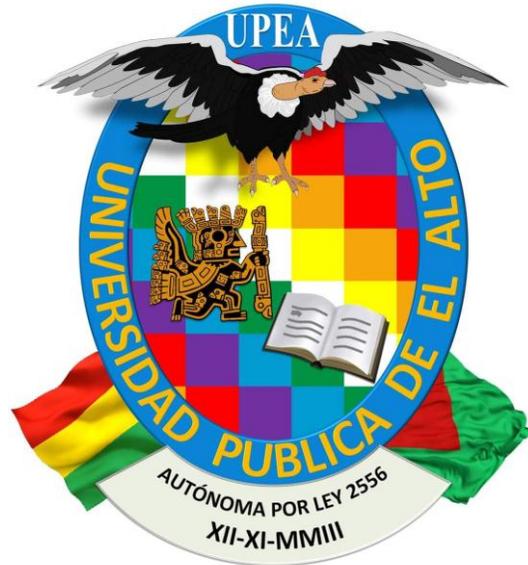


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ACADÉMICA
EN UN ENTORNO WEB PARA LA UNIDAD DE KARDEX
ACADEMICO”**

**CASO: CARRERA CONTADURIA PUBLICA,
UNIVERSIDAD PÚBLICA DE EL ALTO**

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: Informática y Comunicaciones

Postulante: Jhonny Vargas Ramos

Tutor Metodológico: ING. Marisol Arguedas Balladares

Tutor Especialista: ING. Paulino Tarqui Baltazar

Tutor Revisor: ING. Ramiro Kantuta Limachi

EL ALTO – BOLIVIA

2020

(TALLER DE GRADO II)

DEDICATORIA:

La concepción del presente proyecto lo dedico a Dios, a mi abuela y mi familia. A Dios por cuidarme y acompañarme día a día, mi querida y extrañada abuela Raymunda, por apoyarme incansablemente antes, durante y después de mis estudios en la universidad, a mi familia por lo paciencia, fortaleza y comprensión en tiempos difíciles.

AGRADECIMIENTOS

A Dios por brindarme un techo donde reposar, el alimento para recargar fuerzas, salud para alcanzar mis metas y sobre todo por las bendiciones que me toco en cada una de las etapas de mi vida.

Agradecer a mis padres Rosendo y Cleofe por darme su apoyo en los primeros años de mi vida. A mi familia gracias por todo el valor en los días de oscuridad, que hicieron una sumatoria para alcanzar una meta con la culminación de este proyecto. Al mi gran amigo Sergio Garcia por los años en la universidad pasando buenos y malos momentos y por su desinteresada paciencia para compartir sus conocimientos que coadyuvaron con la culminación del presente proyecto.

Al Ing. Paulino Tarqui, por brindarme su apoyo, colaboración, paciencia y desprendimiento aconsejarme en el desarrollo de este proyecto.

Al Ing. Ramiro Kantuta, por la colaboración, orientación y guía.

A la Ing. Marisol Arguedas por su incansable paciencia por guiarme no solo a mi sino también a mis compañeros de estudio, por sus constantes sugerencias, apoyo y por su paciencia más allá del compromiso.

A todo el personal docente de la carrera de Ingeniería de Sistemas que me guiaron y forjaron a lo largo de mi estudio, mi gratitud por haberme dado la herramienta base para culminar mis estudios en la universidad. A la comunidad universitaria docente-estudiantil de la Universidad Pública de El Alto por la valerosa lucha para la creación y consolidación de nuestra casa superior de estudios, que ofrendaron su vida por nuestra convicción.

Agradecer también a la Carrera de Contaduría Publica de la Universidad Pública de El Alto a su Director Lic. R. Ricardo Nogales Q. por la oportunidad de realizar y plasmar el presente proyecto.

RESUMEN

Hoy en día la información en el área académica es un recurso que adquiere mucha importancia en función del tiempo para toda Institución dedicada a la educación y formación, en consecuencia, administra grandes cantidades de información, además que existe el factor de la demanda de estudiantes por cursar o cursaran una carrera o un curso de formación en una determinada institución por lo que va creciendo continuamente. Las Instituciones de carácter educativo no son la excepción, razón por la cual, se debe prestar especial atención al manejo de datos que se genera en las mismas.

La Unidad de Kardex de la carrera Contaduría Pública - UPEA está experimentando un crecimiento significativo en la cantidad de estudiantes. El presente trabajo hace énfasis a los procesos del área académica como gestión del pensum, planificación académica, inscripción, registro de notas y seguimiento académico de estudiantes, son manipulados y procesados de forma semi manual y desorganizada.

Sin embargo este proyecto pretende solucionar los inconvenientes anteriores, desarrollando el Sistema de información para la gestión académica en un entorno Web Carrera Contaduría Pública. Al tratarse de una Universidad, los procesos del área académica deben presentar resultados con datos fidedignos, para asegurar el prestigio y el éxito de dicha entidad.

Para el desarrollo del sistema se toma a la metodología UWE (Ingeniería Web basada en UML), utilizando el lenguaje de programación PHP, un gestor de base de datos MariaDB, Servidor Web Apache, Framework CodeIgniter, usando modelo vista controlador, HTML5, (Gomez J. , 2011) (Pantaleo, 2012) (Suarez, 2012)CSS3.

INDICE

	Pág.
1. CAPÍTULO I: MARCO PRELIMINAR	1
1.1. Introducción	1
1.2. Antecedentes	2
1.2.1. Antecedentes Institucionales.....	2
1.2.2. Organigrama Institucional de la Carrera Contaduría Pública	4
1.2.3. Misión	4
1.2.4. Visión	5
1.2.5. Antecedentes Académicos	5
1.2.6. Antecedentes Nacionales	5
1.2.7. Antecedentes Internacionales	6
1.3. Planteamiento del Problema	7
1.3.1. Problema Principal.....	9
1.3.2. Problemas Secundarios	9
1.4. Objetivos	10
1.4.1. Objetivo General.....	10
1.4.2. Objetivos Específicos	10
1.5. Justificación	11
1.5.1. Justificación Técnica.....	11
1.5.2. Justificación Económica	11
1.5.3. Justificación Social	12
1.6. Metodología Web.....	12
1.6.1. Metodología de Ingeniería Web.....	12
1.6.2. Técnicas de Investigación	12
1.6.3. Métricas de Calidad	12
1.6.4. Métodos de Estimación de Costos	13
1.7. Herramientas	13
1.7.1. Lenguaje de programación.....	13
1.7.2. Sistema gestor de base de datos.....	13
1.7.3. Servidor Web Apache.....	13
1.7.4. Otras herramientas	14
1.8. Límites y Alcances.....	14
1.8.1. Límites	14
1.8.2. Alcances.....	14
1.9. Aportes.....	15
CAPITULO II: MARCO TEÓRICO	16
2. Marco Teórico	16
2.1. Sistema.....	16
2.2. Información.....	16
2.3. Sistema de Información	17
2.4. Gestión	19
2.6. Metodología Web.....	20
2.6.1. Metodología Basada en UML (UWE)	20
2.6.2. Actividades de modelado UWE.....	21
2.6.2.1. Modelo de casos de uso.....	21

2.6.2.2.	Modelo de contenido	22
2.6.2.3.	Modelo de navegación.....	23
2.6.2.4.	Modelo de presentación.....	24
2.6.3.	Fase de desarrollo UWE	24
2.7.	Pruebas de Software.....	26
2.8.	Métricas de Calidad de Software	28
2.8.1.	Métricas de Calidad ISO/IEC 9126	28
2.8.1.1.	Funcionalidad	29
2.8.1.2.	Confiabilidad	30
2.8.1.3.	Usabilidad.....	30
2.8.1.4.	Eficiencia.....	31
2.8.1.5.	Mantenibilidad.....	31
2.8.1.6.	Portabilidad.....	31
2.9.	Tecnologías para el Desarrollo de Aplicaciones Web.	32
2.9.1.	Lenguaje de programación PHP	32
2.9.2.	CodeIgniter3	33
2.9.2.1.	Modelo Vista Controlador	34
2.10.	Herramientas para la Implementación de Aplicaciones Web.....	36
2.10.1.	Servidor Http Apache	36
2.10.1.1.	Servidor Web	36
2.10.2.	MariaDB	37
2.11.	Modelo de Estimación de Costos.....	38
2.11.1.	Modelo COCOMO II.....	38
2.11.1.1.	Métricas orientadas a la función: Punto Función (PF)	39
2.11.1.2.	Factores.....	41
CAPITULO III: MARCO APLICATIVO.....		45
3.	Marco Aplicativo.....	45
3.1.	Modelo del Negocio.....	45
3.1.1.	Análisis actual de la Carrera Contaduría Pública-Upea.....	45
3.1.2.	Obtención de requerimientos	45
3.1.3.	Procesos académicos generales	46
3.2.	Modelo del Sistema	47
3.2.1.2.	Requerimientos NO Funcionales.....	48
3.2.2.	Definición de actores (del sistema).....	48
3.3.	Aplicación del Modelo UWE	50
3.3.1.	Modelo de Casos de Uso	50
3.3.1.1.	Análisis de requerimientos	50
3.3.1.1.1.	Diagrama de casos de uso comercial	50
3.3.1.1.2.	Modelo de Caso de Uso General del Sistema	51
3.3.2.	Modelo de contenido	57
3.3.3.	Modelo de Navegación	59
3.3.4.	Modelo de Presentación.....	63
3.3.5.	Modelo de Implementación	66
3.2.	Métricas de Calidad de Software	69
3.2.1.	Factores de Calidad ISO 9126	69
3.2.1.1.	Funcionalidad	70

3.2.1.2.	Confiabilidad.....	74
3.2.1.3.	Usabilidad.....	75
3.2.1.4.	Eficiencia.....	76
3.2.1.5.	Mantenibilidad.....	77
3.2.1.5.1.	Mantenimiento Correctivo.....	79
3.2.1.5.2.	Mantenimiento de Mejora.....	79
3.2.1.6.	Portabilidad.....	79
3.2.1.7.	Resultado Final.....	81
3.3.	Método de Estimación de Costos.....	81
3.3.1.1.	Modelo COCOMO II.....	81
3.3.1.1.1.	Estimación del punto función.....	81
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.....		88
4.	Conclusiones y Recomendaciones.....	88
4.1.	Conclusiones.....	88
4.2.	Recomendaciones.....	89
BIBLIOGRAFÍA.....		90
ANEXOS.....		92

INDICE DE FIGURAS

	PÁG.
Figura Nº 1.1: Organigrama Carrera Contaduría Publica	4
Figura Nº2 1: Actividades de un Sistema de Información	18
Figura Nº2 2: Ingeniería Web Basada en UML.....	21
Figura Nº2 3: Diagrama de caso de uso en la Metodología UWE	22
Figura Nº2 4: Diagrama de contenido en la Metodología UWE	22
Figura Nº2 5: Diagrama de navegación en la Metodología UWE	23
Figura Nº2 6: Estereotipos del Diagrama de navegación	23
Figura Nº2 7: Diagrama de presentación en la Metodología UWE	24
Figura Nº 3. 2: Modelo de Caso de Uso General del Sistema	51
Figura Nº 3. 3: Diagrama de Caso de Uso: “GESTIÓN DE PENSUM”	51
Figura Nº 3. 4: Diagrama de Caso de Uso: “PLANIFICACIÓN ACADÉMICA”	53
Figura Nº 3. 5: Diagrama de Caso de Uso: “INSCRIPCIÓN”	54
Figura Nº 3. 6: Diagrama de Caso de Uso: “REGISTRO DE NOTAS”	55
Figura Nº 3. 7: Diagrama de Caso de Uso: “SEGUIMIENTO ACADEMICO DE ESTUDIANTE”	56
Figura Nº 3. 9: Modelo Conceptual	58
Figura Nº 3. 10: Modelo de Navegación Gestión de Pensum.....	59
Figura Nº 3. 11: Modelo de Navegación Planificación Académica	60
Figura Nº 3. 12: Modelo de Navegación Inscripción	61
Figura Nº 3. 13: Modelo de Navegación Registro de Notas	62
Figura Nº 3. 14: Modelo de Navegación Seguimiento Académico de Estudiante	63
Figura Nº 3. 15: Modelo de Presentación Autenticación	64
Figura Nº 3. 16: Modelo de Presentación Gestión de Pensum.....	64
Figura Nº 3. 17: Modelo de Presentación Planificación Académica.....	65
Figura Nº 3. 18: Modelo de Presentación Inscripción	65
Figura Nº 3. 19: Modelo de Presentación Registro de Notas	66
Figura Nº 3. 20: Autenticación inicial con el sistema	67
Figura Nº 3. 21: Interfaz de Gestión de Pensum (Administrador)	67
Figura Nº 3. 22: Interfaz de Planificación Académica (Encargado de kardex)	68
Figura Nº 3. 23: Interfaz de Inscripción (Estudiante)	68
Figura Nº 3. 24: Interfaz de Registro de Notas (Encargado de Kardex)	69

INDICE DE TABLAS

Tabla Nº 1. 1: Cuadro Comparativo	93
Tabla Nº 3. 1: Tareas Realizadas Para Obtención De Requisitos	45
Tabla Nº 3. 2: Requerimientos Funcionales	47
Tabla Nº 3. 3: Requerimientos Funcionales	48
Tabla Nº 3. 4: Definición de Actores	49
Tabla Nº 3. 5: Caso de Uso “GESTIÓN DE PENSUM”	52
Tabla Nº 3. 6: Caso de Uso “PLANIFICACIÓN ACADÉMICA”	53
Tabla Nº 3. 7: Caso de Uso “INSCRIPCIÓN”	54

Tabla Nº 3. 8: Caso de Uso “REGISTRO DE NOTAS”	55
Tabla Nº 3. 9: Caso de Uso “SEGUIMIENTO ACADÉMICO DE ESTUDIANTE” ...	57
Tabla Nº 3. 11: Número de Entradas Externas	70
Tabla Nº 3. 12: Número de Salidas Externas.....	70
Tabla Nº 3. 13: Número de Consultas Externas.....	70
Tabla Nº 3. 14: Número de Archivos Lógicos Internos	71
Tabla Nº 3. 15: Número de Interfaces Externas.....	71
Tabla Nº 3. 16: Conteo Total.....	71
Tabla Nº 3. 17: Rangos para evaluar el PF.....	72
Tabla Nº 3. 18: Factor de Complejidad	72
Tabla Nº 3. 19: Rangos para Evaluar la Usabilidad	75
Tabla Nº 3. 20: Preguntas para la valoración de Usabilidad	75
Tabla Nº 3. 21: Rangos para Evaluar la Eficiencia.....	76
Tabla Nº 3. 22: Valoración para la Eficiencia	77
Tabla Nº 3. 23: Mantenimiento Correctivo	79
Tabla Nº 3. 24: Mantenimiento de Mejora.....	79
Tabla Nº 3. 25: Rangos para Evaluar la Portabilidad	80
Tabla Nº 3. 26: Valoración de Portabilidad.....	80
Tabla Nº 3. 27: Resultados de las Características de Calidad	81
Tabla Nº 3. 28: Valores del dominio de Información.....	82
Tabla Nº 3. 29: Valores de ajustes de la complejidad	82
Tabla Nº 3. 30: Conversión de los puntos de función a KDLC	84
Tabla Nº 3. 31: Ecuaciones del modelo intermedio que define COCOMO	84
Tabla Nº 3. 32: Aplicación del Modelo Intermedio.....	85
Tabla Nº 3. 33: Cálculo de Atributos FAE.....	85

CAPÍTULO I

MARCO PRELIMINAR

1. CAPÍTULO I: MARCO PRELIMINAR

1.1. Introducción

En las primeras décadas del siglo XXI las innovaciones tecnológicas con inventos o creaciones se presentan como una novedad en la forma de suplir las necesidades de la sociedad actual que sin el avance de la tecnología no existirían. Gracias al avance extraordinario de la ciencia en los últimos años, la tecnología creada e implementada con la ayuda de ingenieros, ha hecho que las innovaciones del siglo XXI sean tan sorprendentes que han superado la ficción. La sociedad en el trajín diario experimenta limitaciones de diferentes aspectos, en este sentido se delimitan dos grandes grupos a consecuencia de estas limitantes, el primer grupo: los que buscan un servicio para satisfacer una necesidad de cualquier nivel o rubro, y el segundo grupo: los que brindan o trabajan para cubrir o satisfacer las necesidades del primer grupo en este segundo grupo es donde se debe innovar o renovar de manera periódica los servicios para maximizar el alcance de los mismos y minimizar procesos y recursos.

La información en el área académica es un recurso que adquiere mucha importancia en función del tiempo para toda Institución dedicada a la educación y formación, en consecuencia, administra grandes cantidades de información, además que existe el factor de la demanda de estudiantes por cursar o cursaran una carrera o un curso de formación en una determinada institución por lo que va creciendo continuamente, en el caso de las universidades no son ajenas a esta situación tal es el caso de la Universidad Pública de El Alto, esta casa de estudios superiores de la Ciudad de El Alto, encargado de la formación de profesionales a nivel Licenciatura y Técnico Superior como también Técnico Medio en algunas de sus ofertas académicas, las metodologías aplicadas en los procesos del área académica como gestión de pensum, planificación académica, inscripción, registro de notas y seguimiento académico de estudiantes no son las adecuadas u óptimas, siendo que en algunos carreras aún se realizan varios de estos procesos forma manual y desorganizada.

Al tratarse de una institución de educación superior, los procesos del área académica deben presentar la información y resultados con datos confiables que muestran la realidad académica de una determinada carrera, para mostrar la seguridad, el prestigio y el éxito de dicha institución.

En el presente proyecto se pretende afrontar los inconvenientes mencionados párrafo anterior, desarrollando el **Sistema de Información para la Gestión Académica en un entorno Web para la Unidad de Kardex Académico en la carrera Contaduría Pública de la Universidad Pública de El Alto**, que contribuya en los procesos del área académica, para proporcionar información confiable, segura y oportuna, disminuyendo al mismo tiempo el trabajo excesivo de esta área. El sistema propuesto se desarrollará bajo la metodología UWE, utilizando el lenguaje de programación PHP, un gestor de base de datos MariaDB, Servidor Web Apache, Framework CodeIgniter 3, usando modelo vista controlador, HTML5, CSS3.

1.2. Antecedentes

1.2.1. Antecedentes Institucionales

La Universidad Pública de El Alto también conocida por su acrónimo UPEA es una universidad de Bolivia, con sede en la ciudad de El Alto del Departamento de La Paz. La institución pública ofrece 35 carreras en las áreas socio-político-económica, salud y tecnología.

El 5 de septiembre de 2000 tras movilizaciones sociales se promulgó la ley 2115 que determinó la creación de la Universidad Pública de El Alto, la misma determina que la UPEA tendría autonomía en 5 años, tiempo durante el cual estaría a cargo de un consejo formado por el Ministerio de Educación de Bolivia y otros organismos gubernamentales. De acuerdo a la ley de su creación, el ente de mayor decisión en la universidad sería el Consejo de Desarrollo Institucional (CDI), mismo en el que estaban insertos miembros de organizaciones sociales de la ciudad de El Alto que tenían poca relación con el que hacer académico.

En noviembre de 2003 durante el gobierno de Carlos Mesa se realiza una modificación a la ley 2115 que garantiza la autonomía universitaria de la UPEA. La universidad ha sido un actor principal de las protestas sociales durante los últimos años. La Universidad Pública de El Alto al igual que todas las Universidades del Sistema de Universidades Públicas en Bolivia se rige bajo el cogobierno docente y estudiantil. Las máximas autoridades docentes son el Rector, el Vicerrector y los Decanos por área académica y la instancia de cogobierno estudiantil la constituyen los Centros de Estudiantes.

En el caso específico de la carrera de Contaduría Pública es una de las carreras con las que nació la Universidad Pública de El Alto, en la actualidad cuenta con alrededor de 6.500 estudiantes en sus diferentes sedes académicas en el departamento de La Paz. Perteneciente al área de Ciencias Económica Administrativas y Financieras junto a carreras como Economía, Administración de Empresas, Comercio Internacional y Gestión Turística y Hotelera.

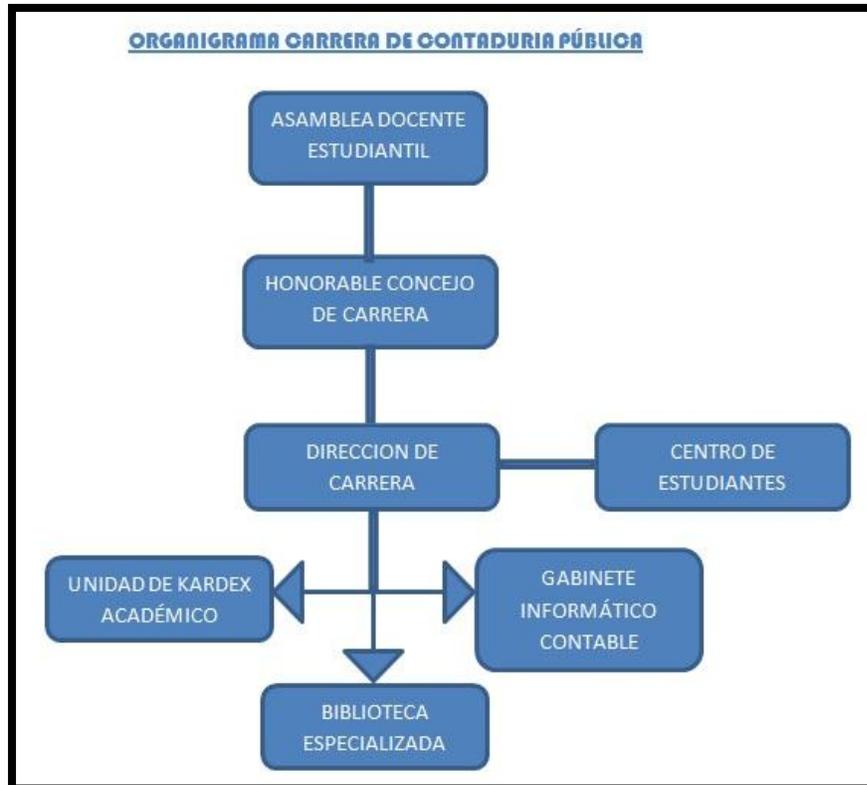
Sedes académicas en diferentes municipios del departamento de La Paz:

- Achacachi
- Viacha
- Palos Blancos

La carrera de Contaduría Pública se encuentra estructurado y organizado jerárquicamente como se puede apreciar en la **Figura 1.1**.

