

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



**PROYECTO DE GRADO  
SISTEMA DE INFORMACIÓN BASADO EN LA WEB PARA LA  
CARRERA DE INFORMÁTICA DE LA UMSA**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**AUTOR: SUSETH VANIA ROMERO BEDOYA**

**TUTOR: Lic. Efraín Silva Sánchez**

**REVISOR: Lic. Rubén Alcón López**

**LA PAZ-BOLIVIA  
2006**

## DEDICATORIA

Este trabajo se lo dedico a mi querida mamita Wilma, y a mi adorado hermano Edson, por que fueron las personas que siempre estuvieron a mi lado apoyándome, me inculcaron valores y ejemplos de vida, que son dignos de reconocer.

Decirte a ti mamá que alcancé una de las metas que me encargaste cuando ingrese a la Universidad; éste es el inicio de una nueva etapa en mi vida y te lo dedico de todo corazón a ti mi querida mamita y a mi ausente y recordado papá que el señor tenga en su reino.

Con mucho amor:

Vania.

## **AGRADECIMIENTOS**

Quiero agradecer a nuestro creador por haberme ofrecido la posibilidad de vida, a mis queridos papas que se esforzaron tanto por hacer de mi una persona educada y útil a la sociedad, creo que este es el complemento que ellos esperaron, el verme enrumbada en el camino de la profesión informática.

A mi tutor Lic. Efraín Silva Sánchez y a mi revisor Lic. Rubén Alcón López, agradecerles por los consejos, la paciencia y las acertadas y oportunas correcciones que han hecho de este trabajo un ejemplo digno del enorme sacrificio que realizan los docentes de la Carrera de Informática en beneficio de los estudiantes. Vaya en nombre de ellos un reconocimiento especial al resto del personal docente y administrativo de la Carrera que permitieron mi formación profesional.

A mis amigos de la carrera agradecerles por su apoyo y amistad para la conclusión de este trabajo y por los varios consejos para no desmayar y seguir adelante.

Al personal administrativo de la Carrera, decirles muchas gracias por su colaboración, a los señores bibliotecarios Daniel y Fernando agradecerles por los innumerables consejos en los libros recomendados para su lectura.

Finalmente agradecer a la Universidad Mayor de San Andrés por permitirme haber visitado y estudiado en sus aulas, por haberme formado en esta hermosa profesión que es la informática y por que estoy orgullosa de ser parte de la mejor universidad de nuestro país.

## RESUMEN

En la década de los años 1990, la aparición de Internet fue totalmente revolucionaria a nivel mundial para los estándares de comunicación conocidos hasta ese momento, generando grandes y positivos cambios en todo el mundo, rápidamente la Web se pobló de información de todo tipo, sin embargo, las páginas estáticas dominaban el panorama en esa etapa y los contenidos dinámicos tardaron unos años más en aparecer. Con el paso del tiempo y la utilización de las tecnologías propias de Internet, se logró el desarrollo de aplicaciones distribuidas, basadas en el modelo cliente/servidor. Actualmente en el ámbito organizacional, la información es considerada como un activo base e importante para la toma de decisiones, y con el servicio de Internet ha surgido la necesidad de utilizar Tecnologías Web para comunicar y transportar información. Considerando lo anteriormente expuesto, se desarrolla el presente proyecto de grado denominado: "Sistema de Información Basado en la Web para la Carrera de Informática" que viene a satisfacer requerimientos específicos de información y comunicación de y con la comunidad informática, las entidades con quienes mantiene una relación de trabajo, y el hecho de informar a todos los interesados que desean conocer los servicios que presta la Carrera de Informática. Para el desarrollo del proyecto, se hace uso de una metodología seleccionada que permite el análisis y diseño de aplicaciones hipertexto orientadas a objeto, paralelamente se utiliza la extensión del Lenguaje de Modelado Unificado para modelar la solución al dominio del problema, antes de proseguir con el diseño de la aplicación, además se toma en cuenta la especificación de requerimientos del software establecida por el Instituto de Ingenieros Eléctricos y Electrónicos de los Estados Unidos, para cumplir con los requerimientos identificados en el análisis, además de establecer una integridad conceptual del sistema. Se utiliza el lenguaje de programación PHP, que está especializado en el desarrollo de aplicaciones Web dinámicas; se hace uso del gestor de base de datos MySQL para manipular la información que se maneja en la Carrera de Informática. El servidor Web Apache permite dar el servicio de almacenamiento físico de los archivos que componen la aplicación. Finalmente se hace una evaluación de calidad Web, utilizada a partir de un marco conceptual de calidad que refleja el aprovechamiento de las metodologías y técnicas utilizadas.

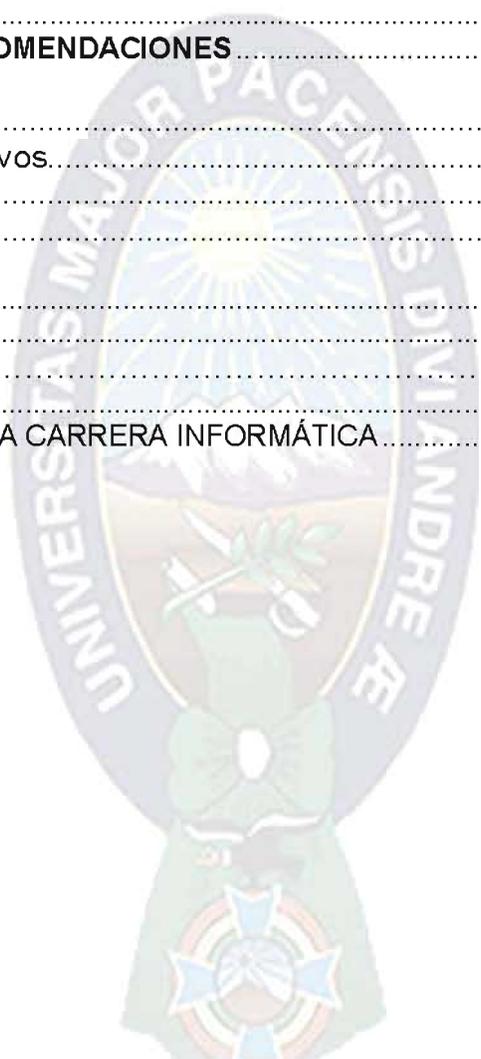
**Palabras clave:** Internet, Web, tecnología, información, comunicación, hipertexto, lenguaje de modelado unificado, servidor Web.

# ÍNDICE

<b>Descripción</b>	<b>Pág.</b>
<b>PRELIMINARES</b> .....	i
DEDICATORIA.....	ii
AGRADECIMIENTOS.....	iii
RESUMEN.....	iv
ÍNDICE.....	v
<b>CAPÍTULO 1</b> .....	1
<b>MARCO REFERENCIAL</b> .....	1
1.1. Introducción.....	1
1.2. Antecedentes .....	2
1.2.1. Marco institucional.....	3
1.4. Planteamiento del problema.....	6
1.4.1. Problema principal.....	6
1.4.2. Problemas específicos .....	6
1. 5. Objetivos .....	7
1. 5.1. Objetivo principal.....	7
1.5.2. Objetivos secundarios.....	7
1.6. Justificación del proyecto .....	8
1.6.1. Justificación social.....	8
1.6.2. Justificación tecnológica.....	9
1.6.3. Justificación operacional .....	9
1.7. Límites y alcances.....	9
1.8. Metodologías.....	10
1.8.1. Planificación de proyectos.....	10
1.8.2. Análisis y diseño de la solución.....	11
1.9. Técnicas y herramientas .....	11
<b>CAPÍTULO 2</b> .....	12
<b>MARCO TEORICO</b> .....	12
2.1. Tecnología orientada a objetos .....	13
2.1.1. Estado actual y tendencias orientadas a objetos .....	13
2.1.2. Procesos de desarrollo y gestión .....	14
2.1.3. Tecnología orientada a objetos.....	14
2.2. Tecnología web .....	15
2.3. Especificacion de requisitos del software.....	15
2.3.1. Definiciones.....	16
2.3.2. Consideraciones para producir un buen requerimiento.....	16
2.4. Metodología de diseño hipermedia orientado a objetos.....	21
2.4.1. Etapas de la metodología oohdm.....	24
2.4.2. Hipertexto www.....	29
2.5. Aplicaciones web con php.....	31

2.5.1. Interacción en internet.....	31
2.5.2. Servidor apache .....	32
2.5.3. Php.....	34
2.6. Lenguaje de modelado unificado.....	36
2.6.1. Modelos.....	37
2.7. Bases de datos relacionales.....	49
2.7.1. Modelo relacional .....	51
2.7.2. Gestor de base de datos relacional mysql .....	52
2.8. Métricas de calidad .....	54
2.8.1 heurísticas de usabilidad para la web .....	55
2.8.2. Métricas de calidad web-site gem .....	58
2.9. Aspectos de seguridad web .....	59
2.9.1. Criptografía asimétrica, claves privada y pública .....	59
2.9.2. Firma de datos .....	60
2.9.3. Políticas de seguridad.....	60
<b>CAPÍTULO 3 .....</b>	<b>62</b>
<b>DESARROLLO DE LA APLICACIÓN .....</b>	<b>62</b>
3.1. Ingeniería de requerimientos.....	62
3.1.1. Especificación de requerimientos para la aplicación .....	62
3.1.2. Descripción general del escenario .....	63
3.1.3. Análisis de requisitos.....	65
3.1.4. Especificación de requerimientos del usuario .....	67
3.2. Diseño del sistema .....	68
3.2.1. Identificación de actores y tareas.....	68
3.2.2. Especificación de escenarios .....	69
3.2.4. Diagramas de caso de uso.....	80
3.2.5. Diagramas de interacción de usuarios .....	82
3.3. Diseño conceptual.....	85
3.3.1. Diagrama de clases.....	86
3.4. Diseño navegacional .....	87
3.4.1. Contexto navegacional del alumno .....	87
3.4.2. Contexto del docente .....	88
3.11.3. Contexto de los funcionarios administrativos .....	88
3.4.4. Contexto del administrador de la aplicación .....	89
3.4.5. Contexto del usuario ocasional .....	90
3.5. Diseño de la interfaz abstracta .....	91
3.5.1. Interfaz abstracta de ingreso a la parte restringida de la aplicación.....	91
3.5.2. Interfaz a la página personal del docente.....	92
3.5.3. Interfaz publicar material intelectual .....	94
3.5.4. Interfaz de mensajes.....	94
3.5.5. Interfaz menú de administración de servicios de la unidad .....	95
3.5.6. Interfaz de avisos y circulares .....	95
3.5.7. Interfaz de noticias .....	96
3.5.8. Interfaz del calendarios de actividades .....	97
3.5.9. Interfaz página personal del alumno .....	97

3.5.10. Interfaz página principal .....	98
3.6. Implementación .....	99
3.6.1. Modelo de datos relacional .....	100
3.6.2. Modelo de datos en mysql .....	101
3.6.3. Interfaces del sistema .....	102
3.7. Métricas de calidad .....	105
3.8. Definición e implementación de la evaluación .....	107
3.9. Evaluación global .....	115
<b>CAPITULO 4</b> .....	<b>117</b>
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>117</b>
4.1. Conclusiones .....	117
4.2. Estado de los objetivos .....	117
4.3. Recomendaciones .....	120
4.4. Trabajos futuros .....	120
<b>BIBLIOGRAFÍA</b> .....	<b>121</b>
<b>NORMAS</b> .....	<b>124</b>
<b>ANEXOS</b> .....	<b>125</b>
ANEXO A GLOSARIO .....	125
ANEXO B ORGANIGRAMA CARRERA INFORMÁTICA .....	127



# ÍNDICE DE FIGURAS Y TABLAS

<b>Figuras</b>	<b>Detalle</b>	<b>Pág.</b>
Fig. 2.1.	Esquema general de las tecnologías Web	28
Fig. 2.2.	Relaciones entre objetos y esquema navegacional y conceptual	35
Fig. 2.3.	Vista abstracta de datos	36
Fig. 2.4.	Definición de un ADV	40
Fig. 2.5.	Arquitectura 2-capas (Cliente/Servidor)	43
Fig. 2.6.	Arquitectura de N-capas	44
Fig. 2.7.	Servidor Apache y peticiones Internet	45
Fig. 2.8.	Historia de UML	49
Fig. 2.9.	Notación para clases a distintos niveles de detalle	51
Fig. 2.10.	Ejemplo de objetos	51
Fig. 2.11.	Ejemplo de asociación con nombre y dirección	52
Fig. 2.12.	Ejemplos de multiplicidad en asociaciones	53
Fig. 2.13.	Ejemplo de roles en una asociación	53
Fig. 2.14.	Ejemplo de agregación	54
Fig. 2.15.	Ejemplo de clase asociación	54
Fig. 2.16.	Ejemplo de asociación ternaria	55
Fig. 2.17.	Ejemplo de herencia	56
Fig. 2.18.	Ejemplo de atributo derivado	56
Fig. 2.19.	Diagrama de casos de uso	57
Fig. 2.20.	Diagrama de secuencia	59
Fig. 2.21.	Diagrama de colaboración	59
Fig. 2.22.	Diagrama de estados	60
Fig. 3.1.	Diagrama de casos de uso general	92
Fig. 3.2.	Diagrama de casos de uso general (continuación)	93
Fig. 3.3.	Diagrama de casos de uso general (continuación)	93
Fig. 3.4.	Diagramas de interacción de usuarios	94
Fig. 3.5.	Diagramas de interacción de usuarios (continuación)	95
Fig. 3.6.	Diagramas de interacción de usuarios (continuación)	96
Fig. 3.7.	Diagramas de interacción de usuarios (continuación)	97
Fig. 3.8.	Diagrama de clases	98
Fig. 3.9.	Diagrama de contexto navegacional del alumno	99
Fig. 3.10.	Diagrama de contexto navegacional del docente	100
Fig. 3.11.	Diagrama de contexto navegacional del funcionario administrativo	101
Fig. 3.12.	Diagrama de contexto navegacional del administrador de la aplicación	102
Fig. 3.13.	Diagrama de contexto navegacional del usuario ocasional	103
Fig. 3.14.	ADV Ingreso de usuarios registrados	104
Fig. 3.15.	ADV página personal del docente	105
Fig. 3.16.	ADV publicación del material intelectual	106
Fig. 3.17.	ADV enviar mensaje	107
Fig. 3.18.	ADV administración de servicios	107
Fig. 3.19.	ADV avisos y circulares	108
Fig. 3.20.	ADV de noticias	108
Fig. 3.21.	ADV de actividad	109
Fig. 3.22.	ADV página personal del alumno	110
Fig. 3.23.	ADV página principal	111
Fig. 3.24.	Modelo relacional del dominio del problema	112
Fig. 3.25.	Base de datos en MySql	113

Fig. 3.26	Interfaz Principal de acceso a la aplicación	114
Fig. 3.27	Interfaz de ingreso a la parte administrativa de la aplicación	114
Fig. 3.28	Interfaz de Ingreso de usuarios registrados	115
Fig. 3.29	Interfaz de Roles de usuario	115
Fig. 3.30	Interfaz de Módulos del sistema	116
Fig. 3.31	Interfaz de Unidades del sistema	116

<b>Tablas</b>	<b>Detalle</b>	<b>Pág.</b>
Tabla 2.1.	Etapas de la metodología OOHDM	37
Tabla 2.2.	Definición de los nodos	39
Tabla 3.1.	Requerimientos del software para la aplicación	74
Tabla 5.1.	Atributos de calidad considerados en Web-site QeM	117
Tabla 5.2.	Resultado de las preferencias elementales de usabilidad	119
Tabla 5.3.	Resultados de la evaluación elemental de funcionalidad	121
Tabla 5.4.	Resultados de la evaluación de confiabilidad	125
Tabla 5.5.	Resultados de la evaluación elemental de eficiencia	126
Tabla 5.6.	Resultados de las evaluaciones parciales y globales	128
Tabla 5.7.	Características de alto nivel y evaluación global	128



# CAPÍTULO 1

## MARCO REFERENCIAL

### Resumen

*Este capítulo presenta el marco inicial para el desarrollo del proyecto, se describen los antecedentes, el marco institucional de la Carrera de Informática, el análisis preliminar, la problemática a resolver, los objetivos, la justificación y los aportes de la solución a los problemas planteados.*

### 1.1. INTRODUCCIÓN

El éxito y la supervivencia de las organizaciones depende de una administración efectiva de la información; en la actualidad la información es considerada como el activo<sup>1</sup> de mayor valor que permite la toma de decisiones; teniendo en cuenta éste hecho la tecnología se preocupó por darle un mejor tratamiento, con el progreso de la informática y las redes de comunicación, ahora la información es presentada en un nuevo escenario, donde los objetos del mundo real están representados por bits y bytes [ALSI, 2006]

Es vital para las organizaciones aprovechar los medios tecnológicos de comunicación. Desde la aparición de Internet<sup>2</sup> se ha ido explotando éste recurso para publicitar productos que se encuentran a la venta, dando lugar a la publicación estática de información en una sociedad globalizada. Con el pasar del tiempo la humanidad se fue dando cuenta que éste recurso podía ser bien utilizado en diferentes áreas, además de las comerciales, viendo la posibilidad de que muchas de las transacciones de información podían ser atendidas a través de la Web; este aspecto despertó las necesidades de obtener y dar servicios a través del Internet, sin restricción de tiempo y lugar; por ejemplo en las áreas de educación, empresas internacionales de negocios, organizaciones que se dedican a ofrecer servicios; viendo la posibilidad de que muchas de las transacciones podían ser atendidas a través de la Web con las medidas de seguridad apropiadas, para generar estrategias de difusión de información efectiva sobre los servicios y productos que ofertan las organizaciones.

---

<sup>1</sup> Un activo es todo aquel elemento que compone el proceso de la comunicación, partiendo desde la información, su emisor, el medio por el cual se transmite, hasta su receptor.

<sup>2</sup> Nombre de la mayor red informática del mundo. Red de telecomunicaciones nacida en 1969 en los EE.UU. a la cual están conectadas centenas de millones de personas, organismos y empresas en todo el mundo, mayoritariamente en los países más desarrollados, y cuyo rápido desarrollo está teniendo importantes efectos sociales, económicos y culturales, convirtiéndose de esta manera en uno de los medios más influyentes de la llamada Sociedad de la Información y en la Autopista de la Información por excelencia. Fue conocida como ARPANET hasta 1974.

Nuevos conceptos sobre como mejorar los servicios de comunicación entre las organizaciones y sus clientes han sido desarrollados gracias a las nuevas tecnologías de información y comunicación (TIC). Uno de los más significativos, es la llamada “ventanilla única” o “ventanilla virtual”; bajo este concepto se han integrado un conjunto de tecnologías que buscan facilitar la interacción de los individuos con diferentes organizaciones y empresas que les brindan diversos servicios, entre otras cosas agilizar los procesos de trámites para que los mismos sean informatizados, automatizar el intercambio de datos y evitar que los trámites sean burocráticos.

El carácter estratégico para la superación de la carrera de Informática de la Universidad Mayor de San Andrés (UMSA) hace que se identifique la necesidad imperiosa de contar con un sistema de comunicación basado en la Web, falencia que es atendida con el presente proyecto de grado, que viene a coadyuvar los procesos de administración, trámites e interacción social en ésta Carrera, en base a la utilización de tecnologías Web.

## **1.2. ANTECEDENTES**

En sus inicios la Carrera de Informática confrontó diversos problemas académicos, provenientes de la falta de profesionales en el área y la carencia de una infraestructura adecuada, lo cual impedía una correcta educación. Posteriormente la carrera creció en población estudiantil y se plantearon nuevos objetivos en cuanto a la administración a pesar que el aspecto comunicativo no significaba inconveniente alguno hasta entonces; como consecuencia se vio la necesidad de declararla, en 1974 como una Carrera de la Facultad de Ciencias Puras y Naturales (FCPN).

La introducción del Bachiller Superior en Ciencias ha significado un impulso en la titulación de nuevos especialistas de la Informática sin descuidar el nivel de la licenciatura que ha tenido un gran impacto tanto cuantitativo como cualitativo.

Actualmente la Carrera de Informática es la más numerosa en la FCPN, constituyendo el 80% de su matrícula, lo que requiere un mayor orden con los expedientes de las personas que se profesionalizan como también para las personas que se encuentran a cargo de la administración y deben mantener una comunicación constante con los beneficiarios para

ponerlos al tanto de cuales son los procedimientos de rigor que se deben cumplir en la carrera; en cuanto al modo de administración de los datos ha ido variando con el tiempo es por eso que los estudiantes, docentes y administrativos de la Carrera de Informática deben mantenerse al tanto de cuál es el correcto funcionamiento de cada trámite por su cuenta, lo cual significa que cada persona se hace una idea diferente de cómo debe proceder con los trámites. Tal es el caso cuando el estudiante se encuentra cursando la materia de noveno semestre (Taller de licenciatura I); y debe iniciar sus trámites de conclusión de materias, posteriormente iniciar los trámites para acceder a la defensa de Tesis de Grado o proyecto de Grado que también es considerado en el plan de estudios correspondiente, luego a la titulación.

La Universidad Mayor de San Andrés (UMSA), en su administración, está estructurada de forma jerárquica y se encuentra compuesta por distintas unidades encargadas de diferentes tareas. La Carrera de Informática como unidad académica de formación de recursos humanos para la sociedad, actualmente cuenta con las siguientes unidades administrativas:

1. Sub Dirección Académica (SDA)
2. Sub Dirección de Investigación (SDI)
3. Laboratorio Superior de Informática (LASIN)
4. Dirección de Carrera (DC)
5. Biblioteca Especializada de Informática (BEI)
6. Centro de Estudiantes de Informática (CEI)

### 1.2.1. Marco Institucional

a) **Sub Dirección Académica.** La SDA es una unidad administrativa que tiene como función fundamental administrar toda la información relacionada con la tarea de formación y capacitación de los estudiantes de la Carrera, en función de las normas y reglamentos vigentes en la Universidad Mayor de San Andrés. Su labor es importante porque a partir de la información que dispone es posible respaldar el nivel de formación obtenido por los estudiantes, además de las experiencias académicas y otros logros que hubieran conseguido los mismos. La SDA se encarga de procesar información útil en las instancias de cogobierno de la UMSA, en las tareas que desarrollan los docentes de la Carrera y en la toma de decisiones de los estudiantes en relación a su formación.

- b) **Sub Dirección de Investigación.** La SDI registra y almacena en la base de datos: la información necesaria de los proyectos que realizan los docentes como parte de su labor académica; además de disponer de información relacionada con tareas de control de proyectos y los recursos asignados; esto con el fin de contar con información fidedigna y esencial de los antecedentes de los docentes en materia de investigación debidamente actualizados por el mismo; además de registrar los resultados de las evaluaciones anuales; este último es considerado para que el docente pueda ser evaluado. Se toma en cuenta los artículos o tareas de investigación extras que podría haber realizado el docente, así como tareas extras que pudiera tener (tareas de investigación interdisciplinarias o de asesoría). Entre otras tareas que realiza la SDI están las de: coordinar, supervisar y controlar los procesos de asignación de personal docente, seguimiento y evaluación de proyectos con otras unidades de la Universidad que norman las actividades de las tareas de investigación en la UMSA (DIPGIS, COFIS, CONGRESOS UNIVERSITARIOS, otros.)
- c) **Laboratorio Superior de Informática.** El LASIN se encarga de planificar, organizar, controlar y ejecutar el conjunto de tareas como ser: dictado de cursos, prueba de programas, manejo e instalación de hardware y software, capacitación en el manejo de paquetes de distinta categoría: lenguajes de programación, gestores de Bases de Datos, Hojas electrónicas, etc., complementarias en la formación académica de los estudiantes de la Carrera. También organiza cursos de formación y capacitación autofinanciados en paquetes de ofimática orientados a personas particulares que lo requieran. El LASIN a través de su jefatura coordina la prestación de servicios de asistencia a otras unidades de la Universidad y elabora informes relacionados con la planificación de sus actividades y con los recursos económicos que genera a fin de ser remitidos a las instancias de cogobierno y administrativas de la Carrera y Facultad (Área Desconcentrada de la Facultad de Ciencias Puras y Naturales, Consejo de Carrera, Jefatura de Carrera, otros).
- d) **Dirección de Carrera.** La DC es una de las unidades más importantes de la Carrera ya que está encargada de; planificar, dirigir, coordinar, supervisar y decidir acerca del conjunto de tareas académicas, administrativas y de servicio que realizará la Carrera durante una gestión académica universitaria. Otra de las tareas de la DC es coordinar, supervisar y controlar los procesos de admisión, designación y seguimiento curricular de

docentes y auxiliares de docencia encargados de las tareas de formación académica, se rige mediante normas y reglamentos para el desarrollo de procedimientos administrativos de los diversos trabajos que se realizan en las actividades de la Carrera.

- e) **Biblioteca.** La función principal de esta unidad es gestionar los recursos bibliográficos de la Carrera de Informática en favor de la comunidad docente-estudiantil; su labor está enmarcada en las normas y reglamentos vigentes de la Universidad. Se encarga también de registrar, clasificar y organizar la producción literaria de docentes y estudiantes de la Carrera; además de actualizar los recursos bibliográficos a través de tareas de coordinación con los docentes y las autoridades de las áreas comprometidas en esta labor. Organiza actividades extras destinadas a promocionar y obtener opiniones acerca de los recursos bibliográficos adquiridos para mejorar y ampliar el stock de libros disponibles.
- f) **Centro de Estudiantes de Informática.** El CEI es una organización representante de la población estudiantil en las distintas asambleas y reuniones donde existen integrantes del cogobierno, para la toma de decisiones respecto a los aspectos relacionados con la formación de los estudiantes, otras de las tareas que desempeña es la organización coordinada de eventos sociales, para el beneficio de los estudiantes, tales como la realización de seminarios, cursos, campeonatos deportivos y otros.

### 1.3. ANÁLISIS PRELIMINAR

Actualmente la carrera de Informática cuenta con un sistema académico, a través del cuál el estudiante puede inscribirse cada gestión, hacer consultas sobre su record académico, descargar la papeleta de inscripción, consultas relativas con horarios y plan de estudios.

Si bien el portal es dinámico, no responde las preguntas frecuentes que tienen los estudiantes, como ser: ¿cuáles son los requisitos y procedimientos que el estudiante tiene que seguir para los trámites que se efectúan en la carrera?; tampoco muestra la información que se genera en la carrera como ser: noticias, avisos, circulares, convocatorias, calendario de actividades, datos de los docente, etc.

Otro aspecto es que los encargados de administrar el sistema no tienen la facilidad de manejo del mismo, es por eso que el servicio es interrumpido en ocasiones, para realizar la actualización de los datos.

#### **1.4. PLANTEAMIENTO DEL PROBLEMA**

Durante la fase de análisis preliminar se aplicaron varias técnicas que ayudaron a identificar el problema:

1. Entrevistas. Se realizaron entrevistas directas con las persona responsables de la administración de la información de la Carrera de Informática; desde el Jefe de Dirección, jefes de subdirección y sus respectivas secretarías que procesan la información, además de atender personalmente a docentes y estudiantes de la Carrera.
2. Recopilación de Documentos. Impresos que se publican en la vitrinas de la Carrera clasificados por: Avisos, Circulares, Convocatorias, Noticias, Calendario de Actividades, Reglamento del Régimen Académico – Docente de la Universidad Boliviana, Normas y Reglamentos Universitarios del Personal Docente.
3. Observación. Se realizaron observaciones del curso de los trámites, la duración de los mismos; la semejanza de procedimientos que existe con otras carreras y unidades administrativas de la UMSA en cuanto la forma de administrar la información que deben dar a conocer a sus usuarios.

##### **1.4.1. Problema Principal**

Los canales de comunicación y difusión de información con los que cuenta la Carrera de Informática de la Universidad Mayor de San Andrés no hacen uso de las Tecnologías Web adecuadas, lo que dificulta dar a conocer los servicios que la Carrera presta a la comunidad informática.

##### **1.4.2. Problemas Específicos**

- a) No se cuenta con los medios apropiados para difundir los avisos, circulares, anuncios y noticias destinados a informar a la comunidad informática.

- b) Gran parte de la comunidad informática desconoce cual es la estructura organizativa de la Carrera de Informática, además de los servicios que ofrece.
- c) Muchos integrantes de la comunidad informática desconocen el cronograma de actividades establecido por la Carrera de Informática.
- d) No se informa adecuadamente a la comunidad informática, acerca de los requisitos y pasos que se deben cumplir para un determinado trámite.
- e) Existe informalidad en la presentación de documentos que se requieren para realizar los diferentes trámites que oferta la Carrera de Informática.
- f) Algunos estudiantes no actualizan su archivo personal, en la oficina de Kardex, con la documentación correcta o faltante, debido a que no cuentan con un medio que les informe acerca del estado del mismo.
- g) Los docentes no cuentan actualmente con un medio para dar a conocer su producción intelectual y hoja de vida.
- h) Una vez iniciado un trámite, el estudiante desconoce el estado en el que se encuentra el mismo.
- i) Existe desconocimiento de las leyes nacionales, normativas internas y reglamentos vigentes por parte de la comunidad informática.

## **1. 5. OBJETIVOS**

### **1. 5.1. Objetivo Principal**

Los canales de comunicación y difusión de información con los que cuenta la Carrera de Informática de la Universidad Mayor de San Andrés, harán uso adecuado de las Tecnologías Web, y facilitaran la información de los servicios que la Carrera presta a la comunidad informática.

### **1.5.2. Objetivos Secundarios**

- a) Construir los medios apropiados para difundir los avisos, circulares, anuncios y noticias destinados a informar a la comunidad informática.
- b) Dar a conocer a la comunidad informática la estructura organizativa de la Carrera, además de los servicios que ofrece.

- c) Presentar a la comunidad informática el cronograma de actividades planificado por las autoridades de la Carrera de Informática.
- d) Informar adecuadamente a la comunidad informática, acerca de los requisitos y pasos que se deben cumplir para un determinado trámite.
- e) Desarrollar los mecanismos formales para la presentación de los documentos que se requieren para realizar los diferentes trámites que oferta la Carrera de Informática.
- f) Proporcionar un medio que permita la visualización del contenido asociado al archivo personal del estudiante, para la actualización de su documentación.
- g) Entregar un medio que se encargue de publicar la producción intelectual y hoja de vida de los docentes.
- h) Mantener informados a los estudiantes sobre el estado y avance de sus trámites.
- i) Mostrar a la comunidad informática, los documentos que componen las leyes nacionales, normativas internas y reglamentos vigentes.
- j) Construir un sistema de información basado en la Web para la administración adecuada de los canales de comunicación y difusión de los trámites que se realizan para beneficio de la comunidad informática.
- k) Realizar las pruebas de calidad al sistema construido.

