

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN ACADÉMICA CASO: INSTITUTO TECNOLÓGICO RVDO. PADRE SEBASTIÁN OBERMAIER”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

Mención: Informática y Comunicaciones

Postulante: Yessica Aguilar Garcia

Tutor Metodológico: ING. Marisol Arguedas Balladares

Tutor Especialista: ING. Juan Regis Muñoz Sirpa

Tutor Revisor: ING.Freddy Salgueiro Trujillo

EL ALTO-BOLIVIA

2019

DEDICATORIA:

El presente proyecto está dedicado con mucho cariño:
A Dios por su amor incondicional, por concederme la vida y fortaleza para terminar este proyecto de grado.
A mi estimado padre Sergio, a mi querida madre Udelia y a mi adorado pequeño Luan, quienes son la principal inspiración de superación y fortaleza en mi vida.

AGRADECIMIENTOS

A nuestro Dios sobre todas las cosas por concederme la vida y la salud para culminar una etapa más en mi vida.

Agradecer a mis padres Sergio y Udelia por darme todo su apoyo y comprensión incondicional para que salga adelante.

A mi pequeño Luan quien fue mi inspiración para culminar este proyecto de grado.

A una persona muy especial y valiosa para mí, que cambio mi vida y por brindarme aliento y ánimo para la culminación de este proyecto.

Al Ing. Sergio Garcia por sus consejos, por guiarme a lo largo del desarrollo de este proyecto que, sin su ayuda no hubiese seguido adelante y llegado a su culminación, gracias a sus constantes sugerencias, pudo ser finalizado de manera exitosa.

Al Ing. Juan Regis Muñoz, por brindarme su apoyo incondicional, colaboración, paciencia y quien me dedico su tiempo para guiarme y aconsejarme en el desarrollo de este proyecto.

Al Ing. Freddy Salgueiro, por su colaboración, paciencia y orientación de este proyecto.

A la Ing. Marisol Arguedas por guiarme y forjarme a lo largo de mi estudio, por sus constantes sugerencias y por su paciencia.

A todos los docentes de la carrera de Ingeniería de Sistemas que me guiaron y forjaron a lo largo de mi estudio agradecerles por haberme dado la herramienta base para salir a una nueva etapa académica.

Agradecer también a la Dirección Académica del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier, especialmente a la Lic. Eufrazio Apaza por darme la oportunidad de realizar este proyecto.

De todo corazón muchas gracias a todos. . .

RESUMEN

Hoy en día la información en el área académica es un recurso de gran importancia para toda Institución, pública y/o privada que administra enormes volúmenes de datos. 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.

El Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O), está experimentando un crecimiento significativo en la cantidad de estudiantes en sus distintas carreras. El presente trabajo hace énfasis a los procesos del área académica como gestión del plan de estudios, 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 web para la gestión Académica en el Instituto Tecnológico Rvdo. Padre Sebastián Obermaier. Al tratarse de un Instituto Superior de Educación, 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, Bootstrap, HTML5, CSS3 y CMS WordPress.

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 del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.).....	3
1.2.3. Misión.....	3
1.2.4. Visión.....	3
1.2.5. Antecedentes Académicos.....	4
1.2.6. Antecedentes Nacionales.....	4
1.2.7. Antecedentes Internacionales.....	5
1.3. Planteamiento del Problema.....	6
1.3.1. Problema Principal.....	8
1.3.2. Problemas Secundarios.....	8
1.4. Objetivos.....	9
1.4.1. Objetivo General.....	9
1.4.2. Objetivos Específicos.....	9
1.5. Justificación.....	10
1.5.1. Justificación Técnica.....	10
1.5.2. Justificación Económica.....	10
1.5.3. Justificación Social.....	11
1.6. Metodología Web.....	11
1.6.1. Metodología de Ingeniería Web.....	11
1.6.2. Técnicas de Investigación.....	11
1.6.3. Métricas de Calidad.....	12
1.6.4. Métodos de Estimación de Costos.....	12
1.7. Herramientas.....	12
1.7.1. Lenguaje de programación.....	12
1.7.2. Sistema gestor de base de datos.....	12
1.7.3. Servidor Web Apache.....	13
1.7.4. Otras herramientas.....	13
1.8. Límites y Alcances.....	13
1.8.1. Límites.....	13
1.8.2. Alcances.....	14
1.9. Aportes.....	14
CAPITULO II: MARCO TEÓRICO.....	16
2. Marco Teórico.....	16
2.1. Sistema.....	16
2.1.1. Sistemas de Información.....	16
2.2. Metodología Web.....	17
2.2.1. Metodología Basada en UML (UWE).....	18
2.2.2. Actividades de modelado UWE.....	18
2.2.2.1. Modelo de casos de uso.....	19

2.2.2.2. Modelo de contenido	19
2.2.2.3. Modelo de navegación	20
2.2.2.4. Modelo de presentación	21
2.2.2.1. Fase de desarrollo UWE	22
2.3. Métricas de Calidad de Software	23
2.3.1. Métricas de Calidad ISO/IEC 9126	23
2.3.1.1. Funcionalidad	25
2.3.1.2. Confiabilidad	25
2.3.1.3. Usabilidad	26
2.3.1.4. Eficiencia	26
2.3.1.5. Mantenibilidad	27
2.3.1.6. Portabilidad	27
2.4. Modelo de Estimación de Costos	28
2.4.1. Modelo COCOMO II	28
2.4.1.1. Métricas orientadas a la función: Punto Función (PF)	29
2.4.1.2. Factores	31
2.5. Tecnologías para el Desarrollo de Aplicaciones Web.	34
2.5.1. Lenguaje de programación PHP	34
2.5.2. CodeIgniter	35
2.5.2.1. Modelo Vista Controlador	37
2.5.3. Bootstrap	38
2.5.4. CMS WordPress	38
2.6. Herramientas para la Implementación de Aplicaciones Web	39
2.6.1. Servidor Http Apache	39
2.6.1.1. Servidor Web	40
2.6.2. MariaDB	41
CAPITULO III: MARCO APLICATIVO	42
3. Marco Aplicativo	42
3.1. Modelo del Negocio	42
3.1.1. Análisis actual del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier	42
3.1.2. Obtención de requerimientos	43
3.1.3. Procesos académicos generales	43
3.2. Modelo del Sistema	44
3.2.1.2. Requerimientos NO Funcionales	46
3.2.2. Definición de actores (del sistema)	46
3.3. Aplicación del Modelo UWE	48
3.3.1. Modelo de Casos de Uso	48
3.3.1.1. Análisis de requerimientos	48
3.3.1.1.1. Diagrama de casos de uso comercial	48
3.3.1.1.2. Modelo de Caso de Uso General del Sistema	49
3.3.2. Modelo de Contenido	55
3.3.3. Modelo de Navegación	57
3.3.4. Modelo de Presentación	61
3.3.5. Modelo de Implementación	64
3.4. Métricas de Calidad de Software	67

3.4.1. Factores de Calidad ISO 9126.....	67
3.4.1.1.Funcionalidad	68
3.4.1.2.Confiabilidad.....	72
3.4.1.3.Usabilidad	73
3.4.1.4.Eficiencia.....	74
3.4.1.5.Mantenibilidad	76
3.4.1.5.1.Mantenimiento Correctivo	77
3.4.1.5.2.Mantenimiento de Mejora.....	77
3.4.1.6.Portabilidad.....	78
3.4.1.7.Resultado Final	79
3.5. Método de Estimación de Costos.....	80
3.5.1.1.Modelo COCOMO II	80
3.5.1.1.1.Estimación del punto función.....	80
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES	86
4. Conclusiones y Recomendaciones	86
4.1. Conclusiones	86
4.2. Recomendaciones	87
BIBLIOGRAFÍA.....	88
ANEXOS	90

INDICE DE FIGURAS

	PÁG.
Figura Nº 1.1: Organigrama (I.T.R.P.S.O.)	3
Figura Nº2 1: Modelo de un Sistema Informático	17
Figura Nº2 2: Ingeniería Web Basada en UML	18
Figura Nº2 3: Diagrama de caso de uso en la Metodología UWE.....	19
Figura Nº2 4: Diagrama de contenido en la Metodología UWE	20
Figura Nº2 5: Diagrama de navegación en la Metodología UWE.....	20
Figura Nº2 6: Estereotipos del Diagrama de navegación.....	21
Figura Nº2 7: Diagrama de presentación en la Metodología UWE	21
Figura Nº2 8: Diagrama de Flujo de la Aplicación	36
Figura Nº2 9: Funcionamiento de Modelo, Vista y Contralor.....	37
Figura Nº2 10: Diseño de Bootstrap.....	38
Figura Nº 3. 1: Carreras del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier	42
Figura Nº 3. 2: Modelo de Caso de Uso General del Sistema	49
Figura Nº 3. 3: Diagrama de Caso de Uso: “GESTIÓN DEL PLAN DE ESTUDIOS”	50
Figura Nº 3. 4: Diagrama de Caso de Uso: “PLANIFICACIÓN ACADÉMICA”	51
Figura Nº 3. 5: Diagrama de Caso de Uso: “INSCRIPCIÓN”	52
Figura Nº 3. 6: Diagrama de Caso de Uso: “REGISTRO DE NOTAS”	53
Figura Nº 3. 7: Diagrama de Caso de Uso: “SEGUIMIENTO ACADEMICO DE ESTUDIANTE”	54
Figura Nº 3. 8: Diagrama de Caso de Uso: “BRINDAR INFORMACIÓN”	55
Figura Nº 3. 9: Modelo Conceptual	56
Figura Nº 3. 10: Modelo de Navegación Gestión Plan de Estudios	57
Figura Nº 3. 11: Modelo de Navegación Planificación Académica	58
Figura Nº 3. 12: Modelo de Navegación Inscripción	59
Figura Nº 3. 13: Modelo de Navegación Registro de Notas.....	60
Figura Nº 3. 14: Modelo de Navegación Seguimiento Académico de Estudiante	61
Figura Nº 3. 15: Modelo de Presentación Autenticación	62
Figura Nº 3. 16: Modelo de Presentación Gestión del Plan de Estudios.....	62
Figura Nº 3. 17: Modelo de Presentación Planificación Académica	63
Figura Nº 3. 18: Modelo de Presentación Inscripción	63
Figura Nº 3. 19: Modelo de Presentación Registro de Notas	64
Figura Nº 3. 20: Autenticación inicial con el sistema	64
Figura Nº 3. 21: Interfaz de Gestión del Plan de Estudios (Administrador)	65
Figura Nº 3. 22: Interfaz de Planificación Académica (Dir. Académico).....	65
Figura Nº 3. 23: Interfaz de Inscripción (Estudiante).....	66
Figura Nº 3. 24: Interfaz de Registro de Notas (Docente).....	66
Figura Nº 3. 25: Interfaz de Sitio web.....	67

INDICE DE TABLAS

Tabla Nº 1. 1: Cuadro Comparativo	91
Tabla Nº 2. 1: Fases de Desarrollo UWE.....	22
Tabla Nº 2. 2: Características y sub-características	24
Tabla Nº 2. 3: Coeficientes	29
Tabla Nº 2. 4: Valores del dominio de Información.	30
Tabla Nº 2. 5: Preguntas de valor de frecuencia	31
Tabla Nº 2. 6: Funcionalidad de los lenguajes de Programación.....	32
Tabla Nº 2. 7: Conductores de Coste	32
Tabla Nº 2. 8: Entorno PHP	35
Tabla Nº 3. 1: Tareas Realizadas Para Obtención De Requisitos.....	43
Tabla Nº 3. 2: Requerimientos Funcionales.....	45
Tabla Nº 3. 3: Requerimientos Funcionales.....	46
Tabla Nº 3. 4: Definición de Actores	47
Tabla Nº 3. 5: Caso de Uso “GESTIÓN DEL PLAN DE ESTUDIOS”	50
Tabla Nº 3. 6: Caso de Uso “PLANIFICACIÓN ACADÉMICA”	51
Tabla Nº 3. 7: Caso de Uso “INSCRIPCIÓN”	52
Tabla Nº 3. 8: Caso de Uso “REGISTRO DE NOTAS”	53
Tabla Nº 3. 9: Caso de Uso “SEGUIMIENTO ACADÉMICO DE ESTUDIANTE” ..	54
Tabla Nº 3. 10: Caso de Uso “BRINDAR INFORMACION”	55
Tabla Nº 3. 11: Número de Entradas Externas	68
Tabla Nº 3. 12: Número de Salidas Externas	68
Tabla Nº 3. 13: Número de Consultas Externas	69
Tabla Nº 3. 14: Número de Archivos Lógicos Internos.....	69
Tabla Nº 3. 15: Número de Interfaces Externas	69
Tabla Nº 3. 16: Conteo Total.....	70
Tabla Nº 3. 17: Rangos para evaluar el PF	70
Tabla Nº 3. 18: Factor de Complejidad	71
Tabla Nº 3. 19: Rangos para Evaluar la Usabilidad.....	73
Tabla Nº 3. 20: Cuestionario para la valoración de Usabilidad	74
Tabla Nº 3. 21: Rangos para Evaluar la Eficiencia	75
Tabla Nº 3. 22: Valoración para la Eficiencia	75
Tabla Nº 3. 23: Mantenimiento Correctivo	77
Tabla Nº 3. 24: Mantenimiento de Mejora.....	77
Tabla Nº 3. 25: Rangos para Evaluar la Portabilidad.....	78
Tabla Nº 3. 26: Valoración para la Portabilidad.....	79
Tabla Nº 3. 27: Resultados de las Características de Calidad.....	79
Tabla Nº 3. 28: Valores del dominio de Información	80
Tabla Nº 3. 29: Valores de ajustes de la complejidad	81
Tabla Nº 3. 30: Conversión de los puntos de función a KDLC.....	82
Tabla Nº 3. 31: Ecuaciones del modelo intermedio que define COCOMO	83
Tabla Nº 3. 32: Aplicación del Modelo Intermedio.....	83
Tabla Nº 3. 33: Cálculo de Atributos FAE.....	84

CAPÍTULO I

MARCO PRELIMINAR

1. CAPÍTULO I: MARCO PRELIMINAR

1.1. Introducción

En esta nueva era de revolución de las tecnologías de información, los sistemas que realizan el tratamiento de información se van haciendo necesarios e imprescindibles, más aún si éstos están interconectados vía Intranet o Internet, pues facilitan la comunicación y la integración de los usuarios, brindando la capacidad de compartir dinámicamente los recursos, disminuir el trabajo redundante y optimizar el flujo de la información.

Hoy en día la información en el área académica es un recurso muy importante para toda Institución que se dedica a la enseñanza y en consecuencia, administra grandes volúmenes de información, la demanda de estudiantes por cursar una carrera en éstas Instituciones va creciendo continuamente, tal es el caso del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O), encargado de la formación de profesionales a nivel Técnico Superior, las metodologías aplicadas en los procesos del área académica como gestión del plan de estudios, planificación académica, inscripción, registro de notas y seguimiento académico de estudiantes no son las adecuadas ya que se lo realiza de forma semi manual y desorganizada. Al tratarse de un Instituto Superior de Educación, los procesos del área académica deben presentar resultados con datos fidedignos, para asegurar el prestigio y el éxito de dicha entidad.

