

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**“SISTEMA DE CARÁCTER ADAPTABLE EN EL AREA ACADÉMICA BASADA  
EN LA METODOLOGÍA ÁGIL ASD BAJO EL ENTORNO WEB “  
CASO: FUNDACIÓN ECOLOGÍA Y DESARROLLO**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA  
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

**AUTOR: Juan Carlos Torrez Laura  
TUTOR: Lic. Efraín Silva Sánchez  
REVISOR: Lic. Roberto Vargas Blacut  
ASESOR: Lic. Jhonny Roberto Felipez Andrade**

**PAZ - BOLIVIA**

**2006**

## **Dedicatoria**

A Dios por iluminarme y  
acompañarme en todo  
momento de mi vida.

A mi querida familia por su  
por su apoyo incondicional para el  
logro de mis metas

A mis queridos abuelitos  
Carmen Condori y Eduardo Torrez  
que están en la gloria de Dios

## **AGRADECIMIENTOS**

Deseo expresar mi agradecimiento en primer lugar a Dios padre que esta en el cielo, porque sin el nada de esto seria realidad.

A mi Tutor Lic. Efraín Silva Sánchez, por confiar en mi persona y en este proyecto, dando sugerencias y enseñanzas enriquecedoras, guía y apoyo constante a mi tarea.

A mi Revisor Lic. Roberto Vargas Blacut, por su disponibilidad en el proceso de revisión del proyecto, por las correcciones, observaciones y sugerencias durante el transcurso del desarrollo del presente proyecto.

A mi Asesor Lic Jhonny Roberto Felipez Andrade, quien me brindo su tiempo generosamente en las observaciones y correcciones de mi proyecto, sobre todo le doy gracias por el gran apoyo moral y académico que el significa para mi persona.

Al Lic. Simón Quisbert Director General de la Fundación Ecología y Desarrollo por darme la oportunidad de aprender y trabajar en el desarrollo del presente proyecto, por su apoyo moral y gran colaboración para seguir adelante en mi proyecto.

A mi familia por su apoyo y comprensión, a mi tío Fidel Torrez por ser la sombra que guía mis pasos desde siempre, a mi abuelita Juana López que siempre esta ahí para apoyarme y orientarme.

A Hermana Sara, Margarita, Mercedes, quienes siempre me han demostrado su aprecio y apoyo espiritual en todas mis actividades.

A mis compañeros de la carrera en especial a mi tan apreciada Claudia Apaza, por acompañarme en esta etapa tan hermosa, que es la vida universitaria.

## RESUMEN

La Fundación Ecología y Desarrollo brinda aportes tecnológicos de información y comunicación a varias instituciones, en el presente proyecto la fundación trabaja con cinco Unidades Educativas, en las cuales se trabajó en el área académica de estas, puesto que el crecimiento de la población estudiantil esta en incremento, por ello el manejo de la información ya no es la adecuada, además se pretende incorporar a estas Unidades Educativas en la nueva sociedad de la información.

En este sentido se ha desarrollado el Sistema de Seguimiento Académico en el entorno Web, para estas Unidades Educativas, realizando el mismo a través de la metodología ágil "ASD" (Desarrollo de Software Adaptable), de la cual se aprovechó el ciclo de vida que presenta, este ciclo de vida permite realizar iteraciones veloces con las que se ha logrado agilizar el trabajo y conseguir la adaptabilidad del sistema a las cinco Unidades Educativas. Como herramienta de apoyo se utilizo el Lenguaje de Modelado Unificado "UML". El Proyecto comprende 5 capítulos los cuales son:

Capitulo 1 Introducción, se describe brevemente los antecedentes, la situación de la problemática que tropiezan las Unidades Educativas, el cual constituyó la base para formular el problema principal, además se representan los objetivos específicos y el general.

Capitulo 2 Marco Teórico, se presentan y describen tanto las definiciones teóricas de la metodología, así como conceptos que se utilizaron en el presente proyecto.

Capitulo 3 Análisis y diseño, donde se realiza un análisis de la situación actual de los procesos académicos que llevan a cabo las unidades educativas para luego aplicar la metodología mencionada y realizar el diseño propuesto para llegar al objetivo trazado.

Capitulo 4 Calidad del Software, donde se realiza una serie de medidas de calidad del sistema a través del modelo Web-site QEM.

Finalmente se presenta el Capitulo 5 Conclusiones y Recomendaciones, de todo el trabajo realizado en el presente proyecto.

# ÍNDICE

<b>CONTENIDO</b>	<b>Pág.</b>
<b>CAPITULO 1 INTRODUCCIÓN</b>	
1.1 Introducción.....	1
1.2 Antecedentes.....	2
1.3 Situación Actual.....	4
1.4 Descripción de la Problemática.....	5
1.5 Lista de Problemas .....	5
1.6 Problema Principal.....	6
1.7 Objetivos.....	6
1.7.1 Objetivo General.....	6
1.7.2 Objetivos Específicos.....	6
1.8 Justificación.....	7
1.8.1 Justificación Técnica.....	8
1.8.2 Justificación Social.....	8
1.8.3 Justificación Económica.....	9
1.9 Metodología.....	9
1.10 Alcances.....	10
1.11 Aportes.....	11
<b>CAPITULO 2 MARCO TEÓRICO</b>	
2.1 Introducción.....	13
2.2 Ingeniería de software.....	14
2.3 El entorno Web.....	15
2.4 Consorcio World Wide Web.....	15
2.5 Arquitectura Web.....	16
2.6 Definición de Internet.....	17
2.7 La Adaptabilidad.....	19

2.8	El aporte de una metodología de desarrollo de software.....	19
2.9	Metodologías ágiles.....	20
2.10	Metodologías ágiles vs. Metodologías tradicionales.....	23
2.11	Desarrollo de Aplicaciones Web con procesos ágiles.....	24
2.12	ASD – Adaptive Software Development.....	26
2.12.1	Fase de Especulación.....	26
2.12.2	Fase de Colaboración.....	27
2.12.3	Fase de Aprendizaje.....	28
2.13	Sistema de base de datos.....	31
2.13.1	Base de Datos.....	31
2.13.2	Los sistemas de Gestión de BD.....	32
2.14	Modelo de Base de Datos Relacional.....	32
2.15	Diagrama entidad-relación (E-R).....	34
2.16	Lenguaje de modelado unificado.....	36
2.16.1	Diagrama de Casos de Uso.....	37
2.16.2	Diagrama de Secuencia.....	40
2.16.3	Diagrama de Colaboración.....	40
2.16.4	Diagrama de Clases.....	40
2.17	Herramientas de construcción de software.....	42
2.17.1	PHP - Hypertext Processor.....	42
2.17.2	My SQL.....	45
2.17.3	Servidor Apache HTTP.....	47
2.18	Seguridad.....	47
2.18.1	Criptografía.....	47
2.19	Evaluación de calidad de sitios Web ( Web-Site QEM ).....	49
2.19.1	Estructura de Agregación de Preferencias Parciales.....	54
2.19.2	Preferencias Elementales.....	57
2.20	Estimación del Costo del Proyecto.....	58
2.20.1	Las Métricas Orientadas al tamaño.....	58
2.20.2	El Método COCOMO II.....	60

## **CAPITULO 3 ANÁLISIS Y DISEÑO**

3.1 Fase de Especulación.....	63
3.1.1 Análisis del Proceso Académico.....	63
3.1.2 Diagrama de flujo de datos lógico.....	65
3.1.3 Captura de Requisitos.....	67
3.1.4 Modelo de casos de uso.....	68
3.1.5 Diagrama de Casos de Uso.....	70
3.1.6 Diagrama de Secuencia y de Colaboración.....	78
3.2 Fase de Colaboración.....	83
3.2.1 Diseño del Diagrama de Clases.....	83
3.2.2 Diseño del Diagrama de Entidad - Relacion.....	84
3.2.3 Diseño del la Base de Datos.....	85
3.2.4 Diccionario de Datos.....	86
3.3 Fase de aprendizaje.....	87
3.3.1 Diseño de Interfaz de Usuario.....	88
3.3.2 Implementación del Sistema.....	93

## **CAPITULO 4 CALIDAD DEL SOFTWARE**

4.1 Evaluación de las Preferencias Elementales.....	94
4.2 Evaluación Global.....	105
4.3 Análisis de Costo.....	108
4.3.1 Costo del Software desarrollado.....	108
4.3.1.1 Estimación de Puntos de Función.....	108
4.3.2 Costo de Elaboración del proyecto.....	112
4.3.3 Costo Total.....	112

## **CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES**

5.1 Conclusiones.....	113
5.2 Recomendaciones.....	114

<b>BIBLIOGRAFÍA</b> .....	115
ANEXO A.....	117
ANEXO B.....	119
ANEXO C.....	120





## INDICE DE FIGURAS

<b>CONTENIDO</b>	<b>Pág.</b>
Figura 2.1 Arquitectura Web.....	16
Figura 2.2 Sitio Dinámico.....	17
Figura 2.3 La red Internet.....	18
Figura 2.4 Ciclo de Vida Adaptativo.....	27
Figura 2.5 Actividades del Ciclo Adaptativo.....	29
Figura 2.6 Ejemplo de diagrama E/R.....	35
Figura 2.7 Modelo Relacional Obtenido del Diagrama E-R.....	35
Figura 2.8 Anexando Atributos a la relación.....	36
Figura 2.9 Ejemplo de Casos de Uso.....	37
Figura 2.10 Ejemplo de Relación <<include>>.....	39
Figura 2.11 Ejemplo de Relación <<extend>>.....	39
Figura 2.12 Representación de una Clase.....	41
Figura 2.13 Asociación.....	42
Figura 2.14 Árbol de requerimientos de calidad para Confiabilidad y Eficiencia.....	50
Figura 2.15 Árbol de requerimientos de calidad para Usabilidad y Funcionalidad.....	51
Figura 2.16 Estructura de Agregación para la Usabilidad y Funcionalidad.....	55
Figura 2.17 Estructura de Agregación para la Confiabilidad y Eficiencia.....	56
Figura 2.18 Estructura de Agregación de Preferencias Parciales.....	56
Figura 2.19 Rango de Aceptación de Calidad.....	58
Figura 3.1 Diagrama de datos lógico del Área Académico.....	66
Figura 3.2 Flujo de Datos Lógico del Proceso de Inscripciones.....	67
Figura 3.3 Diagrama de Casos de Uso del Seguimiento Académico.....	71
Figura 3.4 Casos de uso del Proceso de inscripciones.....	75
Figura 3.5 Secuencia Correspondiente al Control de Acceso del Usuario....	78

Figura 3.6	Diagrama de Colaboración Correspondiente al Control de Acceso del Usuario.....	79
Figura 3.7	Secuencia Correspondiente a la Inscripción de Alumnos.....	79
Figura 3.8	Diagrama de Colaboración Correspondiente a la Inscripción de Alumnos.....	80
Figura 3.9	Secuencia Correspondiente a la Solicitud de Reportes.....	80
Figura 3.10	Diagrama de Colaboración correspondiente a la solicitud de Reportes.....	81
Figura 3.11	Secuencia Correspondiente a los Horarios.....	81
Figura 3.12	Diagrama de Colaboración Correspondiente a los Horarios.....	82
Figura 3.13	Secuencia correspondiente a las Notas.....	82
Figura 3.14	Diagrama de Colaboración Correspondiente a las Notas.....	83
Figura 3.15	Diagrama de Clases del Sistema.....	84
Figura 3.16	Diseño Diagrama Entidad-Relación.....	85
Figura 3.17	Esquema de la Base de Datos.....	86
Figura 3.18	Interfaz se Usuario, Pantalla Inicial.....	88
Figura 3.19	Interfaz se Usuario, Panel de Control Administrador.....	89
Figura 3.20	Interfaz se Usuario, Panel de Control Docentes.....	90
Figura 3.21	Interfaz se usuario, Panel de Control Alumnos.....	90
Figura 3.22	Nomina de Alumnos.....	91
Figura 3.23	Elaboración de Horarios.....	91
Figura 3.24	Exportación de Datos de los Alumnos.....	92
Figura 3.25	Actualización de Notas.....	92
Figura 3.26	Esquema de Funcionamiento del Sistema.....	93

## INDICE DE TABLAS

CONTENIDO	Pág.
Tabla 2.1 Características de las metodologías.....	23
Tabla 2.2 Estructura del modelo relacional.....	33
Tabla 2.3 Categoría de un Proyecto en Función de sus Líneas de Código.....	59
Tabla 2.4 Coeficientes COCOMO.....	61
Tabla 2.5 Conversión Líneas de Código a Puntos Función.....	62
Tabla 3.1 Actores de las Unidades Educativas.....	68
Tabla 4.1 Resultados de Evaluación Elemental de Usabilidad.....	95
Tabla 4.2 Resultado de las evaluaciones elementales de Funcionalidad.....	98
Tabla 4.3 Resultado de las evaluaciones elementales de Confiabilidad.....	101
Tabla 4.4 Resultado de las evaluaciones elementales de Eficiencia.....	103
Tabla 4.5 Resultado del proceso de evaluación.....	107
Tabla 4.6 Cálculo de Puntos de Función no ajustados.....	108
Tabla 4.7 Determinación de Complejidad.....	109
Tabla 4.8 Conversión de Puntos de Función a KLDC.....	110
Tabla 4.9 Coeficientes $a_b$ y $c_b$ y los exponentes $b_b$ , y $d_b$ .....	111
Tabla 4.10 Costo de Elaboración del Proyecto.....	112
Tabla 4.11 Costo Total del Proyecto.....	112

# CAPITULO 1

## INTRODUCCIÓN

### 1.1 INTRODUCCIÓN

Las Tecnologías de la Información y Comunicación (TIC) son incuestionables y están ahí, forman parte de la cultura tecnológica que nos rodea y con la que debemos convivir. Amplían nuestras capacidades físicas y mentales. Además las posibilidades de desarrollo social.

Sus principales aportaciones a las actividades humanas se concretan en una serie de funciones que nos facilitan la realización de nuestros trabajos porque, sean éstos los que sean, con frecuencia requieren una cierta información para realizarlo, un determinado proceso de datos y a menudo también la comunicación con otras personas; y esto es precisamente lo que nos ofrecen las TIC. Sin duda la administración en las Unidades Educativas es también una actividad que requiere la participación de las TIC para poder evitar las muchas dificultades que conllevan los procesos administrativos, tal es el caso de muchas personas que deben realizar largas y tediosas colas para pre-inscribir o inscribir a sus hijos a una Unidad Educativa. Si una persona desearía saber el estado de sus notas o el rendimiento de un estudiante en particular, tropieza primeramente con la pérdida de tiempo y con el protocolo del establecimiento para poder recibir dichas notas. Si un padre de familia quiere conocer el historial académico (Kardex), de su hijo(a), debe realizar un sin fin de trámites y seguir el procedimiento administrativo de la Unidad Educativa; muchas veces la persona interesada necesita la información al instante, por ejemplo para tramitar algún tipo de beca, concursos, premios, etc. y el no contar con esta información a tiempo puede determinar la pérdida de dicha oportunidad. Tener a la mano cifras estadísticas de aprobación de alumnos por trimestre, por año, cálculo de tasas de incremento de alumnado por año, el cual se

sería de mucha utilidad para poder tomar cierto tipo de decisiones respecto a la política de manejo de la enseñanza en la Unidad Educativa.

Es así que la utilidad de las TIC a través de los sistemas de apoyo en las Unidades Educativas, aportan a la automatización de sus procesos administrativos y académicos, logrando el manejo ordenado de su información y sobre todo tener más credibilidad de la información manejada en la institución.

El motivo inicial del presente proyecto es elaborar un sistema informático que logre adaptarse a los requerimientos del proceso académico, de una o varias Unidades Educativas, en este caso el sistema trabajará con cinco Unidades Educativas, las cuales podrán desenvolverse en un entorno Web, de manera local o vía Internet, con lo cual se logrará un flujo eficiente de la información, actualizaciones al instante y la centralización de la información. Todo ello a través de la “Fundación Ecología y Desarrollo” que es la encargada de estas Unidades Educativas.

## **1.2 ANTECEDENTES**

La Fundación Ecología y Desarrollo, es un institución sin ánimos de lucro, cuyo fundamento principal está basado en la defensa del medio ambiente y la incorporación del desarrollo en comunidades alejadas de éste alcance, persigue el logro de la autosostenibilidad en todos los ámbitos de trabajo, desarrollan actividades en los sectores de la educación, la capacitación técnica, la salud y la implementación de tecnologías que vayan en el camino de la defensa y la conservación del medio ambiente.

La institución no tiene afinidades políticas mucho menos de discriminación en credos y religiones o clasificaciones de capas sociales o económicas. Actualmente, trabajan en dos de los nueve departamentos de la República de Bolivia, ellos son: La Paz y Cochabamba enfocados sobre todo en las áreas peri urbanas y rurales.

En busca de brindar el apoyo necesario para el proceso académico de cada Unidad Educativa con la que trabaja, ha visto la necesidad de implementar la tecnología Web, que elevara la eficiencia de este proceso. Trabajos similares al presente se han puesto en marcha en varias Unidades Educativas, cuyo fin siempre fue lograr mejorar el proceso académico de la institución con la que se trabajo. Tal es el caso de los siguientes proyectos que a continuación se citara:

- El proyecto del “Sistema Administrativo y Académico EDUCA” el cual utilizo la metodología de ciclo de vida estructurado y metodologías orientadas al diseño de base de datos relacionales cuyo desarrollo se efectuó en el lenguaje de programación Visual Basic 6.0. Logro mejorar los procesos administrativos y académicos del Centro Educativo Maria Auxiliadora.
- El proyecto del “Sistema de información integral para el colegio San Ignacio”, la cual utiliza la metodología basada en la filosofía orientada a objetos cuyo desarrollo se efectuó en el lenguaje de programación Visual Basic. Lograron tener un mejor seguimiento y control de la toma de decisiones del Colegio San Ignacio.
- El proyecto de “Seguimiento académico al colegio Instituto Americano“, cuyo objetivo es la implementación de un nuevo sistema bajo el paradigma Orientado a Objetos y permitir mejorar el tratamiento de información automatizando el registro de alumnos, control de pago de pensiones, control de docentes, reporte de notas y seguimiento del alumno en el lenguaje de programación Delphi, Versión 3.0.

El presente proyecto de grado a diferencia de los anteriores será desarrollado bajo una metodología ágil como es la ASD (***Adaptive Software Development***), con el fin de lograr la adaptabilidad y hacer frente a los posibles cambios en el

requerimiento académico de las Unidades Educativas con las cuales se trabajara , el proyecto será desarrollado en el entorno Web, lo cual le dará la posibilidad de funcionar vía Internet o de manera local, usando para su desarrollo el lenguaje “Hypertext Preprocessor (PHP) “.

### 1.3 SITUACIÓN ACTUAL

La Fundación Ecología y Desarrollo (FECODES), trabaja en dos de los nueve departamentos de la República de Bolivia, ellos son: La Paz y Cochabamba enfocados sobre todo en las áreas Peri-urbanas y rurales, es conocida por el apoyo educativo y tecnológico que brinda. Actualmente trabaja con cinco Unidades Educativas [ Ver Anexo B ], las cuales solicitaron apoyo tecnológico a la fundación, para optimizar el seguimiento del área académica de cada una de estas. En cuanto a las Unidades Educativas se distingue una base de administración conformada a su vez por tres áreas: administrativa, contable y académica.

- **Área administrativa.-** La cual esta encargada de la organización de la Unidad Educativa, de efectuar el control de pagos, contrataciones, pensiones, otros pagos, control de los deudores, becas, agendas, transporte, biblioteca, enfermería, deportes y otros servicios.
- **Área contable.-** La cual cumple la función del manejo y registro de los recursos de la institución.
- **Área académica.-** El cual se encarga del correcto manejo de los alumnos en cuanto al grado educativo en el que se encuentra, rendimiento, materias destacadas, reprobadas, aprobadas, la asignación de horarios, actualizaciones de notas, seguimiento de inscritos, historial de los alumnos. En cuyas áreas el manejo adecuado de la información carece de eficacia

por lo cual estas Unidades Educativas han solicitado a la fundación el apoyo para lograr la eficiencia en el manejo de su información. En el presente proyecto se trabajará con el área académica en la cual se realizará un desglose de los procesos que contiene como base del análisis del sistema a implementar.

#### **1.4 DESCRIPCIÓN DE LA PROBLEMÁTICA**

La implementación de las nuevas Tecnologías de Información y Comunicación (TIC) cada vez se hacen más necesarias en un mundo que con el transcurrir del tiempo se afianza mucho más a la era de la información. Es el caso de las Unidades Educativas que ya ven a las TIC como una necesidad para ellas y para sus educandos, mas aun para el manejo eficiente del proceso académico de estas, es el caso de cinco Unidades Educativas a cargo de la Fundación Ecología y Desarrollo que tropiezan con muchas falencias en el proceso académico de ellas. Para las cuales se hace necesario un aporte tecnológico que logre adaptarse a las necesidades de cada una de ellas. Además que se ve la necesidad de la Fundación Ecología y Desarrollo de centralizar esta información en miras de obtener los datos del rendimiento académico que están alcanzando los alumnos de cada una de las Unidades Educativas que están a su cargo.

#### **1.5 LISTA DE PROBLEMAS**

- Técnica de recolección de datos obsoleta.
- Demora en la verificación del estado final de las inscripciones.
- Retardo en la inscripción de alumnos nuevos y antiguos.
- Pérdida de información personal y académica de los alumnos.
- Deficiencia en organizar los horarios y aulas.
- La nomina de alumnos llega muy a destiempo a los maestros.
- El reporte del rendimiento del alumnado es muy demorativo y suele carecer de exactitud.



- Los administrativos de las Unidades Educativas tienden a la confusión y al cansancio
- La actualización de notas es morosa
- La emisión del reporte final, del rendimiento académico de los alumnos, dirigida a la Fundación, no es adecuada.
- Falta de control permanente del rendimiento académico de los estudiantes por parte de los padres o apoderados
- La búsqueda de los datos de un estudiante carece de rapidez
- El control de asistencia a docentes es lento y obsoleto

## **1.6 PROBLEMA PRINCIPAL**

¿El sistema de seguimiento académico superará las dificultades y demoras en el seguimiento académico que tropiezan las cinco Unidades Educativas dependientes de la Fundación Ecología y Desarrollo?

## **1.7 OBJETIVOS**

### **1.7.1 Objetivo General**

Desarrollar e implementar un sistema de carácter adaptable en el área académica de cinco Unidades Educativas bajo un entorno Web, de esta forma lograr una eficiente gestión en el área académica de estas Unidades y una eficiente administración de la información, por parte de la Fundación Ecología y Desarrollo.

### **1.7.2 Objetivos Específicos**

- Construir módulos automatizados de inscripción de alumnos, reservas, incorporación de docentes, asignación de horarios, aulas y docentes.

- Recopilar la Información de los alumnos y docentes a través de procesos de búsqueda combinada.
- Construir módulos adaptables de disposición de aulas para los diferentes ciclos, disposición de horarios de cada curso, disposición de periodos, paralelos, notas mínimas a asignarse.
- Desarrollar el sistema en un entorno Web, que permite ser parte de Internet, con lo que se logra la eficiente transferencia de datos desde cualquier ubicación a las Unidades Educativas y a la Fundación Ecología y Desarrollo, además de contar con la capacidad de trabajar con el sistema de manera local o intranet en cada Unidad Educativa.
- Elaborar módulos automatizados para la actualización de notas por parte de administrativos y docentes, estos a través de Internet o de manera local.
- Elaborar cuadros estadísticos, módulos de consulta y módulos de reportes de la información de los alumnos, esto vía Internet o de manera local.
- Construir un módulo de asistencia, que controle la asistencia de los docentes de manera rápida confiable y económica.

## **1.8 JUSTIFICACIÓN**

El aporte de un medio tecnológico en los días que nos encontramos es casi imprescindible en las Unidades Educativas ya que estas no deben quedar relegadas a la obsolescencia en miras a lograr una mejor aceptación en una sociedad cada vez mas ligada a la informática. Por ello se toman en cuenta los siguientes aspectos para la realización del proyecto.

### **1.8.1 Justificación Técnica**

La implementación de un sistema adaptable en el seguimiento académico proporcionará una gran flexibilidad y adecuación en su uso a diferentes Unidades Educativas. El entorno Web da la facilidad de utilizar la tecnología más destacada de las TIC, que es el Internet; por la cual las Unidades Educativas tendrán mayor presencia en la era de las tecnologías de información, cabe mencionar que el servicio de Internet es proporcionada por la Fundación. El uso de las metodologías ágiles nos permiten el desarrollo de software rápido y respondiendo a los cambios que puedan surgir a lo largo del proyecto

El sistema será desarrollado bajo un entorno de software libre, es así que como servidor se utilizara el servidor Apache, como lenguaje de programación en el servidor se utilizara PHP, como gestor de base de datos MySQL y estos bajo una plataforma LINUX. Tomando en cuenta también que el sistema no tiene ninguna dificultad para su correcto funcionamiento en la plataforma Windows.

Los requerimientos de hardware para el funcionamiento del sistema son mínimos debido al espacio reducido de este y al gran ahorro de recursos del computador que logra el PHP, por lo que requiere como mínimo un equipo Pentium I de 200 Mhz con memoria ram de 64 Mbits, cuyo requerimiento es cubierto, por la Fundación y las Unidades Educativas que contarán con la implementación del sistema.

### **1.8.2 Justificación Social**

Socialmente el presente proyecto contribuye a nivel académico a las cinco Unidades Educativas dependientes de la fundación ecología y desarrollo brindándoles beneficios en lo que respecta a prestigio rapidez, proporcionando información oportuna y confiable a los diferentes usuarios del sistema.

Contribuye a que la Fundación Ecología y Desarrollo, este al tanto en cuanto al rendimiento académico de los alumnos miembros de cada una de las Unidades Educativas de una manera eficiente. Logra hacer parte a las Unidades Educativas de la nueva sociedad de la información.

### **1.8.3 Justificación Económica**

La ejecución del proyecto permitirá mejorar el rendimiento de los procesos manuales, el ahorro de tiempo de los procesos académicos con los que cuentan las Unidades Educativas, el costo de documentos impresos y papelería, transporte, costos horas/ hombre en trabajos operacionales y repetitivos.

### **1.9 METODOLOGÍA**

La metodología a emplear en el presente proyecto es una metodología ágil, la ASD (Adaptive Software Development) . Las metodologías ágiles permiten el desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Pretende ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos donde todas las decisiones de los clientes son tomadas al inicio para el desarrollo del software volviéndose cerradas, las metodologías ágiles buscan retrasar las decisiones lo cual no significa no tomarlas sino tomar las que sean necesarias en cada momento.

La filosofía ASD (Adaptive Software Development) fue impulsada por Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: Especulación, Colaboración y Aprendizaje.

En la primera de ellas se inicia el proyecto y se planifican las características del software.

En la segunda desarrollan las características

En la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

## 1.10 ALCANCES

El presente proyecto de grado se centrará en el área académica, en donde se identificaron los siguientes módulos, semejantes en todas las Unidades Educativas, en las cuales trabajará el sistema:

- **Modulo de nueva gestión.-** Este modulo lograra actualizar la información para el inicio de un nuevo año académico
- **Modulo de estadísticas de la gestión anterior.-** El cual permitirá tener: La nomina general de los alumnos antiguos de la Unidad Educativa, la verificación del nuevo año en curso, el control de alumnos antiguos y graduados de la Unidad Educativa.
- **Modulo de adaptación.-** Este modulo debe lograr acomodares a los distintos cambios referentes a: la disposición de aulas, materias y horarios de la Unidad Educativa; la disposición de maestros y materias, la observación del horario general de la Unidad Educativa, la observación del horario por cursos de la Unidad Educativa, la disposición del cupo para cada curso.
- **Modulo de inscripciones.-** El modulo conllevara las siguientes características: las inscripciones para alumnos antiguos, el registro e inscripción de nuevos alumnos, la asignación de un código de única identificación, El registro de la fecha y hora de cada inscripción, la verificación del resultado final de las inscripciones.