## **1.6. JUSTIFICACIÓN DEL PROYECTO**

Uno de los elementos centrales para el mejoramiento de los servicios dentro de las instituciones, es el fortalecimiento de la comunicación entre sus usuarios y la organización. Para lograr esto dentro de la Carrera se busca la inclusión de la Tecnología Web, como elemento central para la divulgación de información que está dirigida a la comunidad universitaria.

### **1.6.1. Justificación Social**

El presente proyecto beneficia principalmente a la comunidad estudiantil, docentes y funcionarios administrativos de la Carrera de Informática de la Universidad Mayor de San Andrés; proporcionándoles el medio por el cuál las unidades pertenecientes a la carrera, darán a conocer la información referente a sus actividades; información que es de gran

interés para estudiantes y público en general; permitiéndoles un acceso inmediato a la información a través de Internet.

### **1.6.2. Justificación Tecnológica**

Se cuenta con la tecnología mínima, necesaria para el desarrollo e implementación del sistema con tecnologías Web propuesto, además del equipamiento necesario de recursos hardware y software y el conocimiento que permite encarar con perspectivas de éxito el presente proyecto de grado, aspectos que ayudan a garantizar el correcto desarrollo e implementación del mismo.

### **1.6.3. Justificación Operacional**

El hecho de que actualmente Internet se haya convertido en una herramienta de gran uso y además considerando que el número de usuarios del mismo está en ascenso, hace que este tipo de sistemas sea fácil de operar por los usuarios, funcionarios administrativos, docentes y estudiantes de la Carrera de Informática.

## **1.7. LÍMITES Y ALCANCES**

La tecnología con la que se desarrolla el sistema no tiene una limitación física o de tiempo respecto al acceso a la información contenida en él.

El sistema permitirá la difusión de Información Institucional con carácter público y de acceso irrestricto, aspectos institucionales indispensables como la misión, visión, objetivos de la carrera, estructura organizacional, antecedentes históricos e información sobre su ubicación serán los tópicos incluidos dentro de la solución final.

El sistema también permitirá a la comunidad informática interactuar y actualizar información sensible en forma segura y por medio de mecanismos que garanticen la confidencialidad de esta, el docente podrá gestionar en línea la información de sus datos personales. El estudiante estará continuamente informado sobre los trámites que mantiene en ejecución y de los documentos que contiene su archivo personal.

Un aspecto que apoya a la gestión interna dentro la carrera es el mantener accesible información relativa a las funciones y procedimientos internos, como herramientas de apoyo para un normal desenvolvimiento de sus actividades, esto hace referencia a que debe existir un centro de documentación donde se consulten las normativas y reglamentos vigentes en la carrera e información organizacional.

Existirán secciones orientadas a esclarecer y precisar los pasos necesarios para realizar un trámite, complementado con noticias, avisos, circulares y convocatorias, que permitan a los usuarios estar enterados sobre algún cambio o una nueva fecha de presentación de documentos.

## **1.8. METODOLOGÍAS**

### **1.8.1. Planificación de Proyectos**

El Marco Lógico [Rosenberg & Posner, 1979] fue desarrollado por la Agencia de los Estados Unidos para el Desarrollo Internacional como una herramienta para ayudar a conceptualizar un proyecto y analizar sus premisas. Desde el desarrollo del Marco Lógico, éste ha sido adoptado, con varias adaptaciones por un gran número de organizaciones bilaterales e internacionales de desarrollo. El Marco Lógico ha sido extremadamente valioso para el diseño, ejecución, monitoreo y evaluación de proyectos.

En la preparación del perfil de proyecto, hay una interrelación lógica entre el Problema total, la Meta, el Objetivo Específico, los Resultados Esperados, y las Actividades. El Marco Lógico facilita un análisis de estas interrelaciones y sus relaciones con el ambiente que las rodea. A partir del análisis del perfil de proyecto, es evidente que hay una intercalación lógica entre la meta, los objetivos específicos, los resultados esperados y las actividades.

Se utilizó el Marco Lógico para la planificación del perfil del proyecto, debido a que esta metodología contribuyó a eliminar las repeticiones y contradicciones que pudieron haber surgido en la definición de los objetivos.

### 1.8.2. Análisis y diseño de la solución

Para el análisis y desarrollo del sistema se utiliza la ingeniería del software orientado a objetos con una extensión natural al tratamiento de aplicaciones Web. Para el análisis, diseño y desarrollo de la solución se emplea también, el método para el diseño de hipermedia<sup>3</sup> que en un principio fue utilizado para el desarrollo de aplicaciones hipermedia y que en la actualidad se utiliza para desarrollar aplicaciones Web.

Como complemento se utiliza el Lenguaje de Modelado Unificado, que ofrece una notación gráfica y fácil de entender que es aplicable a cualquier tipo de proyecto, este lenguaje es utilizado para la consolidación de los requerimientos, para el planteamiento del modelo de la solución y para el desarrollo de la navegación y las interfaces.

### 1.9. TÉCNICAS Y HERRAMIENTAS

Para el desarrollo de éste proyecto se considera la utilización de las siguientes herramientas:

- Sistema Operativo. Debido a la naturaleza del producto final, este podrá estar bajo una plataforma Windows o Linux.
- Lenguajes de Programación. Los lenguajes de programación que se utilizarán serán: PHP, JavaScript.
- Bases de Datos. MySql
- Herramienta de diseño gráfico. Flash, Free Hand, FIRE Works
- Herramientas CASE. Power Designer.

---

<sup>3</sup> OOHDM (Object Oriented Hypermedia Desing Method)

## CAPÍTULO 2

### MARCO TEORICO

#### Resumen

*Este capítulo presenta la teoría asociada a contribuir con la solución de la problemática planteada en el capítulo 1, las tecnologías usadas por el proyecto como la arquitectura del sistema y del software, también se plantean las metodologías usadas para el análisis y diseño de la solución, se hace una descripción de la metodología OOHDM, así como del lenguaje de modelado UML, por último se habla del modelo relacional para implementar las bases de datos utilizadas.*

La aparición de Internet a nivel mundial en la década de los años 1990 fue totalmente revolucionaria para los estándares de comunicación conocidos hasta ese momento. Rápidamente la Web se pobló de información de todo tipo. Sin embargo, los contenidos dinámicos tardaron unos años más en aparecer, y las páginas estáticas dominaban el panorama en esa etapa. La llegada de lenguajes de programación que pueden embeberse dentro del código HTML como por ejemplo PHP, cuyo acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos Web, supuso un adelanto importante en la interactividad en las páginas Web que componían la World Wide Web. Un tiempo después, gracias a los desarrollos alcanzados en la comunidad del software libre, Internet tomó un impulso importante y se convirtió en fuente principal de información a nivel mundial [Vujacich & Rodríguez, 2005].

En la segunda mitad de los años 90, el lenguaje PHP y el sistema de base de datos MySQL, y el servidor Web Apache, se fueron convirtiendo en los estándares de las páginas dinámicas de todo tipo de información. A partir de los últimos cinco años aproximadamente, éste conjunto de programas de código abierto, son un verdadero estándar para la construcción de sistemas de información dinámicos en Internet.

Un sistema de información en Internet es un conjunto de páginas Web dinámicas e interactivas, alojadas en un servidor Web específico, y con soporte en un sistema de base de datos relacional preparado para tal fin. Los portales de información pueden clasificarse según sus objetivos en varios tipos: Portales de información multipropósito o generales: son los llamados Content Management Systems (CMS) o sistemas administradores de contenido,

Blogs o bitácoras de noticias, foros de discusión, galerías de imágenes, Wikis o sistemas de texto comunitario para documentación, sistemas de educación a distancia (e-learning) y Groupware o portales para trabajo en grupo [Vujacich & Rodríguez, 2005].

## 2.1. TECNOLOGÍA ORIENTADA A OBJETOS

Desde el nacimiento de los primeros lenguajes de programación orientados a objetos, en la década de los años 1960, la Tecnología de Objetos (TO) se ha presentado como la solución a muchos de los problemas del desarrollo de software, consolidándose, en la década de los años 1990, como el paradigma de desarrollo para sistemas de complejidad media o alta<sup>4</sup>. Además, proporciona otra serie de ventajas como facilitar la reutilización de código o permitir obtener aplicaciones más fácilmente escalables, [Heinckiens, 1998]. En la actualidad, la TO parece ser también la que mejor se adapta a los nuevos tipos de desarrollos hipertexto y Web [Schawe y Rossi, 1995].

Hablar de TO lleva consigo la necesidad de hablar de objetos, de componentes, de patrones, de lenguajes, de *frameworks*, de arquitecturas, de persistencia, etc. Por otra parte, la TO hoy pasa por la integración de cada una de las técnicas y herramientas de desarrollo, tanto de análisis y diseño, como de programación. Cada una de estas palabras clave, así como la problemática de la integración de las mismas, constituye un área de interés que podría, sin duda alguna, dar lugar a una propuesta como esta. Por ello, no se pretende profundizar en ninguna de estas áreas sino, más bien, ofrecer una visión general de la TO en la actualidad así como haciendo una arriesgada previsión de futuro, constituyen las principales tendencias de la misma.

### 2.1.1. Estado actual y tendencias orientadas a objetos

En cuanto a tendencias, hay que separar claramente dos aspectos: tendencias en cuanto a *métodos para el desarrollo y gestión*, que ayudan en la concepción, diseño, codificación y pruebas así como la gestión de un sistema orientado a objetos (OO); y tendencias en cuanto a la *tecnología o infraestructura de desarrollo y explotación* propiamente dicha.

---

<sup>4</sup> Aplicaciones CAD/CAM, multimedia, aplicaciones distribuidas, etc.

### 2.1.2. Procesos de desarrollo y gestión

Como consecuencia del nacimiento de los primeros lenguajes de programación OO así como de la complejidad creciente de las aplicaciones, comienzan a surgir, en la década de los años 1990 tres autores ampliamente conocidos: Booch, Rumbaugh y Jacobson<sup>5</sup>. Estos tres autores se unieron para definir el Lenguaje de Modelado Unificado (UML) [Booch et al., 1999], como un conjunto de técnicas y un lenguaje de definición para el modelado de sistemas OO. UML versión 1.1 fue aprobada por el OMG como notación estándar de modelado en septiembre de 1997. En 1998 se aprobó la versión 1.2 y en 1999 la 1.3. Al año 1999 se esperaba UML versión 1.4, que incluiría revisiones menores al estándar [Kobrin, 1999] y UML 2.0 [OMG, 1999].

Sin embargo, y pese a toda la funcionalidad que parece poseer UML y de la clara aceptación que ha tenido en el mercado, existen otras propuestas relevantes como OPEN [Henderson-Sellers, 1996], Catalysis [D'Souza y Cameron, 1999] para el desarrollo basado en componentes y frameworks con UML u OOHDM [Schwabe y Rossi, 1995], para el desarrollo OO en la Web.

### 2.1.3. Tecnología orientada a objetos

Se llama tecnología orientada a objetos a la infraestructura que permite desarrollar sistemas orientados a objetos que son: servidores de transacciones, *middleware*, Sistemas de Gestión de Bases de Datos (SGBD), Lenguajes de Programación (LP), etc.

En cuanto a *middleware*, CORBA es probablemente la solución a corto plazo para arquitecturas distribuidas, aunque DCOM también mantiene un puesto importante. Las ventajas de los monitores transaccionales (TUXEDO) y los ORB se están fundiendo en los CTM (*Component Transaction Monitors*) o servidores de aplicación (Net Dinamycs o MTS - *Microsoft Transaction Server*). Estos servidores de aplicación combinan la fluidez y facilidad de acceso de los sistemas de objetos distribuidos basados en ORB con un monitor transaccional.

---

<sup>5</sup> Con una propuesta más enfocada al análisis de sistemas de información con grandes volúmenes de datos.

Por otra parte, los Sistemas de Gestión de Bases de Objetos (SGBO), como ObjectStore o POET, no tienen la madurez suficiente para soportar aplicaciones con grandes volúmenes de datos que requieren tiempos rápidos de respuesta. Para este tipo de aplicaciones se utilizarán las nuevas versiones de los productos objeto-relacionales (Oracle 8i, Universal Server de Informix). Sin embargo, los SGBO pueden ofrecer buenas prestaciones para BD, de tamaño pequeño o medio, que estén orientadas a Internet.

## 2.2. TECNOLOGIA WEB

La idea fundamental es que los navegadores o *browsers*, presentan documentos escritos en HTML que se han obtenido de un servidor Web. Estos documentos HTML habitualmente presentan información de forma estática, sin más posibilidad de interacción con ellos.

El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías Web pasando desde las primeras páginas escritas en HTML almacenadas en un archivo en el servidor Web hasta aquellas que se generan al vuelo como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Además, el modo de generar páginas dinámicas ha evolucionado, desde la utilización de la Interfaz Gateway Común (CGI, *Common Gateway Interface*) hasta las aplicaciones que se ejecutan del lado del servidor. Todas estas tecnologías se encuadran dentro de aquellas conocidas como *Server Side*, ya que se ejecutan en el servidor Web.

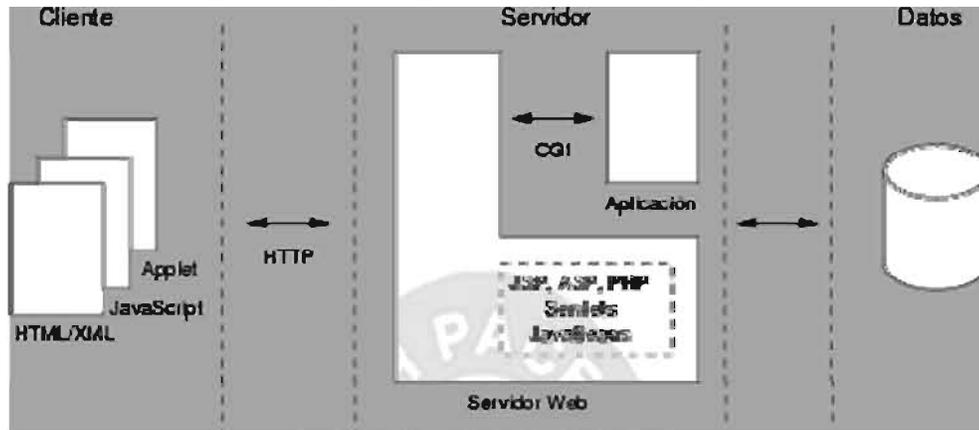
Otro aspecto que completa el panorama son las inclusiones del lado del cliente, *Client Side*, que se refieren a las posibilidades de que las páginas lleven incrustado código que se ejecuta en el cliente, como por ejemplo JavaScript [Vegas, 2002].

El esquema general de la situación se puede ver en la figura 2.1, donde se muestran cada tipo de tecnología involucrada en la generación e interacción de documentos Web.

## 2.3. ESPECIFICACION DE REQUISITOS DEL SOFTWARE

De acuerdo con la norma IEEE publicada el año 1998, específicamente el estándar IEEE-STD-830-1998: Especificaciones de los Requisitos del Software (SRS), cuenta con el siguiente cuerpo metodológico.

Fig. 2.1. Esquema general de las tecnologías Web.



Fuente: [Vegas, 2002]

### 2.3.1. Definiciones

En general las definiciones de los términos utilizados en estas especificaciones están de acuerdo a las definiciones proporcionadas en el estándar IEEE-STD-610.12-1990.

- a) **Contrato:** Un documento es legalmente obligatorio y en el estarán de acuerdo a las partes del cliente y proveedor. Esto incluye los requisitos técnicos y requerimientos de la organización, costo y tiempo para un producto. Un contrato también puede contener la información informal pero útil como los compromisos o expectativas de las partes involucradas.
- b) **Cliente:** Las personas que pagan por el producto, normalmente (pero no necesariamente) definen los requisitos. En la práctica el cliente y el proveedor pueden ser miembros de la misma organización.
- c) **Proveedor:** Las personas que producen un producto para un cliente.
- d) **Usuario:** Las personas que operan o actúan recíprocamente directamente con el producto. El usuario y el cliente no son a menudo las mismas personas.

### 2.3.2. Consideraciones para producir un buen requerimiento

Estas cláusulas proporcionan información a fondo que deben ser consideradas al momento de producir un buen requerimiento, a través de la especificación correcta del SRS. Esto incluye lo siguiente:

- a) Naturaleza del SRS;
- b) Características de un buen SRS ;

- c) Preparación de las reuniones para el SRS;
- d) Evolución de SRS;
- e) Prototipos;
- f) Generación del diseño en el SRS;
- g) Generación de los requisitos del proyecto en el SRS.

### **2.3.2.1. Naturaleza de la especificación**

El SRS hace referencia a las especificaciones para un producto software en particular, programa, o juego de programas que realizan ciertas funciones en un ambiente específico. El SRS puede escribirse por uno o más representantes del proveedor, uno o más representantes del cliente, o por ambos. Los problemas básicos que se presentan al escribir un SRS están dirigidos a lo siguiente:

- a) La funcionalidad. ¿Cuáles son las funciones supuestas que realizará el software?
- b) Las interfaces externas. ¿Cómo el software actúa recíprocamente con las personas, el hardware de los sistemas, otro hardware, y otro software?
- c) La actuación. ¿Cuál es la velocidad, la disponibilidad, tiempo de respuesta, tiempo de recuperación de varias funciones del software, etc.?
- d) Los atributos. ¿Qué portabilidad tiene, exactitud, mantenimiento, seguridad, las consideraciones, etc.?
- e) Las restricciones del diseño que impusieron en una aplicación. ¿Hay algún requerimiento estándar, idioma de aplicación, las políticas para la integridad de la base de datos, los límites de los recursos, en que ambiente opera, etc.?

### **2.3.2.2. Características de una buena especificación**

Una buena especificación de requerimientos del software debe ser:

- a) Correcto. Un SRS es correcto si y solamente si, cada requisito declarado se encuentra en el software.
- b) Inequívoco. Un SRS es inequívoco si y sólo si, cada requisito declarado tiene sólo una interpretación.

- c) Completo. Un SRS está completo si y sólo si, incluye los elementos siguientes: requisitos, definición de las respuestas, etiquetas llenas y referencias a todas las figuras, tablas y diagramas.
- d) Consistente. La consistencia se refiere a la consistencia interior. Si un SRS no está de acuerdo con algún documento del nivel superior, como una especificación de requisitos de sistema, entonces no es correcto.
- e) Consistencia interior. Un SRS es internamente consistente si y sólo si, ningún subconjunto de requisitos individuales genera conflictos en el requerimiento.
- f) Importancia y estabilidad. Un SRS debe delinear la importancia y estabilidad si cada requisito tiene un identificador para indicar la importancia o estabilidad de ese requisito en particular. Típicamente, todos los requisitos que relacionan a un producto software no son igualmente importantes. Algunos requisitos pueden ser esenciales, sobre todo para las aplicaciones de vida crítica, mientras otros pueden ser solo deseables.
- g) Comprobable. Un SRS es comprobable si y sólo si, cada requisito declarado es capaz de comprobarse.
- h) Modificable. Un SRS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requisitos de manera fácil, completa y de forma consistente mientras conserva la estructura y estilo.
- i) Identificable. Un SRS es identificable si el origen de cada uno de sus requisitos está claro y si facilita las referencias de cada requisito en el desarrollo futuro o documentación del mismo.

### **2.3.2.3. Preparación de las reuniones para la especificación**

El proceso de desarrollo de software debe empezar con el proveedor y con el acuerdo del cliente, en lo que el sistema software debe hacer. Este acuerdo, en la forma de un SRS, debe prepararse juntamente. Esto es importante porque ni el cliente ni el proveedor son calificables para escribir exclusivamente un buen SRS.

- a) Los clientes normalmente no entienden de manera adecuada el diseño del software y el proceso de desarrollo, como para escribir un SRS utilizable.
- b) Los proveedores normalmente no entienden bien el problema de los clientes y el campo de acción del mismo, como para especificar los requisitos de un sistema satisfactorio.

Por consiguiente, el cliente y el proveedor deben trabajar juntos para producir un SRS bien escrito y completamente entendible. Una situación especial existe cuando el sistema y su software se definen concurrentemente. Entonces la funcionalidad, interfaces, desarrollo y otros atributos, como las restricciones del software, no son los predefinidos, sino se deben definir de manera conjunta y estar sujetos a la negociación y al cambio.

#### **2.3.2.4. Evolución de la especificación**

El SRS necesita evolucionar así como cambiar para el desarrollo de las actualizaciones del producto software. Puede ser imposible especificar a detalle en el momento que el proyecto se inicia (por ejemplo, puede ser imposible definir toda la estructura de la pantalla para un programa interactivo durante la fase de requisitos). Los cambios adicionales pueden suceder según como se vayan descubriendo las deficiencias, las limitaciones e inexactitudes en el SRS.

Dos consideraciones en este proceso son las siguientes:

- a) Debe especificarse los requisitos completamente como se es conocido en el momento de la interacción con el cliente, aún cuando las revisiones evolutivas pueden preverse como inevitables. Debe anotarse el hecho que ellos están incompletos.
- b) Un proceso de cambio formal debe comenzar para identificar el control, dejar huella e informes de los cambios proyectados.

Los cambios aprobados en los requisitos deben incorporarse en el SRS de tal manera que:

- a) Proporcione un lineamiento de la auditoria exacta y completa de los cambios.
- b) Efectúe el permiso de la revisión actual y el reemplazo de los cambios en el SRS.

#### **2.3.2.5. Prototipos.**

Los prototipos estáticos frecuentemente se utilizan durante la fase de los requisitos de un proyecto. Muchas herramientas existen para generar un prototipo que exhiba algunas características de un sistema, pueda ser creado muy rápida y fácilmente. Los prototipos son útiles por las siguientes razones:

- a) El cliente puede ver el prototipo y reaccionar a él, que leer el SRS y reaccionar a él. Así, el prototipo proporciona una visión rápida del sistema.
- b) El prototipo despliega aspectos de anticiparse a la conducta de los sistemas. Así, no sólo produce las respuestas sino también las nuevas preguntas. Esto ayuda a ver el alcance en el SRS.
- c) Un SRS basado en un prototipo tiende a sufrir menos cambios durante el desarrollo, así se acorta el tiempo de desarrollo.

Un prototipo debe usarse como una manera de obtener los requisitos del software. Pueden extraerse algunas características como pantallas o formatos de reportes directamente del prototipo. Otros requisitos pueden ser inferidos ejecutando los experimentos con el prototipo.

#### **2.3.2.6. Generación de diseño en el SRS**

Un requisito especifica una función externa visible o un atributo de un sistema. Un diseño describe un subcomponente particular de un sistema o sus interfaces con otros subcomponentes. El diseñador del SRS debe distinguir claramente entre identificar las restricciones del diseño requeridos y proyectar un plan específico. La nota es que cada requisito en el SRS limita las alternativas del plan. Esto no significa, sin embargo, que cada requisito es el plan propiamente dicho.

El SRS debe especificar qué funciones serán realizadas, con qué datos, para producir qué resultados, en qué situación y para quien. El SRS se debe enfocar en los servicios a ser realizados. El SRS normalmente no debe especificar los puntos del plan como los siguientes:

- a) Partir el software en módulos;
- b) Asignar las funciones a los módulos;
- c) Describir el flujo de información o los controles entre los módulos;
- d) Escoger las estructuras de datos.

Los ejemplos de restricciones del diseño válidos son requisitos físicos, requisitos del desarrollo, normas de desarrollo del software y software de calidad según los estándares. Por consiguiente, los requisitos deben declararse desde un punto de vista completamente externo. Al utilizar modelos para ilustrar los requisitos, se debe recordar que el modelo sólo indica la conducta externa y no especifica un plan de desarrollo.

### 2.3.2.7. Requisitos del proyecto

El SRS debe dirigir el desarrollo del producto software. Los requisitos del proyecto representan una comprensión entre el cliente y el proveedor sobre materias contractuales que pertenecen a la producción del software y deben ser incluidos en el SRS. Éstos normalmente incluyen los puntos como:

- a) Costos;
- b) Los tiempos de la entrega;
- c) Información acerca de los procedimientos;
- d) Los métodos de desarrollo de Software;
- e) La convicción de calidad;
- f) La aprobación y criterio de la comprobación;
- g) Los procedimientos de aceptación.

Se especifican los requisitos del proyecto en otros documentos, generalmente en un plan de desarrollo de software, un software de calidad o una declaración de trabajo [IEEE, 1998].

## 2.4. METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A OBJETOS

El modelo Diseño de Hipertexto Orientado a Objetos (OOHDM, *Object Oriented Hypermedia Design Methodology*), para diseño de aplicaciones hipertexto y para la Web, fue diseñado por D. Schwabe, G. Rossi, y S.D.J. Barbosa y es una extensión de la metodología de diseño de hipertextos (HDM<sup>6</sup>) con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipertexto como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios Web [Schwabe y Rossi, 1997].

---

<sup>6</sup> El HDM o Hypertext Design Model, fue creado por Franca Garzotto, Paolo Paolini y Daniel Schwabe en 1991, dentro del marco del proyecto HYTEA de la Comunidad Europea. El objetivo era crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto. El modelo HDM divide el proceso de diseño de una aplicación de hipertexto en 2 partes: el *authoring-in-the-large*, que se refiere a la especificación y diseño de los aspectos globales y estructurales de la aplicación, y el *authoring-in-the-small*, que se refiere al desarrollo del contenido de los nodos.

Al igual que RMM<sup>7</sup>, este método se inspira en el modelo HDM, pero lo que le distingue claramente del primero es el proceso de creación orientado a objetos. OOHDM propone el desarrollo de aplicaciones hipertexto mediante un proceso de 4 etapas:

- a) diseño conceptual
- b) diseño navegacional
- c) diseño de interfaces abstractas
- d) implementación

Cada etapa de la creación define un esquema objeto específico en el que se introducen nuevos elementos o clases. Según Silva y Mercerat (1999), en la primera etapa se construye un esquema conceptual representado por los objetos del dominio o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos semántico estructurado (como el modelo entidad relación). El modelo OOHDM se propone como un esquema conceptual basado en clases, relaciones y subsistemas.

En la segunda etapa, el diseñador define clases navegacionales tales como nodos, enlaces y estructuras de acceso (índices y visitas guiadas) obtenidas del esquema conceptual. Los enlaces derivan de las relaciones y los nodos representan ventanas lógicas (*views*) sobre las clases conceptuales. A continuación, el diseñador describe la estructura navegacional en términos de contextos navegacionales. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos y otros contextos navegacionales (contextos anidados) que pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros. Los nodos se enriquecen con un conjunto de clases especiales que permiten presentar atributos así como métodos o comportamientos cuando se navega en un contexto particular. Durante esta etapa, es posible adaptar los objetos navegacionales para cada contexto.

OOHDM no propone un modelo enriquecido para el dominio de la aplicación, por lo que deja libre al diseñador para elegir el modelo de especificación del dominio. Sin embargo, el modelo hipertexto está definido en dos niveles de abstracción: las clases navegacionales y los contextos navegacionales. En el momento de la especificación de las clases navegacionales es cuando el diseñador define las correspondencias. Se pueden precisar los

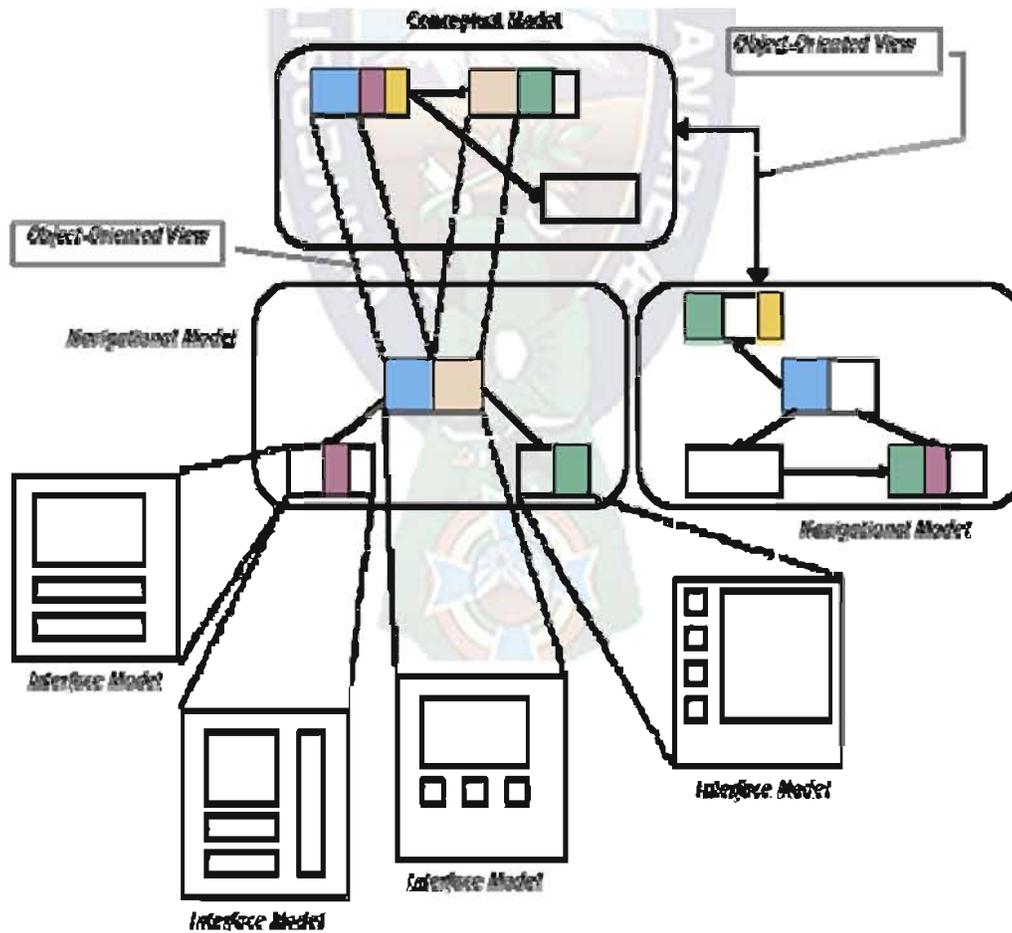
---

<sup>7</sup> La RMM o Relationship Management Methodology se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipertexto. Los elementos principales de este método son el modelo E-R (Entidad-Relación) y el modelo RMDM (Relationship Management Data Model) basado en el modelo HDM. La metodología fue creada por Isakowitz, Stohr y Balasubramanian. Según sus autores, está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos.

nodos obtenidos de las clases del modelo del dominio y los enlaces inducidos de las relaciones del modelo del dominio. Como el segundo nivel está consagrado a la especificación de la navegación, expresada exclusivamente sobre los objetos navegacionales (no sobre los elementos del modelo del dominio), constituye un mecanismo que permite enriquecer el modelo hipertexto.