En el presente proyecto se pretende afrontar los inconvenientes mencionados arriba, desarrollando el **“Sistema de Información Web para la Gestión Académica en el Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.)”**, que coadyuve en los procesos del área académica, para proporcionar información fidedigna, 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, usando modelo vista controlador, Bootstrap, HTML5, CSS3 y CMS WordPress.

1.2. Antecedentes

1.2.1. Antecedentes Institucionales

El Instituto Tecnológico Santo Toribio de Mogrovejo comienza su andadura en 1997, por iniciativa de Fe y Alegría con el denominativo de Instituto Superior de Electrónica, Informática y Telecomunicaciones Santo Toribio de Mogrovejo y fue el artífice del nombre de la institución, inspirada en el párroco Toribio de Mogrovejo, por la labor inalcanzable de evangelización. Dicha Institución vio la necesidad de implementar un centro a nivel superior y en nuevas tecnologías como son la electrónica, la informática y las telecomunicaciones. En 1998 se realizó la construcción del edificio gracias al Padre Sebastián Obermaier. Ese mismo año y gracias a una financiación del Gobierno Vasco de España, el centro se equipó con equipos electrónicos de primer nivel.

En febrero de 1999, se abrió el centro con el ingreso de los primeros estudiantes y en septiembre del 2000 se obtuvo la Resolución Ministerial Nro. 363/00, lo cual cualifica al I.S.E.I.T. como centro fiscal de enseñanza superior. El 8 de agosto de 2008 se obtiene la Resolución Ministerial Nro. 624/08, la que renueva la R.M. Nro. 363/00. A partir de la Resolución Ministerial Nro. 066/2012, se adecua las carreras de acuerdo a los Diseños Curriculares.

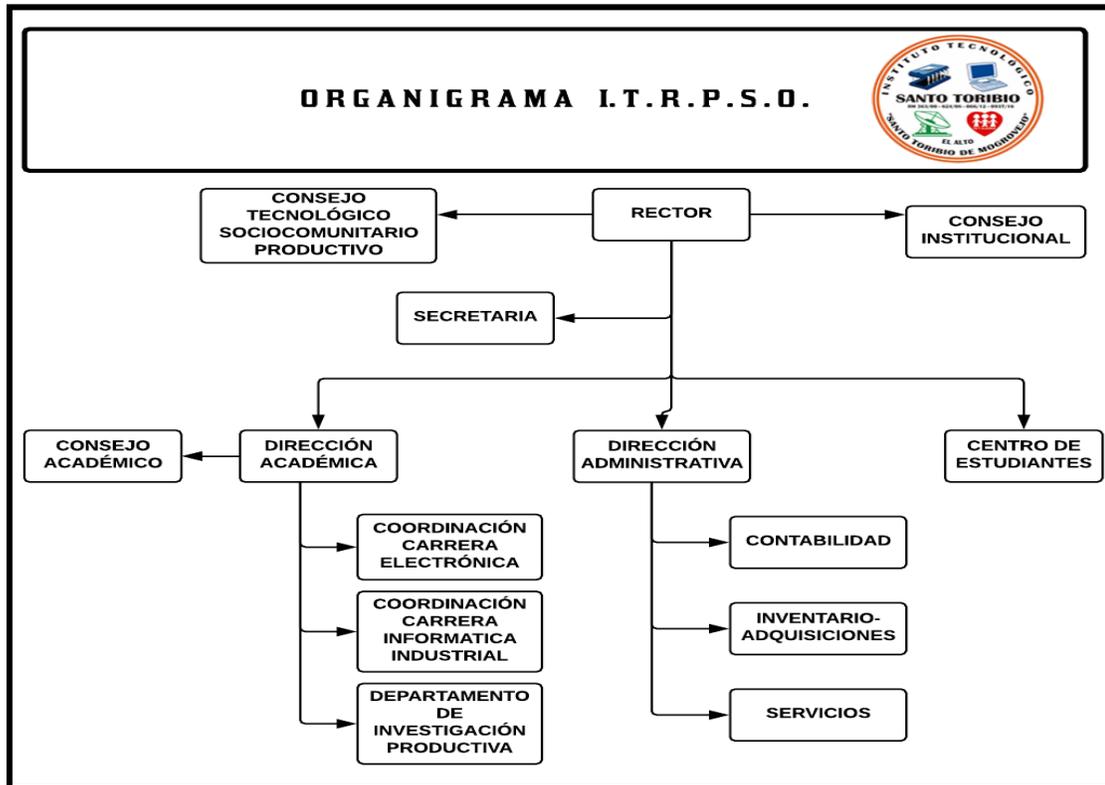
En la actualidad el Instituto Tecnológico Santo Toribio de Mogrovejo (I.S.E.I.T) bajo Resolución Ministerial Nro. 0402/2018 cambio su nominación por “Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.)”, cuenta con 295 estudiantes aproximadamente, 24 docentes y 4 como personal administrativo, con Resolución Ministerial Nro. 0937/16 vigente hasta el día de hoy .Ubicado en el Plan 88 Calle Montero Nro. 123 de la Zona Villa Adela de la ciudad de El Alto de La Paz con las carreras a nivel Técnico Superior de:

- **Informática Industrial**
- **Electrónica**

El I.S.E.I.T. se encuentra estructurado y organizado jerárquicamente como se puede apreciar en **Figura 1.1.**

1.2.2. Organigrama del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.)

Figura Nº 1.1: Organigrama (I.T.R.P.S.O.)



Fuente: Instituto Tecnológico Rvdo. Padre Sebastián Obermaier

1.2.3. Misión

Ser una Institución de Educación Superior comprometidos con el cambio social bajo la visión institucional de Fe y Alegría, formando personas competentes para responder a las demandas productivas de la comunidad, buscando el mejoramiento de la calidad de vida de las personas.

1.2.4. Visión

Ser líder y referente en la ciudad de El Alto en el área de electrónica e informática, que utilice las nuevas tecnologías, medios de comunicación, dirigidos a la investigación, la excelencia, potencialidades de la región y las políticas institucionales de Fe y Alegría.

1.2.5. Antecedentes Académicos

1.2.6. Antecedentes Nacionales

En relación a los trabajos llevados a cabo en el entorno de estudio, se puede hacer 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. Aplicó 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, registró 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, realizara 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, registró 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.

“Desarrollo del sistema de gestión académica de postgrados de la Universidad de Cuenca (SGAP)”. Llevado a cabo por Ximena Paulina Castillo Solórzano, Carmen Jackeline Fernández Cruz, Ruth Jaqueline Rea Rojas y Edison Esaú Tapia Zhingri, Cuenca- Ecuador, Julio del año 2012. En su implementación se utilizó plataforma Java, Servidor GlassFish para poner a prueba la aplicación y el patrón MVC (Modelo Vista Controlador). El sistema tiene por objetivo facilitar la administración de inscripción y los segundos disfruten de las ventajas en cuanto a información. Este proyecto tiene afinidad con el sistema formulado, automatizará el proceso de inscripción, así también se usará el patrón MVC (Modelo Vista Controlador) con el framework CodeIgniter.

1.3. Planteamiento del Problema

El incremento de la población estudiantil en el **Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.)**, hace que la información sea excesiva y difícil de manejarla adecuadamente, además de que este arduo trabajo se realiza de forma semi manual y está concentrado en pocas personas, haciendo dificultosa y morosa estas tareas. Actualmente en la institución no existe un sistema de información para los procesos en el área académica, produciéndose volúmenes considerables de información, lo cual hace notar que en ciertas actividades como ser: **gestión de plan de estudios, planificación académica, inscripción, registro de notas y seguimiento de estudiantes**, el trabajo se acumula y hasta resulta caótico lo que lleva que las tareas no se realicen de manera ordenada y eficiente. (Información Director Académico I.T.R.P.S.O. Lic. Eufrazio Apaza Quispe)

A continuación, se presenta los procesos mencionados anteriormente:

- **Gestión del plan de estudios**, no se establece un orden gradual y armónico de asignaturas con sus respectivas características (código, asignatura, nivel descripción, régimen, menciones, prerrequisitos, homologación de asignaturas, 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, ramos por semestre, asignación de aulas, asignaturas, paralelos, y otros. Este proceso se realiza al inicio de cada semestre 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 la secretaria se encarga de ordenar los mismos por asignaturas o semestres.
- **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, horario, récord, historial de notas, calificaciones de las diferentes asignaturas del semestre correspondiente.

1.3.1. Problema Principal

Los métodos utilizados actualmente para la gestión del plan de estudios, planificación académica, inscripción, registro de notas y seguimiento de estudiantes, procesos que forman parte del área académica del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.) no son apropiadas, dado que se los realiza de forma semi manual y desorganizada, generando información poco confiable, redundante e inoportuna.

1.3.2. Problemas Secundarios

- Dificultad en el proceso que se manejan actualmente para la gestión del plan de estudios, por realizarse de forma semi manual 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 informes estadísticos y reportes (tales como mejores calificaciones, estudiantes más destacados mayores promedios por asignatura y otros.), debido a que se realiza la recopilación de información de manera manual, lo cual es una tarea lenta y repetitiva.

- Deficiente manejo y difusión de información con fines educativos acerca de la institución en la web, lo cual hace que pierdan un potencial Mercado que son los usuarios de Internet.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un Sistema de Información Web para la Gestión del Área Académica del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.), haciendo que la información generada sea veraz, confiable y oportuna.

1.4.2. Objetivos Específicos

- Realizar el registro del plan de estudios de forma automatizada mediante el desarrollo de un módulo para dar estricto cumplimiento de los prerrequisitos establecidos.
- Automatizar la planificación académica, inscripción de estudiantes y el registro de notas para evitar el retraso, duplicidad, pérdida y acumulación de la documentación física.
- Sistematizar la consulta de estudiantes para que realicen su seguimiento académico y puedan acceder a la información: asignaturas inscritas, récord, historial de notas y registro personal.
- Generar reportes automáticos mediante el desarrollo de un módulo, que permita obtener información oportuna, confiable y en tiempo real.
- Difundir la información acerca de la institución mediante un sitio Web, todo lo referente a carreras, cursos, actividades alcanzando llegar de esta manera a la población de la ciudad de El Alto.

1.5. Justificación

1.5.1. Justificación Técnica

El Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.) dispone con equipamiento completo de hardware, software y una infraestructura de red con acceso a Internet, aspectos vitales para el desarrollo del proyecto. Para el desarrollo del sistema se lo hará bajo software libre utilizando: Lenguaje de programación PHP, Servidor web Apache, Sistema gestor de base de datos MariaDB y Framework CodeIgniter.

La factibilidad técnica del proyecto propuesto es viable, ya que la Institución dispone con equipos de computación, Internet, conexiones de red, asimismo cuenta con personal altamente calificado para el manejo del sistema, las tecnologías usadas para el desarrollo son libres de licencias.

1.5.2. Justificación Económica

El sistema web se desarrollará bajo herramientas de software libre, el cual no generará costo alguno para llevar a cabo su realización. Los procesos serán automatizados y permitirá reducir el tiempo de acceso a la información, reducir el excesivo manejo de documentación 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 generará mayores ingresos para la institución, por la inscripción de nuevos estudiantes, así como la opción de difundir información por tener atención de calidad.

1.5.3. Justificación Social

La implementación del sistema optimizará en tiempo de respuesta y atención a:

- Los estudiantes podrán inscribirse vía web, realizarán consultas de las notas obtenidas de acuerdo a la carrera que cursa.
- El plantel docente publicará las notas, que estarán disponibles para los estudiantes.
- La secretaria agilizará los trámites de los estudiantes, registros obtenidos del sistema.
- Los coordinadores de área podrán obtener consultas inmediatas y fidedignas de los estudiantes y docentes registrados.
- El Rector y dirección académica gozarán de información confiable y oportuna.

El contribuir a los objetivos estratégicos de la Institución, permitirá principalmente consolidación, crecimiento y mantendrá su prestigio Educativo. De esta manera la institución será competitiva frente a las demás instituciones.

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. (Céspedes, 2001).**

1.6.3. Métricas de Calidad

Para la evaluación de la calidad se hará uso el 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 (*CO*nstru*ctive CO*st *MO*del)**, 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 (MVC).** Es la técnica más ordenada para crear una aplicación independiente del lenguaje de programación que se utilice. **(Gomez, 2015)**
- **Diseño Responsive con Bootstrap.** Esencialmente está formado por una colección de hojas de estilo y funciones JavaScript auxiliares que permiten construir muy rápidamente un front-end responsive **(GitBook, 2016)**
- **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)**
- **CMS WordPress (Content Management System).** Sistema Gestor de Contenidos, es una plataforma que nos permite gestionar los contenidos de nuestra web, con la que podemos crear, administrar y publicar todo tipo de informaciones en una página web. **(Strato, 2016)**

1.8. Límites y Alcances

1.8.1. Límites

El presente proyecto de grado sistema de información web para la gestión académica del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier se limita solo a la:

- Gestión del plan de estudios
Planificación académica
- Inscripción
- Registro de notas
- Seguimiento de estudiantes

Procesos que se siguen para la gestión del área académica. El sistema funcionara siempre y cuando se mantenga el sistema académico actual. Por otra parte, el sistema no contemplará procesos como inventario, planillas, seguimiento de trámites, control de correspondencia y área administrativa.

1.8.2. Alcances

Para la implementación del sistema se contemplarán los siguientes procesos:

- Habilidad de usuarios al sistema con distintos niveles de acceso de acuerdo a sus requerimientos.
- Registrar en la base de datos la información de inscripción del estudiante, plan de estudios, carreras, asignaturas, menciones, homologaciones, prerrequisitos, datos personales, notas y otros.
- Proceder a la administración de gestión del plan de estudios.
- Registro y asignación de docentes a paralelos por asignatura.
- Llevar a cabo la inscripción de estudiantes nuevos y antiguos.
- Llevar a cabo el registro de notas parciales y finales.
- Emitir hoja de ingreso personal, hoja de inscripción, boletín de notas.
- Generará historial, record académico, reportes e informes.

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 el desarrollar un sistema para Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (**I.T.R.P.S.O.**), que concentre la administración y difusión de información para la gestión académica que se genera en la institución a través de la web, brindando a toda la población de la Ciudad de El Alto, permitiendo a la institución fortalecimiento y prestigio Educativo.

De igual manera promover la importancia y beneficios del uso de sistemas de gestión de versiones (Git) para este tipo de proyectos.