- **Modulo de seguimiento académico.-** Este modulo es el que optimiza el trabajo de cada unidad Educativa tomo en cuenta: La emisión de la nomina de alumnos por curso a cada docente que corresponda, la actualización de notas vía Web, la actualización de notas vía disket, la actualización de notas de los docentes por cada trimestre, la emisión de la calificación trimestral por alumno, la actualización de la notas finales de los alumnos, el control de alumnos que aprobaron el curso, los que reprobaron el curso y los graduados de la Unidad Educativa, la emisión de la nomina de rezagados, todos los reportes de información de alumnos ya sea individual o por curso.
- **Modulo de cierre gestión.-** Para el cierre de una gestión se debe tener en cuenta: la verificación del estado final del rendimiento académico de los alumnos miembros de la Unidad Educativa, la finalización de la gestión que permite acomodarse al inicio de la próxima gestión
- **Modulo de control de asistencia de docentes a través de diskets.-** Este debe tener un registro diario de la hora de ingreso y de salida de cada docente facilitando el control de asistencia de estos , lo realiza a través de un disco flexible para economizar el gasto de adquirir un aparato de registro sofisticado

### 1.11 APORTES

El sistema a desarrollar tiene la virtud de ser un producto tecnológico que logrará adaptarse a las necesidades del área académica de cinco Unidades Educativas, las cuales también contarán con el manual de usuario de tal forma que la persona encargada de trabajar con este software pueda manejar el mismo.

El sistema gracias a su desarrollo en un entorno Web, podrá trabajar sin dificultad a través del Internet, lo que facilitará las reservaciones de alumnos nuevos, las inscripciones de alumnos antiguos, las consultas de los alumnos y padres, la actualización de notas de manera inmediata, además que gracias a este aporte tecnológico, se da presencia a estas Unidades Educativas en la nueva era de la información que conlleva a elevar el prestigio con el que cuentan.



# CAPITULO 2

## MARCO TEÓRICO

### 2.1 INTRODUCCIÓN

El marco teórico presentado a continuación es el producto de la exposición de los principios de la ciencia de la computación relacionadas a los sistemas, métodos y enfoques metodológicos que se consideran validos para el desarrollo de proyectos de software en el medio. Como ser la utilización del lenguaje UML para especificación del problema y objetivos del sistema, tomando en cuenta la implementación de la metodología ágil ASD (*Adaptive Software Development*) que apoya el desarrollo del sistema elaborado.

Hoy en día el software ha adquirido mayor importancia en el convivir cotidiano, pues se encuentra presente en la mayoría de las actividades de nuestra sociedad, ha pasado a ser una de las industrias más importantes del mundo. En este escenario es imperativo para los ingenieros de software encontrar formas de desarrollo más eficaces y eficientes que permitan enfrentar esta compleja tarea.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodologías que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas mas tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usar. Estas propuestas han demostrado ser efectivas y necesarias en un gran numero de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo mas actividades, mas artefactos y mas restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo mas complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse



en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software

## **2.2 INGENIERÍA DE SOFTWARE**

La Ingeniería de Software es el establecimiento y uso de principios de ingeniería para obtener software que sea confiable y que funcione eficientemente en máquinas reales, esta aparece a finales de los años sesenta y principios de los setenta, comenzando con las Técnicas de Programación Estructurada, incorporándolas a las fases del ciclo vital de software. La Programación Estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño. Además, comenzaron a usarse tecnologías orientadas a objetos. El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" [JAC1998]. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [JAC 1998]

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un

plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

## 2.3 EL ENTORNO WEB

La Web nació alrededor de 1989 a partir de un proyecto del CERN, en el que Tim Berners-Lee construyó el prototipo que dio lugar al núcleo de lo que hoy es la World Wide Web. La intención original era hacer más fácil el compartir textos de investigación entre científicos y permitir al lector revisar las referencias de un artículo mientras lo fuera leyendo. Un sistema de hipertexto enlazaría todos los documentos entre sí para que el lector pudiera revisar las referencias de un artículo mientras lo fuera leyendo. El nombre original del prototipo era "*Enquire Within Upon Everything*". [DWP2000]

La funcionalidad elemental de la Web se basa en tres estándares:

- El Localizador Uniforme de Recursos (URL), que especifica cómo a cada página de información se asocia una "dirección" única en donde encontrarla.
- El Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo el navegador y el servidor intercambian información en forma de peticiones y respuestas,
- El Lenguaje de Marcación de Hipertexto (HTML), un método para codificar la información de los documentos y sus enlaces.

## 2.4 CONSORCIO WORLD WIDE WEB

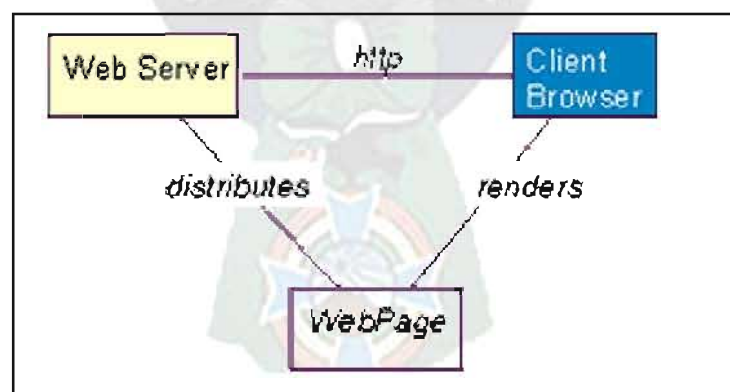
El W3C es el Consorcio World Wide Web, una asociación internacional formada por organizaciones miembros del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares web. W3C pretende guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la web. [WIF2006]

El valor social de la Web radica en su capacidad para hacer posible la comunicación, el comercio y facilitar la oportunidad de compartir conocimiento. Uno de los objetivos principales del W3C es hacer que estas ventajas estén disponibles para todo el mundo, independientemente del hardware, software, infraestructura de red, idioma, cultura, localización geográfica, o habilidad física o mental. El trabajo realizado en áreas como accesibilidad Web, internacionalización e independencia de dispositivo, es especialmente importante en la labor del W3C por alcanzar una Web para todos. Es también destacable el gran esfuerzo realizado en la traducción de los estándares del W3C, informes y notas de prensa a diferentes idiomas.

## 2.5 ARQUITECTURA WEB

La arquitectura de un sitio Web tiene tres componentes principales: un servidor Web, una conexión de red, y uno o más clientes (browsers). El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

Figura 2.1 Arquitectura Web

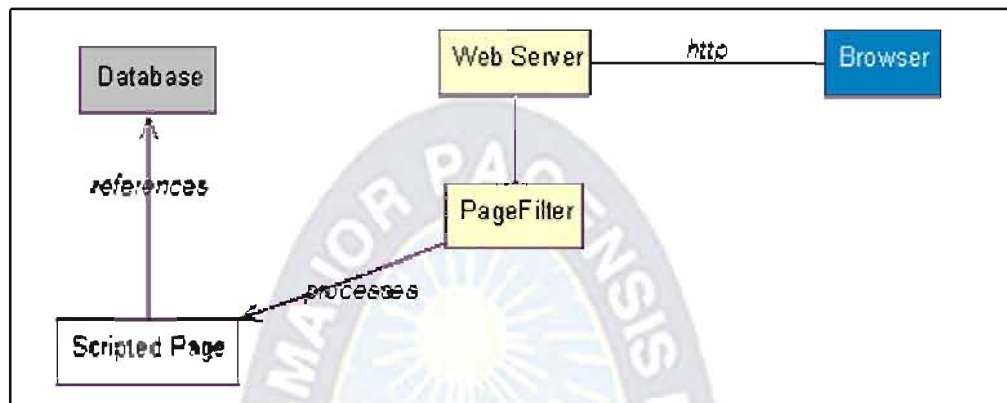


Fuente: [WIF2006]

La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está almacenada en una base de

datos, y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema, son llamados sitios dinámicos. [WIF2006]

Figura 2.2 Sitio Dinámico



Fuente: [WIF2006]

## 2.6 DEFINICIÓN DE INTERNET

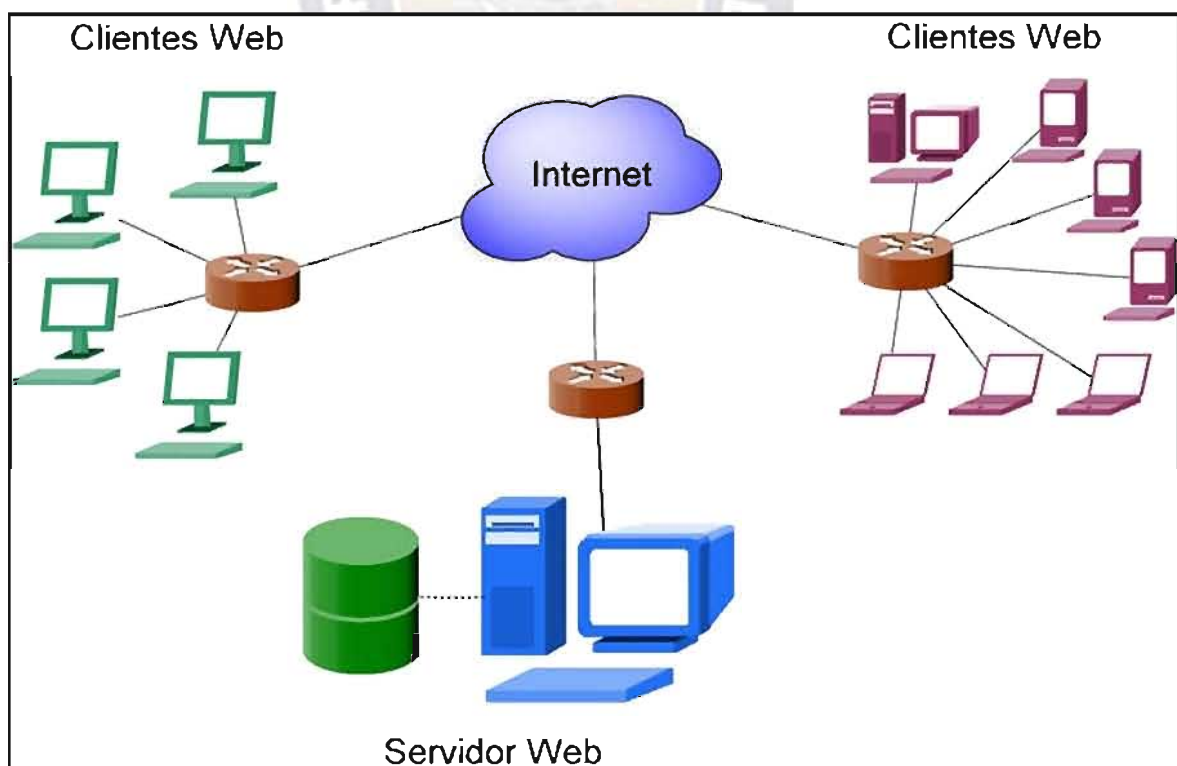
El Internet algunas veces llamado simplemente la red, es un sistema mundial de redes de computadoras, un conjunto integrado por las diferentes redes de cada país del mundo, por medio del cual un usuario en cualquier computadora puede, en caso de contar con los permisos apropiados, acceder a la información de otra computadora y poder tener inclusive comunicación directa con otros usuarios en otras computadoras (Ver Figura 2.3). Internet como una herramienta de comunicación e información, en un principio satisfacía a sus usuarios con las páginas estáticas Web que solo brindaban información, pero hoy en día se hizo necesario que las páginas dejen de ser estáticas y pasen a ser dinámicas de tal manera que satisfagan a los usuarios que son cada día más exigentes a consecuencia de esto nacen métodos y técnicas que hacen el dinamismo de las páginas, entre estas técnicas podemos mencionar a los CGI, ASP, Oracle, PHP, y otros que trabajan bajo la plataforma Windows, Linux [WIF2006]

- **CGI.** (Interfaz de pasarela común) que es una técnica de poder transmitir información de forma dinámica, con la desventaja que las aplicaciones CGI

crean un proceso para cada consulta o cual en determinado momento puede llegar a saturar el equipo y ocasionar que el equipo no responda

- **PHP** (Hypertext Processor) Es un lenguaje interpretado de alto nivel en paginas HTML, es un lenguaje de programación Web actualmente el mas utilizado, la mayoría de la sintaxis es similar a los lenguajes de programación C, Java, Perl, aunque tiene ciertas características propias de lenguaje PHP precompila el código fuente antes de ejecutarlo aumentando el rendimiento lo cual es muy importante en grandes aplicaciones. Una de sus grandes ventajas es hecho de permitir el acceso a distintas bases de datos, conteniendo funciones nativas para acceder a ACCES, SQL SERVER, POTSGRESS, INFORMIX, ORACLE , MY SQL .

Figura 2.3 La red Internet



Fuente: Elaboración Propia

## 2.7 LA ADAPTABILIDAD

Una de las características que mas se busca de un sistema software es la capacidad de lograr adaptarse a las exigencias del usuario, por ello se debe tener claro este concepto el cual puede definirse de la siguiente manera:

Acomodar, ajustar una cosa a otra.

Hacer que un objeto o mecanismo desempeñe funciones distintas de aquellas para las que fue construido.

Acomodarse a diversas circunstancias.

La adaptabilidad es uno de los objetivos del presente proyecto puesto que se desea que el sistema pueda adaptarse a los requerimientos de varias Unidades Educativas de tal manera que no se concentre mas esfuerzo del necesario para que se pueda cumplir dicho objetivo, por ende una de las metodologías que mas se adecua a este tipo de trabajo es la ASD ( Desarrollo de Software Adaptable )

## 2.8 EL APOORTE DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo ayudan a comprender la realidad del sistema a desarrollar, entender el contexto, entender las partes y sus interrelaciones; en síntesis apoya a adquirir un *conocimiento pertinente* respecto de una realidad. Un conocimiento pertinente debe evidenciar una forma de referirse de manera multidimensional a lo complejo en una concepción global.

Sabemos perfectamente que todo conocimiento conlleva siempre un riesgo de error e ilusión, en la más mínima transmisión de información de un emisor a un receptor existe siempre el riesgo de errar, concientes de esta realidad debemos

procurar que una metodología nos ayude a minimizar ese nivel de error, favoreciendo la comunicación del equipo de desarrollo.

Las metodologías tradicionales llevan un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño donde por lo general se exige un alto grado de cumplimiento a las actividades del proceso.

Pero no todos los proyectos de desarrollo cumplen las características descritas y debemos enfrentarnos a entornos cuyos requerimientos cambian con frecuencia, donde el cliente exige que los tiempos de desarrollo sean mínimos y el resultado sea de alta calidad; entonces el enfoque de las metodologías tradicionales no resulta ser el más adecuado. En este escenario, las metodologías ágiles se constituyen en una alternativa de solución aportando una elevada simplificación y exhortando la ejecución de prácticas esenciales para asegurar la calidad del producto.

## **2.9 METODOLOGÍAS ÁGILES**

Varios grupos de desarrolladores preocupados ante los problemas surgidos de la aplicación de las metodologías tradicionales proponen otras metodologías que tratan de adaptarse de mejor manera la realidad del desarrollo de software. Y es así como en el año 2001 los principales promotores de estas teorías se reunieron para debatir los puntos que tenían en común y el resultado fue la creación “*The Agile Alliance*” quienes dan a conocer el Manifiesto Ágil. Este manifiesto establece las ideas principales de las metodologías ágiles:

Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados

Es más importante crear un producto software que funcione, antes que escribir documentación exhaustiva

- La colaboración con el cliente debe prevalecer sobre la negociación de contratos
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Hay diversas metodologías ágiles, de las cuales podemos citar las siguientes:

- *Adaptive Software Development (ASD)*: Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes de software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.
- *eXtreme Programming (XP)*: la más difundida de las metodologías ágiles, definida por Kent Beck como "una forma ligera, eficiente, flexible, predecible, científica y divertida de generar software".
- *Scrum*: Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, se caracteriza por dividir los proyectos en iteraciones de 30 días (llamados carreras "sprints"), el resultado de cada sprint es un incremento ejecutable que se muestra al cliente, el equipo se reúne todos los días, de quince a treinta minutos ("scrums") para coordinación e integración., la planificación es iterativa y se hace énfasis en el seguimiento de procesos. Aplica para proyectos cuyos requisitos cambian rápidamente.



- *Crystal Methodologies*: Este conjunto de metodologías para el desarrollo de software, desarrolladas por Alistair Cockburn, se caracterizan por estar centradas en las personas que componen el equipo y procuran reducir al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas definidas para el trabajo en equipo. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Cristal Orange (25 a 50 miembros).
- *Dynamic Systems Development Method (DSDM)*:. Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Se caracteriza por ser un proceso iterativo e incremental en el cual el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.
- *Feature -Driven Development (FDD)*: Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.
- *Lean Development (LD)*: Definida por Bob Charette a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan

adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

Después de haber estudiado estas metodologías todas buscan el manifiesto ágil, pero se puede observar que el ciclo de vida de XP es mayor a ASD por ello una iteración es mas demorativa, Cabe observar que XP es optimo al contar con un equipo de trabajo, pero al trabajar individualmente XP es mas lento que ASD. Por otra parte scrum presenta iteraciones, denominadas sprints, con una duración de 30 días lo cual alarga el proyecto aun más. Como en el presente proyecto se desea satisfacer las necesidades de cinco unidades educativas, el sistema debe lograr adaptarse a estas y se debe agilizar el desarrollo del sistema en tanto el producto sea lo suficientemente bueno, es por esta razón que en la línea de las metodologías ágiles se ha optado por ASD

## 2.10 METODOLOGÍAS ÁGILES VS. METODOLOGÍAS TRADICIONALES

El siguiente cuadro presenta una resumida comparación entre las características de las metodologías ágiles versus las metodologías tradicionales.

Tabla 2.1 Características de las Metodologías

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, imponen un proceso disciplinado.
Ofrecen una buena solución para entornos cambiantes	Cierta resistencia a los cambios
El costo del cambio es mínimo, su estrategia es retrasar las decisiones	El costo de un cambio es mayor cuanto más tarde se produce
Énfasis en la comunicación del grupo	Énfasis en los roles
Impuestas internamente (por el equipo)	Impuestas externamente

Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con Numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo, participa permanentemente del desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones, el cliente está forzado a tomar todas las decisiones al principio
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente: [ JCH2000 ]

## 2.11 DESARROLLO DE APLICACIONES WEB CON PROCESOS ÁGILES

En la actualidad, debido generalmente al exceso de burocracia de las metodologías tradicionales se ha pasado, en la mayoría de los casos, a no utilizar ningún método de trabajo específico y a trabajar “a destajo” con el único y erróneo objetivo de ahorrar tiempo y dinero. Aplicar cierto grado de disciplina ayudará en el proceso de desarrollo y siempre es mejor utilizar un proceso ágil que ninguna otra cosa. Sin embargo, un proceso ágil es adecuado para el desarrollo de cierto tipo de aplicaciones, por lo que no se deberían utilizar estos métodos para cualquier tipo de desarrollo software. Los procesos ágiles son una buena elección cuando trabajamos con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previas a la entrega final, lo cual agradará al cliente. Pero la mayor barrera que habrá que salvar será convencer al

cliente de que no existe una planificación y una forma fija de hacer las cosas. En cualquier caso, lo que se garantiza es un menor riesgo ante la posibilidad de cambios en los requisitos. Porque los cambios existen, y los procesos adaptativos permitirán estos cambios lo que en definitiva, garantizará que el producto final sea el deseado por el cliente. Según afirma Booch [BCH2001].

En general, las aplicaciones Web cumplen la mayor parte de las características mencionadas en el párrafo anterior, por lo que la utilización de procesos ágiles podría ser beneficiosa para este tipo de desarrollos. La necesidad del cliente que contrata un desarrollo Web es que su producto esté disponible en la red lo más pronto posible. Si no se contempla esta necesidad, la aplicación no resultará un producto satisfactorio para el cliente. Puesto que los procesos ágiles permiten obtener versiones de producto previas a la versión final, si se aplican adecuadamente estos procesos el cliente podrá disponer de forma rápida de alguna versión intermedia. Además el ciclo de desarrollo de la mayoría de los sitios y aplicaciones Web es extremadamente corto. Esto implica que generalmente no se aplique ningún tipo de proceso, pero sin duda y como se mencionaba anteriormente más vale un proceso ágil que nada. Por otra parte, los desarrollos Web se perciben como desarrollos sencillos y los desarrolladores son sometidos a una gran presión de trabajo para terminar lo más pronto posible.

Esta forma de trabajar va a implicar sin duda alguna modificaciones. Luego sería conveniente garantizar un proceso de desarrollo adaptable a los cambios. Otra cuestión fundamental a tener en cuenta es que las aplicaciones Web se desarrollan sin conocer los perfiles de los usuarios finales de las mismas, o lo que es lo mismo sin conocer los requisitos de usuario del sistema.

Sin lugar a dudas esto implicará cambios en los requisitos inicialmente detectados, lo que nos lleva de nuevo a la elección de un proceso adaptativo. Por lo tanto, podríamos concluir que este tipo de procesos son especialmente aplicables al desarrollo de aplicaciones para la Web.

## 2.12 ASD – ADAPTIVE SOFTWARE DEVELOPMENT

Jim Highsmith en su libro [HIJ2000] es la mente detrás de este proceso ágil. ASD consiste en un cambio de filosofía en las organizaciones pasando de la transición del modelo Comando-Control al modelo Liderazgo-Colaboración.

ASD propone utilizar en cambio el ciclo de vida de la Figura 2.4, el cual toma en cuenta las fases de: Especulación Colaboración y Aprendizaje.

### 2.12.1 Fase de Especulación

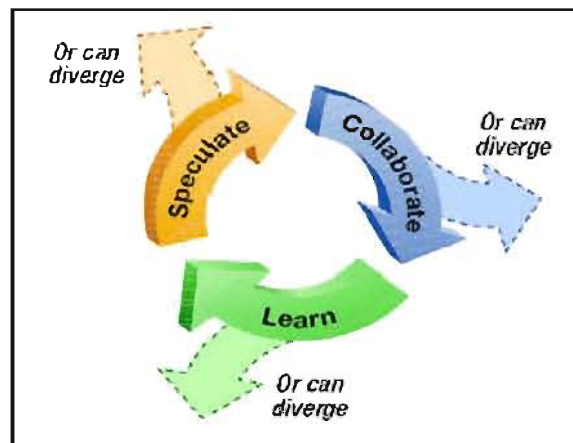
El proyecto comienza con una fase de especulación en que se lleva a cabo la planificación tentativa del proyecto en función de las entregas que se irán realizando.

La utilización del verbo Especular demuestra el interés de Highsmith en demostrar la naturaleza impredecible de los sistemas complejos. En esta etapa se fija un rumbo determinado a ser seguido en el desarrollo, sabiendo a partir de ese momento que no será el lugar en que finalizará el proyecto.

En cada iteración, se aprenderán nuevas funcionalidades, se entenderán viejas cuestiones, y cambiarán los requerimientos. Gracias a centrarse en la especulación, ASD permite administrar estos proyectos de alto cambio y rápido desarrollo que se encuentran en el *borde del caos*. Sin embargo, no es más que una especulación ya que el carácter adaptativo del proceso permite pequeñas desviaciones en un sentido – por lo que Highsmith sugiere que cada ciclo se componga de un mix entre funcionalidades críticas, útiles, y opcionales, previendo los posibles retrasos que puedan existir mediante el movimiento de las funcionalidades de menor prioridad a futuros ciclos y grandes desviaciones en otro, las cuales son utilizadas para la exploración del dominio y de la aplicación,

que puede llevar a cambiar el rumbo del proyecto estos desvíos está representado por las flechas de divergencia en la Figura 2.4.

Figura 2.4 Ciclo de Vida Adaptativo



Fuente: [HIJ2000]

### 2.12.2 Fase de Colaboración

La siguiente fase del ciclo de vida, Colaborar, es aquella en la que se construye la funcionalidad definida durante la especulación. ASD define un *Componente* como un grupo de funcionalidades o entregables a ser desarrollados durante un ciclo iterativo. Durante cada iteración el equipo colabora intensamente para liberar la funcionalidad planificada. También, existe la posibilidad de explorar nuevas alternativas, realizar pruebas de concepto, pudiendo eventualmente alterar el rumbo del proyecto profundamente. ASD no propone técnicas ni prescribe tareas al momento de llevar a cabo la construcción simplemente mencionando que todas las prácticas que sirvan para reforzar la colaboración serán preferidas, siguiendo de esta forma la línea de las metodologías ágiles respecto a la orientación a componentes. El énfasis se ubica en la relaciones entre las personas que deben estar lo suficientemente lubricadas para generar una propiedad imprescindible de los organismos complejos: emergencia. La emergencia es una propiedad de los sistemas adaptativos complejos que crea alguna propiedad más grande del todo (comportamiento del sistema) a partir de la interacción entre las partes

(comportamiento auto-organizativo de los agentes). Gracias a esta propiedad los grupos de desarrollo logran sacar lo mejor de si en la el *borde del caos*.

### 2.12.3 Fase de Aprendizaje

La fase final de ASD, Aprender, consiste en la revisión de calidad que se realiza al final de cada ciclo. En la misma se analizan cuatro categorías de cosas para aprender [HIJ2000]:

- Calidad del resultado de la desde la perspectiva del cliente
- Calidad del resultado de la desde la perspectiva técnica
- El funcionamiento del equipo de desarrollo y las prácticas que este utiliza
- El status del proyecto

Para evaluar la calidad desde el punto de vista del cliente se sugieren utilizar grupos de enfoque en el cliente, mediante los cuales se explora un modelo de la aplicación y se anotan los requerimientos de cambio del cliente.

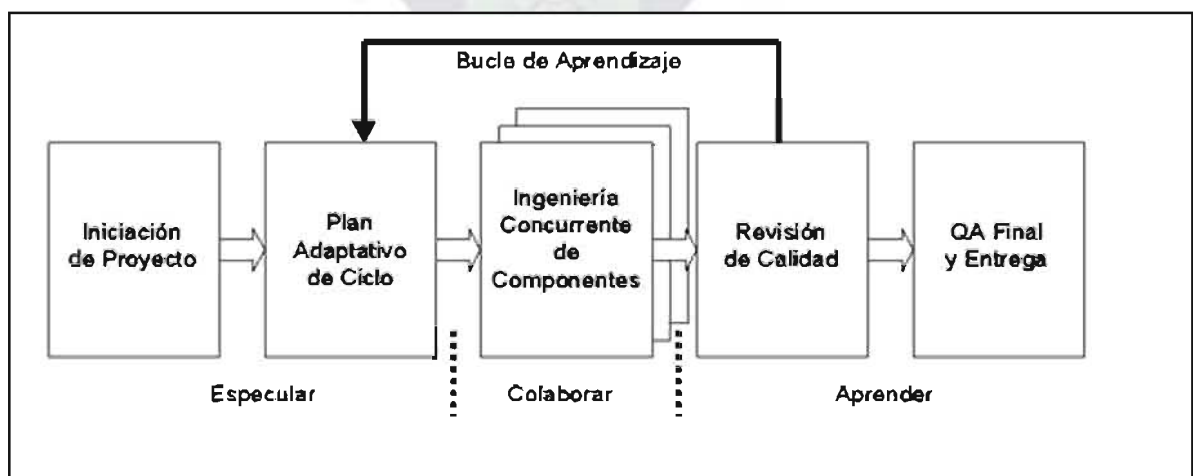
Las revisiones al diseño, al código o a las pruebas permitirán aprender sobre la calidad de los mismos. En este caso, el énfasis estará puesto en aprender cuales han sido los errores o desvíos y poder resolverlos, y no en encontrar culpables. Asimismo, está es la etapa en que se evaluarán las exploraciones que se hayan realizado dando la capacidad de poder modificar la arquitectura del sistema si se ha encontrado algún camino que se ajusta mejor a lo que necesita el usuario o si han cambiado los requerimientos. El tercer proceso de feedback está relacionado con la interacción entre las partes, la dinámica de grupo, y las técnicas empleadas. Para medir la performance y el grado de cohesión del mismo, se podrán realizar al final de cada ciclo pequeñas reuniones de postmortem. En las mismas se discuten los aspectos del proceso que contribuyen al desarrollo y aquellos que deben ser descartados por su influencia negativa.