La tercera etapa está dedicada a la especificación de la interfaz abstracta. En la misma se define la forma en la cual deben aparecer los contextos navegacionales. También se incluye aquí el modo en que dichos objetos de interfaz activarán la navegación y el resto de funcionalidades de la aplicación, esto es, se describirán los objetos de interfaz y se los asociará con objetos de navegación. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional.

Fig. 2.2. Relaciones entre objetos y esquema navegacional y conceptual



Fuente: Tomado de [Schwabe y Rossi, 1997]



**Tabla 2.3. Etapas de la Metodología OOHD**

<b>Etapas</b>	<b>Productos</b>	<b>Mecanismos</b>	<b>Intereses del Proyecto</b>
Diseño Conceptual	Clases, subsistemas, relaciones, perspectivas de atributos.	Clasificación, composición, generalización y especialización	Modelado de la semántica del dominio de la aplicación
Diseño Navegacional	Estructuras de acceso, contextos de navegación, transformaciones navegacionales	Mapeo entre objetos conceptuales y de navegación. Descripción de la estructura general de la aplicación	Se toma en cuenta el perfil de usuario y las tareas que se realizan
Diseño de Interfaz Abstracta	Objetos de interfaz abstracta, relaciones con eventos externos, transformaciones de interfaces	Mapeo entre objetos de navegación y objetos de interfaz	Modelado de objetos perceptibles, descripción de interfaces para objetos navegacionales
Implementación	Aplicación en ejecución	Aquellos proporcionados por el ambiente de trabajo	Desempeño y conclusión

Fuente: [Schwabe, Rossi y Barbosa, 1996]

#### 2.4.1.1. Diseño conceptual

En OOHD el modelo conceptual está construido en base a clases, relaciones y subsistemas. Las clases están construidas al igual que en un modelo orientado a objetos, además sus atributos son multitypos representando diferentes perspectivas de la misma entidad. El proceso de modelado del dominio puede ser básicamente el mismo en diversas aplicaciones, las decisiones tomadas en esta etapa se pueden referir a la estructura de una aplicación hipertexto.

Este modelo consiste en un conjunto de objetos y clases unidos entre si mediante las denominadas relaciones, ya que los objetos son instancias de las clases, las relaciones entre las clases son las relaciones entre los objetos abstraídos. Es importante mencionar que el objetivo de esta etapa es la construcción de un modelo de datos, y para lograr este objetivo se utiliza algunas metodologías adicionales, en el caso de la presente propuesta se usa el Lenguaje de Modelado Unificado, el cual es utilizado para la construcción de modelos orientados a objetos [Schwabe y Rossi, 1997].

### 2.4.1.2. Diseño navegacional

En OOHDM se propone la construcción de un modelo conceptual compartido, el cual está focalizado en los objetos y relaciones del dominio, del cual, diferentes visiones navegacionales serán derivadas tomando en cuenta los perfiles de los futuros usuarios del sistema. El modelo conceptual funciona como un repositorio de información del modelo, a partir del cual se construyen diferentes aplicaciones del dominio del problema, esto significa que diferentes comunidades de usuarios vean el mismo objeto con perspectivas diferentes.

Los contextos navegacionales ayudan en la organización de los objetos navegacionales formando espacios de navegación consistentes que disminuyen la posibilidad de que el usuario pueda perderse en el ciberespacio. Según Schwabe y Rossi (1997) los esquemas asociados al diseño navegacional son:

- a) Esquema de clases navegacionales. Uno de los productos de la actividad de diseño navegacional es el esquema navegacional; las clases navegacionales en esta actividad son definidas como especializaciones de un conjunto de clases básicas que definen la semántica de los objetos navegacionales. Los diseños navegacionales son derivados del diseño conceptual a través de mecanismos de definición de vistas, pero en algunas aplicaciones el diseño navegacional y el diseño conceptual son muy similares debido a la simplicidad del modelo o a la semejanza de las tareas desempeñadas por los distintos usuarios. El lenguaje de definición de vistas se basa en el lenguaje orientado a objetos, a su vez la definición de vistas se basa en el área de base de datos, y en particular las bases de datos orientadas a objetos.
- b) Esquema de contexto navegacional. Cualquier aplicación bien proyectada debe tomar en cuenta el modo como el usuario explora el espacio hipermedia, y evitar la información redundante, ayudando al usuario a escoger la forma de navegación a través de la aplicación. El concepto de contexto navegacional que se presenta esta basado sobre la premisa de que los elementos de información de un hiperdocumento son mas fácilmente entendidos cuando se presentan en un contexto y que el uso de los contextos minimiza el problema de desorientación común en aplicaciones hipermedia complejas. Un contexto navegacional está constituido por un conjunto de nodos (clases), instancias de clases (objetos), relaciones entre objetos, y otros contextos navegacionales. En OOHDM el esquema navegacional

muestra la relación entre los objetos navegacionales, mientras que los contextos navegacionales muestran la navegación en general.

- c) Clases navegacionales. Los nodos contienen información básica en las aplicaciones. La estructura de los nodos depende de la semántica, esto quiere decir, de los intereses particulares de los usuarios a los cuales está dirigida la aplicación. En OOHDm se define una clase de nodos como una cadena lógica entre una o más clases en el esquema conceptual. Los nodos son descritos como un grupo de atributos y un conjunto de métodos que implementan su comportamiento y son organizados en jerarquías: **de parte de** y **es un**. Los comportamientos predefinidos de la clase nodo implementan el modo como los nodos reaccionan cuando son activados durante la navegación, como reaccionan cuando un enlace es seleccionado, y la manera en la que interactúan con los objetos (instancias) y sus interfaces. Los nodos son definidos utilizando un lenguaje de definición orientado a objetos.

Los enlaces en OOHDm son tratados como gatillos navegacionales, esto se produce cuando un enlace es seleccionado y ocurre un evento de navegación. En OOHDm los enlaces se definen como: **área: Enlace (Pertenece)**

**Tabla 2.2. Definición de los nodos**

Nodo :	nombre [DE NombreClase:var NombreClase] [HEREDE DE ClaseSuper]
Atributo 1 :	tipo1 [SELECCIONAR nombre1, ..., nombren] [DE Clase:var Clase DONDE expresion]
Atributo 2 :	tipo2 [idem]
...	
Atributo n :	Tipon [idem]

Donde:

*nombre*: Nombre de la clase que se está creando.

*NombreClase*: Nombre de la clase conceptual a partir de la cual se realiza la correspondencia con el nodo.

*ClaseSuper*: nombre de la superclase.

*nombren*: Nombre de los elementos de la consulta.

*expresión*: Permite la definición de nodos cuyas instancias son una combinación de objetos definidos en el esquema conceptual.

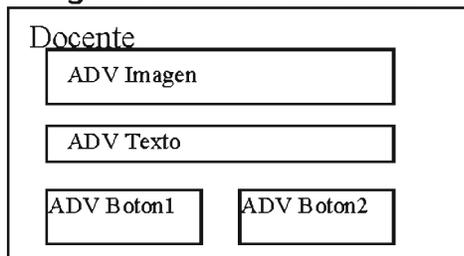
### 2.4.1.3. Diseño de interfaz abstracta

Según Schwabe y Rossi (1997), la construcción de una interfaz hipermedia es un aspecto crítico en una aplicación hipermedia. Para especificar el modelo abstracto de interfaces es necesario describir sus propiedades estáticas y dinámicas, así como sus relaciones con el modelo navegacional.

En OOHDM se utilizan los ADV's (Abstract Data Views) para especificar el modelo de interfaz abstracta. El modelo ADV fue creado originalmente para especificar clara y formalmente la separación entre la interfaz de usuario y los componentes del software, y ofrecer un método independiente de la implementación, generando altos grados de reusabilidad de componentes. Los ADV's son objetos que poseen un estado y una interfaz, los cuales pueden ser ejecutados o accedidos a través de mensajes de otros objetos o mediante eventos de entrada y salida, son abstractos en el sentido que representan la interfaz y el estado, no la implementación.

Los ADV's son usados generalmente para representar interfaces entre dos medios diferentes, por ejemplo, un usuario y una red, o como una interfaz entre dos o más objetos de datos abstractos (Abstract Data Objects (ADO's)), siendo que los ADV's no soportan eventos externos. En una aplicación que utiliza ADV's hay un conjunto de ADO's que administran la estructura de datos y el control de la aplicación, así como un conjunto de objetos de interfaz (instancias de ADV's) administrando aspectos de la interfaz de la aplicación, tales como entradas y salidas de los usuarios al sistema. Un ADV puede ser visto como un objeto de interfaz conteniendo un conjunto de atributos que definen sus propiedades de percepción y un conjunto de eventos, como aquellos generados por el usuario. Diferentes mecanismos de abstracción y composición son usados en ADV, pueden ser compuestos por agregación o composición de ADV's, permitiendo la construcción de interfaces de usuario con objetos perceptibles anidados.

**Fig. 2.4. Definición de un ADV**



**Fuente: [Larman, 1999]**

#### 2.4.1.4. Implementación

La implementación en OOHDM se basa en la implementación de los ADV's, es responsabilidad del programador o diseñador, construir todos los elementos del software de acuerdo al análisis y diseño realizados en las etapas anteriores [Schwabe y Rossi, 1997].

#### 2.4.2. Hipertexto WWW

La World Wide Web, comúnmente llamada Web o WWW, nace a principios de los años 1990, aunque sus orígenes se remontan a mucho tiempo antes, ya que es fruto de la confluencia de la teoría hipertextual y de las redes de computadoras. Fue creada por *Tim Berners-Lee* en el Centro Europeo de Física Nuclear (CERN) con el objetivo de servir como herramienta para la búsqueda y transmisión de información entre los científicos [W3C, 2002].

El hipertexto es la base funcional y estructural de la World Wide Web. Se podría decir que la Web es un hipertexto de **escala** planetaria puesto que cualquier usuario puede poner su página en la red y establecer **enlaces** a cualquiera de los documentos disponibles en ella. La Web es un sistema hipertextual preparado para recorrer diferentes páginas Web dispuestas en servidores accesibles desde cualquier computadora conectada a Internet y enlazadas unas con otras conformando una estructura similar a la de una telaraña, típicamente una red hipertextual asociativa, o mejor dicho, la red hipertextual por antonomasia. Las páginas Web se enlazan unas a otras dentro de cada hiperdocumento o sitio Web y pueden conectarse a otros hiperdocumentos o sitios Web llevando al usuario de un servidor a otro sin necesidad de teclear ninguna ruta [García y Tramullas, 1996].

En la Web, cada nodo es una página, y **cada palabra** remarcada representa la entrada de un enlace. La Web también permite relacionar documentos multimedia (imágenes, sonidos, vídeo, etc.) y recursos residentes en múltiples servidores mundiales y ofrece un nuevo y más extenso medio para estudiar las consecuencias del hipertexto convertido en hipermedia.

La Web se ha convertido en uno de los servicios principales de Internet. En pocos años, casi toda la información disponible en la red se ha volcado a la Web y se han ido abandonando otros métodos como Gopher, con su estructura jerárquica de acceso a la información, o las técnicas documentales que ofrecía Wais. La World Wide Web ha triunfado y millones de

documentos se encuentran accesibles mediante este sistema de almacenamiento y acceso a la información.

La World Wide Web constituye una gran red documental de estructura hipertextual. El éxito de este modo de presentar la información, se debe principalmente a la flexibilidad del protocolo HTTP y a las capacidades del lenguaje HTML. En la base de ambos, está la estructura hipertextual de los documentos creados, de donde protocolo y lenguaje, toman sus nombres: *Hypertext Transfer Protocol* e *Hypertext Markup Language*. Lo cierto es que en la Web hoy existen otros muchos lenguajes tanto para estructurar la información como para conceptualizarla y que a través de la Web no sólo se accede a información, sino que la Web se ha convertido en la interfaz más utilizada hoy en día para la prestación de muchos de los servicios de Internet debido, sobre todo, además de su capacidad hipertextual, a sus capacidades hipermedia, dinámicas e interactivas [Hughes, 2000].

Las páginas Web precisan de una conexión a Internet y de un programa de navegación que deberá estar disponible en la computadora del usuario o cliente Web, ya que el funcionamiento de la Web se basa en el modelo cliente-servidor. El programa de navegación se denomina navegador, *browser*, visualizador, explorador, visor, lector u hojeador<sup>8</sup>. El primer navegador fue Mosaic desarrollado por el NCSA (*The National Center for Supercomputing Applications*), y hoy los navegadores más populares son Internet Explorer, Netscape Navigator, Opera, Firefox, Mozilla, etc; aunque existen otros muchos.

Hay que tener en cuenta que la Web no es Internet, sino una pequeña parte de ella. Por lo tanto, es preciso analizar los distintos aspectos relacionados con la red Internet, como sus aspectos tecnológicos (protocolos, direcciones IP, URL, dominios, etc.) y los distintos servicios que presta la red de redes. También merece una mención aparte la puesta en práctica de nuevos desarrollos en el campo de los lenguajes hipertextuales que permiten dotar de una semántica a los documentos contenidos en la Web - lo que se ha venido en denominar la Web Semántica-, y el desarrollo de los nuevos Servicios Web, así como el auge y popularidad de la llamada Web 2.0 basada en redes sociales y cooperativas [García y Tramullas, 1996].

---

<sup>8</sup> La traducción de la palabra *browser* no está claramente definida, aunque en los últimos años parece haber triunfado finalmente la palabra navegador.

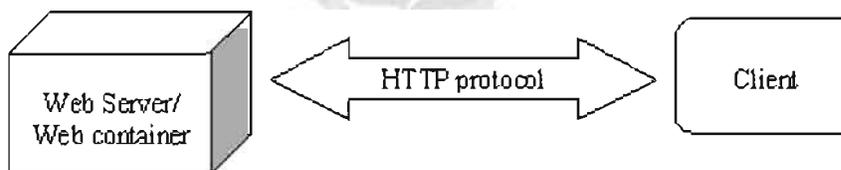
## 2.5. APLICACIONES WEB CON PHP

Los primeros sitios Web no eran sino colecciones de páginas Web enlazadas entre ellas por el Lenguaje HTML. Hoy en día, los sitios Web incluyen multimedia, aplicaciones de comercio electrónico, y otro tipo de sofisticadas aplicaciones basadas en Web como voz IP y banca online. Además de los avances en contenido, se han producido avances en las tecnologías de servidores Web, como el desarrollo de servidores de aplicaciones, y tecnologías de creación dinámica de contenido, como las páginas PHP y las páginas ASP.

### 2.5.1. Interacción en Internet

Según Pursnani (2004), cuando visitas un restaurante, el camarero coge tu pedido y se lo pasa al personal de la cocina. Cuando el pedido se ha completado, el camarero te trae la comida. Las actividades que se desarrollan en la cocina, son invisibles. La cocina, al igual que la Web, opera según un modelo cliente/servidor. Al escribir la dirección de un sitio Web o una URL en un navegador, o cliente, se inicia una petición al servidor que alberga el sitio. Cualquier acción que realice el servidor es transparente para los visitantes del sitio. En una arquitectura cliente/servidor típica de 2-capas, el contenido del sitio Web<sup>9</sup> como las páginas HTML, los PHP, los servlets, las JSP, las ASP, software de correo, los scripts CGI, etc. se almacenan en el servidor Web. Los sistemas de 2-capas más viejos, como los de la figura 2.5, estaban hechos de un servidor Web o contenedor, y el navegador.

**Fig. 2.5. Arquitectura 2-capas (Cliente/Servidor)**

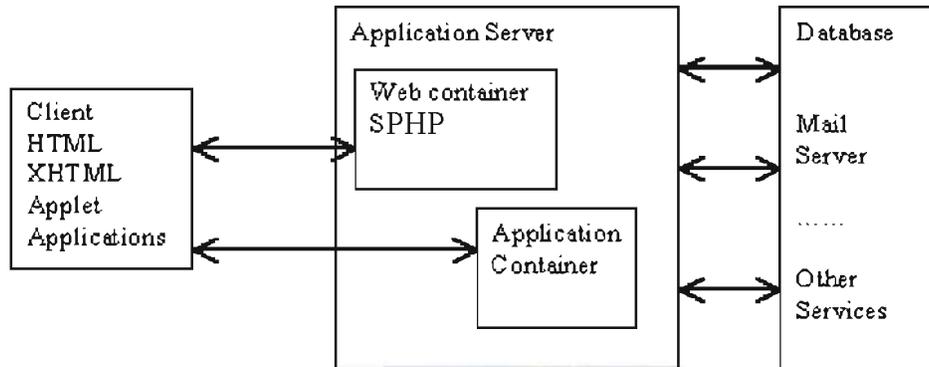


**Fuente: [Pursnani, 2004]**

El servidor de aplicaciones es una extensión del modelo cliente/servidor 2-capas a una aproximación de N-capas. El servidor de aplicaciones tiene un servidor o contenedor Web integrado y un contenedor de aplicaciones. La figura 2.6 demuestra la aproximación de N-capas.

<sup>9</sup> Es decir la capa de presentación del sitio Web.

**Fig. 2. 6. Arquitectura de N-capas**



**Fuente: [Pursnani, 2004]**

La aproximación de N-capas es ideal para la generación de contenido dinámico.

### **2.5.2. Servidor APACHE**

El servidor HTTP Apache fue construido por un grupo de webmasters conectados por Internet que se dieron cuenta de que era más inteligente unir sus esfuerzos para mejorar un código base que desarrollar un gran número de esfuerzos de desarrollo paralelos. Haciendo esto fueron capaces de capturar la mayoría de las ventajas del desarrollo propio y el poderoso efecto de depuración de la revisión entre pares masivamente paralelo.

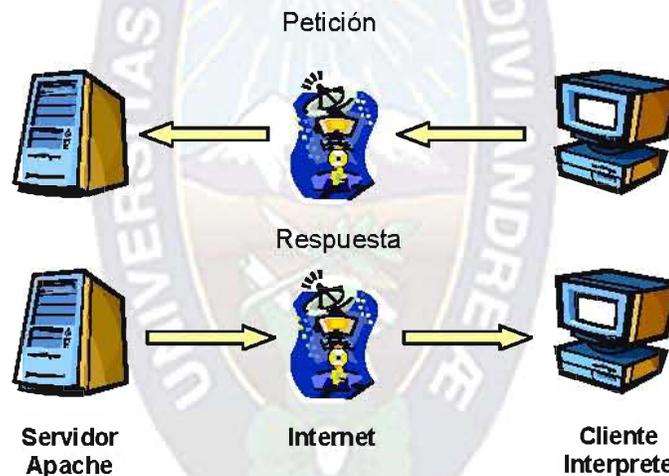
Este software es parte de un proyecto que se encuentra en desarrollo y mantenimiento constante cuya finalidad es la creación de un servidor HTTP de código abierto en cualquier computadora y en cualquier sistema operativo moderno, como UNIX y Windows. Según el sitio Web en Internet <http://httpd.apache.org>, la meta de dicho proyecto es proporcionar un servidor seguro y eficiente que pueda ofrecer los servicios HTTP en sincronía con los estándares actuales de HTTP.

La última versión de Apache HTTP Server está disponible en la página oficial <http://httpd.apache.org>. En este sitio Web no se menciona las características relevantes de la seguridad empotrada en este software. Sin embargo es notable, desde la configuración en el código abierto, que este software impida el acceso a otros directorios que se encuentren en niveles inferiores al especificado en la configuración.

En el archivo de configuración debe especificarse que esta ruta de acceso al disco duro, será la dirección raíz de la página Web. De modo que en este directorio se ha de colocar el archivo principal, que por defecto tiene el nombre: `index.html`. En el cual se tiene el contenido principal o la página principal del sitio Web. Basta con esta configuración para que el visitante de la página, no pueda acceder a páginas inferiores a `htdocs`.

La seguridad constituye una de las más grandes ventajas cuando se decide la utilización de Apache. Aunque dentro del mismo archivo de configuración para Apache, existen otras configuraciones, desde el uso de puertos para entregar respuestas a los clientes, hasta el uso de cookies y la utilización de versiones diferentes de PHP y MySQL.

**Fig. 2.7. Servidor Apache y peticiones**



Fuente: modificado de [Apache, 2005]

Apache se encarga de responder a las peticiones de los visitantes de la página por medio de un puerto. El visitante manda un paquete, conformado por un encabezado y un cuerpo del mensaje, por Internet desde su computadora de cliente con los datos necesarios del servidor, en este caso únicamente la dirección de la página<sup>10</sup>, en cuanto el servidor Apache sepa que le fue efectuada una petición, responderá con otro paquete con su encabezado y su cuerpo, donde en el cuerpo tiene la copia del archivo `index.html`. De este modo la

<sup>10</sup> La dirección viene en el cuerpo del mensaje, el encabezado tiene la información del cliente.

computadora cliente, por medio de un visualizador o interprete<sup>11</sup> traducirá el código HTML, de modo que el cliente lo vea como texto, animaciones e imágenes.

El servidor HTTP Apache es un servidor Web poderoso y flexible. Implementa los últimos protocolos incluyendo el HTTP/1.1. Es altamente configurable y extensible con módulos de terceros. Puede ser personalizable escribiendo módulos a través de una API. Provee acceso completo al código fuente, y utiliza una licencia no restrictiva. Corre bajo Windows NT/9x, Netware 5.x, OS/2, y en la mayoría de las versiones de Unix, así como en varios otros sistemas operativos. Todas estas características lo convierten en el servidor HTTP más utilizado en Internet, con un 59% de presencia entre los servidores Web en el ámbito mundial<sup>12</sup>, en febrero del 2001. Este porcentaje supera al representado por la suma de los todos los sitios que usan otros servidores.

### **2.5.3. PHP**

PHP es uno de los lenguajes de lado del servidor más extendidos en la Web. Nacido en 1994, se trata de un lenguaje para el desarrollo de aplicaciones Web relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters, debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP es un lenguaje de programación de código abierto que permite la generación de contenidos dinámicos en páginas Web en un servidor. Su nombre oficial es PHP: HiperText Preprocessor. Entre sus principales características se pueden destacar su potencia, alto rendimiento y facilidad de aprendizaje. Es una herramienta eficaz de desarrollo para los programadores Web, debido a que proporciona elementos que permiten generar de manera rápida y sencilla sitios Web dinámicos.

PHP fue originalmente creado por Rasmus Lerdorf como un conjunto de utilidades para añadir contenido dinámico en las páginas Web. Ganó rápidamente popularidad y fue posteriormente rediseñado de manera completa por Zeev Suraski y Andi Gutmans y rebautizado como PHP 3.0, actualmente existen otras versiones entre las cuales se destaca la versión PHP 4.0.

---

<sup>11</sup> Es decir: Internet Explorer, Netscape, etc.

<sup>12</sup> Información obtenida en <http://www.netcraft.com/survey/>

PHP es un lenguaje de programación que contiene conceptos de C, Perl y Java, también su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje. La cuestión es que aún no existen muchos textos de apoyo para aprender a programar en este lenguaje. Sin embargo existen muchos sitios Web donde se pueden conseguir tutoriales, explicaciones y ejemplos para aprender este lenguaje.

PHP es un lenguaje para la creación de sitios Web del cual se pueden destacar las siguientes características:

- a) Potente y robusto. Con un lenguaje de programación embebido<sup>13</sup> en documentos HTML.
- b) Conectado. Dispone de bibliotecas de conexión con la gran mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información en el servidor.
- c) Comunicación variada. Proporciona soporte a múltiples protocolos de comunicación en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.)
- d) Código fuente abierto. El código del interprete se encuentra disponible y accesible, permitiendo posibles mejoras o sugerencias acerca de su desarrollo (PHP fue escrito en lenguaje C)
- e) Gratuito. No es necesario realizar ninguna aportación económica para programar en este lenguaje.
- f) Portátil y multiplataforma. Existen versiones del interprete para múltiples plataformas (Windows, Unix, Linux) en el servidor. Esto permite que las aplicaciones puedan ser portadas de una plataforma a otra sin necesidad de modificar una sola línea de código.
- g) Eficiente. PHP consume muy pocos recursos en el servidor, por lo que con un equipo relativamente sencillo es posible desarrollar aplicaciones interesantes.
- h) Alta velocidad de desarrollo. PHP permite desarrollar rápidamente sitios Web dinámicos. Proporciona gran cantidad de bibliotecas muy útiles y bien documentadas que ahorran mucho trabajo a los programadores.

El intérprete de PHP reside y se ejecuta en la computadora en la que está instalado el servidor Web. De esta manera, el cliente lo único que recibe es el resultado generado (páginas HTML, imágenes en formato .gif o .jpg etc.) y no le afecta si previamente fue

---

<sup>13</sup> Se entiende como el proceso de encajar una cosa dentro de otra.

producido por un programa CGI escrito en lenguaje C o por un script escrito en Perl o por una página con código PHP o ASP.

A diferencia de JavaScript, o de un applet Java, PHP es una tecnología que solamente se ejecuta del lado del servidor. Es importante saber dónde y para qué se puede aplicar una tecnología, y más importante aún, donde no se puede aplicar o para qué no se puede aplicar. PHP ha sido diseñado para generar de manera dinámica contenidos y uno de sus puntos fuertes es su gran capacidad para interactuar con una multitud de gestores de bases de datos. Además, también puede producir código de manera dinámica<sup>14</sup> que se envía por la página HTML y se ejecuta en el lado del cliente. En consecuencia para desarrollar aplicaciones Web empleando PHP es necesario disponer de un servidor Web y del intérprete de PHP instalado.

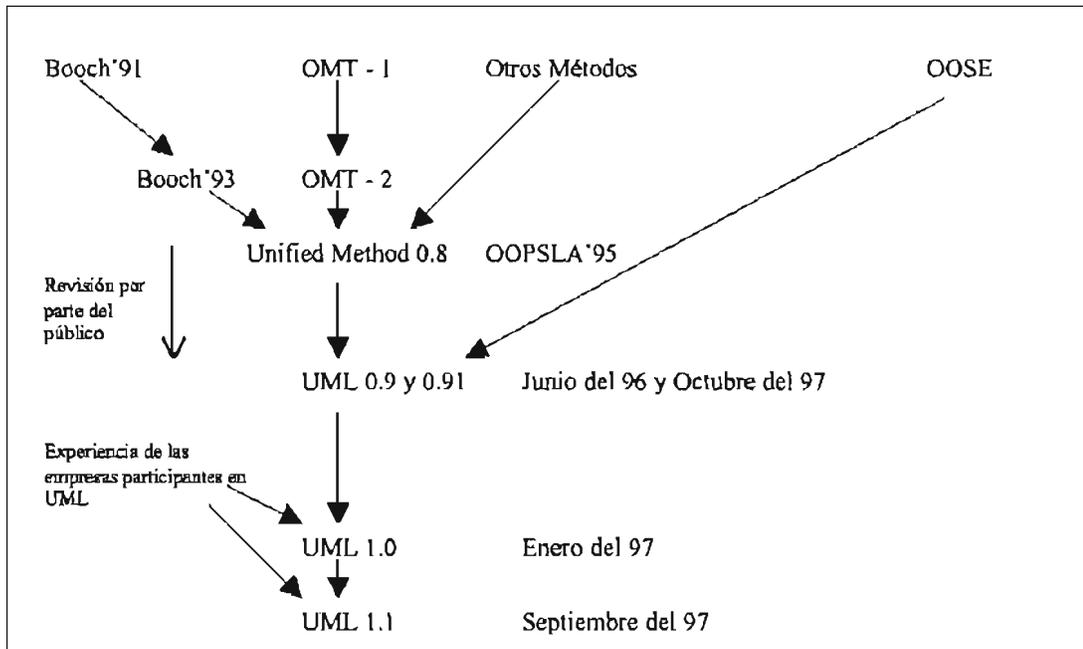
## 2.6. LENGUAJE DE MODELADO UNIFICADO

UML (*Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido 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. Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT y OOSE. UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

---

<sup>14</sup> Incluyendo por ejemplo JavaScript.

Fig. 2.8. Historia de UML



Fuente: [Ferré, 2003]

El objetivo principal cuando se empezó a gestar UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común. En la figura 2.8. se puede ver cuál ha sido la evolución de UML hasta la creación de UML 1.1.

Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación. En este curso se sigue el proceso propuesto por Craig Larman (1999) que se ajusta a un ciclo de vida evolutivo e incremental dirigido por casos de uso.

### 2.6.1. Modelos

Según Ferré (2003), un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico muestran la misma figura vista desde distintos ángulos, cada modelo permite establecer un aspecto distinto del sistema.

Los modelos de UML con los que normalmente se trabaja son los siguientes:

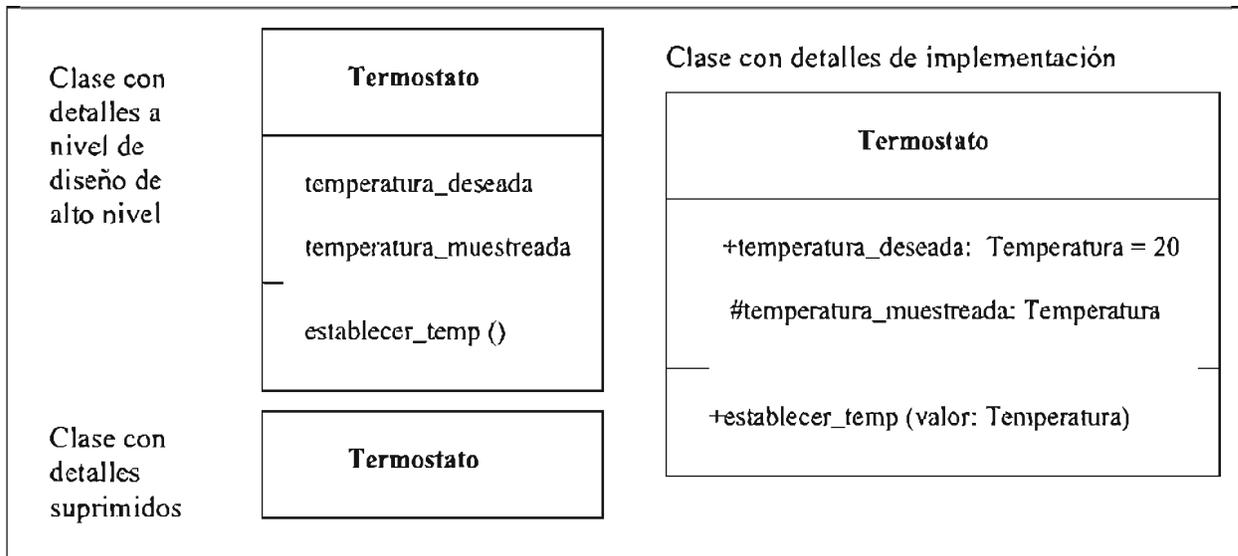
- a) Diagrama de Estructura Estática.
- b) Diagrama de Casos de Uso.
- c) Diagrama de Secuencia.
- d) Diagrama de Colaboración.
- e) Diagrama de Estados.
- f) Diagrama de Paquetes.

