quispe

**DIRECTOR DE CARRERA DE INGENIERÍA DE SISTEMAS**

Presente. -

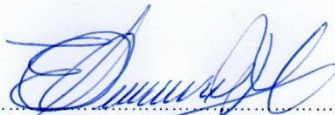
Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del Proyecto de Grado titulado **"SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORIA Y CAPACITACION" caso: "NUAM" (NUEVO AMANECER)**. Que propone la postulante universitaria: **RAQUEL APAZA CUPANA** con **CI: 7045129 LP**. Para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



M. Sc. Ing. Enrique Flores Baltazar  
TUTOR METODOLÓGICO

**ANEXO D**

---

**MANUAL DE USUARIO**

# MANUAL DE USUARIO

## SISTEMA DE CONTROL ADMINISTRATIVO Y SEGUIMIENTO PARA CONSULTORÍA Y CAPACITACIÓN





## 1. INTRODUCCIÓN

Actualmente las instituciones académicas necesitan de un sistema de control y seguimiento para el registro de estudiantes, pedagogos, materias, registro de inscripciones y control de pagos de mensualidades oportuno y de forma automática. Estos procesos se realizan de forma manual o algún registro en Word, por lo que representa un problema para encargado de Kardex al momento de búsqueda de información por las pérdidas de información.

Para lo cual, el sistema de control administrativo y seguimiento de consultoría y capacitación tiene como objetivo brindar información relevante de forma rápida y oportuna haciendo uso de herramientas tecnológicas que permitan tener una mejor organización dentro de la institución.

## 2. OBJETIVO

Guiar al usuario para poder acceder al sistema mediante este manual de forma correcta para su correcto funcionamiento.

## 3. REQUERIMIENTOS DEL SISTEMA

### 3.1 Requerimientos de Hardware

- ✓ Una computadora o laptop
- ✓ Conexión a internet

### 3.2 Requerimientos del Software

- ✓ Sistema Operativo Windows
- ✓ Navegadores (Mozilla Firefox, Google Chrome)

## 4. USUARIO

El sistema tiene un usuario que podrá realizar los cambios del sistema como modificar, eliminar, adicionar datos para poder mostrar los reportes deseados.

## 5. MANUAL DE USUARIO

### 5.1. Interfaz Inicio de Sesión al Sistema

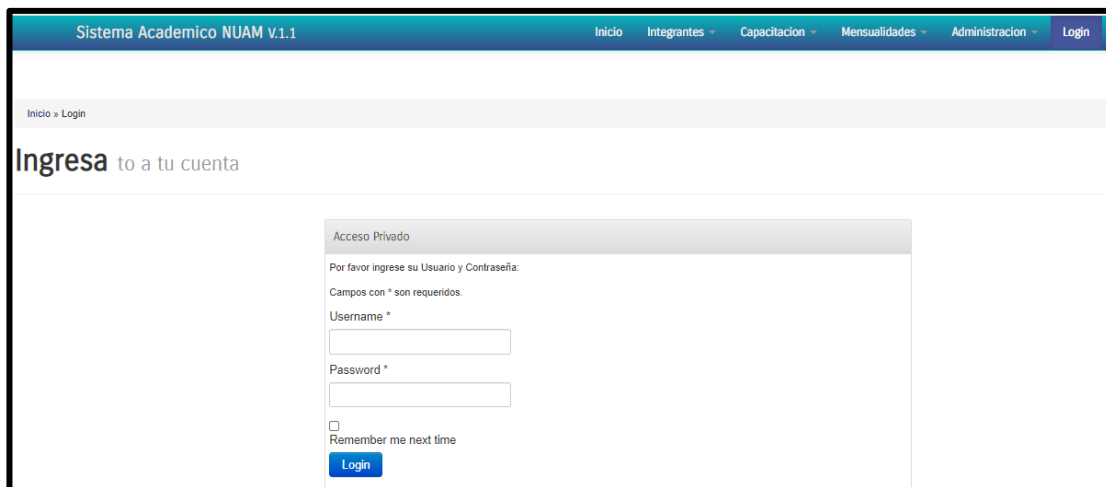
Se muestra en la pantalla de inicio la interfaz de inicio donde se muestra la pantalla principal para la navegacion.