En relación al status del proyecto, se realizarán revisiones para determinar el estado del mismo en relación a lo planificado. En este momento, se detectarán posibles diferencias que pueden surgir de la exploración y que cambiarán el rumbo a que apuntaba el proyecto.

En la Figura 2.5 se puede ver el detalle interno de cada fase como ya fue explicado, mostrándose con una flecha que trasciende las tres fases en sentido inverso, el bucle de aprendizaje. Este bucle es algo crítico para ASD ya que denota un cambio en el esquema tradicional de la vista de un sistema en que se tenía un bucle de control para detectar diferencias y corregirlas. Es decir, en las metodologías tradicionales las diferencias respecto a lo planificado eran vistas como errores que debían ser enmendados para que cumplieran lo pautado. ASD y las metodologías ágiles plantean la necesidad de que el feedback necesario sea para aprender, nos da la posibilidad de entender más respecto al dominio y construir la aplicación que mejor satisfaga las necesidades del cliente. Highsmith lo expone claramente en la siguiente frase:

En ambientes complejos, el seguir un plan al pie de la letra produce el producto que pretendíamos, pero no el producto que necesitamos.

Figura 2.5 Actividades del Ciclo Adaptativo



Fuente: [HIJ2000]



Este método ágil pretende abrir una tercera vía entre el “desarrollo monumental de software” y el “desarrollo accidental”, o entre la burocracia y la adhocracia. En el cual se que indica que deberíamos buscar más bien, “el rigor estrictamente necesario”; para ello hay que situarse en coordenadas apenas un poco fuera del caos y ejercer menos control que el que se cree necesario [HIJ2000].

ASD presupone que las necesidades del cliente son siempre cambiantes. La iniciación de un proyecto involucra definir una misión para él, determinar las características y las fechas y descomponer el proyecto en una serie de pasos individuales, cada uno de los cuales puede abarcar entre cuatro y ocho semanas. Los pasos iniciales deben verificar el alcance del proyecto; los tardíos tienen que ver con el diseño de una arquitectura, la construcción del código, la ejecución de las pruebas finales y el despliegue..

Un ciclo de vida es una iteración; este ciclo se basa en componentes y no en tareas, es limitado en el tiempo, orientado por riesgos y tolerante al cambio. Que se base en componentes implica concentrarse en el desarrollo de software que trabaje, construyendo el sistema pieza por pieza. En este paradigma, el cambio es bienvenido y necesario, pues se concibe como la oportunidad de aprender y ganar así una ventaja competitiva; de ningún modo es algo que pueda ir en detrimento del proceso y sus resultados.

Highsmith piensa que los procesos rigurosos (repetibles, visibles, medibles) son encomiables porque proporcionan estabilidad en un entorno complejo, pero muchos procesos en el desarrollo (por ejemplo, el diseño del proyecto) deberían ser flexibles. La clave para mantener el control radica en los “estados de trabajo” (la colección de los productos de trabajo) y no en el flujo de trabajo (workflow). Demasiado rigor, por otra parte, acarrea rigor mortis, el cual impide cambiar el producto cuando se introducen las inevitables modificaciones. En la moderna teoría económica del retorno creciente, ser capaz de adaptarse es significativamente más importante que ser capaz de optimizar [HIJ2000].

El modelo de Highsmith es, naturalmente, complementario a cualquier concepción dinámica del método; no podría ser otra cosa que adaptable, después de todo, y por ello admite y promueve integración con otros modelos y marcos como el lenguaje de modelado UML. Un estudio de Dirk Riehle compara ASD con XP, encontrando similitudes y diferencias de principio que pueden conciliarse con relativa facilidad, al lado de otras variables que son incompatibles. La actitud de ambos métodos frente a la redundancia de código, por ejemplo, es distinta; en XP se debe hacer todo “una vez y sólo una vez”, mientras que en ASD la redundancia puede ser un subproducto táctico inevitable en un ambiente competitivo y debe aceptarse en tanto el producto sea “suficientemente bueno”. En materia de técnicas, ASD las considera importantes pero no más que eso; para XP, en cambio, los patrones y la refactorización son indispensables.

### **2.13 SISTEMA DE BASE DE DATOS**

En el entorno informático, la gestión de bases de datos ha evolucionado desde ser una aplicación más disponible para los computadores, a ocupar un lugar fundamental en los sistemas de información. En la actualidad, un sistema de información será más valioso cuanto de mayor calidad sea la base de datos que lo soporta, la cual resulta a su vez un componente fundamental del mismo, de tal forma que puede llegarse a afirmar que es imposible la existencia de un sistema de información sin una base de datos, que cumple la función de "memoria", en todas sus acepciones posibles, del sistema.

#### **2.13.1 Base de Datos**

Las bases de datos almacenan, como su nombre dice, datos. Estos datos son representaciones de sucesos y objetos, a diferente nivel, existentes en el mundo real: en su conjunto, representan algún tipo de entidad existente. En el mundo real

se tiene percepción sobre las entidades u objetos y sobre los atributos de esos objetos; en el mundo de los datos, hay registros de eventos y datos de eventos.

### **2.13.2 Los Sistemas de Gestión de Bases de Datos**

Pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

El SGBD incorpora como herramienta fundamental un lenguaje, para la definición y la manipulación de los datos. El SQL (Structured query language), lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores. Como su nombre indica, el SQL nos permite realizar consultas a la base de datos. Pero el nombre se queda corto ya que SQL además realiza funciones de definición, control y gestión de la base de datos.

### **2.14 MODELO DE DATOS RELACIONAL**

En este modelo los datos y las relaciones entre los datos se presentan por medio de una serie de tablas, cada una de las cuales tiene varias columnas con nombres únicos y una columna de una tabla representa una relación entre un conjunto de valores. Las bases de datos relacionales son el tipo de bases de datos actualmente más difundido. Los motivos de este éxito son fundamentalmente dos:

Ofrecen sistemas simples y eficaces para representar y manipular los datos  
Se basan en un modelo, el relacional, con sólidas bases teóricas

El modelo relacional fue propuesto originariamente por E.F. Codd en un ya famoso artículo de 1970. Gracias a su coherencia y facilidad de uso, el modelo se ha convertido en los años 80 en el más usado para la producción de DBMS.

La estructura fundamental del modelo relacional es precisamente esa, "relación", es decir una tabla bidimensional constituida por líneas (tupla) y columnas (atributos).

Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad. Por ejemplo, si en la base de datos se tienen que representar personas, se podrá definir una relación llamada "Personas", cuyos atributos describen las características de las personas (tabla siguiente).

Cada tupla de la relación "Personas" representará una persona concreta.

Tabla 2.2 Estructura del modelo relacional

Persona				
Nombre	Apellido	Nacimiento	Sexo	Estado Civil
Juan	Loza	15/06/1971	H	Soltero
Isabel	Galvez	23/12/1969	M	Casada
Micaela	Ruiz	02/10/1985	M	Soltera

Fuente: Elaboración Propia

En realidad, siendo rigurosos, una relación es sólo la definición de la estructura de la tabla, es decir su nombre y la lista de los atributos que la componen. Cuando se puebla con las tuplas, se habla de "instancia de relación". Por eso, la tabla anterior representa una instancia de la relación persona. Una representación de la definición de esa relación podría ser la siguiente:

Personas (nombre, apellido, fecha\_nacimiento, sexo, estado\_civil)

## 2.15 DIAGRAMA ENTIDAD-RELACIÓN (E-R)

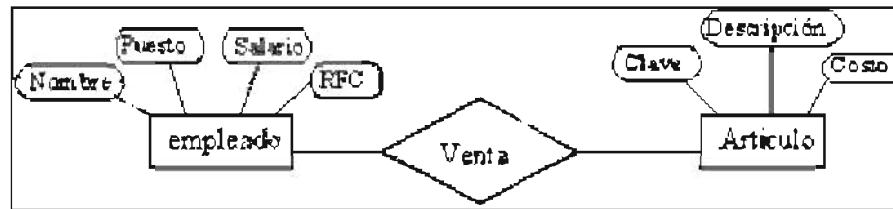
Este modelo representa a la realidad a través de un esquema gráfico empleando los terminología de *entidades*, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades esta representada por la relación del modelo.

El modelo E/R aporta una herramienta de modelado para representar las entidades, propiedades y relaciones: los diagramas Entidad/Relación mediante éstos, el esquema conceptual abstracto puede ser mostrado gráficamente y mantener una independencia conceptual con respecto a la implementación propiamente dicha.

En realidad, podemos hacer que los diagramas sean un reflejo fiel de las relaciones, interrelaciones y atributos del modelo relacional de datos o podemos englobar diversas relaciones en una sola entidad o conjunto de propiedades.

Los diagramas E/R son parecidos a los diagramas de flujo (organigramas) clásicos en que utilizan rectángulos, rombos y óvalos, pero los significados de estos elementos son distintos.

Figura 2.6 Ejemplo de diagrama E/R



Fuente: Elaboración Propia

-Los rectángulos representan entidades, los rombos relaciones y los óvalos propiedades.

Un diagrama E-R, puede ser representado también a través de una colección de tablas. Para cada una de las entidades y relaciones existe una tabla única a la que se le asigna como nombre el del conjunto de entidades y de las relaciones respectivamente, cada tabla tiene un número de columnas que son definidas por la cantidad de atributos y las cuales tienen el nombre del atributo.

El diagrama E-R de la Figura 2.7 en la que intervienen las entidades de Vendedor con los atributos RFC, nombre, puesto, salario y Artículo con los atributos Clave, descripción, costo. Puede ser representada en una colección de tablas o en su modelo relacional de la siguiente forma:

Figura 2.7 Modelo relacional obtenido del diagrama E-R

EMPLEADO				ARTICULO		
Nombre	Puesto	Salario	RFC	Clave	Descripción	Costo
Teófilo	Vendedor	2000	TEAT701210XYZ	A100	Abanico	460
Cesar	Auxiliar ventas	1200	COV741120ABC	C260	Colcha matrimonial	1200

VENTA	
RFC	Clave
TEAT701210XYZ	C260
COV741120ABC	A100

Fuente: Elaboración Propia

Nótese que en la tabla de relación - Venta -, contiene como atributos a las llaves primarias de las entidades que intervienen en dicha relación, en caso de que exista un atributo en las relaciones, este atributo es anexado como una fila más de la tabla.

Por ejemplo si anexamos el atributo fecha a la relación venta, la tabla que se originaría sería la siguiente:

Figura 2.8 Anexando atributos a la relación

RELACION - VENTA		
RFC	Clave	Fecha
TEAT701210XYZ	C260	10/12/96
COV741120ABC	A100	11/12/96

Fuente: Elaboración Propia

## 2.16 LENGUAJE DE MODELADO UNIFICADO

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software . Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE.

En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores. UML al ser un lenguaje de modelado de sistemas y no un a metodología puede adecuarse en varias metodologías tales

como las ágiles para lograr un modelado óptimo y colaborar en el ciclo de vida de estas.

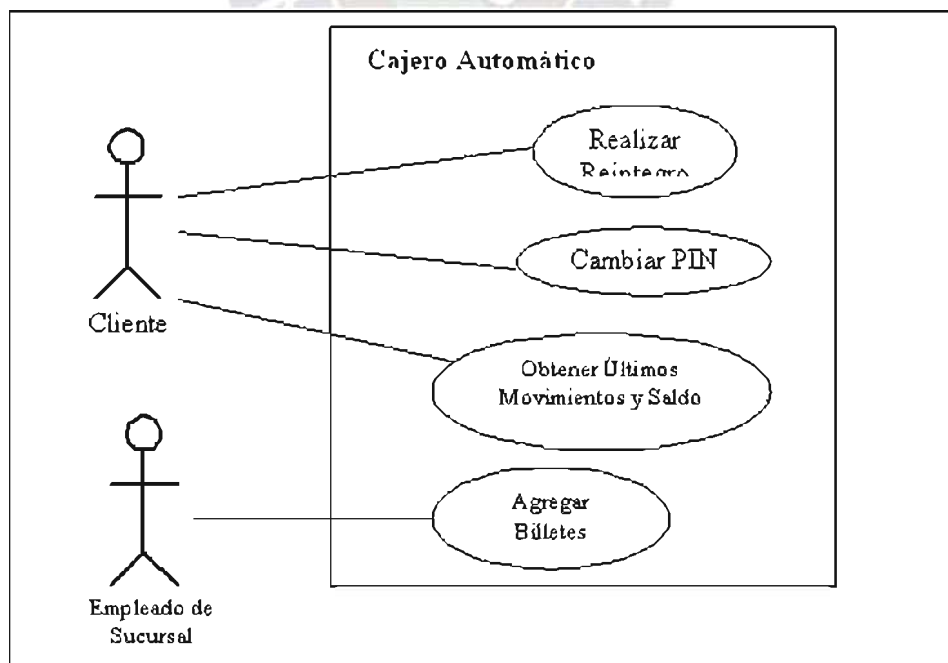
### 2.16.1 Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

En la Figura 2.9 se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

Figura 2.9 Ejemplo de Casos de Uso



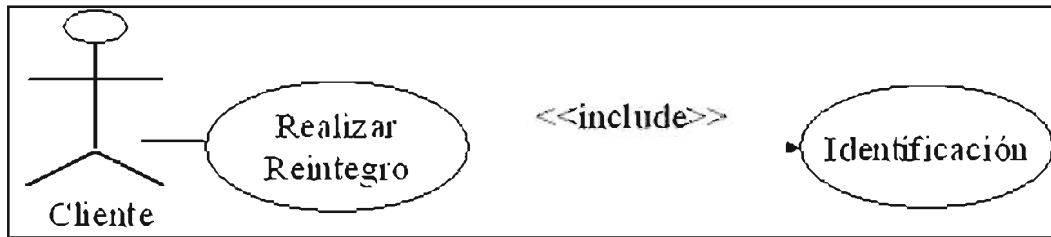
Fuente: Elaboración Propia



Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

- **Actores-** Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores.
- **Casos de Uso.-** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.
- **Relaciones entre Casos de Uso.-** Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación en el equipo de desarrollo, el manejo de la documentación de casos de uso. Para el caso de que queramos utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos: • Incluye (<>): Un caso de uso base incorpora explícitamente a otro caso de uso en un lugar especificado en dicho caso base. Se suele utilizar para encapsular un comportamiento parcial común a varios casos de uso. En la Figura 2.10 se muestra cómo el caso de uso Realizar Reintegro puede incluir el comportamiento del caso de uso Autorización.

Figura 2.10 Ejemplo de Relación <<include>>



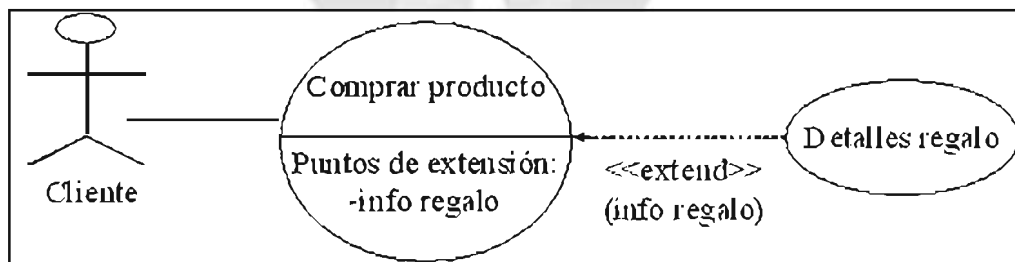
Fuente: Elaboración Propia

Cuando un caso de uso base tiene ciertos puntos (puntos de extensión) en los cuales, dependiendo de ciertos criterios, se va a realizar una interacción adicional.

El caso de uso que extiende describe un comportamiento opcional del sistema (a diferencia de la relación incluye que se da siempre que se realiza la interacción descrita).

En la Figura 2.11 se muestra como el caso de uso Comprar Producto permite explícitamente extensiones en el siguiente punto de extensión: info regalo. La interacción correspondiente a establecer los detalles sobre un producto que se envía como regalo están descritos en el caso de uso Detalles Regalo.

Figura 2.11 Ejemplo de Relación <<extend>>



Fuente: Elaboración Propia

Ambos tipos de relación se representan como una dependencia etiquetada con el estereotipo correspondiente (<> o <>), de tal forma que la flecha indique el sentido en el que debe leerse la etiqueta. Junto a la etiqueta <> se puede detallar el/los puntos de extensión del caso de uso base en los que se aplica la extensión.

## 2.16.2 Diagramas de Secuencia

El Diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia se modela para cada caso de uso. Mientras que el diagrama de caso de uso permite el modelado de una vista general del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

## 2.16.3 Diagrama de colaboración

Es una forma de representar interacción entre los objetos, es decir, las relaciones entre ellos y la secuencia de los mensajes de las iteraciones que están indicadas por un número. Muestra como varios objetos colaboran en un solo caso de uso

## 2.16.4 Diagrama de Clases

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia.

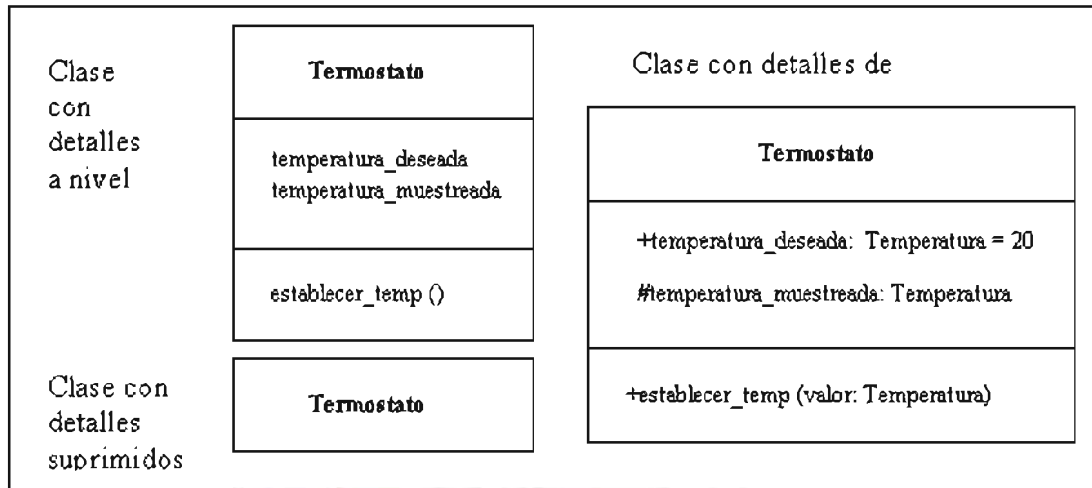
Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones.

- **Clases.**- Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones.

Una clase puede representarse de forma esquemática, con los atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

En la Figura 2.12 se ve cómo una misma clase puede representarse a distinto nivel de detalle según interese, y según la fase en la que se esté.

Figura 2.12 Representación de una Clase

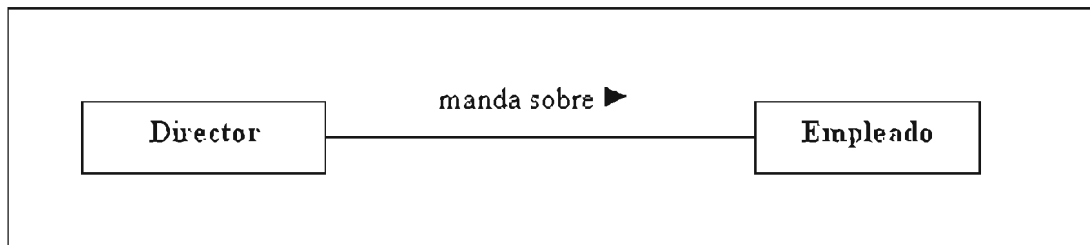


Fuente: Elaboración Propia

- **Objetos.**- Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre\_del\_objeto: nombre\_de\_la\_clase Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase.
- **Asociaciones.**- Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los más importantes de entre dichos elementos gráficos.
- **Nombre de la Asociación y Dirección.**- El nombre de la asociación es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo negro sólido que indique la dirección en la cual leer el nombre de la asociación.

En el ejemplo de la Figura 2.13 se puede leer la asociación como “Director manda sobre Empleado”.

Figura 2.13 Asociación



Fuente: Elaboración Propia

Los nombres de las asociaciones normalmente se incluyen en los modelos para aumentar la legibilidad. Sin embargo, en ocasiones pueden hacer demasiado abundante la información que se presenta, con el consiguiente riesgo de saturación. En ese caso se puede suprimir el nombre de las asociaciones consideradas como suficientemente conocidas. En las asociaciones de tipo agregación y de herencia no se suele poner el nombre.

## 2.17 HERRAMIENTAS DE CONSTRUCCIÓN DE SOFTWARE

El presente proyecto buscado el mas optimo rendimiento en cuanto al desarrollo de un sistema que pueda trabajar en el entorno Web, se a optado por usar los siguientes programas que optimizan el desarrollo de paginas dinámicas, el manejo de datos y poder contar con las ventajas de trabajar en intranet , en Internet, además de ser multiplataforma

### 2.17.1 PHP - Hypertext Processor

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten crear aplicaciones complejas con una curva de aprendizaje muy suave. También permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas. Su interpretación y ejecución se da en el servidor, en el cual se

encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET / C+ /VB.NET, a ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE comercial llamado

#### **a) Historia**

PHP fue originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador Danés-Canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio del 1995 fue publicado "Personal Home Page Tools" después de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI.

Dos programadores israelíes del Technion, Zeev Suraski y Andi Gutmans, reescribieron el analizador sintáctico (*parser* en inglés) en el año 1997 y crearon la base del PHP 3, cambiando el nombre del lenguaje a la forma actual. Inmediatamente comenzaron experimentaciones públicas de PHP 3 y fue lanzado oficialmente en junio del 1998.

Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como Zend Engine o motor Zend. También conformaron Zend Technologies en Ratmat Gan, Israel. En mayo de 2000 PHP 4 fue lanzado bajo el poder del motor Zend Engine 1.0. El 13 de julio de 2004, fue lanzado PHP 5, utilizando el motor Zend Engine II (o Zend Engine 2). La versión más reciente de PHP es la 5.1.6, que incluye el novedoso PDO (Objetos de Datos de PHP o PHP Data Objects) y mejoras utilizando todas las ventajas que provee el nuevo Zend Engine 2.

## **b) Usos de PHP**

Los principales usos del PHP son los siguientes:

Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.

Programación en consola, al estilo de Perl o Shell scripting.

Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

## **c) Ventajas de PHP**

Es un lenguaje multiplataforma.

Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.

Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.

Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.

Posee una amplia documentación en su página oficial .

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Permite las técnicas de Programación Orientada a Objetos.

Nos permite crear los formularios para la Web.

### **2.17.2 My SQL**

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.



## a) Historia

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Se comenta en el manual [\[MySQL Manual\]](#) que no se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

La versión estable de este gestor a días de hoy es la 3.23.49. Se puede encontrar más información sobre este gestor en el manual [\[MySQL Manual\]](#).

## b) Características

Las principales características de este gestor de bases de datos son las siguientes:

Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

Soporta gran cantidad de tipos de datos para las columnas.

Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).

Gran portabilidad entre sistemas.

Soporta hasta 32 índices por tabla.

Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

### **2.17.3 Servidor Apache HTTP.-**

El Servidor Apache HTTP es un servidor Web de tecnología Open Source sólido originalmente basado en el servidor mas popular de la epoca: NCSA httpd a principios de 1995. El nombre Apache es "A PAtCHy server".

Desde su origen a evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache es de acuerdo al estudio hecho por Netcraft, el servidor WWW más popular del momento.

Apache a demostrado ser substancialmente mas rápido que muchos otros servidores libres y compite de cerca con los mejores servidores comerciales.

### **2.18 SEGURIDAD**

La seguridad es uno de los aspectos más importantes y conflictivos en el uso de Internet. La falta de una política de seguridad global esta frenando el desarrollo de Internet en diferentes áreas, por lo que es necesario crear entornos seguros.

Para el desarrollo de sistema se desarrolla un entorno de seguridad para proteger la información enviada del servidor hacia el cliente y viceversa, de esta manera evitar el uso no autorizado de las funciones que ofrece el sistema, dotando al mismo de servicios de seguridad.

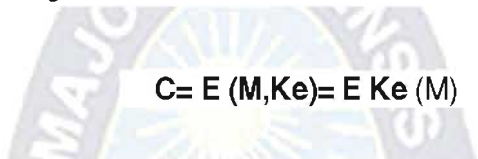
#### **2.18.1 Criptografía**

Las raíces etimológicamente de la palabra criptografía son Kriptos, que significa oculto y graphos, que significa como escribir, lo que da una clara idea de su definición clásica; arte de escribir mensajes en clave secreta o enigmáticamente.

Ahora bien la criptografía corresponde solo a una parte de la comunicación. Si se requiere secreto para la comunicación, es por que existe desconfianza o peligro de que el mensaje transmitido sea interpretado por un enemigo y ese tal vez quiera

alterar el mismo para su beneficio, para esto utilizará un conjunto de técnicas y métodos que constituyen una ciencia conocida como criptoanálisis [ASO05].


- **Modelo de Criptografía:** La encriptación esta asociada con la transformación de un mensaje inteligente (M) a una forma no inteligible (C) con la ayuda de una clave secreta (Ke) antes de que sea colocada en un medio inseguro. La encriptación se define como una función matemática (algoritmo) de la siguiente forma:


$$C = E(M, K_e) = E_{K_e}(M)$$

- Donde E puede ser una función computacional.
  - Un algoritmo de encriptación debe tener la propiedad de que el dato original pueda ser recuperado de su forma encriptada si se conoce en valor de su llave.
- **Criptografía de clave pública**

En los cifrados de la clave pública la característica fundamental es que la clave de descifrado no se puede calcular a partir del cifrado. Con un sistema asimétrico cualquier usuario puede enviar un mensaje cifrado usando la clave pública, de este último, pero solo aquellos que conozcan la clave secreta correspondiente pueden descifrar correctamente.

- **Algoritmo básico del md5**



El mensaje se divide en bloques de 512 bits, añadiendo bits si es necesario al último bloque (de 1 a 448, por una parte, más 64 fijos de un valor  $k \bmod 264$ ). Con cada bloque de 512 se realizan 64 iteraciones, en las que intervienen cuatro funciones primitivas con operaciones: suma  $\bmod 2^{32}$ , lógicas (AND, OR y NOT). En cada iteración la función MD5 se alimenta con

el bloque correspondiente y con el resultado de la iteración anterior, un vector de cuatro elementos de 32 bits cada uno (con valores predeterminados en la primera iteración). La salida de cada iteración se convierte en el nuevo vector (128 bits). En la última iteración, el vector resultante contendrá el resumen de 128 bits.