#### 2.6.1.1. Diagramas de Estructura Estática

Con el nombre de Diagramas de Estructura Estática se engloba tanto al Modelo Conceptual de la fase de Diseño de Alto Nivel como al Diagrama de Clases de Diseño. Ambos son distintos conceptualmente, mientras el primero modela elementos del dominio el segundo presenta los elementos de la solución software. Sin embargo, ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

- a) **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 (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase. En la figura 2.9. se ve cómo una misma clase puede representarse a distinto nivel de detalle según interese, y según la fase en la que se esté.

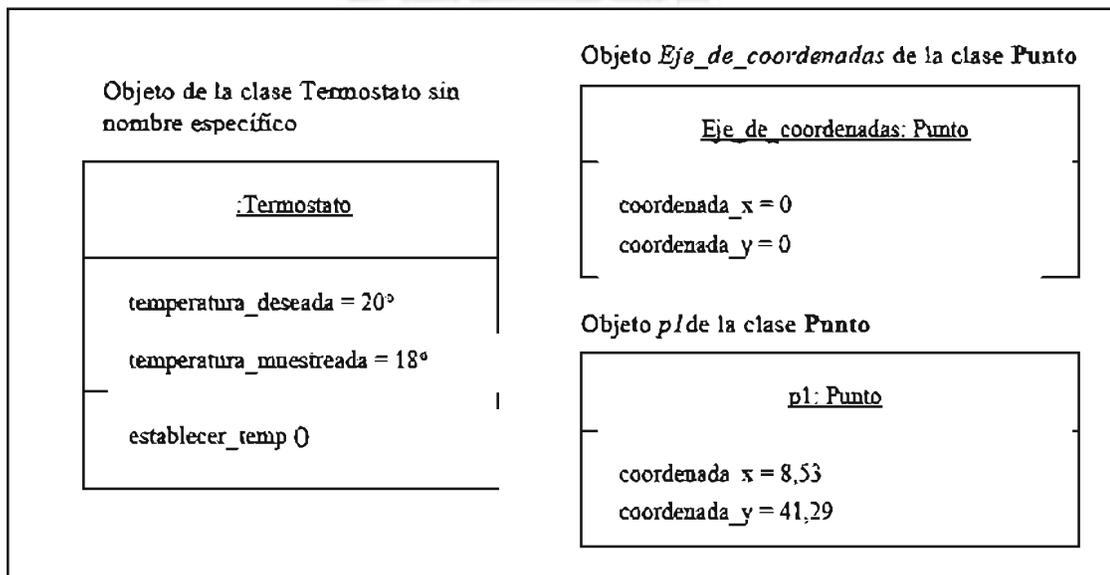
Fig. 2.9. Notación para clases a distintos niveles de detalle.



Fuente: [Ferré, 2003]

- b) **Objetos.** Un objeto se representa de la misma forma que una clase. En el compartimiento superior aparece el nombre del objeto junto con el nombre de la clase subrayado, 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.

Fig. 2.10. Ejemplo de objetos



Fuente: [Ferré, 2003]

c) **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 2.11. se puede leer la asociación como “Un objeto de la clase Perro es mascota de un objeto de la clase Persona”. 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.

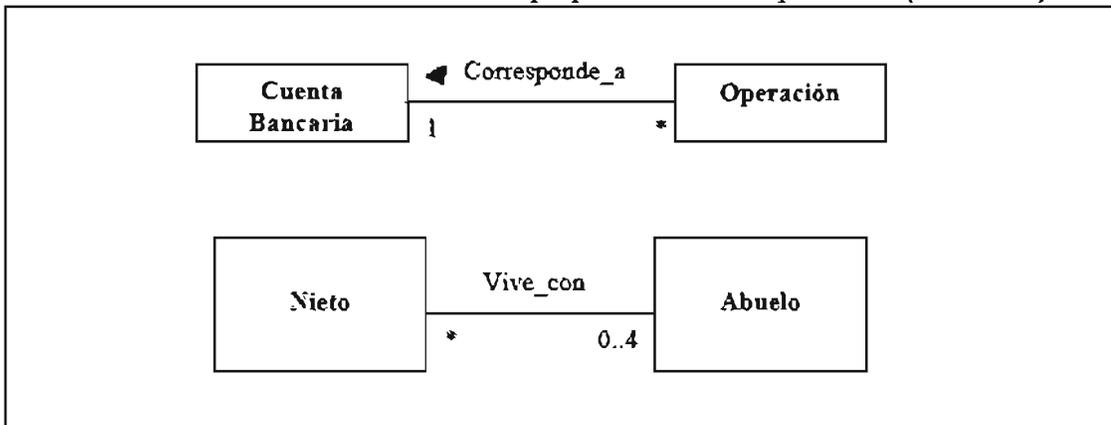
**Fig. 2.11. Ejemplo de asociación con nombre y dirección.**



**Fuente: [Ferré, 2003]**

- **Multiplicidad** La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase. Puede expresarse de las siguientes formas: (a) Con un número fijo: 1. (b) Con un intervalo de valores: 2..5. (c) Con un rango en el cual uno de los extremos es un asterisco. Significa que es un intervalo abierto. Por ejemplo, 2..\* significa 2 o más. (d) Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7, 15..\*. (e) Con un asterisco: \*. En este caso indica que puede tomar cualquier valor (cero o más).

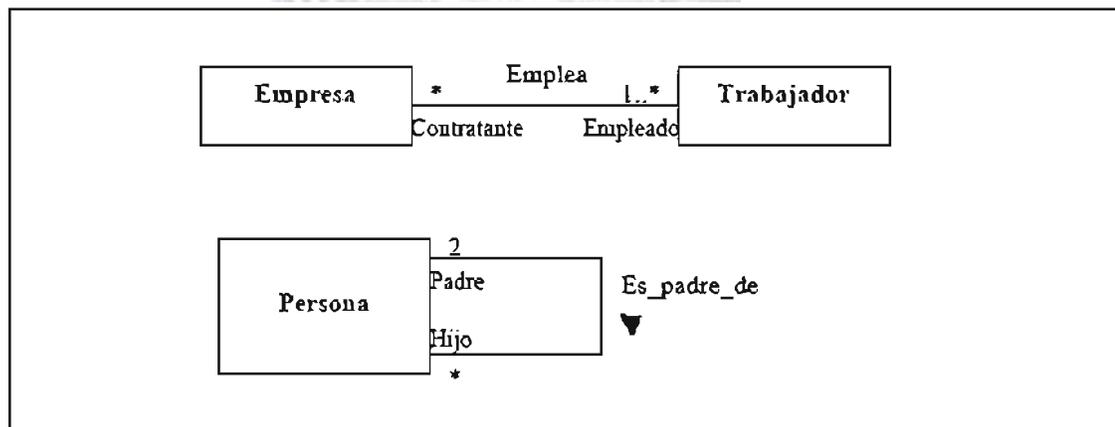
Fig. 2.12. Ejemplos de multiplicidad en asociaciones.



Fuente: [Ferré, 2003]

- **Roles** Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol. Se representa en el extremo de la asociación junto a la clase que desempeña dicho rol.

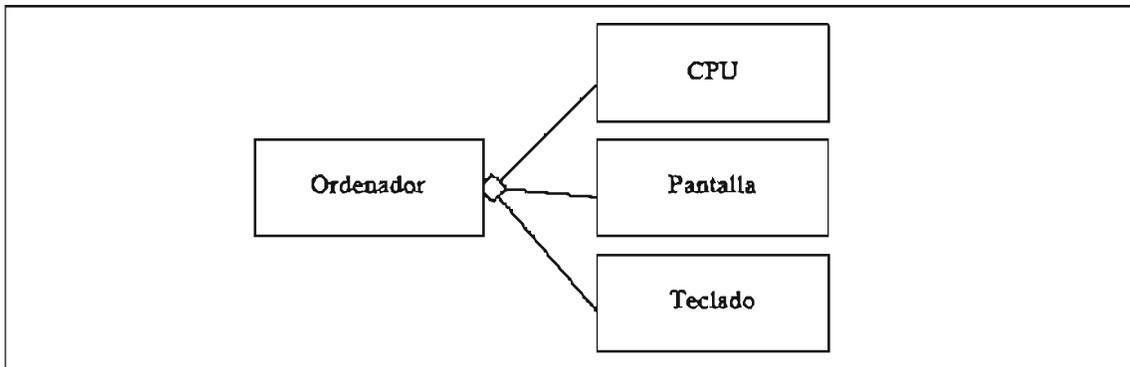
Fig. 2.13. Ejemplo de roles en una asociación.



Fuente: [Ferré, 2003]

- **Agregación.** El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

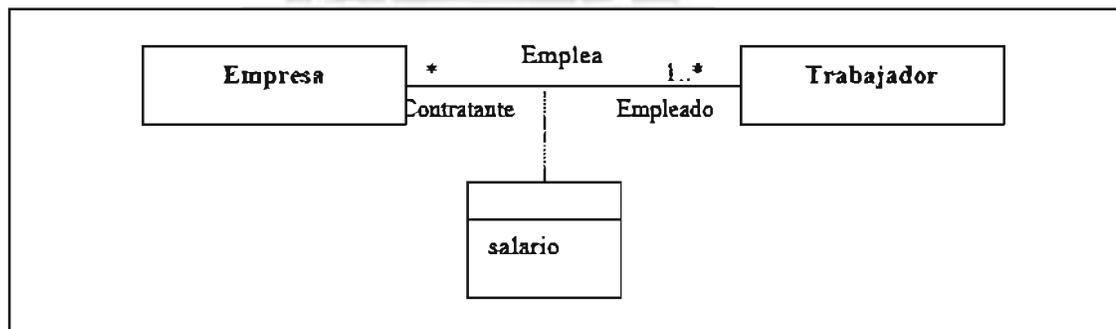
Fig. 2.14. Ejemplo de agregación



Fuente: [Ferré, 2003]

- **Clases Asociación** Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea a trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación. Por tanto ambos tienen el mismo nombre, el de la asociación. Cuando la clase asociación sólo tiene atributos el nombre suele ponerse sobre la línea (como ocurre en el ejemplo de la figura 2.15). Por el contrario, cuando la clase asociación tiene alguna operación o asociación propia, entonces se pone el nombre en la clase asociación y se puede quitar de la línea.

Fig. 2.15. Ejemplo de clase asociación.

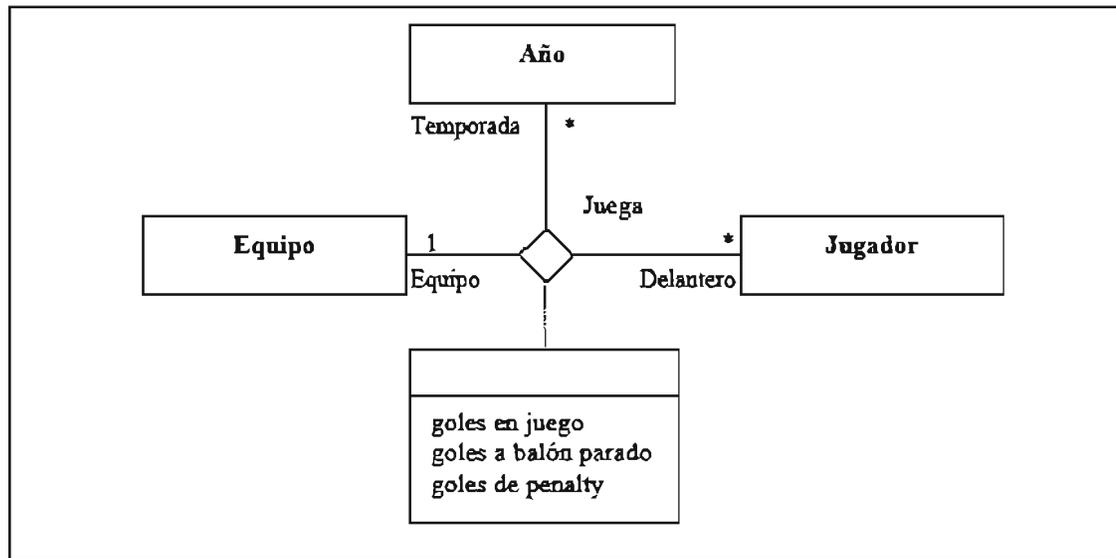


Fuente: [Ferré, 2003]

- **Asociaciones N-Arias** En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante central. Si se muestra multiplicidad en un rol, representa el número potencial de tuplas de instancias en la asociación cuando el resto de los N-1 valores están fijos. En la figura 2.16 se ha impuesto la restricción de que un jugador no puede jugar en dos equipos distintos a lo

largo de una temporada, porque la multiplicidad de “Equipo” es 1 en la asociación ternaria.

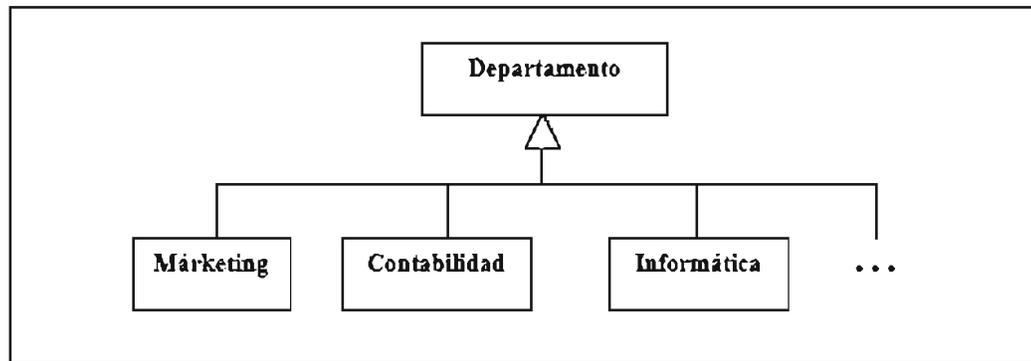
Fig. 2.16. Ejemplo de asociación ternaria.



Fuente: [Ferré, 2003]

- **Navegabilidad** En un extremo de una asociación se puede indicar la navegabilidad mediante una flecha. Significa que es posible "navegar" desde el objeto de la clase origen hasta el objeto de la clase destino. Se trata de un concepto de diseño, que indica que un objeto de la clase origen conoce al objeto(s) de la clase destino, y por tanto puede llamar a alguna de sus operaciones.
- d) **Herencia.** La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre”. Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama concreto no se quieren poner todas, esto se representa mediante puntos suspensivos. En el ejemplo de la figura 2.17., sólo aparecen en el diagrama 3 tipos de departamentos, pero con los puntos suspensivos se indica que en el modelo completo (el formado por todos los diagramas) la clase “Departamento” tiene subclases adicionales, como podrían ser “Recursos Humanos” y “Producción”.

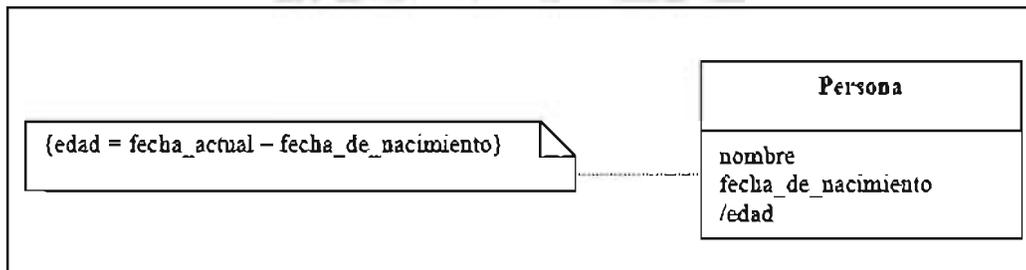
Fig. 2.17. Ejemplo de herencia



Fuente: [Ferré, 2003]

- e) **Elementos Derivados.** Un elemento derivado es aquel cuyo valor se puede calcular a partir de otros elementos presentes en el modelo, pero que se incluye en el modelo por motivos de claridad o como decisión de diseño. Se representa con una barra “/” precediendo al nombre del elemento derivado.

Fig. 2.18. Ejemplo de atributo derivado.



Fuente: [Ferré, 2003]

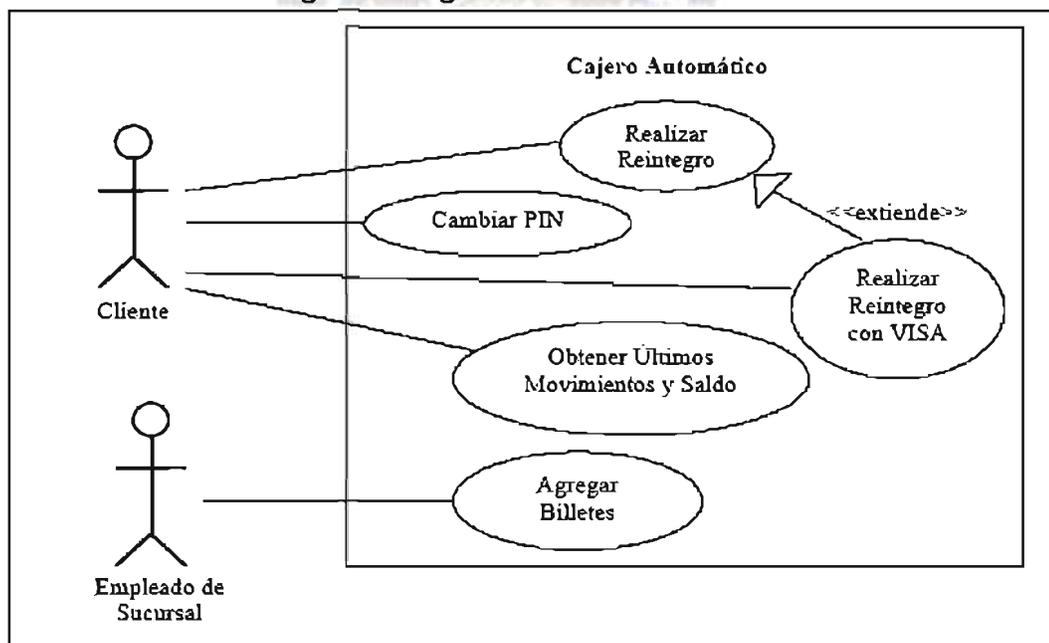
### 2.6.1.2. Diagrama de Casos de Uso

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

- a) **Actores.** Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

- b) **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.
- c) **Relaciones entre Casos de Uso** Entre dos casos de uso puede haber las siguientes relaciones: (1) **Extiende:** Cuando un caso de uso especializa a otro extendiendo su funcionalidad. (2) **Usa:** Cuando un caso de uso utiliza a otro. Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación. 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 2.19 se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

**Fig. 2.19. Diagrama de Casos de Uso.**



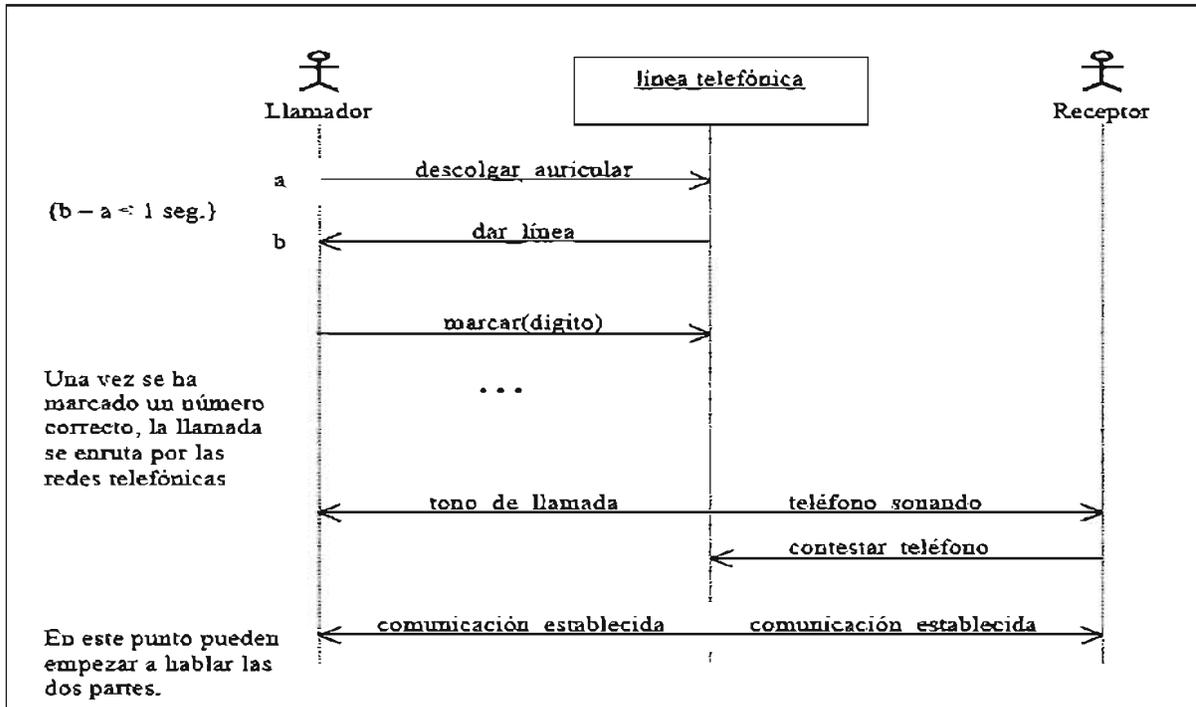
**Fuente: [Ferré, 2003]**

### 2.6.1.3. Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

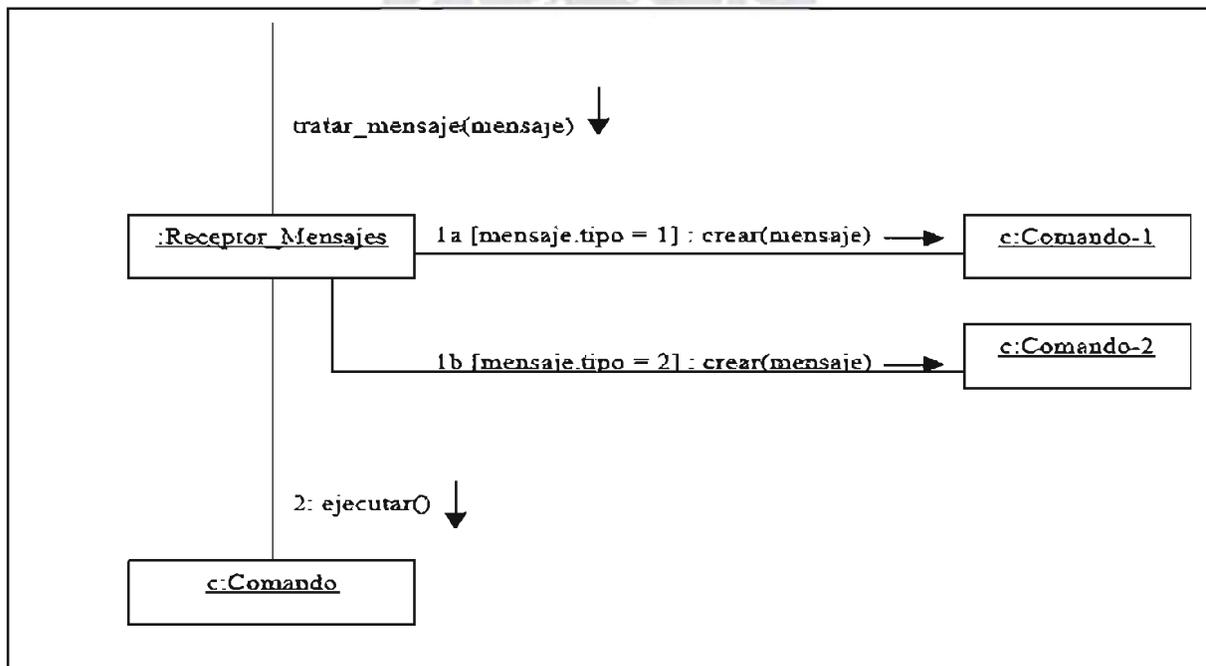
- a) **Diagrama de Secuencia.** Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren. En la figura 2.20. se representa el Diagrama de Secuencia para la realización de una llamada telefónica.
- b) **Diagrama de Colaboración.** Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia. En cuanto a la representación, un Diagrama de Colaboración muestra a una serie de objetos con los enlaces entre los mismos, y con los mensajes que se intercambian dichos objetos. Los mensajes son flechas que van junto al enlace por el que “circulan”, y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis. Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el mensaje que inicia el diagrama, que no lleva número de secuencia. Se pueden indicar alternativas con condiciones entre corchetes (por ejemplo `3 [condición_de_test] : nombre_de_método()`), tal y como aparece en el ejemplo de la figura 2.21. También se puede mostrar el anidamiento de mensajes con números de secuencia como `2.1`, que significa que el mensaje con número de secuencia 2 no acaba de ejecutarse hasta que no se han ejecutado todos los `2.x`.

Fig. 2.20. Diagrama de Secuencia.



Fuente: [Ferré, 2003]

Fig. 2.21. Diagrama de Colaboración

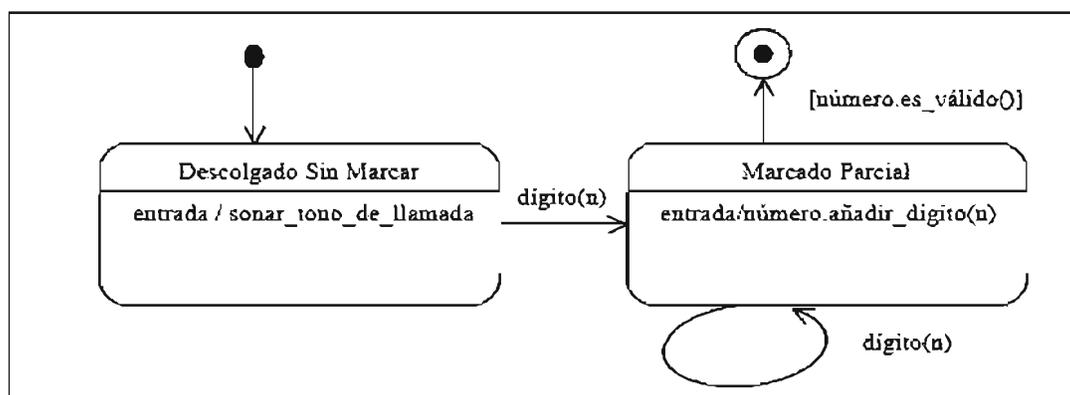


Fuente: [Ferré, 2003]

#### 2.6.1.4. Diagrama de Estados

Un Diagrama de Estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera. En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino. La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas. Una acción de entrada aparece en la forma *entrada/acción\_asociada* donde *acción\_asociada* es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta. Una acción de salida aparece en la forma *salida/acción\_asociada*. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta. Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma *nombre\_de\_evento/acción\_asociada*. Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (del caso de uso o del objeto). El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad, los estados inicial y final son pseudoestados, pues un objeto no puede “estar” en esos estados, pero sirven para saber cuáles son las transiciones inicial y final [Ferré, 2003].

Fig. 2.22. Diagrama de Estados.



Fuente: [Ferré, 2003]

### **2.6.1.5. Levantamiento de requisitos con OOHDM**

La representación gráfica del modelado de la aplicación está respaldada con la notación del UML, a pesar de que ésta representación es realizada en la etapa de análisis de aplicaciones estructuradas y no así para aplicaciones orientadas a objetos, en éste caso y por las características de éste proyecto se puede rescatar ésta forma de análisis para el levantamiento de requisitos con la utilización de los Diagramas de Interacción de Usuario que es propuesta por la metodología OOHDM.

La propuesta de OOHDM para el levantamiento de requisitos, es la identificación de los casos de uso, a partir de ésta premisa en la versión cuatro de la metodología OOHDM, se implementa la utilización de los Diagramas de Interacción de Usuarios (UIDs), consistente en los siguientes procesos:

- Identificación de actores o tareas
- Especificación de escenarios
- Especificación de Casos de Uso
- Especificación de Diagramas de Interacción de Usuarios (UIDs)
- Validación de Casos de Usos de UIDs.

### **2.7. BASES DE DATOS RELACIONALES**

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas. Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo. El modelo relacional representa un sistema de bases de datos en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente, de la misma manera como, por ejemplo, un lenguaje del tipo de PL/1 representa un sistema de programación con un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente. De hecho, el modelo relacional puede considerarse como un lenguaje de programación más bien abstracto, orientado de manera específica hacia las aplicaciones de bases de datos [Date, 1993].

En términos tradicionales una relación se asemeja a un archivo, una tupla a un registro, y un atributo a un campo. Pero estas correspondencias son aproximadas, en el mejor de los casos. Una relación no debe considerarse como "solo un archivo", sino mas bien como un archivo disciplinado, siendo el resultado de esta disciplina una simplificación considerable de las estructuras de datos con las cuales debe interactuar el usuario, lo cual a su vez simplifica los operadores requeridos para manejar esas estructuras.

Según Date (1993) las características principales de los "archivos" relacionales son:

- a) Cada "archivo" contiene solo un tipo de registros
- b) Los campos no tienen un orden específico, de izquierda a derecha
- c) Los registros no tienen un orden específico, de arriba hacia abajo
- d) Cada campo tiene un solo valor
- e) Los registros poseen un campo identificador único (o combinación de campos) llamado clave primaria

Así, todos los datos en una base de datos relacional se representan de una y solo una manera, a saber, por su valor explícito<sup>15</sup>. En particular, las conexiones lógicas dentro de una relación y entre las relaciones se representan mediante esos valores; no existen "ligas" o apuntadores visibles para el usuario, ni ordenamientos visibles para el usuario, ni grupos repetitivos visibles para el usuario, etc.

Actualmente algunos de los manejadores de bases de datos, utilizan un sistema de búsqueda con algoritmos de árboles b. Pero las búsquedas que se pueden realizar con estos algoritmos son sólo para memoria principal. Los algoritmos implementados para realizar búsquedas con listas salteadas o por bloques (*skip lists*) son eficientes para realizar búsquedas en memoria secundaria. Como tienen varios niveles en cada nodo de la lista, permite dar saltos mas largos al realizar las búsquedas, esto provoca que las mismas sean más rápidas.

---

<sup>15</sup> Esta se denomina en ocasiones "principio básico del modelo relacional".

### 2.7.1. Modelo Relacional

Este modelo fue propuesto por Codd el año 1970, quien propone un modelo de datos basados en la teoría de las relaciones, donde los datos se estructuran lógicamente en forma de relaciones (tablas), siendo un objetivo fundamental mantener la independencia de la estructura lógica respecto al modelo de almacenamiento y a otras características del tipo físico.