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 (modelo vista controlador,), Bootstrap 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. **(Ortega Z., 2012).**

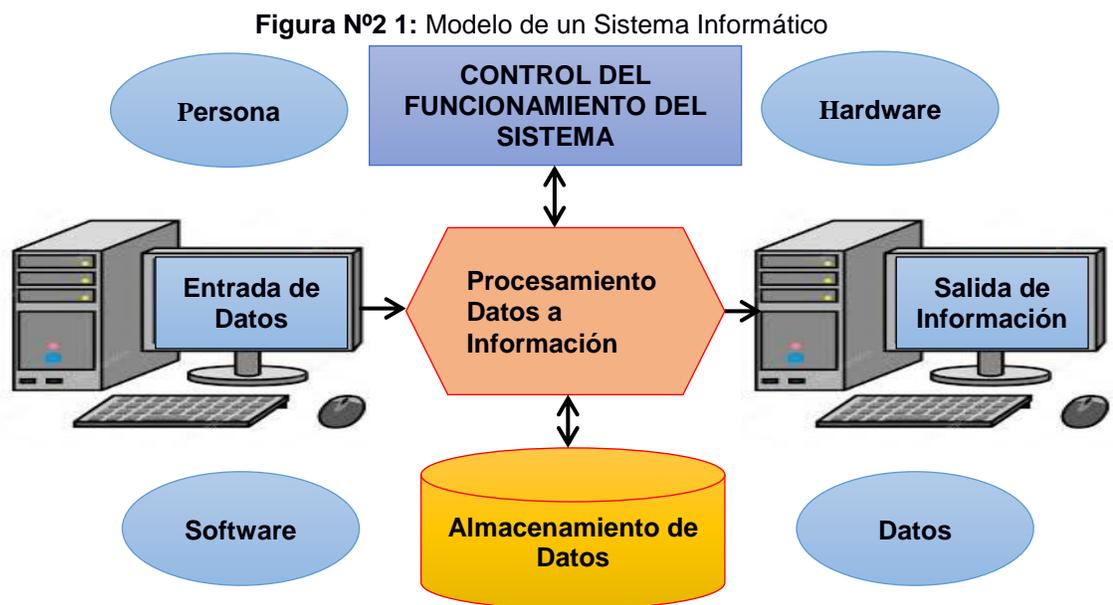
2.1.1. Sistemas de Información

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.

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.

Su uso logra importantes mejoras en la organización e institución al automatizar los procesos operativos y facilitar el logro de ventas competitivas a través de su implementación.

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.



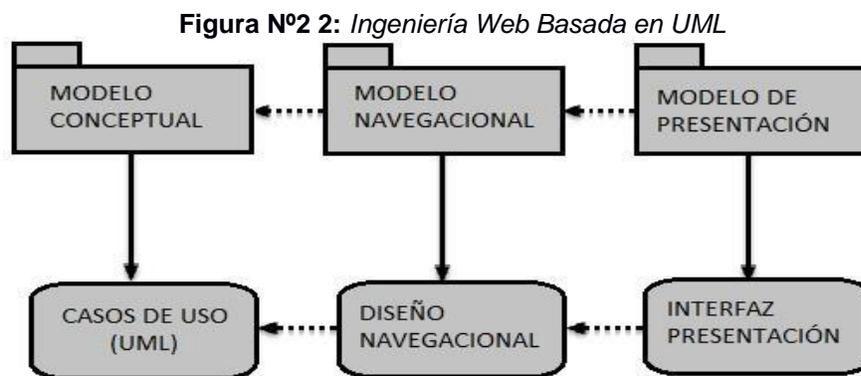
Fuente: Sistema de Información Basado en Computadoras 1992

2.2. 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.

2.2.1. 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 enfocada 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.



Fuente: (Vargas, 2017)

2.2.2. Actividades de modelado UWE

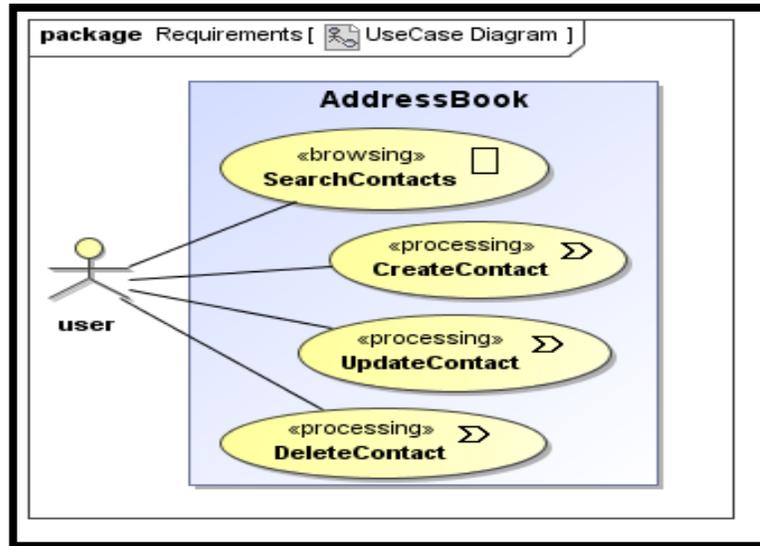
UWE proporciona guías para la construcción de modelos de forma sistemática enfocadas en el estudio de casos de uso. Las actividades del modelado son Análisis de Requerimientos, Diseño Conceptual, Diseño Navegacional y Diseño de Presentación producen los siguientes artefactos:

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

2.2.2.1. 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 N°2 3: Diagrama de caso de uso en la Metodología UWE

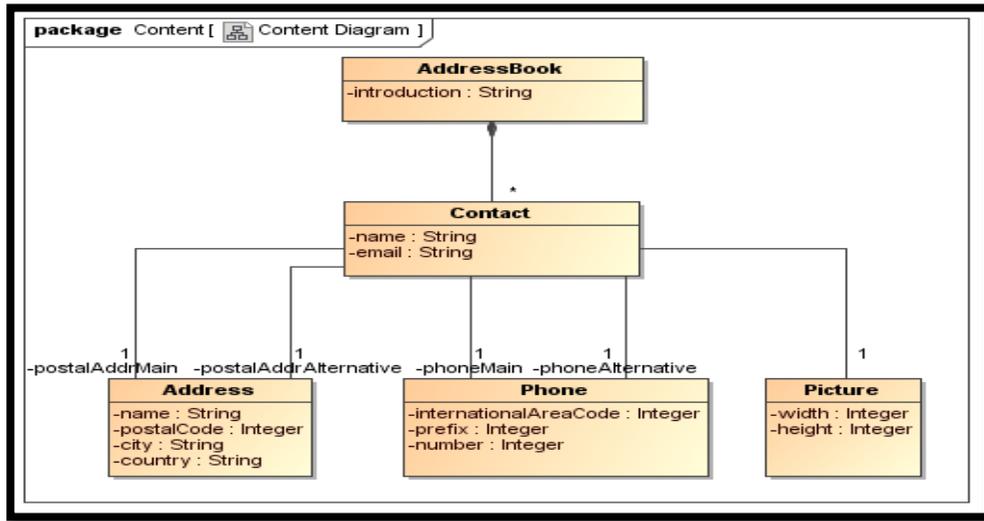


Fuente: (Zeballos, 2017)

2.2.2.2. 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 N°2 4: Diagrama de contenido en la Metodología UWE

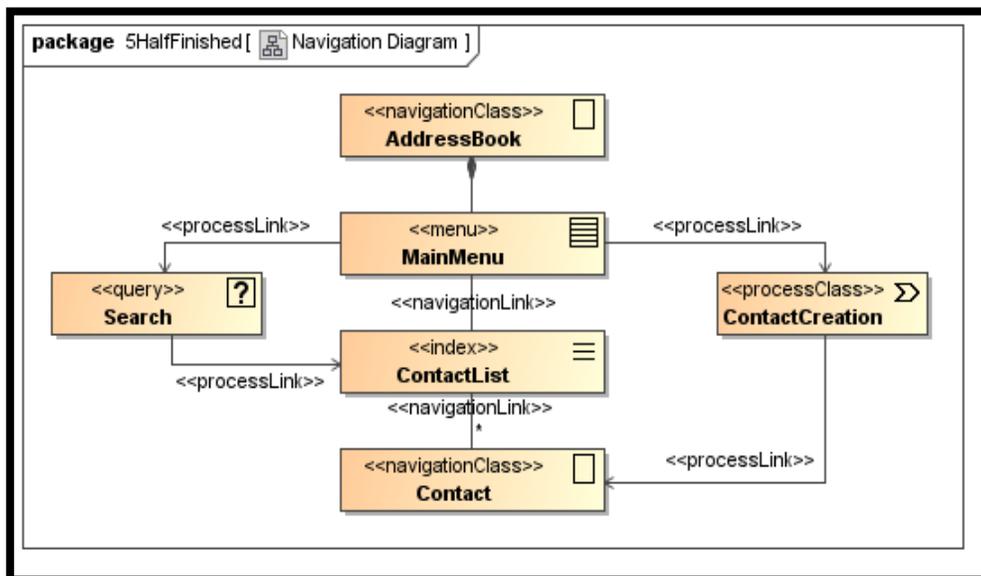


Fuente: (Zeballos, 2017)

2.2.2.3. Modelo de navegación

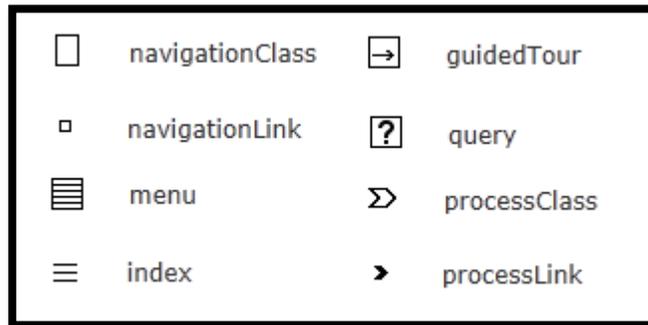
Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. Ello significa que necesitamos un diagrama de conteniendo nodos y enlaces.

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



Fuente: (Zeballos, 2017)

Figura N°2 6: Estereotipos del Diagrama de navegación

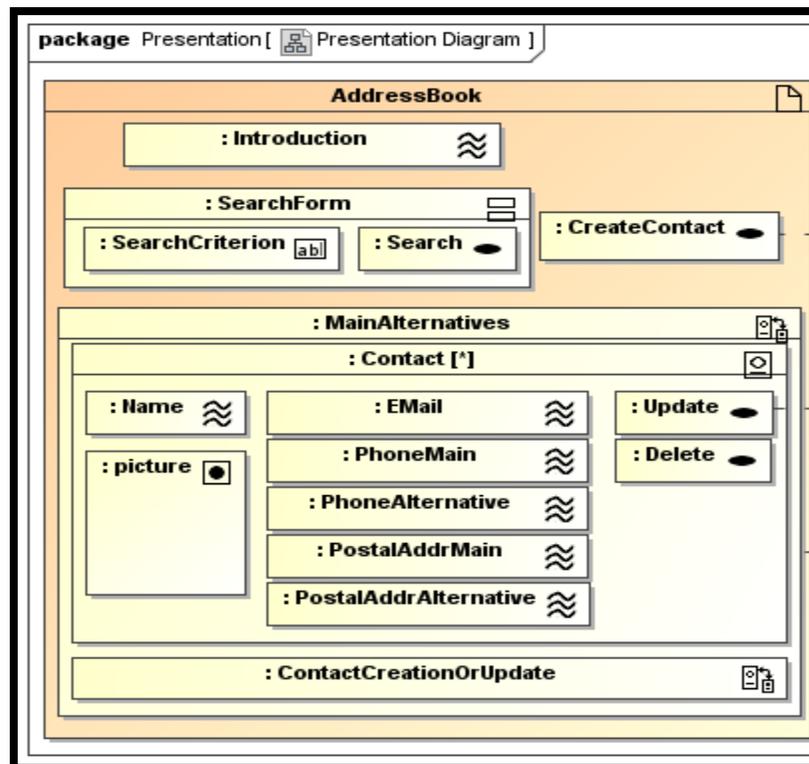


Fuente: (Zeballos, 2017)

2.2.2.4. 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)

2.2.2.1. 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 Lenguaje**, 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 cuatro actividades:

Tabla Nº 2. 1: Fases de Desarrollo UWE

Análisis de Requisitos	Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.. Centra el trabajo en el estudio de los casos de uso.
Diseño del Sistema	Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fases de análisis), el diseño define como estos requisitos se cumplirán, la construcción que debe darse a la aplicación web.
Codificación del Software	Durante esta etapa se realizan tareas comúnmente se conocen como programación, esto consiste esencialmente en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
Pruebas	Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
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.
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.

Fuente: Elaboración Propia

2.3. 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.

2.3.1. 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.

La primera parte contiene las características y sub-características para la determinación de la calidad de un producto. Estas se listan a continuación:

Tabla Nº 2. 2: Características y sub-características

a. Funcionalidad	Adecuación
	Exactitud
	Interoperabilidad
	Conformidad
	Seguridad
b. Fiabilidad	Madurez
	Tolerancia o fallos
	Recuperabilidad
	Conformidad
c. Usabilidad	Comprensibilidad
	Facilidad de Aprendizaje
	Operabilidad
	Atracción
	Conformidad
d. Eficiencia	Comportamiento de Tiempos
	Utilización de recursos
	Conformidad
e. Mantenibilidad	Capacidad de Análisis
	Cambiabilidad
	Estabilidad
	Facilidad de prueba
	Conformidad
f. Portabilidad	Adaptabilidad
	Facilidad de instalación
	Coexistencia
	Capacidad de Reemplazo
	Conformidad

Fuente: (Olsina, 1999)

Veamos sus descripciones muy claramente.

2.3.1.1. 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.

2.3.1.2. 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.

2.3.1.3. 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.

2.3.1.4. 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.

2.3.1.5. 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 funciones 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.

2.3.1.6. 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.

2.4. 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.

2.4.1. Modelo COCOMO II

Entre los distintos métodos de estimación de costes, el modelo **COCOMO** (***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. 3: 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.

2.4.1.1. 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. 4: 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.

2.4.1.2. 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. 5: 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. 6: 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. 7: 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:
 $P = E/T \text{ personas}$

2.5. Tecnologías para el Desarrollo de Aplicaciones Web.

2.5.1. 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 N° 2. 8: 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)

2.5.2. CodeIgniter

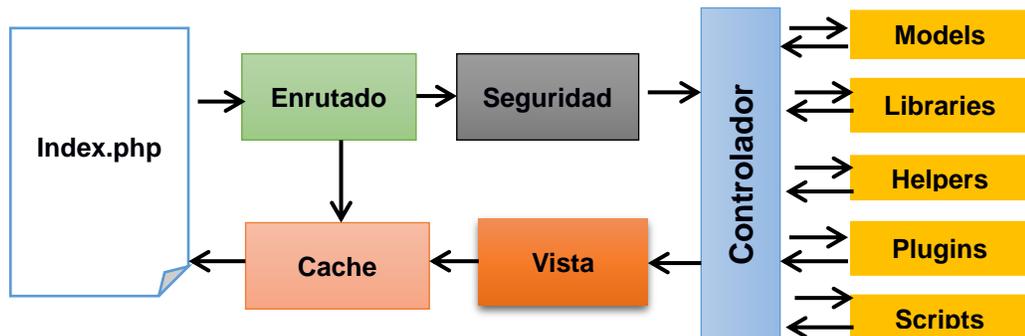
Es un potente Framework PHP con un tamaño muy pequeño, construido para programadores PHP que necesitan una herramienta simple y elegante para crear aplicaciones Web con todas las funciones. Es un producto de código libre, libre de uso para cualquier aplicación. **(Gomez, 2015)**

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 8: Diagrama de Flujo de la Aplicación



Fuente: (Gomez, 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.