## **2.19 EVALUACIÓN DE CALIDAD DE SITIOS WEB ( WEB-SITE QEM )**

Este enfoque propuesto, es esencialmente integral, flexible y robusto, y cubre la mayor parte de las actividades en el proceso de evaluación, comparación, y selección del entorno Web.

La estrategia propuesta, denominada Metodología de Evaluación de Calidad de Sitios Web (o, en inglés, Web-site Quality Evaluation Method, o, metodología Web-site QEM) elaborada por el Dr. Luís Olsina, pretende realizar un aporte ingenieril al proponer un enfoque sistemático, disciplinado y cuantitativo que se adecue a la evaluación, comparación y análisis de calidad de sistemas de información centrados en la Web (más o menos complejos). En términos generales, y teniendo como marco conceptual al estándar ISO [ISO/IEC 9126], se consideran tres perfiles de usuario, a un alto nivel de abstracción para dominios Web, a saber: visitantes, desarrolladores, y gerenciadorees. Siguiendo un mecanismo de categorización y descomposición podemos dividir, por ejemplo a la categoría visitante en clases más específicas como veremos para los estudios que iremos desarrollando.

Particularmente, el estándar citado afirma que la relativa importancia de las características de calidad (como usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad, y mantenibilidad) varían dependiendo del punto de vista considerado y de la criticidad de los componentes del artefacto a evaluar. Por ejemplo, la visión del usuario (que nosotros denominamos punto de vista o perfil del visitante, para el dominio Web), concierne a interés de los mismos en usar el artefacto de software, como así también su

eficiencia, su facilidad de uso, entre otros aspectos. Los usuarios no están interesados en características internas o de desarrollo de los artefactos (sin embargo, atributos internos contribuyen a la calidad de uso).

En cambio, la visión de calidad del desarrollador debe considerar no sólo los requerimientos del producto para la visión del usuario sino también la calidad para los productos intermedios resultantes de las actividades del desarrollo. Para poder evaluar a la calidad de los productos intermedios de las diferentes actividades de la fase de desarrollo, los evaluadores deberán considerar las métricas apropiadas para la misma característica de acuerdo al proceso.

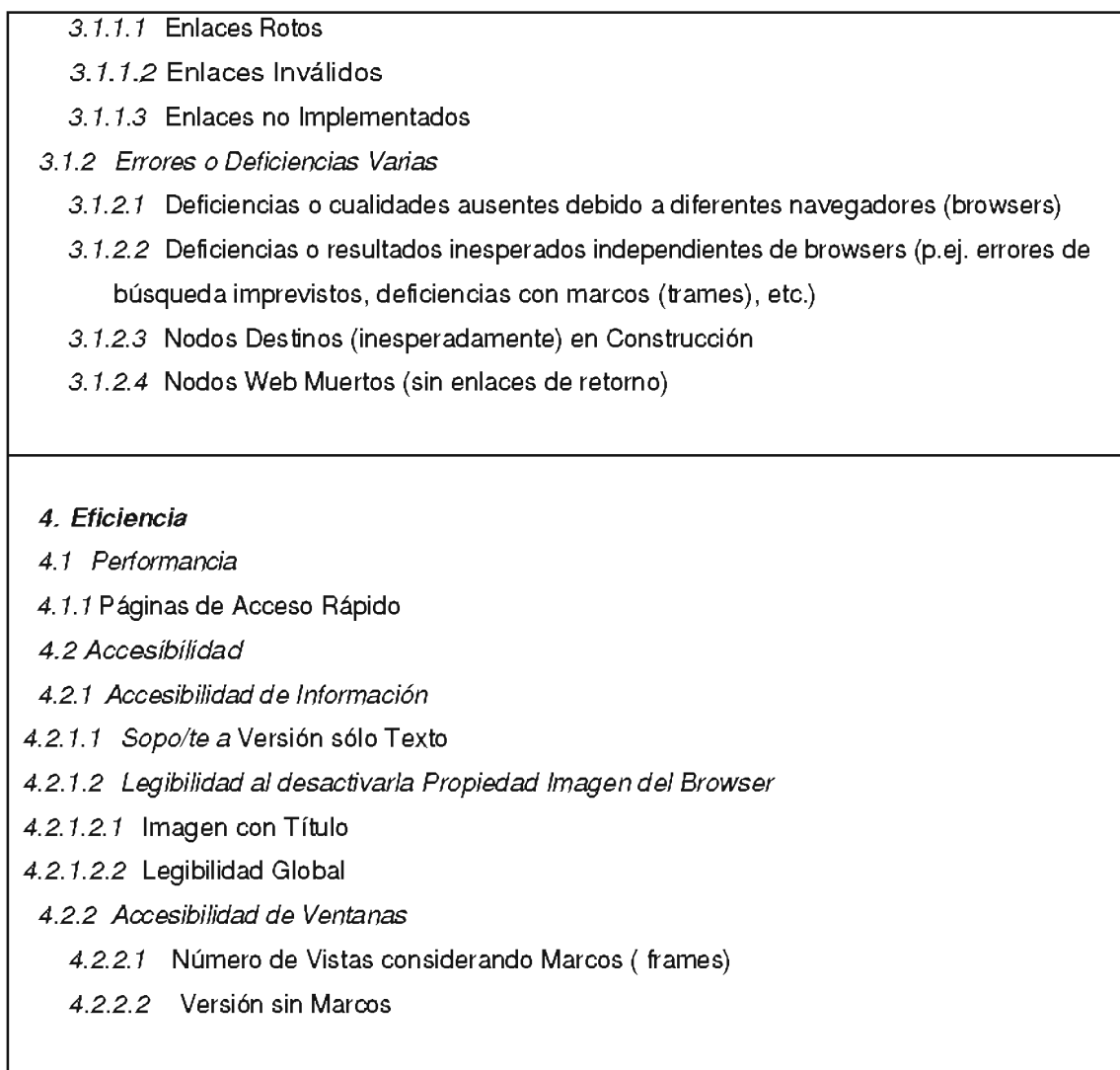
- **Hacia un Árbol de Requerimientos de Calidad General para Dominios Web** .- Un aspecto importante, reside en determinar un árbol estándar que pueda ser reusado y personalizado para distintos subdominios de aplicaciones, dentro del dominio de la Web. Contar con un árbol de requerimientos general pero a su vez fácilmente adaptable a diversos dominios, sería de gran beneficio para tareas de aseguramiento y control de calidad posibilitando potencialmente la automatización de las mismas [OLS1999].
- **Características, Subcaracterísticas y Atributos en la Web**.- En esta sección, mostraremos el árbol de requerimientos de calidad, correspondiente al dominio Web de una Unidad Académica específicamente presentaremos los requerimientos de calidad el cual esta establecido por el Dr. Olsina

Figura 2.14 Árbol de requerimientos de calidad para Confiabilidad y eficiencia

### **3. Confiabilidad**

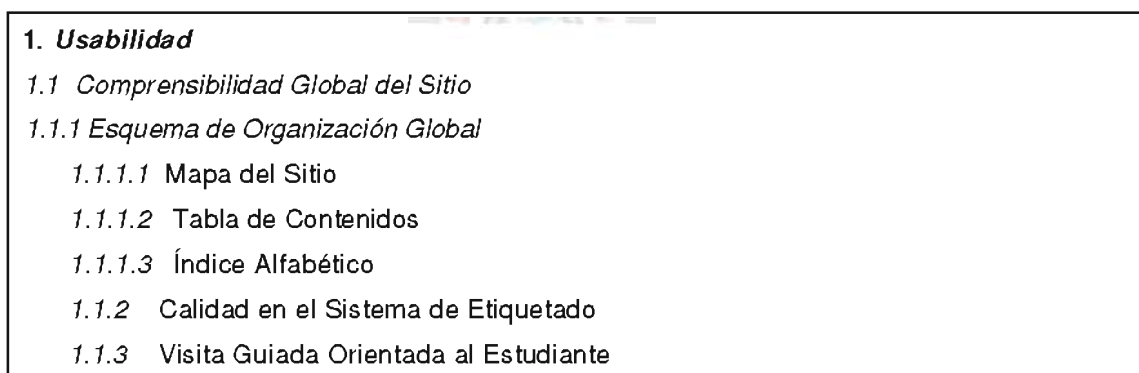
#### **3.1 No Deficiencia**

##### **3.1.1 Errores de Enlaces**



Fuente: [OLS1999]

Figura 2.15 Árbol de requerimientos de calidad para Usabilidad y Funcionalidad



<ul style="list-style-type: none"><li>1.1.4 Mapa de Imagen (Campus/Edificio)</li><li>1.2 <i>Mecanismos de Ayuda y Retroalimentación en línea</i><ul style="list-style-type: none"><li>1.2.1 <i>Calidad de la Ayuda</i><ul style="list-style-type: none"><li>1.2.1.1 Ayuda Explicatoria Orientada al Estudiante</li><li>1.2.1.2 Ayuda de la Búsqueda</li></ul></li><li>1.2.2 <i>Indicador de Última Actualización</i><ul style="list-style-type: none"><li>1.2.2.1 Global (de todo el sitio Web)</li><li>1.2.2.2 Restringido (por subsitio o página)</li></ul></li><li>1.2.3 <i>Directorio de Direcciones</i><ul style="list-style-type: none"><li>1.2.3.1 Directorio E-mail</li><li>1.2.3.2 Directorio TE-Fax</li><li>1.2.3.3 Directorio Correo Postal</li></ul></li><li>1.2.4 Facilidad FAQ</li><li>1.2.5 <i>Retroalimentación</i><ul style="list-style-type: none"><li>1.2.5.1 Cuestionario</li><li>1.2.5.2 Libro de Invitados</li><li>1.2.5.3 Comentarios/Sugerencias</li></ul></li><li>1.3 <i>Aspectos de Interfaces y Estéticos</i><ul style="list-style-type: none"><li>Cohesividad al Agrupar los Objetos de Control Principales</li><li><i>Permanencia y Estabilidad en la Presentación de los Controles Principales</i></li><li>Permanencia de Controles Directos</li><li>Permanencia de Controles Indirectos</li><li>Estabilidad</li><li>1.3.3 <i>Aspectos de Estilo</i><ul style="list-style-type: none"><li>Uniformidad en el Color de Enlaces</li><li>Uniformidad en el Estilo Global</li><li>Guía de Estilo Global</li></ul></li><li>1.3.4 Preferencia Estética</li></ul></li><li>1.4 <i>Misceláneas</i><ul style="list-style-type: none"><li>Soporte a Lenguaje Extranjero</li><li>Atributo "Qué es lo Nuevo"</li><li>Indicador de Resolución de Pantalla</li></ul></li></ul></li></ul>
<ul style="list-style-type: none"><li><b>2. <i>Funcionalidad</i></b><ul style="list-style-type: none"><li>2.1 <i>Aspectos de Búsqueda y Recuperación</i><ul style="list-style-type: none"><li>2.1.1 <i>Mecanismo de Búsqueda en el Sitio Web</i></li></ul></li></ul></li></ul>

*2.1.1.1 Búsqueda Restringida*

De Personas

De Cursos

de Unidades Académicas

*2.1.1.2 Búsqueda Global*

*2.1.2 Mecanismos de Recuperación*

Nivel de Personalización

Nivel de Retroalimentación en la Recuperación

*2.2 Aspectos de Navegación y Exploración*

*2.2.1 Navegabilidad*

*2.2.1.1 Orientación*

Indicador del Camino

Etiqueta de la Posición Actual

*2.2.1.2 Promedio de Enlaces por Página*

*2.2.2 Objetos de Control Navegacional*

*2.2.2.1 Permanencia y Estabilidad en la Presentación de los Controles Contextúales (Subsitio)*

Permanencia de los Controles Contextúales

Estabilidad

*2.2.2.2 Nivel de Desplazamiento*

Desplazamiento Vertical

Desplazamiento Horizontal

*2.2.3 Predicción Navegacional*

Enlace con Título (enlace con texto explicatorio)

Calidad de la Frase del Enlace

*2.3 Aspectos del Dominio orientados al Estudiante*

*2.3.1 Relevancia de Contenido*

*2.3.1.1 Información de Unidad Académica*

Índice de las Unidades

Sub-sitios de las Unidades

*2.3.1.2 Información de Inscripción*

*2.3.1.2.1 Información de los Requerimientos de Ingreso/Admisión*

*2.3.1.2.2 Formulario para Rellenar/Bajar*

*2.3.1.3 Información de Carreras*

Índice de Carreras

Descripción de Carrera

Plan de Carrera/Oferente de Cursos



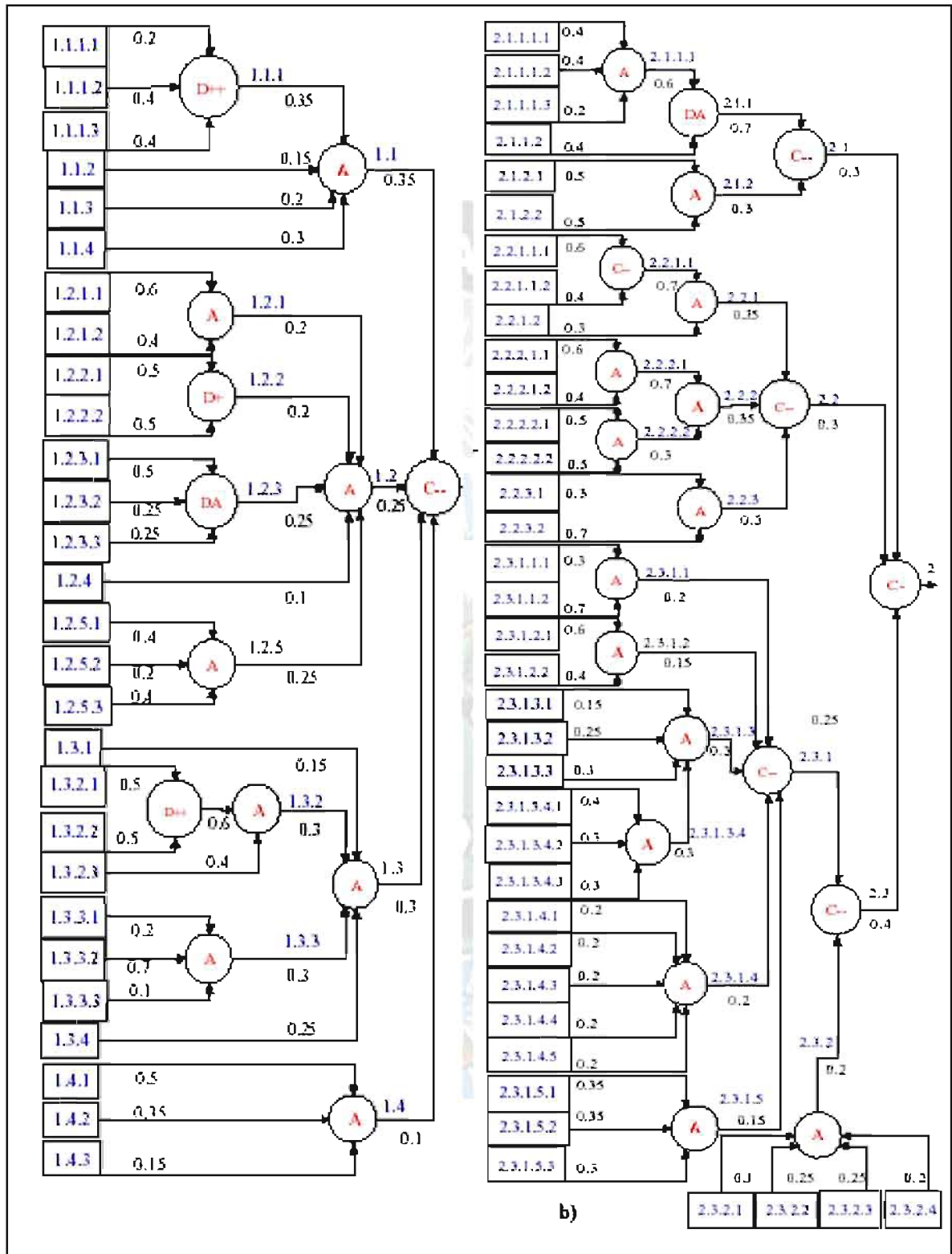
<p><i>Descripción de Cursos</i></p> <p>Comentarios</p> <p>Programa Sinté./Anal.</p> <p>Programación Cursos</p> <p><i>2.3.1.4 Información de Servicios al Estudiante</i></p> <p>índice de Servicios</p> <p>Información de Salud</p> <p>Información de Becas</p> <p>Información de Residencias</p> <p>Información Cultural/Deport,</p> <p><i>2.3.1.5 Información de Infraestructura Académica</i></p> <p>    2.3.1.5.1 Información de Bibliotecas</p> <p>    2.3.1.5.2 Información de Laboratorios</p> <p>    2.3.1.5.3 Información Resultados I+D</p> <p><i>2.3.2 Servicios On-line</i></p> <p>2.3.2.1 Información Aranceles, Aprobación de Cursos.</p> <p>2.3.2.2 Servicio de Páginas Web</p> <p>2.3.2.3 Servicio FTP</p> <p>2.3.2.4 Servicio de Grupo de Noticias</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: [OLS1999]

## **2.19.1 ESTRUCTURA DE AGREGACIÓN DE PREFERENCIAS PARCIALES**

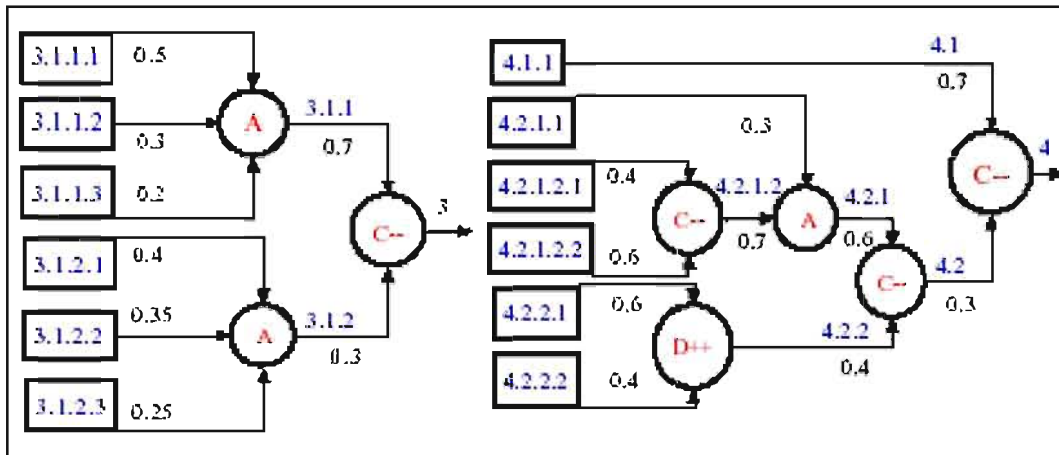
En este punto se realiza una descripción de las características, subcaracterísticas y atributos. Para determinar el valor de la variable "peso"  $p_i$  (que se utilizará para calcular los indicadores de calidad Global (IGi) de cada característica y atributo del árbol de requerimientos de calidad, en el presente proyecto se tomara como referencia los valores que propone el Dr. Olsina en sus casos de estudio de sitios Web académicos. En las siguientes figuras se muestra las estructuras de agregación de preferencias parciales. A continuación se muestra la estructura de agregación para la característica Usabilidad, la estructura de agregación para la característica Funcionalidad (ver Figura 2.16), para la característica Confiabilidad, y por último para la Eficiencia.(Ver Figura 2.17).

Figura 2.16 Estructura de agregación para la Usabilidad y Funcionalidad



Fuente: [OLS1999].

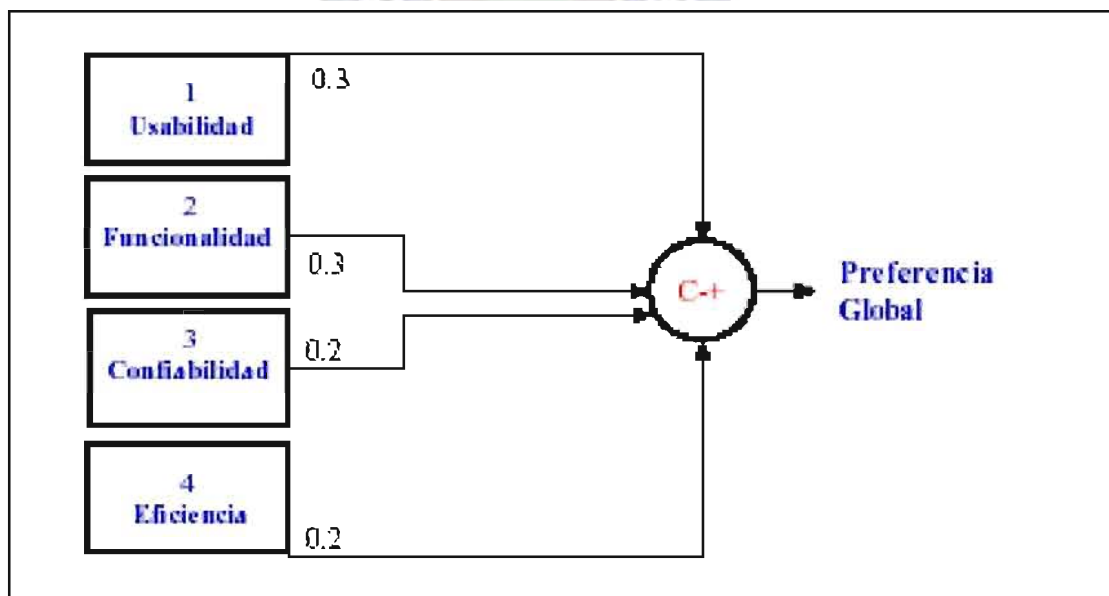
Figura 2.17 Estructura de agregación para la Confiabilidad y Eficiencia



Fuente: [OLS1999].

En la Figura 2.18 se muestra los valores ponderados de los pesos  $P_i$  (0.3, 0.3, 0.2, 0.2) de las características de más alto nivel, los cuales son útiles para calcular el indicador de calidad global IG. Esta estructura producida es útil para cuantificar el indicador de calidad global IG para el presente proyecto.

Figura 2.18 Estructura de Agregación de preferencias parciales



Fuente: [OLS1999].

## 2.19.2 PREFERENCIAS ELEMENTALES

A partir del árbol de calidad (ver Figura 2.15) antes esquematizado, la idea central es determinar los valores de las variables de preferencia de calidad elemental ( $IE_i$ ) para cada atributo  $A_i$  (hojas de calidad), es importante mencionar que cada atributo  $A_i$  tendrá asociada una variable  $X_i$ , que tomará un valor real a partir de un proceso de medición el cual producirá un valor de  $IE_i$  que interpreta el porcentaje del requerimiento satisfecho.

Se utiliza el criterio binario y absoluto donde se pregunta si esta disponible o no esta disponible, es decir:

$X=0$  si el atributo de calidad esta ausente  $\rightarrow IE_i = 0 \%$

$X=1$  si el atributo de calidad esta presente  $\rightarrow IE_i = 100 \%$

Los valores obtenidos  $IE_i$  será la base principal para la evaluación global, a través de la función de la media potencia pesada (ver ecuación 1)

$$IG(r) = (P_1 IE_1^r + P_2 IE_2^r + \dots + P_m IE_m^r)^{1/r} \quad (\text{Ecuación 1})$$

Donde:

$$\begin{aligned} r &\leq +\infty; \quad 0 \leq IE_i \leq 1 \\ (P_1 + P_2 + \dots + P_m) &= 1; \quad P_i > 0; \quad i = 1 \dots m \end{aligned}$$

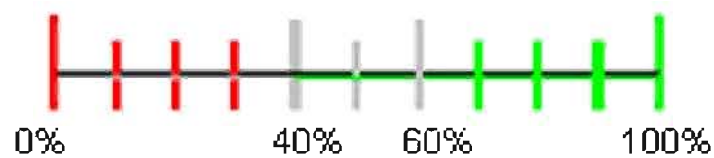
$P_i$  = representa al peso de cada elemento del árbol de requerimientos; estos valores fueron asignados ya por el autor de la metodología web site QEM Dr. Olsina (ver Figuras 2.17,2.18).

$r$  = es el exponente de la ecuación (1), guarda el valor real conforme al operador lógico y a la cantidad de entradas seleccionadas para una función de agregación dada.

A continuación podemos mencionar el rango de aceptación del modelo de calidad:

Figura 2.19 Rango de aceptación de calidad

Satisfactorio	[%]	$60 < IE \leq 100$
Insatisfactorio	[%]	$0 < IE \leq 40$
Marginal	[%]	$40 < IE \leq 60$



Fuente: [OLS1999].

## 2.20 ESTIMACIÓN DEL COSTO DEL PROYECTO

Existen una serie de métricas propuestas por la **Ingeniería del Software** para determinar el esfuerzo de un proyecto, el alcance del mismo y la productividad de sus programadores. Vamos a aplicar algunas de las mismas a este desarrollo, para calibrar su dificultad y rendimiento obtenido.

### 2.20.1 Las métricas orientadas al tamaño

La métrica del software es un factor realmente importante en el análisis de un proyecto. Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso por el cual se desarrolla. Se basan en la medición del número de Líneas De Código - LDC - que contiene el desarrollo,

entendiendo por línea de código una sentencia del lenguaje de programación (se excluyen comentarios y líneas en blanco de los fuentes).

Una forma de clasificarlos es atendiendo al número de líneas de código, como se muestra en la tabla 2.3

Tabla 2.3 Categoría de un Proyecto en Función de sus Líneas de Código

<b>Categoría</b>	<b>Programadores</b>	<b>Duración</b>	<b>Líneas de código</b>	<b>Ejemplo</b>
Trivial	1	0 – 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 – 6 semanas	1k- 3k	Biblioteca de funciones
Media	2-5	0,5 – 2 semanas	3k - 50k	Compilador de C
Grande	5-20	2 – 3 años	50k- 100k	SO pequeño
Muy grande	100-1000	4 – 5 años	100k- 1M	Grandes SO
Gigante	1000 - 5000	5 – 10 años	> 1M	Sistema de Distribución

Fuente: [COT1994]

Si en un determinado proyecto en este caso digamos el proyecto SpiderBot en el cual se han generado más de 8.200 líneas de código, esto quiere decir que nos enfrentamos a un proyecto software con una clasificación de complejidad media, para el cual se necesitarían de 2 a 5 programadores trabajando de medio año a 2 años.

## 2.20.2 EL MÉTODO COCOMO II

Una metodología que se encarga de medir proyectos software es COCOMO. La metodología COCOMO (CONstructive COst MOdel) se debe a Barry Boehm, y está orientada a líneas de código.

Hay una jerarquía de modelos COCOMO: básico, intermedio y avanzado, la cual se aplica a tres tipos diferentes de software [BWB1990]:

1. **Orgánico**; proyectos relativamente sencillos, menores de 50.000 líneas de código. Se tiene experiencia en proyectos similares y se encuentra en un entorno estable.
2. **Semiacoplado**: proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
3. **Empotrado**: proyectos bastante complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Se trabaja con unos requisitos muy restrictivos y de gran volatilidad

Dado que sólo se va a emplear una variable para la estimación (la línea de código), se empleará COCOMO básico, ya que es un modelo uní variable estático, con lo que se obtiene una **valoración objetiva** del esfuerzo realizado. Este proyecto será considerado como software orgánico, ya que posee menos de 50.000 líneas de código.