1.2.2. Organigrama Institucional de la Carrera Contaduría Pública

Figura Nº 1.1: Organigrama Carrera Contaduría Pública



Fuente: Carrera Contaduría pública

1.2.3. Misión

Formar profesionales competitivos, idóneos con alto grado moral y ética en el contexto de exigencia y calidad en un ambiente de respeto, con pensamiento crítico reflexivo e investigadores con conciencia social y alto nivel de eficiencia académica en el área de contaduría y auditoría, acorde al avance de la ciencia y la tecnología, y a los requerimientos del mercado laboral, para contribuir al desarrollo de la sociedad y resolver los problemas relacionados a todos los sectores económicos en general.

1.2.4. Visión

Ser una carrera líder en la formación de profesionales con calidad de excelencia académica comprometida con su entorno social y la pluriculturalidad del país, ser una carrera acreditada a nivel nacional e internacional, proyectada al desarrollo de sus actividades académicas, científicas, productivas y tecnológicas de interacción social, para priorizar e incentivar la investigación en los campos de la contaduría y auditoría, relacionando la teoría con la práctica para contribuir a la transformación de la estructura económica, social, ambiental, cultural y política a nivel local, regional, nacional e internacional.

1.2.5. Antecedentes Académicos

1.2.6. Antecedentes Nacionales

En relación a los trabajos realizados y/o implementados dentro el entorno de estudio, se hace referencia a los siguientes proyectos:

“Sistema de seguimiento académico carrera de arquitectura Universidad Pública de El Alto”. Esmerado por Víctor Hugo Calcina López del año 2007. Empleó PHP como lenguaje de programación y MYSQL como gestor de base de datos en su implementación. Tiene como propósito facilitar el trabajo de su personal y de los estudiantes, haciendo que los primeros mejoren la administración que se genera durante las actividades, especialmente de inscripción que se lleva a cabo en esta institución académica y los segundos disfruten de las ventajas en cuanto a información. Este proyecto tiene una analogía con el sistema indicado, automatizar el proceso de inscripción, así también se usará para la implementación del lenguaje de programación PHP.

“Sistema de inscripción y seguimiento académico vía Web caso carrera de electrónica Escuela Industrial Superior Pedro Domingo Murillo”. Hecho por Víctor Hugo Márquez del año 2010. Para su implementación utilizó el lenguaje de programación PHP y MYSQL como gestor de base de datos.

El sistema tiene como objetivo facilitar el proceso de inscripción de estudiantes a materias, registro y seguimiento académico de estudiantes que ingresan y egresan, notas por periodo, récord académico y asignación de horarios de esta forma se optimiza el trabajo académico administrativo de la carrera. El proyecto citado tiene una relación con el sistema propuesto, automatizará los procesos de inscripción, seguimiento académico y consultas de los estudiantes vía web.

“Sistema de gestión académica con portal Web para la Escuela de Idiomas del Ejército Cochabamba”. Procedido por Jorge Fernando Cortez Guillen del año 2008. Para su implementación se utilizó el lenguaje de programación PHP, CodeIgniter junto a Ajax y CMS (Manejador de contenidos). El mismo tiene como propósito llevar a cabo el sitio web, cobro para los alumnos inscritos y gestión académica.

El proyecto citado tiene una relación con el sistema planteado, realizará la gestión académica y desarrollo del sitio web haciendo uso de las herramientas del lenguaje de programación de PHP, CodeIgniter (Modelo Vista Controlador).

1.2.7. Antecedentes Internacionales

Respecto a los trabajos similares realizados en el campo de estudio, se puede mencionar los siguientes:

“Desarrollo del sistema de gestión académica para la Escuela Gonzalo Rubio Orbe de Otavalo”. Realizado por Marco Vinicio Gualacata Puma, Sangolquí-Ecuador, enero del año 2012. Se usó Microsoft Visual Studio y Microsoft SQL Server 2005 como gestor de base de datos para su implementación. El objetivo que tiene es brindar una serie de servicios como: matriculación, gestión de notas, registro de estudiantes y otros.

Este proyecto tiene semejanza con el sistema planteado, dado que automatizará los procesos de matriculación, inscripción de estudiantes y consulta de notas, consulta de materias inscritas vía web.

“**Sistema de información para gestión académica (SIGA)**”. Elaborado por Angélica María Díaz Ortega, William Orlando Duran Sandoval y Rafael Enrique Quijano Giraldo, Bucaramanga-Colombia, agosto del año 2010. Para su implementación se usó el lenguaje de programación PHP y Mysql como gestor de base de datos.

El sistema tiene por objetivo realizar inscripciones de los aspirantes, matrículas, ingreso de notas, y consultas a través de Internet, y está dedicado a la Gestión Académica de la Institución d Fundación Tecnológica (**FITEC**). Este proyecto tiene similitud con el sistema propuesto, se automatizará los procesos de inscripción, matriculación, registro de notas y consultas de los estudiantes Vía Web, así también se usará para la implementación el lenguaje de programación PHP.

1.3. Planteamiento del Problema

En cada gestión académica el incremento de la población estudiantil en la Universidad Pública De El Alto es adsorbida por las 35 carreras que ofrece esta casa superior de estudios y la carrera de Contaduría Pública siendo una de las carreras de mayor demanda académica, crece anualmente con un promedio constante, este parámetro hace que la información se genere en grandes cantidades y en función del tiempo hace que sea difícil manejarla o manipularla adecuadamente, sin dejar de lado que existen otros procesos que requieren datos ordenados para un determinado fin. Además se incrementa algunos sub procesos realizados de forma semi manual, si valga el termino y está concentrado en pocas personas que administran los mismos, generando dificultad y mora en estas actividades para generar resultados aceptable.

En la actualidad la carrera Contaduría Pública cuenta con un sistema genéricamente implementado, que cubre algunos procesos en lo que se refiere a la gestión académica por lo que la comunidad docente y estudiantil de la carrera observa dificultades en actividades como: **gestión de pensum, planificación académica, inscripción, registro de notas y seguimiento académico de**

estudiantes, por lo que se genera un ambiente desordenado que conlleva una inseguridad documental además de la información.

A continuación, se presenta los procesos mencionados anteriormente:

- **Gestión de pensum**, no se establece un orden gradual y armónico de asignaturas con sus respectivas características (código, asignatura, nivel descripción, régimen, menciones, prerrequisitos, tiempo y número de horas de teoría y de práctica de cada asignatura) que corresponden a una carrera. Lo cual dificulta la emisión de documentos como historiales y récord académicos por el manejo simultáneo de varios planes de estudio.
- **Planificación académica**, existen muchos factores que inciden en esto: asignar docentes por sección y horario, número de estudiantes inscritos, asignación de aulas, asignaturas, paralelos y otros. Este proceso se realiza al inicio de cada gestión académica en la que se distribuye aulas y docentes de acuerdo con el número de estudiantes y bloques horarios.
- **Inscripción de estudiantes**, el proceso se realiza manualmente mediante el llenado de formularios de inscripción por los estudiantes, posteriormente el personal de Kardex se encarga de ordenar los mismos por asignaturas o según el record del estudiante.
- **Registro de notas**, este proceso es realizado manualmente al finalizar la gestión académica, son transcritas en hojas de texto de Microsoft Excel, denominadas Acta de Notas, los cuales son sellados y archivados en las actas de cada gestión.
- **Seguimiento académico de estudiantes**, se realiza de manera manual, la información se guarda en libro de actas y medios físicos sin respaldo alguno, se produce una demora considerable en la obtención de la información como: datos personales, asignaturas inscritas, récord, historial de notas, calificaciones de las diferentes asignaturas de la gestión académica correspondiente.

1.3.1. Problema Principal

Los métodos utilizados actualmente para la gestión de pensum, planificación académica, inscripción, registro de notas y seguimiento de estudiantes, procesos que forman parte del área académica de la carrera de Contaduría Pública no son apropiadas, dado que se los realiza mayormente de manera manual y con un comportamiento desorganizado, generando información poco confiable, redundante e inoportuna.

1.3.2. Problemas Secundarios

- Dificultad en procesos que actualmente se las realiza de manera manual para la gestión de pensum, por realizarse de esta forma y otra en forma semiautomática presentando deficiencias e inconvenientes a la hora de generar reportes y tomar decisiones
- Inconveniente en el proceso de planificación académica, por realizarse de forma mecánica, lo que genera repetir y duplicar las mismas tareas.
- Lentitud en el proceso de inscripción, por realizarse de forma manual, lo que ocasiona a cometer errores ortográficos en los datos generales de los estudiantes, afectando significativamente a la hora de emitir los certificados, asimismo se produce redundancia de datos, trabajo excesivo en el traslado, búsqueda de información y pérdida de documentación.
- Dificultad en el registro de notas, por realizarse de forma manual, lo que provoca un retardo en la entrega de notas a las instancias correspondientes.
- Demora considerable en el proceso de seguimiento académico, por realizarse de manera manual, haciendo que la información sea excesiva y difícil de manejarla adecuadamente.
- Tardanza en la entrega de reportes como record e historial académico, certificaciones estudiantiles, boletas de inscripción, entre otros, debido a que se realiza la recopilación de información de manera manual, lo cual es una tarea lenta y repetitiva.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un **Sistema de Información para la Gestión Académica en un entorno Web para la Unidad de Kardex Académico Carrera de Contaduría Pública de la Universidad Pública de El Alto**, con la finalidad de que la información sea generada de manera oportuna, efectiva y confiable para el uso y catalogación de la misma.

1.4.2. Objetivos Específicos

- Análisis e implementación de la Base de Datos del sistema.
- Gestionar el registro de estudiantes activos, pasivos y nuevos, además del personal docente.
- Realizar la administración y centralización de notas de estudiantes.
- Plasmar el registro de pensum de forma automatizada mediante el desarrollo del módulo para establecer el cumplimiento de los prerrequisitos establecidos en las asignaturas
- Automatizar los procesos de planificación académica como ser: inscripción de estudiantes y el registro de notas, para evitar el retraso, duplicidad, pérdida y acumulación de la documentación física.
- Simplificar la consulta de estudiantes que realicen el seguimiento académico personal, sistematizando el proceso para acceder a su información como: récord académico, historial de notas, asignaturas inscritas e información personal.
- Generar reportes sistematizados mediante el desarrollo de un módulo, que permita obtener información confiable, oportuna y ordenada.
- Incentivar la competencia académica en la comunidad estudiantil y docente, dentro de los procesos de gestión académica en coordinación con la parte ejecutiva y técnica de la carrera Contaduría Pública.

1.5. Justificación

1.5.1. Justificación Técnica

La carrera de Contaduría Pública de la Universidad Pública de El Alto cuenta con equipamiento de hardware, software y una estructura de red con acceso a Internet además de la infraestructura para alojar un servidor propio, cualidades importantes para el desarrollo del proyecto. El desarrollo del sistema tendrá como base el software libre para lo que se usara: Lenguaje de programación PHP, Servidor web Apache, Sistema gestor datos MariaDB y Framework CodeIgniter 3.

Con las condiciones detalladas la factibilidad técnica del proyecto es viable, porque la carrera objeto cuenta con equipos de computación, acceso a Internet, conexiones de red, como también posee el personal asignado para el manejo del sistema y las tecnologías usadas para el desarrollo son libres, no requieren licencias.

1.5.2. Justificación Económica

El sistema web se desarrollará bajo herramientas de software libre, el cual no generará costo económico para llevar a cabo su desarrollo. Los procesos serán automatizados, esto permitirá optimizar el tiempo de acceso a la información u datos en general, reducir el excesivo manejo de documentación física y reducir los gastos que representa el material de escritorio.

El proyecto es factible económicamente en su elaboración e implementación, ya que las posibilidades tecnológicas así lo han permitido, es decir, que las herramientas de construcción de proyecto de grado son de uso libre lo cual contribuye a la optimización de los costos del proyecto. La implementación del sistema web podrá generar ingresos para la carrera de manera más óptima y catalogada, por los aportes por inscripción de estudiantes nuevos y antiguos, así como la opción de difundir información por tener atención de calidad, estableciendo normativas que reglamenten el cumplimiento de los mismos.

1.5.3. Justificación Social

Se optimizará en tiempo de atención y la respuesta a los usuarios (docentes y estudiantes) además de dotar a la Unidad de Kardex Académico con una herramienta integral para ejecutar los procesos administrativos correspondientes a la unidad:

- Los estudiantes podrán realizar su inscripción de manera virtual vía web, así mismo realizarán consultas de las notas de su carrera universitaria.
- La Unidad de Kardex Académico optimizará la atención de los trámites de los estudiantes universitarios.
- Se contará con datos estadísticos de gestiones anteriores y actuales.
- El director de carrera tendrá acceso a información confiable y oportuna mediante reportes generados por el sistema mediante Kardex Académico.

1.6. Metodología Web

En el desarrollo del sistema propuesto se adoptará los siguientes puntos:

1.6.1. Metodología de Ingeniería Web

Metodología **UWE** (Ingeniería Web basado en UML) que proporciona técnicas y actividades para el modelado y construcción de las aplicaciones Web, tiene como herramienta principal al UML (Lenguaje de Modelado Unificado) para la recolección de requerimientos en el desarrollo Orientado a Objetos para un estudio personalizado a detalle. (Galiano, 2012)

1.6.2. Técnicas de Investigación

En el presente proyecto se recurrió a las técnicas como: **Observación, Entrevista personal y Revisión de documentación.** (Cespedes, 2001)

1.6.3. Métricas de Calidad

Para la evaluación de la calidad se hará uso del estándar **ISO/IEC (International Standard Organization) 9126** que identifica seis factores clave de calidad de software, los mismos coinciden con los definidos para la evaluación de calidad de

aplicaciones web, como son factores de: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad. (Olsina, 1999)

1.6.4. Métodos de Estimación de Costos

Para la estimación de costos se empleará el modelo **COCOMO II (COnsturctive COst MOdel)**, que permite la estimación del esfuerzo y tiempo requerido para desarrollar un producto. (Boehm, 1981)

La estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos.

1.7. Herramientas

Para la implementación del sistema se mantiene el lineamiento de utilizar software libre, se puede mencionar las siguientes herramientas:

1.7.1. Lenguaje de programación

Para el desarrollo del sistema se tomó en cuenta tres lenguajes de programación más usadas como se presenta en el siguiente cuadro **TABLA 1.1 (VER ANEXO A)**. (Villalobos, 2010). Tomando en cuenta lo anterior, se eligió el lenguaje de programación **PHP**, que se utiliza para procesar instrucciones orientadas a la Web.

1.7.2. Sistema gestor de base de datos

Se optó por **MariaDB**, el mismo permite modificar, almacenar y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, recuperación de información entre otras.

1.7.3. Servidor Web Apache

Programa de servidor HTTP Web de código abierto, actualmente es uno de los servidores Web más utilizado en la red.

1.7.4. Otras herramientas

- **Framework CodeIgniter 3 (MVC).** Es la técnica más ordenada para crear una aplicación independiente del lenguaje de programación que se utilice. (Gomez F. , 2015)
- **HTML5 (Hypertext Markup Language).** Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es que gracias a HTML5 podemos crear la estructura de una página web. (Guioteca, 2011)

1.8. Límites y Alcances

1.8.1. Límites

El presente proyecto de grado sistema de información para la gestión académica en un entorno Web para la Unida de Kardex Académico de la carrera Contaduría Pública ,Universidad Pública de El Alto, no realizara lo siguiente:

- No realizara convalidaciones.
- No llevara el control de asistencia del personal docente.
- No llevara el control de asistencia del estudiante.
- No contempla el modulo docente por la susceptibilidad por parte de los docentes y establecida por la carrera.
- En el módulo de evaluación docente solo gestionara datos numéricos del criterio estudiantil durante el proceso evaluación al docente no así el resultado final.
- El sistema no contempla aspectos como inventario de activos, generación y seguimiento de planillas, seguimiento de trámites, control de correspondencia ni control del personal administrativo.

1.8.2. Alcances

Para la implementación del sistema se contemplarán los siguientes procesos:

- Habilitación de usuarios al sistema con distintos niveles de acceso de acuerdo a sus requerimientos.

- Registrará en la base de datos la información de inscripción del estudiante, plan de estudios, carreras, asignaturas, menciones, homologaciones, prerrequisitos, datos personales, notas y otros.
- Procederá a la administración de gestión del plan de estudios.
- Registro y asignación de docentes a paralelos por asignatura según nombramiento docente y asignatura.
- Llevará a cabo la inscripción de estudiantes nuevos y antiguos.
- Llevará a cabo el registro de notas finales.
- Seguimiento académico de estudiantes.
- Obtención de certificaciones, record e historial de notas, para trámites administrativos.
- Coadyuvará en la evaluación académica de docentes en cada gestión según reglamentación.

El sistema se centrará en sistematizar los procesos de gestión del plan de estudios, planificación académica, inscripción, registro de notas y seguimiento de estudiantes, procesos que se siguen para la gestión del área académica.

1.9. Aportes

El principal aporte es desarrollar un sistema en un entorno Web para la Unidad de Kardex Académico de la carrera de Contaduría Pública de la Universidad Pública de El Alto, que concentre la administración y difusión de información para la gestión académica que se genera en la carrera a través de la web, brindando a toda la población universitaria Docente-Estudiantil una herramienta disponible en todo momento, así también permitiendo a la institución fortalecimiento y prestigio académico.

De la misma forma el presente proyecto aportará como guía para el desarrollo de sistemas, además el sistema será desarrollado con tecnologías de desarrollo de software actuales, (Framework CodeIgniter 3 (modelo vista controlador,) y otros), lo cual servirá como referencia a futuros proyectos.

CAPITULO II: MARCO TEÓRICO

2. Marco Teórico

Es importante considerar el camino más adecuado, en la utilización de herramientas y métodos que permitan el análisis, para lograr alcanzar los objetivos planteados en el marco preliminar. En el presente capítulo se da conocer el marco teórico, recopilando toda la información para el sustento de estudio que se realiza, estos elementos teóricos extraídos de la revisión de la literatura consisten en la base para la descripción y explicación del problema a solucionar.

2.1. Sistema

El termino sistema se utiliza habitualmente con múltiples sentidos, tantos que resulta difícil dar una definición única que los abarque todos al mismo tiempo y que sea lo suficientemente precisa para servir a propósitos específicos. Podemos partir de la definición de sistema como conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto. Se trata de una definición sencilla pero que pone de manifiesto los caracteres relevantes de lo que constituye el denominado enfoque sistémico, contemplación del todo y no de las partes aisladamente, acento en las relaciones entre las partes y consideración teleológica al tener en cuenta los propósitos u objetivos del sistema, especialmente válida para los sistemas creados por el hombre. (MCLeod, 2000)

2.2. Información

Múltiples son las definiciones que se encuentran presentes a la hora de la determinación de su contenido. La información no se ha definido solo desde el punto de vista matemático o técnico; su conceptualización abarca enfoques filosóficos, cibernéticos y otros, basados en las denominadas ciencias de la información. (Long, 1999)

La información social es un sistema de control, en tanto que es la propagación de consignas que deberíamos de creer o hacer que creemos. En tal sentido la

información es un conjunto organizado de datos capaz de cambiar el estado de conocimiento en el sentido de las consignas transmitidas. (Flores J., 2011)

La información está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información es un recurso que otorga significado o sentido a la realidad, ya que mediante códigos y conjuntos de datos, da origen a los modelos de pensamiento humano.

2.3. Sistema de Información

Es un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control de la organización.” (Scott, 2010)

Un sistema de información puede definirse técnicamente como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución. Además, para apoyar a la toma de las decisiones, la coordinación y el control, los sistemas de información pueden también ayudar a los administradores y al personal a analizar problemas, visualizar cuestiones complejas y crear nuevos productos. (Suarez, 2012)

La finalidad de los sistemas de información, dentro de una organización, es procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

Además de apoyar en la toma de decisiones, la coordinación y el control, los sistemas de información son indispensables para ayudar a los gerentes a mantener ordenada su compañía, a analizar todo lo que por ella. Existen elementos de naturaleza diversa y están incluyen:

- El equipo computacional: es decir el hardware necesario para que el sistema de información pueda operar. Lo constituyen las computadoras y el equipo periférico que se conectan a ellas.