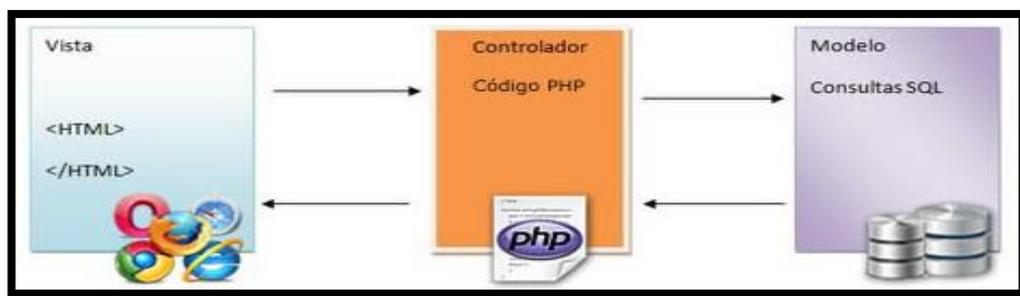
2.5.2.1. Modelo Vista Controlador

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 9: Funcionamiento de Modelo, Vista y Contralor



Fuente: (Gomez, 2015)

2.5.3. Bootstrap

Es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Te permite crear sitios y aplicaciones web responsive, con un diseño espectacular, de una forma fácil y sencilla. Está enfocado a la parte front-end del desarrollo web (HTML, CSS y JavaScript). Bootstrap contiene plantillas de diseño, tipografías, tablas, imágenes, formularios, botones y otros elementos de presentación basados en HTML y CSS. **(Del Valle Hernandez, 2017)**

Figura N°2 10: Diseño de Bootstrap



Fuente: (Del Valle Hernandez, 2017)

Algunas características son:

- Permite crear de manera fácil sitios web que son adaptables o *responsives* para ajustarse a cualquier dispositivo y tamaños de pantalla.
- Podemos conseguir un sitio web muy organizado visualmente.
- Posee un sistema de grillas en la cual maquetamos el sitio mediante columnas que lo hace bastante fácil.
- Se integra perfectamente con librerías de JavaScript.
- Posee una comunidad muy grande y activa que lo respalda.

2.5.4. CMS WordPress

CMS es el acrónimo de **Content Management System** o Sistema Gestor de Contenidos en español. Como su nombre indica, es una plataforma que nos permite gestionar los contenidos de nuestra web.

Por lo tanto, un CMS es una herramienta con la que crear, administrar y publicar todo tipo de informaciones en una página web o blog. WordPress es la opción más popular en el mercado, el rey de los CMS, plataforma sencilla y flexible y con muchas posibilidades. **(Strato, 2016)**

A modo de resumen, podemos decir que las principales características de WordPress son:

- Es una plataforma sencilla que puede gestionar cualquier usuario, sin importar sus conocimientos previos.
- No se necesitan conocimientos de programación o diseño web para hacer una página web
- La instalación y la configuración es muy sencilla.
- La infinidad de diseños disponibles se adaptan a cualquier tipo de página web.
- Incluye la opción de aumentar las capacidades y posibilidades de tu web a través de plugins.
- El mantenimiento y las actualizaciones las puedes hacer tu mismo, sin recurrir a terceros.
- Este CMS cuenta con opciones sencillas para mejorar la posición en los motores de búsqueda como Google.

2.6. 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.

2.6.1. Servidor Http Apache

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

2.6.1.1. 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.

2.6.2. MariaDB

MariaDB es un sistema gestor de bases de datos (**SGBD**), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe, entre otras.

MariaDB surge a raíz de la compra, de la compañía desarrolladora de otro (SGBD) llamado MySQL, por la empresa **Sun Microsystems**. El desarrollador original, decide tomar el código fuente original de MySQL y genera un derivado con mejoras y cambios a los que llama MariaDB. Permitiendo así la existencia de una versión de este producto con licencia GPL (**General Public License**). (**Sólida, 2017**).

Características:

- **Más motores de almacenamiento:** Adicionalmente a los motores estándar MyISAM, Blackhole, CSV, Memory, y Archive.
- **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.
- d) Mejores Pruebas.**
 - a. Más pruebas en la suite de pruebas.
 - b. Errores en pruebas corregidos.
 - c. Pruebas construidas con diferentes opciones de configuración para obtener mejores pruebas.

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.

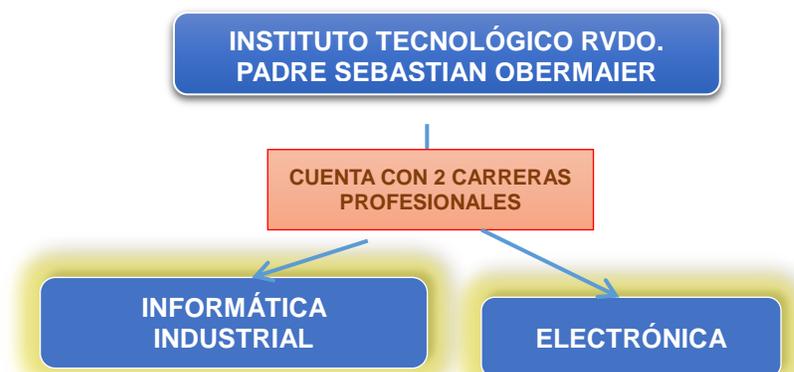
En el presente capítulo se describe y desarrolla el análisis y diseño del “**Sistema de Información Web para la Gestión Académica Instituto Tecnológico Rvdo. Padre Sebastián Obermaier**”, siguiendo las etapas de la metodología UWE (Ingeniería Web basada en UML), normas y técnicas mencionadas en el marco teórico del capítulo II.

3.1. Modelo del Negocio

3.1.1. Análisis actual del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier

El “Instituto Tecnológico Rvdo. Padre Sebastián Obermaier (I.T.R.P.S.O.)”, ubicado en el Plan 88 Calle Montero Nro. 123 de la Zona Villa Adela de la ciudad de El Alto. Inicialmente cuenta con dos carreras de formación Superior que son: Informática Industrial y Electrónica.

Figura N° 3. 1: Carreras del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier



Fuente: Elaboración Propia

3.1.2. Obtención de requerimientos

La tarea de ingeniería de requerimiento es muy 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">✓ Rector✓ Director Académico✓ Secretaria✓ Docentes✓ Estudiantes
Observaciones	En el "Instituto Tecnológico Rvdo. Padre Sebastián Obermaier. Se pudo evidenciar que los procesos que se siguen en el área académica como: gestión del plan de estudios, 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 del plan de estudios**, no se establece un orden gradual y armónico de asignaturas con sus respectivas características (código, asignatura, nivel descripción, régimen, menciones, prerrequisitos, homologación de asignaturas, 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 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, ramos por semestre, asignación de aulas, asignaturas, paralelos, y otros. Este proceso se realiza al inicio de cada semestre 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 la secretaria se encarga de ordenar los mismos por asignaturas o semestres.
- **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 de usuarios 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 y administrativos	Evidente
R.5	Administración de gestión académica (plan de estudios, paralelos Asignaturas, carreras, menciones, prerequisites, homologaciones).	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	Verificación interna de homologación de materias.	
R.10	Control de cupos por paralelo	Oculto
R.11	Inscripción de estudiantes.	Evidente
R.12	Emitir Boleta de Inscripción.	Evidente
R.13	Despliegue de listas ordenadas de estudiantes inscritos por paralelo.	Evidente
R.14	Emisión de reportes sobre estudiantes registrados por carrera y especialidad por gestión.	Evidente
R.15	Registro de Notas de los estudiantes.	Evidente
R.16	Emisión del boletín de Notas.	Evidente
R.17	Emisión de historial de notas récord académico,	Evidente
R.18	Emisión automática de reportes.	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 del Instituto Tecnológico Rvdo. Padre Sebastián Obermaier.

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
Rector	<ul style="list-style-type: none"> ✓ Realizar consultas y emitir reportes generales y específicos de estudiantes por semestres, especialidad y paralelos.
Dirección Académica	<p>Necesita conocer listas, reportes y estadísticas de estudiantes, docentes y administrativos, como ser:</p> <ul style="list-style-type: none"> ✓ Realizar la gestión del plan de estudios por carreras. ✓ Realizar la planificación académica por semestre. ✓ Realizar consultas y emitir reportes generales y específicos de estudiantes por semestres, especialidad y paralelos ✓ Ver listas de estudiantes inscritos por semestres, especialidad y paralelo. ✓ Emitir reporte de datos personales de los estudiantes (edad, promedio mínimo y máxima, sexo entre otros.).
Secretaria	<p>Encargada del manejo de la información académica quien necesita:</p> <ul style="list-style-type: none"> ✓ Información detallada de los estudiantes. ✓ Emitir la Hoja de Registro Personal después del proceso de Registro de estudiantes nuevos. ✓ Verificación automática de materias a cursar y materias de arrastre de un estudiante. ✓ Emitir boletas de inscripción y matriculación. ✓ Emitir listas oficiales de estudiantes inscritos por semestres y paralelo. ✓ Emitir notas semestrales, récord e historial académico. ✓ Actualización de documentos. ✓ Realizar la difusión de la información con referente a la Institución (carreras, turnos, horarios).
Docente	<p>Profesional que necesita:</p> <ul style="list-style-type: none"> ✓ Realizar el registro de notas de estudiantes inscritos en su materia. ✓ Imprimir boletines de notas. ✓ Listado de los estudiantes inscritos en su materia.
Estudiante	<p>Usuario Indirecto quien necesita:</p> <ul style="list-style-type: none"> ✓ Conocer sus datos de Registro Personal. ✓ Realizar su inscripción. ✓ Obtener la Boleta de inscripción. ✓ Conocer Materias Inscritas y paralelos. ✓ Obtener Historial de Notas. ✓ Obtener Récord Académico.

Fuente: Elaboración Propia

3.3. Aplicación del Modelo UWE

3.3.1. Modelo de Casos de Uso

3.3.1.1. 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 Instituto. 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.

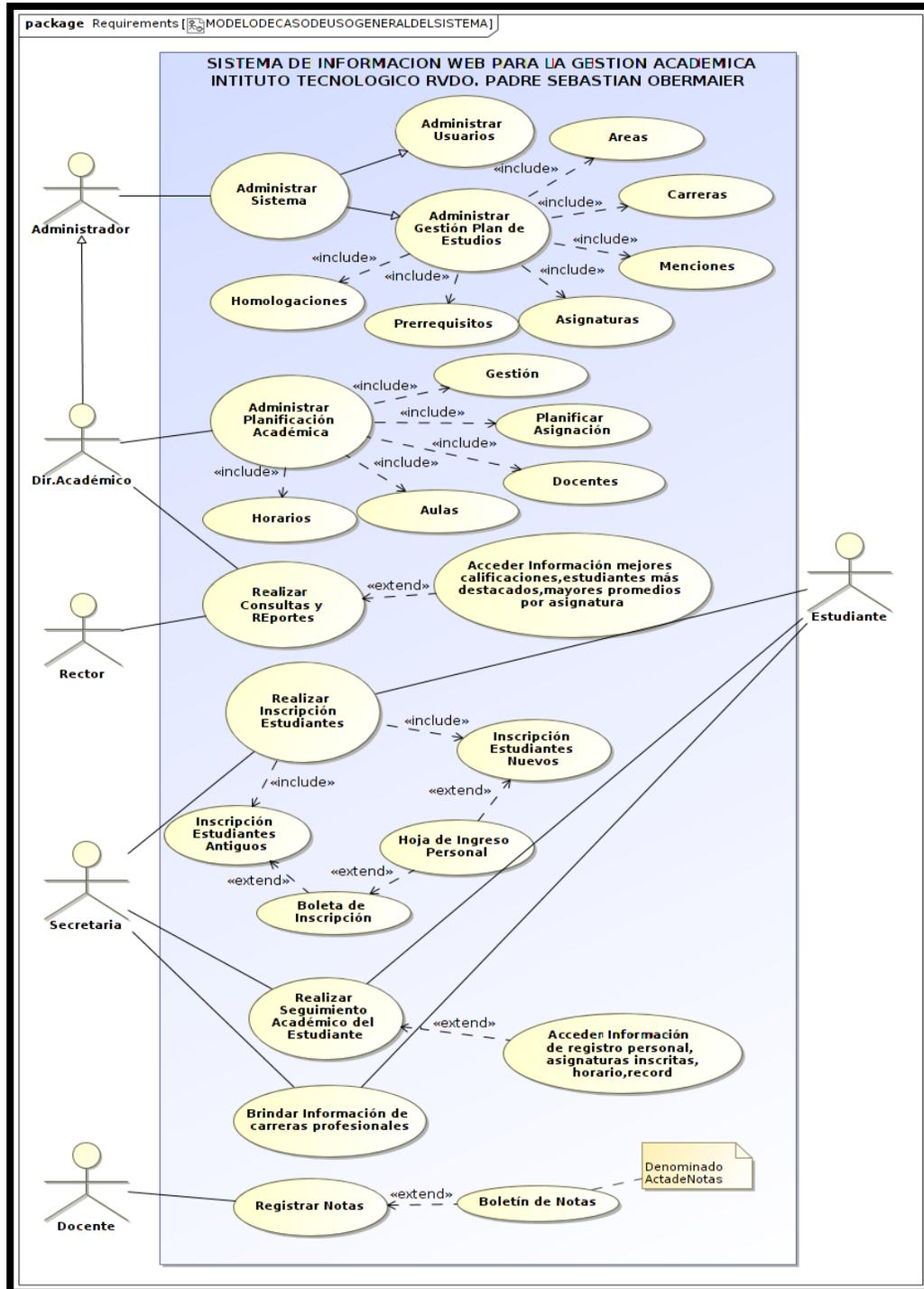
3.3.1.1.1. Diagrama de casos de uso comercial

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 del Plan de Estudios**
- **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**
- **Diagrama de caso de uso: Brindar Información**

3.3.1.1.2. Modelo de Caso de Uso General del Sistema

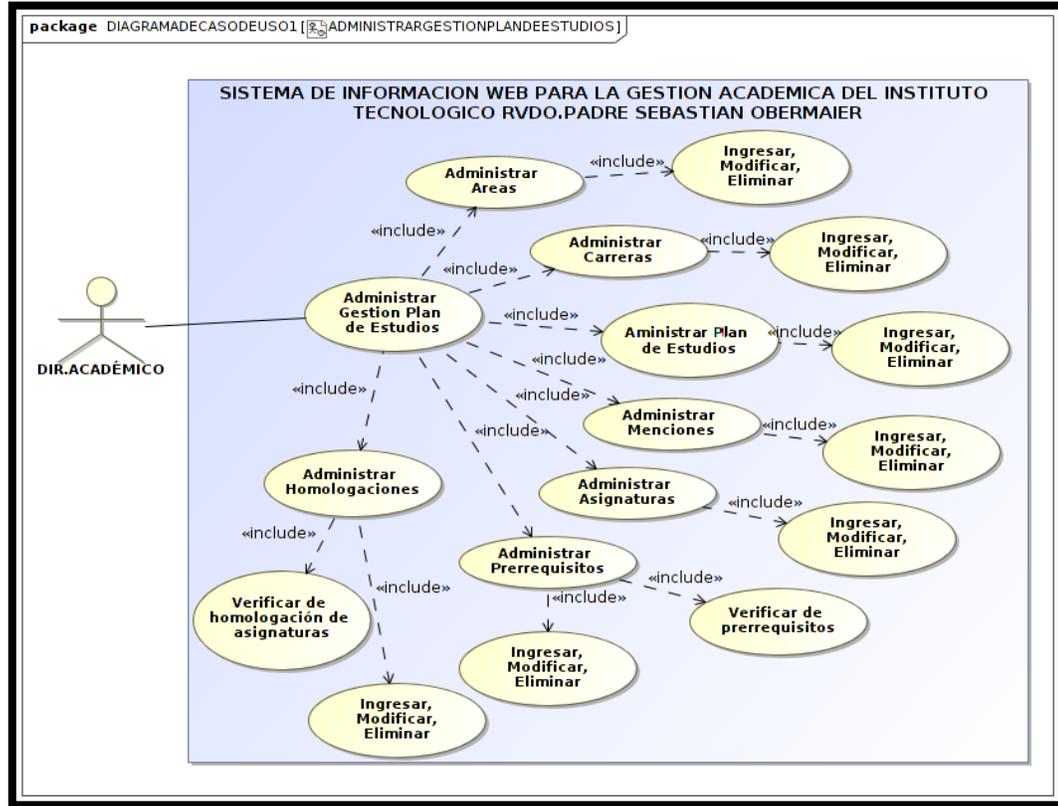
Figura Nº 3. 2: Modelo de Caso de Uso General del Sistema