Dentro las ventajas del modelo relacional se pueden mencionar:

- a) Independencia física. Que el modo en que se almacenan los datos no influya en su manipulación lógica, y por tanto, no sea necesario modificar los programas por cambios en el almacenamiento físico.
- b) Independencia Lógica. Que la modificación de objetos en la base de datos no repercuta en los programas y/o usuarios que estén accediendo al subconjunto parcial de la base de datos.
- c) Flexibilidad. Poder presentar a cada usuario los datos de la forma que prefiera.
- d) Uniformidad. Las estructuras lógicas de datos presentan un estado uniforme.
- e) Sencillez. Es de fácil diseño y comprensión.

Para conseguir esto Codd (1970) introduce el concepto de *Relación* → *Tabla* como estructura básica del modelo. Todos los datos (Entidades / Interrelaciones) de un base de datos se representan en forma de Tablas cuyo contenido varía en el tiempo. Una tabla, es un conjunto de filas, con una misma estructura que es representada por la primera fila, en la terminología relacional. Surge la *Teoría de la Normalización* cuyas tres primeras formas normales fueron introducidas por Codd y constituyen un soporte para un diseño de bases de datos relacionales.

Tablas → Regulares/Uniformes. Deben evitar anomalías.

Con respecto a la componente dinámica del modelo se proponen una serie de operadores que se aplican a las relaciones, algunos clásicos de la Teoría de conjuntos. Todos ellos forman el *Álgebra relacional*.

El elemento básico del modelo relacional es la Tabla (Relación). Se puede distinguir un conjunto de columnas, (atributos), que representan las propiedades de la misma y que están caracterizadas por un nombre. Un conjunto de filas llamadas tuplas, que son las ocurrencias de la tabla. El número de filas de la relación corresponde a la cardinalidad. El número de filas es el grado (numero de atributos). Los dominios son los lugares de donde los atributos toman sus valores.

- a) Dominio y atributo. El dominio es un conjunto de valores homogéneos y atómicos, caracterizados por un nombre. Todo dominio debe tener un nombre por el que uno puede referirse a el y un tipo de datos. Los dominios pueden definirse por intención y por extensión. Un atributo en el papel tiene un determinado dominio en la relación.
- b) Relación. Intención o esquema de la relación. Es la parte definitoria y estática de la relación. Extensión, ocurrencia o instancia de la relación. Es el conjunto de tuplas, que en un instante determinado satisface el esquema correspondiente.
- c) Claves. Las claves tienen la siguiente tipificación (1) Clave candidata. Conjunto no vacío de atributos que identifican unívocamente cada tupla. (2) Clave primaria. La que el usuario escoge de las claves candidatas. (3) Claves alternativas. Claves candidatas que no han sido escogidas. (4) Clave ajena. Conjunto de atributos de la tabla cuyos valores han de coincidir con los de la clave primaria de otra tabla. (Clave ajena y primaria debe estar definida sobre los mismos dominios) [Codd, 1970].

### **2.7.2. Gestor de base de datos relacional MySQL**

El diseño de la base de datos es crucial para el buen funcionamiento de una aplicación Web, si no se cuenta con un diseño de calidad, la operación de la aplicación Web se puede ver entorpecida o de plano anulada. Con el fin de obtener el mejor desempeño y flexibilidad de la aplicación Web, es recomendable seguir ciertas guías para el diseño de la base de datos. Bajo estos lineamientos, es posible desarrollar un sistema de base de datos escalables y que pueden manejar desde la más sencilla consulta hasta la más compleja. Es muy importante cotejar los requerimientos con el diseño de la base de datos, además, conforme se progresa en la fase de desarrollo de la aplicación o incluso después de que ésta es terminada, es posible que surjan nuevos requerimientos; la base de datos debe de ser lo suficientemente flexible para permitir cambios que satisfagan esos requerimientos. Antes de comenzar a crear las tablas se debe crear un diccionario de datos. El diccionario de datos es una referencia que describe las entidades y sus cada uno de sus atributos. En él se especifica el

nombre del campo, su descripción, el tipo de dato (entero, cadena de texto, monetario, etc.) y el tamaño, si aplica. El siguiente paso es crear la base de datos.

MySQL es un administrador de base de datos, el software abierto más popular de SQL<sup>16</sup>. Es desarrollado, distribuido y soportado por MySQL AB, una compañía comercial fundada por los desarrolladores de MySQL.

Se ha optado elegir MySQL como administrador de la base de datos debido a su popularidad y amplitud de uso en los entornos de desarrollo de aplicaciones Web. Esta elección no se debe a que este administrador sea mejor que otros en todos los aspectos, sino debido a una serie de razones que en conjunto hacen del mismo el gestor adecuado para el desarrollo de aplicaciones Web que utilizan PHP como herramienta de desarrollo<sup>17</sup>.

Además de ser gratuito para las principales plataformas en el mercado del software, las licencias pueden verse en su página oficial <http://www.mysql.com>. Al margen de unas herramientas y pequeños programas para diferentes usos, básicamente se compone de un programa que actúa de servidor y que atiende las peticiones de las computadoras cliente (vía TCP), en este caso se hace referencia al mencionado HTTP Server.

MySQL es realmente ligero, no exige muchos recursos del sistema, se maneja por medio del prompt de comando. Es decir que se maneja por consola y no por una aplicación de win32, además que corre como un servicio al instalarlo, así no será necesario ejecutar un programa específico para que la base de datos pueda funcionar correctamente y sin problemas. Esto implica que no llevará al sistema operativo a detenerse, ciclarse y mucho menos saturar en memoria física. Además, como se ha mencionado previamente, es completamente compatible con las plataformas más comunes e importantes del mercado, de igual forma, MySQL es ligero para todas y cada una de estas plataformas.

El uso de MySQL es muy sencillo, es suficiente conocer los comandos básicos en consola para poder crear tablas, definir columnas y su tipo de contenidos e introducirles datos. Existen comandos no básicos que hacen a las tablas aun mas dinámicas, con estos comandos es posible no solo agregar líneas a la base de datos, sino que también es posibles buscar con

---

<sup>16</sup> Se considera al Structured Query Language (SQL) como el precursor de los administradores de las bases de datos modernas.

<sup>17</sup> Entre otras cosas se dispone de licencias, estilo software libre, accesibilidad al código fuente, funcionamiento en las principales plataformas, alto rendimiento bajo determinadas condiciones, etc.

facilidad en este gestor, modificar, buscar rangos, agrupar, organizar y muchas otras cosas mas que se pueden hacer en una base de datos y todo manejado por consola. La mayor ventaja es que constituyen datos que pueden ser manejados de las mismas formas por medio de un script PHP o de algún otro programa compatible.

Una ventaja particular, es que para acceder a la base de datos es necesario (si así se requiere) un nombre de usuario y contraseña, los mismos que están encriptados cuando se conecta al servidor. Esto se realiza a través de un script, el mismo que accede a la base de datos de donde se solicita la información. Este nombre de usuario y contraseña son los que se usarían siempre en los scripts para modificar las tablas y pueden ser modificados a través del archivo de códigos de MySQL (my.ini)

## **2.8. METRICAS DE CALIDAD**

Las aplicaciones basadas en la Web presentan un escenario bastante diferente al de las aplicaciones diseñadas para ser usadas sin conexión a una red (standalone). La naturaleza misma de la Web, prácticamente sin barreras y accesible desde cualquier punto, hacen que la manera de atacar los problemas de usabilidad requieran de un enfoque distinto al de una aplicación de software tradicional. La clave en la usabilidad de un sitio Web es asegurarse de que el sitio sea útil y usable para la audiencia objetivo [Murray & Costanzo, 1999].

Un documento en la Web se enfrenta a una audiencia global, con distintas necesidades, bagajes culturales y niveles educativos. Cada usuario que visita el sitio Web lo hace con un equipamiento de hardware y software distinto. Se debe tomar en consideración que habrá gente que tenga una computadora de última generación con la versión de navegador más actualizada, y una conexión de banda ancha a la red. Y en el otro extremo se puede tener a una persona con un equipo lento, una conexión mala a Internet y software totalmente desactualizado. Si nuestro objetivo es llegar a todo usuario y atenderlos con el mismo nivel de calidad se debe diseñar un sitio Web para que se adapte a las posibles limitaciones de cada usuario.

Quizás una de las decisiones más importantes al momento de comenzar a desarrollar un sitio Web, sea precisamente el definir la audiencia que se va a manejar. Existen varios métodos para recolectar información para sustentar nuestra definición de audiencia; entre

estos se incluye la utilización de focus groups, entrevistas individuales, investigación demográfica y la recolección de datos por parte de usuarios en el caso de un sitio Web pre-existente (encuestas on-line) [Baeza Yates & Rivera, 2003]. En el caso de este último método, una encuesta para definir la audiencia podría recolectar la siguiente información:

- a) Perfil del usuario (información demográfica, ocupación, preferencias en cuanto a recreación, etc.)
- b) Perfil de navegación (cómo usa la Web el usuario).
- c) Uso del sitio (lo que le gusta, lo que no le gusta, que es lo que realiza en él regularmente).
- d) Nivel de tecnología (hardware, tipo de navegador, velocidad de conexión).

### **2.8.1 Heurísticas de usabilidad para la Web**

Jakob Nielsen, en 1990, enumeró diez heurísticas de usabilidad. En ellas se hacían una serie de recomendaciones para verificar puntos críticos en interfaces de usuario, para asegurar que estas tuvieran un alto nivel de usabilidad. Estas heurísticas no fueron escritas específicamente para aplicaciones basadas en la Web [Nielsen, 1990]. En 1997 Keith Instone retomó las heurísticas de Nielsen y las adaptó específicamente para la Web [Instone, 1997].

- a) **Visibilidad del estado del sistema.** El usuario siempre debe saber exactamente qué es lo que el sistema está haciendo. En sitios Web, esto se puede lograr informándole al usuario dónde se encuentra, siempre. Uno de los mayores problemas para los usuarios al navegar por la red es la desorientación. Es vital asegurar una consistencia de todo el sitio Web. Otro aspecto no menos importante, es ofrecer al usuario una permanente retroalimentación a las acciones que este realice en cada página Web. Esto puede lograrse con efectos tan sutiles como el efecto hover de las hojas de estilo (CSS) sobre los hipervínculos.
- b) **Similitud entre el sistema y el mundo real.** El sistema debe hablar el lenguaje del usuario, las frases, palabras y conceptos deben ser familiares para el usuario. Además, se deberán seguir las convenciones usadas en el mundo real, haciendo que la información aparezca en un orden lógico y natural.

- c) **Control por parte del usuario y libertad.** Los usuarios frecuentemente eligen funciones por error y necesitarán de “salidas de emergencia” claramente marcadas. Se deben proveer al visitante del sitio; con funciones para deshacer y rehacer las acciones que haya realizado.
- d) **Consistencia y cumplimiento de estándares.** Los usuarios no tienen por qué preguntarse si distintas palabras, situaciones o acciones significan lo mismo. Hay que seguir las convenciones de las plataformas en las que se está desplegando el sitio Web. Preferir los estilos por default de botones, barras de scroll, etc. provistas por la plataforma. El cumplimiento con las recomendaciones emitidas por el W3C (World Wide Web Consortium) referentes tecnologías Web (HTML, CSS, XML, etc.) aseguran en muchos casos una portabilidad absoluta del sitio Web. Existen una gran cantidad de herramientas (validadores) que permiten verificar si un sitio Web cumple con las especificaciones del W3C. Hay que ser cuidadoso cuando se desarrollan sitios con múltiples páginas, especialmente aquellos elaborados por grupos grandes de desarrolladores. Es importante usar de manera consistente el fraseo, imágenes y fuentes a través del sitio para dar la imagen de consistencia. El uso de hojas de estilo facilita mantener la consistencia del diseño gráfico del sitio. Se recomienda fuertemente desarrollar un plan claro en donde se defina claramente el estilo y layout del sitio Web.
- e) **Prevención de errores** Aún mejor que el desarrollar buenos mensajes de error es tener un diseño cuidadoso que evite la ocurrencia de errores. Hay que asegurarse de que las instrucciones estén escritas de una manera clara y que sean desplegadas de manera conveniente, evitando cualquier tipo de contaminación visual. Si se requiere el llenado de un formulario con campos obligatorios, destáquelos por sobre el resto de las entradas. Es muy conveniente hacer una validación de la forma antes de enviarla al servidor (esto se puede hacer fácilmente con Javascript).
- f) **Preferencia al reconocimiento frente a la memorización.** Haga que los objetos, acciones y opciones sean visibles. El usuario no tiene porque recordar información de una parte de un diálogo a otra parte. Las instrucciones de uso del sistema deben ser visibles y accesibles cuando el usuario lo considere necesario. Tenga mecanismos de búsqueda. Cualquier sitio de más de 200 páginas necesita acceso directo por contenido, no espere que el usuario entienda y navegue hasta encontrar lo que busca.

- g) Flexibilidad y eficiencia de uso.** Los aceleradores –invisibles para el usuario novato – pueden hacer más rápida la interacción para el usuario experto. El sistema debe tratar eficientemente tanto a los usuarios expertos como a los inexpertos. Para lograr esto, es conveniente permitir a los usuarios que personalicen ciertas acciones frecuentes (un ejemplo de esto es la opción "1-Click" de Amazon que ofrece a los compradores habituales el pasar directamente a la confirmación de la venta, sin realizar el procedimiento estándar de facturación). Los sitios Web se deben descargar lo más rápidamente posible, independientemente del tipo de conexión a la red utilizado por el usuario. Hay que dar prioridad al uso de HTML y el reuso de imágenes en el sitio Web.
- h) Estética y diseño minimalista.** Los diálogos no deben contener información que sea irrelevante o que rara vez sea de utilidad. Cada información extra en un diálogo compite con unidades relevantes de información y disminuye su visibilidad relativa. Se recomienda reducir el número de imágenes al mínimo. Hay que recordar que cada imagen implica una descarga desde el servidor, y esto en conexiones lentas puede ser un problema serio. Es altamente recomendable que la información más importante sea colocada en la parte superior de la página, pues esta es la región que siempre es visible en el navegador.
- i) Ayuda para que el usuario reconozca, diagnostique y se recupere de los errores.** Los mensajes de error deben ser expresados en un lenguaje claro (sin ambivalencias), indicando exactamente el problema, y proveyendo constructivamente de una solución. Los mensajes de error pueden ayudar a restablecer la confianza en el sitio Web.
- j) Ayuda y documentación.** Lo más probable es que lo mejor sería que un sistema no requiriera de documentación, pero generalmente se requiere de documentación y una opción de ayuda en línea. Cualquier información debe ser fácil de buscar, y debe estar orientada a las acciones del usuario. En cuanto un sitio Web ofrezca alguna característica fuera de la norma, o ligeramente complicada, será necesario prestar ayuda y dar documentación a los usuarios. Tómese el tiempo de desarrollar un sistema de ayuda que dé auxilio relevante cuando el usuario lo requiera

## 2.8.2. Métricas de calidad Web-Site QEM

Se ha observado que los enfoques utilizados para la evaluación, no se han concentrado en medir a la calidad como una característica de muy alto nivel que contiene y describe a otras características como *usabilidad*, *funcionalidad*, *eficiencia*, *confiabilidad*, u otras como *mantenibilidad* y *portabilidad*. Antes bien, se especializan en evaluar usabilidad y satisfacción del usuario para cierto perfil, sin que siempre se tenga muy en cuenta esa visión integral de la calidad [Olsina, 1999].

Según Olsina (1999), la Metodología de Evaluación de Calidad de Sitios Web (Web-site QEM) es esencialmente integral, flexible y robusto, y cubre la mayor parte de las actividades en el proceso de evaluación, comparación, y selección de artefactos Web. La estrategia propuesta, pretende realizar un aporte de ingeniería 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.

Las aplicaciones de software centradas en la Web son cada vez más complejas y están creciendo rápidamente. Por lo tanto, este rápido crecimiento genera nuevos desafíos, como por ejemplo: cómo diseñar y producir artefactos Web para diferentes dominios, necesidades y comportamientos de usuarios, teniendo a la calidad como objetivo esencial (en consideración de costos); o, desde otro punto de vista, cómo medir, evaluar y potencialmente mejorar la calidad de los sitios Web. (*“Un mapa no sirve si no se sabe dónde se está parado”*). Una estrategia efectiva para encarar estos desafíos es el modelado de procesos y productos, usando enfoques prescriptivos y descriptivos [Olsina, 1998].

Particularmente, el modelado de productos permite comprender, analizar, optimizar, y predecir la calidad de los mismos. Es desde esta perspectiva, la de modelado de procesos y, particularmente, de productos (artefactos Web), que se propone construir y especificar a la metodología. Si bien la metodología puede ser empleada en cualquier fase del ciclo de vida de los productos (a saber, fase de exploración, fase de desarrollo y fase operativa), las etapas en este estudio se focalizarán primeramente en la fase operativa, esto es, en la evaluación de artefactos Web ya existentes u operativos. No obstante, se puede emplear sistemáticamente a Web-site QEM, en la evaluación de nuevos proyectos de desarrollo. Web-site QEM, incluyendo un conjunto de fases, actividades, productos, modelos y

constructores de proceso. Una de las metas principales de la evaluación y comparación de calidad de artefactos Web, radica en comprender el grado de cumplimiento de un conjunto de características y subcaracterísticas con respecto a los requerimientos de calidad establecidos. De este modo, otro aporte interesante consiste en la definición de características, subcaracterísticas y atributos cuantificables considerando dominios de aplicaciones Web particulares. Estas investigaciones permitirán generalizar subcaracterísticas y atributos para conjuntos de dominios: así, patrones de reuso se pueden catalogar a semejanza de los patrones de diseño [Rossi et al., 1997]

## **2.9. ASPECTOS DE SEGURIDAD WEB**

Se verá una presentación sobre la seguridad y los servicios Web. Utilizando el trabajo realizado por Palos (2002) se empieza con una breve introducción de los conceptos y arquitecturas que hay bajo la seguridad de un Servicio Web.

### **2.9.1. Criptografía Asimétrica, Claves Privada y Pública**

Durante la Segunda Guerra Mundial, la marina de los EEUU encontró que una de las mejores formas de encriptar las transmisiones de radio era hablar en el idioma de los Indios Navajos, un idioma al que no se le conocía ninguna conexión lingüística con ningún otro idioma. Nunca se pudo romper esta técnica rudimentaria, y se crearon libros de código más complejos y mecanismos computacionales.

La criptografía moderna ha avanzado considerablemente desde entonces. Para los propósitos de esta explicación, hay dos tipos básicos de algoritmos de criptografía: simétrico (o convencional) y asimétrico (también conocido como criptografía de clave pública-privada). Para la encriptación simétrica, la clave (o código) que se usa para encriptar el mensaje es el mismo que se usa para desencriptarlo. Para la encriptación asimétrica, se utilizan dos claves diferentes para bloquear y desbloquear (encriptar y desencriptar) los mensajes y archivos. Las dos clase se enlazan matemáticamente pero la derivación de una de la otra no es factible matemáticamente hablando. Una clave pública de un individuo se distribuye a otros usuarios y estos la utilizan para encriptar mensajes para ese individuo. El individuo mantiene en secreto la clave privada y la usa para desencriptar los mensajes enviados con la clave pública.

Desde el punto de vista computacional, pasar mensajes usando la encriptación simétrica es más eficiente y consume menos recursos que el método asimétrico. La asimetría tiene sus ventajas cuando se trabaja en grandes comunidades porque se puede distribuir libremente la clave 'pública'.

### **2.9.2. Firma de Datos**

La firma de datos es sólo proporcionar el origen de los datos. Si uno encripta los datos usando la clave privada entonces cualquiera que pueda des encriptarlos correctamente puede fácilmente acceder al origen de los datos: una des encriptación correcta significa que los datos fueron encriptados con la correspondiente clave privada. Como la clave privada siempre se mantiene privada y nunca abandona la posesión de la persona que la generó, la correcta decodificación de los datos protege su origen.

Desafortunadamente, los algoritmos asimétricos son bastante lentos, por eso se usan funciones especiales (como MD5 o HASH). Estas funciones hash especiales primero calculan la huella de los datos (por ejemplo, 16 bytes para MD5 y 20 bytes para HASH) que luego es firmada. La parte receptora calcula la huella usando la misma función, desencripta la huella firmada usando la clave pública, y finalmente compara las huellas calculadas y la desencriptada. Si ambas huellas son iguales entonces se ha verificado el origen de los datos.

### **2.9.3. Políticas de seguridad**

Por otra parte es bueno conocer la importancia del plan de implementación de seguridad. Conocer las amenazas a la información por el hecho de la fragilidad del ambiente que se encuentra en el momento de ser transportada a través de la Internet. Para combatir las amenazas, se debe instalar herramientas, divulgar reglas, concienciar a los usuarios sobre el valor de la información, configurar los permisos. Son varias las medidas de protección que se recomienda para la reducción de las vulnerabilidades de la información. Entre otras:

- a) Protección contra virus
- b) Control de acceso a los recursos de la red

- c) Políticas generales o específicas de seguridad
- d) Entrenamiento
- e) Clasificación de la información
- f) Acceso remoto seguro
- g) Monitoreo y gestión de la seguridad
- h) Infraestructura de llaves públicas (ICP)

Se tiene que tener en cuenta que el objetivo de la implementación de las medidas de seguridad excede los límites de la *informática*, definida como "la ciencia que tiene en vista el tratamiento de la información a través del Internet".

Por otra parte, los servidores Web son designados para recibir solicitudes anónimas desde auténticos hosts en la Internet y a liberar las solicitudes de información en una manera rápida y eficiente. De tal forma, ellos proveen un portal que puede ser usado por amigos y enemigos igualmente. Por su naturaleza, son complicados programas que demandan un alto nivel de seguridad. El tipo de tecnología que mejor cumple con estas demandas se deduce a través de estudios que se realizan para la implementación de servidores Web seguros [Palos, 2002].

## **CAPÍTULO 3**

# **DESARROLLO DE LA APLICACIÓN**

### **Resumen**

*En éste capítulo se describe el proceso de la aplicación de los argumentos teóricos propuestos en anteriores capítulos, se describe de manera secuencial la ingeniería de requerimientos del proyecto, el diseño de los elementos a través de los cuales se hallará la solución a los problemas planteados*

### **3.1. INGENIERÍA DE REQUERIMIENTOS**

La ingeniería de requisitos, tiene como propósito resolver los problemas inherentes en la captura, especificación, validación de los requisitos del sistema. Además, a dichas actividades de la ingeniería de requerimientos se suma el desafío de derivar y gestionar un diseño que satisfaga los requisitos del usuario. Las arquitecturas software son descripciones de alto nivel centradas en definir los elementos que conforman el sistema y las relaciones entre ellos. En una arquitectura software, de manera inicial y antes de su desarrollo, se deben reflejar decisiones de diseño determinadas por los requisitos. En la actualidad existe una convergencia de estas dos áreas que se han ido desarrollando en paralelo. [Navarro,2004]

#### **3.1.1. Especificación de requerimientos para la aplicación**

De acuerdo a lo establecido en el punto 2.3. de capítulo 2, donde se hace una referencia puntual a la especificación de requerimientos para el diseño y desarrollo de productos software, que utiliza el estándar IEEE-STD-830-1998: Especificaciones de los Requisitos del Software, a continuación se presenta un análisis de los requerimientos identificados en las visitas al lugar de estudio, en conversaciones, entrevistas y observación directa de los problemas. Haciendo un análisis de la especificación de requerimientos del software con el uso del estándar IEEE-830 se ha podido identificar los requisitos del cliente para el desarrollo del producto software y la disponibilidad de los mismos. Al proponer una aplicación Web, se debe tener en cuenta la utilización de las herramientas tecnológicas propias de la WWW, que permiten por defecto el servicio de aplicaciones distribuidas basadas en el modelo

cliente/servidor, además del requerimiento de los siguientes componentes de software tanto para el diseño, desarrollo e implementación de la aplicación.

**Tabla 3.1: Requerimientos del software para la aplicación**

TIPO	NOMBRE	FASE DE UTILIZACIÓN
Lenguaje de programación de código abierto.	PHP versión 5.0	Etapa de la implementación, en el desarrollo de la aplicación.
Servidor de aplicaciones Web	Apache versión 2.0.48	Funcionamiento de la aplicación
Gestor de Base de Datos	MySQL versión 3.23	Funcionamiento de la aplicación
Entorno de desarrollo integrado (IDE)	Macromedia Dreamweaver MX 2004	Etapa de la implementación, en el desarrollo de la aplicación.
Diseño gráfico	Macromedia Fireworks MX 2004 Macromedia Free hand MX 2004 Adobe Photoshop 7.0	Etapa de la implementación, en el desarrollo de la aplicación.

El sistema está destinado para su uso en la Carrera de Informática y para servir a la comunidad informática, que en éste caso representan los usuarios que interactúan con el sistema de manera posterior a la etapa de implementación. Además se beneficiará al usuario ocasional; que está conformado por los estudiantes pre-universitarios que tienen interés de formar parte de la Carrera de Informática y a todas las personas que tengan un interés manifiesto en obtener información de la mencionada Carrera.

### 3.1.2. Descripción general del escenario

El punto de partida para el diseño; es precisamente el análisis de la situación actual de los procesos que se realizan en la administración de la Carrera de Informática, específicamente en el área relacionada con la comunicación.

La Carrera de Informática, al igual que las demás carreras pertenecientes a la UMSA, administra su propia información, haciendo uso de procedimientos ya establecidos de acuerdo a sus necesidades; el hecho de publicar avisos y circulares, para mantener informada a la comunidad de temas de interés, el envío y despacho de correspondencia (impresa, firmada y sellada), recopilación de información de los estudiantes, que está

documentada y almacenada en archivos personales en kardex, avisos, anuncios y circulares, entre otros. Los procesos de difusión son aquellos que por lo general hacen uso de medios impresos. El acceso a ésta información por parte de los estudiantes y docentes requiere que se apersonen a las oficinas en el horario establecido de atención al cliente y esperen su turno para ser atendido personalmente, el problema que se tiene es que cierta cantidad de personas llegan para formular un conjunto de preguntas recurrentes, las cuales pueden ser atendidas de manera mecánica.

Actualmente cada subdirección de la Carrera define sus propias actividades, cada subdirección es una unidad integrante de la Carrera; todas las unidades tienen la necesidad de comunicar cierta información de su área, como ser: calendario de actividades, avisos y circulares, convocatorias, actividades y eventos, seminarios, conferencias, cursos, reuniones, asambleas, normativas, noticias, reglamentos, etc.; es por tanto que ésta información debe estar disponible para la comunidad informática en todo momento y el único medio con que se cuenta para publicar parte de ésta información, son la vitrinas y el resto de la información debe ser consultada en bibliotecas; todo éste procedimiento requiere de la presencia del interesado en las instalaciones de la Carrera e invertir tiempo en ese procedimiento resulta complejo para los administradores de la Carrera.

Internamente la administración de cada Unidad en la Carrera, tiene ya definidas las actividades para satisfacer con información a las personas que se apersonan para preguntar, pero que pasa con aquellas personas que por falta de tiempo no pueden ir en el horario establecido a la Unidad para que respondan a sus preguntas, o quieran enterarse de las nuevas noticias referentes a las labores de las cuales está encargada la Unidad. No quedan otras opciones, más que recurrir a otras fuentes de información de la Carrera (estudiantes, docentes, administrativos) y preguntarles de manera personal, ésta última opción no es la más adecuada debido a que no todos están informados, otra opción es llamar por teléfono a la oficina, esperar la atención o la comunicación con el interno correspondiente y esperar a que la persona que informa se encuentre desocupada y de buen humor, para responder a las peticiones de información.

En conversaciones realizadas con los usuarios estudiantes, estos manifestaron las siguientes inquietudes:

- ¿Cuáles son los trámites que se realizan en la Carrera de Informática?,
- ¿En qué circunstancias se debe iniciar algún trámite?,
- ¿Cuándo se debe iniciar un trámite determinado?,
- ¿Cuál es el procedimiento que sigue un trámite?,
- ¿Cuáles son los requisitos que se deben cumplir?,
- ¿En que Unidad se debe realizar el trámite?,
- ¿Cuál es el estado del trámite en proceso?

Las respuestas a estas interrogantes se las puede encontrar haciendo las consultas directas a la Unidad encargada o a las personas que ya han pasado por el proceso de elaboración de un trámite de interés. En el caso de que el trámite ya fuera iniciado por el estudiante, el mismo espera recoger la respuesta del trámite o en definitiva, si es que se requiere un procedimiento más largo para el mismo, debe ir a informarse acerca del estado de su trámite y si es que tiene alguna observación. La gestión del trámite es realizada por el estudiante de manera personal en la Unidad correspondiente al trámite.

Con base en las razones anotadas es menester recalcar la importancia de contar con un sistema de información basado en la Web que proporcione soluciones a los problemas mencionados, utilizando de manera eficiente los procedimientos de comunicación y difusión de información útiles a la comunidad informática.

### **3.1.3. Análisis de requisitos**

La descripción general del escenario además de las características inherentes al proyecto planteado, constituyen los parámetros fundamentales para la identificación de requerimientos, sobre los mismos se debe hacer un análisis y estudio pertinente antes de elaborar el diseño de la aplicación.

Los requisitos globales identificados son las características con las que debe contar la aplicación para proporcionar un manejo fácil de la información para los usuarios que son los docentes, alumnos y personas interesadas en informarse acerca de los servicios que ofrece la Carrera de Informática. Estos requisitos globales hacen referencia a los siguientes puntos de interés:

- Entrega de información actualizada y organizada.
- Administración de la información de fácil acceso, y que la interfaz de usuario proporcione suficiente referencia en la presentación, respecto a los contenidos del sistema.
- Que las interfaces destinadas a la actualización de la información, sean sencillas y de fácil manejo.

De manera complementaria a los requerimientos globales, se requiere que el sistema a ser desarrollado:

- Pueda ser visitado desde cualquier parte del mundo.
- Muestre la información institucional de la Carrera de Informática.
- Proporcione documentos normativos, reglamentos internos y resoluciones.
- Muestre noticias, eventos, avisos, comunicados y circulares, entre otros.
- Proporcione el cronograma de actividades establecido para el año en curso, con información referente a inicio de clases, cursos de verano, temporadas de matriculación, etc.
- Muestre los tipos de trámites que se ejecutan en la Carrera.
- Dé a conocer los requisitos de cada trámite.
- Muestre el procedimiento de cada trámite.
- Proporcione el tiempo de duración de un determinado trámite.
- Muestre recursos reutilizables, como cartas tipo, que puedan ser descargados de modo inmediato.
- Proporcione información referente al estado del trámite que se encuentra en ejecución
- Proporcione seguridad en el acceso, con autenticación, a la página personal.