- El recurso humano: que interactúa con el sistema de información, el cual está formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.
- Los datos o información fuente: que son introducidos en el sistema, son todas las entradas que este necesita para generar como resultados la información que se desee.
- Los programas: que son ejecutados por las computadoras y producen diferentes tipos de resultados. Los programas son parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

El uso eficaz de los sistemas de información implica entender sobre organización, administración y tecnología de información que da forma a los sistemas como se puede apreciar en la **Figura N° 2.1**.

Figura N°2 1: Actividades de un Sistema de Información



Fuente: (Suarez, 2012)

2.4. Gestión

Gestión es la acción y el efecto de gestionar y administrar. De una forma más específica, entendida como un trámite necesario para conseguir algo o resolver un asunto, habitualmente de carácter administrativo o que conlleva documentación. (Gomez J. , 2011)

Gestión es también un conjunto de acciones u operaciones relacionadas con la administración y dirección de una organización. (Flores J., 2011)

Este concepto se utiliza para hablar de proyectos o en general de cualquier tipo de cualquier actividad que requiera procesos de planificación, desarrollo, implementación y control.

2.5. Gestión Académica

La gestión académica es aquel ámbito de acción orientada a facilitar y mejorar los procesos formativos que imparten las instituciones de educación superior. Esto incluye la definición, implementación y seguimiento de procesos de mejoramiento continuo que les permitan responder y anteponerse a las necesidades formativas de sus estudiantes, a las exigencias del entorno, del mercado laboral y la política pública. (CnaChile, 2018)

Facilita a los estudiantes todos los trámites y procedimientos que comporta su formación: matriculas, becas de régimen general y de equidad, expediente, obtención de títulos, entre otros, además, también se ocupa de los planes de estudios y las normativas de ámbito académico y docente , la elaboración de esta oferta y su difusión, los procesos de preinscripción, admisión y matrícula. (Coulouris, 2001)

La Gestión Académica comprende las siguientes cuestiones:

- ✓ Para solicitar información general.
- ✓ Certificaciones.
- ✓ Compulsa de documentos.
- ✓ Convalidaciones de estudios.
- ✓ Matriculación.

- ✓ Solicitudes de traslados de expedientes.
- ✓ Tramitación de títulos.
- ✓ Accesos al segundo ciclo.
- ✓ Informaciones relacionadas con la normativa académica y gestión del expediente, así como otras informaciones de tipo general.
- ✓ Cambios de asignaturas.
- ✓ Cambios de grupo.
- ✓ Información de los planes de estudios.
- ✓ Tramitación de solicitudes.
- ✓ Convocatorias extraordinarias de exámenes.
- ✓ Prácticas en empresas, entidades e instituciones.

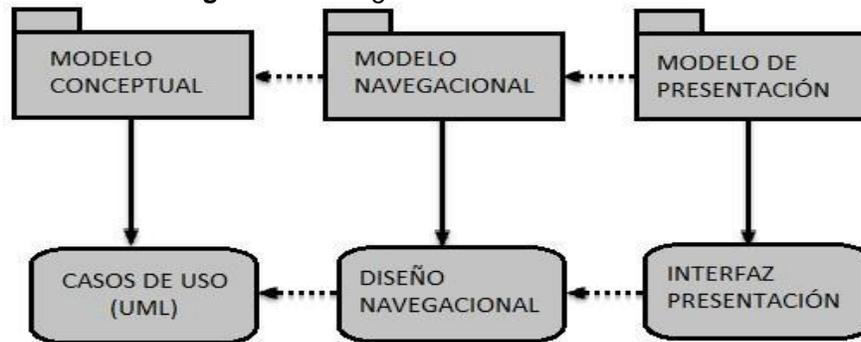
2.6. Metodología Web

Son procesos que permiten estructurar, comunicar, entender, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios de software.

2.6.1. Metodología Basada en UML (UWE)

UWE (UML Web Engineering), en español **Ingeniería Web Basada en UML** es una metodología que permite especificar de mejor manera una aplicación web enfocado sobre el diseño sistemático, personalización y generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web, UWE describe una metodología de diseño sistemática basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Figura N°2 2: Ingeniería Web Basada en UML



Fuente: (Vargas, 2017)

2.6.2. Actividades de modelado UWE

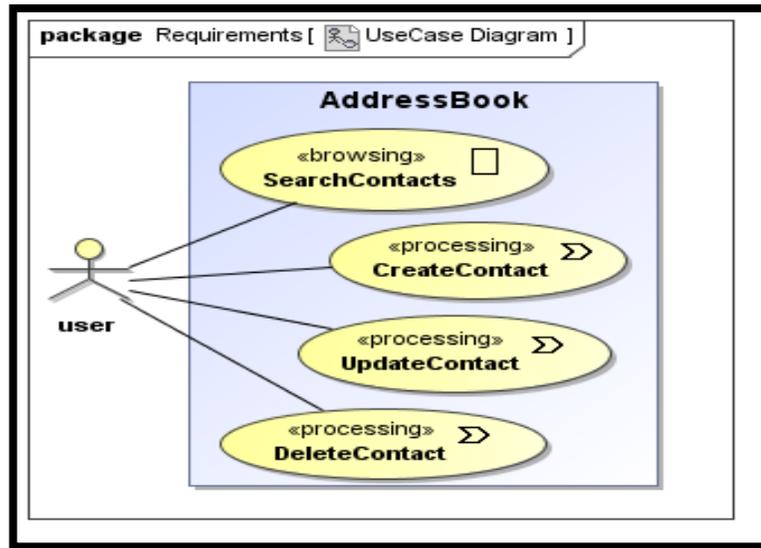
UWE proporciona guías para la construcción de modelos de forma sistemática enfocadas en el estudio de casos de uso. Las actividades del modelado son Análisis de Requerimientos, Diseño Conceptual, Diseño Navegacional y Diseño de Presentación producen los siguientes artefactos:

- Modelo de Casos de Uso
- Modelo Conceptual
- Modelo de Navegación
- Modelo de Presentación.

2.6.2.1. Modelo de casos de uso

En UWE, se utilizan diagramas de casos de uso UML para definir los requerimientos funcionales de la aplicación Web. Un modelo de casos de uso Web define la funcionalidad de la aplicación como se puede apreciar en la siguiente figura.

Figura Nº2 3: Diagrama de caso de uso en la Metodología UWE

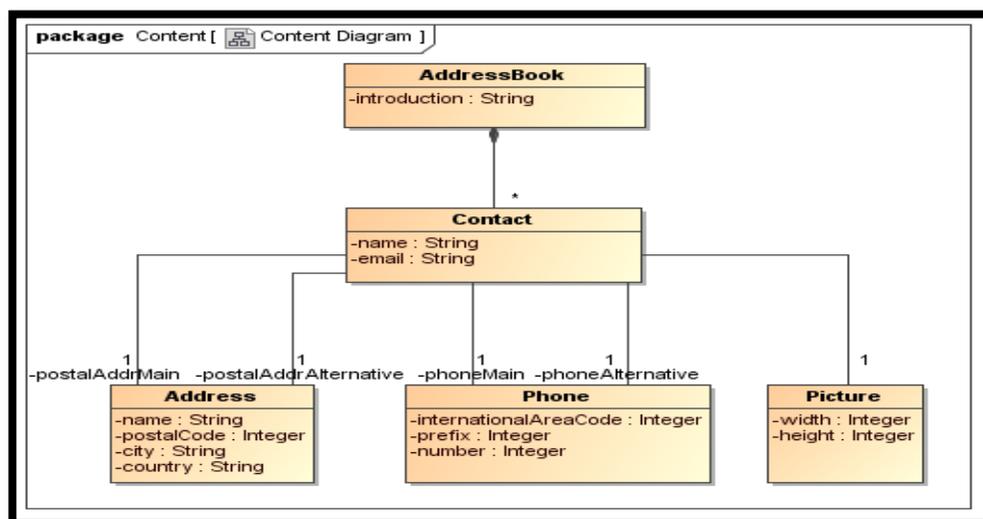


Fuente: (Zeballos, 2017)

2.6.2.2. Modelo de contenido

La construcción de este modelo de contenido se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. Este es un diagrama normal de clases, los cuales describen las estructuras estáticas de un sistema.

Figura Nº2 4: Diagrama de contenido en la Metodología UWE

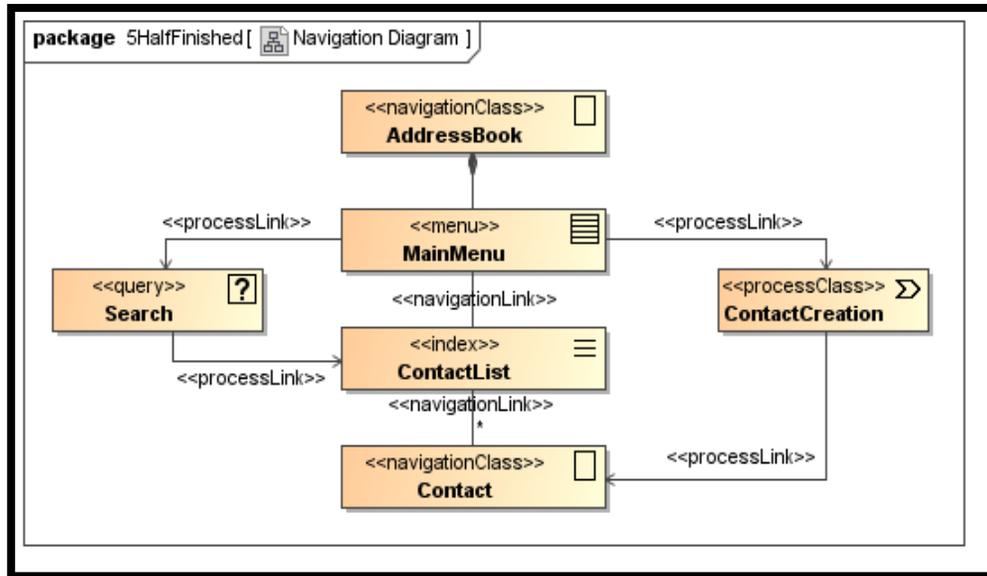


Fuente: (Zeballos, 2017)

2.6.2.3. Modelo de navegación

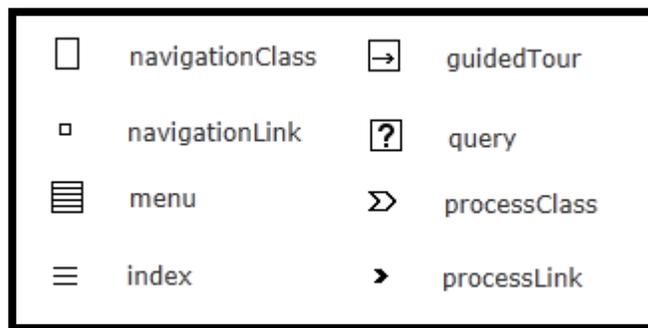
Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. Ello significa que necesitamos un diagrama de conteniendo nodos y enlaces.

Figura N°2 5: Diagrama de navegación en la Metodología UWE



Fuente: (Zeballos, 2017)

Figura N°2 6: Estereotipos del Diagrama de navegación

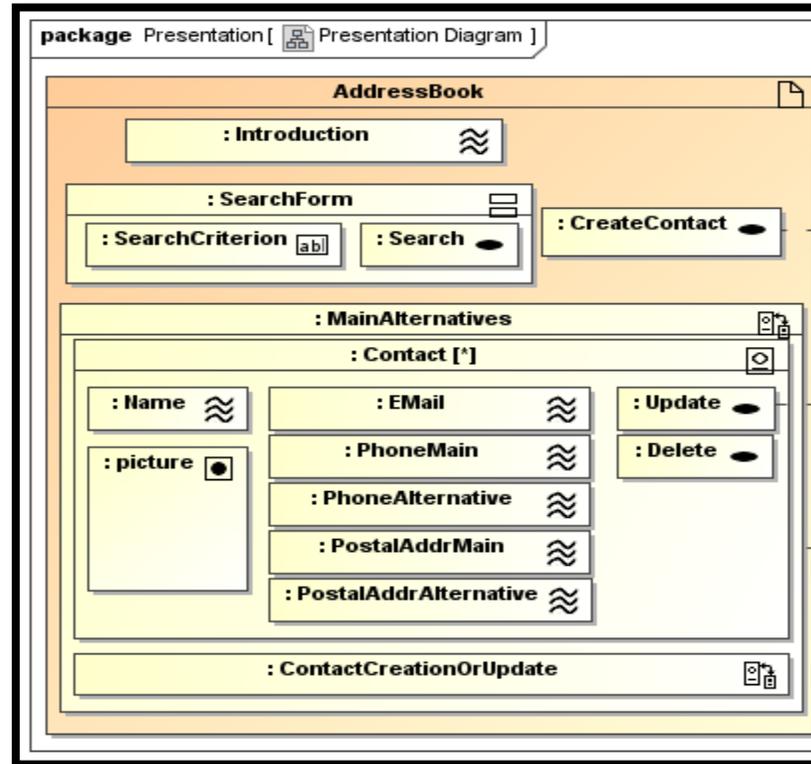


Fuente: (Zeballos, 2017)

2.6.2.4. Modelo de presentación

Describe donde y como los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

Figura N°2 7: Diagrama de presentación en la Metodología UWE



Fuente: (Zeballos, 2017)

2.6.3. Fase de desarrollo UWE

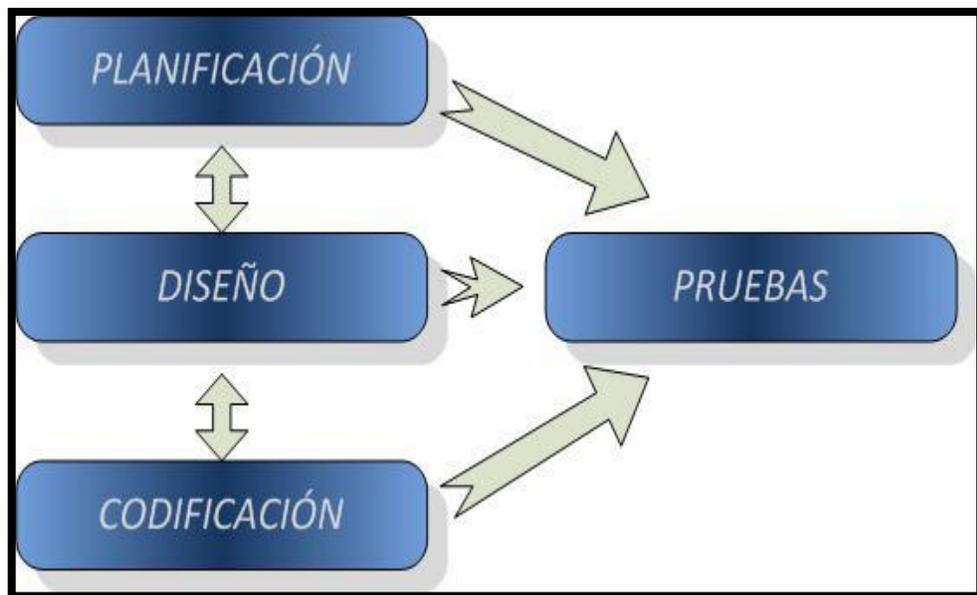
Con respecto al proceso de creación de la aplicación, UWE se vale mediante el uso de metodologías estándares reconocidas como UML principalmente y también del lenguaje de especificaciones de restricciones asociados **OCL** (**Object Constraint Language**, en español lenguaje de restricciones para objetos). Para recolectar los requerimientos necesarios de las aplicaciones web, esta metodología propone una aplicación utilizada en el proceso de creación, mismo que se divide en las siguientes actividades:

- a) Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.
- b) 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.
- Diagrama de Casos de Usos
 - Diagrama Conceptual
 - Diagrama Físico
 - Diagrama de Clases
 - Modelo Navegacional
 - Modelo de Presentación
- c) Codificación del software:** Durante esta etapa se realizan las tareas que se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- d) Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- e) La Instalación o Fase de Implementación:** es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.
- Implementación y Lanzamiento: En la implementación de la Pagina Web es recomendable utilizar estándares (HTML, XHTML...) para asegurar la futura compatibilidad y escalabilidad del sitio. Una vez implementada la página web y aprobada su funcionalidad se procede al lanzamiento del sitio.
- f) El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de

errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

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

Figura N°2 8: Fases de desarrollo UWE



Fuente: (Zeballos, 2017)

2.7. Pruebas de Software

Las pruebas de software son procesos que permiten verificar la calidad de un producto de software. Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene. Para determinar el nivel de calidad se deben efectuar unas pruebas que permiten comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Las pruebas del software, testing o beta testing es un proceso usado para identificar posibles fallos de implementación, calidad o usabilidad de un programa de ordenador. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Únicamente un proceso de verificación formal puede probar que no existen errores.

2.7.1. Pruebas de Caja Blanca

En programación se denomina cajas blancas a un tipo de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas.

Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hayan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritméticos, pruebas de camino de datos (definición de uso de variables), comprobación de bucles (se verifican los bucles para 0,1 y n y luego para las iteraciones máximos menos uno y más uno).

Las pruebas de caja blanca se llevan a cabo en primer lugar; sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas. En los sistemas orientados a objetos, las pruebas de caja blanca pueden aplicarse a los métodos de la clase, pero según varias opciones, ese esfuerzo debería dedicarse a otro tipo de pruebas más especializadas (un argumento podría ser que los métodos de una clase suelen ser menos complejos que los de una función de programación estructurada). La prueba de la ruta básica es una técnica de la prueba de la caja blanca, la cual se basa en el manejo de grafos, en donde se maneja la definición de complejidad ciclomática, esta es una métrica de software que proporciona una medida cuantitativa de la complejidad lógica de un programa.

2.7.2. Pruebas de Caja Negra

En teoría de sistemas y física, se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio

que le rodea (en ocasiones, otros elementos que también podrían ser cajas negras) entendiendo que es lo que se hace, pero sin dar importancia a como lo hace. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, es decir, en cambio no se precisa definir ni conocer los detalles internos de su funcionamiento.

2.8. Métricas de Calidad de Software

Una de las metas principales de la evaluación y comparación de calidad del software que se está realizando, radica en medir, analizar y comprender el grado de cumplimiento de un conjunto de características y atributos con respecto a los requerimientos de calidad establecidos, para un perfil de usuario y dominio de aplicación dados.

Por una parte, los desarrollos centrados en la Web en los más diversos dominios de aplicación como comercio electrónico, sistemas académicos, financieros, entre otros, se están tornando cada vez más en sistemas complejos.

Estos estándares en particular, han sido desarrollados teniendo en cuenta que las características y métricas asociadas se pueden usar no sólo para evaluar los productos software, sino también para definir requisitos de calidad y otros usos.

2.8.1. Métricas de Calidad ISO/IEC 9126

El estándar ISO/IEC 9126 define un modelo de calidad del software en el que la calidad se define como la totalidad de características relacionadas con su habilidad para satisfacer necesidades establecidas o especificadas. Los atributos de calidad se clasifican según seis características (funcionales, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), las cuales a su vez se subdividen en sub-características.

La ISO/IEC 9126 se basa en los siguientes fundamentos:

El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso del software por parte de los usuarios. Es necesario comprender las necesidades

reales de los usuarios con tanto detalle como sea posible (requisitos), razones por la cual se definen los siguientes aspectos de calidad.

- a) **Interna:** Medible a partir de las características intrínsecas, como el código fuente.
- b) **Externa:** Medible en el comportamiento del producto, como en una prueba.
- c) **En Uso:** Durante la utilización efectiva por parte del usuario.

ISO/IEC 9126 nos permite definir un modelo de calidad, para nuestra empresa, en base a las 6 características que se indican en el estándar .El modelo de calidad que definamos nos dará como resultado el grado de calidad de cada uno de nuestros productos software. Estas se listan a continuación:

2.8.1.1. Funcionalidad

En este grupo se conjunta una serie de atributos que permiten calificar si nuestro producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

- **Adecuación:** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- **Exactitud:** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- **Interoperabilidad:** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- **Conformidad:** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- **Seguridad:** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

2.8.1.2. Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub-características que el estándar sugiere son:

- **Madurez:** Permite medir la frecuencia de falla por errores en el software.
- **Tolerancia o fallas:** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperabilidad:** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.
- **Conformidad:** La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

2.8.1.3. Usabilidad

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

- **Comprensibilidad:** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprendizaje:** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad:** Agrupa los conceptos que evalúan la operación y el control del sistema.
- **Atracción:** La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario.
- **Conformidad:** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

2.8.1.4. Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- **Comportamiento de Tiempos:** Atributos del software relativo a los tiempos de respuesta u de procesamiento de los datos.
- **Utilización de Recursos:** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.
- **Conformidad:** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

2.8.1.5. Mantenibilidad

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya que por la corrección de errores o por el incremento de funcionalidad. En este caso se tienen los siguientes factores:

- **Capacidad de Análisis:** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- **Cambiabilidad:** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- **Estabilidad:** Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- **Facilidad de Prueba:** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.
- **Conformidad:** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

2.8.1.6. Portabilidad

En este caso se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- **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.
- **Capacidad de Reemplazo:** Se refiere a la oportunidad u el esfuerzo usado en sustituir el software por otro producto con funciones similares
- **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.