Fuente: Elaboración Propia

✓ **Diagrama de caso de uso: Gestión del Plan de Estudios**

Figura Nº 3. 3: Diagrama de Caso de Uso: “GESTIÓN DEL PLAN DE ESTUDIOS”



Fuente: Elaboración Propia

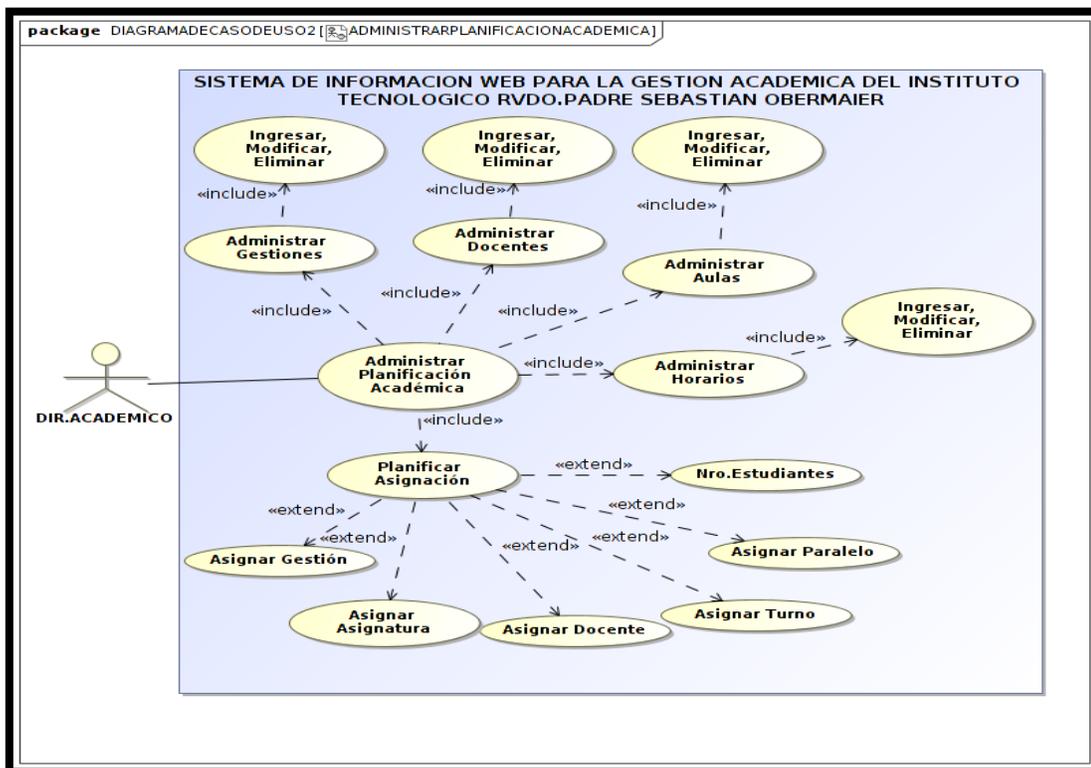
Tabla Nº 3. 5: Caso de Uso “GESTIÓN DEL PLAN DE ESTUDIOS”

CASO DE USO:	GESTIÓN DEL PLAN DE ESTUDIOS
ACTOR:	Dir. Académico
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Dir. Académico ingresa al sistema mediante su usuario, contraseña asignado, accede al módulo correspondiente y realiza la administración de Gestión del Plan de Estudios, estableciendo CRUD (crear, leer, actualizar y borrar) de áreas, carreras, menciones, plan de estudios, asignaturas, prerrequisitos y homologaciones correspondientes a una carrera.

Fuente: Elaboración Propia

✓ Diagrama de caso de uso: Planificación Académica

Figura N° 3. 4: Diagrama de Caso de Uso: “PLANIFICACIÓN ACADÉMICA”



Fuente: Elaboración Propia

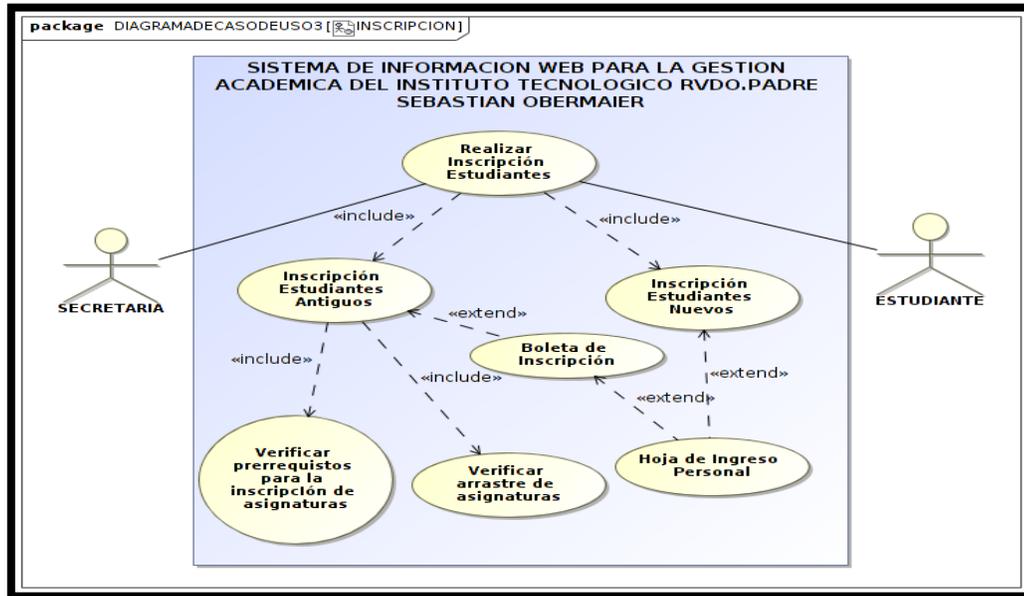
Tabla N° 3. 6: Caso de Uso “PLANIFICACIÓN ACADÉMICA”

CASO DE USO:	PLANIFICACIÓN ACADÉMICA
ACTOR:	Dir. Académico
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Dir. Académico 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, docentes, aulas, horarios, de la misma manera procede a planificar la asignación registrando datos como docente, asignatura, turno, paralelo, número de estudiantes, entre otros, correspondientes a un plan de estudios, periodo y gestión vigente.

Fuente: Elaboración Propia

✓ Diagrama de caso de uso: Inscripción

Figura N° 3. 5: Diagrama de Caso de Uso: “INSCRIPCIÓN”



Fuente: Elaboración Propia

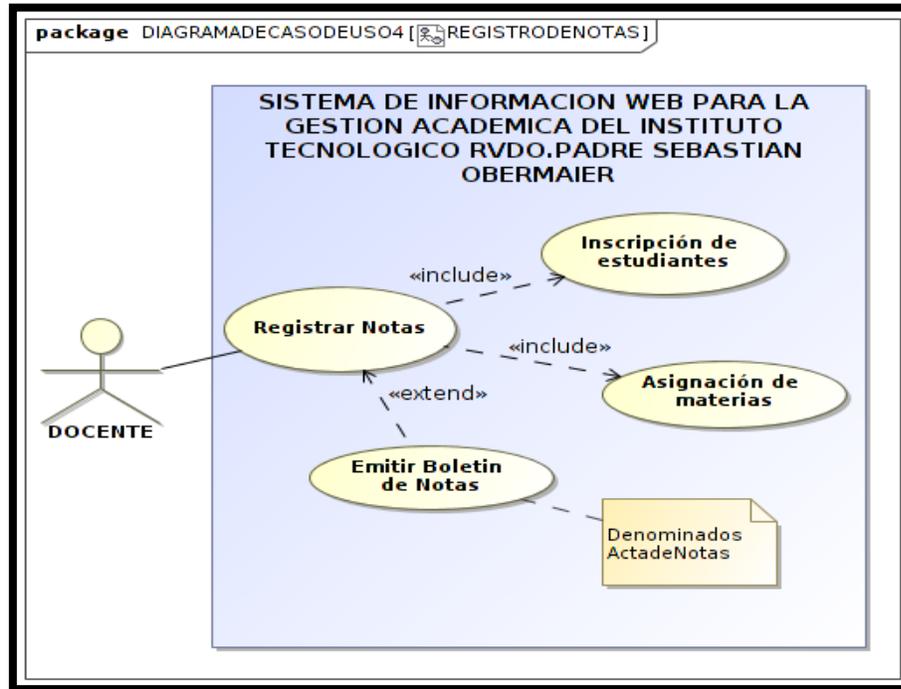
Tabla N° 3. 7: Caso de Uso “INSCRIPCIÓN”

CASO DE USO:	INCRIPCIÓN
ACTOR:	Estudiante, Secretaria
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>La secretaria 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 el instituto, así mismo procede a la impresión de la misma.</p> <p>Estudiante Antigo, 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. 6: Diagrama de Caso de Uso: “REGISTRO DE NOTAS”



Fuente: Elaboración Propia

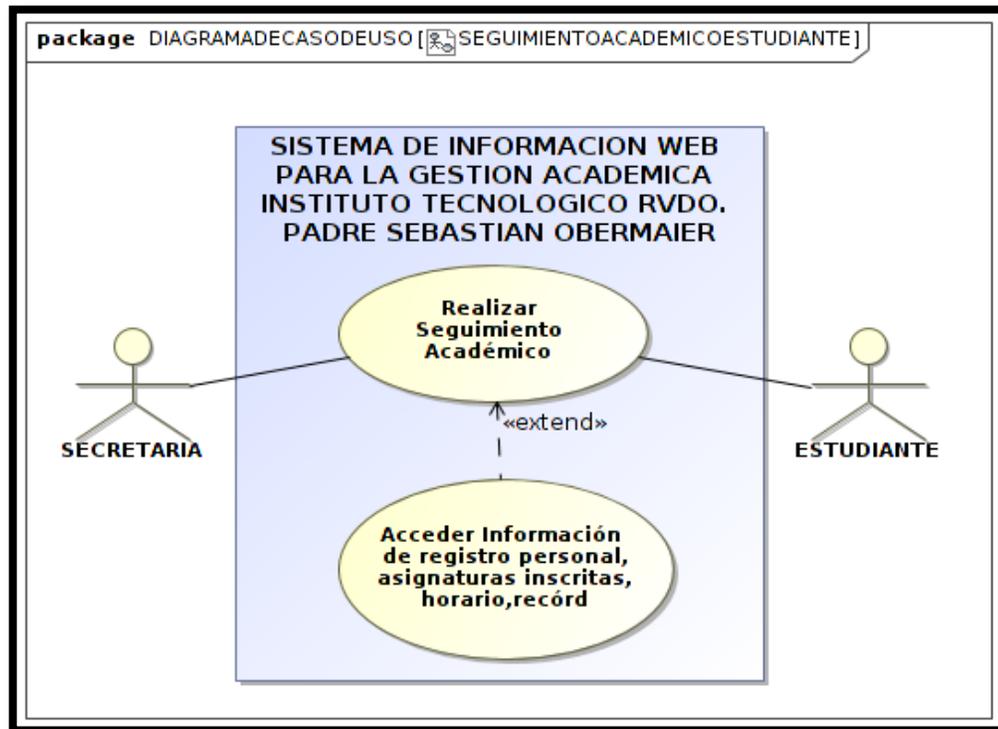
Tabla Nº 3. 8: Caso de Uso “REGISTRO DE NOTAS”

CASO DE USO:	REGISTRO DE NOTAS
ACTOR:	Docente
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Docente ingresa al sistema mediante su usuario, contraseña asignado, accede al módulo Registro de Notas, el mismo realiza el llenado de datos correspondientes y confirma su centralización.