De manera funcional el sistema debe encontrarse organizado a través de roles de administración.

- Administrador principal, que se encargue de administrar los permisos a los demás administradores.
- Administrador de unidad, que administre los módulos asignados.
- Sub administrador de unidad, que puede ingresar con permiso a realizar ciertas modificaciones en la información que se está publicando.

- Página personal del docente, que autoriza a cada docente de forma individual el acceso, con la concesión de permiso a publicar el material intelectual o material de referencia.
- Página personal del estudiante, que permite a los estudiantes revisar la información que les llega, ver la documentación que tiene su archivo personal en Kardex.

En relación con las características adicionales, el sistema debe permitir:

- Que la navegación del sistema sea sencilla y que el tiempo de recuperación de funciones del software sea rápido.
- Que el acceso al área restringida sea con autenticación de usuario
- Que el acceso a la administración de servicios del sistema sea con la autenticación de usuario.
- Que se verifique con autenticación al usuario que edita la información que se publica en el sitio.
- Que identifique a los actores que intervienen en el sistema, las tareas que desarrollan y las relaciones que existen entre los actores y las tareas.

#### **3.1.4. Especificación de requerimientos del usuario**

El usuario que forma parte de la comunidad informática, conformada por docentes administrativos y estudiantes, además de los usuarios ocasionales, identificaron las siguientes necesidades que conforman la base de sus requerimientos:

- Información institucional de la Carrera de Informática.
- Calendario anual de actividades planificadas por la Carrera de Informática.
- Normas y reglamentos de uso cotidiano en la Carrera de Informática.
- Publicaciones de los docentes y estudiantes que oferta la Carrera de Informática.
- Noticias de importancia que se generan en la Carrera de Informática.
- Tramites que se realizan en la Carrera de Informática.
- Requisitos para emprender los tramites que se efectúan en la Carrera de Informática
- Estado de los tramites en las Unidades de la Carrera de Informática.
- Consulta y actualización de los archivos personales de docentes y estudiantes de la Carrera de Informática.

## 3.2. DISEÑO DEL SISTEMA

### 3.2.1. Identificación de Actores y Tareas

Los actores que se identifican en esta sección están mencionados de acuerdo a las personas, los sistemas y los roles de usuario, que interactúan con el sistema.

1. **Visitante:** Cualquier persona que ingrese al portal para consultar información general. Este actor puede realizar las siguientes acciones:
  - I. Consultar la Información Institucional: Misión, objetivos, estructura organizacional, autoridades, ubicación, entre otros.
  - II. Consultar el calendario de actividades.
  - III. Consultar las normas y reglamentos con los que se trabajan en la carrera de Informática.
  - IV. Consultar las noticias generadas en la carrera de Informática.
  - V. Revisar las publicaciones de los docentes de la carrera de Informática.
  - VI. Consultar los trámites que se pueden efectuar en la carrera de Informática.
  
2. **Secretaria de Unidad:** Es la encargada de los asuntos administrativos de la Unidad. Este actor puede realizar las siguientes acciones:
  - I. Recibir los requisitos de los trámites
  - II. Actualizar el estado de los trámites y realizar su seguimiento
  - III. Actualizar el Archivo Personal del alumno (en el caso del Área Académica)
  - IV. Atender las consultas de los alumnos
  - V. Solicitar información específica a otras unidades
  - VI. Enviar información específica a otras unidades
  
3. **Auxiliar de Oficina:** Es la persona encargada de colaborar con el trabajo de la secretaria. Este actor puede realizar las siguientes acciones:
  - I. Publicar avisos o circulares relativos a esta unidad.
  - II. Atender las consultas de los alumnos
  - III. Actualizar y publicar el calendario de actividades de la unidad

4. **Docente:** Es la persona encargada del proceso de enseñanza-aprendizaje al interior de la Carrera. Este actor puede realizar las siguientes acciones:
  - I. Actualizar la información personal con datos generales.
  - II. Actualizar su hoja de vida.
  - III. Publicar el material intelectual de su autoría o interés.
  
5. **Alumno:** Es la persona receptora del proceso de enseñanza-aprendizaje. Este actor puede realizar las siguientes acciones:
  - I. Consultar la información de su archivo personal.
  - II. Consultar la información y el estado de los trámites que está realizando en la unidad.
  - III. Solicitar el inicio de un trámite de su interés.
  - IV. Enviar mensajes a la unidad pidiendo que se le aclare algún tema de su interés.
  
6. **Administrador de la Aplicación:** Es la persona encargada de la administración del sistema. Este actor puede realizar las siguientes acciones:
  - I. Definir los roles de los usuarios del sistema.
  - II. Registrar los módulos del sistema.
  - III. Asignar roles a los usuarios.
  - IV. Crear las unidades de la carrera, en el sistema.

### 3.2.2. Especificación de escenarios

#### Actor 1: Visitante

Escenario 1: Consultar la Información Institucional

Descripción: Se desea obtener información general acerca de la Carrera de Informática; para obtener esta información se debe seleccionar las opciones de: **Objetivos, Estructura Organizacional, Autoridades y Ubicación** en la ventana principal.

Escenario 2: Consultar el calendario de actividades

Descripción: Se requiere saber cuando se llevará a cabo la conferencia sobre seguridad informática. En la aplicación se debe seleccionar la opción de **Calendario de Actividades** y en la ventana que aparece, se muestra todas las actividades del mes; luego al seleccionar la

actividad deseada se muestra los detalles de la misma: nombre de la actividad, fecha, hora, lugar, el responsable y una breve descripción.

Escenario 3: Consultar las normas y reglamentos de la carrera de Informática.

Descripción: Se desea saber cuales son los deberes de los docentes de la carrera de informática; se selecciona la opción de **Normas y Reglamentos**, luego se muestra una lista con el nombre del documento, el tipo de documento (norma, reglamento, políticas, etc.), una descripción del documento y un archivo para descargar.

Escenario 4: Consultar las noticias generadas en la carrera de Informática.

Descripción: Se quiere saber quien ganó en las elecciones del Centro de Estudiantes; para esto se selecciona la opción de **Noticias** de la ventana principal y a continuación se muestra las noticias generadas durante la última semana.

Escenario 5: Consultar las publicaciones de los docentes de la carrera de Informática.

Descripción: Se quiere saber que artículos publicó el Lic. Silva, se selecciona la opción de **Docentes** en la ventana principal y se muestra una lista con todos los docentes de la carrera, se elige el nombre de un docente y se muestra la información general del docente, su hoja de vida y todas las publicaciones.

Escenario 6: Consultar los trámites que se pueden efectuar en la carrera de Informática.

Descripción: En la ventana principal se selecciona la opción de **Trámites** y luego se muestra una lista con todos los trámites que se pueden realizar en la Carrera de Informática más una descripción del trámite y los procedimientos que se deben seguir.

## **Actor 2: Secretaria de Unidad**

Escenario 1: Recepciona los requisitos de los trámites

Descripción: Se quiere registrar el trámite de convalidación de materias para el alumno Juan Pérez; se ingresa como Secretaria a la aplicación y se selecciona la opción de **Trámites de Alumno** se busca al alumno mediante su código, se selecciona el trámite que solicita, se registra los requisitos del trámite en el sistema y luego el estado en el que se encuentra el trámite.

Escenario 2: Actualizar el estado de los trámites y realizar su seguimiento

Descripción: Se quiere registrar el trámite de convalidación de materias para el alumno Juan Pérez; se ingresa como Secretaria a la aplicación y se elige la opción de **Trámites de Alumno** se busca al alumno mediante su código, se selecciona el trámite del alumno para actualizar, y se actualiza el estado actual del trámite.

Escenario 3: Actualizar el Archivo Personal del alumno (caso particular del Área Académica)

Descripción: Se quiere actualizar el Archivo Personal del alumno Juan Pérez; se ingresa como Secretaria a la aplicación y se selecciona la opción **Alumnos**, luego se selecciona de la lista de alumnos a Juan Pérez o mediante la **Opción de Búsqueda** se busca al alumno, se selecciona la opción de archivo personal y a continuación se actualiza el archivo personal del alumno.

Escenario 4: Atender consultas de los alumnos

Descripción: Se desea saber cuáles son las consultas de los alumnos; se ingresa como Secretaria a la aplicación y se selecciona la opción **Mensajes de Alumnos**, se selecciona la opción de mensajes no leídos o no atendidos de alumnos, se selecciona un mensaje y se responde a la consulta realizada por el alumno.

Escenario 5: Solicitar información específica a otras unidades

Descripción: Se solicita la lista de alumnos de la materia de INF-111; se ingresa como Secretaria a la aplicación y se elige la opción **Mensajes de Unidad**, se selecciona la opción de enviar mensaje y se solicita la información deseada a la unidad correspondiente, se selecciona la Unidad que solicita la información y se envía el mensaje respectivo.

Escenario 6: Enviar información específica a otras unidades

Descripción: Se envía la lista de alumnos de la materia de INF-111; se ingresa como Secretaria a la aplicación y se elige la opción **Archivos de Unidad**, se selecciona la opción **Enviar Archivo**, se selecciona la Unidad que solicita esta información y se envía el archivo con la información solicitada.

### **Actor 3: Auxiliar de Oficina**

Escenario 1: Publicar avisos o circulares

Descripción: Se desea dar a conocer a los docentes que tienen que presentar sus formularios para el descargo RC-IVA; se ingresa como Auxiliar de oficina a la aplicación y se selecciona la opción **Avisos y Circulares**, se elige la opción **Nuevo** aviso o circular y se registra el tipo (aviso, circular u otro), fecha, una descripción, el responsable y a quienes va dirigido.

Escenario 2: Atender las consultas de los alumnos

Descripción: Se desea saber cuales son las consultas de los alumnos, se ingresa como Auxiliar de oficina a la aplicación y se selecciona la opción **Mensajes de Alumnos**, se selecciona la opción de mensajes no leídos o no atendidos de los alumnos, se selecciona un mensaje y se envía la respuesta a la consulta del estudiante.

Escenario 3: Actualizar o publicar una actividad

Descripción: Se desea dar a conocer que se llevará a cabo el seminario sobre Internet; se ingresa como Auxiliar de oficina a la aplicación y se selecciona la opción **Calendario de Actividades**, se selecciona la fecha en la cual se realizará la actividad, se selecciona **Nueva Actividad** y se registra el nombre de la actividad, fecha inicial, fecha de finalización, la hora, el lugar, el responsable y una descripción de la actividad.

#### **Actor 4: Docente**

Escenario 1: Actualizar información general del docente

Descripción: Se desea actualizar el correo electrónico; se ingresa como Docente a la aplicación y se selecciona la opción de **Modificar** datos generales, se actualizan los datos necesarios y se guarda esta información.

Escenario 2: Actualizar hoja de vida

Descripción: Se desea actualizar la hoja de vida del docente; se ingresa como Docente a la aplicación y se selecciona la opción **Hoja de Vida**, luego se procede a actualizar el archivo con la hoja de vida del docente.

Escenario 3: Publicar el material intelectual de interés del docente

Descripción: Se desea publicar el artículo “La Informática y la Educación”; se ingresa como docente a la aplicación y se elige la opción de **Publicaciones**, se selecciona un tipo de

publicación o se procede a crear un nuevo tipo de publicación que se ajuste a las necesidades del docente, se define el tipo y las partes de la publicación, y luego se registra el contenido de la publicación. Si solo se quiere publicar un archivo con el contenido de la publicación se selecciona la opción de **Publicar Archivo**, se registra el nombre, una descripción y el archivo con el contenido total.

#### **Actor 5: Alumno:**

Escenario 1: Consultar la información del archivo personal

Descripción: Se desea revisar que documentos se encuentran en el archivo personal; se ingresa como alumno a la aplicación y se selecciona la opción **Archivo Personal**, luego se muestra la documentación que se encuentra registrada en el archivo personal.

Escenario 2: Consultar la información y el estado del trámite que está realizando en la unidad

Descripción: Se desea saber cuál es el estado del trámite en proceso: Conclusión de Materias; se ingresa como alumno a la aplicación y se selecciona la opción **Trámites de Alumno**, se muestra una lista con todos los trámites que se están efectuando actualmente en las unidades de la Carrera, se selecciona el trámite de interés y se verifica el estado del mismo.

Escenario 3: Enviar mensaje a las unidades

Descripción: Se ingresa como alumno a la aplicación y se selecciona la opción **Mensajes**, se selecciona **Nuevo Mensaje** y se registra el asunto del mensaje, la fecha y el contenido del mensaje, luego se envía el mensaje.

#### **Actor 6: Administrador de la Aplicación:**

Escenario 1: Definir los roles de los usuarios del sistema

Descripción: Se desea crear el rol Auxiliar de Oficina; se ingresa como Administrador de la Aplicación y se elige la opción **Roles de Usuario**, se selecciona **Nuevo Rol de Usuario** se registra el nombre del rol y una descripción del mismo.

Escenario 2: Registrar los módulos del sistema.

Descripción: Se desea registrar el módulo de Calendario de Actividades; se ingresa como Administrador de la Aplicación y se elige la opción **Módulos del Sistema**, se selecciona **Nuevo Módulo** y se registra el nombre del módulo, una descripción del mismo, el nombre con el que aparecerá en los menús y el archivo de origen.

Escenario 3: Asignar roles a los usuarios

Descripción: Se desea asignar el rol Editor de Calendario a Juan Pérez; se ingresa como Administrador de la Aplicación y se elige la opción **Buscar Usuario**, se selecciona la opción **Asignar Rol de Usuario** y se asigna a Juan Pérez el rol Editor de Calendario.

Escenario 4: Crear las unidades de la carrera, en el sistema.

Descripción: Se desea crear la Unidad Laboratorio Superior de Informática; se ingresa como Administrador de la Aplicación y se elige la opción **Unidades**, se selecciona **Nueva Unidad** y se registra el nombre, la sigla, la dirección, teléfono, el correo electrónico y la dependencia de la Unidad.

### 3.2.3. Casos de Uso

**Caso de Uso 1:** Consultar Información Institucional

Escenario: 1.1

Descripción:

1. El usuario selecciona una de las opciones del menú Información Institucional
2. El sistema retorna en una ventana la información relacionada a la opción escogida

**Caso de Uso 2:** Consultar el calendario de actividades

Escenario: 1.2

Descripción:

1. El usuario selecciona la opción del menú Calendario de Actividades
2. El sistema retorna un calendario visual, con año, mes y los días del mes seleccionado
3. El usuario selecciona un año o mes
4. El sistema retorna con una lista de las actividades del mes seleccionado
5. El usuario selecciona una actividad
6. El sistema retorna los detalles de la actividad: nombre de la actividad, fecha, hora, lugar, el responsable y una breve descripción

### **Caso de Uso 3:** Consultar normas y reglamentos

Escenario: 1.3

Descripción:

1. El usuario selecciona la opción del menú Normas y Reglamentos
2. El sistema retorna una lista de documentos que contiene: el título del documento, el tipo de documento, una breve descripción, y un archivo para descargar.
3. El usuario descarga el archivo requerido

### **Caso de Uso 4:** Consultar Noticias

Escenario: 1.4

Descripción:

1. El usuario selecciona la opción del menú Noticias.
2. El sistema retorna una lista de las noticias generadas en la Carrera de Informática, donde se observa: el título de la noticia, la fecha y una breve descripción.
3. El usuario selecciona una noticia.
4. El sistema retorna los detalles de la noticia: el título de la noticia, la fecha, el cuerpo, una foto, la fuente y el autor.

### **Caso de Uso 5:** Consultar Publicaciones de la Carrera de Informática

Escenario: 1.5

Descripción:

1. El usuario selecciona la opción del menú Publicaciones.
2. El sistema retorna una lista de las publicaciones generadas en la Carrera de Informática, donde se ve: el título de la publicación, la fecha, una breve descripción y un archivo para descargar.
3. El usuario selecciona una publicación.
4. El sistema retorna los detalles de la publicación; se muestra las partes de la publicación con sus respectivos contenidos.

### **Caso de Uso 6:** Consultar Lista de Trámites

Escenario: 1.6

Descripción:

1. El usuario selecciona la opción del menú Trámites.

2. El sistema retorna una lista de los trámites que se pueden efectuar en la Carrera de Informática y la unidad en la que se efectúa el mismo.
3. El usuario selecciona un trámite.
4. El sistema retorna los detalles del trámite: el nombre del trámite, una descripción del trámite, el procedimiento que se debe seguir y los requisitos necesarios.

**Caso de Uso 7:** Registrar requisitos de los trámites del alumno

Escenario: 2.1

Descripción:

1. El usuario ingresa a buscar al alumno.
2. El sistema retorna una lista con los resultados de la búsqueda.
3. El usuario selecciona el alumno y selecciona la opción Nuevo Trámite.
4. El sistema muestra los requisitos del trámite seleccionado.
5. El usuario registra los requisitos presentados por el alumno.

**Caso de Uso 8:** Actualizar el estado del trámite

Escenario: 2.2

Descripción:

1. El usuario selecciona la opción del menú Buscar Alumno.
2. El sistema retorna una lista con los resultados de la búsqueda.
3. El usuario selecciona el alumno y selecciona la opción Trámites de Alumno.
4. El sistema muestra los detalles del trámite seleccionado y actualiza el estado actual del trámite.

**Caso de Uso 9:** Actualizar el archivo personal del alumno

Escenario: 2.3

Descripción:

1. El usuario selecciona la opción del menú Buscar Alumno.
2. El sistema retorna una lista con los resultados de la búsqueda.
3. El usuario selecciona el alumno y selecciona la opción Archivo Personal.
4. El sistema muestra los documentos del archivo personal del alumno seleccionado.
5. El usuario registra los documentos presentados por el alumno.

**Caso de Uso 10:** Atender consultas de los alumnos

Escenario: 2.4, 3.2

Descripción:

1. El usuario selecciona la opción del menú Mensajes.
2. El sistema retorna una lista con los mensajes enviados por los alumnos.
3. El usuario selecciona un mensaje de la lista.
4. El sistema muestra el contenido del mensaje.
5. El usuario responde al mensaje enviado por el alumno.

**Caso de Uso 11:** Solicitar información específica a otras unidades

Escenario: 2.5

Descripción:

1. El usuario selecciona la opción del menú Unidades.
2. El sistema retorna una lista con los nombres de las unidades.
3. El usuario selecciona Nuevo Mensaje.
4. El sistema muestra el formulario de nuevo mensaje.
5. El usuario registra el asunto del mensaje, la unidad a la que está destinada, el contenido.

**Caso de Uso 12:** Enviar información específica a otras unidades

Escenario: 2.6

Descripción:

1. El usuario selecciona la opción del menú Unidades.
2. El sistema retorna una lista con los nombres de las unidades.
3. El usuario selecciona Enviar Archivo.
4. El sistema muestra el formulario de envío de archivos.
5. El usuario registra el archivo con la información solicitada y una breve descripción.

**Caso de Uso 13:** Publicar Avisos o Circulares

Escenario: 3.1

Descripción:

1. El usuario ingresa al módulo de Avisos o Circulares.
2. El sistema retorna una lista de los avisos o circulares publicados por la unidad.
3. El usuario selecciona Nuevo aviso o circular.
4. El sistema muestra el formulario de publicación de aviso o circular.

5. El usuario registra el aviso o circular.

**Caso de Uso 14:** Actualizar y publicar una actividad

Escenario: 3.3

Descripción:

1. El usuario ingresa al módulo de Calendario de Actividades.
2. El sistema retorna una descripción visual de un calendario con: días, meses y años; y también muestra una lista con las actividades y horas, asociadas a la fecha actual.
3. El usuario selecciona Nueva actividad.
4. El sistema solicita el registro de la actividad.
5. El usuario registra los datos relacionados con la actividad.

**Caso de Uso 15:** Actualizar información general del docente

Escenario: 4.1

Descripción:

1. El usuario ingresa a la página personal del docente.
2. El sistema retorna los datos generales del docente.
3. El usuario selecciona Modificar datos generales.
4. El sistema solicita la actualización de los siguientes datos: dirección, teléfono, correo electrónico y fotografía.
5. El usuario modifica la información presentada.

**Caso de Uso 16:** Actualizar hoja de vida del docente

Escenario: 4.2

Descripción:

1. El usuario ingresa a la página personal del docente.
2. El sistema retorna la información de la hoja de vida del docente.
3. El usuario selecciona Modificar datos de la hoja de vida.
4. El sistema solicita que actualice la información de la hoja de vida.
5. El usuario modifica la información presentada.

**Caso de Uso 17:** Publicar material intelectual del docente

Escenario: 4.3

Descripción:

1. El usuario ingresa a la página personal del docente.
2. El sistema retorna una lista con todas las publicaciones disponibles.
3. El usuario selecciona Nueva publicación.
4. El sistema pide que registre información de la publicación que se encuentra de acuerdo a una plantilla de documento.
5. El sistema pide el registro de un archivo para descargar.
6. El usuario registra la información solicitada.
7. El usuario crea una nueva plantilla y vuelve al paso 2.

#### **Caso de Uso 18:** Consultar Archivo Personal

Escenario: 5.1

Descripción:

1. El usuario ingresa a la página personal del alumno.
2. El sistema retorna dos listas: la primera con los documentos faltantes u observados, y la segunda con los documentos que se encuentran registrados correctamente.
3. El usuario pasa al Caso de Uso 9, si necesita alguna aclaración o consulta.

**Caso de Uso 19:** Consultar la información y el estado de un trámite que se esta realizando en una unidad

Escenario: 5.2

Descripción:

1. El usuario ingresa a la página personal del alumno.
2. El sistema retorna una lista con trámites efectuados por el alumno.
3. El usuario selecciona uno de los trámites de la lista.
4. El sistema muestra los detalles de la información del trámite, el estado del mismo y los mensajes relacionados con el trámite.
5. El usuario revisa la información presentada.
6. El usuario pasa al Caso de Uso 8 si lo requiere.

**Caso de Uso 20:** Enviar mensajes a la unidad

Escenario: 5.4

Descripción:

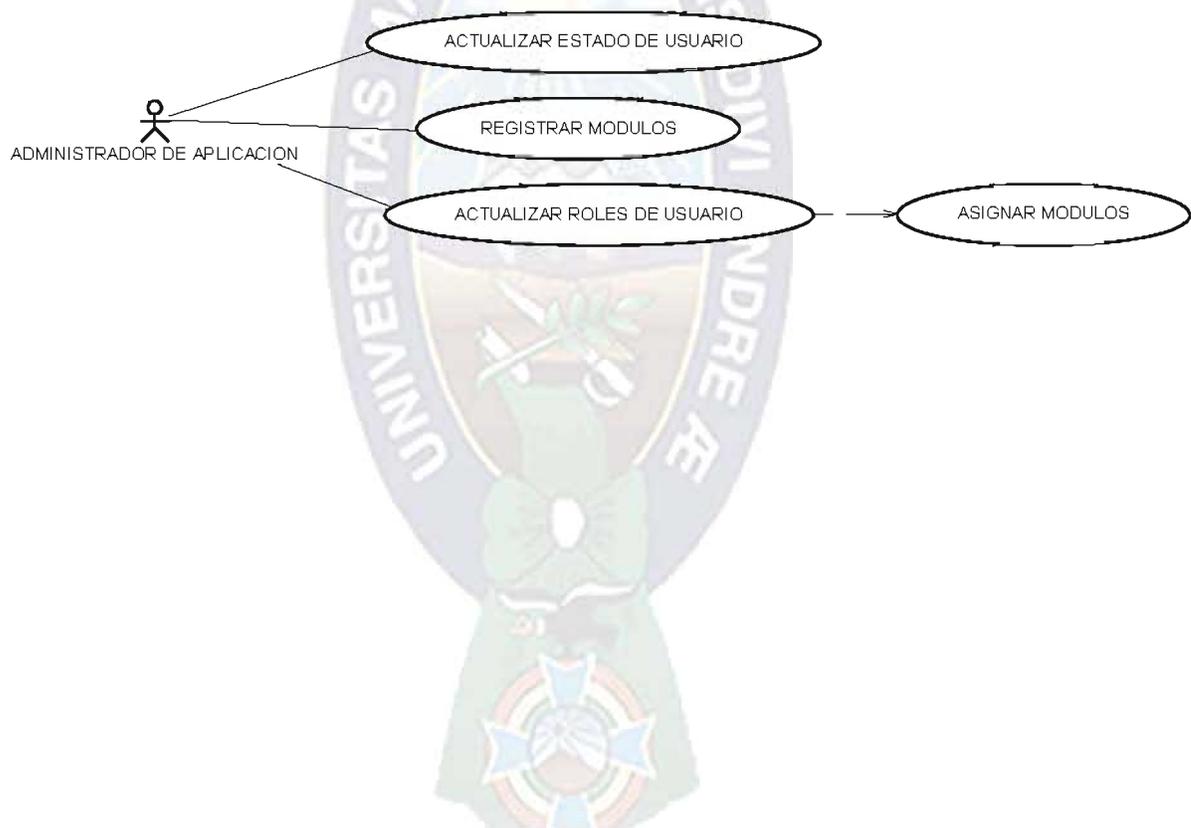
1. El usuario ingresa a la página personal del alumno.
2. El sistema retorna una lista con los mensajes no leídos y otra con los mensajes leídos.

3. El usuario selecciona Nuevo Mensaje.
4. El sistema le pide que registre el mensaje y la unidad de la Carrera de Informática a la que va dirigido.
5. El usuario registra la información solicitada.

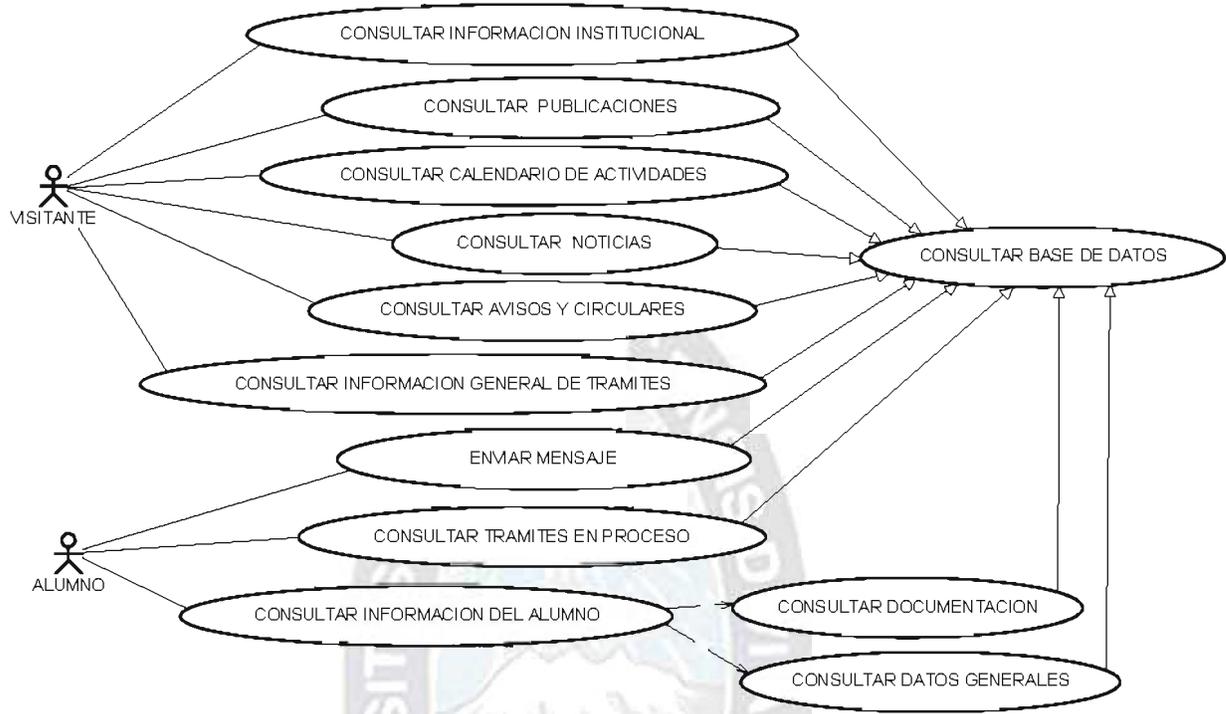
### 3.2.4. Diagramas de caso de uso

Un conjunto de casos de uso identifica gráficamente los casos de uso de un sistema, los actores y las relaciones entre estos y los casos de uso propiamente dicho .

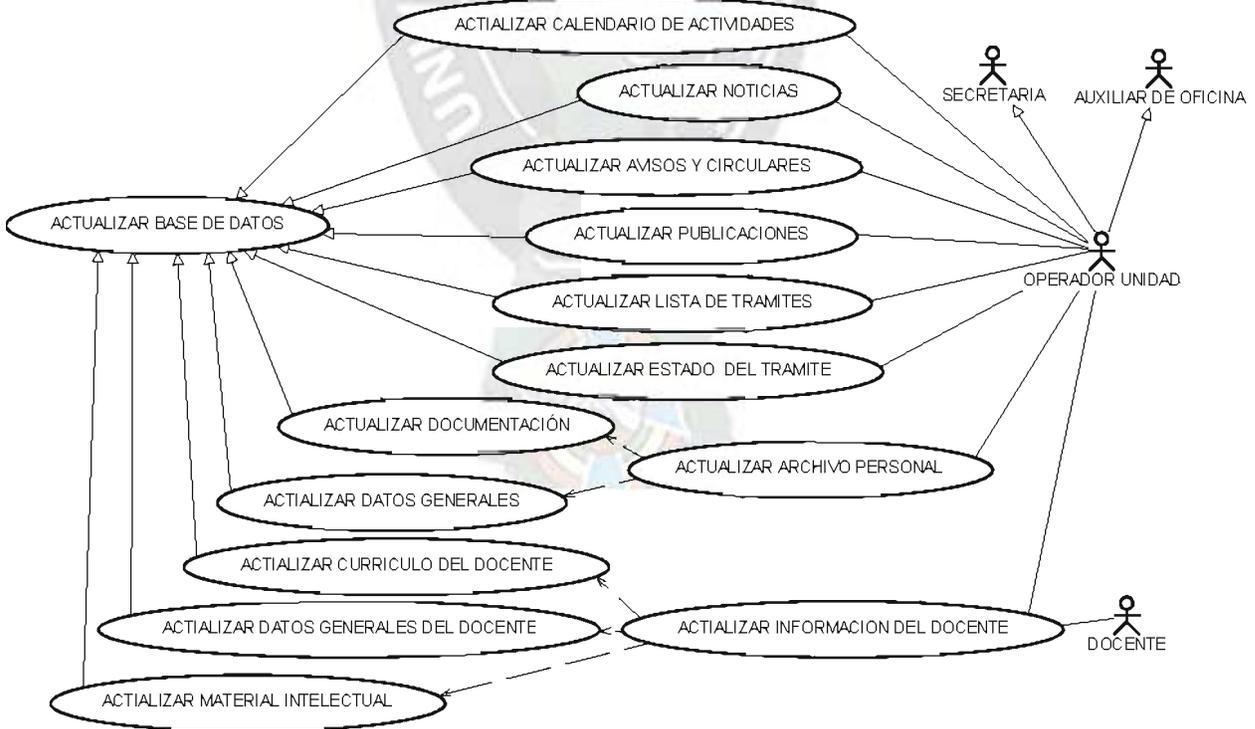
**Fig. 3.1. Diagrama de Casos de Uso general**



**Fig. 3.2. Diagrama de Casos de Uso General (Continuación)**



**Fig. 3.3. Diagrama de Casos de Uso General (Continuación)**



### 3.2.5. Diagramas de interacción de usuarios

Los diagramas de interacción de usuarios que se presentan a continuación indican como se relacionan los actores del sistema con el mismo, además permiten identificar las clases, atributos y operaciones que se necesitan para el diseño conceptual y establecer claramente el dominio del problema.