2.9. Tecnologías para el Desarrollo de Aplicaciones Web.

2.9.1. Lenguaje de programación PHP

La herramienta tecnológica a utilizar será el PHP, cuyas siglas responden a un acrónimo recursivo (**PHP: hypertext preprocessor**) que es una herramienta orientada a objetos que nos proporciona un nivel extremadamente alto de velocidad y rendimiento. (Guioteca, 2011)

Características de PHP:

- Capacidad de acceder a la mayoría de las bases de datos que se utilizan en la actualidad.
- Leer los datos desde diferentes fuentes, incluyendo datos que pueden introducir los usuarios desde formas HTML y manipularlos de forma sencilla.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones)
- Es Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de programación orientada a objetos.

En el mercado nacional se puede encontrar personal altamente capacitado con experiencia en este producto. PHP proporciona un juego completo de herramientas que facilitan el desarrollo rápido de aplicaciones.

Tabla Nº 2. 1: Entorno PHP

ENTORNO PHP				
SOLUCION DEL CLIENTE				
PHP	AJAX	JAVASCRIPT	HTML	LENGUAJE DE PROGRAMACION
ECLIPSE	NETBEANS		ZEND STUDIO	AMBIENTE INTEGRADO DE DESARROLLLO
SAPPHIRE		CODEIGNITER		FRAMEWORK
ORACLE	MYSQL	POSTGRESQL	MARIADB	BASE DE DATOS
WINDOWS		LINUX		SISTEMA OPERATIVO

Fuente: (Guioteca, 2011)

2.9.2. CodeIgniter

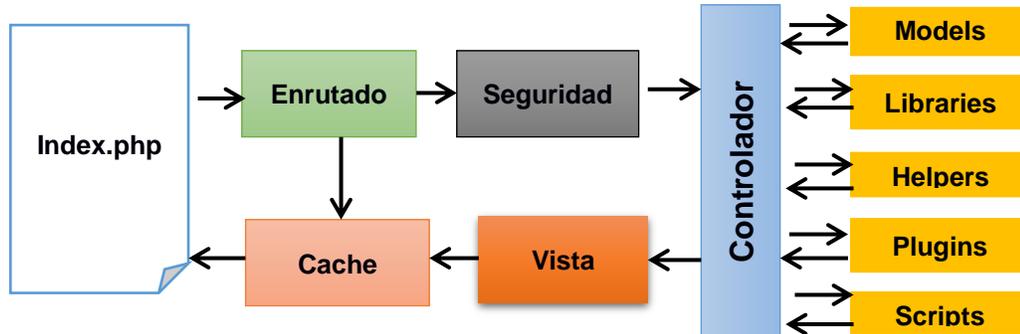
Es un potente Framework PHP con un tamaño muy pequeño, construido para programadores PHP que necesitan una herramienta simple y elegante para crear aplicaciones Web con todas las funciones. Es un producto de código libre, libre de uso para cualquier aplicación. **(Gomez, 2015)**

Características principales:

- Sistema basado en Modelo-Vista-Controlador.
- Compatible con PHP.
- Extremadamente liviano.
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Soporte de Active Record para Base de Datos.
- Formulario y Validación de datos.
- Seguridad y filtro XSS.
- Manejo de Sesión.
- Librería de manipulación de imagen (cortar, redimensionar, rotar, etc.).

- Encriptación de datos.

Figura N°2 9: Diagrama de Flujo de la Aplicación



Fuente: (Gomez F. , 2015)

- El **index.php** sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
- El **Router** examina la petición HTTP para determinar que debe ser hecho con él.
- Si un archivo de **caché** existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
- Seguridad**. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
- El **controlador** carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.
- La **Vista** finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas.

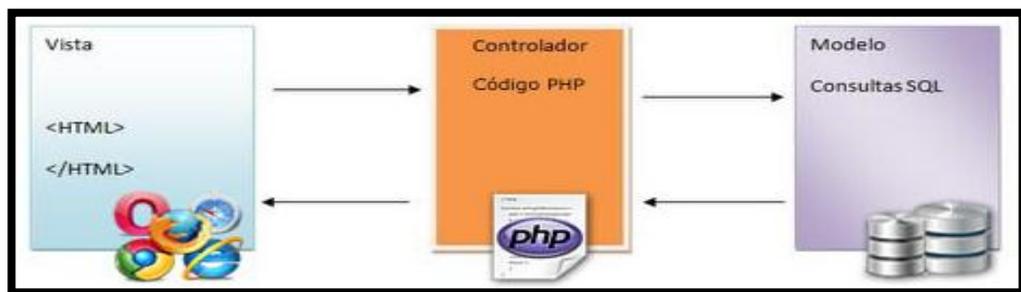
2.9.2.1. Modelo Vista Controlador

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación. En la práctica, permite que sus páginas web contengan mínima codificación ya que la presentación es separada del código PHP.

- El **Modelo** representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos.
- La **Vista** es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".
- El **Controlador** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web.

CodeIgniter tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. Si no necesita agregar separación, o descubre que mantener los modelos requiera más complejidad que quería, puede ignorarlos y construir su aplicación mínimamente usando Controladores y Vistas. También permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar en una forma que hace que tenga más sentido.

Figura N°2 10: Funcionamiento de Modelo, Vista y Contralor



Fuente: (Gomez F. , 2015)

2.10. Herramientas para la Implementación de Aplicaciones Web

Para desarrollar aplicaciones Web se debe hacer uso de sistemas que puedan ser accedidos mediante cualquier navegador Web. El estándar dominante para la creación de una aplicación Web ha sido HTML, el Lenguaje de Marcado de Hipertexto, que posibilita al desarrollador proporcionar una serie de etiquetas que describen objetos como textos, videos, gráficos.

2.10.1. Servidor Http Apache

Para entender mejor lo que es Apache, primeramente, definiremos lo que es un servidor web. **(Fumas Cases, 2015).**

2.10.1.1. Servidor Web

La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc.). Estos servidores web utilizan el protocolo http. Los servidores web están alojados en un ordenador que cuenta con conexión a Internet.

El **web server**, se encuentra a la espera de que algún navegador le haga alguna petición, como, por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.

El **servidor Apache HTTP**, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la **Apache Software Foundation**.

Características:

- **Instalación/Configuración:** Software de código abierto.
- **Coste.** El servidor web Apache es completamente gratuito.
- **Funcional y Soporte:** Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente

con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.

- **Multi-plataforma:** Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- **Rendimiento:** Capacidad de manejar más de un millón de visitas/día.
- **Soporte de seguridad:** SSL y TLS.

2.10.2. MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Está desarrollado por Michael (Monty) Widenius (fundador De MySQL) y la comunidad de desarrolladores de software libre. Es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos (Sólida, 2017)

Características:

Más motores de almacenamiento

Mejoras de Velocidad

- Existen algunas mejoras al código DBUG para hacer su ejecución más rápida cuando se compila, pero no se usa.
- La tabla de chequeo de redundancia es más rápida.
- Replicación rápida y segura.
- La suite de pruebas está extendida y ahora corre mucho más rápido que antes aun si se prueban más cosas.
- MariaDB puede manejar hasta 32 segmentos clave por clave (sobre los 16 originales).
- Eliminación de Tablas.
- Extensiones de prueba mysqltest.

Mejores Pruebas.

- Más pruebas en la suite de pruebas.
- Errores en pruebas corregidos.

- Pruebas construidas con diferentes opciones de configuración para obtener mejores pruebas.

2.11. Modelo de Estimación de Costos

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismo, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto de grado eficientemente. La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimiento y tiempo incidiendo en la gestión empresarial.

2.11.1. Modelo COCOMO II

Entre los distintos métodos de estimación de costes, el modelo **COCOMO II** (***CO**nstructive **CO**st **MO**del*) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto software.

Por un lado, **COCOMO** define tres modos de desarrollo o tipos de proyectos:

- a. **Orgánico:** Proyectos relativamente sencillos, menores de 50 KLDC (Kilos de líneas de código), en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- b. **Semi-acoplado:** Proyectos intermedios en complejidad y tamaño (menores de 300 KLDC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- c. **Empotrado:** Proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Tabla Nº 2. 2: Coeficientes

PROYECTO SOFTWARE	a	E	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi-acoplado	3,0	1,12	2,5	0,35
Empotrado	2,8	1,20	2,5	0,32

Fuente: (Pressman, 2005)

Por otro lado, existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC (Líneas de Código).
- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Las fórmulas utilizadas serán las siguientes:

- ✓ **E = Esfuerzo = a KLDC^e * FAE** (persona x mes)
- ✓ **T = Tiempo de duración del desarrollo = c Esfuerzo^d** (meses)
- ✓ **P= Personal = E/T** (personas)

Dónde:

KLDC= (PF * Líneas de código por cada PF)/1000

FAE, se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste.

2.11.1.1. Métricas orientadas a la función: Punto Función (PF)

La medida de punto de función se propuso en 1979 y trata de medir la funcionalidad o utilidad del software.

✓ **Cálculo del punto de función (PF)**

Para calcular el PF se estima cada uno de los valores necesarios que se muestran en la siguiente tabla.

Tabla Nº 2. 3: Valores del dominio de Información.

Parámetro	Cuenta	Factor de ponderación			Subtotal
		Simple	Medio	Complejo	
Número de entradas de usuario		3	4	6	
Número de salidas de usuario		4	5	7	
Número de peticiones de usuario		3	4	6	
Número de archivos		7	10	15	
Número de interfaces externas		5	7	10	
Total (T)					

Fuente: (Pressman, 2005)

Donde la columna **Cuenta** debe ser llenado con:

1. **Entradas de usuario.** Son entradas que proporcionan diferentes datos a la aplicación. No confundirlos con las peticiones de usuario. (cantidad de datos que introduces por teclado).
2. **Salidas de usuario.** Son reportes, pantallas o mensajes de error que proporcionan información. Los elementos de un reporte, no se cuentan de forma separada. (los resultados que te muestra tu sistema).
3. **Peticiones de usuario.** Es una entrada interactiva que produce la generación de alguna respuesta del software en forma de salida interactiva. (como impresiones, almacenamiento).
4. **Archivos.** Son los archivos que pueden ser parte de una base de datos o independientes. (número de tablas que tienes).

5. **Interfaces externas.** Son los archivos que se usan para transmitir información a otro sistema. (número de pantallas de interfaz externa que tiene tu sistema).

Responder a cada una de las siguientes catorce preguntas y asignarles un valor entre **0** y **5**, donde **0** es no influencia, **1** es incidental, **2** es moderado, **3** es medio, **4** es significativo y **5** es esencial.

2.11.1.2. Factores

Preguntas de valor de frecuencia. Se calculará la cantidad de líneas de código y la métrica respecto al punto de función.

Tabla Nº 2. 4: Preguntas de valor de frecuencia

FACTOR	VALOR
Copia de seguridad y recuperación	
Comunicaciones de datos	
Proceso distribuido	
Rendimiento crítico	
Entorno operativo existente	
Entrada de datos en línea	
Transacciones de entrada en múltiples pantallas	
Archivos maestros actualizados en línea	
Complejidad de valores del dominio de información	
Complejidad del procesamiento interno	
Código diseñado para ser rehusado	
Conversión/instalación en diseño	
Instalaciones múltiples	
Aplicación diseñada para el cambio	
Total (F)	$\sum(F_i)$

Fuente: (Pressman, 2005)

Dónde:

F= Frecuencia de critica

✓ El punto de función FP se calcula con la siguiente ecuación:

$$PF = T * (0.65 + 0.01 * F)$$

Calculo de KLDC (Kilo de líneas de código) y FAE (Factor de ajuste del esfuerzo)

Tabla Nº 2. 5: Funcionalidad de los lenguajes de Programación

LENGUAJE	LDC/PF
Ensamblador	320
C	150
C++	64
PHP	29

Fuente: (Pressman, 2005)

- ✓ El número de Kilos de Líneas de código (KLDC) se calcula con la siguiente ecuación:

$$\text{KLDC} = (\text{PF} * \text{Líneas de código por cada PF}) / 1000$$

- ✓ Calculo de la variable **FAE**, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste.

Tabla Nº 2. 6: Conductores de Coste

CONDUCTORES DE COSTE	VALORACIÓN					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Fiabilidad requerida del Software	0,75	0,88	1.00	1,15	1,40	-
Tamaño de la 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
Restricciones del tiempo de ejecución.	-	-	1.00	1,11	1,30	1,66
Restricciones del almacenamiento principal.	-	-	1.00	1,06	1,21	1,56
Volatilidad de la máquina Virtual.	-	0,87	1.00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1.00	1,07	1,15	-
Capacidad del analista.	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 los Programadores.	1,42	1,17	1.00	0,86	0,70	-
Experiencia en S.O. utilizado.	1,21	1,10	1.00	0,90	-	-
Experiencia en el lenguaje de programación.	1,14	1,07	1.00	0,95	-	-
Prácticas de programación moderna.	1,24	1,10	1.00	0,91	0,82	-
Utilización de herramientas Software.	1,24	1,10	1.00	0,91	0,83	-
Limitaciones de planificación del proyecto.	1,23	1,08	1.00	1,04	1,10	-

Fuente: (Pressman, 2005)

Justificación de los valores de la tabla anterior:

a) **Atributos de software**

- **Fiabilidad requerida del software:** Si se produce un fallo por el pago de un pedido, o fallo en alguna reserva, etc. puede ocasionar grandes pérdidas a la empresa (Valoración Alta).
- **Tamaño de la base de datos:** La base de datos del producto software será de tipo estándar (Valoración Nominal).
- **Complejidad del producto:** La aplicación no va a realizar cálculos complejos (Valoración Baja).

b) **Atributos de hardware**

- **Restricciones del tiempo de ejecución:** En los requerimientos se exige alto rendimiento (Valoración Alta).
- **Restricciones del almacenamiento principal:** No hay restricciones al respecto (Valoración Nominal).
- **Volatilidad de la máquina virtual:** Se usarán sistemas de la “Familia Windows” (Valoración Nominal).
- **Tiempo de respuesta del ordenador:** Deberá ser interactivo con el usuario (Valoración Alta).

c) **Atributos del personal**

- **Capacidad del analista:** Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta).
- **Experiencia en la aplicación:** Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).
- **Capacidad de los programadores:** Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).
- **Experiencia en S.O. utilizado:** Con Windows 2000 Professional la experiencia es a nivel usuario (Valoración Nominal).

- **Experiencia en el lenguaje de programación:** Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).

d) **Atributos del proyecto**

- **Prácticas de programación modernas:** Se usarán prácticas de programación mayormente convencional (Valoración Nominal).
 - **Utilización de herramientas software:** Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).
 - **Limitaciones de planificación del proyecto:** Existen pocos límites de planificación. (Valoración Baja).
- ✓ Cálculo del esfuerzo del desarrollo, se tiene que:
 $E = a \text{ KLDC}^e * \text{FAE personas /mes.}$
- ✓ Cálculo del tiempo de desarrollo se tiene:
 $T = c \text{ Esfuerzo}^d \text{ meses}$
- ✓ Cálculo de la Productividad del proyecto se tiene:
 $PR = \text{LDC/Esfuerzo LDC/personas mes.}$
- ✓ Cálculo del número de Personal Promedio se tiene:
 $P = E/T \text{ personas}$

CAPITULO III: MARCO APLICATIVO

3. Marco Aplicativo

Al establecer un sistema de información se debe tener la certeza que cumpla dos objetivos; que sea un sistema correcto en el análisis y diseño y que este correctamente implementado y funcionando, siendo útil y mejorando así la toma de decisiones de la Institución. La función del análisis y diseño es desarrollar un producto que pueda cumplir con los requerimientos establecidos por el usuario y no presente fallas posteriores en su uso.

En el presente capítulo se describe y desarrolla el análisis y diseño del “**Sistema de Información para la Gestión Académica en un entorno Web para la unidad de Kardex Académico de la carrera Contaduría Pública - UPEA**”, siguiendo las etapas de la metodología UWE (Ingeniería Web basada en UML), normas y técnicas mencionadas en el marco teórico del capítulo II.

3.1. Modelo del Negocio

3.1.1. Análisis actual de la Carrera Contaduría Pública-Upea

Con el modelado del negocio, se logrará entender mejor el funcionamiento de los respectivos procesos que se llevan a cabo en el área académica en la unidad de kardex de la carrera de Contaduría Pública.

3.1.2. Obtención de requerimientos

La tarea de ingeniería de requerimiento es fundamental para que un sistema tenga éxito, en este sentido para el presente proyecto se realizaron las actividades correspondientes, como se puede apreciar en la siguiente tabla.

Tabla Nº 3. 1: Tareas Realizadas Para Obtención De Requisitos

TAREAS REALIZADAS	CARACTERISTICAS
Entrevistas	Se realizaron las entrevistas con los siguientes personajes: <ul style="list-style-type: none">✓ Director de carrera✓ Encargado de Kardex✓ Encargado de caja de recaudación

	✓ Docentes ✓ Estudiantes
Observaciones	En la carrera de Contaduría Pública se pudo observar y tomar nota de los procesos que se siguen en el área académica como: gestión de pensum, planificación académica, inscripción y seguimiento no son las adecuadas ya que se lo realiza de forma semi manual y desorganizada.
Documentación	Se ha podido recabar, con toda la documentación necesaria para el proceso de desarrollo del sistema.

Fuente: Elaboración Propia

3.1.3. Procesos académicos generales

A continuación, se detallan los principales procesos detectados:

- **Gestión de pensum**, no se establece un orden gradual y armónico de asignaturas con sus respectivas características (código, asignatura, nivel descripción, régimen, menciones, prerrequisitos, tiempo y número de horas de teoría y de práctica de cada asignatura). Lo cual dificulta la emisión de documentos como historiales y record académicos por el manejo simultáneo de varios planes de estudio.
- **Planificación académica**, existen muchos factores que inciden en esto: asignar docentes por sección y horario, número de estudiantes inscritos, asignación de aulas, asignaturas, paralelos, y otros. Este proceso se realiza al inicio de cada gestión en la que se distribuye aulas y docentes de acuerdo con el número de estudiantes y bloques horarios.
- **Inscripción de estudiantes**, el proceso se realiza manualmente mediante el llenado de formularios de inscripción por los estudiantes, posteriormente kardex se encarga de ordenar los mismos por asignaturas.
- **Registro de notas**, este proceso es realizado manualmente en el transcurso de cada semestre y a finales de la misma son copiadas en hojas de texto de Microsoft Excel, denominadas Acta de Notas, los cuales son sellados y archivados en las actas de cada gestión.

- **Seguimiento académico de estudiantes**, se realiza de manera manual, la información se guarda en libro de actas y medios físicos sin respaldo alguno, se produce una demora considerable en la obtención de la información como: datos personales, asignaturas inscritas, historial de notas, calificaciones de las diferentes asignaturas de las gestiones correspondientes.

3.2. Modelo del Sistema

La **lista de requerimientos** del sistema se lo debe realizar a partir de los problemas identificados y de los objetivos planteados como solución a éstos.

3.2.1. Requerimientos del Sistema

3.2.1.1. Requerimientos Funcionales

Tabla Nº 3. 2: Requerimientos Funcionales

Ref. #	FUNCION	CATEGORIA
R.1	Administración y niveles de usuario.	Evidente
R.2	Determinar el estado de cada usuario activo.	Evidente
R.3	Control de acceso al sistema.	Oculto
R.4	Registro y actualización de datos de estudiantes, docentes.	Evidente
R.5	Administración de gestión académica (pensum, paralelos, asignaturas, prerrequisitos, carrera, sedes académicas, gestión).	Evidente
R.6	Asignación de materias y lista de estudiantes a docentes.	Evidente
R.7	Verificación interna de notas para controlar materias a cursar y materias de arrastre.	Oculto
R.8	Verificación interna de Prerrequisitos.	Oculto
R.9	Control de cupos por paralelo	Oculto
R.10	Inscripción de estudiantes.	Evidente
R.11	Emitir formulario de Inscripción.	Evidente
R.12	Registro de evaluación docente.	Evidente
R.13	Emitir formulario de evaluación de docente.	Evidente
R.14	Despliegue de listas ordenadas de estudiantes inscritos por paralelo.	Evidente
R.15	Registro de documentación presentada (formulario de inscripción)	Evidente

R.16	Registro de cobro por gestión.	Evidente
R.17	Registro de Notas de los estudiantes.	Evidente
R.18	Emisión de récord académico.	Evidente
R.19	Emisión de certificado de deudas pendientes.	Evidente
R.20	Emisión de certificado estudiante regular.	Evidente
R.21	Emisión de recibo de aporte.	Evidente
R.22	Generar estadísticas por gestión, sede y género.	Evidente
R.23	Generar estadística de estudiantes sobresalientes.	Evidente
R.24	Generar estadísticas de inscripción por gestión.	Evidente

Fuente: Elaboración Propia

3.2.1.2. Requerimientos NO Funcionales

Tabla Nº 3. 3: Requerimientos Funcionales

Ref. #	FUNCION
RN.1	El sistema debe visualizarse y funcionar correctamente en cualquier navegador, Internet Explorer, Mozilla Firefox, Chrome.
RN.2	El sistema contara con restricciones de acceso de forma que los usuarios tendrán acceso solamente a la información que ellos deban conocer o manejar.
RN.3	El sistema brindara una interfaz amigable y comprensible para los usuarios, permitiendo que la capacitación en el manejo del mismo se haga en el menor tiempo.
RN.4	El sistema debe evitar la redundancia de información.
RN.5	El sistema debe validar la información ingresada en la base de datos.
RN.6	Se almacenarán respaldos periódicos de la información de forma que esta pueda ser recuperada en caso de fallas del sistema o pérdidas de información por accidentes no previstos.
RN.7	El sistema no debe tardar más de diez segundos en mostrar los resultados de una búsqueda.
RN.8	El sistema debe mostrar información comprensible para los usuarios.