La ecuación del esfuerzo de COCOMO básico tiene la siguiente forma:

$$E = \text{Esfuerzo} = a KLDC^b \text{ (persona x mes)}$$

Donde KLDC es el número de líneas de código, distribuidas en millares, para el proyecto.

La ecuación del tiempo de desarrollo es:

$$T = \text{Tiempo de duración del desarrollo} = c \text{ Esfuerzo}^d \text{ (meses)}$$

Por su parte los coeficientes a, b, c y d se obtienen empíricamente del estudio de una serie de proyectos, y sus valores son:

Tabla 2.4 Coeficientes COCOMO

Proyecto de software	A	B	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Fuente: [BWB1990]

En el desarrollo de SpiderBot se han codificado 8,2 miles de líneas de código.

$$\text{Esfuerzo realizado} = 2,4 * 8,2^{1,05} = 21,9 \text{ personas-mes}$$

$$T = 2,5 * 21,9^{0,38} = 8,1 \text{ mes [BWB1990]:}$$

$$N^{\circ} \text{ de personas para desarrollar el proyecto} = E/T = 21,9 / 8,1 = 3 \text{ personas}$$

#### - La controversia: Líneas de código frente a puntos de función

Existe en el mundo de la Ingeniería del Software una viva polémica sobre qué tipo de métricas son mejores para evaluar un proyecto: las orientadas a tamaño o las que utilizan puntos de función.



El centro de controversia está en considerar las líneas de código como medida clave, ya que los que se oponen a su uso, aducen que las medidas basadas en líneas de código son dependientes del lenguaje de programación.

En cualquier caso esta polémica queda apartada gracias a Casper Jones, quién creó la siguiente tabla 2.4 de correspondencia entre algunos de los lenguajes de programación más conocidos con su número de equivalencia entre líneas de código por punto de función:

Tabla 2.5 Conversión Líneas de Código a Puntos de Función

Lenguaje	LDC / PF
Ensamblador	320
C	150
Cobol	106
Pascal	91
Basic	64
TCL	64
java	53
PHP	29
C++	29

Fuente: [COT1994]

# CAPITULO 3

## ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se efectuara el análisis y diseño correspondiente al sistema tomando en cuenta las necesidades comunes en las que puedan coincidir estas cinco unidades educativas, utilizando los conceptos mencionados en el marco teórico.

### 3.1 FASE DE ESPECULACION

Esta es la fase inicial del ciclo de vida ASD, que comienza con la Especificación esta consiste en el inicio del proyecto y la planificación de las características del mismo, para este fin sin duda el primer paso es la captura de requisitos, para posteriormente realizar una planificación adecuada del proyecto. Para lo cual se utilizaron herramientas del lenguaje de modelado unificado "UML", estas son los Diagramas de casos de uso, diagramas de secuencia y los diagramas de colaboración que se describieron en el capítulo anterior

#### 3.1.1 ANÁLISIS DEL PROCESO ACADÉMICO

Luego de las entrevistas efectuadas a las Unidades Educativas se logro distinguir una base de administración conformada a su vez por tres áreas: administrativa, contable y académica

- **Área administrativa.-** La cual esta encargada de la organización , de efectuar el control de pagos, contrataciones, pensiones, otros pagos, control de los deudores, becas, agendas, transporte, biblioteca, enfermería, deportes y otros servicios

- **Área contable.-** La cual cumple la función del manejo y registro de los recursos de la institución
- **Área académica.-** El cual se encarga del correcto manejo de los alumnos en cuanto al grado educativo en el que se encuentra, rendimiento, materias destacadas, reprobadas, aprobadas, la asignación de horarios, actualizaciones de notas, seguimiento de inscritos, historial de los alumnos. En el presente proyecto se trabajara con el área académica en la cual se efectuó el análisis correspondiente para iniciar el mismo:

Se identifico las siguientes actividades que realizan cada una de las Unidades Educativas con respecto al área académica que estas manejan

- **NUEVA GESTIÓN.-** En la cual se debe comenzar la nueva gestión dando lugar a la recepción de nuevos estudiantes
- **ESTADÍSTICAS DE LA GESTIÓN ANTERIOR.-** Esta implica las siguientes actividades: se toma la nomina general de los alumnos antiguos de la institución educativa, se verifica el nuevo año en curso, se controlan los alumnos antiguos y graduados de la institución educativa
- **ADECUACIÓN Y ASIGNACIÓN.-** Aquí es donde mayormente existen cambios para cada gestión y además son diferentes en cada Unidad Educativa. Se toman en cuenta los siguientes puntos: se asignan aulas, materias y horarios; se asignan docentes y materias, se revisan los horarios de forma general y por cursos , se asignan cupos por curso
- **INSCRIPCIONES.-** Las inscripciones implican varias actividades estas son: se habilitan las inscripciones para alumnos antiguos, se habilita el registro e inscripción de nuevos alumnos, se habilitan las inscripciones para alumnos nuevos, se le asigna un código de única identificación, se registra la fecha y

hora de cada inscripción, se verifica el proceso de inscripción en cada curso, se cierran las inscripciones, se verifica el resultado final de las inscripciones

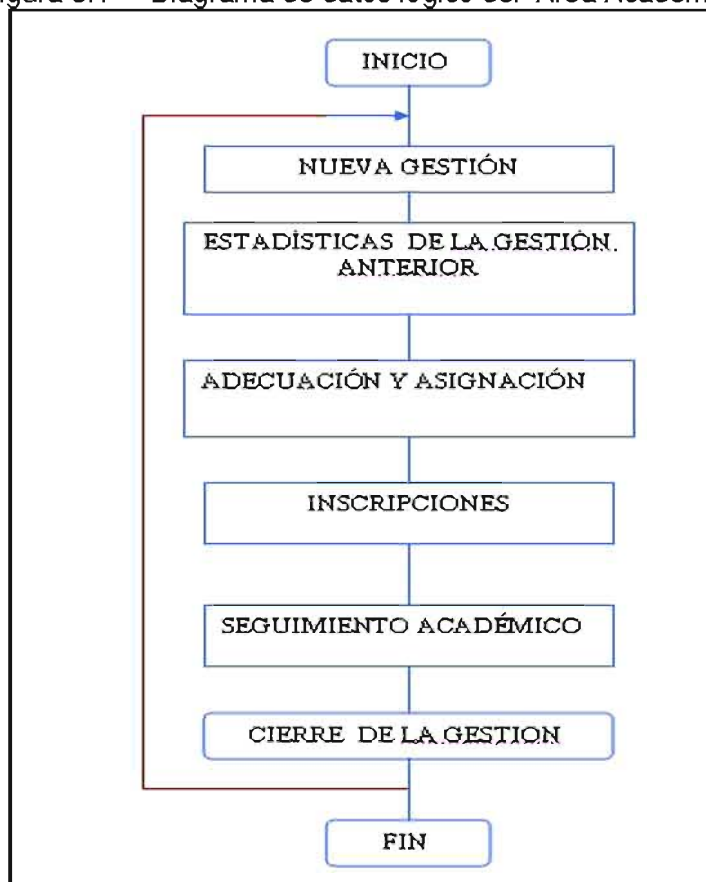
- **SEGUIMIENTO ACADÉMICO.-** Este es el eje torcal de las Unidades Educativas en la cual se notan las siguientes actividades: se emite la nomina de alumnos por curso a cada docente, se actualizan las notas de los alumnos por cada trimestre, se emite la calificación trimestral por alumno, se actualiza la nota final de los alumnos, se controla los alumnos que aprobaron el curso, los que reprobaron el curso y los graduados de la Unidad Educativa, se efectúa la nomina de rezagados.
- **CIERRE DE LA GESTIÓN .-** Esta implica la verificación del estado final del rendimiento académico de los alumnos miembros de la Unidad Educativa
- **CONTROL DE ASISTENCIA DE DOCENTES.-** Se hace el control de hora de llegada y salida de cada docente donde se identifica los retrasos y faltas que puedan tener alguno de los docentes

### **3.1.2 DIAGRAMA DE FLUJO DE DATOS LÓGICO**

Este diagrama nos permite observar el flujo real de datos que ocurre en el proceso académico de las Unidades Educativas (Figura 3.1).

En el Área académica se logro observar una característica importante que es un ciclo de retro alimentación al finalizar e iniciar una nueva gestión académica, como se distingue en el siguiente diagrama de flujo de datos lógico:

Figura 3.1 Diagrama de datos lógico del Área Académico

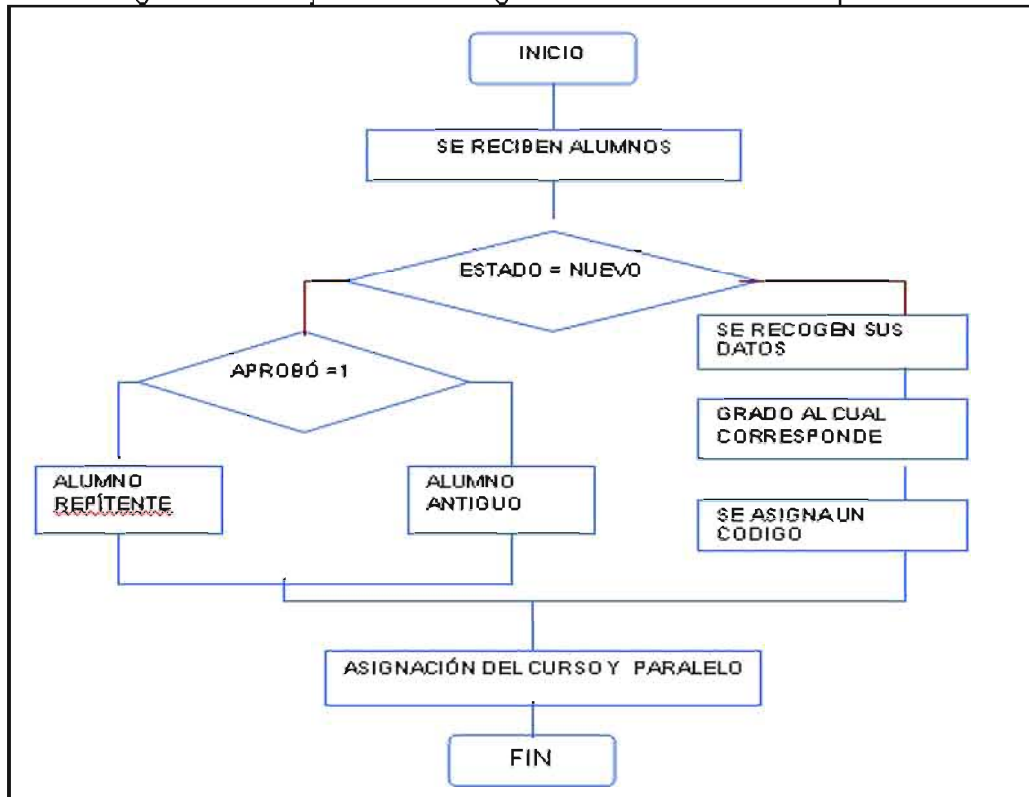


Fuente: Elaboración Propia

En esta área el proceso con mas retardo es el de inscripciones puesto que acá el registro de alumnos nuevos el muy lento y esto ocurre en todas las Unidades Educativas.

El presente sistema en su fase de Especulación toma en cuenta las características de inscripción de todas las unidades educativas y se logro obtener un flujo de datos lógico similar en cada una de estas, la cual la podemos observar en la Figura 3.2 . En el proceso de inscripción presenta dos criterios para incorporar alumnos los cuales pueden ser alumnos que ya estaban inscritos que resultan ser antiguos y aquellos que se inscriben por primera ves son los nuevos.

Figura 3.2 Flujo de Datos Lógico del Proceso de Inscripciones



Fuente: Elaboración Propia

### 3.1.3 CAPTURA DE REQUISITOS

La captura de requisitos es el proceso de averiguar normalmente en circunstancias difíciles lo que se debe construir. Los desarrolladores y clientes deben acordar qué es lo que el sistema debe hacer. Para identificar los requisitos funcionales del sistema utilizaremos los casos de uso, que se describen como conjuntos de secuencias.

Cada una de estas secuencias refleja la interacción entre los elementos externos al sistema y el propio sistema (se trata de la descripción de escenarios o situaciones posibles donde se pone de relieve el comportamiento del sistema ante su uso por parte del usuario). Para la óptima captura de requisitos, el

proyecto se afianzará del modelado de casos de uso. Así pues, los objetivos principales de la realización de casos de uso son:

Definir el límite entre el sistema a desarrollar y los elementos externos a ese sistema (actores usuarios del sistema).

Capturar el conjunto de funcionalidades y comportamientos del sistema a desarrollar.


Cada caso de uso se documenta mediante una representación gráfica y un texto con la descripción de las situaciones o escenarios ante los que el usuario se pueda encontrar en su interacción con el sistema.




### 3.1.4 MODELO DE CASOS DE USO

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones.



A continuación se describen las características de los actores identificados en las Unidades Educativas con las que se trabajo.

Tabla 3.1 Actores de las Unidades Educativas

ACTORES	DESCRIPCIÓN
	<p><b>Administrador de Sistema.</b> Este usuario será el encargado de realizar transacciones en el sistema, efectuar cambios de configuración, levantar y bajar servicios, asignar enlaces, acceder a la información estadística y monitorear el sistema.</p>

	<p><b>Estudiante.</b> Es un usuario que accederá al sistema con más frecuencia, para esto tendrá asignado una contraseña que le permitirá realizar su inscripción y consultas.</p>
	<p><b>Kardixa.</b> Es un usuario que accederá al sistema mediante una contraseña que le será asignado, con el cual obtendrá y registrará datos e información del estudiante para almacenar en el respectivo file académico.</p>
	<p><b>Docente.</b> Es el usuario que recibe información del sistema sobre los estudiantes inscritos en las materias que le corresponde, horarios. Registrará las notas de los estudiantes respectivos, visualiza el reporte de pre-actas y actas de finales de notas, para esto se le asignará una contraseña.</p>



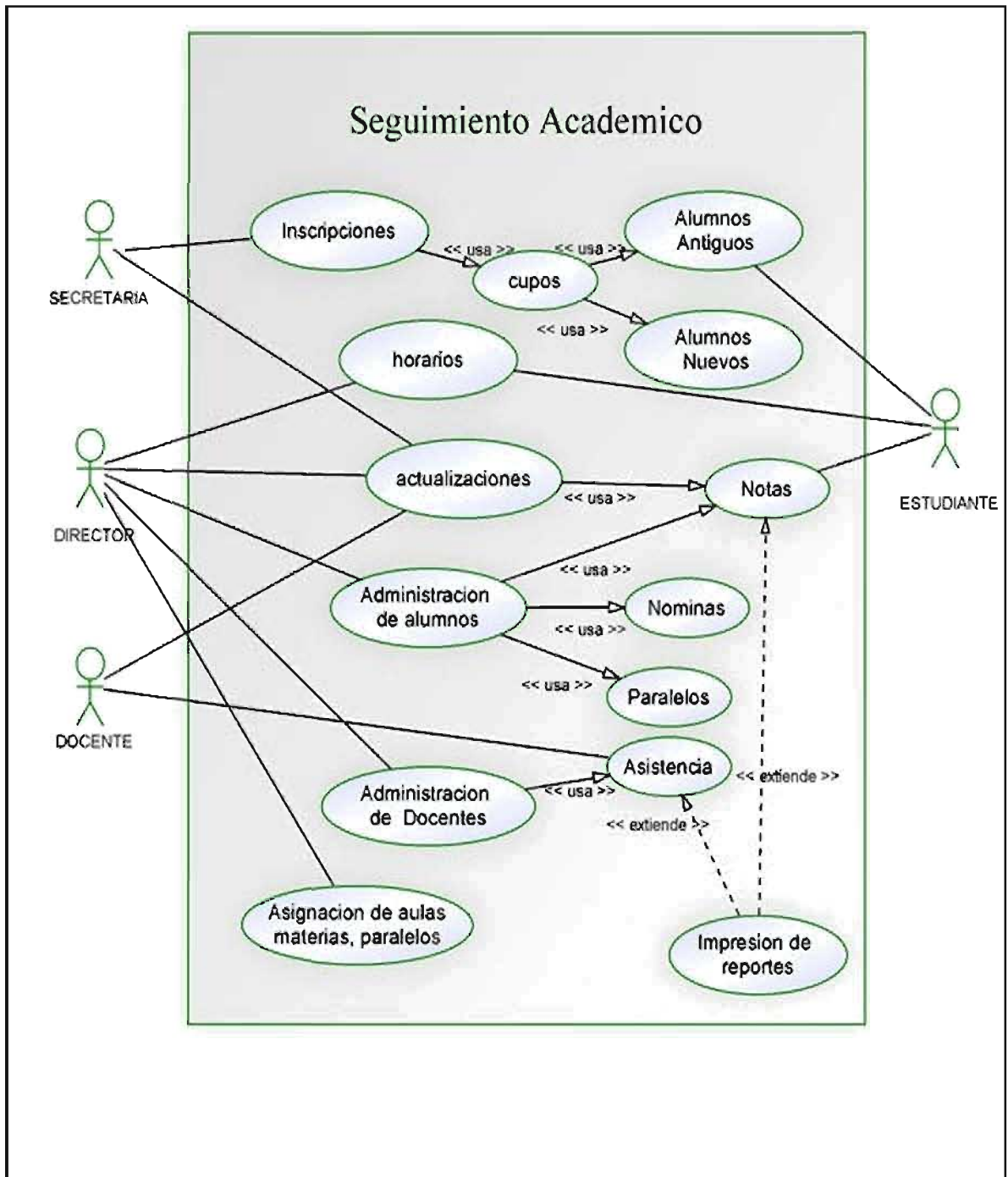
	<p><b>Director.</b> Este usuario accederá a la información estadística del sistema, podrá realizar consultas con respecto a la población estudiantil e intervendrá en casos irregulares que surjan en cualquier proceso. También obtendrá información del sistema en cuanto al rendimiento académico y asistencia de docentes. Es quien habilita materias en la gestión académica y controla la emisión de las actas de notas.</p>
	<p><b>Secretaria General.</b> Este usuario accederá a la información académica ya sea para verificar o registrar ciertos datos para llevar a cabo ciertos trámites. Es quien genera e imprime las actas de notas y encargada de las inscripciones de alumnos antiguos y nuevos.</p>

Fuente: Elaboración Propia

### 3.1.5 DIAGRAMA DE CASOS DE USO

Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema. A continuación se muestran los casos de uso principales agrupados por paquetes y la descripción narrativa de cada caso de uso.

Figura 3.3 Diagrama de Casos de Uso del Seguimiento Académico



Fuente: Elaboración Propia

Nombre	<b>ACTUALIZACIONES</b>	
Actores	Él actor que interviene en este <b>caso</b> es: Director, Docentes, Secretaria	
Propósito	Actualizar notas vía Web o diskett	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de actualizar	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema. 3. El usuario ingresa código clave. 5. El usuario ingresa las notas	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información 6. El sistema registra la notas de acuerdo al tipo de ingreso que se desee
Precondición	El sistema tiene almacenada las nominas de alumnos de cada curso	
Poscondición	Ninguna	

Nombre	<b>HORARIOS</b>	
Actores	Él actor que interviene en este <b>caso</b> es: Director, Estudiantes	
Propósito	Registrar y actualizar los horarios con los cuales contara la Unidad Educativa y para el estudiante consultar su horario	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de horarios	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema. 3. El usuario ingresa código clave. 5. El usuario ingresa los datos del horario	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información 6. El sistema registra la información del nuevo horario
Precondición	El sistema tiene almacenada los periodos	
Poscondición	Ninguna	

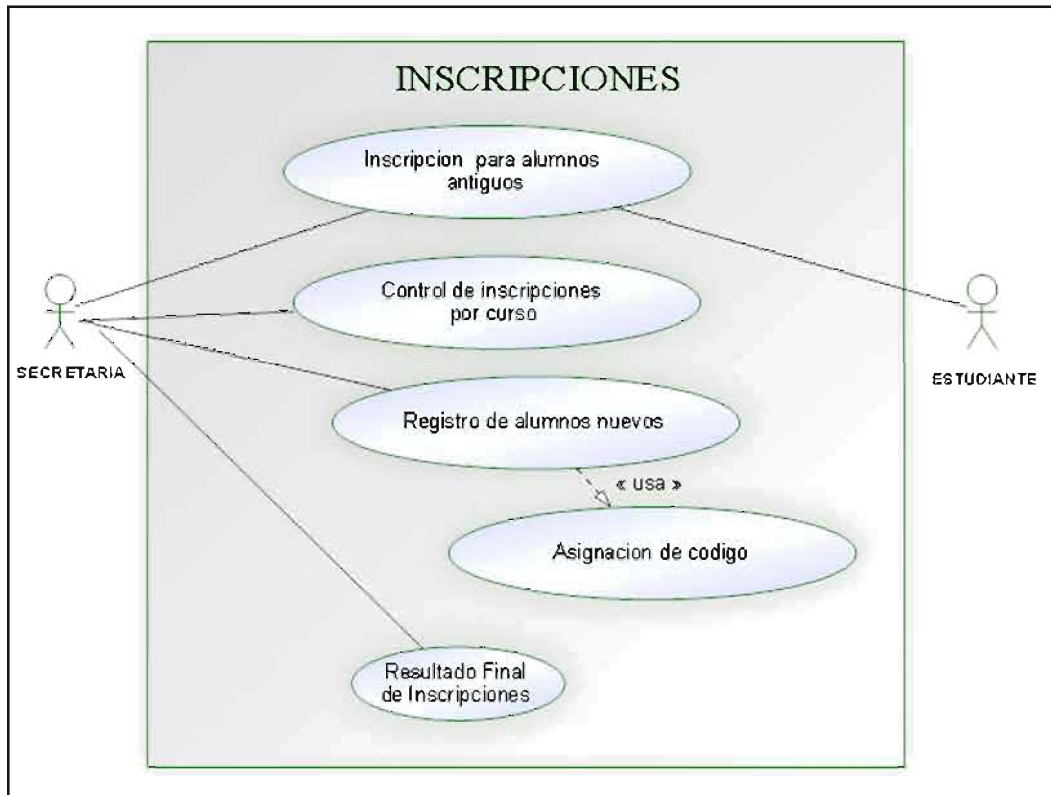
Nombre	<b>ASIGNACIÓN</b>	
Actores	El actor que interviene en este caso es: Director	
Propósito	Realizar asignaciones o modificaciones tanto de paralelos, materias, aulas y cupos	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de asignación	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema. 3. El usuario ingresa código clave. 5. El usuario elige la opción	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información 6. El sistema realiza la opción deseada
Precondición	El sistema tiene datos de los cursos y paralelos con los cuales cuenta la Unidad Educativa	
Poscondición	Ninguna	

Nombre	<b>INSCRIPCIÓN</b>	
Actores	El actor que interviene en este <b>caso</b> es: Secretaria	
Propósito	Registrar los datos del nuevo alumno e incorporarlo en el curso que corresponda y paralelo	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de inscripción para antiguos como nuevos	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema. 3. El usuario ingresa código clave. 5. El usuario ingresa los datos	2. El sistema le solicita código y clave. 4 El Sistema autentifica código y clave y las casillas para introducir información 6. El sistema registra la información del nuevo alumno
Precondición	El sistema tiene almacenada la información de los cursos y paralelos existentes	
Poscondición	Ninguna	

Nombre	<b>ADMINISTRACIÓN DE DOCENTES</b>	
Actores	El actor que interviene en este <b>caso</b> es: Director	
Propósito	Controlar la asistencia de los docentes como consultar la información	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de administración de docentes	
	Evento Actor	Evento Sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema.</li> <li>3. El usuario ingresa código clave.</li> <li>5. El usuario ingresa la opción</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema le solicita código y clave.</li> <li>4 El Sistema autentifica código y clave y las casillas para introducir información</li> <li>6. El sistema busca la información del docente</li> </ol>
Precondición	El sistema tiene almacenada la información de los docentes	
Poscondición	Ninguna	

Nombre	<b>ADMINISTRACIÓN DE ALUMNOS</b>	
Actores	El actor que interviene en este <b>caso</b> es: Director	
Propósito	Realizar cambios tales como la información del alumno ,notas, además de realizar consultas del rendimiento	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitudes de administración de alumnos	
	Evento Actor	Evento Sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario ingresa al sistema.</li> <li>3. El usuario ingresa código clave.</li> <li>5. El usuario elige la opción</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema le solicita código y clave.</li> <li>4 El Sistema autentifica código y clave y las casillas para introducir información</li> <li>6. El sistema realiza la opción deseada</li> </ol>
Precondición	El sistema tiene almacenada la información de los alumnos	
Poscondición	Ninguna	

Figura 3.4 Casos de Uso del Proceso de inscripciones



Fuente: Elaboración Propia

Nombre	<b>INSCRIPCIÓN PARA ALUMNOS ANTIGUOS</b>	
Actores	Él actor que interviene en este caso es: Estudiante, Secretaria	
Propósito	Confirmar que el alumno seguirá en la Unidad Educativa	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitud de inscripción	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema. 3. El usuario ingresa código clave. 5. El usuario confirma que el alumno seguirá estudiando	2. El sistema le solicita código y clave del alumno. 4 El Sistema autentifica código y clave para verificar si el alumno existe 6. El sistema registra la inscripción en el curso que le corresponda
Precondición	El sistema tiene almacenada las nominas de alumnos de cada curso	
Poscondición	Ninguna	

Nombre	<b>REGISTRO DE ALUMNOS NUEVOS</b>	
Actores	El actor que interviene en este <b>caso</b> es: Secretaria	
Propósito	Ingresar la información del alumno a la BD	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitud de inscripción	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema.  3. El usuario ingresa código clave.  5. El usuario ingresa lops datos del alumno	2. El sistema le solicita código y clave del usuario.  4 El Sistema autentifica código y clave para verificar autenticidad  6. El sistema almacena la información en la BD
Precondición	El sistema acepta el ingreso de alumnos nuevos	
Poscondición	Ninguna	

Nombre	<b>CONTROL DE INSCRIPCIONES POR CURSO</b>	
Actores	El actor que interviene en este <b>caso</b> es: Secretaria	
Propósito	Asignar al alumno un curso y paralelo que no este lleno	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitud de inscripción	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema.  3. El usuario ingresa código clave.  5. El usuario revisa que curso y paralelo tiene menos alumnado	2. El sistema le solicita código y clave del usuario.  4 El Sistema autentifica código y clave para verificar autenticidad  6. El sistema prepara el curso y paralelo al que será asignado el alumno
Precondición	El sistema tiene almacenada las nominas de alumnos de cada curso	
Poscondición	Ninguna	

Nombre	<b>RESULTADO FINAL DE INSCRIPCIONES</b>	
Actores	Él actor que interviene en este <b>caso</b> es: Secretaria	
Propósito	Ver la cantidad de inscritos por curso y de manera general	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitud de inscripción	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema.  3. El usuario ingresa código clave.  5. El usuario consulta el modulo de inscripciones	2. El sistema le solicita código y clave del usuario.  4 El Sistema autentifica código y clave para verificar autenticidad  6. El sistema muestra numéricamente los inscritos por paralelo y de manera general
Precondición	Tener registrado a los alumnos inscritos	
Poscondición	Ninguna	

Nombre	<b>ASIGNACIÓN DE CÓDIGO</b>	
Actores	Él actor que interviene en este <b>caso</b> es: Secretaria	
Propósito	Asignar un código para identificar a los alumnos	
Resumen	El sistema deberá comportarse como se describe en el presente caso de uso cuando reciba las solicitud de inscripción	
	Evento Actor	Evento Sistema
Flujo principal	1. El usuario ingresa al sistema.  3. El usuario ingresa código clave.  5. El usuario inscribe al alumno	2. El sistema le solicita código y clave del usuario.  4 El Sistema autentifica código y clave para verificar autenticidad  6. El sistema le asigna un código de acuerdo a sus apellidos y la fecha de nacimiento
Precondición	No permite códigos repetidos	
Poscondición	Ninguna	

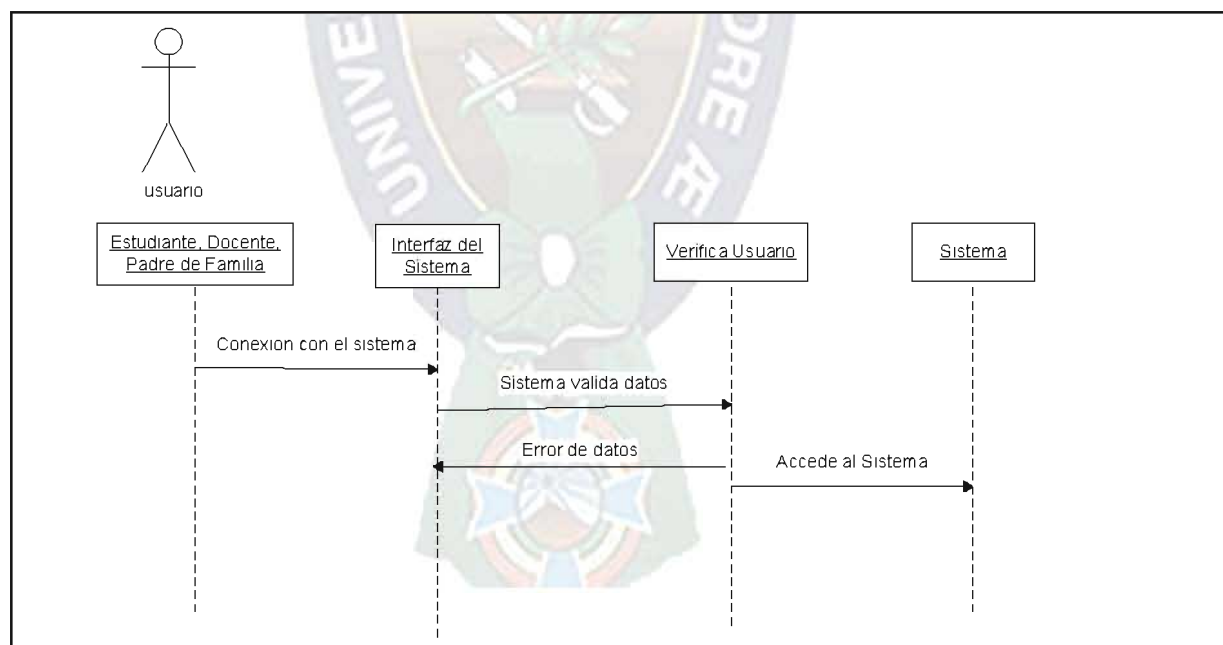


### 3.1.6 DIAGRAMAS DE SECUENCIA Y COLABORACIÓN

Los diagramas de secuencia muestran de manera gráfica los eventos u operaciones del sistema, como es que este responde a alguna determinada operación y con la aplicación de los diagramas de colaboración se tiene un enfoque más claro de la relación entre las operaciones, lo cual nos ayudara en la etapa de implementación.