En la figura 3.3 se muestra los UID's del sistema.

**Fig. 3.4. Diagramas de Interacción de Usuarios**

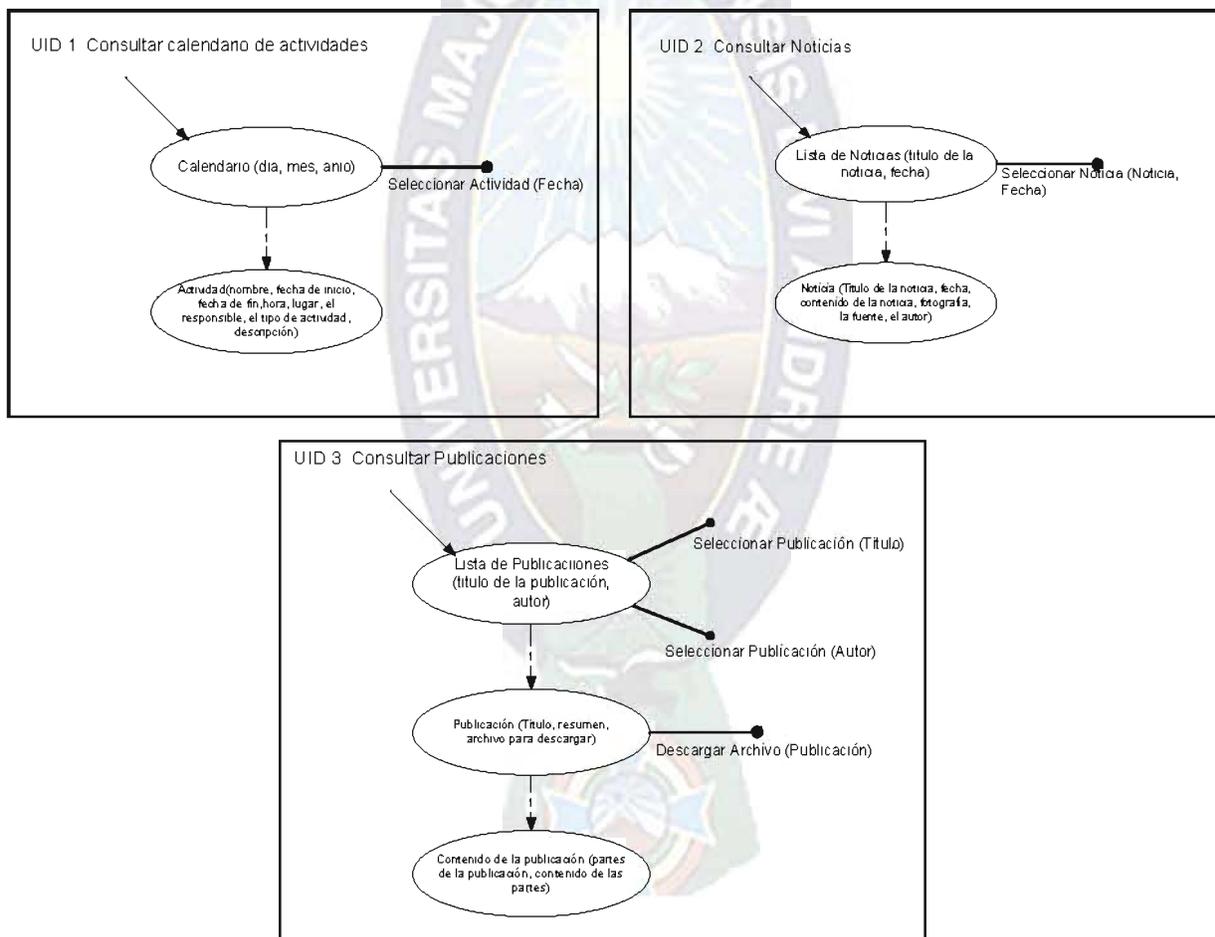
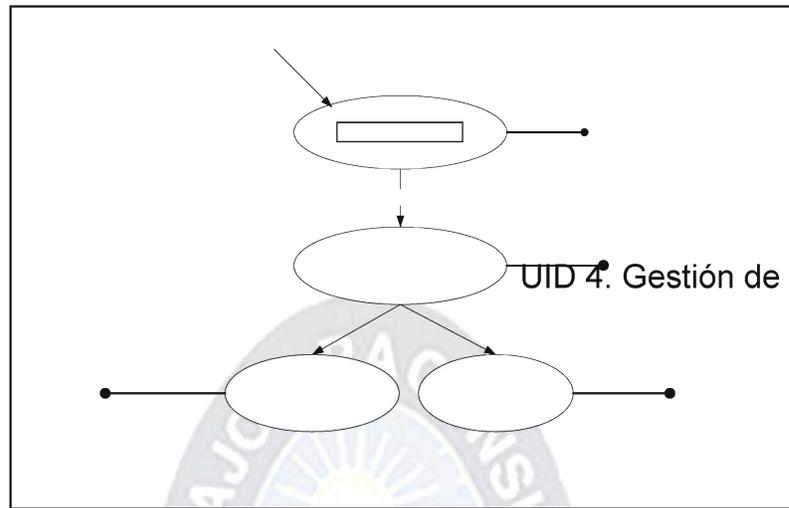
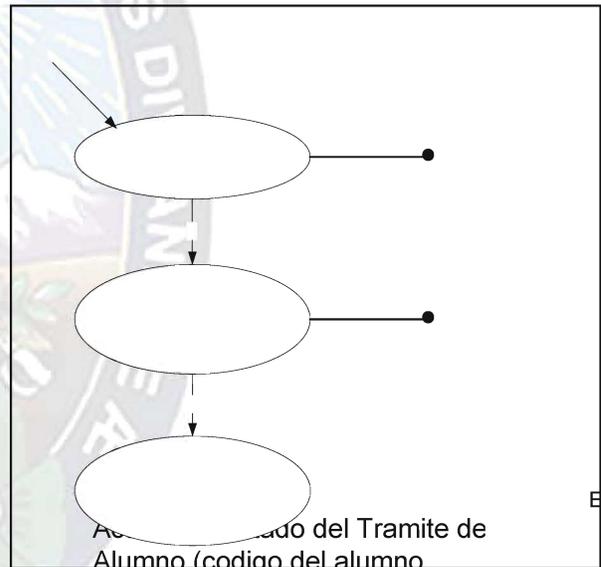
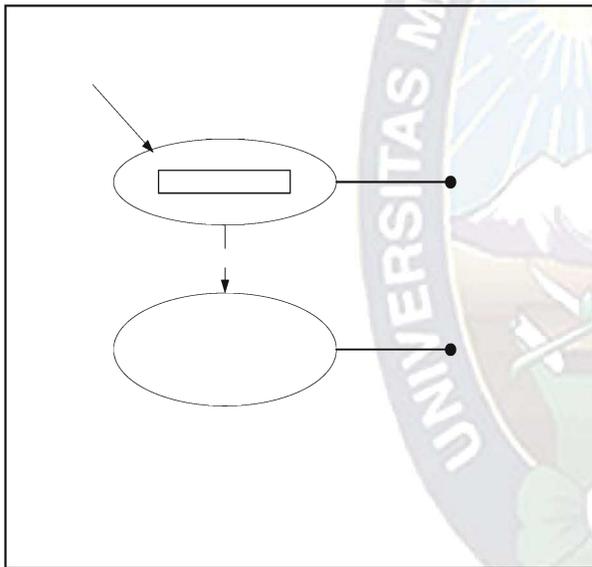


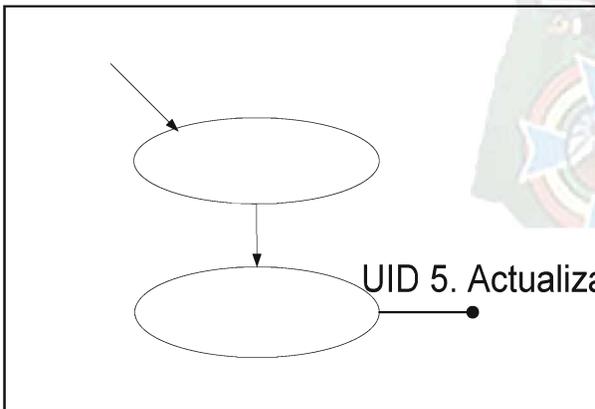
Fig. 3.5. Diagramas de Interacción de Usuarios (continuación)



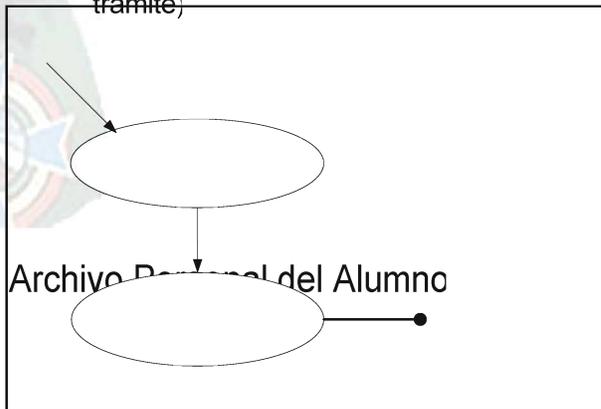
UID 4. Gestión de Trámites del Alumno



Estado del Trámite de Alumno (codigo del alumno, trámite)



UID 5. Actualizar Archivo Personal del Alumno



Cód

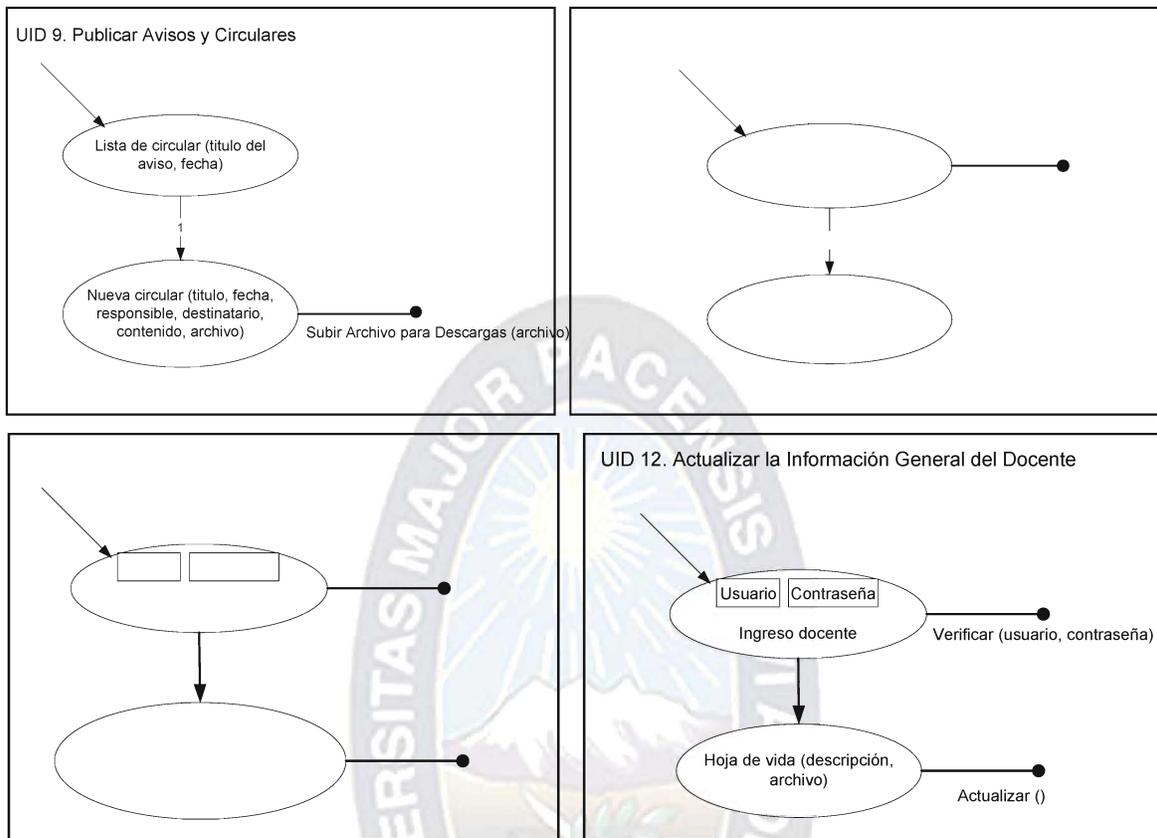
Alumno  
no  
Lista de  
(nombre d

Estado del Trámite

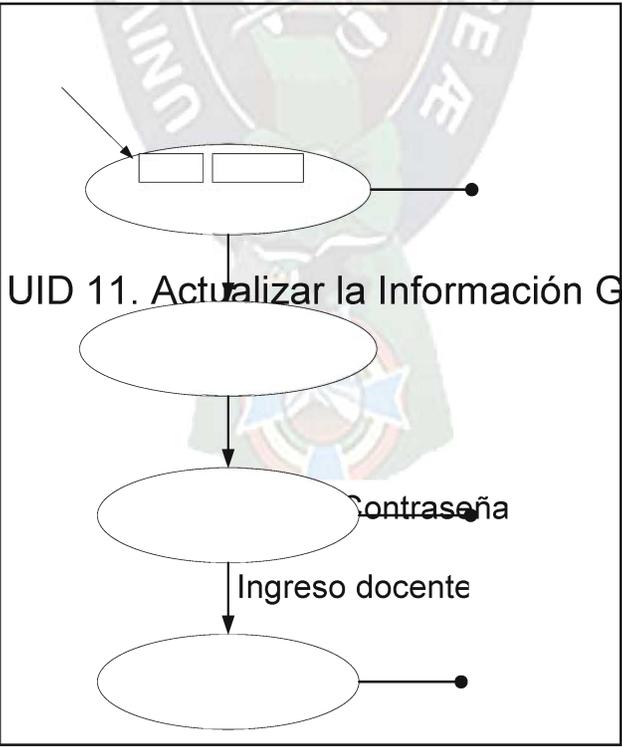
Código del alumno

Busqueda de Alumnos  
(Codigo del Alumno)

**Fig. 3.6. Diagramas de Interacción de Usuarios (continuación)**



**UID 11. Actualizar la Información General del Docente**



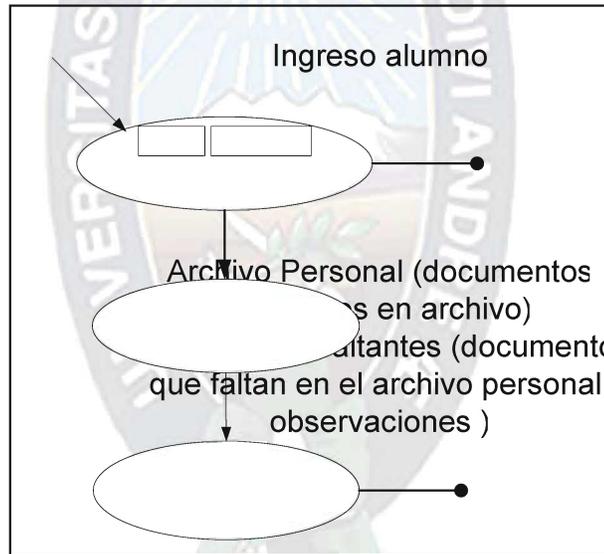
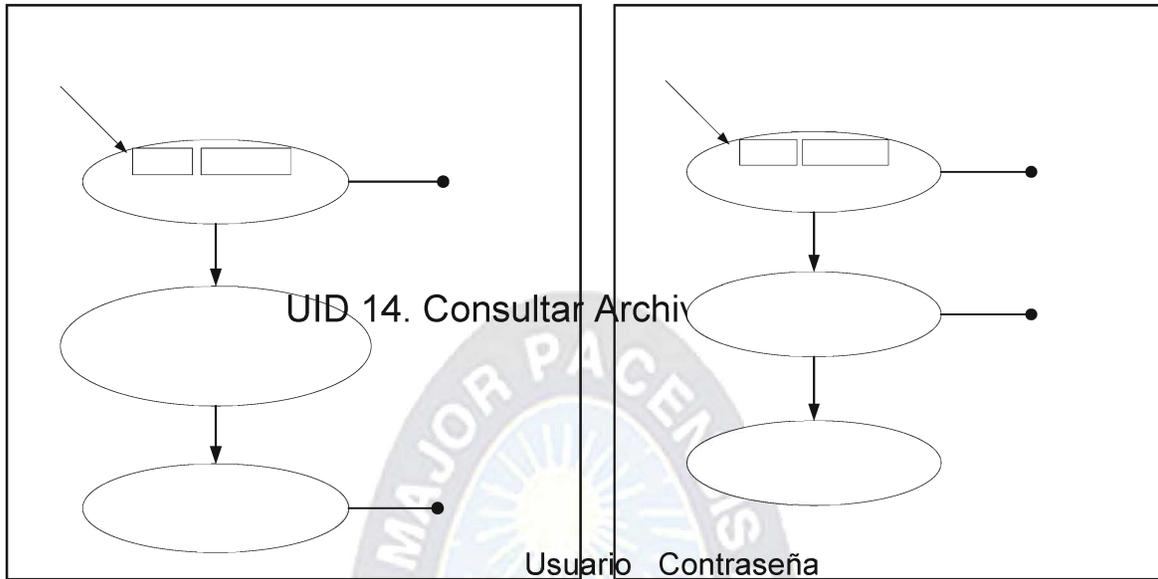
Verificar (usuario, contraseña)

Fuente. Elaboración Propia

Datos Generales (nombre completo, ci, fecha de nacimiento, direccion, teléfono, correo electrónico, fotografía)

Actualizar ()

Fig. 3.7. Diagramas de Interacción de Usuarios (continuación)



Fuente. Elaboración Propia

3.3. DISEÑO CONCEPTUAL

Mensaje de consulta (unidad, mensaje)

Enviar mensaje (unidad, mensaje)

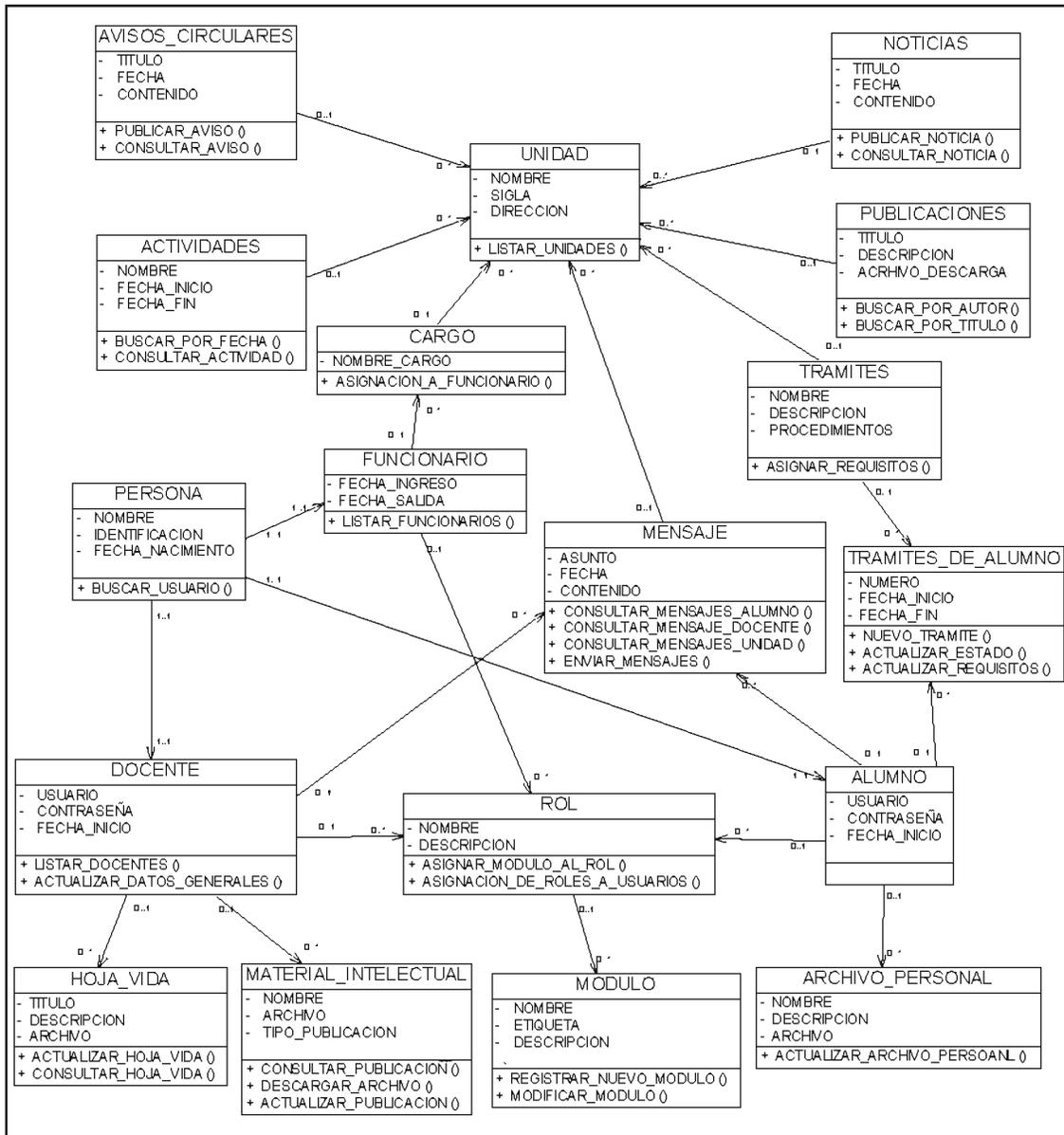
En el diseño conceptual se definen los elementos requeridos para el sistema, de tal forma que estos respondan a la solución del dominio del problema.

UID 16. Enviar mensaje a la

### 3.3.1. Diagrama de Clases

El diagrama que se presenta en la figura 3.8 refleja las necesidades presentadas en la identificación de actores del sistema, identificación de escenarios y casos de uso.

Fig. 3.8. Diagrama de Clases



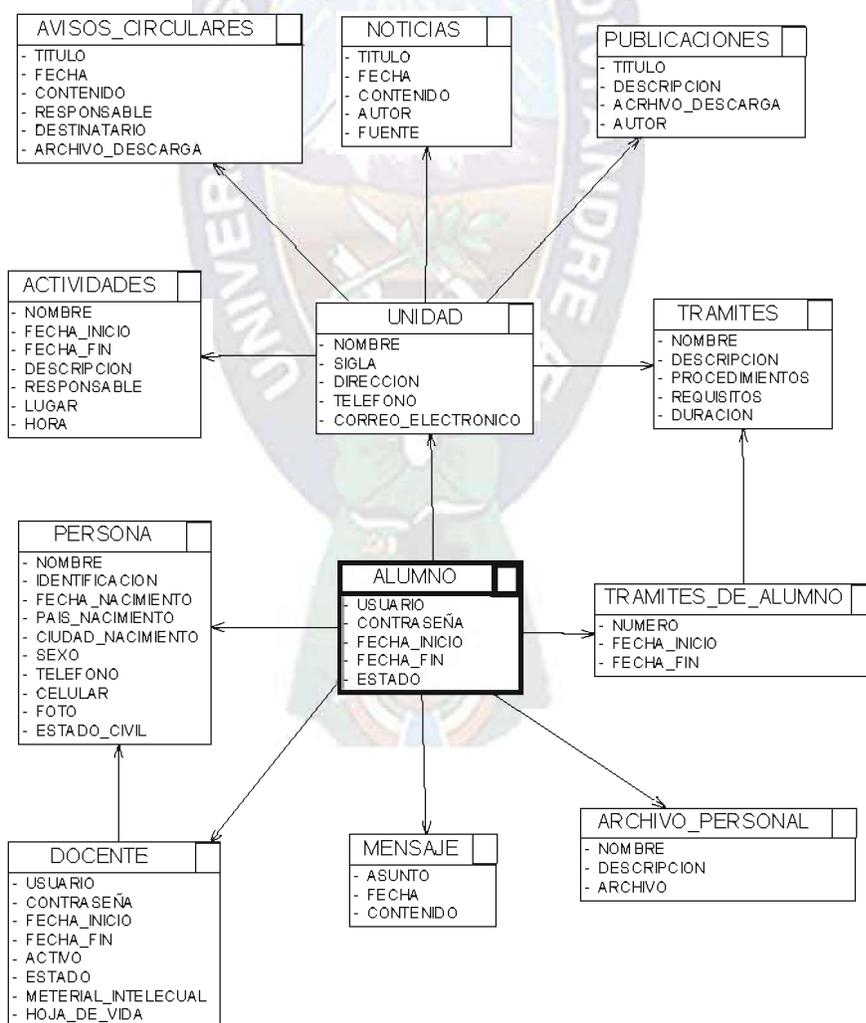
### 3.4. DISEÑO NAVEGACIONAL

El diseño navegacional se especifica para cada actor, cada uno de los diagramas está relacionado con la información a la que tendrá acceso el usuario dentro la aplicación dependiendo del rol que tiene. El sistema tiene como actores: Alumnos, Docentes, Secretaria, Auxiliar de oficina, Administrador de la aplicación

#### 3.4.1. Contexto Navegacional del Alumno

En este contexto se describe la información a la que el alumno tiene acceso para satisfacer sus necesidades, según se observa en la figura 3.9.

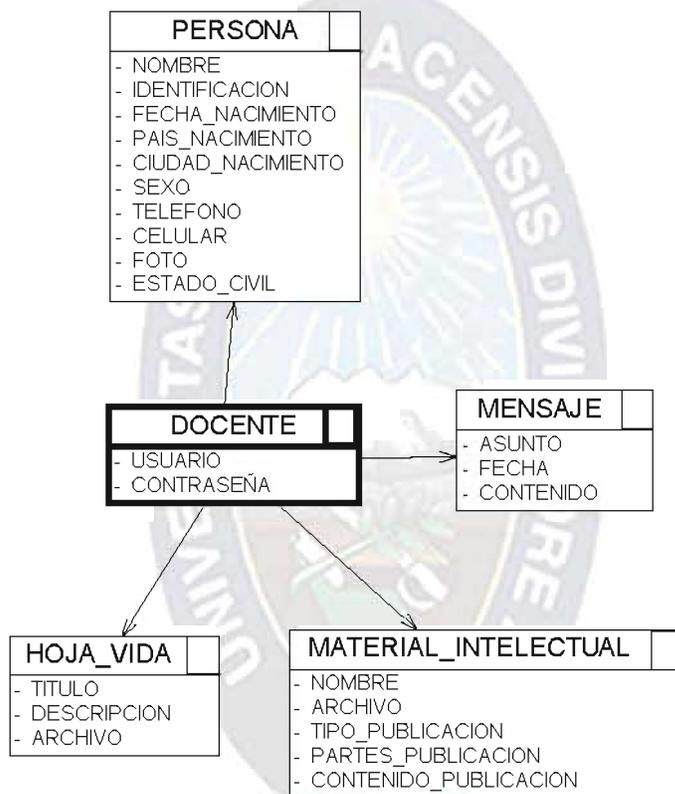
**Fig. 3.9. Diagrama de Contexto Navegacional del Alumno**



### 3.4.2. Contexto del Docente

El docente como tal tiene acceso a su información personal, a su hoja de vida, al material intelectual que requiera publicar y al envío y recepción de mensajes, como se observa en la figura 3.10.

Fig. 3.10. Diagrama de Contexto Navegacional del Docente



### 3.11.3. Contexto de los Funcionarios Administrativos

Los funcionarios administrativos de la Carrera de Informática, tienen acceso a la información que se procesa en cada Unidad dependiendo del rol que se les asigna.

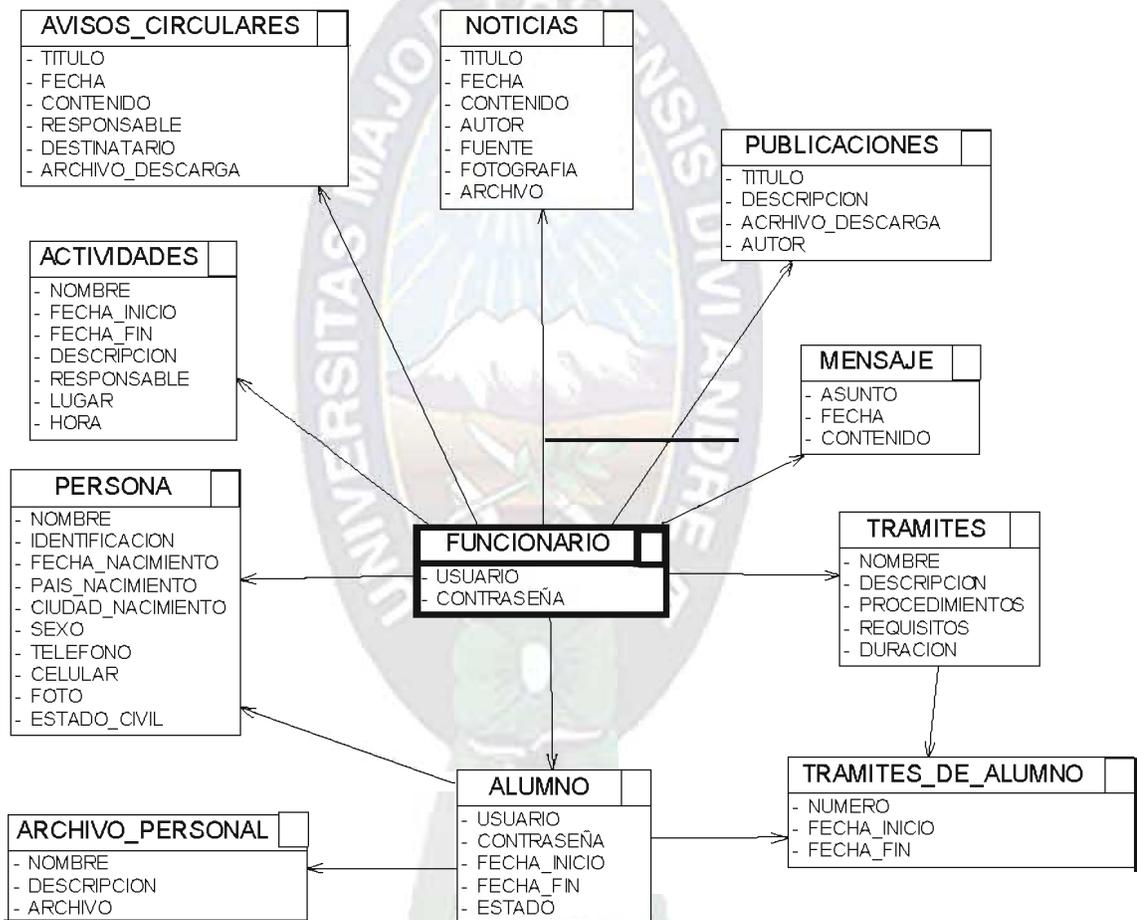
Entre la información a la que tienen acceso se encuentra:

- Publicación de avisos y circulares
- Publicación de Noticias
- Publicación de Actividades en el calendario de actividades

- Envío y recepción de mensajes
- Actualización de la información del alumno: datos generales, archivo personal, trámites del alumno.