Fuente: Elaboración Propia

3.2.2. Definición de actores (del sistema)

Las identificaciones de los actores, nos permite conocer a los personajes involucrados en los procesos que contemplan el área académica de la Carrera Contaduría Pública.

A objeto de modelar los casos de uso, en la siguiente tabla se presenta la lista de los actores y la descripción de sus actividades.

Tabla Nº 3. 4: Definición de Actores

ACTOR	DESCRIPCION
Administrador	Encargado del manejo de la información del área académica quien necesita habilitar y administrar: <ul style="list-style-type: none"> ✓ Usuarios. ✓ Gestión para inscripción. ✓ Cobros para recaudación. ✓ Prerrequisitos. ✓ Asignatura. ✓ Pensum. ✓ Sede. ✓ Parámetros del sistema.
Encargado de Kardex	Responsable del manejo de la información académica quien necesita: <ul style="list-style-type: none"> ✓ Realizar la planificación académica por gestión. ✓ Asignación de materias a docente. ✓ Información detallada de los estudiantes. ✓ Inscripción de estudiantes. ✓ Verificación automática de materias a cursar y materias de arrastre de un estudiante. ✓ Verificación de nómina de Inscritos. ✓ Verificación de evaluación docente. ✓ Registro de presentación de documento (formulario de inscripción). ✓ Registro de Notas de estudiantes. ✓ Emitir formulario de inscripción. ✓ Emitir listas oficiales de estudiantes inscritos por paralelo y materia. ✓ Emitir récord académico ✓ Emitir certificado de deudas pendientes. ✓ Emitir certificado de estudiante regular. ✓ Realizar consultas y emitir reportes generales y específicos de estudiantes por gestión, sede, género y paralelos ✓ Ver listas de estudiantes inscritos por gestión y paralelo. ✓ Emitir reporte de datos personales de los estudiantes (inscritos por gestión, género, sede, mejores estudiantes)

Encargado Caja de recaudación	Responsable que necesita: ✓ Recaudar el aporte por inscripción. ✓ Generar el recibo del aporte. ✓ Generar el formulario de inscripción final. ✓ Generar lista del total recaudado.
Estudiante	Usuario Indirecto quien necesita: ✓ Conocer sus datos de Registro Personal. ✓ Realizar su pre-inscripción. ✓ Obtener la Formulario de inscripción. ✓ Conocer Materias Inscritas y paralelos. ✓ Verificar Historial de Notas. ✓ Obtener Récord Académico. ✓ Registrar la evaluación de docente.

Fuente: Elaboración Propia

3.3. Aplicación del Modelo UWE

3.3.1. Modelo de Casos de Uso

3.1.1.1. Análisis de requerimientos

En este punto se plasma el análisis de requerimientos del sistema mediante el diseño de diagrama de caso de uso comercial el cual describe el comportamiento del Carrera Contaduría Pública. El diagrama de caso de uso permite que describa el comportamiento del sistema frente a las acciones de los diferentes actores, así como las funcionalidades del sistema.

3.1.1.1.1. Diagrama de casos de uso comercial

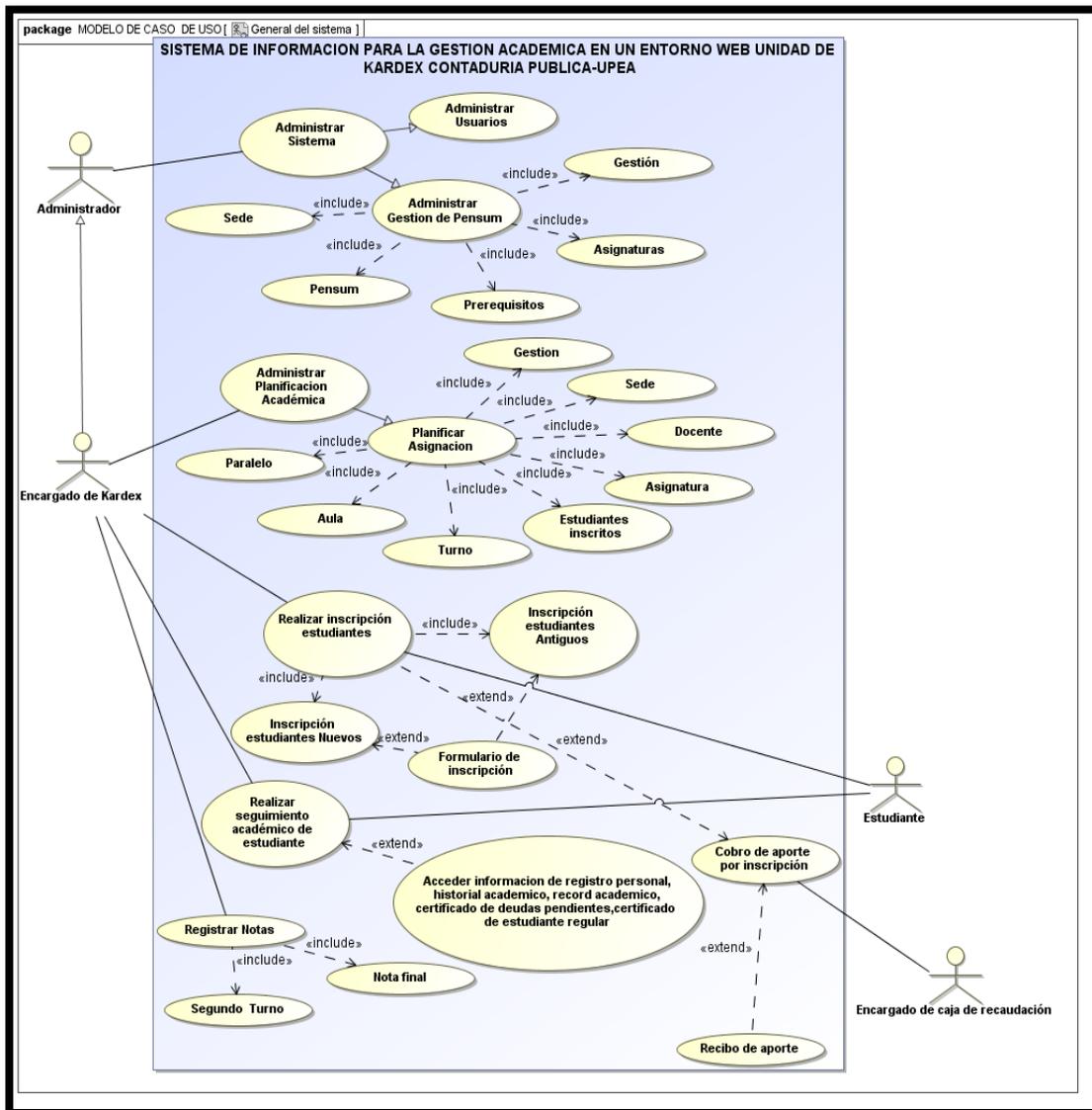
A continuación, se hace el modelamiento donde se puede apreciar cómo interactúan los actores sobre los casos de uso. Presentación previa de los diagramas de caso de uso, se describen a continuación:

- **Modelo de caso de uso: General del Sistema**
- **Diagrama de caso de uso: Gestión de Pensum**
- **Diagrama de caso de uso: Planificación Académica**
- **Diagrama de caso de uso: Inscripción**
- **Diagrama de caso de uso: Registro de Notas**

- Diagrama de caso de uso: Seguimiento Académico de Estudiante

3.1.1.1.2. Modelo de Caso de Uso General del Sistema

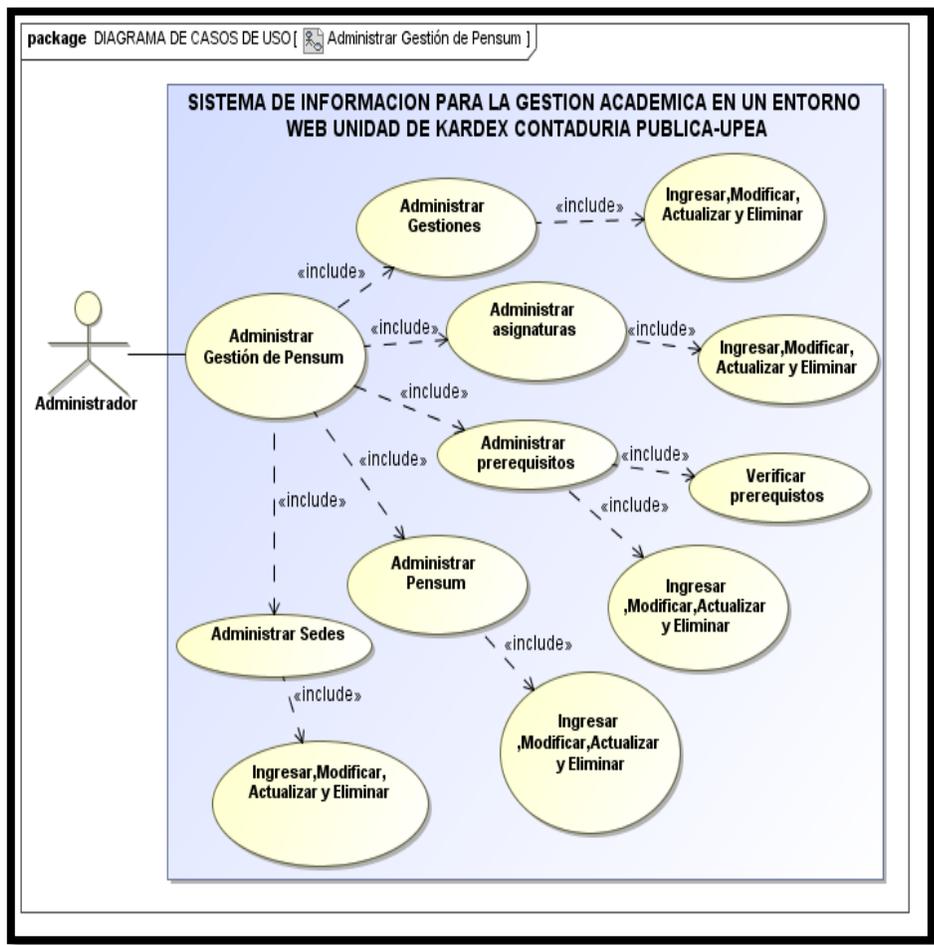
Figura Nº 3. 1: Modelo de Caso de Uso General del Sistema



Fuente: Elaboración Propia

- ✓ Diagrama de caso de uso: Gestión de Pensum

Figura Nº 3. 2: Diagrama de Caso de Uso: "GESTIÓN DE PENSUM"



Fuente: Elaboración Propia

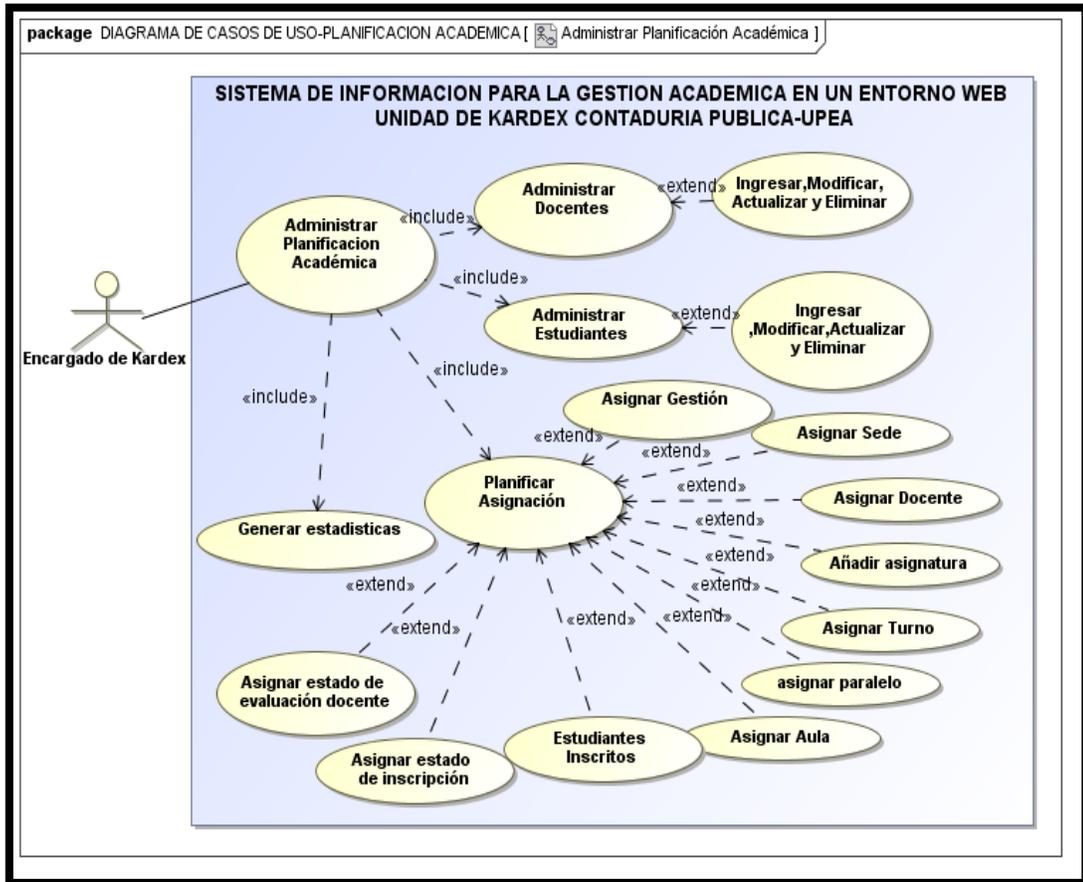
Tabla Nº 3. 5: Caso de Uso “GESTIÓN DE PENSUM”

CASO DE USO: GESTIÓN DE PENSUM	
ACTOR:	Administrador
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Administrador ingresa al sistema mediante su usuario, contraseña asignado, accede al módulo correspondiente y realiza la administración de Gestión de Pensum, estableciendo CRUD (crear, leer, actualizar y borrar) de gestion, pensum, asignaturas, prerequisite y sedes correspondientes a la carrera.

Fuente: Elaboración Propia

✓ Diagrama de caso de uso: Planificación Académica

Figura N° 3. 3: Diagrama de Caso de Uso: “PLANIFICACIÓN ACADÉMICA”



Fuente: Elaboración Propia

Tabla N° 3. 6: Caso de Uso “PLANIFICACIÓN ACADÉMICA”

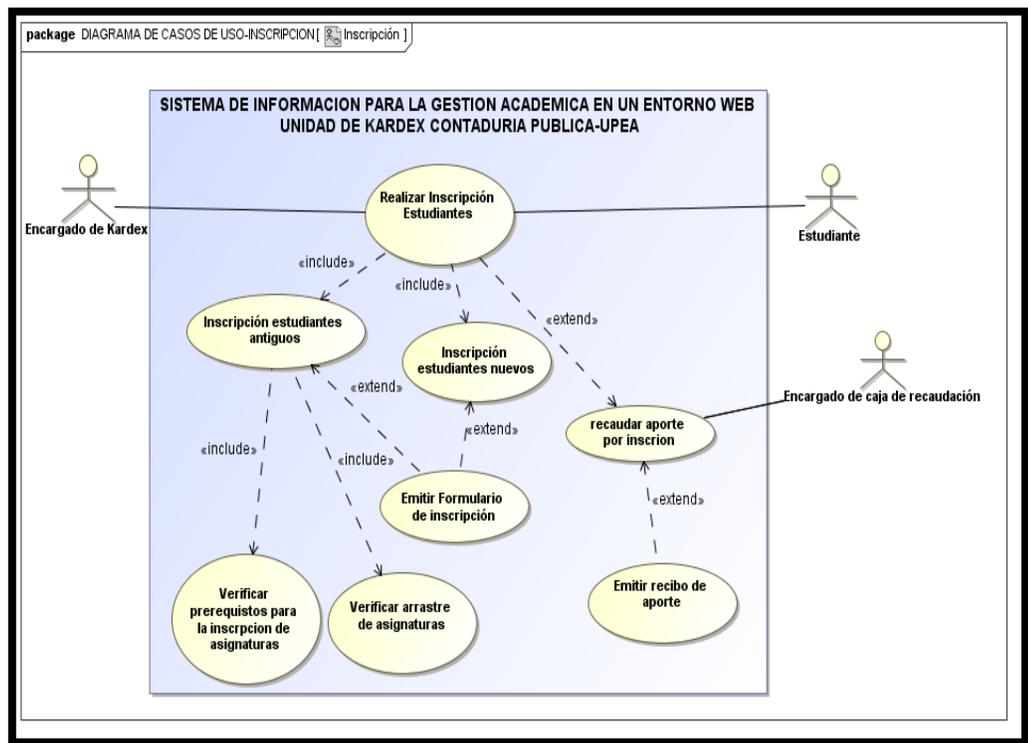
CASO DE USO: PLANIFICACIÓN ACADÉMICA	
ACTOR:	Encargado de Kardex
TIPO:	Primario Esencial

DESCRIPCIÓN:	Este caso de uso comienza cuando kardex ingresa al sistema mediante su usuario, contraseña asignado, accede al módulo correspondiente y efectúa la administración de Planificación Académica, estableciendo CRUD (crear, leer, actualizar y borrar) de gestión, sedes y de la misma manera procede a planificar la asignación registrando datos como docente, asignatura, turno, paralelo, estudiantes inscritos, estado de inscripción, estado de evaluación docente y otros, correspondientes a la gestión vigente.
---------------------	---

Fuente: Elaboración Propia

✓ Diagrama de caso de uso: Inscripción

Figura N° 3. 4: Diagrama de Caso de Uso: "INSCRIPCIÓN"



Fuente: Elaboración Propia

Tabla N° 3. 7: Caso de Uso "INSCRIPCIÓN"

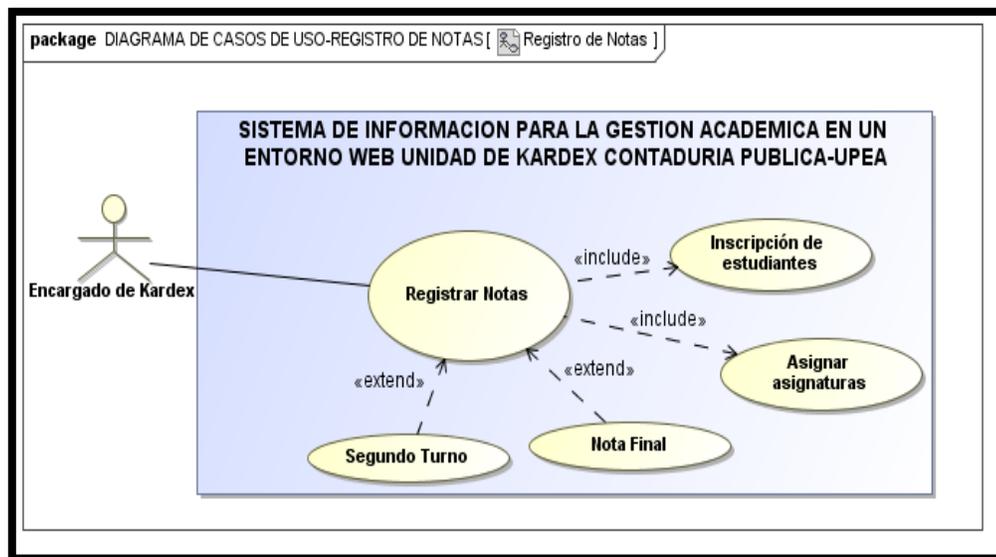
CASO DE USO: INCRIPCIÓN	
ACTOR:	Encargado de kardex, Estudiante, Encargado de caja de recaudación

TIPO:	Primario Esencial
DESCRIPCIÓN:	<p>Este caso de uso comienza cuando el Estudiante procede a realizar su inscripción como: Estudiante Nuevo, ingresa al Sitio Web de la Institución indaga y opta por la carrera de su interés e inmediatamente realiza su preinscripción haciendo el llenado de sus datos requeridos por el sistema, así mismo hace la impresión de la boleta de inscripción.</p> <p>Encargado de Kardex ingresa al módulo inscripción del sistema mediante su usuario y contraseña asignado y realiza la inscripción del estudiante si cumple con los requisitos exigidos por la carrera, así mismo procede a la impresión de la misma.</p> <p>Estudiante Antiguo, ingresa al sistema mediante su usuario y contraseña asignado y realiza su respectiva inscripción e impresión de la misma.</p>

Fuente: Elaboración Propia

✓ **Diagrama de caso de uso: Registro de Notas**

Figura Nº 3. 5: Diagrama de Caso de Uso: “REGISTRO DE NOTAS”



Fuente: Elaboración Propia

Tabla Nº 3. 8: Caso de Uso “REGISTRO DE NOTAS”

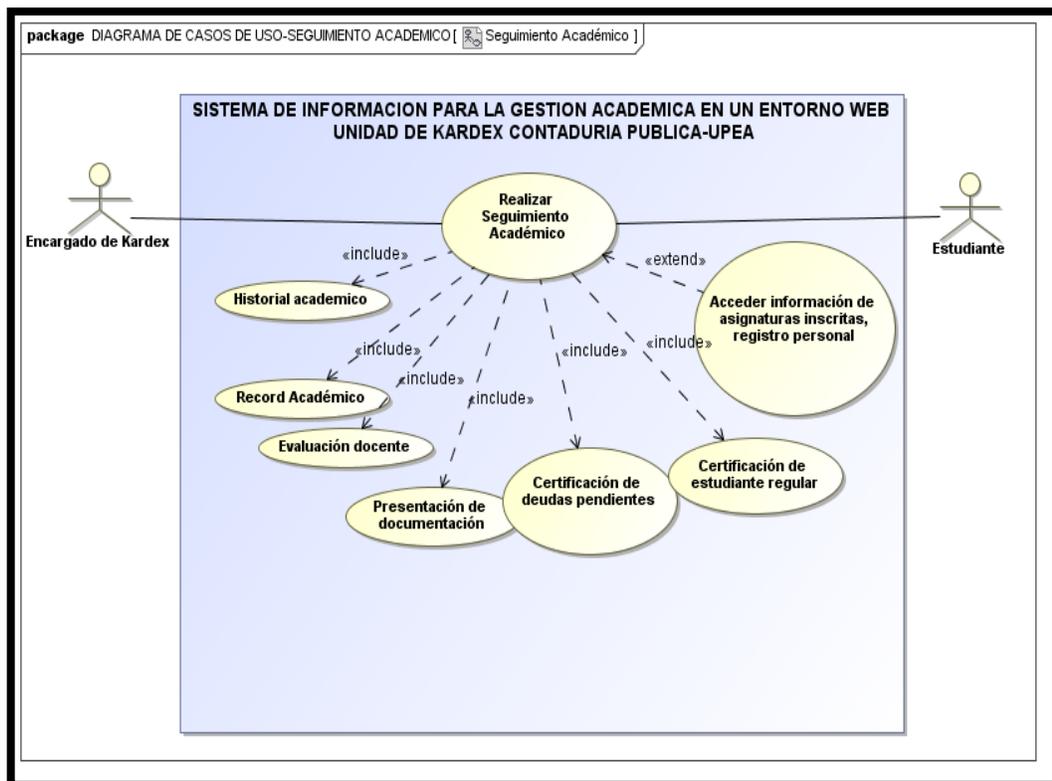
CASO DE USO: REGISTRO DE NOTAS	
ACTOR:	Encargado de Kardex

TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el encargado de kardex ingresa al sistema mediante su usuario, contraseña asignado, accede al módulo Registro de Notas, el mismo realiza el llenado de datos correspondientes .