El sistema verifica al usuario, para poder ingresar con su usuario y contraseña escriba de forma correcta para poder acceder al sistema, de lo contrario no podrá acceder a la información o se pondrá la pantalla en blanco sin poder acceder.

USUARIO: admin

CONTRASEÑA: admin



## 5.2. Funcionalidad General

El sistema se divide según los requerimientos desplegándose por las pestañas, iniciando con los integrantes, capacitacion, mensualidades, administracion. Los cuales al desplegar se muestran cada uno de los módulos con su respectiva informacion, listos para adicionar, modificar o eliminar datos.



## 5.3. Módulos que Integran al Sistema

### 5.3.1. Módulo Integrantes

Para observar la lista de los coordinadores, docentes, estudiantes existentes de la institución con los datos personales, se debe hacer clic en la pestaña integrantes seguido de la opción que desee ingresar, como podemos ver el despliegue de las siguientes figuras.

Sistema Academico NUAM v.1.1

Inicio Integrantes Capacitacion Mensualidades Administracion Logout (admin)

Inicio » Coordinador » Administrador

Coordinadores  
Docentes  
Estudiantes

## Administrador de Coordinadores

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Nombre	Apellido Paterno	Apellido Materno	Fecha Nacimiento	Ci	Modificar
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
SIN NOMBRE					
BENITA	RAMOS	GONZALES	0000-00-00		
ROGELIA	MAMANI	QUINO	0000-00-00		
SIXTO	FERNANDEZ	RAMIREZ	0000-00-00		
WALDO	TERRAZAS	QUISPE	0000-00-00		
OLGA	BENAVIDEZ	PEREZ	0000-00-00		
WILFREDO	QUISBERT	ZAMBRANA	1986-11-21	6543214	
JOSE	QUISPE	MAMANI	2010-06-09	5678345	

Sistema Academico NUAM v.1.1

Inicio Integrantes Capacitacion Mensualidades Administracion Logout (admin)

Inicio » Docente » Administrador

Coordinadores  
Docentes  
Estudiantes

## Administrador Docentes

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Nombre	Apellido Paterno	Apellido Materno	Ci	Id Gestion	Nombre Coordinador	Modificar
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
SANDOVAL	YANIQUE	QUISPE	4567489	2020	LUCIA QUIROZ MAMANI	
VLADIMIR SERGIO	VILLAFAN	SANTOS	7845678	2020	LUCIA QUIROZ MAMANI	
MARISOL ISAUARA	CHALCO	QUISPE	7845634	2020	OLGA BENAVIDEZ PEREZ	
FRANZ	VICTOR	LUGONES	8067835	2020	ANA FLORES QUISPE	
GABY	BARRIOS	LOPEZ	8067356	2020	BENITA RAMOS GONZALES	
ZULMA	CHOQUE	ZEGARRA	7567567	2020	WILFREDO QUISBERT ZAMBRANA	
JOSE MIGUEL	BARRIOS	LOPEZ	6756478	2020	ANA FLORES QUISPE	
GILBERTH HECTOR	LANZA	OPORTO	0	2016	SIN NOMBRE	

Sistema Academico NUAM v.1.1 Inicio Integranes Capacitacion Mensualidades Administracion Logout (admin)

Inicio » Participantes » Manage

Coordinadores  
Docentes  
Estudiantes

## Administrador de Participantes

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Nombre	Apellido Paterno	Apellido Materno	Ci	Edad	Sexo	Inscrito	Modificar
NATIVIDAD	CALLISAYA	DE COPA	2217322	58	F	Sin Inscripcion	
ELSA	CASI	TIPO DE SARZURI	2480166	48	F	Sin Inscripcion	
MARGARITA	PEREZ	JIMENEZ	2609916	50	F	Sin Inscripcion	
JUSTA	CANAVIRI	HUALLPA	2590186	48	F	Sin Inscripcion	
NELLY SANTUSA	TORREZ	DE ARUQUIPA	2187931	59	F	Sin Inscripcion	
TERESA	QUISPE	CALLIZAYA	4765149	50	F	Sin Inscripcion	
PAULINA	SARZURI	BERNA	2427606	53	F	Sin Inscripcion	
SALOME	TARQUI	FABIO	4247360	39	F	Sin Inscripcion	