Fuente: Elaboración Propia

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

Figura Nº 3. 7: Diagrama de Caso de Uso: "SEGUIMIENTO ACADEMICO DE ESTUDIANTE"



Fuente: Elaboración Propia

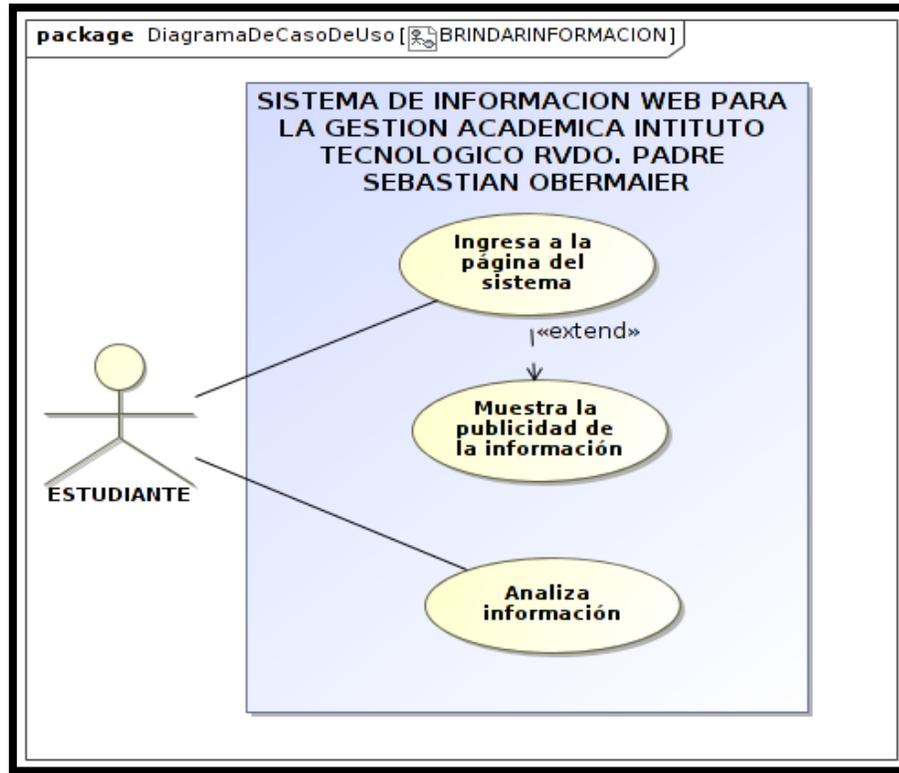
Tabla Nº 3. 9: Caso de Uso "SEGUIMIENTO ACADÉMICO DE ESTUDIANTE"

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

Fuente: Elaboración Propia

✓ **Diagrama de caso de uso: Brindar Información**

Figura N° 3. 8: Diagrama de Caso de Uso: "BRINDAR INFORMACIÓN"



Fuente: Elaboración Propia

Tabla N° 3. 10: Caso de Uso "BRINDAR INFORMACION"

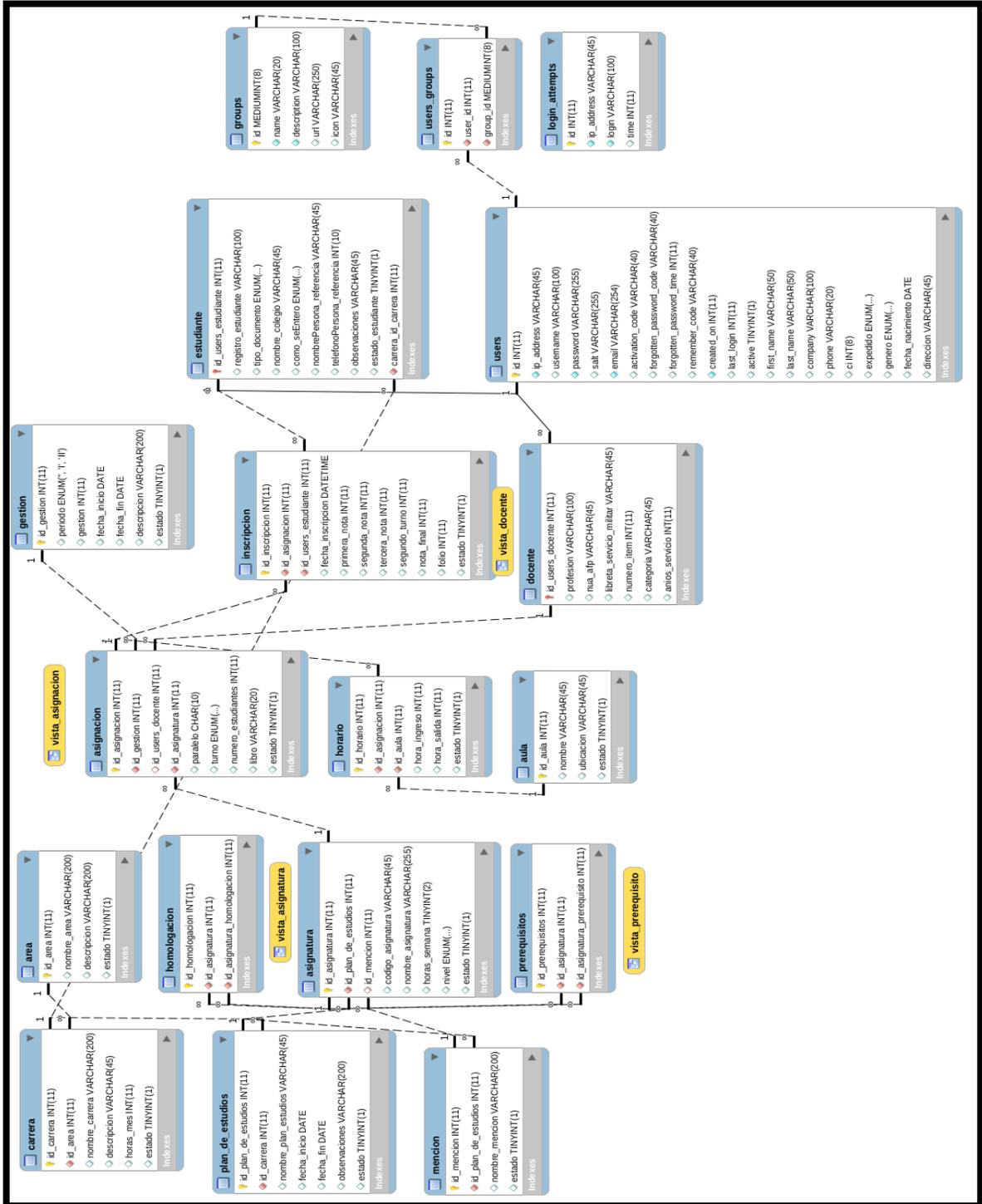
CASO DE USO:	BRINDAR INFORMACION
ACTOR:	Estudiante
TIPO:	Primario Esencial
DESCRIPCIÓN:	Este caso de uso comienza cuando el Estudiante ingresa al Sitio Web para indagar la publicidad que ofrece la Institución.

Fuente: Elaboración Propia

3.3.2. Modelo de contenido

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

Figura N° 3. 9: Modelo Conceptual



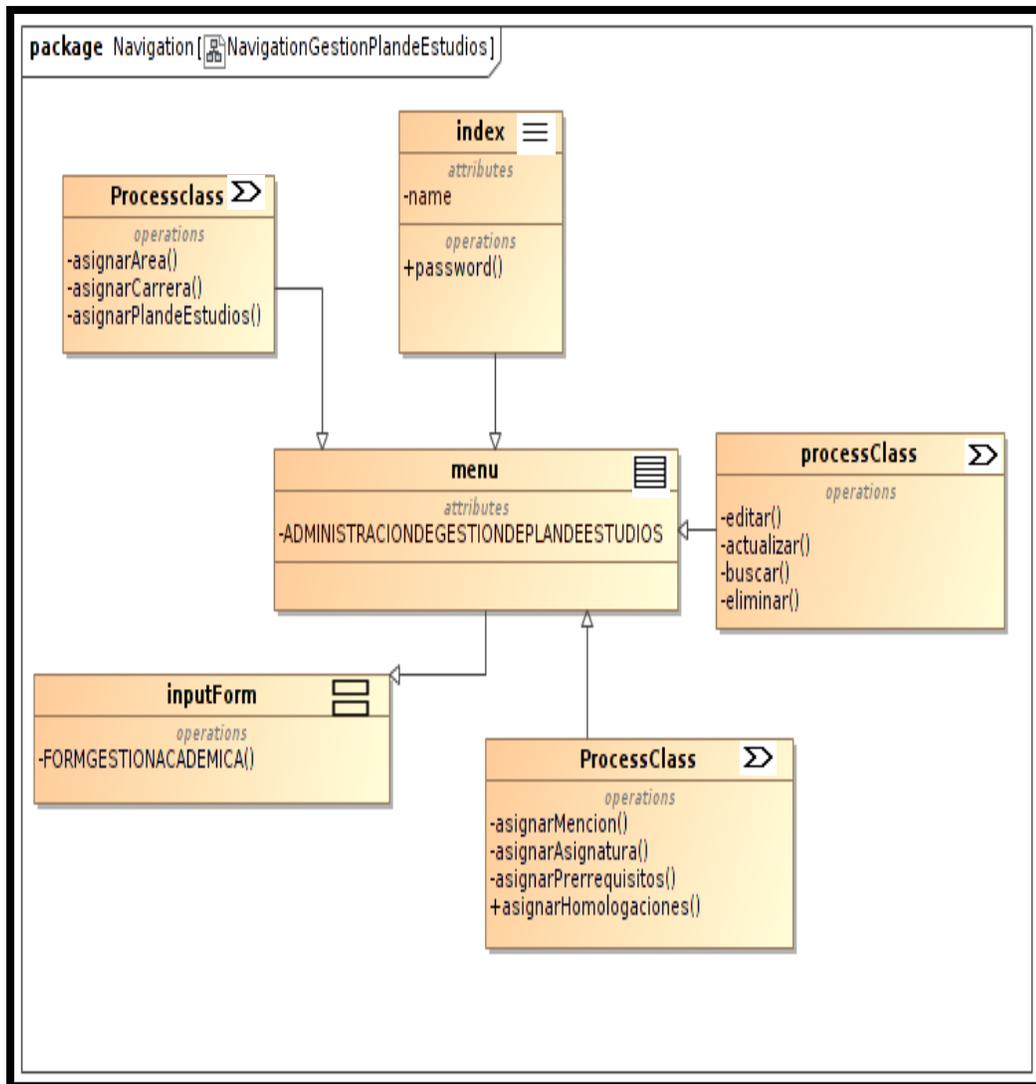
Fuente: Elaboración Propia

3.3.3. 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 Plan de Estudios

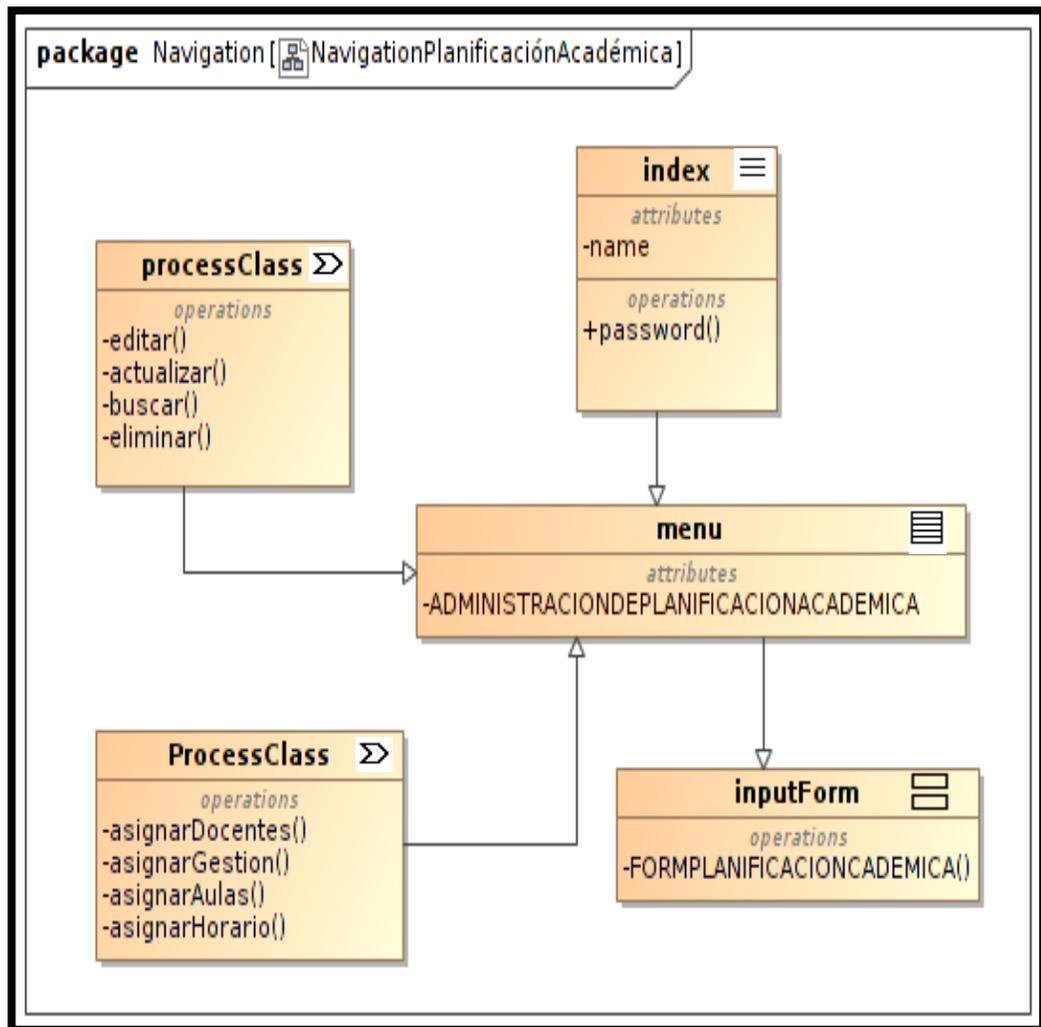
Figura Nº 3. 10: Modelo de Navegación Gestión Plan de Estudios



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Planificación Académica**

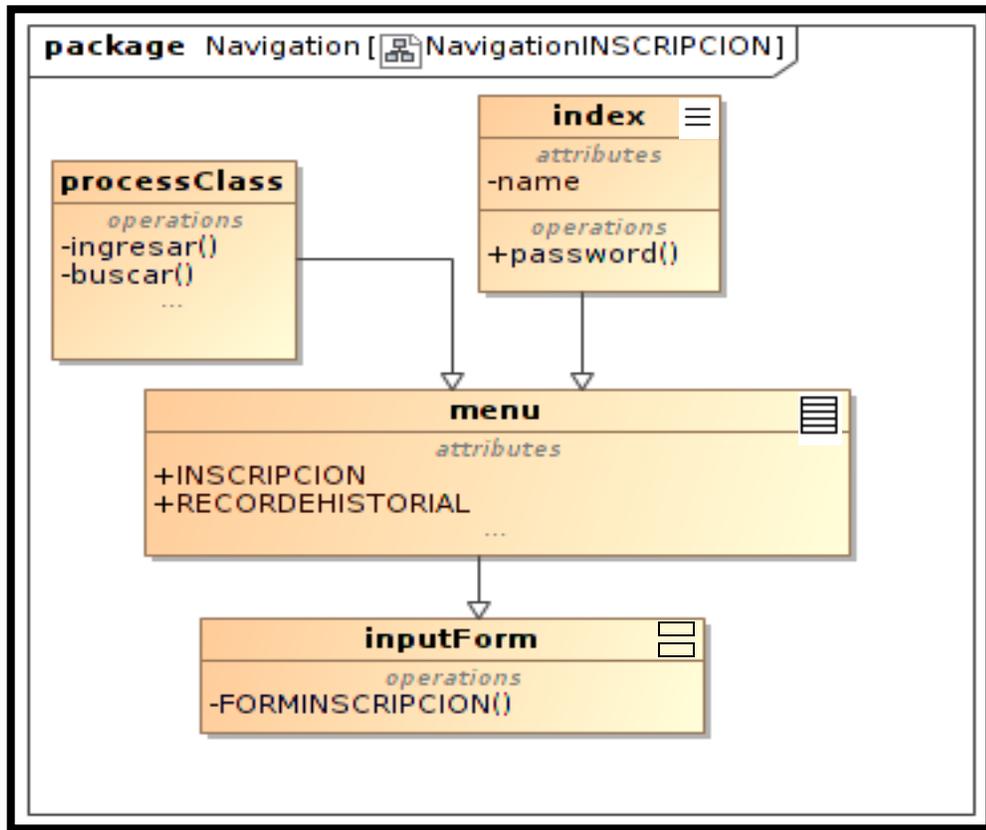
Figura Nº 3. 11: Modelo de Navegación Planificación Académica