Fuente: Elaboración Propia

✓ **Diagrama de caso de uso: Seguimiento Académico de Estudiante**

Figura Nº 3. 6: Diagrama de Caso de Uso: “SEGUIMIENTO ACADEMICO DE ESTUDIANTE”



Fuente: Elaboración Propia

Tabla N° 3. 9: Caso de Uso “SEGUIMIENTO ACADÉMICO DE ESTUDIANTE”

CASO DE USO: SEGUIMIENTO ACADÉMICO DE ESTUDIANTE	
ACTOR:	Estudiante, Encargado de kardex
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Encargado de kardex ingresa al sistema mediante su usuario, contraseña asignado y accede a la información de su registro personal, asignaturas inscritas, historial académico, evaluación docente. Así también la Estudiante accede a la misma información.

Fuente: Elaboración Propia

3.1.2. Modelo de contenido

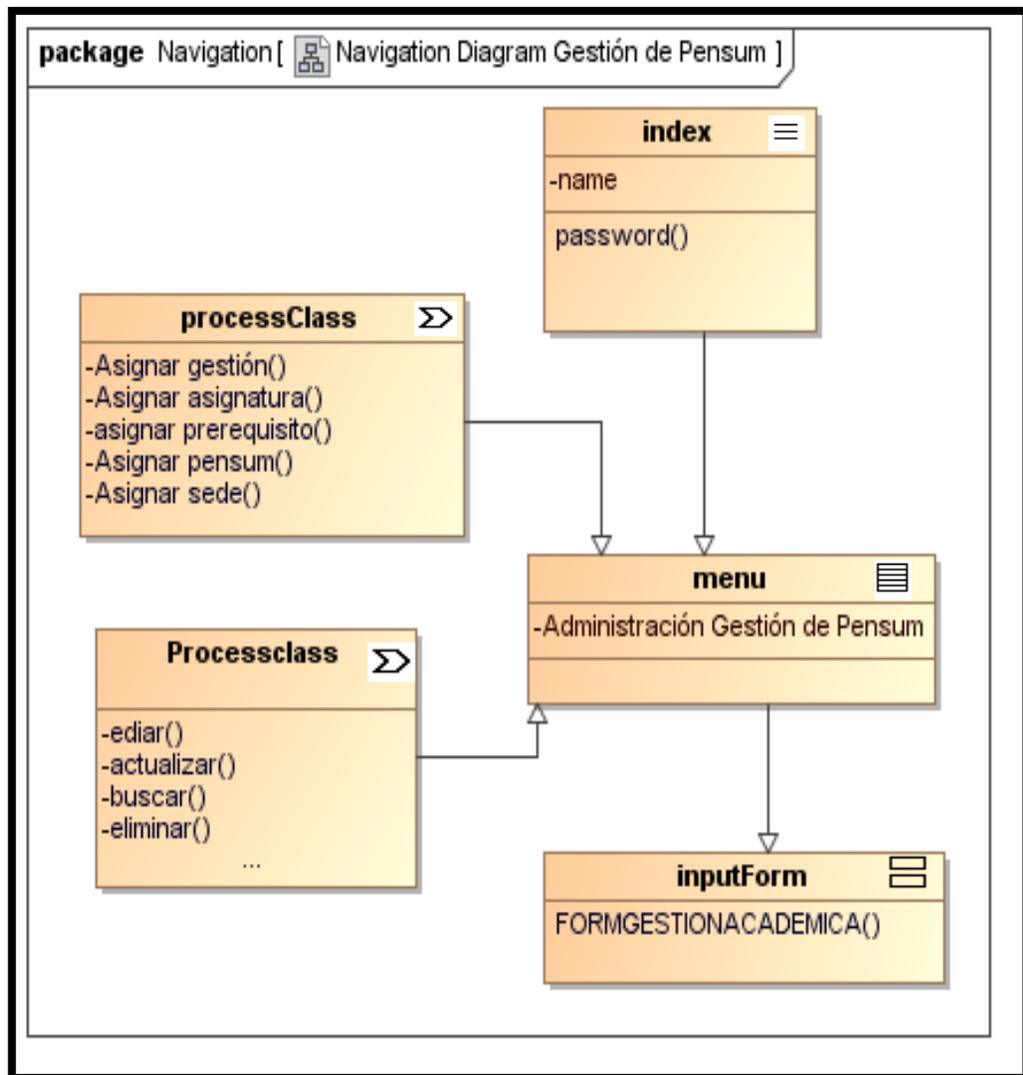
El modelo de contenido es un diagrama UML normal de clases, por ello se modela las clases necesarias para el sistema.

3.1.3. Modelo de Navegación

A continuación se realiza el modelamiento donde se puede apreciar cómo interactúan los usuarios en el modelo de navegación del sistema.

Modelo de Navegación: Gestión de Pensum

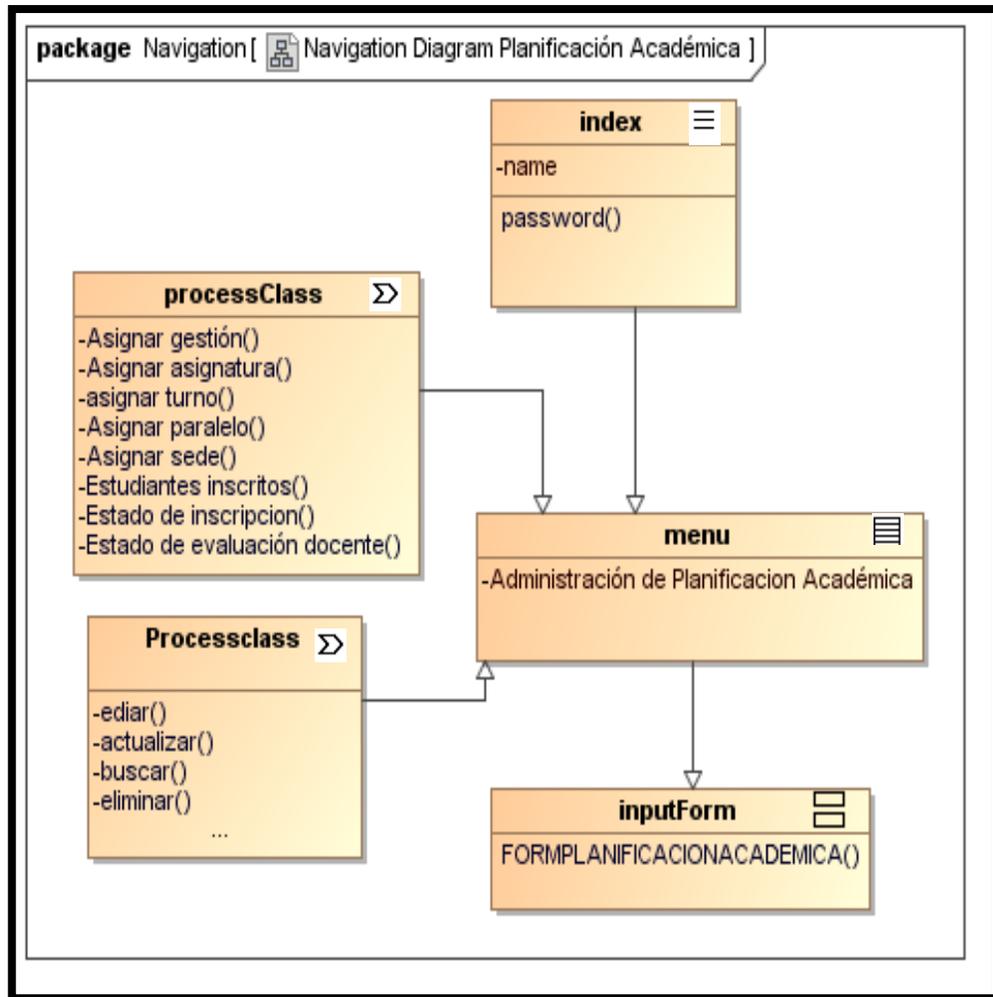
Figura Nº 3. 8: Modelo de Navegación Gestión de Pensum



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Planificación Académica**

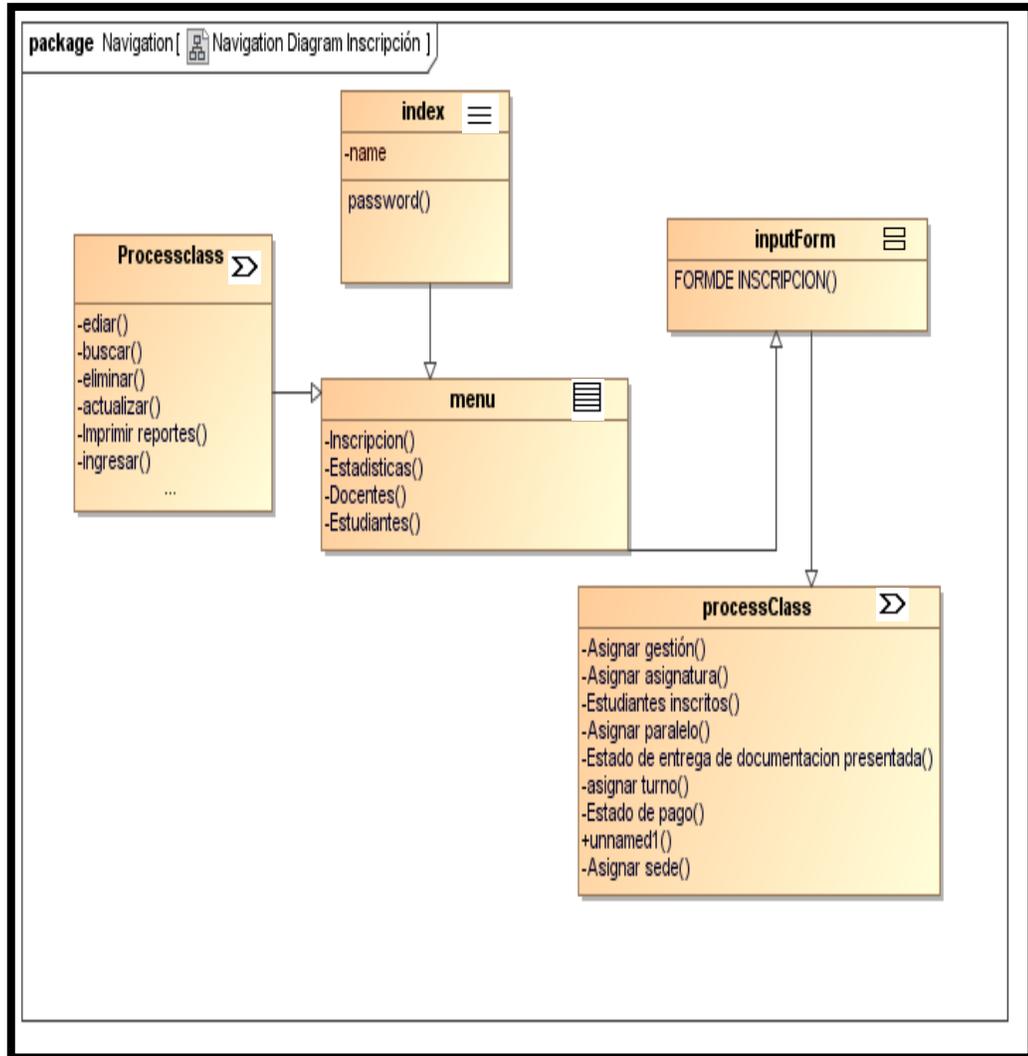
Figura Nº 3. 9: Modelo de Navegación Planificación Académica



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Inscripción**

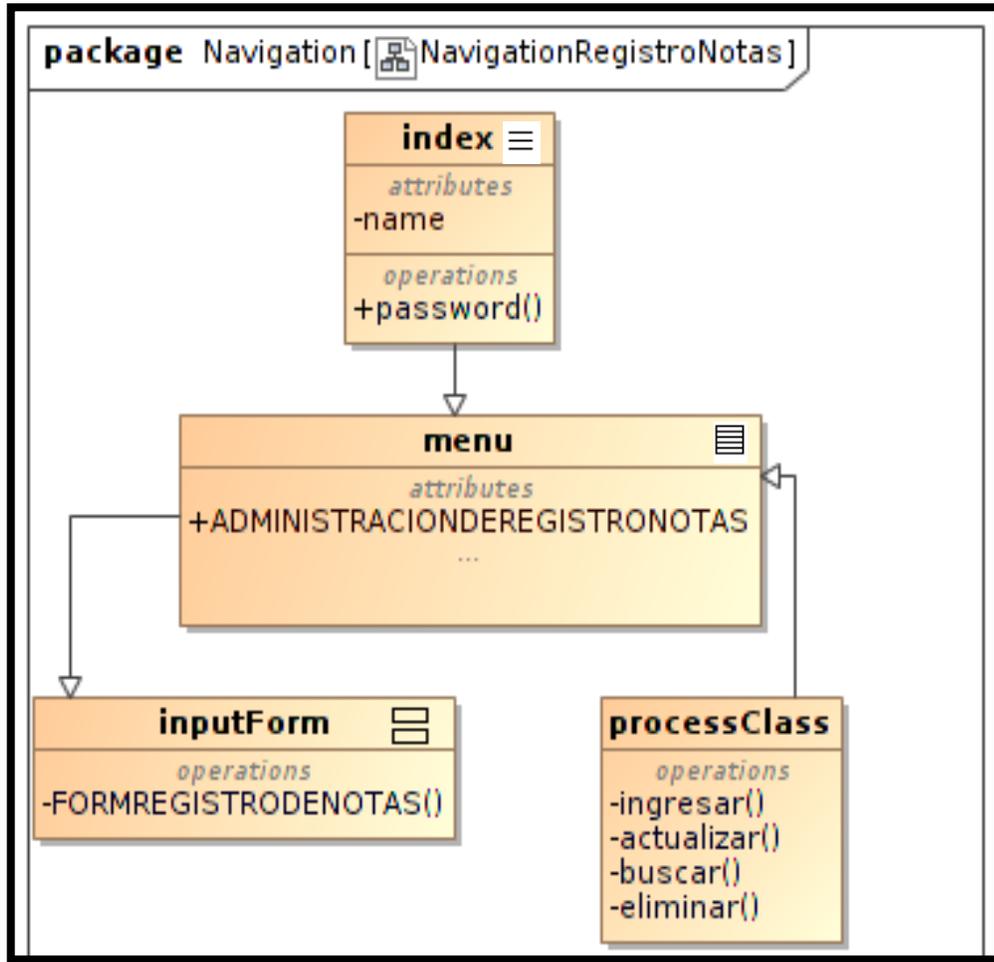
Figura N° 3. 10: Modelo de Navegación Inscripción



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Registro de Notas**

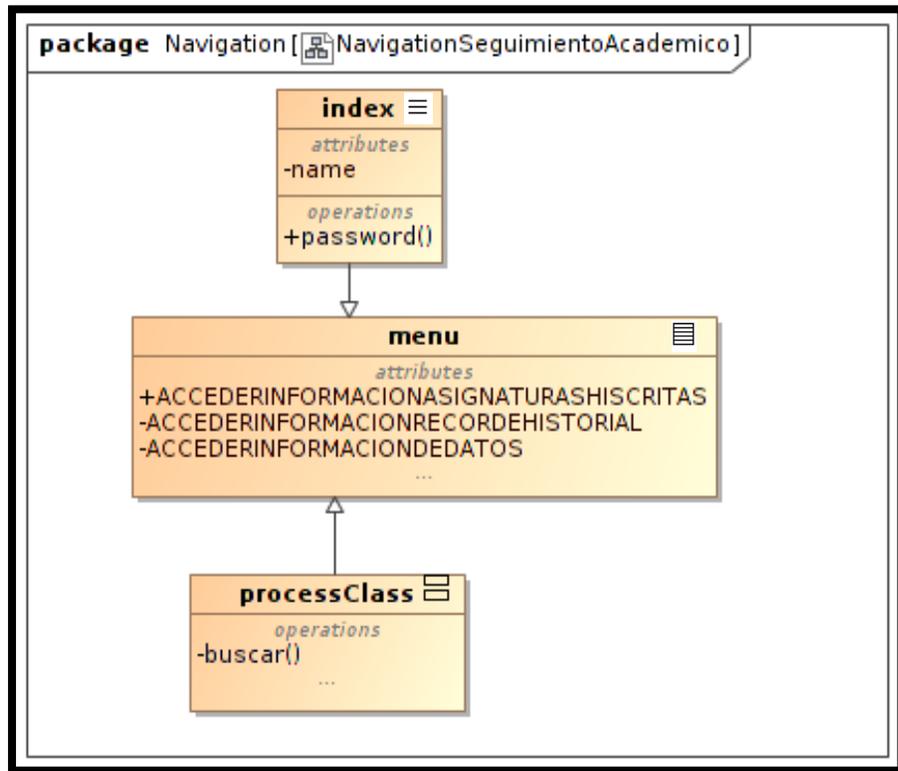
Figura Nº 3. 11: Modelo de Navegación Registro de Notas



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Seguimiento Académico de Estudiante**

Figura N° 3. 12: Modelo de Navegación Seguimiento Académico de Estudiante



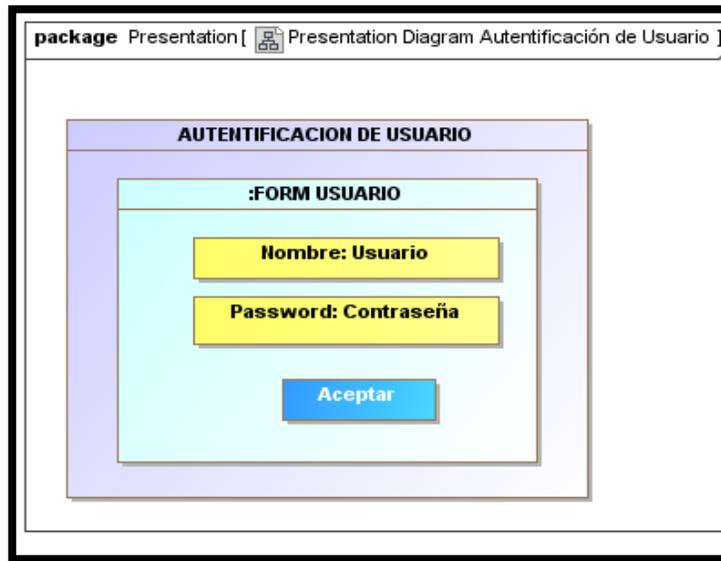
Fuente: Elaboración Propia

3.1.4. Modelo de Presentación

A continuación, se muestran los modelos de presentación para el sistema, según UWE propone para la construcción de las páginas en forma de bosquejos.