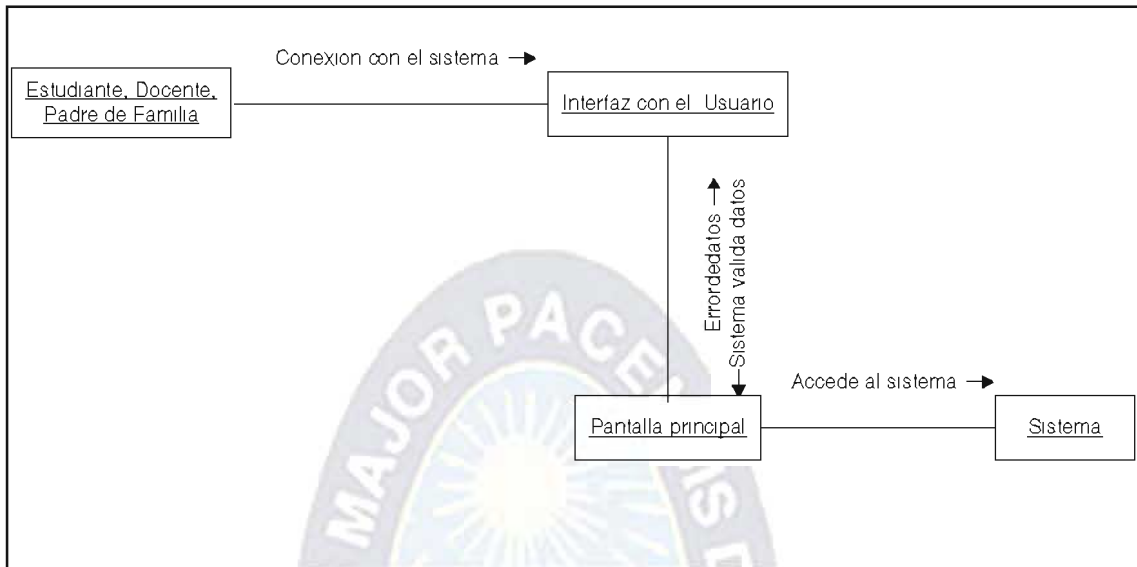
Con lo desarrollado hasta ahora se tiene una perspectiva como para poder elaborar los diagramas de secuencia y sus respectivos diagramas de colaboración, a continuación se detallarán las siguientes figuras más importantes: Es importante el control de acceso del usuario como muestra en la *Figura. 3.5*

Figura 3.5 Secuencia Correspondiente al Control de Acceso del Usuario



Fuente: Elaboración Propia

Figura 3.6 Diagrama de Colaboración Correspondiente al Control de Acceso del Usuario



Fuente: Elaboración Propia

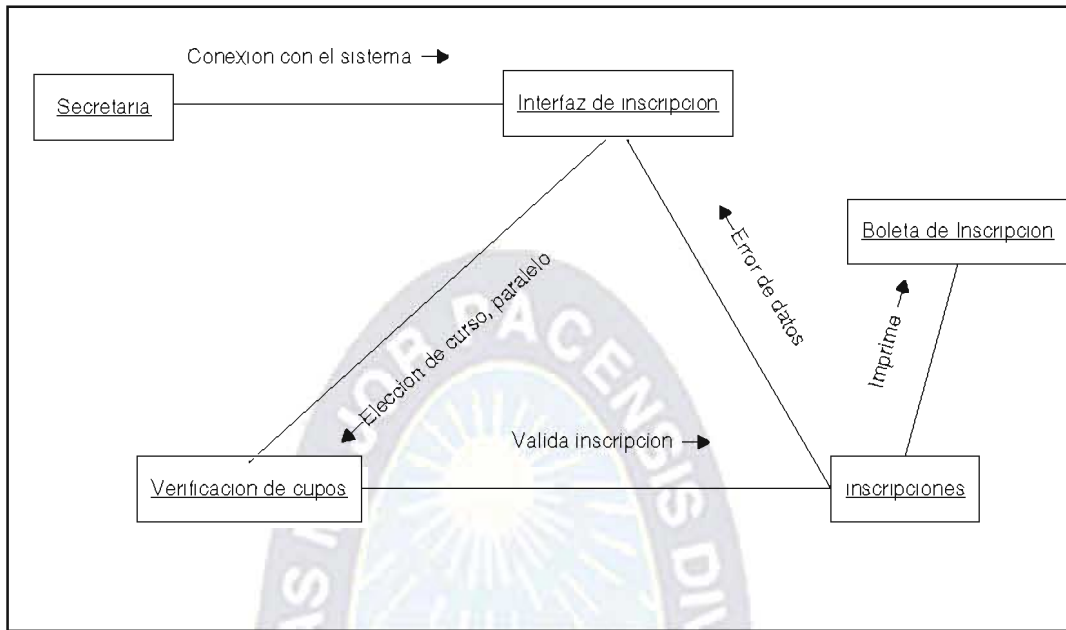
La secuencia correspondiente a la inscripción se inicia cuando el padre de familia o apoderado solicita la inscripción de su hijo. Una vez realizada la inscripción, la persona encargada de realizar este trabajo es la que debe grabar la inscripción para luego emitir la boleta de inscripción como se puede observar en la Figura. 3.7

Figura 3.7 Secuencia Correspondiente a la Inscripción de Alumnos



Fuente: Elaboración Propia

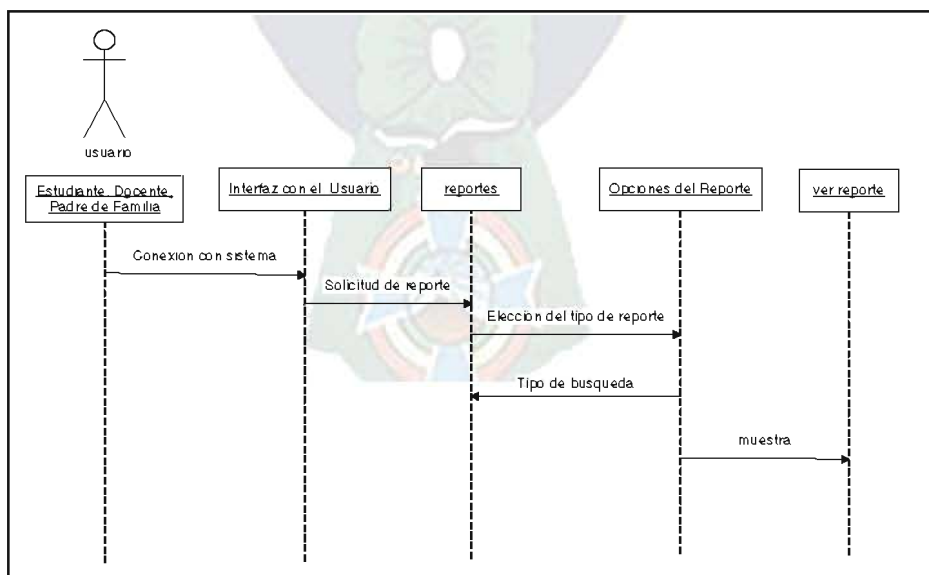
Figura 3.8 Diagrama de Colaboración Correspondiente a la Inscripción de Alumnos



Fuente: Elaboración Propia

Luego pasada la etapa de Inscripción el sistema podrá dar los reportes respectivos así: listado de docentes, listado de alumnos, reportes de los datos tanto por alumno como por docente y otros (Figura 3.9)

Figura 3.9 Secuencia Correspondiente a la Solicitud de Reportes



Fuente: Elaboración Propia

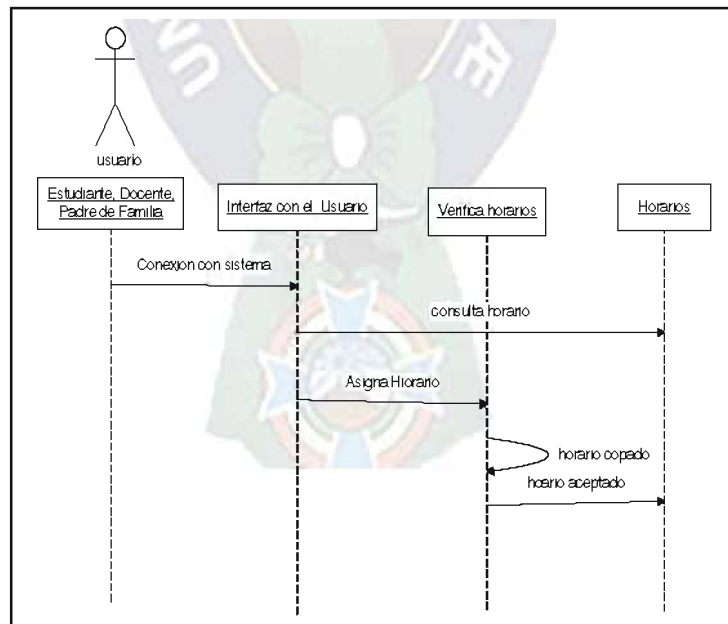
Figura 3.10 Diagrama de Colaboración Correspondiente a la Solicitud de Reportes



Fuente: Elaboración Propia

También podremos observar los Horarios correspondiente a los estudiantes y docentes, el sistema podrá dar los respectivos cuadros de horarios (Figura 3.11 )

Figura 3.11 Secuencia Correspondiente a los Horarios



Fuente: Elaboración Propia

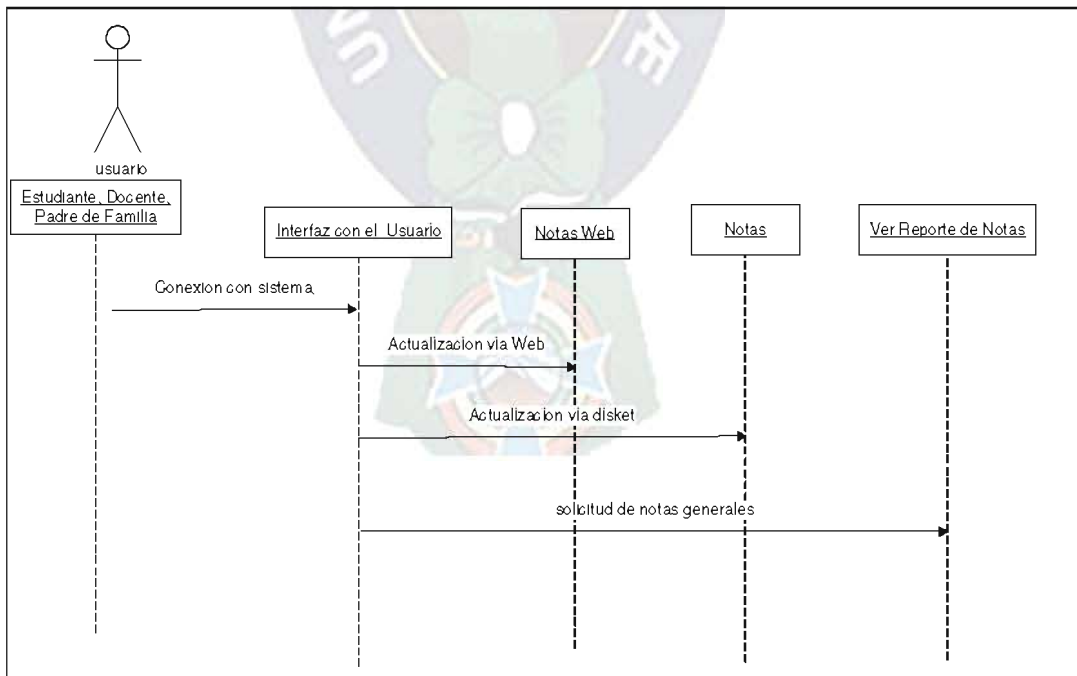
Figura 3.12 Diagrama de Colaboración Correspondiente a los Horarios



Fuente: Elaboración Propia

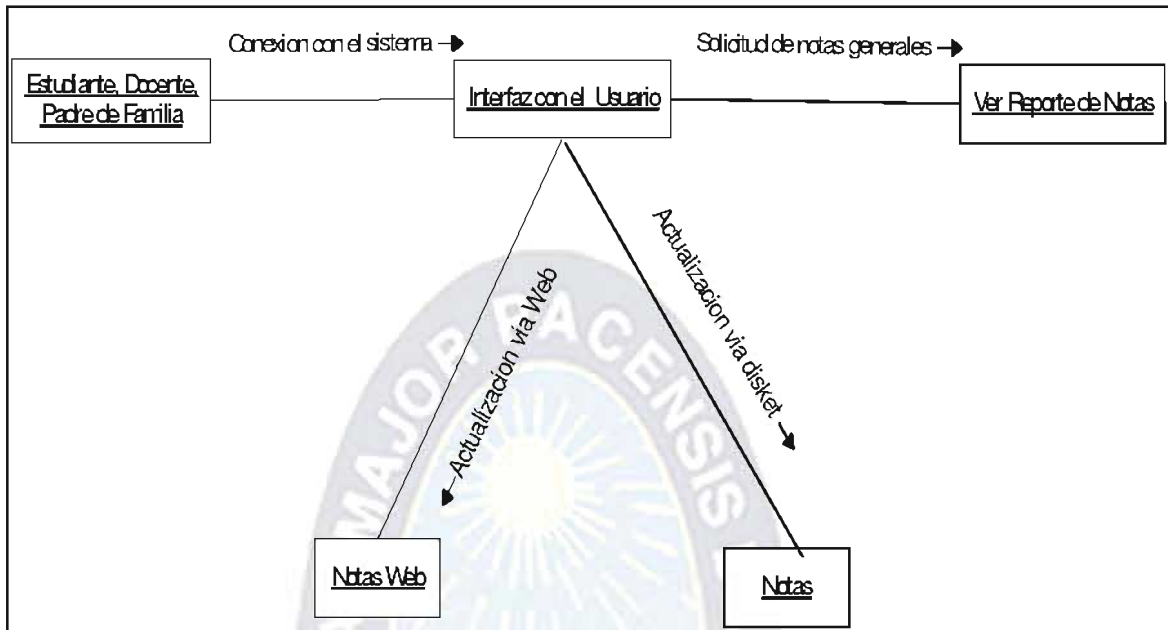
También podremos observar las Notas de los Estudiante(s) Trimestralmente, este evento le corresponde al rol del estudiante y Padre de Familia o Apoderado, también los administrativos (Figura 3.13).

Figura 3.13 Secuencia Correspondiente a las Notas



Fuente: Elaboración Propia

Figura 3.14 Diagrama de Colaboración Correspondiente a las Notas



Fuente: Elaboración Propia

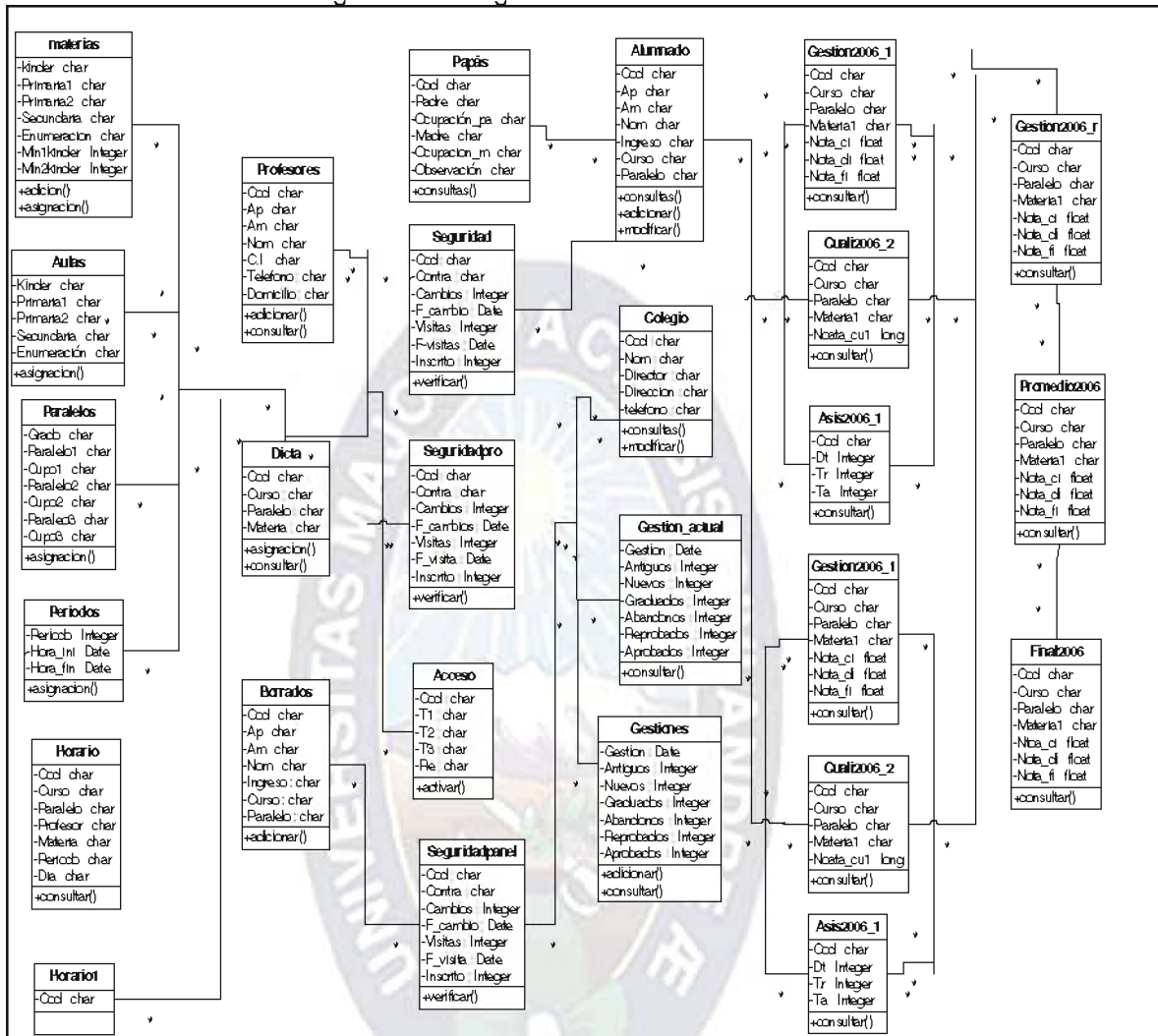
## 3.2 FASE DE COLABORACION

Esta es la segunda fase del ciclo de vida ASD, se desarrollan las características del sistema, pero luego de haber realizado varias iteraciones el sistema finalmente presenta el siguiente esquema en cuanto a su estructura de funcionamiento, el cual también se halla representado por herramientas del lenguaje de modelado unificado UML apoyándose en el diagrama de clases y también se utilizó los diagramas de E-R para modelar la estructura de la base de datos del sistema

### 3.2.1 DISEÑO DEL DIAGRAMA DE CLASES

El sistema se ha logrado modelar de acuerdo al siguiente diagrama de clases que se muestra en la figura 3.15

Figura 3.15 Diagrama de Clases del Sistema



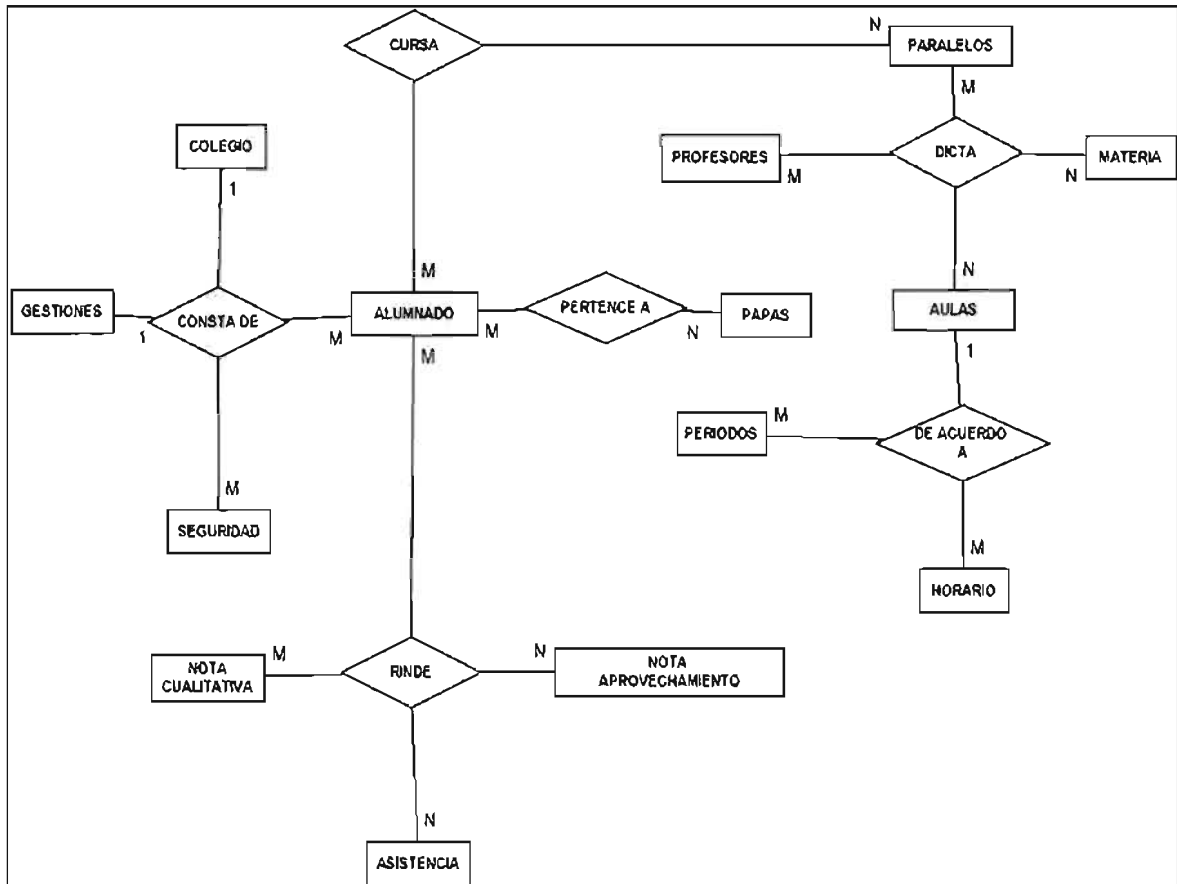
Fuente: Elaboración Propia

### 3.2.2 DISEÑO DEL DIAGRAMA DE ENTIDAD – RELACIÓN

Del diagrama de clases de la Figura 3.15 se ha logrado identificar las entidades, atributos y relaciones, esto considerando sólo las clases más importantes.