Fuente: Elaboración Propia

✓ **Modelo de Navegación: Inscripción**

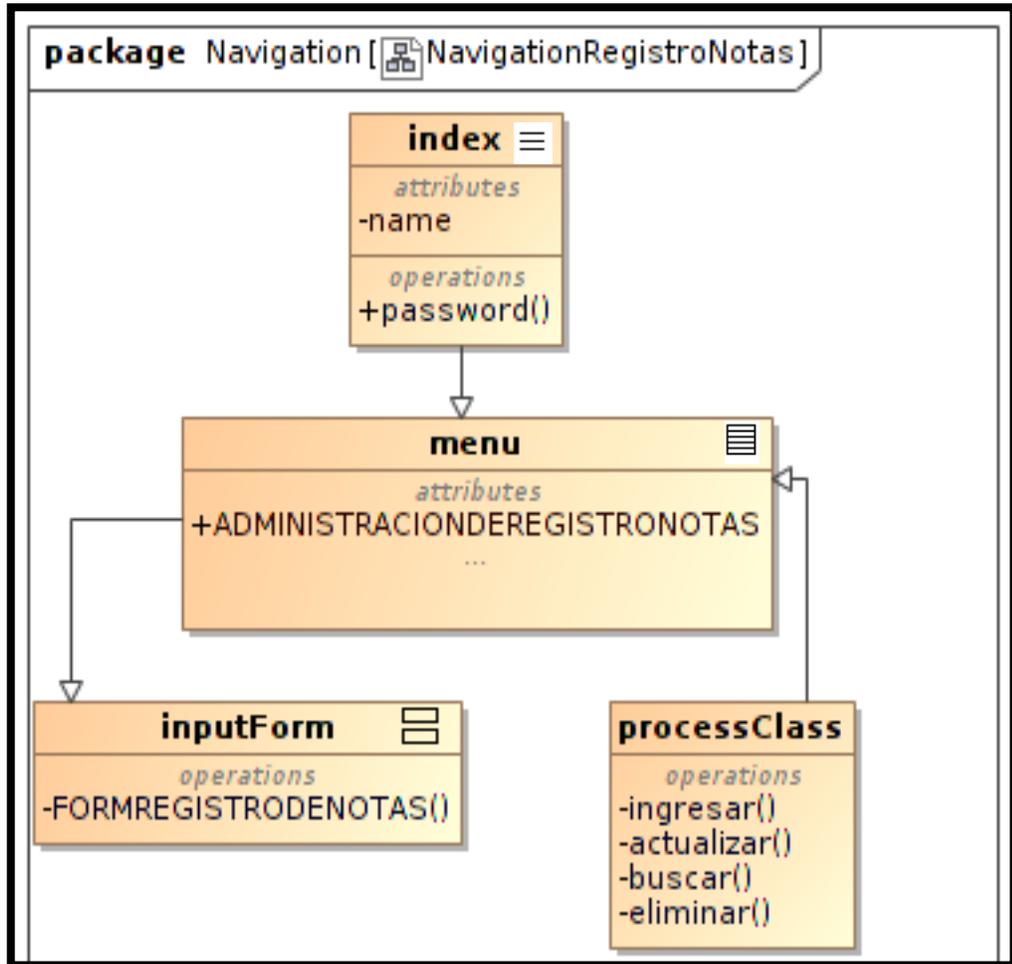
Figura N° 3. 12: Modelo de Navegación Inscripción



Fuente: Elaboración Propia

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

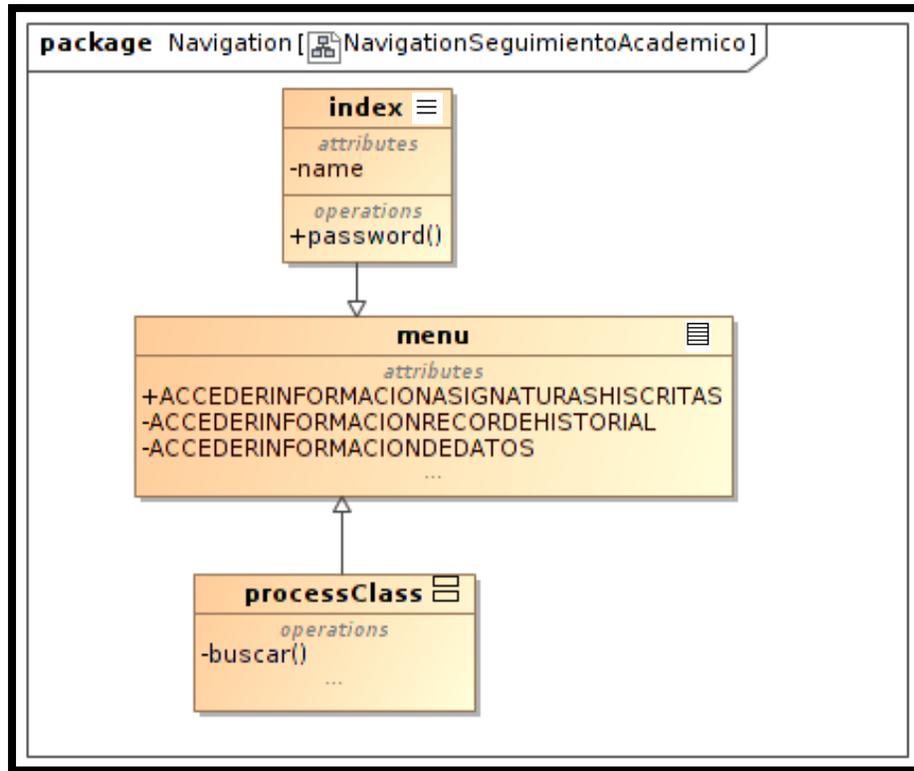
Figura Nº 3. 13: Modelo de Navegación Registro de Notas



Fuente: Elaboración Propia

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

Figura N° 3. 14: Modelo de Navegación Seguimiento Académico de Estudiante



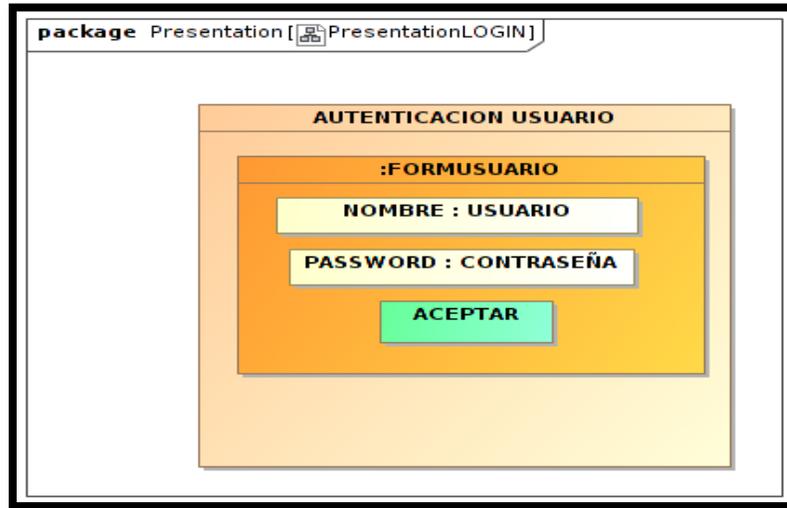
Fuente: Elaboración Propia

3.3.4. 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: AUTENTICACION**

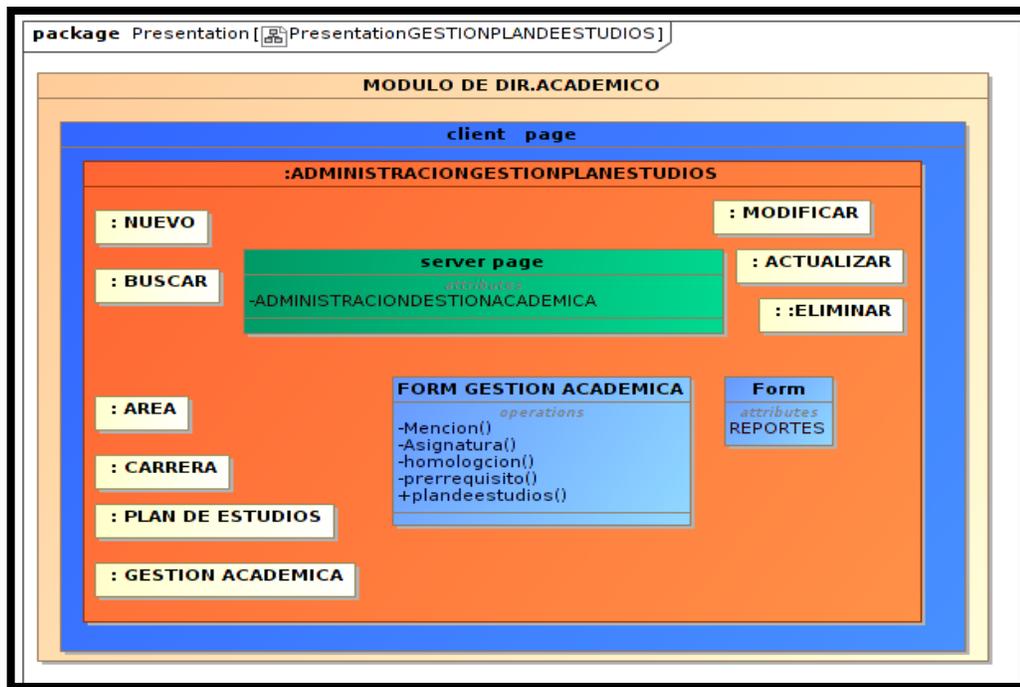
Figura Nº 3. 15: Modelo de Presentación Autenticación



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Gestión del Plan de Estudios**

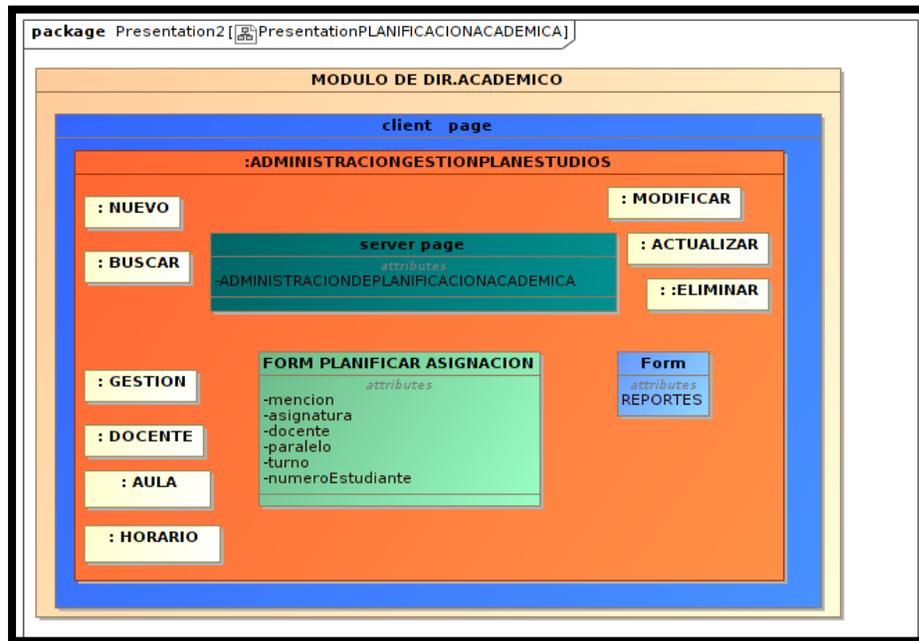
Figura Nº 3. 16: Modelo de Presentación Gestión del Plan de Estudios



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Planificación Académica**

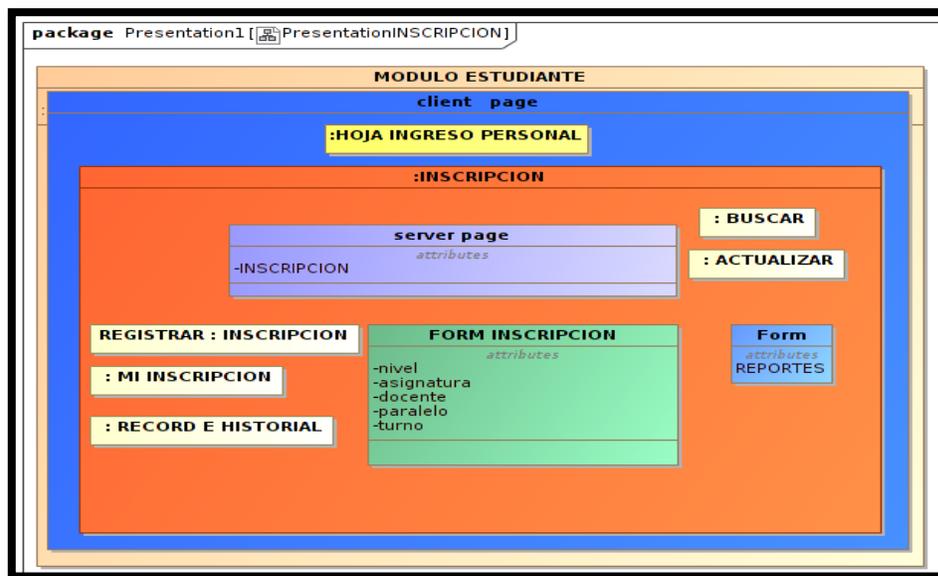
Figura Nº 3. 17: Modelo de Presentación Planificación Académica



Fuente: Elaboración Propia

✓ **Modelo de Presentación: Inscripción**

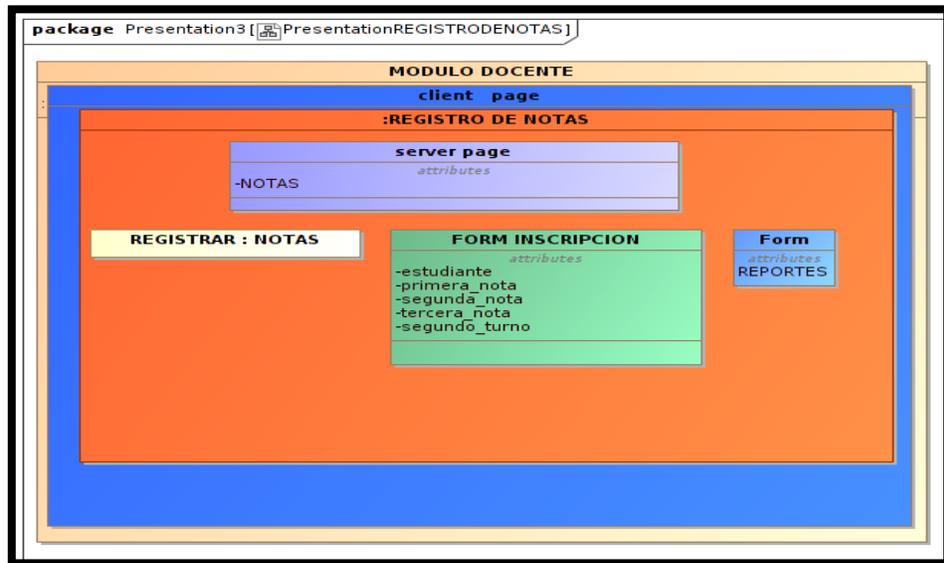
Figura Nº 3. 18: Modelo de Presentación Inscripción