✓ **Modelo de Presentación: Autenticación**

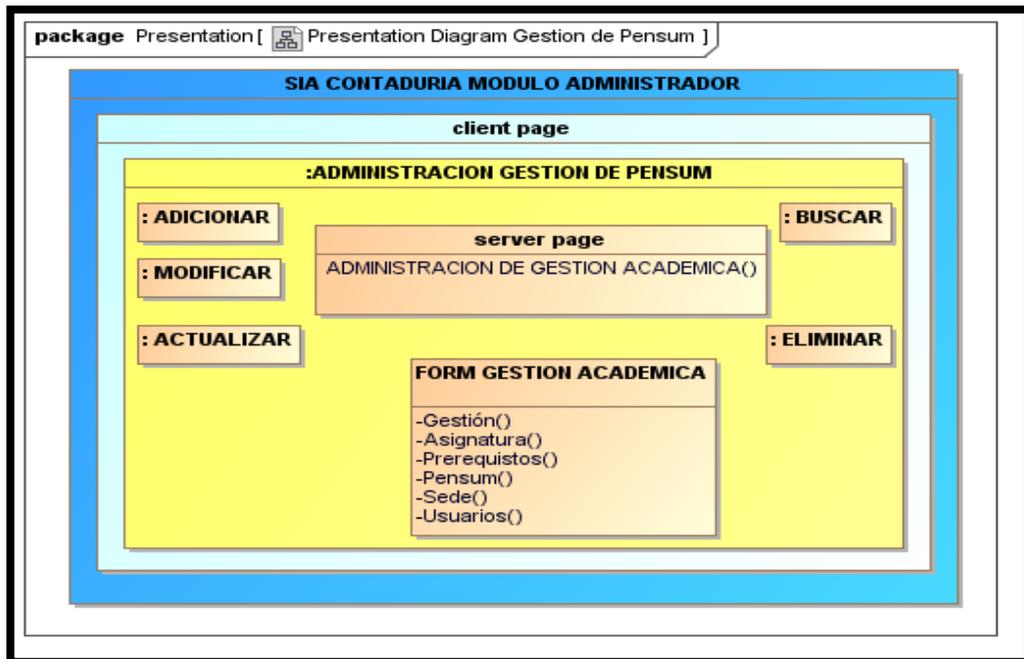
Figura Nº 3. 13: Modelo de Presentación Autenticación



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Gestión de Pensum**

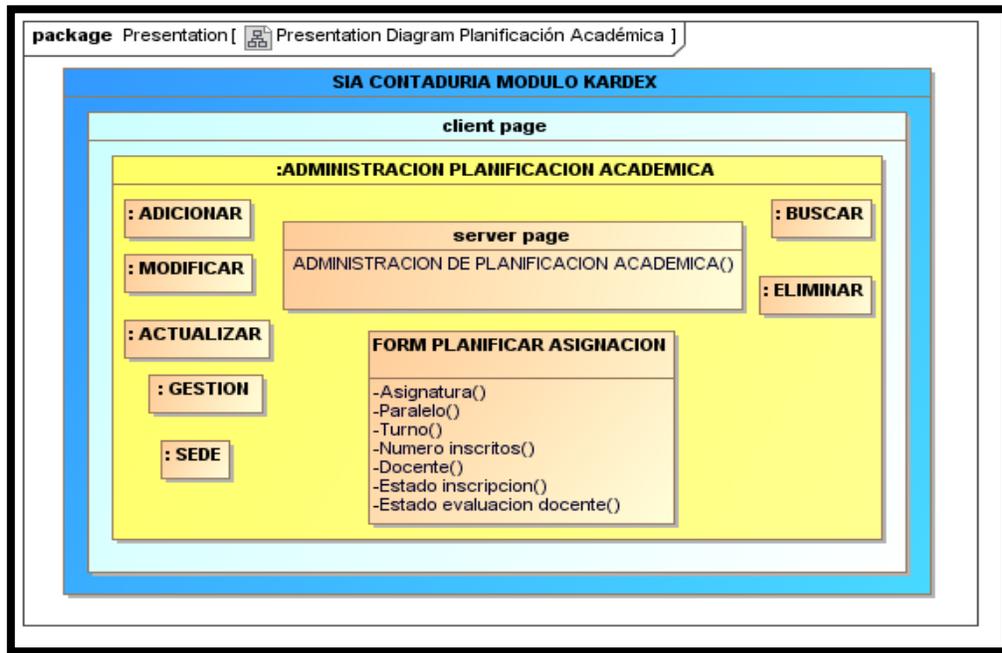
Figura Nº 3. 14: Modelo de Presentación Gestión de Pensum



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Planificación Académica**

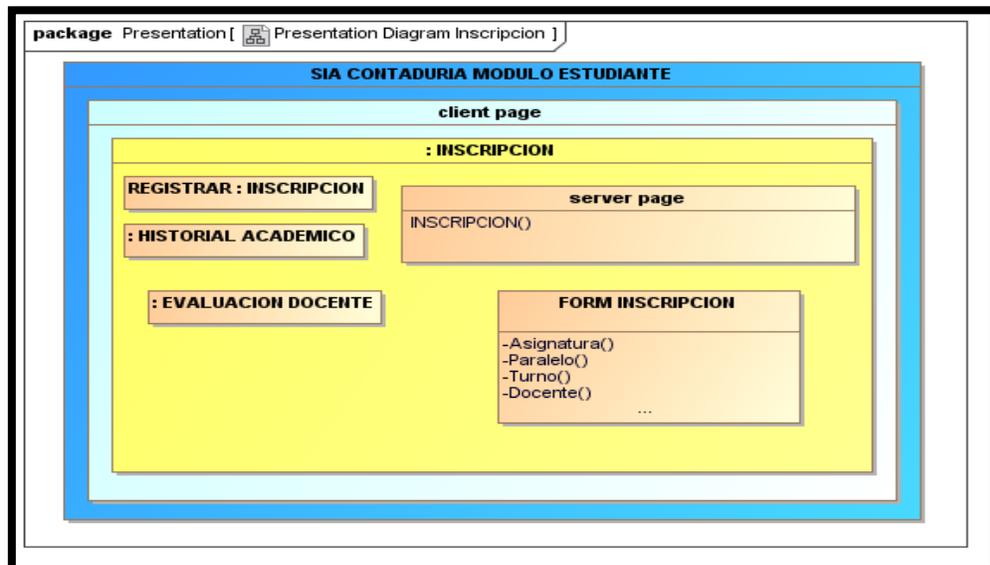
Figura Nº 3. 15: Modelo de Presentación Planificación Académica



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Inscripción**

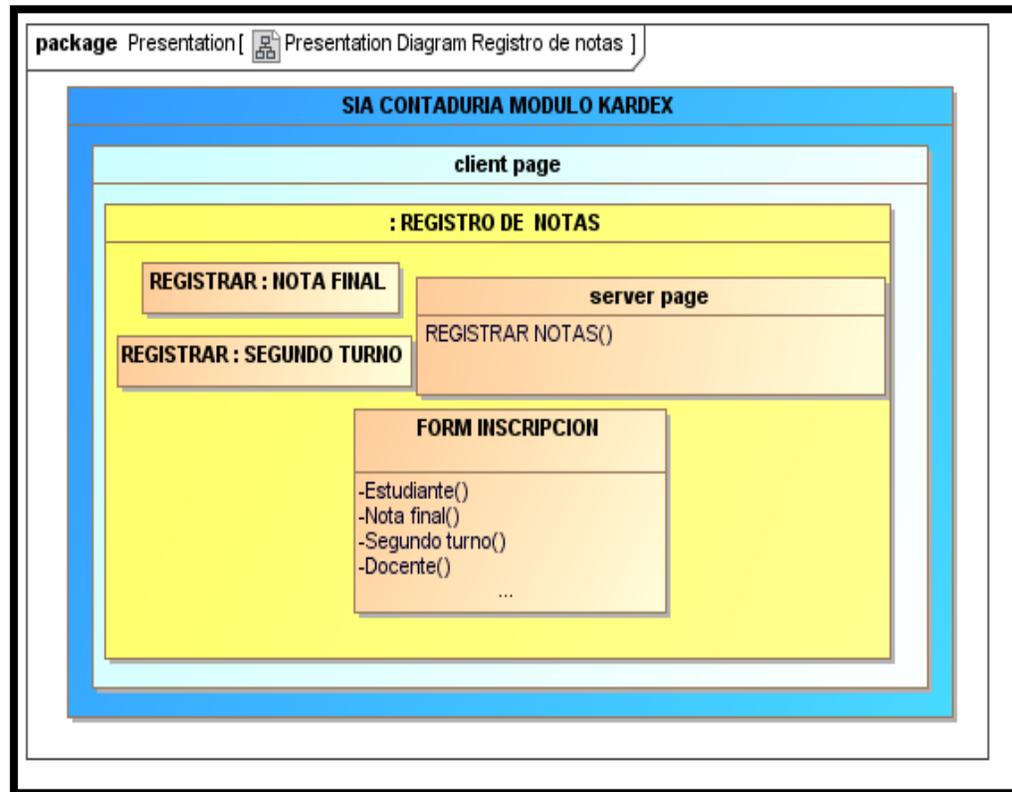
Figura Nº 3. 16: Modelo de Presentación Inscripción



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Registro de Notas**

Figura N° 3. 17: Modelo de Presentación Registro de Notas



Fuente: Elaboración Propia

3.1.5. Modelo de Implementación

En esta fase de implementación consiste en mostrar el desarrollo de la presentación de las interfaces del sistema y sus elementos construidos.

✓ **Interfaz de Inicio de Sesión: Autenticación**

Figura Nº 3. 18: Autenticación inicial con el sistema

CONTADURIA PUBLICA - UPEA

Inicia sesión con tu nombre de usuario y contraseña.

ingsis.jhonny@gmail.com

.....

Aceptar

Fuente: Elaboración Propia

✓ Interfaz de Gestión de Pensum

Figura Nº 3. 19: Interfaz de Gestión de Pensum (Administrador)

SIACONTADURIA ADMIN: JHONNY VARGAS RAMOS

Inicio

Añadir Asignatura Exportar Imprimir

25 Pagina 1 de 2 Mostrando 1 a 25 de 44 registros

Acciones	Pensum id	Sigla asignatura	Nombre	Estado
	CONTADURIA PÚBLICA	CON-101	CONTABILIDAD BASICA	A
	CONTADURIA PÚBLICA	MAT-101	ANÁLISIS MATEMÁTICO	A
	CONTADURIA PÚBLICA	DER-101	DERECHO COMERCIAL Y CORPORATIVO	A
	CONTADURIA PÚBLICA	ADM-101	ADMINISTRACIÓN GENERAL Y PÚBLICA	A
	CONTADURIA PÚBLICA	INV-101	TÉCNICAS DE APRENDIZAJE Y METODOLOGÍA DE LA INVESTIGACIÓN	A
	CONTADURIA PÚBLICA	MAT-102	MATEMATICAS FINANCIERAS	A
	CONTADURIA PÚBLICA	HIS-101	HISTORIA, CULTURA Y ETICA DE LA SOCIEDAD ORIGINARIA	A
	CONTADURIA PÚBLICA	ECO-101	ECONOMÍA Y MICROECONOMIA	A
	CONTADURIA PÚBLICA	CON-201	CONTABILIDAD INTERMEDIA	A
	CONTADURIA PÚBLICA	CON-202	CONTABILIDAD DE SOCIEDADES	A

Fuente: Elaboración Propia

✓ **Interfaz de Planificación Académica**

Figura Nº 3. 20: Interfaz de Planificación Académica (Encargado de kardex)

The screenshot displays the Academic Planning Interface. At the top, there are filters for 'Gestion: 2018 - REGULAR' and 'Sede: VILLA ESPERANZA'. Below these are buttons for '+ Añadir Asignatura', 'Exportar', 'Imprimir', and 'Buscar'. A pagination bar shows 'Mostrando 1 a 25 de 181 registros Pagina 1 de 8'. The main table lists courses with the following data:

Acciones	Asignatura	Paralelo	Turno	Estudiantes Inscritos	Estado Inscripción	Estado Evaluación Docente	Docente
Acciones	1 - ADM-101 - ADMINISTRACIÓN GENERAL Y PÚBLICA	1A	MAÑANA	102	activo	inactivo	JAVIER SELAEZ HELGU
Acciones	CONTABILIDAD BASICA	1A	MAÑANA	103	activo	inactivo	JAVIER SELAEZ HELGU
Acciones	1 - DER-101 - DERECHO COMERCIAL Y CORPORATIVO	1A	MAÑANA	89	activo	inactivo	JAVIER SELAEZ HELGU

Fuente: Elaboración Propia

✓ **Interfaz de Inscripción**

Figura Nº 3. 21: Interfaz de Inscripción

The screenshot shows the Enrollment Interface for 'GESTIÓN ACADÉMICA 2018'. The user information is: UNIVERSITARIO(A) ADUVIRI ALFREDO, CI 10022759 LP, and RU 12003769. Below this is a table titled 'MATERIAS INSCRITAS' with the following data:

No	AÑO	SIGLA	MATERIA	MODALIDAD	PARALELO	TURNO	SEDE	PREREQUISITO
1	4	CON-403	DISEÑO E IMPLEMENTACION DE SISTEMAS CONTABLES	ANUAL	4C	NOCHE	VILLA ESPERANZA	CON-301, CON-201, CON-202, CON-203, INF-201, MAT-201, CON-204, ECO-202, DER-202

Fuente: Elaboración Propia

✓ Interfaz de Registro de Notas

Figura Nº 3. 22: Interfaz de Registro de Notas (Encargado de Kardex)

Inicio
Inscripcion
PERSONA
Docente
Estudiante

Volver

ADMINISTRACIÓN GENERAL Y PÚBLICA
Paralelo: 1A - Turno: MAÑANA

Nivel: 1 Sigla: ADM-101
Gestion: 2018 Sede: VILLA ESPERANZA
Docente: JAVIER SELAEZ HELGUERO

Editar Estudiante

Estudiante* : 6078581 YUJRA LAURA, SHAYDA JERU..X

Nota final : 78

Segundo turno :

Actualizar cambios Actualizar y volver a la lista Cancelar

Fuente: Elaboración Propia

3.2. Métricas de Calidad de Software

La calidad del software nos permite medir, supervisar y mejorar el proceso para alcanzar la calidad del software, uno de ellos es el organismo de normalización ISO (*International Standards Organization*) que ha definido los requisitos de un sistema de gestión de calidad de software de carácter general que cubre el desarrollo de cualquier producto. Para el presente proyecto se toma en cuenta esta métrica estándar de calidad **ISO 9126**.

3.2.1. Factores de Calidad ISO 9126

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos claves de calidad para el software, pero no necesariamente son utilizados para medidas directas. En cualquier caso facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad de un sistema.

A continuación se desarrollara los seis principales atributos de calidad propuestos por esta métrica.

3.2.1.1. Funcionalidad

Para determinar la funcionalidad del sistema nos basaremos en la métrica puntos función (PF), se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. Los valores del dominio de la información se definen de la siguiente manera:

- ✓ **Número de entradas externas (EE).** Son todos los datos que el usuario introduce al sistema.

Tabla Nº 3. 10: Número de Entradas Externas

Nº	ENTRADAS EXTERNAS	CANTIDAD
1	Modulo Administrador	4
2	Módulo Gestión de Pensum	11
3	Módulo Planificación Académica	8
4	Módulo Inscripción	4
5	Módulo Registro de Notas	1
Total		28

Fuente: Elaboración Propia

- ✓ **Número de salidas externas (SE).** Son datos que proporciona la aplicación al usuario, estos pueden ser: informes, pantallas, mensajes de error, etc.

Tabla Nº 3. 11: Número de Salidas Externas

Nº	SALIDAS EXTERNAS	CANTIDAD
1	Modulo Administrador	9
2	Módulo Gestión de Pensum	6
3	Módulo Planificación Académica	7
4	Módulo Inscripción	5
5	Módulo Registro de Notas	3
Total		30

Fuente: Elaboración Propia

- ✓ **Número de consultas externas (CE).** se define como una entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software.

Tabla Nº 3. 12: Número de Consultas Externas

Nº	CONSULTAS EXTERNAS	CANTIDAD
1	Modulo Administrador	4

2	Módulo Gestión de Pensum	6
3	Módulo Planificación Académica	8
4	Módulo Inscripción	3
5	Módulo Registro de Notas	2
Total		23

Fuente: Elaboración Propia

- ✓ **Número de archivos lógicos internos (ALI).** Es agrupamiento lógico de datos dentro de los límites de las aplicaciones y que se mantiene entradas externas.

Tabla N° 3. 13: Número de Archivos Lógicos Internos

Nº	ARCHIVOS LÓGICOS INTERNOS	CANTIDAD
1	Modulo Administrador	4
2	Módulo Gestión de Plan de Pensum	7
3	Módulo Planificación Académica	5
4	Módulo Inscripción	2
Total		18

Fuente: Elaboración Propia

- ✓ **Número de archivos de interfaz externos (AIE).** Es un agrupamiento lógico de datos externos a la aplicación.

Tabla N° 3. 14: Número de Interfaces Externas

Nº	INTERFACES EXTERNAS	CANTIDAD
1	Internet	2
Total		2

Fuente: Elaboración Propia

Donde el conteo total es la suma de todas las entradas de PF y el Factor ponderado es Medio, como se puede apreciar en la tabla 3.16.

Tabla N° 3. 15: Conteo Total

PARÁMETRO	CONTEO	FACTOR	SUBTOTAL
Número de entradas externas (EE)	28	4	112
Número de salidas externas (SE)	30	5	150
Número de consultas externas (CE)	23	4	92

Número de archivos lógicos internos (ALI)	18	10	180
Número de archivos de interfaces externas (AIE)	2	7	14
TOTAL			548

Fuente: Elaboración Propia

Tomado el siguiente criterio para el Factor de Ajuste de Valor (FAV):

Tabla Nº 3. 16: Rangos para evaluar el PF

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

Fuente: Elaboración Propia

Valores de ajuste de complejidad según las respuestas a las siguientes preguntas que se muestra en la tabla Nº 3.18.

Tabla Nº 3. 17: Factor de Complejidad

Nº	FACTOR DE COMPLEJIDAD	VALOR
1	¿El sistema requiere respaldo y recuperación confiable?	5
2	¿Se requiere comunicación de datos?	4
3	¿Hay funciones distribuidas de proceso?	3
4	¿El desempeño es crítico?	4
5	¿Se ejecuta el sistema en un entorno operativo existente y frecuentemente utilizado?	5
6	¿Requiere el sistema entrada de datos interactivo?	4

7	¿Requiere la entrada de datos interactiva que las entradas lleven a cabo múltiples operaciones?	5
8	¿Se actualiza los archivos maestros en forma interactiva?	2
9	¿Las entradas, las salidas, los archivos o las consultas son complejas?	2
10	¿Es complejo el procesamiento interno?	4
11	¿El código diseñado será reutilizado?	4
12	¿Se incluye la conversión e instalación en diseño?	2
13	¿Es diseñado el sistema para instalaciones múltiples en diferentes organizaciones?	0
14	¿La aplicación está diseñada para facilitar el cambio y para que el usuario lo use fácilmente?	5
Factor de Complejidad Total (FCT)		49

Fuente: Elaboración Propia

La ecuación de punto de función con la que se calculara la funcionalidad del sistema es la siguiente:

$$PF = \text{Conteo total} * [\beta + (\alpha * \sum(Fi))]$$

Dónde:

PF: funcionalidad de estimación del sistema.

Conteo total: representa la suma de todas las entradas.

β : representa la portabilidad subjetiva estimada de la confiabilidad.

α : error estimado del sistema (0.01).

$\sum(Fi)$: valores de ajuste de complejidad(0,65).

Remplazamos en la ecuación para un nivel de confianza del 65%

$$PF_{\text{real}} = 548 * [0.65 + (0.01 * 49)]$$

$$PF_{\text{real}} = 624,72$$

Ahora calcularemos a un nivel de confianza del 100%.

$$PF_{\text{esperado}} = 548 * [1 + (0.01 * 49)]$$

$$PF_{\text{esperado}} = 816.52$$

El porcentaje de funcionalidad será:

$$\% \text{ PF} = \text{PF}_{\text{real}} / \text{PF}_{\text{esperado}}$$

$$\% \text{ PF} = 624,72 / 816,52$$

$$\% \text{ PF} = 0,7651 * 100$$

$$\% \text{ PF} = 77\%$$

Por lo tanto se tiene una funcionalidad de un **77%**, lo que indica que el producto de software está en grado aceptable, en la satisfacción de las necesidades para las cuales fue diseñado.

3.2.1.2. Confiabilidad

La confiabilidad representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la probabilidad de que un sistema presente fallos según la función:

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

Sean los datos:

$$\text{PF} = 77\%$$

$$\mu = 0,1 \text{ (es decir 1 error en cada 10 ejecuciones)}$$

$$t = 20 \text{ días}$$

Conociendo la funcionalidad del 77% del sistema calculamos para el periodo establecido.

$$F(12) = (0,77) * e^{-(0,1 * 20)}$$

$$F(12) = 0,104$$

Por tanto en un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla.

$$G(t) = 1 - F(t)$$

Reemplazando datos se tiene:

$$G(t) = 1 - 0,104$$

$$G(t) = 0,90$$

$$G(t) = 90\%$$

Por tanto la confiabilidad del sistema es **90%** en un periodo de 20 días como tiempo de prueba.