De donde se ha logrado elaborar el diagrama Entidad- Relación que nos proporciona una representación grafica mas clara de nuestra base de datos la cual se puede observar en la Figura 3.16

Figura 3.16 Diseño diagrama Entidad- Relación



Fuente: Elaboración Propia

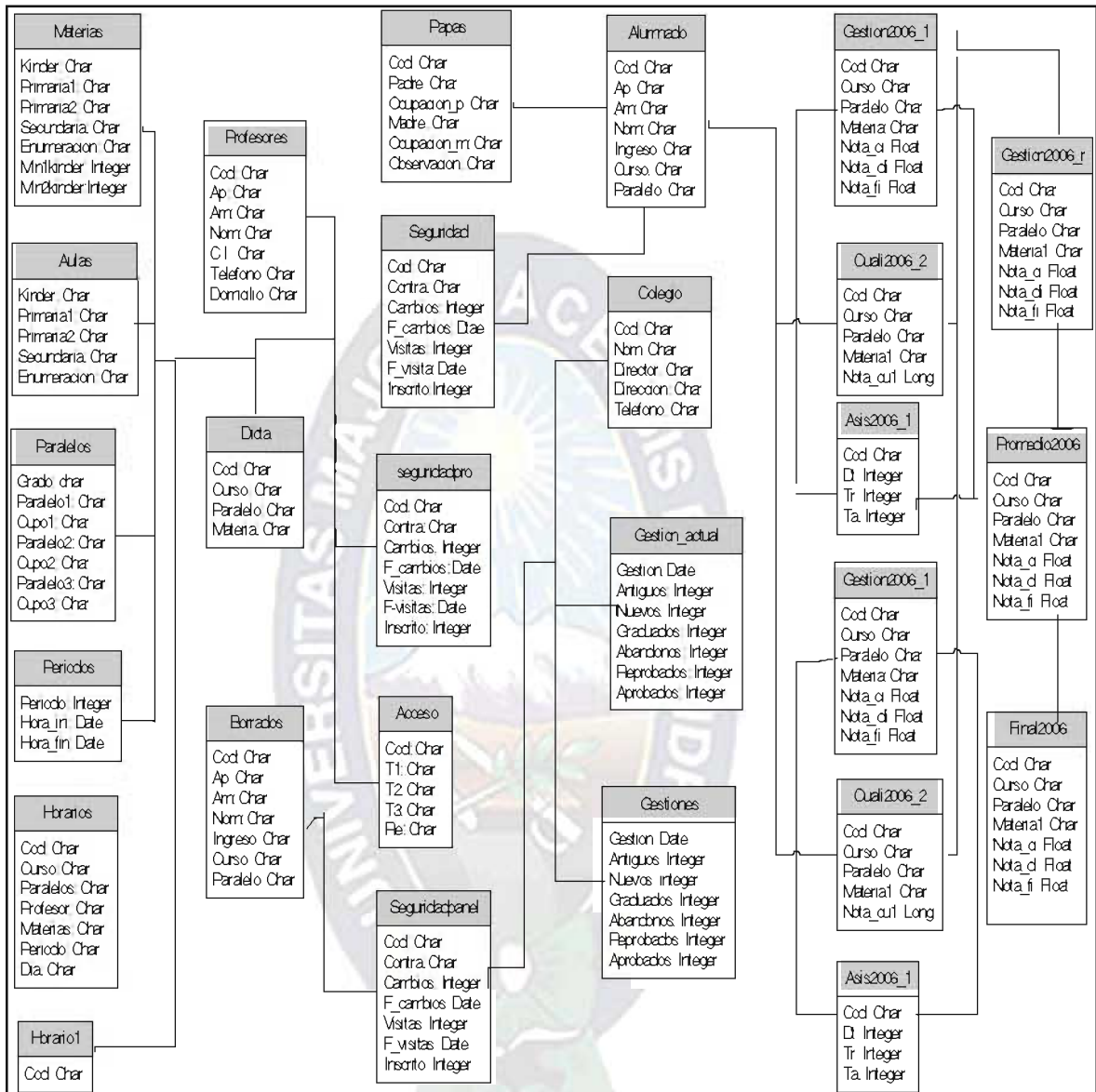
### 3.2.3 DISEÑO DE LA BASE DE DATOS

Se logró obtener una estructura de la base de datos, en la cual se busca que puedan efectuarse combinaciones de consultas dependiendo de la necesidad del usuario, además tomando en cuenta que los datos deben ser precisos y consistentes, obteniendo un almacenamiento, una recuperación y actualización de datos eficiente.

A continuación en la Figura 3.17 se ve el esquema de la base de datos con la que el sistema funciona de manera adecuada en las Unidades Educativas.



Figura 3.17 Esquema de la base de datos



Fuente: Elaboración propia

### 3.2.4 DICCIONARIO DE DATOS

Esta es una herramienta necesaria con la que organizaremos todos los datos pertinentes del sistema, lo cual nos permitirá un entendimiento común de todas las entradas salidas que forman parte de las tablas elaboradas en nuestra base de datos a continuación podemos observar alguna de estas.

<b>Alumnado = cod +ap +am +nom +ingreso + curso + paralelo</b>		
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
Cod	<i>varchar</i>	** Clave primaria que identifica al estudiante **
Ap	<i>char</i>	** Apellido Paterno de estudiante **
Am	<i>char</i>	** Apellido Materno del estudiante **
Nom	<i>char</i>	** Nombre completo del estudiante **
Ingreso	<i>date</i>	** Fecha de ingreso del estudiante **
Curso	<i>varchar</i>	** Curso en el cual el estudiante se inscribió **
Paralelo	<i>varchar</i>	**Paralelo que se asigna al estudiante **

<b>Papas = cod + padre + ocupación_p +madre +ocupacion_m +observación</b>		
<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCION</b>
Cod	<i>varchar</i>	**Clave primaria que identifica al estudiante**
Padre	<i>char</i>	**Nombre y apellido del padre del estudiante**
Ocupación_p	<i>char</i>	** Ocupación con la que cuenta el padre de familia**
Madre	<i>char</i>	**Nombre y apellido de la madre del estudiante**
Ocupación_m	<i>char</i>	** Ocupación con la que cuenta la madre de familia**
Observación	<i>char</i>	** Se identifica si son los padres o apoderados **

### 3.3 FASE DE APRENDIZAJE

Esta es la tercera fase del modelo ASD esta consiste en la revisión de calidad que se realiza al final de cada ciclo. En la misma se analizan las siguientes categorías de cosas para aprender

- Calidad del resultado desde la perspectiva del cliente
- Calidad del resultado desde la perspectiva técnica

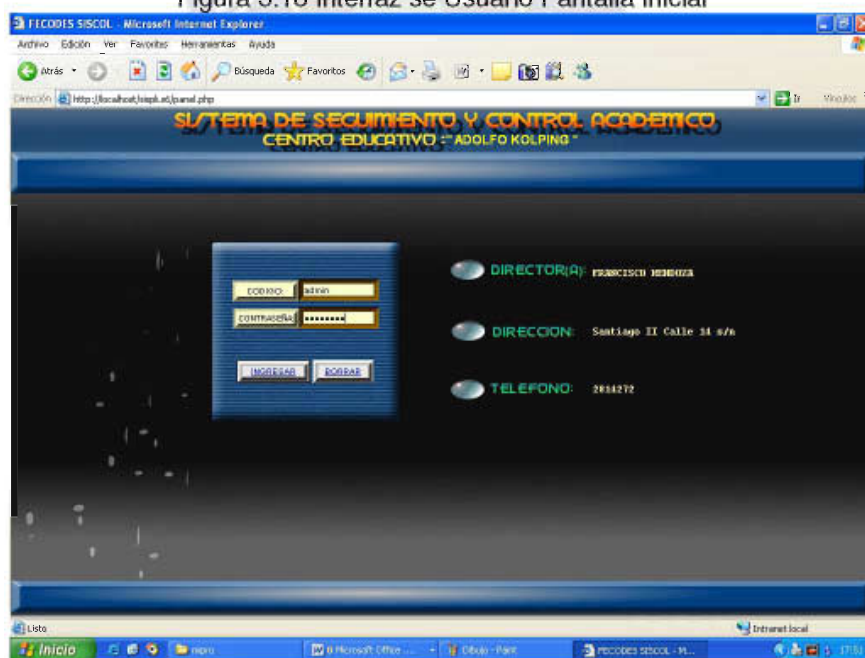
El calculo de calidad desde la perspectiva técnica se encuentra elaborado en el capitulo siguiente en el cual se ve que el proyecto se encuentra en el margen de calidad optimo. Una vez que estos puntos estén sin problemas puede realizarse la entrega al cliente. En el presente proyecto luego de optimizar los puntos mencionados, esto se logro después de varias iteraciones a través del ciclo de vida de ASD, se ha podido aprender de errores y finalmente se logro obtener el producto requerido. Los diseños finales con los que proyecto llego a su conclusión los podemos observar a continuación.

### 3.3.1 DISEÑO DE INTERFAZ DE USUARIO

A continuación detallamos algunas de las pantallas del sistema las cuales se realizaron en base a las normas de W3C

- **PANTALLA INICIAL.-** Esta es la encargada para recibir los datos de ingreso del usuario al sistema y si es correcta logra acceder, al ingresar presenta tres tipos de usuario administradores, docentes y alumnos

Figura 3.18 interfaz se Usuario Pantalla inicial



Fuente: Elaboración Propia

- **PANEL DE CONTROL.-** El Panel de control para los diferentes usuarios es el siguiente:

**- Panel de control de administradores**

En acá los encargados de la administración del sistema logran realizar todas las funciones que aviamos descrito en nuestros casos de uso, siendo esta la pantalla principal para el uso del sistema

Figura 3.19 Interfaz de Usuario, Panel de Control Administrador

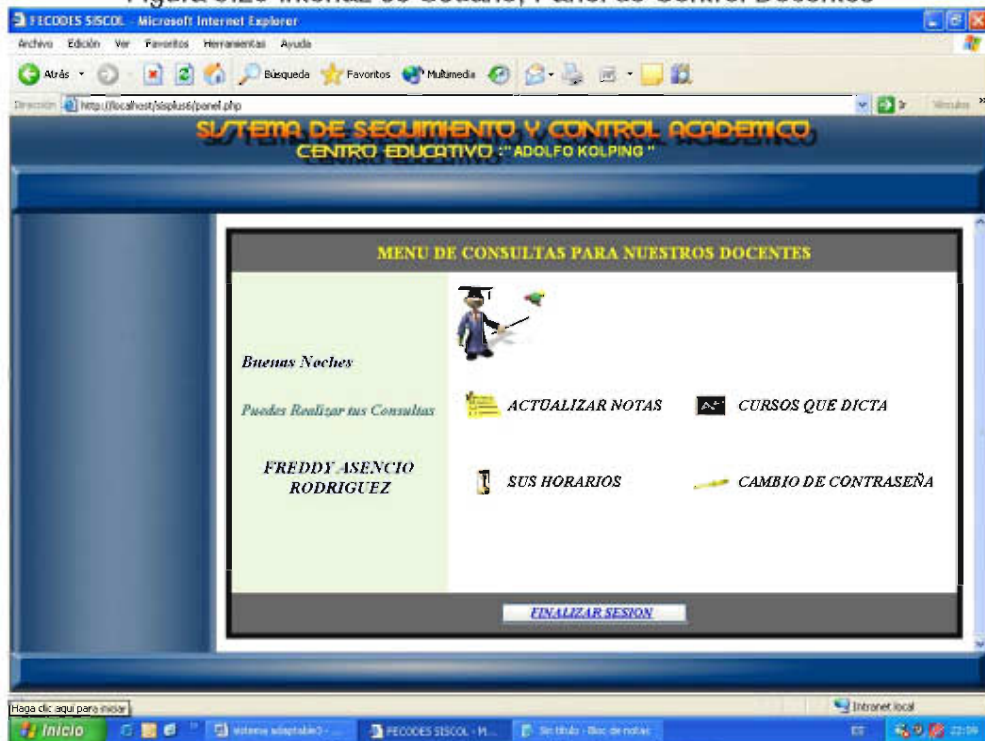


Fuente: Elaboración Propia

**- Panel de control para docentes y alumnos**

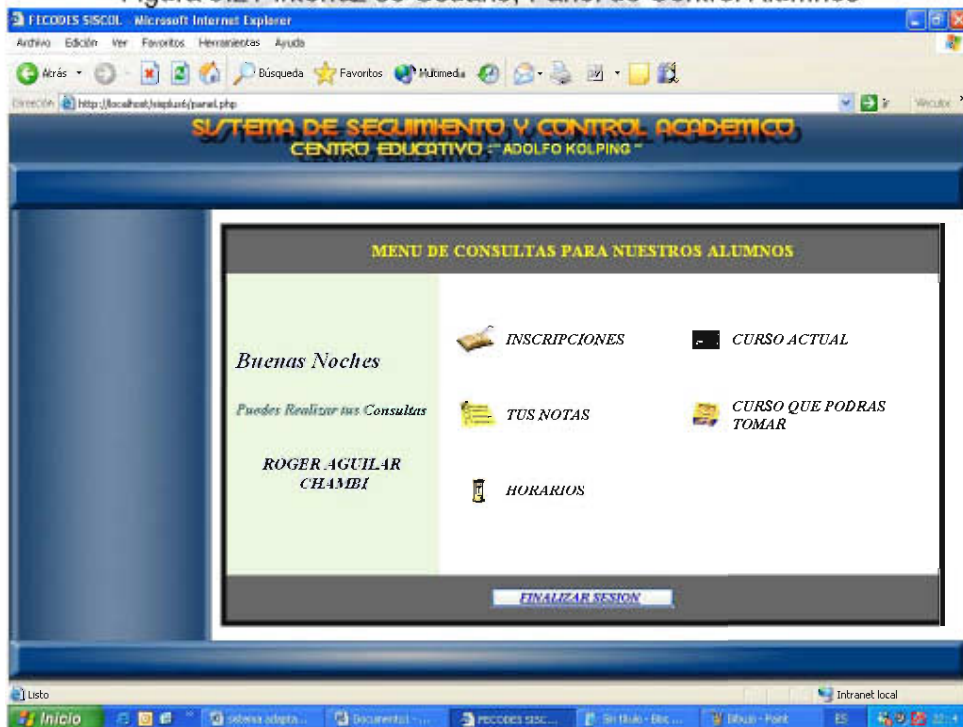
En acá los docentes y los alumnos, como padres de familia pueden acceder al sistema para las diversas consultas que tengan

Figura 3.20 Interfaz se Usuario, Panel de Control Docentes



Fuente: Elaboración Propia

Figura 3.21 Interfaz se Usuario, Panel de Control Alumnos



Fuente: Elaboración Propia

- **CONSULTAS.-** Se puede consultar los alumnos en particular por curso, por código, fecha de nacimiento, por apellidos. El sistema nos mostrara un reporte similar a la Figura 3.20

Figura 3.22 Nomina de Alumnos

CURSO	SP	A	N	PATERNO	MATERNO	NOMBRE	CODIGO	DIRECCION	TELEFONO
1			1	ADUVIR	ZANSPANA	ROGANGELA	ADR100488	ZONA ZARAGOZA ESQUELANA N. 1055	70337859
2			2	ALCONZ	DESPEDES	RONALDO DANIEL	AC050408	ZONA ZARAGOZA 5700 DE MADRUGA N. 2036	203615
3			3	ALCJO	MUÑECA	JOSE CARLOS	AHJ03	ZONA N.	
4			4	APAZA	DRUZ	DIEGO RUBEN	ACD196997	ZONA ZARAGOZA KANTUTANI N. 2055	
5			5	CALLIPA	ELIANA	MARIA	CE0505997	ZONA ZARAGOZA KALAPACHTA N. 2194	7033752
6			6	DESPEDES	LASTARDO	SHIRLEY JEMINA	ELS	ZONA N.	
7			7	ENKQUE	ALCONZ	DIADYS	EAD	ZONA N.	
8			8	ENKQUE	QUIRPE	JEROME LUIS	EQ0401997	ZONA 28 DE DIC. AV. CALLAPA N. 1005	
9			9	DONDON		APAZA, JESUS ADALEI	C_AY07990	ZONA ZARAGOZA CYNICA N. 1096	
10			10	COSME	MARCA	WILMER	CMW	ZONA N.	
11			11	CRUZ	ROJAS	JROSELINE	CRJ102997	ZONA ZARAGOZA CALLE 7 N. 2034	
12			12	FERNANDEZ	FLORES	PAMELA ESTEFANY	FF051197	ZONA ZARAGOZA CALLE 16 N. 2095	
13			13	FLORES	LINA	ESTEFANI	FL021197	ZONA 8 Y "E" INCLUYO N. 2025	
14			14	GUTIERREZ	LEPEZ	MOSHE RUTH	GM070598	ZONA 19 DE BELL LATA N. 1018	
15			15	GUTIERREZ	DISCO	NESTOR AGRAMAN	GD140798	ZONA 11 Y "A" SANTA ROSA N. 1126	
16			16	HUANGA	ROJAS	VERALDEN YESSICA	MV011097	ZONA ZARAGOZA TOLAMARCA N. 1184	
17			17	LARICO	CORDON	SILVIA	ELS	ZONA N.	
18			18	LARIC	KUNO	FRANKLIN	FLF	ZONA N.	
19			19	MAMANI	CORDERO	RUBEN	MC0210097	ZONA 8 Y "E" JANKO DIRCA N. 1195	
20			20	MAMANI	CRUZ	MARINA	MC051298	ZONA 8 Y "E" PALCOMA N. 2015	77653308
21			21	MAMANI	GUTIERREZ	WALLY DAYAN	BM0021197	ZONA 8 Y "A" CALLE 7 N. 2084	
22			22	MAMANI	HUANGA	EDWIN	ME050497	ZONA ZARAGOZA AV. RIO BLANCO N. 2034	7138381
23			23	MAYTA	DARICA	JUAN FREDDY	MDJ	ZONA N.	
24			24	NETA	LAURA	MARIANDELA	RLM	ZONA V	

Fuente: Elaboración Propia

- **HORARIOS.-** A través de este modulo se puede consultar horarios y elaborar horarios.

Figura 3.23 Elaboración de Horarios

Horas	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
P-1 08:00-09:45	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE		Matemática Pro: JAVIER CHUMBI RONALDO	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE
P-2 09:45-11:00	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE	Ciencias Sociales Pro: JAVIER CHUMBI RONALDO	Ciencias Sociales Pro: JAVIER CHUMBI RONALDO	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE
P-3 09:30-10:15	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: CHINO MAMANI DANA RUBEN	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE
P-4 10:15-11:00	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: CHINO MAMANI DANA RUBEN	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE
P-5 11:00-11:45	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE
P-6 11:45-12:30	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Lenguaje Pro: PLATERES RUILOVA ANDREA YANETTE	Matemática Pro: PLATERES RUILOVA ANDREA YANETTE	Ciencias Sociales Pro: PLATERES RUILOVA ANDREA YANETTE
P-7 12:30-13:15					
P-8 13:15-14:00					
P-9 14:00-14:45					
P-10 14:45-15:30					

Fuente: Elaboración Propia

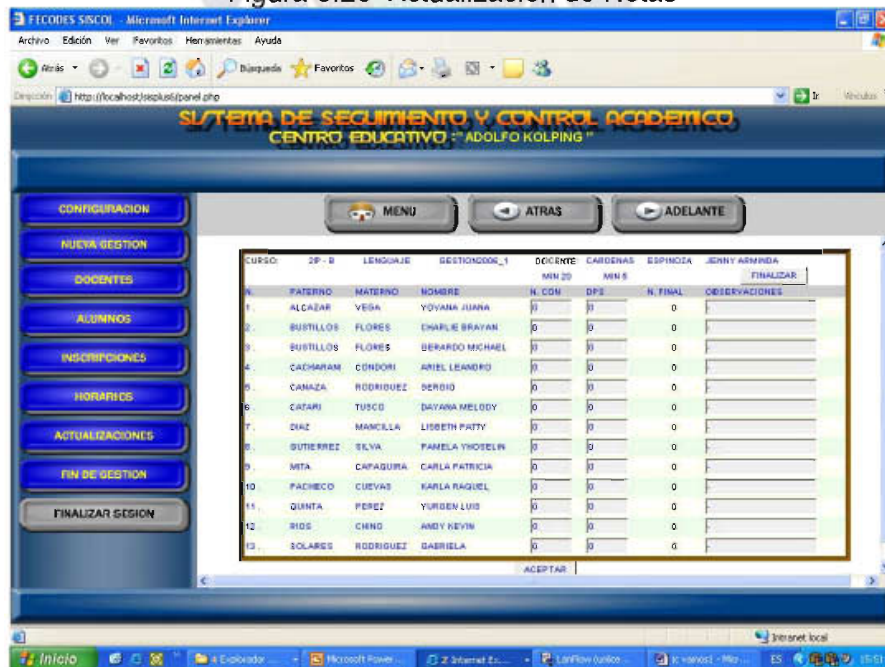
- **ACTUALIZACIONES.-** Se logra actualizar el sistema con respecto a las notas de alumnos a través de la Web o vía disket, para ello previamente se exportan los datos de los alumnos que correspondan a un docente

Figura 3.24 Exportación de datos de los Alumnos



Fuente: Elaboración Propia

Figura 3.25 Actualización de Notas

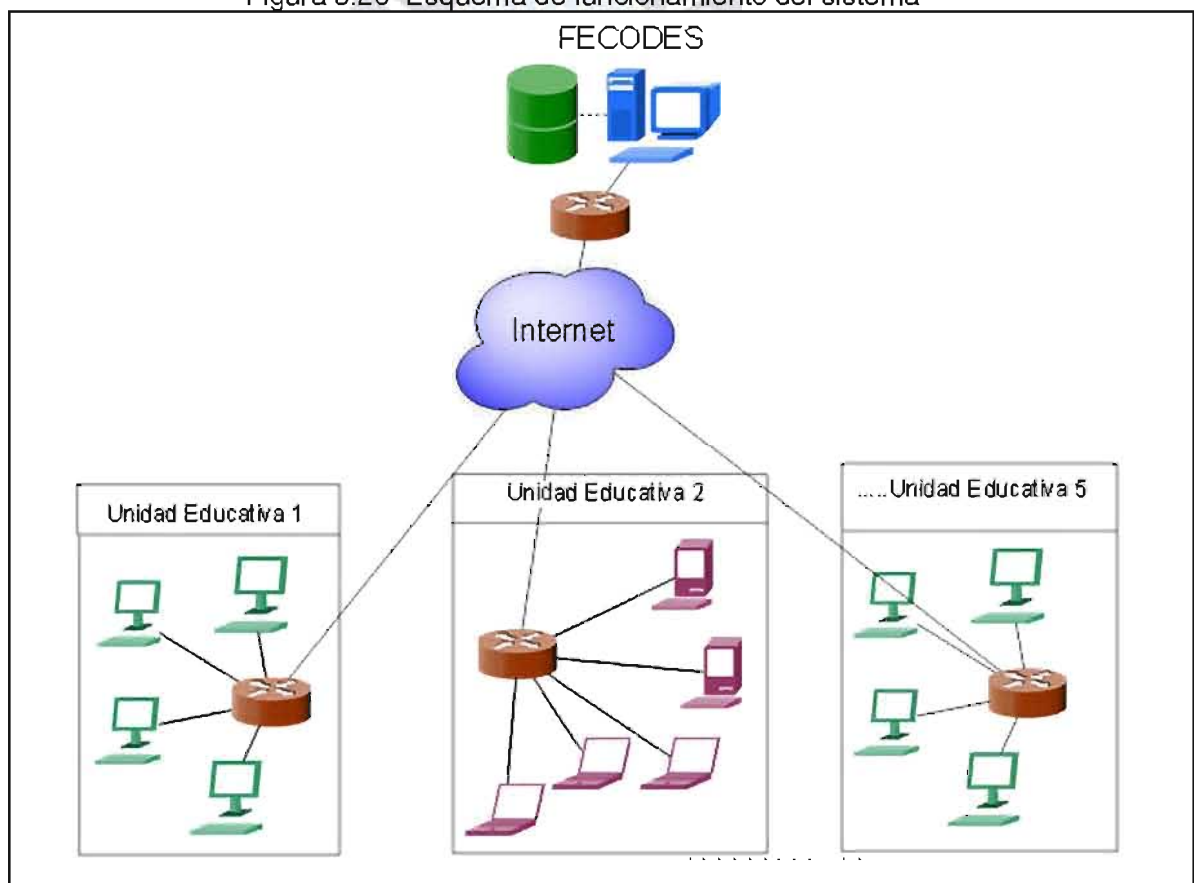


Fuente: Elaboración Propia

### 3.3.2 IMPLEMENTACIÓN DEL SISTEMA

Cada Unidad Educativa contara con el sistema estas trabajaran inicialmente de manera local, y posteriormente a través de Internet de tal modo que la Fundación Ecología y Desarrollo podrá centralizar la información de estas, el esquema general de funcionamiento se muestra en la Figura 3.26

Figura 3.26 Esquema de funcionamiento del sistema



Fuente: Elaboración Propia



# CAPITULO 4

## CALIDAD DEL SOFTWARE

El objetivo de un producto software es que posea la calidad necesaria y suficiente para que satisfaga las necesidades de usuario explícitas e implícitas. En el presente proyecto para el aseguramiento y control de la calidad se presenta un modelo de calidad orientado a sitios Web, el modelo Web-site QEM propuesta por el Dr. Luis Olsina.

### 4.1 EVALUACION DE LAS PREFERENCIAS ELEMENTALES

Efectuando la evaluación elemental de usabilidad tomando en cuenta el árbol de calidad del Dr. Olsina (Figura 2.15) donde para cada punto se desarrollo el siguiente esquema que nos permite el calculo del porcentaje de satisfacción (Ei).

Título: Mapa del Sitio; Código: 1.1.1.1; Tipo: Atributo

Característica de más Alto Nivel: Usabilidad

Definición: Un mapa del sitio es una representación con componentes gráficos, que muestra la estructura o arquitectura global (a menudo jerárquica) del sitio Web como un todo.

Tipo de Criterio Elemental: es un criterio binario, discreto y absoluto: sólo se pregunta sí está disponible (1) o si no está disponible (0).

Escala de Preferencia:

$E_i$  = La preferencia elemental computada es 100%. Peso: con un peso de 0.2 (ver Figura 2.17).

Título: Mapa de imagen del campus; Código: 1.1.4; Tipo: Atributo

Característica de más Alto Nivel: Suavidad

Definición: Un mapa es una imagen (tal vez navegable) con funciones de ampliación (zooming); y/o visita virtual.

Tipo de Criterio Elemental es un criterio multinivel, discreto y absoluto:

$X=0$  no disponible, entonces  $IE_i=0\%$

$X=1$  disponible, permitiendo sólo referencias no enlazadas y/o zooming, y  $IE_i=60\%$

$X=2$  disponible, permitiendo navegación enlazada directa o indirectamente y/o zooming, y/o visita virtual, y  $IE_i=100\%$ .

Escala de Preferencia:

$E_i$  = La preferencia elemental computada es 60%.

Peso: con un peso de 0.3 (ver Figura 2.17)

En las siguientes tablas se denota los resultados obtenidos para cada atributo del árbol de calidad (Ver Figura 2.15)

Tabla 4.1 Resultados de evaluación elemental de Usabilidad

1.14	Mapa de imagen del campus	Un mapa de imagen es una imagen (tal vez navegable), con funciones de ampliación (zooming); y/o visita virtual. Sea $X=0$ si esta no es disponible, $X=1$ disponible, permitiendo navegación no enlazada y $X=2$ disponible, permitiendo navegación enlazada y visita virtual.	Es un criterio multinivel, discreto y absoluto.	60
		pregunta si esta disponible (1) o si no esta disponible (0).		