Fuente: Elaboración Propia

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

Figura N° 3. 19: Modelo de Presentación Registro de Notas



Fuente: Elaboración Propia

3.3.5. 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**

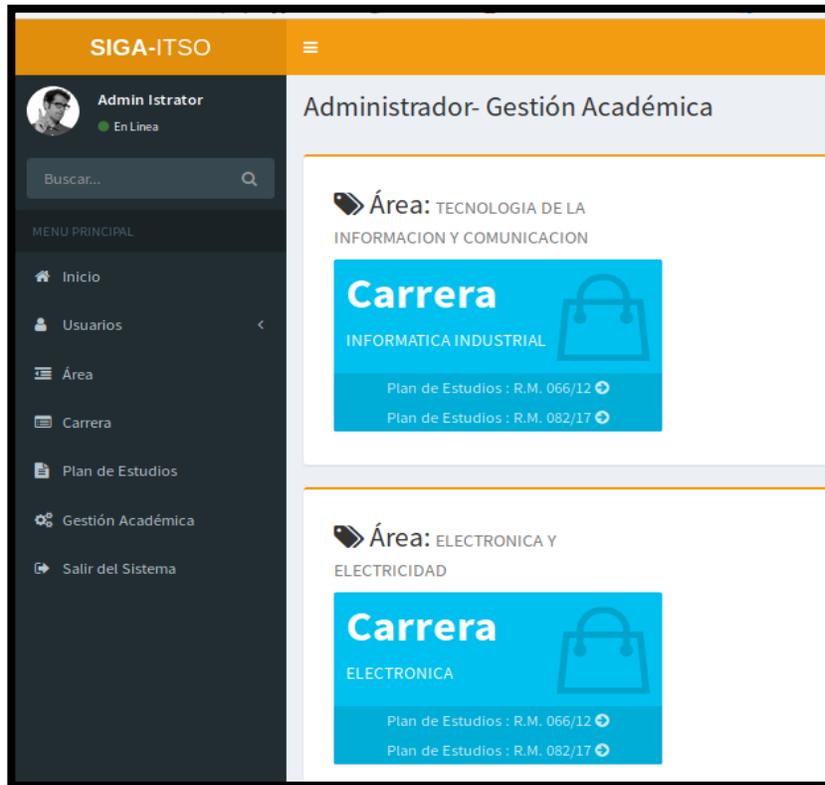
Figura N° 3. 20: Autenticación inicial con el sistema



Fuente: Elaboración Propia

✓ **Interfaz de Gestión del Plan de Estudios**

Figura Nº 3. 21: Interfaz de Gestión del Plan de Estudios (Administrador)



Fuente: Elaboración Propia

✓ **Interfaz de Planificación Académica**

Figura Nº 3. 22: Interfaz de Planificación Académica (Dir. Académico)



Fuente: Elaboración Propia

✓ Interfaz de Inscripción

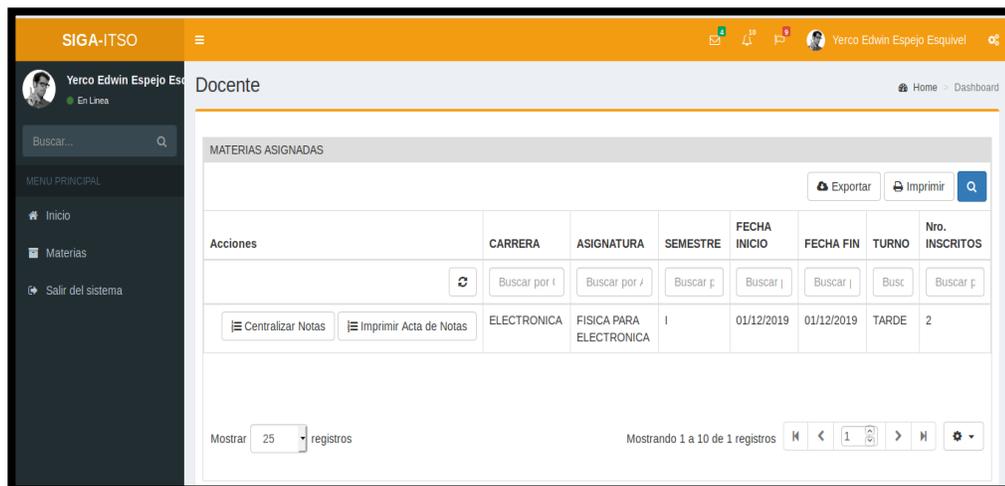
Figura N° 3. 23: Interfaz de Inscripción (Estudiante)



Fuente: Elaboración Propia

✓ Interfaz de Registro de Notas

Figura N° 3. 24: Interfaz de Registro de Notas (Docente)



Fuente: Elaboración Propia

✓ Interfaz de Sitio web

Figura Nº 3. 25: Interfaz de Sitio web



Fuente: Elaboración Propia

3.4. Métricas de Calidad de Software

La calidad del software nos permite medir, supervisar y mejorar el proceso para alcanzar la calidad del software. Existen varias métricas de medición que nos permiten medir la calidad del software, uno de ellos 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**.

3.4.1. 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.

3.4.1.1. 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. 11: Número de Entradas Externas

Nº	ENTRADAS EXTERNAS	CANTIDAD
1	Modulo Administrador	4
2	Módulo Gestión de Plan de estudios	11
3	Módulo Planificación Académica	8
4	Módulo Inscripción	3
5	Módulo Registro de Notas	1
Total		27

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. 12: Número de Salidas Externas

Nº	SALIDAS EXTERNAS	CANTIDAD
1	Modulo Administrador	8
2	Módulo Gestión de Plan de estudios	16
3	Módulo Planificación Académica	7
4	Módulo Inscripción	5
5	Módulo Registro de Notas	3
Total		39

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. 13: Número de Consultas Externas

Nº	CONSULTAS EXTERNAS	CANTIDAD
1	Modulo Administrador	4
2	Módulo Gestión de Plan de estudios	8
3	Módulo Planificación Académica	6
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 Nº 3. 14: Número de Archivos Lógicos Internos

Nº	ARCHIVOS LÓGICOS INTERNOS	CANTIDAD
1	Modulo Administrador	4
2	Módulo Gestión de Plan de estudios	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 Nº 3. 15: 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 Nº 3. 16: Conteo Total

PARÁMETRO	CONTEO	FACTOR	SUBTOTAL
Número de entradas externas (EE)	27	4	108
Número de salidas externas (SE)	39	5	195
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			589

Fuente: Elaboración Propia

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

Tabla Nº 3. 17: 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. 18: 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).

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

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

$$PF_{\text{real}} = 589 * [0.65 + (0.01 * 49)]$$

$$PF_{\text{real}} = 671,46$$

Ahora calcularemos a un nivel de confianza del 100%.

$$PF_{\text{esperado}} = 589 * [1 + (0.01 * 49)]$$

$$PF_{\text{esperado}} = 877,61$$

El porcentaje de funcionalidad será:

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

$$\% PF = 671,46 / 877,61$$

$$\% PF = 0,7651 * 100$$

$$\% PF = 77\%$$

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.

3.4.1.2. 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:

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

Sean los datos:

$$PF = 77\%$$

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

$$t = 20 \text{ días}$$

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

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

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

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.

3.4.1.3. 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 cuestionario y rango de valores para la evaluación, según la función:

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

Dónde:

X_i=Sumatoria de valores

N= Número de preguntas

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

Tabla N° 3. 19: Rangos para Evaluar la Usabilidad

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

Fuente: Elaboración Propia

Tabla N° 3. 20: Cuestionario 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 llego 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.

3.4.1.4. 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 Nº 3. 21: 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 apreciaciones, tal como se muestra en la tabla 3.22.

Tabla Nº 3. 22: 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.

3.4.1.5. Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la Institución 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:

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

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:

$$M_T = 6$$

$$F_c = 0$$

$$F_a = 1$$

$$F_d = 0$$

Reemplazando tenemos:

$$IMS = [6 - (1 + 0 + 0)] / 6$$

$$IMS = 0.83$$

Interpretando podemos decir que el nuevo sistema tiene una estabilidad de **83%** que es la facilidad de mantenimiento, el **17%** 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.

3.4.1.5.1. Mantenimiento Correctivo

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

Tabla N° 3. 23: 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.

3.4.1.5.2. Mantenimiento de Mejora

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

Tabla N° 3. 24: 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.

3.4.1.6. 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 N° 3. 25: 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 N° 3. 26: Valoración para la 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.

3.4.1.7. Resultado Final

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

Tabla N° 3. 27: Resultados de las Características de Calidad

ESCALA	VALOR
Funcionalidad	77%,
Confiabilidad	90%
Usabilidad	86 %
Eficiencia	88%
Mantenibilidad	83%
Portabilidad	85%
PROMEDIO	84,83%

Fuente: Elaboración Propia

Se puede observar que el resultado de grado de calidad del software es **84,83%**, satisfacción que el usuario tendrá al momento de utilizar el sistema.

3.5. 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.

3.5.1.1. 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.

3.5.1.1.1. Estimación del punto función

a) Calculo del punto función (PF)

Tabla Nº 3. 28: Valores del dominio de Información

PARAMETRO	CUENTA	FACTOR DE PONDERACION			SUB-TOTAL
		SIMPLE	MEDIO	COMPLEJO	
Número de entradas externa (EE)	27	3	4	6	108
Número de salidas externas (SE)	39	4	5	7	195
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					589

Fuente: Elaboración Propia

b) Cálculo 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 Nº 3. 29: Valores de ajustes de la 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
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:

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

$$\text{PF} = 589 * 1.14$$

$$\text{PF} = 671,46$$

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. 30: 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:

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

$$\text{LDC} = 671,46 * 29$$

$$\text{LDC} = 19472,34$$

Las líneas de código en su totalidad son **19472,34**, entonces el número estimado de líneas de código distribuidas en miles es:

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

$$KLDC=19472,34/1000$$

$$KLDC=19,47$$

Por tanto existen **19,47** 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 Nº 3. 31: Ecuaciones del modelo intermedio que define COCOMO

VARIABLE	ECUACION	TIPO/UNIDAD
Esfuerzo requerido por el proyecto	$E=a*(KLCD)^b*FAE$	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 **19,47**. 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. 32: Aplicación del Modelo Intermedio

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semiacoplado	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. 33: 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:

$$\text{FAE} = 15 * 1,00 * 1,15 * 1,00 * 1,00 * 1,00 * 1,00 * 0,86 * 0,82 * 0,86 * 1,00 * 0,95 * 0,91 * 1,00 * 1,04$$

$$\text{FAE} = 0,72$$

Aplicando y reemplazando valores a las fórmulas, se tiene:

✓ **Cálculo Esfuerzo**

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

$$E = 2,4 * (19,47)^{1,05} * 0,72$$

$$E = 39,03 \text{ (Personas/mes)}$$

✓ **Cálculo del Tiempo**

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

$$T = 2,5 * 39,03^{0,38}$$

$$T = 10,06 \text{ (Meses)}$$

✓ **Cálculo de la Productividad**

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

$$\text{PR} = 19472,34 / 39,03 \text{ (Meses)}$$

$$\text{PR} = 498,90 \text{ (LCD/Personas Mes)}$$

✓ **Cálculo del Personal Requerido**

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

$$P = 39,03 / 10,06 \text{ (Personas)}$$

$$P = 3,88 \text{ (Personas)}$$

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

✓ **Costo Total del Proyecto**

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

$$\text{Costo Total} = P * \text{Salario medio entre los programadores y analistas}$$

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 y recomendaciones referentes al presente proyecto de Grado.

4.1. Conclusiones

Se concluyó con los objetivos planteados en el presente proyecto, se desarrolla un Sistema de Información para la Gestión Académica del Instituto tecnológico Rvdo. Padre Sebastián Obermaier, el cual cumple con las expectativas de la institución, logrando cumplir los objetivos establecidos inicialmente en el capítulo I.

Se lograron cumplir con los siguientes objetivos específicos:

- Se logró desarrollar un módulo del plan de estudios, para dar estricto cumplimiento de los prerrequisitos establecidos.
- .Se consiguió realizar un módulo de planificación académica, inscripción de estudiantes y registro de notas.
- Posee con consulta de estudiantes permitiendo realizar su seguimiento académico (asignaturas inscritas, récord, historial de notas y registro personal).
- Cuenta con los reportes necesarios para una toma de decisiones en la información correspondiente.
- Se alcanzó a diseñar una página web para el instituto, para la difusión de sus carreras, cursos, actividades y otros.
- 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 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 institución.
- Realizar copias de seguridad de la base de datos (Backup), así mismo que la información publicada en el sitio web sea actualizada constantemente, brindando mayor satisfacción al usuario.
- Tomar en cuenta para un futuro la implementación de un módulo de pagos de inscripción en línea mediante la web, es decir el pago de matrículas al momento de la inscripción se haga mediante vía online, tomando en cuenta la seguridad que conlleva el mismo.
- En cuanto al análisis y diseño del sistema, cuando la institución 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

- Pressman, R. (2005). Ingeniería de Software, Quinta Edición. México: Mc.Graw Hill.
- Vargas, M. S. (25 de junio de 2017). Ingeniería guiada por modelos. Obtenido de Ingeniería Web Basada en UML: marcelosalasvargas.blogspot.com/2017
- Zeballos, D. (19 de Febrero de 2017). Tutorial UWE. Obtenido de Tutorial sobre UWE: es.scribd.com/document/tutorial-UWE
- Boehm, B. (1981). Software Engineering Economics. Pearson: Prentice Hall.
- Cespedes, J. (2001). Metodología de la Investigación. La Paz: Universidad Técnica de Oruro.
- Galiano, L. (3 de 11 de 2012). Metodología UWE. Recuperado el 08 de 04 de 2018, de Metodología UWE aplicada a la solución Informática: elproyectedeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html
- GitBook. (s.f.). Recuperado el 10 de 04 de 2018, de Bootstrap: www.ajgallego.gitbook.io/bootstrap-3/content/capitulo_frameworks.html
- Gomez, M. (s.f.). CodeIgniter en Español. Recuperado el 10 de 04 de 2018, de CodeIgniter: www.codeigniter.es
- Guioteca. (2011). HTML5. Recuperado el 11 de 04 de 2018, de www.guioteca.com/internet/¿que-es-html5-y-que-cambios-introduce/
- Olsina, L. (2 de 11 de 1999). Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web. Recuperado el 03 de 04 de 2018, de Ingeniería de Software en la Web: www.dsi.uclm.es/personal/FranciscoMSimarro/cedasi/Olsina-Tesis.pdf
- Rukbottoland. (21 de 04 de 2016). Rukbottoland;. Recuperado el 11 de 04 de 2018, de Diseño de Interfaces Web con React.js: www.rukbottoland.com/blog/disenio-de-interfaces-web-con-react-js/

- Sólida, T. (12 de 11 de 2017). Comparación de propiedades de SGBD MariaDB VS PostgreSQL. Recuperado el 08 de 04 de 2018, de DB-ENGINES: db-engines.com/en/system/MariaDB%3BPostgreSQL
- Strato, A. (2016). Recuperado el 12 de 04 de 2018, de WordPress: www.strato.es/wordpress-hosting/cms-wordpress/
- Villalobos, J. (05 de 09 de 2010). Comparación PHP.JAVA,ASP.NET. Recuperado el 01 de 04 de 2018, de Código Programación: codigoprogramacion.com/programacionweb/comparacion-php-jsp-asp.html

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: Jorge Villalobos