3.2.1.3. Usabilidad

La usabilidad se evalúa tomando en cuenta factores humanos, está relacionado con los usuarios del sistema, es la valoración individual y toma en cuenta los criterios de aprendizaje, comprensión, operatividad y atractividad. Para esto se utilizó un interrogatorio y nivel de valores para la evaluación, según la función:

$$FU = \left[\frac{\text{Sum}(xi)}{n} \right] * 100$$

Dónde:

Xi=Sumatoria de valores

N= Número de preguntas

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

Tabla Nº 3. 18: Rangos para Evaluar la Usabilidad

ESCALA	VALOR
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Fuente: Elaboración Propia

Tabla Nº 3. 19: Preguntas para la valoración de Usabilidad

Nº	PREGUNTAS	RESPUESTAS		PONDERACION %
		SI	NO	
1	¿Puede utilizar con facilidad el sistema?	4	1	80%
2	¿Puede controlar operaciones que el sistema solicita?	4	1	80%
3	¿Las respuestas del sistema son complicadas?	1	4	80%

4	¿El sistema permitió la retroalimentación de información?	5	0	100%
5	¿El sistema cuenta con interface agradable a la vista?	5	0	100%
6	¿La respuesta del sistema es satisfactoria?	4	1	80%
7	¿Le parecen complicadas las funciones del sistema?	1	4	80%
8	¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	4	80%
9	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	100%
10	¿El sistema llevo a cumplir con todas sus expectativas?	4	1	80%
USABILIDAD				86%

Fuente: Elaboración Propia

Por tanto se concluye que el sistema tiene un grado del **86 %** aceptable en su facilidad de uso, comprensión y aprendizaje.

3.2.1.4. Eficiencia

Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los sub atributos de comportamiento del tiempo y de los recursos.

Tabla N° 3. 20: Rangos para Evaluar la Eficiencia

ESCALA	VALOR
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Fuente: Elaboración Propia

Valorando la eficiencia del sistema, se tienen las siguientes valoraciones, tal como se muestra en la tabla 3.22.

Tabla Nº 3. 21: Valoración para la Eficiencia

Nº	PREGUNTAS	VALOR	PONDERACION%
1	¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5	100%
2	¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	4	80%
3	¿Los datos de salida están presentados de modo que se entienden de inmediato?	4	80%
4	¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	5	100%
5	¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	4	80%
EFICIENCIA			88%

Fuente: Elaboración Propia

Por tanto se concluye que el sistema tiene un grado del **88 %** aceptable en los recursos del sistema.

3.2.1.5. Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la carrera tenga. Para este fin Pressman nos sugiere un Índice de Madurez del Software (IMS) para determinar la estabilidad de un producto software. Dicha IMS es calculada según la ecuación:

$$IMS = [M_T - (F_a + F_c + F_d)] / MT$$

Dónde:

M_T = Número de módulos de la versión actual.

F_c = Número de módulos en la versión actual que se han cambiado.

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

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

Los parámetros que se obtiene con el sistema es:

$$M_T = 5$$

$$F_c = 0$$

$$F_a = 1$$

$$F_d = 0$$

Reemplazando tenemos:

$$IMS = [5 - (1 + 0 + 0)] / 5$$

$$IMS = 0.8$$

Interpretando podemos decir que el nuevo sistema tiene una estabilidad de **80%** que es la facilidad de mantenimiento, el **20%** restante es el margen de error correspondiente a los cambios y modificaciones efectuados desde el prototipo de la versión actual.

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

Así mismo para medir la facilidad de mantenimiento aplicamos la métrica de tiempo medio TMC.

$$TMC = TA + TD + TI + TP$$

Dónde:

TA= Tiempo en el que se tarda en analizar la petición de cambio.

TD= Tiempo empleado en diseñar una modificación adecuada.

TI= Tiempo necesario para implementar el cambio.

TP= Tiempo empleado en probar y distribuir el cambio a todos los usuarios.

3.2.1.5.1. Mantenimiento Correctivo

Se toma en cuenta los cambios que se pueda realizar al sistema considerando las correcciones en minutos u horas.

Tabla N° 3. 22: Mantenimiento Correctivo

EN EL MEJOR DE LOS CASOS[MIN]	EN EL PEOR DE LOS CASOS [HRS]
TMC= 20+30+50+100	TMC=24+24+48+25
TMC=200=3.3 HRS.	TMC=121 HRS.

Fuente: Elaboración Propia

El promedio de $3.3+121/2 = 62$ horas.

La facilidad de mantenimiento adaptivo es de **62 horas**. Lo que indica que en dos días aproximadamente demora en algún cambio.

3.2.1.5.2. Mantenimiento de Mejora

Se toma en cuenta las mejoras al sistema considerando que se puede implementar nuevos módulos.

Tabla N° 3. 23: Mantenimiento de Mejora

EN EL MEJOR DE LOS CASOS[HRS.]	EN EL PEOR DE LOS CASOS [DIAS]
TMC= 5+5+8+12	TMC=15+30+25+10
TMC=30 HRS.	TMC=80=1920 HRS.

Fuente: Elaboración Propia

El promedio de $(30+1920/2)=975$ horas.

Por tanto la facilidad de mantenimiento de mejora es de **975 HRS.**, lo que indica en un mes se daría una nueva mejora al sistema, teniendo en cuenta el cambio para la mejora será de todo el sistema.

3.2.1.6. Portabilidad

Se define como la facilidad con que el software puede ser llevado de un entorno de Hardware y Software a otro. Donde se deben tomar atributos internos como ser: Facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio y soporte de sistemas operativos.

Para tener una idea más clara de la portabilidad del sistema se valorara a cada atributo con el rango de valores de la tabla 3.25.

Tabla Nº 3. 24: Rangos para Evaluar la Portabilidad

ESCALA	VALOR
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Fuente: Elaboración Propia

Valorando la portabilidad del sistema, se tienen las siguientes apreciaciones:

- **Facilidad de instalación**, para la instalación del sistema se debe tener instalado en el servidor el gestor de base de datos MariaDB, servidor Web Apache y soporte de PHP.
- **Facilidad de ajuste**, es fácil de ajustar el servidor donde se lo instale, se tendrá que ajustar el archivo de conexión a la base de datos, con los parámetros propios del servidor.
- **Facilidad de adaptación al cambio**, si tomamos en cuenta la migración del sistema a otro gestor de base de datos, la base de datos del sistema puede ser migrada a MySQL, SQL server, Oracle y Postgres.
- **Soporte de sistemas operativos**, el sistema puede ser instalado en servidores con sistemas Windows y Linux, los cuales son muy utilizados en la actualidad.

Tabla Nº 3. 25: Valoración de Portabilidad

ATRIBUTO	VALOR	PONDERACION %
Facilidad de instalación	4	80%
Facilidad de ajuste	4	80%
Facilidad de adaptación al cambio	4	80%
Soporte de sistemas operativos	5	100%

PORTABILIDAD	85%
---------------------	------------

Fuente: Elaboración Propia

Por lo tanto se concluye que el sistema tiene un grado de portabilidad aceptable del **85%**, lo que permite afirmar que el sistema puede adaptarse y transportarse a nuevo entorno.

3.2.1.7. Resultado Final

En resumen, los resultados de las características de la normas ISO/IEC 9126 son:

Tabla N° 3. 26: Resultados de las Características de Calidad

ESCALA	VALOR
Funcionalidad	77%,
Confiabilidad	90%
Usabilidad	86 %
Eficiencia	88%
Mantenibilidad	80%
Portabilidad	85%
PROMEDIO	84%

Fuente: Elaboración Propia

Se puede observar que el resultado de grado de calidad del software es **84%**, satisfacción que el usuario tendrá al momento de utilizar el sistema.

3.3. Método de Estimación de Costos

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

3.3.1.1. Modelo COCOMO II

Para la determinación del costo del software desarrollado, se hará uso del modelo constructivo de costos COCOMO II, orientado a los puntos de función.

De la misma forma se hará uso el **Modelo Intermedio** que define COCOMO, dado que realiza las estimaciones con bastante precisión.

3.3.1.1.1. Estimación del punto función

a) Calculo del punto función (PF)

Tabla N° 3. 27: Valores del dominio de Información

PARAMETRO	CUENTA	FACTOR DE PONDERACION			SUB-TOTAL
			MEDIO		
Número de entradas externas (EE)	28	3	4	6	112
Número de salidas externas (SE)	30	4	5	7	150
Número de consultas externas (CE)	23	3	4	6	92
Número de archivos lógicos internos (ALI)	18	7	10	15	180
Número de archivos de interfaces externas (AIE)	2	5	7	10	14
TOTAL					548

Fuente: Elaboración Propia

b) Calculo de valores de ajustes de la complejidad

Los valores son respondidos usando una escala desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial).

Tabla N° 3. 28: Valores de ajustes de la complejidad

N°	FACTOR DE COMPLEJIDAD	VALOR
1	¿El sistema requiere respaldo y recuperación confiable?	5
2	¿Se requiere comunicación de datos?	4
3	¿Hay funciones distribuidas de proceso?	3
4	¿El desempeño es crítico?	4

5	¿Se ejecuta el sistema en un entorno operativo existente y frecuentemente utilizado?	5
6	¿Requiere el sistema entrada de datos interactivo?	4
7	¿Requiere la entrada de datos interactiva que las entradas lleven a cabo múltiples operaciones?	5
8	¿Se actualiza los archivos maestros en forma interactiva?	2
9	¿Las entradas, las salidas, los archivos o las consultas son complejas?	2
10	¿Es complejo el procesamiento interno?	4
11	¿El código diseñado será reutilizado?	4
12	¿Se incluye la conversión e instalación en diseño?	2
13	¿Es diseñado el sistema para instalaciones múltiples en diferentes organizaciones?	0
14	¿La aplicación está diseñada para facilitar el cambio y para que el usuario lo use fácilmente?	5
TOTAL (F)		49

Fuente: Elaboración Propia

Aplicando la siguiente fórmula:

$$\text{Factor de Ajuste} = 0.65 + 0.01 * F$$

$$\text{Factor de ajustes} = 0.65 + 0.01 * 49$$

$$\text{Factor de ajuste} = 1.14$$

c) Cálculo de punto función

El cálculo de punto función se basa en la fórmula:

$$PF = \text{Cuenta Total} * \text{Factor de ajuste}$$

$$PF = 548 * 1.14$$

$$PF = 624,72$$

d) Conversión de los puntos de función(PF) a KDLC

Ahora convertimos los PF a miles de líneas de código. Para ello veremos la tabla 3.30.

Tabla N° 3. 29: Conversión de los puntos de función a KDLC

LENGUAJE	NIVEL	FACTOR LDC/PF
Ensamblador	10	320
C	2.5	128
Visual C++	9.5	34
Java	6	53
ASP	9	36
<u>PHP</u>	<u>11</u>	<u>29</u>

Fuente: Elaboración Propia

Reemplazamos datos en las siguientes fórmulas:

$$\text{LDC} = \text{PF} * \text{Factor LDC/PF}$$

$$\text{LDC} = 624.72 * 29$$

$$\text{LDC} = 18117$$

Las líneas de código en su totalidad son **18117**, entonces el número estimado de líneas de código distribuidas en miles es:

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

$$\text{KLDC} = 18117 / 1000$$

$$\text{KLDC} = 18$$

Por tanto existen **18** líneas de código distribuidas para el proyecto.

e) Aplicación de las fórmulas del modelo intermedio

A continuación presentamos las ecuaciones que nos permitirán el costo total del software.

Tabla N° 3. 30: Ecuaciones del modelo intermedio que define COCOMO

VARIABLE	ECUACION	TIPO/UNIDAD
Esfuerzo requerido por el proyecto	$E = a * (\text{KLCD})^b \text{FAE}$	Persona/mes

Tiempo requerido por el proyecto	$T=c*(E)^n$	Meses
Número de personas requeridas para el proyecto	$P=E/T$	Personas
COSTO TOTAL	CT=sueldo mes*P*T	\$us.

Fuente: (Boehm, 1981)

Es decir que el número de Kilos de líneas de código es de **18**. Así en este caso se optó por el tipo **orgánico**, porque es el apropiado, pues el número de líneas de código no supera los 50 KLC, por consiguiente, los coeficientes que se usaran serán los valores que se detallan en la tabla 3.32.

Tabla Nº 3. 31: Aplicación del Modelo Intermedio

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2,8	1.2	2.5	0.32

Fuente: (Pressman, 2005)

Para hallar la variable FAE, se utilizara la tabla de multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observa en la siguiente tabla.

Tabla Nº 3. 32: Cálculo de Atributos FAE

CONDUCTORES DE COSTE	VALORACIÓN					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Atributos de Software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de Hardware						
Restricciones del tiempo de ejecución	-	-	1,00	1,11	1,30	1,66

Restricciones de memoria visual	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina Virtual.	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador.	-	0,87	1,00	1,07	1,15	-
Atributos de Personal						
Capacidad del analista.	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 los Programadores.	1,42	1,17	1,00	0,86	0,70	-
Experiencia en S.O. Utilizado.	1,21	1,10	1,00	0,90	-	-
Experiencia en el Lenguaje de programación.	1,14	1,07	1,00	0,95	-	-
Atributos del Proyecto						
Prácticas de Programación moderna	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas de Software.	1,24	1,10	1,00	0,91	0,83	-
Limitaciones de planificación del proyecto.	1,23	1,08	1,00	1,04	1,10	-
TOTAL			0,72			

Fuente: Elaboración Propia

Por lo tanto, nuestra variable FAE será:

$$FAE = 15 * 1,00 * 1,15 * 1,00 * 1,00 * 1,00 * 1,00 * 0,86 * 0,82 * 0,86 * 1,00 * 0,95 * 0,91 * 1,00 * 1,04$$

$$FAE = 0,72$$

Aplicando y reemplazando valores a las fórmulas, se tiene:

✓ **Cálculo Esfuerzo**

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

$$E = 2,4 * (18)^{1,05} * 0,72$$

$$E = 36 \text{ (Personas/mes)}$$

✓ **Cálculo del Tiempo**

$$T = c * (E)^n \text{ (Meses)}$$

$$T = 2,5 * 36^{0,38}$$

T=10 (Meses)

✓ **Cálculo de la Productividad**

$$PR=LCD/E \text{ (Meses)}$$

$$PR=18117/36 \text{ (Meses)}$$

$$PR=503(LCD/Personas \text{ Mes})$$

✓ **Cálculo del Personal Requerido**

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

$$P=36/10 \text{ (Personas)}$$

$$P=3,6 \text{ (Personas)}$$

$$P=4 \text{ (Personas)}$$

✓ **Costo Total del Proyecto**

El salario de un programador puede fluctuar entre los 715\$us, cifra que será tomada en cuenta para la siguiente estimación.

Costo Total=P*Salario medio entre los programadores y analistas

Aplicando la fórmula anterior tenemos:

$$\text{Costo Total} =4*715=2860\$$$

En resumen, se requiere 4 personas estimando un trabajo de 10 meses y con un costo total **2860\$**.

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4. Conclusiones y Recomendaciones

En este capítulo se dará las conclusiones pertinentes y recomendaciones que son necesarias para el buen funcionamiento del sistema.

4.1. Conclusiones

Con el desarrollo del presente proyecto la Unidad de Kardex Académico de la carrera Contaduría Pública-UPEA se beneficia por contar con una base de datos que contiene información trascendental para el desempeño de sus actividades en el área académica, logrando cumplir los objetivos establecidos inicialmente en el capítulo I.

Se lograron cumplir con los siguientes objetivos específicos:

- Se logró desarrollar un módulo de pensum, para dar estricto cumplimiento de los prerrequisitos establecidos.
- .Se consiguió realizar un módulo de planificación académica, inscripción de estudiantes y registro de notas.
- Posee consulta de estudiantes permitiendo realizar su seguimiento académico (asignaturas inscritas, récord, historial de notas, certificaciones y registro personal).
- Cuenta con los reportes necesarios para una toma de decisiones en la información correspondiente (estadísticas de inscritos por gestiones, género y aprovechamiento estudiantil).
- El sistema fue desarrollado haciendo el uso de la metodología UWE, que guiaron de forma efectiva en el desarrollo del sistema en todas sus etapas.
- Así mismo los conocimientos adquiridos durante los ciclos de estudio en la Universidad Pública de El Alto (UPEA) se integraron y coadyuvaron a la conclusión satisfactoria de este trabajo.

4.2. Recomendaciones

El personal de la carrera de Contaduría Pública encargada de administrar el sistema debe incorporar normas y políticas de uso del sistema.

Así mismo el sistema está desarrollado y orientado al uso de personas que tengan conocimiento básico en sistemas computacionales.

Si bien el sistema cuenta con un mayor nivel de confiabilidad y seguridad, es necesario realizar acciones para permitir mantener la madurez del sistema, para ello se recomienda:

- Es recomendable tener cuidado respecto a las claves de acceso al sistema, para evitar la vulnerabilidad de los datos que puedan afectar a la carrera.
- Realizar copias de seguridad de la base de datos (Backup).
- Debido a las características del sistema se recomienda llevar a cabo la capacitación correspondiente.
- Tomar en cuenta para un futuro la implementación de un módulo de docente para el registro de notas en línea mediante la web, tomando en cuenta la seguridad que conlleva el mismo.
- En cuanto al análisis y diseño del sistema, cuando la carrera requiera incluir un módulo nuevo, se recomienda primero revisar el documento para poder tomar una buena decisión, ya que el diseño del sistema representa componentes reutilizables que pueden ayudar a reutilizar el diseño del sistema actual.

BIBLIOGRAFÍA

- *CnaChile*. (2018, 09 26). Retrieved 10 29, 2020, from Sistema de Gestion Académica: <https://www.cnachile.cl/noticias/Paginas/GESTI%C3%93N-ACAD%C3%89MICA.aspx>
- Boehm, B. (1981). *Software Engineering Economic*. Pearson: Prentice Hal.
- Cespedes, J. (2001). *Metodología de la Investigación*. La Paz: Universidad Técnica de Oruro.
- Coulouris, G. (2001). *Gestió Académica conceptos y diseño*. Madrid España: Tercera Edición.
- Flores J., A. (2011, 6 24). Retrieved 5 9, 2020, from Tesis Portal Academico utilizando herramientas con GNU: www.nitropdf.com
- Galiano, L. (. (2012, 11 3). *Metodología UWE*. Retrieved Junio 05, 2020, from Metodología UWE aplicada a la solución Informática: elproyectedeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.htm
- Gomez, F. (2015, 09 6). *Codeigniter*. Retrieved 4 7, 2019, from Codeigniter: www/miproyecto/versionescodeigniter/codeigniter
- Gomez, J. (2011). *Gestión Académica*. S.A.: Editorial.
- Guioteca. (2011, 2 19). *HTML5*. Retrieved 4 12, 2018, from HTML5: www.guioteca.com/internet/¿que-es-html5-y-que-cambios-introduce/
- Long, L. (1999). *Introducion a los sistemas de informacion*. Buenos Aires.
- MCLeod, R. (2000). *Sistemas de Información Gerencial*. Mexico: Prentice Hall.
- Olsina, L. (1999, 11 2). *Ingeniería de Software en la Web*. Retrieved 6 8, 2019, from Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web: www.dsi.uclm.es/personal/FranciscoMSimarro/cedasi/Olsina-Tesis.pdf
- Pantaleo, G. (2012). *Calidad en el desarrolllo de Software*. RA-MA.

- Pressman, R. (2005). *Ingeniería de Software, Quinta Edición*. Mexico: Mc.Graw Hil.
- Scott, G. (2010). *Sistema de Información*. McGraw-Hill.
- Sólida, T. (2017, 11 12). *DB-ENGINES*:. Retrieved 12 4, 2019, from Comparación de propiedades de SGBD MariaDB VS PostgreSQL: db-engines.com/en/system/MariaDB%3BPostgreSQL
- Suarez, C. (2012). *Sistema de Información.Herramientas prácticas para la gestión empresarial*. Alfaomega.
- Vargas, M. S. (2017, junio 25). *Ingeniería guiada por modelos*. Retrieved from Ingeniera Web Basada en UML: marcelosalasvargas.blogspot.com/2017
- Villalobos, J. (2010, 9 5). *Código Programación*. Retrieved 11 15, 2019, from Comparación PHP.JAVA,ASP.NET: codigoprogramacion.com/programacionweb/comparacion-php-jsp-asp.html
- Zeballos, D. (2017, Febrero 19). *Tutorial UWE*. Retrieved from Tutorial sobre UWE: es.scribd.com/document/tutorial-UWE

ANEXOS

ANEXO A

Tabla Nº 1. 1: Cuadro Comparativo

CRITERIO	PHP	ASP.NET	JAVA
Bajo Costo	4	1	3
Portabilidad	4	3	4
Seguridad	2	4	4
Estabilidad	4	4	4
Acceso a Bases de Datos	4	4	4
Multiplataforma	4	3	4
Programación Orientado a Objetos	3	4	4
Bajo Requerimiento de Hardware	4	2	3
Aplicaciones con Alta Complejidad	3	4	4
Fácil Desarrollo	4	3	2
Facilidad de Ayuda	4	2	2
Soporte XML	4	4	4
Velocidad de Ejecución	4	3	3
Soporte Técnico	2	4	3
IDEs Disponibles	2	4	4
Curva de Aprendizaje	4	2	2
Servidores Web disponibles en Internet	4	2	1
TOTAL:	60	53	55

Fuente: (Villalobos, 2010)