Para poder adicionar a un nuevo Coordinador, docente, estudiante se debe ir a la opción:

***Nuevo Pedagogo***, y se despliega la ventana de adición.

Sistema Academico NUAM v.1.1 Inicio Integranes Capacitacion Mensualidades Administracion Logout (admin)

Inicio » Coordinador » Adicionar

## Insertar Nuevo Pedagogo

Campos con \* Son requeridos.

Nombre \*

Apellido Paterno \*

Apellido Materno

Fecha Nacimiento

Ci

Expedido

Esta acción se realiza para Coordinadores, docentes, estudiantes.

Para poder ver, modificar o eliminar, se va a la opción modificar ubicado en la parte derecha de la pantalla.

Sistema Academico NUAM v.1.1

Inicio » Participantes » Manage

## Administrador de Participantes

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Nombre	Apellido Paterno	Apellido Materno	Ci	Edad	Sexo	Inscrito	Modificar
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
NATIVIDAD	CALLISAYA	DE COPA	2217322	58	F	Sin Inscripcion	
ELSA	CASI	TIPO DE	2480166	48	F	Sin	

#### 5.4. Módulo capacitación

En la pestaña capacitacion se muestra a los cursos para poder detallar la fecha de inicio y fin del curso asi como el pedagogo con el curso que esta asignado.

Sistema Academico NUAM v.1.1

Inicio Integrantes - Capacitacion - Mensualidades - Administracion - Logout (admin)

Inicio » Curso » Administrador

## Administrador de Cursos

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Id Punto	Fecha Inicio	Fecha Fin	Facilitador	Curso	Id Gestion
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1	2020-06-01	2020-06-21	LOURDES CLEMENTINA ROMERO	MATEMATICA NIÑOS	2020
2	2020-07-06	2020-09-26	GILBERTH HECTOR LANZA OPORTO	LENGUAJE	2020
3	2020-06-15	2020-07-25	EMMA MASCO CHAMBI	MATEMATICA NIÑOS	2020
4	2020-07-06	2020-08-28	DAVID MAMANI LOZA	MATEMATICA NIÑOS	2020
5	2020-07-06	2020-08-21	NANCY CONDORI FLORES	REPOSTERIA	2020
6	2020-09-07	2020-11-27	GILBERTH HECTOR LANZA OPORTO	ROBOTICA	2020

En la pestaña inscripción se inserta al estudiante con la fecha que ingreso y su respectiva materia de asignación.

Sistema Academico NUAM v.1.1

Inicio Integrantes - Capacitación - Mensualidades - Administracion - Logout (admin)

Inicio » Inscripciones » Manage

### Administrar Inscripciones

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o \*) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Id Inscripción	Participante	Fecha Inscripción	Nivel Inscripción	Punto Asignado	Id Gestion	Modificar Inscripción
1	BENIGNA YUJRA DE ALVAREZ	2020-02-06	ROBOTICA	ROBOTICA	2020	
2	BETTY APAZA NAVIA	2020-02-06	ROBOTICA	ROBOTICA	2020	
3	JORGE GONZALES ALVARO	2020-02-06	CALCULO I	CALCULO I	2020	
4	RAQUEL APAZA CUPANA	2020-02-06	CALCULO I	CALCULO I	2020	
5	BETTY APAZA NAVIA	2020-06-12	REPOSTERIA	REPOSTERIA	2020	
6	BERTHA LAYME RAMIREZ DE CONDORI	2020-09-12	LENGUAJE	LENGUAJE	2020	

Para poder adicionar a un nuevo Curso o Inscripción se debe ir a la opción: **Nuevo curso, Nueva Inscripción** y se despliega la ventana de adición.

Sistema Academico NUAM v.1.1

Inicio Integrantes - Capacitación - Mensualidades - Administracion - Cerrar sesión (administrador)

Inicio » Inscripciones » Crear

### Nueva Inscripcion

Los campos con \* son obligatorios.