1.2.2.1	Fecha última de actualización Global de todo el sitio Actualización global.	Permite verificar la última fecha de Actualización Solo se pregunta si esta disponible (1) o si no esta disponible (0).	Es un criterio, binario, discreto, absoluto.	100
1.2.2.2	Última Actualización por página	Es la fecha que se realizó la última actualización en las sesiones o páginas P. Ej. Noticias, Agenda, etc, (cada ítem tiene la fecha de última actualización)	Es un criterio binario discreto y absoluto	100
1.2.3.1	Directorio E-mail	Es el lugar en el sitio donde se agrupan las direcciones electrónicas para enviar E-mail (como mecanismo de retroalimentación). Solo se pregunta si esta disponible (1) o si esta no esta disponible (0).	Es un criterio, binario discreto, absoluto.	100
1.2.3.2	Directorio Teléfonos- Fax	Es el lugar en el sitio donde se agrupan las direcciones números para contactarse vía telefónica o por medio de Fax. Solo se pregunta si esta disponible (1) o si esta no esta disponible (0).	Es un criterio, binario, discreto, absoluto.	100
1.2.4	Facilidad FAQ	Esto mide si el Sitio Web es capaz e responder a las preguntas más frecuentes del usuario. Solo se pregunta si es capaz (1) o no es capaz (0).	Es un criterio, binario, discreto, absoluto.	0
1.2.5.3	Comentarios y Sugerencias	Representa un formulario el cual el usuario puede llenar y puede hacer alguna sugerencia o comentario al web master.	Es un criterio, binario discreto, absoluto.	100

1.3.2.1	Permanencia de controles directos	Este atributo representa la permanencia directa de los controles del menú principal del sitio que permiten navegación.	Es un criterio binario, discreto, absoluto	100
1.3.2.2	Permanencia de Controles indirectos	Es un control o referencia indirecta a la página principal. Solo se pregunta si esta disponible (1) o si esta no esta disponible (0)	Es un criterio, binario, discreto, absoluto	100
1.3.2.3	Estabilidad	Se refiere a la ubicación de los controles principales directos o Indirectos en los nodos del sitio. Un control es estable si se encuentra siempre en la misma ubicación al navegar por los nodos. Solo se pregunta si esta siempre en la misma Ubicación (1 ) o si no lo esta (0).	Es un criterio, binario, discreto, absoluto.	100
1.3.3.1	Uniformidad sn el color de os Enlaces.	Se refiere si todos los enlaces que posee el sitio Web son de un mismo color. Se empleara la experiencia del evaluador para ver si todos los enlaces son de un mismo color (1) o si no lo son (0).	Es un criterio de referencia de calidad directa.	100
1.3.3.2	Uniformidad en el Estilo Global.	Si todo sitio Web posee una uniformidad de color, estilo y fuente. Sea X=0 si se interpreta si no posee una uniformidad de estilo, X=1 si posee parcialmente una uniformidad de estilo y X=2 si posee una uniformidad de estilo total.	Es un criterio multi-nivel, discreto, absoluto	100
1.4.1	Soporte a Lenguaje Extranjero	Este atributo modela la disponibilidad parcial o total de lenguajes extranjeros soportados por el sitio Web. No se computa el lenguaje nativo como lenguaje extranjero. Sea X=0 si no soporta lenguajes extranjeros. X=1	Es un criterio multi-nivel, discreto, absoluto	60

En la Tabla 4.2 se muestra valores de los criterios elementales para cada uno de los atributos de la característica de *Funcionalidad*. Estos valores se obtuvieron a partir de los criterios de Preferencia Elemental.

Tabla 4.2 Resultado de las evaluaciones elementales de Funcionalidad

Código	Atributo	Definición	Criterio	IEÍ (%)
2.1.1.1	Búsqueda restringida	Búsqueda de un elemento particular de la aplicación.	Es un criterio elemental binario	0
2.1.1.2	Búsqueda global	Búsqueda en la aplicación	Es un criterio multi-nivel, absoluto, discreto	0
2.2.2.1.2	Promedio de enlaces por página	Da una idea sobre la cantidad de puntos de partida en una página, que indica como una página esta interconectada con hacia nodos destino.	Es un criterio multi-nivel, discreto, absoluto	60
2.2.2.1.1	Permanencia de los controles contextuales	Es la permanencia de los controles de navegación de un subsitio, de modo que los visitantes tengan acceso permanente a los principales nodos del sitio, y realizando el menor esfuerzo cognitivo y de	Es un criterio multi-nivel, discreto, absoluto	60
2.2.2.1.2	Estabilidad	Se refiere a la ubicación de los controles en los nodos del subitio. Un control es estable si se encuentra siempre en la misma ubicación durante la navegación de los subsitios.	Es un criterio multi-nivel, discreto, absoluto	60

2.2.2.2.1	Desplazamiento Vertical	Nivel de desplazamiento vertical que el visitante debe realizar para ajustar la interfase (considerando 640 x 480 como la mínima resolución)	Es un criterio variable normalizada	50
2.2.2.2.2	Desplazamiento Horizontal	Nivel de desplazamiento horizontal que el visitante debe realizar para ajustar la interfase (considerando 640 x 480 como la mínima resolución)	Es un criterio variable normalizada.	100
2.2.3.1	Enlace con titulo	Este atributo trata de predecir los temas o contenidos que están asociados al enlace.	Es un criterio multi - nivel, discreto, absoluto	60
2.2.3.2	Calidad de la Frase del Enlace	Cuando los enlaces no son suficientemente descriptivos, el visitante podría no tener una buena pista respecto de lo que dichos enlaces significan, principalmente cuando no hay un buen contexto. Por lo tanto, se tiene una buena	Es un criterio multi - nivel, discreto, absoluto	60
2.3.1.1	Información de la unidad académica	Este atributo da una idea si están presentes los datos relevantes de la unidad académica.	Es un criterio elemental binario,	100
2.3.1.2.1	Información de los requerimientos de ingreso/admisión	Establece que la información referente a inscripciones de la unidad académica esta presente. Solo se pregunta si esta cuenta con información de los requerimientos de ingreso (1) o si no cuenta con este atributo (0).	Es un criterio multi-nivel, discreto	100
2.3.1.3	Información de Carreras	Establece que la información referente a las Carreras de la unidad académica. Solo se pregunta si esta cuenta con información de las carreras (1) o si no cuenta con este atributo (0).	Es un criterio elemental binario, discreto, absoluto	100

2.3.1.4.1	Información de convenios	Es una representación de los convenios con que cuenta esta unidad académica. Solo se pregunta si esta cuenta con información (1) o si no cuenta con este atributo (0).	Es un criterio elemental binario, discreto, absoluto	100
2.3.1.4.2	Información de Noticias	Es la representación de todas las noticias que se generan en la Facultad y sus carreras	Es un criterio elemental binario, discreto, absoluto	100
2.3.1.4.3	Información de Becas	Es una representación de las becas que existe en la aplicación. Solo se pregunta si esta cuenta con información (1) o si no cuenta con este atributo (0).	Es un criterio elemental binario, discreto, absoluto	100
2.3.1.4.5	Información de publicaciones	Es una representación de las publicaciones que son desarrolladas en la Facultad. Solo se pregunta si esta cuenta con información de publicaciones (1) o si no cuenta con este atributo (0).	Es un criterio elemental binario, discreto, absoluto	100
2.3.1.5.1	Información de bibliotecas	Enlace existente a la biblioteca de la unidad académica.	Es un criterio elemental binario, discreto, absoluto	100
2.3.1.5.2	Información de Laboratorios	Información referente a los laboratorios con que cuenta la Facultad.	Es un criterio elemental binario, discreto, absoluto	100
2.3.2.1	Servicio de grupo de noticias.	Servicio de noticias en la aplicación. Solo se pregunta si esta cuenta con información de noticias (1) o si no cuenta con este atributo (0).	Es un criterio elemental binario, discreto, absoluto	100

2.3.2.2	Servicio FTP	Servicio de transferencia de archivos	Es un criterio elemental binario, discreto, absoluto	100
---------	--------------	---------------------------------------	------------------------------------------------------	-----

En la Tabla 4.3 se muestra valores de los criterios elementales para cada uno de los atributos de la característica de confiabilidad. Estos valores se obtuvieron a partir de los criterios de Preferencia Elemental.

Tabla 4.3 Resultado de las evaluaciones elementales de Confiabilidad

Código	Atributo	Definición	Criterio	IEÍ (%)
3.1.1.1	Enlaces Rotos	Se mide la cantidad de enlaces que están rotos. Se utiliza la siguiente formula: $X = 100 - (ER * 100 / ET) * 10$ Donde: ER= Cantidad total de enlaces rotos. ET= Cantidad total de enlaces. $X = 100 - 0 = 100$	Variable normalizada, continua, absoluta	100
3.1.1.2	Enlaces Inválidos	Se mide los enlaces que no son validos. Se utiliza la siguiente formula: $X = 100 - (EI * 100 / ET) * 10$ Donde: EI= Cantidad total de enlaces inválidos. ET= Cantidad total de enlaces. $X = 100 - 0 = 100$	Variable normalizada, continua, absoluta	100
3.1.1.1	Enlaces Rotos	Se mide la cantidad de enlaces que están rotos. Se utiliza la siguiente formula: $X = 100 - (ER * 100 / ET) * 10$ ER= Cantidad total de enlaces rotos. ET= Cantidad total de enlaces. $X = 100 - 0 = 100$	Variable normalizada, continua, absoluta	100
3.1.1.2	Enlaces Inválidos	Se mide los enlaces que no son validos. Se utiliza la siguiente formula: $X = 100 - (EI * 100 / ET) * 10$	Variable normalizada, continua,	100



		$100 - (ER * 100 / EI) * 10$ Donde: EI= Cantidad total de enlaces inválidos. ET= Cantidad total de enlaces. $X = 100 - 0 = 100$	absoluta	
3.1.1.3	Enlaces no implementados	Se mide los enlaces en el sitio Web, Sea $X=0$ si no existen enlaces implementados, $X=1$ si existen algunos enlaces implementados y $X=2$ si todos los enlaces están Implementados.	Multi-nivel, discreto, absoluto	60
3.1.2.1	Deficiencias o cualidades ausentes debido a diferentes Navegadores.	Representa la existencia de alguna deficiencia en la presentación del portal Web, debido a diferentes navegadores. Sea $X=0$ si no existen deficiencias en la presentación, $X=1$ si existen deficiencias de forma parcial $X=2$ la presencia total del atributo.	Es un criterio de multi-nivel, discreto, absoluto	100
3.1.2.2	Deficiencias o resultados inesperados independientes del browser.	Se mide si existe una mala presentación del sitio Web sin que afecte el navegador que utilizemos.	Es un criterio de referencia de calidad directa.	100
3.1.2.2	Nodos Destinos en construcción	Este atributo indica si existe paginas en construcción, la formula para computar la variable es : $X = 100 - (NC * 100 / NT) * 10$ En donde si $NC =$ numero de nodos en construcción encontrados. $NT =$ Numero total de nodos del sitio Si $X < 0$ entonces $X = 0$ $X = 100 - 0$ $X = 100$	Criterio de variable normalizada	100

3.1.2.2	Nodos web muertos	Este atributo indica si existe nodos del portal muertos o sea sin funcionamiento y la formula para computar la variable es : $X=100-(NM*100/NT)*10$ En donde si NM=numero de nodos muertos encontrados. NT=Numero total de nodos del sitio Si $X<0$ entonces $X=0$ $X=100-0$ $X=100$	Criterio de variable normalizada	100
---------	-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------	-----

En la Tabla 4.4 se muestra valores de los criterios elementales para cada uno de los atributos de la característica de *Eficiencia*. Estos valores se obtuvieron a partir de los criterios de Preferencia Elemental.

Tabla 4.4 Resultado de las evaluaciones elementales de Eficiencia

Código	Atributo	Definición	Criterio	IEÍ (%)
4.1.1	Paginas de acceso rápido	Se mide el tamaño de todas las páginas de la aplicación considerando todos sus componentes. Tamaño aceptable 35,2 Kb, requiere 20 seg. Para bajar a una tasa de 1 4400 bps. $X=0$ si no cuenta con paginas de acceso rápido de forma total, entonces $IEi=0\%$ $X=1$ si cuenta con paginas de acceso rápido de forma parcial, entonces $IEi=60\%$ $X=2$ si no cuenta con paginas de acceso rápido de forma total, entonces $IEi=100\%$	Multi-nivel, discreto, absoluto	60

4.2.1.1	Soporte a versión solo texto	Representa la accesibilidad a la información que esta en las Páginas. Es relevante que la página sea editada en una versión de solo texto. $X=0$ si no esta preparado para versión solo texto, entonces $IE_i=0\%$ $X=1$ si esta preparado parcialmente, entonces $IE_i=60\%$ $X=2$ si esta preparado para versión solo texto de manera total, entonces $IE_i=100\%$	Multi - nive discreto absoluto.	60
4.2.1.2.1	Imagen con Título	Mide el porcentaje de la presencia la etiqueta <ALT> de manera de incluir texto alternativo a la imagen cuando se desactiva la propiedad imágenes del navegador. Sea $X=100-X_j$ $X_j=100-(TA*100/TT)$ Si $TA$ =Cantidad de títulos ausentes $TT$ =Número total de títulos. $X_j$ ^Representa el % de imágenes con título. Entonces si $X=100-(0*100/20)*10$ $X=100$	Variable normalizada, continua, absoluta	100
4.2.1.2.2	Legibilidad global	Representa la presencia de calidad en consideración del nivel de legibilidad global del sitio cuando desactiva la propiedad ver imágenes del navegador. $X=0$ si no existe legibilidad, entonces $IE_i=0\%$ $X=1$ si esta preparado parcialmente, entonces $IE_i=60\%$ $X=2$ si existe legibilidad global, entonces $IE_i=100\%$	Multi-nivel discreto, absoluto	60
4.2.2.1	Número de vistas considerando marcos	Los trames o marcos organizan a una ventana en diferentes áreas o subvistas, tanto de control como de contenido. Cuanto mayor es la cantidad de marcos, menores la accesibilidad de la ventana. $X=1$ entonces 100% $X=2$ 90% $X=3$ 80% $X=4$ 50% $x>4$ 0% y según el trabajo no lleva ninguna sub vista , entonces $x=100$	Multi-nivel, discreto, absoluto	100

4.2.2.2	Versión sin marcos	Cuando se utilizan marcos, es deseable que cuente con una de versión del sitio sin marcos. Solo se pregunta si esta disponible versión sin trames (1) o no es aplicable en caso de no estar diseñado con trames (0)	Elemental Binario, discreto y absoluto.	100
---------	--------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------	-----

## 4.2 EVALUACIÓN GLOBAL

La evaluación global se la realiza con el fin de obtener un indicador de calidad para el sistema. Aplicando un mecanismo de agregación paso a paso, las preferencias de calidad elementales deben estructurarse y agregarse de un modo de abajo hacia arriba (estrategia bottom-up) para permitir el posterior cálculo de las preferencias parciales respectivas. A su vez, repitiendo el proceso de agregación recursivo al final puede obtenerse la estructura de agregación de todo el sistema.

Los valores obtenidos  $IE_i$  de la evaluación realizada en las anteriores Tablas serán la base principal para la evaluación global, utilizando LPS (Logia Scoring of Preference) con la función de la media potencia pesada (ver ecuación 1)

$$IG(r) = (P_1 |E_1|^r + P_2 |E_2|^r + \dots + P_m |E_m|^r)^{1/r} \quad (\text{Ecuación 1})$$

Donde:

$$r \leq +\infty; \quad 0 \leq IE_i \leq 1$$

$$(P_1 + P_2 + \dots + P_m) = 1; \quad P_i > 0; \quad i = 1 \dots m$$

$P_i$  = representa al peso de cada elemento del árbol de requerimientos; estos valores fueron asignados ya por el autor de la metodología web site QEM Dr: Olsina (ver Figuras 2.17).

$r$  = es el exponente de la ecuación (1), guarda el valor real conforme al operador lógico y a la cantidad de entradas seleccionadas para una función de agregación dada.

En la Ecuación 1 se reemplazan los valores obtenidos en la evaluación elemental ( $E_i$ ) y los valores de agregación o peso de cada elemento ( $P_i$ ) para cada etapa de agregación y al final se genera la estructura de agregación de toda la aplicación. Los valores parciales obtenidos de las características usabilidad, funcionalidad, confiabilidad y eficiencia, nos permitirán determinar el valor de la calidad global. A continuación realizamos los cálculos parciales para la usabilidad.

Comenzamos con  $r=3$  pues el numero de entradas de la primera agregación son 3 (ver figura 4.1)

-calculando  $IG(r)$  para la sub-característica 1.1

Reemplazando los valores en la ecuación tenemos:

$$IG_{(3)} = (0.2*100^3 + 0.4*0^3 + 0.4*0^3)^{1/3} = 56.15$$

$$IG_{(4)} = (0.35*56.15^4 + 0.15*0^4 + 0.2*0^4 + 0.3*60^4)^{1/4} = 52.14$$

$$IG_{(2)} = (0.5*100^2 + 0.5*100^2)^{1/2} = 100$$

$$IG_{(3)} = (0.5*100^3 + 0.25*100^3 + 0.25*0^3)^{1/3} = 86.85$$

-calculando  $IG(r)$  para la sub-característica 1.2

$$IG_{(5)} = (0.2*0^5 + 0.2*100^5 + 0.25*86.85^5 + 0.1*60^5 + 0.25*42.56)^{1/5} = 73.85$$

$$IG_{(2)} = (0.5*100^2 + 0.5*100^2)^{1/2} = 100$$

$$IG_{(2)} = (0.4*100^2 + 0.6*100^2)^{1/2} = 100$$

$$IG_{(3)} = (0.2*100^3 + 0.7*60^3 + 0.1*0^3)^{1/3} = 67.65$$

-calculando IG(r) para la sub-característica 1.3

$$IG_{(4)} = (0.15 \cdot 0^4 + 0.3 \cdot 100^4 + 0.3 \cdot 67.61^4 + 0.25 \cdot 0^4)^{1/4} = 74.75$$

-calculando IG(r) para la sub-característica 1.4

$$IG_{(3)} = (0.5 \cdot 60^3 + 0.35 \cdot 0^3 + 0.15 \cdot 0^3)^{1/3} = 45$$

Reemplazando los valores calculados de las sub-características 1.1, 1.2, 1.3, 1.4 en la ecuación obtenemos el IG(r) para la característica de usabilidad.

$$IG_{(4)} = (0.35 \cdot 52.14^4 + 0.3 \cdot 73.27^4 + 0.3 \cdot 74.37^4 + 0.1 \cdot 45.75^4)^{1/4} = 65.10$$

. Finalmente el proceso general de evaluación del proyecto presenta los siguientes resultados

Tabla 4.5 Resultado del proceso de evaluación

Característica	IG(r)
Usabilidad	65.10
Funcionalidad	72.67
Confiabilidad	92.63
Eficiencia	67.42
<b>Calidad global</b>	<b>75.57</b>

Fuente: Elaboración Propia

El 75.57 de acuerdo al rango de aceptación de calidad, que se encuentra representado en la figura 4.4 por tanto el resultado se halla dentro de los márgenes de satisfacción. , por tanto se concluye que el usuario visitante tendrá una grado de satisfacción del 75.57%

### 4.3 ANÁLISIS DE COSTOS

Para determinar el costo total del proyecto se tomará en cuenta los siguientes costos:

- Costo del Software desarrollado.
- Costo de elaboración del proyecto.

#### 4.3.1 Costo del Software desarrollado

Para la determinación del costo del Software desarrollado, se hará uso del Modelo Constructivo de Coste COCOMO II, orientado a los Puntos de Función. La metodología de este modelo se encuentra explicada en el capítulo anterior.

##### 4.3.1.1 Estimación de Puntos de Función

##### a) Cálculo de Puntos de Función no ajustados

Tabla 4.6 Cálculo de Puntos de Función no ajustados

Parámetros de Medición	Cuenta	Factor de Ponderación	Total
Número de entradas de	40	4	160
Número de salidas de	25	5	125
Número de peticiones de	15	4	60
Número de archivos en	22	10	220
Número de interfaces	0	7	0
<b>Cuenta Total</b>			<b>565</b>

Fuente: [PRE 2000]

## b) Cálculo de valores de ajuste de la complejidad

Tabla 4.7 Determinación de Complejidad.

Factor	Valor
Copia de seguridad y recuperación	5
Comunicación de datos	2
Proceso distribuido	3
Rendimiento crítico	3
Entorno operativo existente	4
Entrada de datos en línea (on-line)	4
Transacciones de entrada en múltiples	4
Archivos maestros actualizados en línea	3
Complejidad de valores del dominio de	2
Complejidad del procesamiento interno	4
Código diseñado para la reutilización	2
Conversión/instalación en diseño	2
Instalaciones múltiples	3
Aplicación diseñada para el cambio	4
$\Sigma (F_i)$	45

Fuente: [PRE 2000]

$$\text{Factor de Ajuste} = 0.65 + 0.01 * \Sigma (F_i)$$

$$\text{Factor de Ajuste} = 0.65 + 0.01 * 45$$

$$\text{Factor de Ajuste} = 1.1$$

## c) Cálculo de Puntos de Función

El cálculo de Puntos de Función se basa en la fórmula:

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

$$PF = 565 * 1.1$$

$$PF = 621.5$$



**d) Conversión de los Puntos de Función a KLDC**

Ahora se debe convertir los Puntos de Función a miles de líneas de código. Tomar en cuenta la siguiente tabla.

Tabla 4.8 Conversión de Puntos de Función a KLDC

LENGUAJE	NIVEL	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Ansi Cobol 74	3	107
Visual Basic	7.00	46
ASP	9.00	36
PHP	11.00	29
Visual C++	9.50	34

Fuente:[JONES96]

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

$$LDC = 621.5 * 29$$

$$LDC = 18023.5$$

$$KLDC = (18023.5/1000) = 18.02$$

**e) Aplicación de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido**

Las ecuaciones del CÓCONO básico tienen la siguiente forma:

$$E = a_b (KLDC)^{b_b} \dots\dots\dots (\text{Ecuación } 2)$$

$$D = c_b (E)^{d_b} \dots\dots\dots (\text{Ecuación } 3)$$

Donde:

E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en meses cronológicos

KLDC: Número estimado de líneas de código distribuidas (en miles)

Tabla 4.9 Coeficientes  $a_b$  y  $c_b$  y los exponentes  $b_b$ , y  $d_b$

Proyecto de Software	$a_b$	$c_b$	$b_b$	$d_b$
Orgánico	2.4	1.05	2.5	0.38
Semi - acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Fuente: [PRE2000]

Son proyectos de software semi-acoplados, los proyectos intermedios (en tamaño y complejidad) en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medio rígidos, tal es el caso del software desarrollado.

$$E = 3.0 * (18.02)^{1.12}$$

$$E = 76.5$$

$$D = 2.5 * (76.5)^{0.35}$$

$$D = 11.4$$

El personal requerido, en este caso el número de programadores se obtiene con la siguiente fórmula:

$$\text{Número de programadores} = E / D$$

$$\text{Número de programadores} = 76.5 / 11.4$$

$$\text{Número de programadores} = 6.71 = 7$$

El salario de un programador puede oscilar entre los \$us 200, cifra que será tomada en cuenta para la estimación siguiente:

Costo del software desarrollado = Número de programadores \* salario de un programador

Costo del software desarrollado = 7 \* 200

Costo del Software desarrollado = \$us 1400

#### 4.3.2 Costo de Elaboración del Proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de análisis y recopilación principalmente, estos costos se representan en la siguiente tabla

Tabla 4.10 Costo de Elaboración del Proyecto

DETALLE	IMPORTE (\$us)
Análisis y Diseño del	300
Bibliografía	50
Material de Escritorio	30
Internet	40
Otros	10
<b>TOTAL</b>	<b>430</b>

Fuente: Elaboración Propia

#### 4.3.3 Costo Total

El costo total es la sumatoria del costo del software desarrollado, costo de elaboración del proyecto, detallados en la tabla 4.11

Tabla 4.11 Costo Total del Proyecto

DETALLE	IMPORTE (\$us)
Costo del Software Desarrollado	1400
Costo de Elaboración del Proyecto	430
<b>TOTAL</b>	<b>1830</b>

Fuente: Elaboración Propia

De acuerdo al análisis de costo de COCOMO el presente proyecto de grado tiene un costo total de 1830 \$us

# CAPITULO 5

## CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

Terminando el proceso de construcción del sistema, se concluye que:

- A la culminación de la elaboración, análisis, diseño e implementación, se concluyo satisfactoriamente, llegando a los objetivos planteados
- Se obtuvo un producto con las particularidades de un sistema Web, que contribuye como un aporte tecnológico, que lleva a ser parte de la nueva sociedad de la información a las Unidades Educativas con las que se trabajó.
- El sistema ha logrado adaptarse satisfactoriamente a los requerimientos de las cinco Unidades Educativas con las que se trabajo, teniendo la capacidad de trabajar de manera local como a través de Internet.
- La automatización de la información, facilita la organización de la misma , agilizando los procesos del seguimiento académico de cada una de las Unidades Educativas.
- Se desarrollo el sistema mediante la nueva corriente de metodologías que son las ágiles en este caso usando ASD, la cual gracias a su ciclo de vida permitió realizar iteraciones mas veloces, lo cual favoreció el que se pueda atender a las cinco Unidades Educativas de buena manera y sin perder tiempo, tomando en cuenta que el trabajo fue realizado de manera individual.

- El desarrollo del sistema logro apoyarse eficientemente en las herramientas de modelado que nos brinda el Lenguaje de Modelado Unificado UML con el cual se especifico el análisis y diseño del sistema.
- Debido a diseño Web , el sistema trabaja tranquilamente en Internet dando la facilidad a la Fundación Ecología y Desarrollo centralizar la información de estas cinco Unidades Educativas logrando quebrar los cercos institucionales, a fin de compartir conocimientos entre estas , haciendo que las mismas se conviertan en productoras y no solo en consumidoras de información, dando así mayor protagonismo a sus actores

## 5.2 RECOMENDACIONES

- El sistema a tomado en cuenta el área académica de las Unidades Educativas, pero no el área contable ni el área administrativa se recomienda diseñar estos módulos faltantes tomando en cuenta el esquema de la base de datos con la que se trabajo
- El presente proyecto puede ampliarse a varias unidades Educativas pudiendo convertirse en un aporte para las correspondientes Entidades Estatales encargadas de centralizar los resultados académicos de Unidades Educativas tal es el caso de la Distrital
- Hacer una supervisión al sistema cada gestión para ver el buen funcionamiento, en el almacenamiento de información y procesos concurrentes como inscripciones, consultas y registros

## BIBLIOGRAFÍA

- [OLS1999] Olsina “Metodología Cuantitativa para la Evolución y comparación de la calidad de sitios Web”, Facultad de Ciencias Exactas de la UNLP, Argentina
- [BGH2001] Booch, G. (2001). DEVELOPING THE FUTURE. SOFTWARE SOLUTIONS vol. 44 (3), pp. 119-121, March 2001.
- [JHC2000] José H. Canós, Patricio Letelier y Carmen Penadés, “Metodologías Ágiles en el Desarrollo de Software” ,Universidad Politécnica de Valencia,2000
- [HIJ1999] Highsmith, Jim, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House, 1999.
- [HIJ2000] Highsmith, Jim, “Retiring Lifecycle Dinosaurs,” *Software Testing & Quality Engineering (STQE)*, (July/August 2000) pp. 22-28.
- [PRE2000 ] Roger S. Presman, R. 2000: INGENIERIA DEL SOFTWARE UN ENFOQUE PRACTICO, Quinta Edicion, Editorial McGRAW-HILL, Mexico
- [JAC1998] Ivar Jacobson, Grady booch, James Rumbaugh ,2000 “PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE”
- [LAR1999] Laman G. 1999, UML Y PATRONES, primera edición, editorial Prentice-Hall
- [COT1994] Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13.

- [OMG1999] Object Management Group. 1999. OMG Unified Modeling Language Specification (Draft). Versión 1.3. alfa R5, marzo de 1999. Disponible en: <http://www.rational.com/uml>
- [DWP2000] Diseño Web, Programación, Usabilidad, Hosting Argentina. Córdoba 6345 Rosario Argentina. Disponible en: <http://www.pabloimpallari.com.ar/webdesign/>
- [WIF2006] Wikimedia Foundation, Inc. 2006. Disponible en: <http://es.wikipedia.org/wiki/W3C>
- [BWB1990] B. W. Boehm a finales de los 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981).
- [MAS2001] *Manifesto for Agile Software Development*, Disponible en: <http://www.agilealliance.org/>