Fecha Inscripción

Nivel Inscripción

Participante

Curso

Gestion

Crear

## 5.5. Módulo Mensualidades

En la pestaña mensualidades se muestra como administrar las mensualidades para poder detallar el curso y estudiante así como el costo del curso y matrícula que está asignado.

Sistema Academico NUAM v.1.1 Inicio Integrantes Capacitacion Mensualidades Administracion Logout (admin)

Mensualidades Pagos de Mensualidad

Inicio Mensualidades Administrador

### Administrar Mensualidades

Puede introducir opcionalmente un operador de comparación (<, >, >=, <= o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search

Viendo 1-5 de 5 resultados.

Codigo Mensualidad	Curso y Participante	Costo Curso	Costo Matricula	
1	ROBOTICA - BENIGNA YUJRA DE ALVAREZ	250	100	
2	ROBOTICA - BETTY APAZA NAVIA	50	100	
3	CALCULO I - JORGE GONZALES ALVARO	150	50	
4	CALCULO I - RAQUEL APAZA CUPANA	200	100	
5	LENGUAJE - BERTHA LAYME RAMIREZ DE CONDORI	100	100	

En la pestaña de administrar pagos se inserta al estudiante con el curso que este asignado para detallar los pagos acumulativos que se realizan cuando el estudiante deposita su pago.



Sistema Academico NUAM v.1.1

Inicio Integrantes - Capacitacion - Mensualidades - Administracion - Logout (admin)

Mensualidades  
Pagos de Mensualidad

Inicio » Mensualidades » Administrador

### Administrar Pagos

Puede introducir opcionalmente un operador de comparación (<, <=, >, >=, <> o =) al comienzo de cada uno de los valores de búsqueda para especificar la forma en la comparación que debe hacerse.

Advanced Search  
Viendo 1-5 de 5 resultados.

Id Mensualidad	Curso y Participante	Curso	Participante	Costo Curso	Costo Matricula	Realizar Pago
1	ROBOTICA - BENIGNA YUJRA DE ALVAREZ	ROBOTICA	BENIGNA YUJRA DE ALVAREZ	250	100	
2	ROBOTICA - BETTY APAZA NAVIA	ROBOTICA	BETTY APAZA NAVIA	50	100	
3	CALCULO I - JORGE GONZALES ALVARO	CALCULO I	JORGE GONZALES ALVARO	150	50	
4	CALCULO I - RAQUEL APAZA CUPANA	CALCULO I	RAQUEL APAZA CUPANA	200	100	

## 5.6. Módulo Administración

En la pestaña administración se muestra como administrar los reportes para poder datallar los cursos, docentes y estudiante , en la cual nos lanza datos requeridos en una hoja de Pdf y Excel.

Sistema Academico NUAM v.1.1

Inicio Integrantes - Capacitacion - Mensualidades - Administracion - Logout (admin)

Reportes  
Agenda  
Usuarios

Inicio » Reportes » Index

### Listar Reportes

Estudiante Detalle Docentes Cursos

Detalle de puntos..

En la pestaña agenda, se puede realizar agenda para las nuevas asignaciones de pedagogos y materias futuras, q algunas instituciones requieren los servicios de enseñanza.

Sistema Academico NUAM v.1.1

Inicio Integrantes Capacitacion Mensualidades Administracion Logout (admin)

Reportes  
Agenda  
Usuarios

Inicio » Agendas » Index

## Listar Agendas

Ordenado Por Fecha

2020-08-12 , 13:00:00

Institucion: Iglesia Agua de la vida  
Nombre Encargado: marco Limachi  
Numero de Contacto: 78645454  
Direccion: Z/ Pedro Domingo Murillo C/ 4 N° 34  
Punto: 3  
Observacion:

Ver

Inicio

Agenda

Otros Proyectos

Reportes y Formularios

OPERATIONS

Nuevo Registro Agenda

Administrador Agenda

Tareas Finalizadas

Nuevo formulario

Nuevo formulario Participante