La figura 3.11 muestra el contexto del funcionario administrativo y la información a la que tiene acceso.

**Fig. 3.11. Diagrama de Contexto Navegacional del Funcionario Administrativo**



### 3.4.4. Contexto del Administrador de la Aplicación

El administrador de la aplicación puede:

- Crear nuevos roles de usuarios según las necesidades de las unidades
- Registrar y actualizar la información de los módulos que componen el sistema

- Crear nuevas unidades donde cada unidad tiene sus propios servicios.
- Asignar roles individuales a las personas registradas en el sistema

La figura 3.12 detalla la información a la que tiene acceso el administrador de la aplicación.

**Fig. 3.12. Diagrama de Contexto Navegacional del Administrador de la Aplicación**

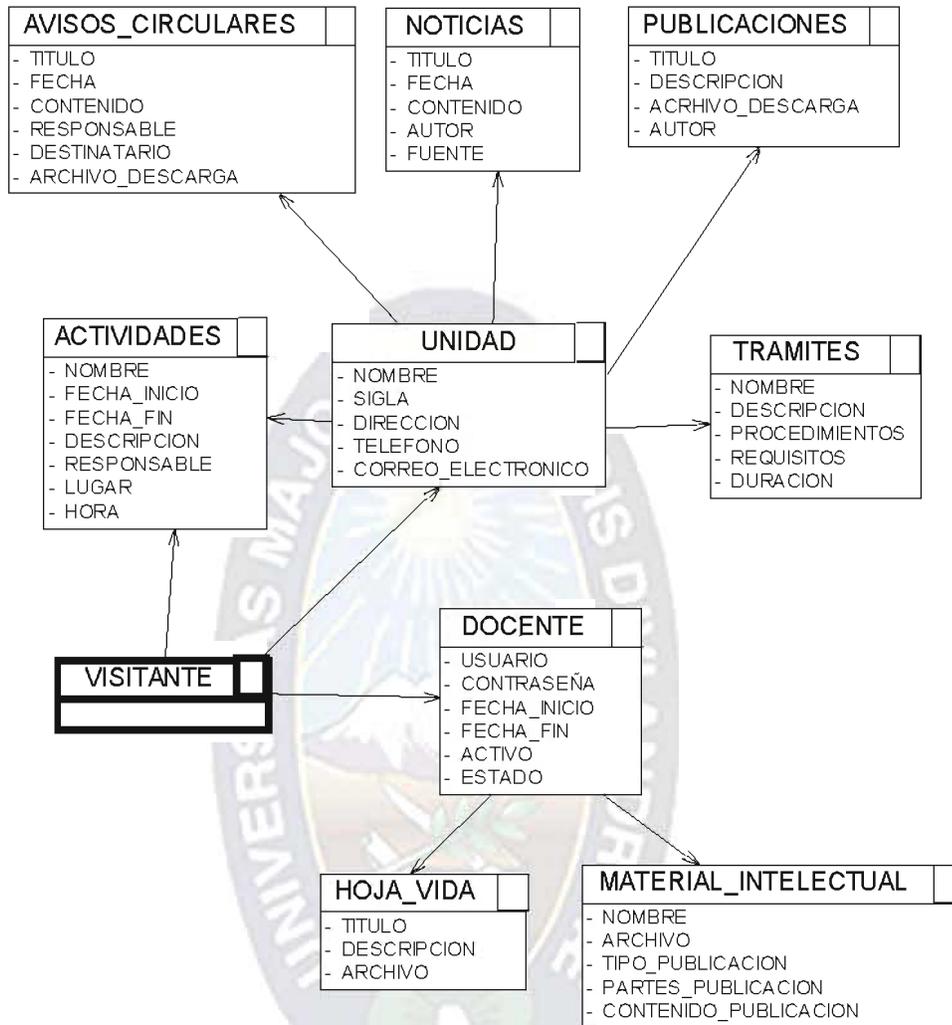


### 3.4.5. Contexto del Usuario Ocasional

Ya que cualquier persona puede ser un usuario ocasional (visitante), este tiene acceso a la información de carácter público de la carrera de Informática.

La figura 3.13 describe claramente la información a la que este usuario tiene acceso.

**Fig. 3.13. Diagrama de Contexto Navegacional del Usuario Ocasional**

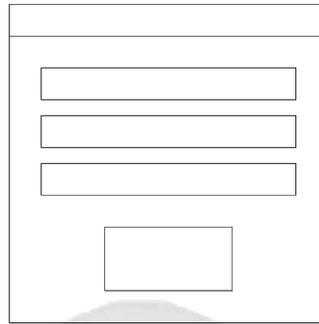


### 3.5. DISEÑO DE LA INTERFAZ ABSTRACTA

#### 3.5.1. Interfaz abstracta de ingreso a la parte restringida de la aplicación

El ADV que se presenta en la figura. 3.14 muestra como accederán los alumnos, docentes y funcionarios administrativos a la aplicación, cada uno de los cuales accede de acuerdo al rol asignado a las interfaces que le correspondan.

**Fig. 3.14. ADV Ingreso de Usuarios Registrados**



The diagram shows a rectangular box representing a login form. It contains four horizontal input fields stacked vertically. The top field is the largest, followed by three smaller fields of equal size. Below these four fields is a single, wider rectangular button.

ADV Ingreso

Lista Tip

Text

Texto

In

### **3.5.2. Interfaz a la Página Personal del Docente**

Al Docente se le presenta el ADV que se muestra en la figura. 3.15 desde la cual puede navegar por las interfaces que le corresponden como usuario docente; tiene acceso a modificar la información irrelevante de sus datos personales, tiene acceso a las herramientas de publicación del material intelectual y puede mandar y recibir mensajes; también puede modificar su hoja de vida.

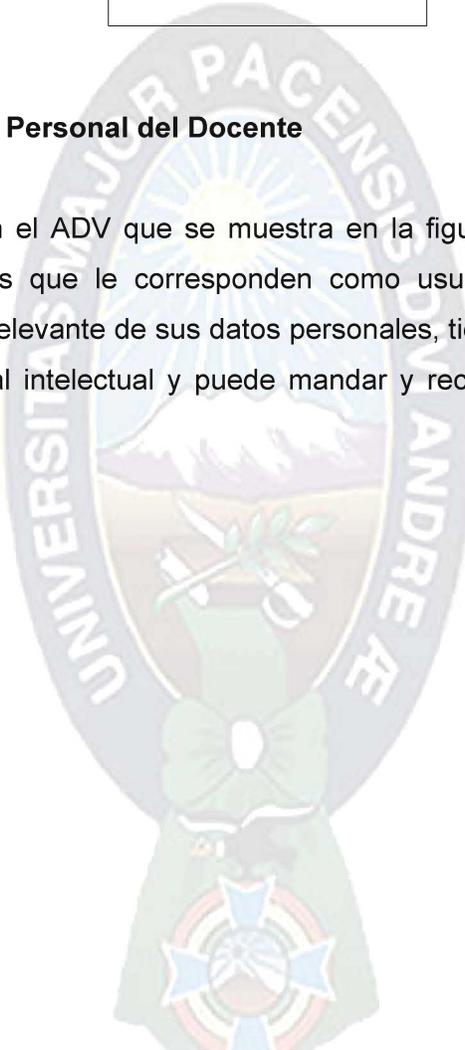
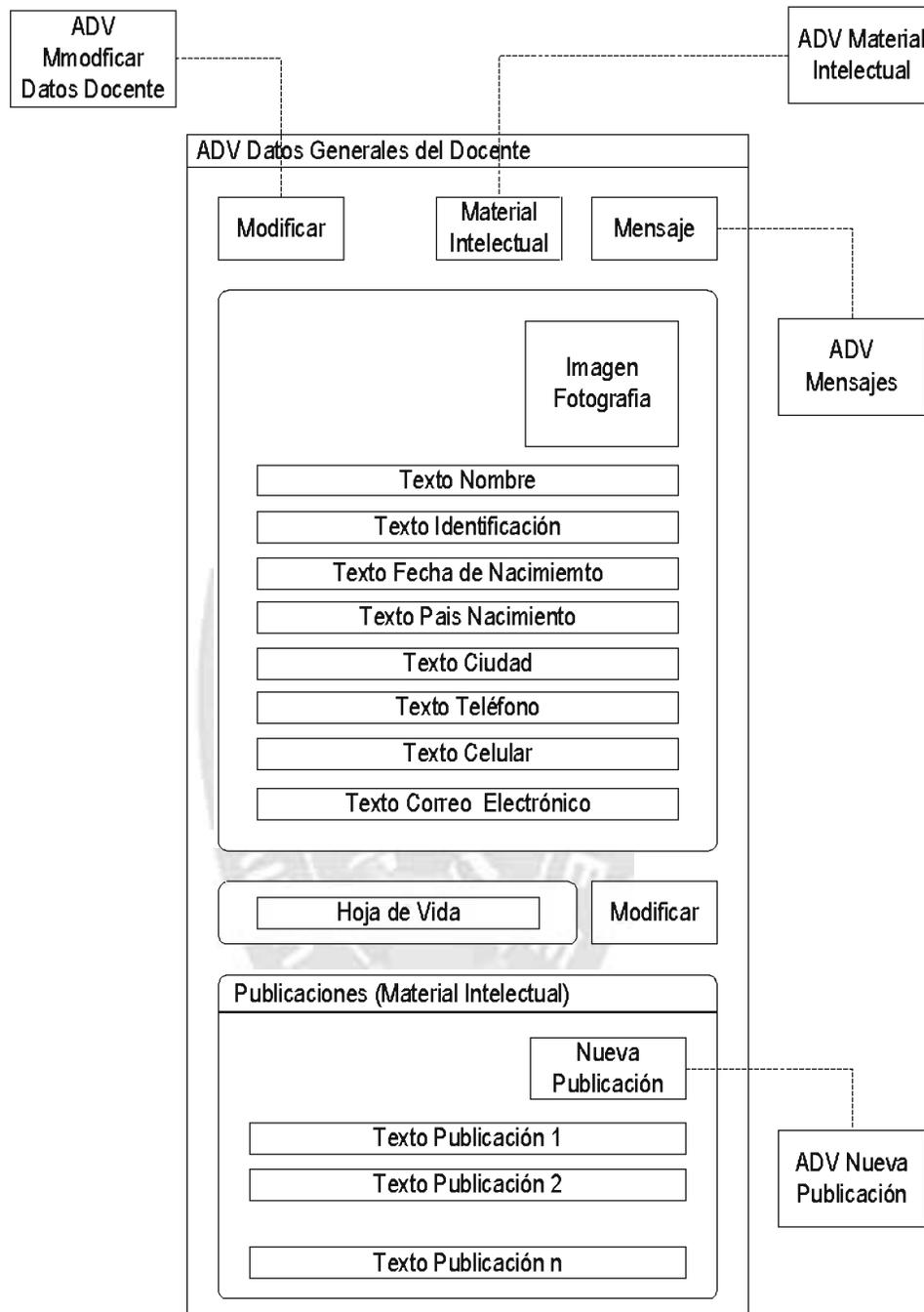


Fig. 3.15. ADV Página Personal del Docente



### 3.5.3. Interfaz Publicar Material Intelectual

Las publicaciones que el docente desee dar a conocer a la comunidad universitaria son publicadas mediante el siguiente ADV, figura. 3.16, donde tiene las opciones de publicación básica y publicación completa (partes y contenidos).

**Fig. 3.16. ADV Publicación del Material Intelectual**

El diagrama muestra un formulario con el título "ADV Publicar Material Intelectual". El formulario contiene los siguientes elementos:

- Campo de texto: "Texto Tipo Publicación"
- Campo de texto: "Texto Nombre"
- Campo de texto: "Texto Fecha"
- Campo de texto: "Texto Archivo"
- Sección de partes y contenidos (encerrada en un recuadro):
  - Campo de texto: "Texto Parte 1" y "Texto Contenido 1"
  - Campo de texto: "Texto Parte 2" y "Texto Contenido 2"
  - Campo de texto: "Texto Parte n" y "Texto Contenido n"
- Botón: "Guardar"

### 3.5.4. Interfaz de Mensajes

Para el envío de mensajes se tiene el ADV mostrado en la figura 3.17, en la cual se describen los datos necesarios para un mensaje.

**Fig. 3.17. ADV Enviar Mensaje**

ADV Mensaje

Enviar

Texto Destinatario

Texto Asunto

Texto Fecha

Texto Contenido

### 3.5.5. Interfaz Menú de Administración de Servicios de la Unidad

El menú con que los usuarios funcionarios de las unidades trabajarán se presenta en la figura 3.18, este menú es generado dinámicamente de acuerdo al rol del usuario.

**Fig. 3.18. ADV Administración de servicios**

ADV Menú Funcionarios

Calendario de Actividades

Noticias

Avisos y Circulares

Lista de Trámites

Publicaciones

...

### 3.5.6. Interfaz de Avisos y Circulares

Para la publicación de avisos y circulares se tiene el ADV mostrado en la figura 3.19, en la cual se describen los datos necesarios para un aviso o circulares.

**Fig. 3.19. ADV Avisos y Circulares**



A schematic diagram of a form titled "ADV Avisos y Circulares". The form is enclosed in a rectangular border. At the top, there is a header bar. Below the header, there are six horizontal input fields of varying lengths, stacked vertically. The bottom-most field is the longest and is positioned above a larger, empty rectangular area. At the very bottom of the form, centered, is a single horizontal input field.

ADV Avisos y Circu

Te

Te

Texto

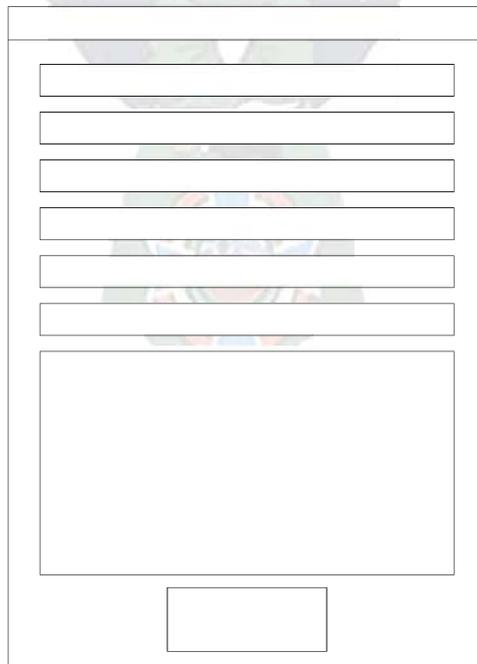
Texto

Texto An

### 3.5.7. Interfaz de Noticias

Para la publicación de las noticias generadas en las unidades se tiene el ADV mostrado en la figura 3.20, en el cual se describen los datos necesarios para una noticia.

**Fig. 3.20. ADV de Noticias**



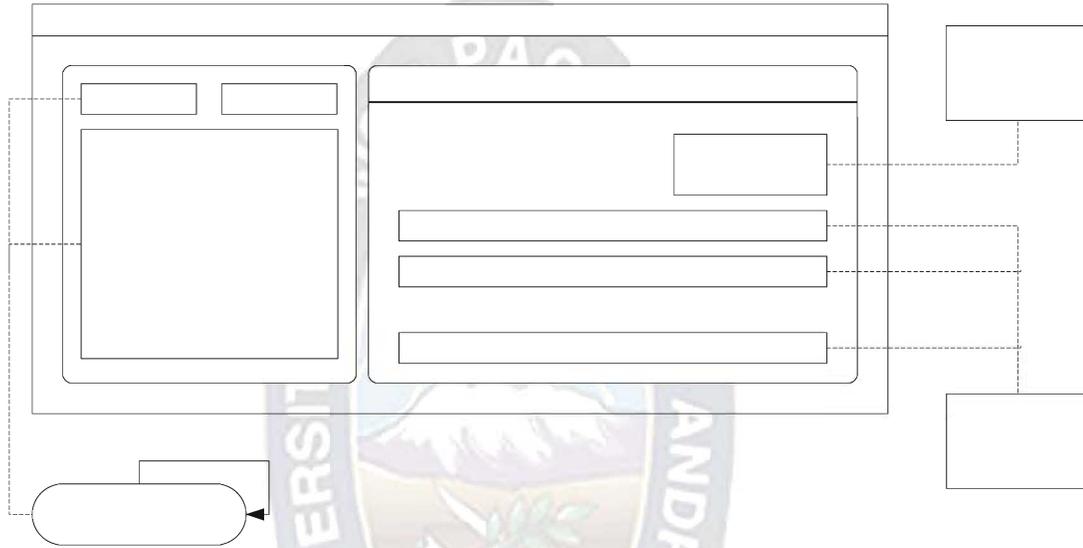
A schematic diagram of a form titled "ADV de Noticias". The form is enclosed in a rectangular border. At the top, there is a header bar. Below the header, there are eight horizontal input fields of varying lengths, stacked vertically. The bottom-most field is the longest and is positioned above a larger, empty rectangular area. At the very bottom of the form, centered, is a single horizontal input field.

Text

### 3.5.8. Interfaz del Calendarios de Actividades

Para la publicación de las actividades generadas en las unidades se tiene el ADV mostrado en la figura 3.21, en la cual se describen los datos necesarios para una actividad.

**Fig. 3.21. ADV de Actividad**



**ADV Calendario de Actividades**

### 3.5.9. Interfaz Página Personal del Alumno

Los alumnos tienen acceso a esta parte mediante el ingreso de los usuarios registrados, es en esta interfaz que el alumno puede monitorear su archivo personal, trámites efectuados en las unidades, mensajes. La interfaz de la pagina personal del alumno se muestra en la figura 3.22.

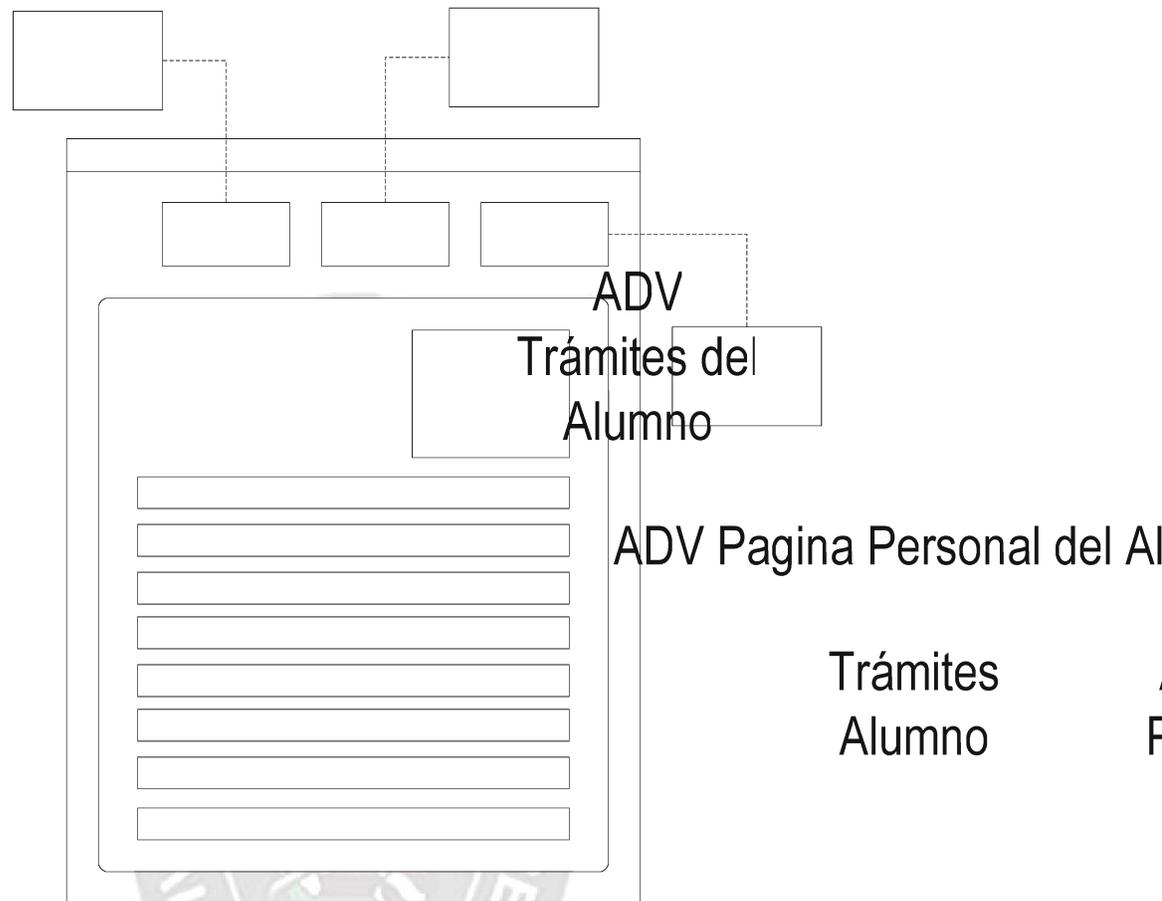
Lista Mes

Lista Año

Actividades

Texto Días del Mes

**Fig. 3.22. ADV Página Personal del Alumno**

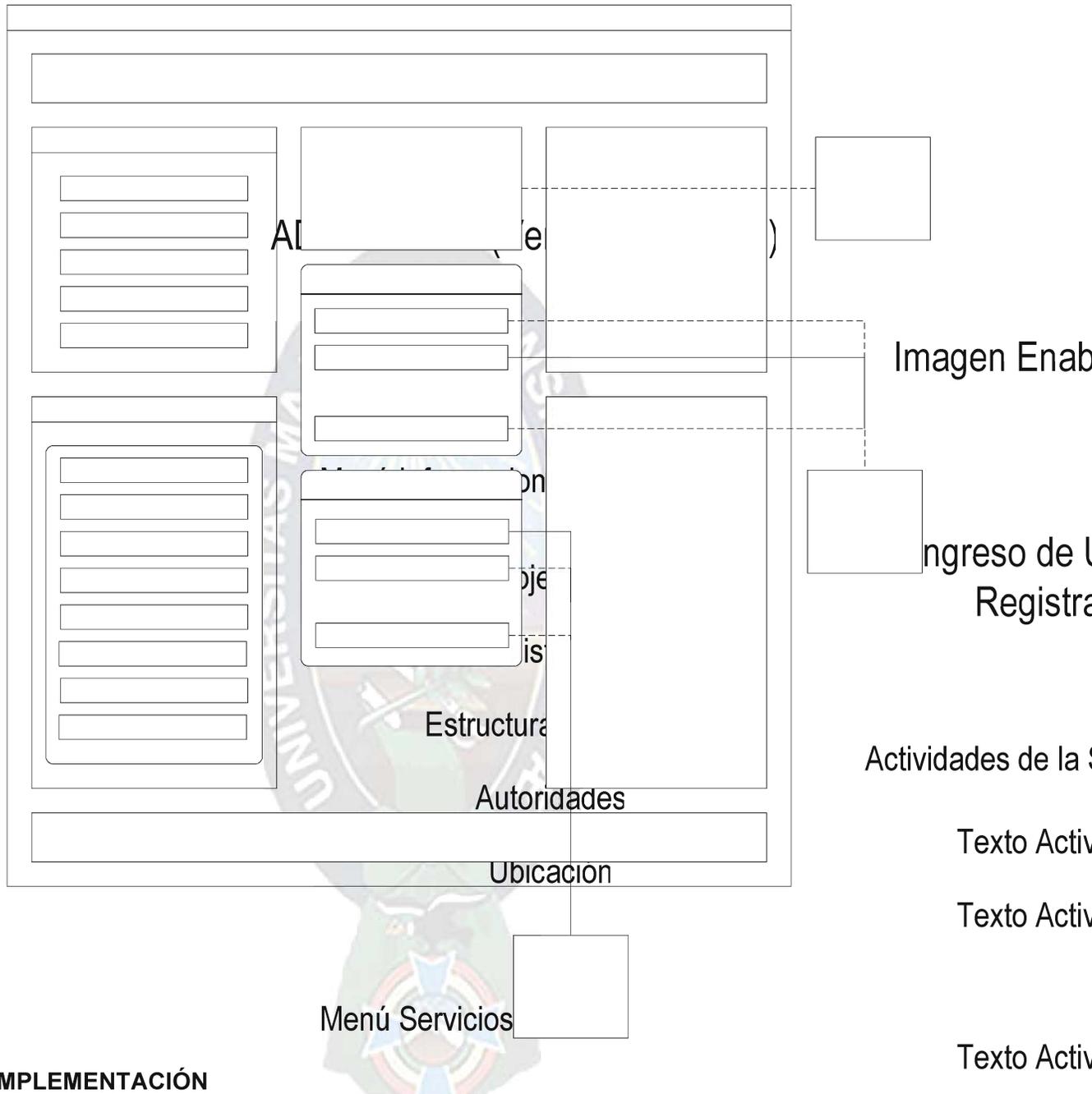


### 3.5.10. Interfaz Página Principal

Esta es la pagina inicial para todos los usuarios ocasionales (visitantes), a partir de esta ventana se tiene acceso a toda la información de carácter público que esta disponible para todos, también se cuenta en esta página con los enlaces para que los usuarios registrados (docentes, alumnos y funcionarios administrativos) puedan acceder a las interfaces restringidas.

La figura 3.23 muestra los detalles de esta página.

Fig. 3.23. ADV Página Principal



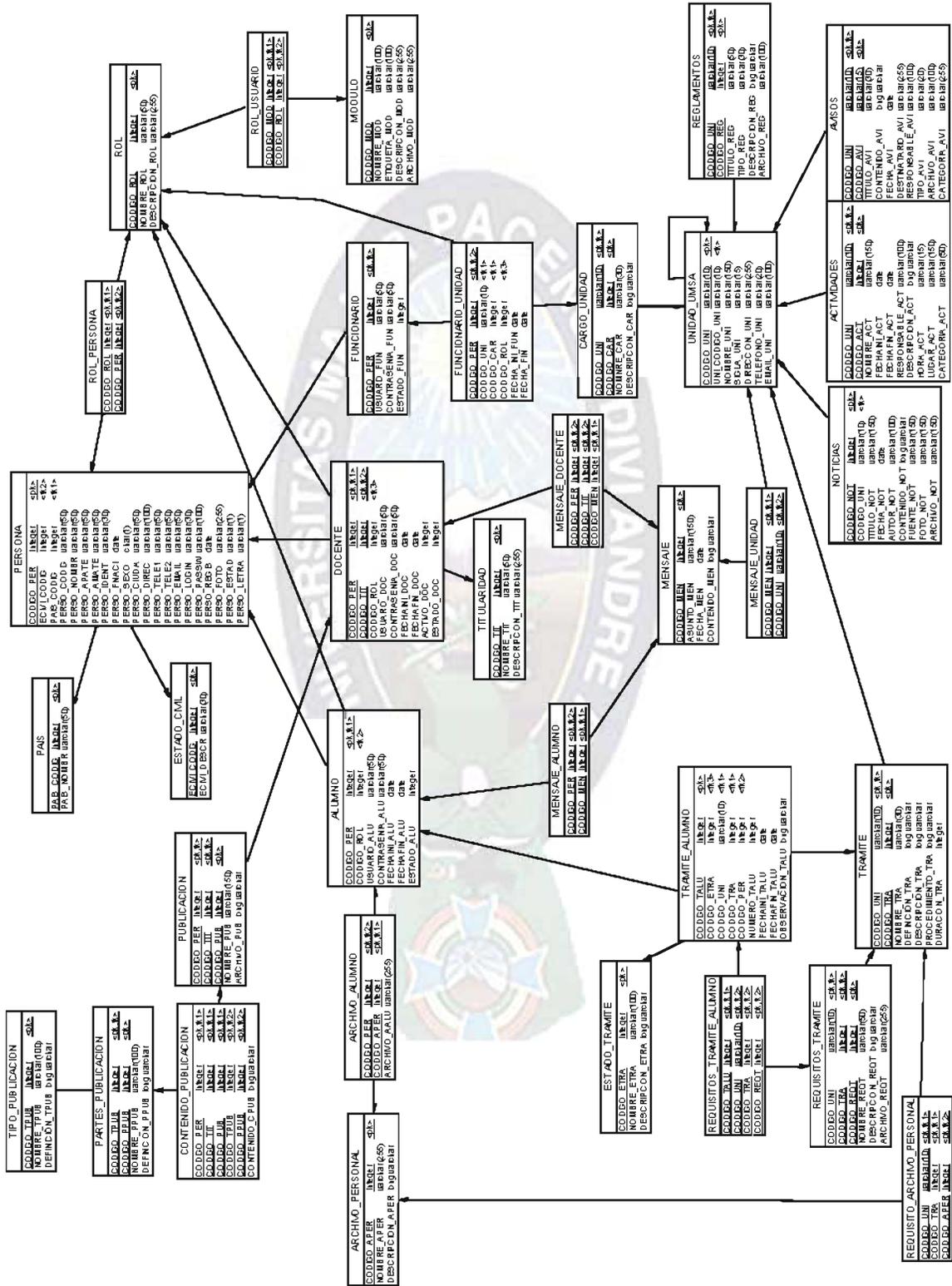
### 3.6. IMPLEMENTACIÓN

En esta etapa se muestra el modelo de datos de la aplicación que refleja el **Calendario de Actividades** Conceptual que se detalla en la figura 3.24, adicionalmente se muestra su aplicación en el gestor de base de datos MySQL presentado en la Tabla.3.1 y por último se presenta las interfaces para los distintos tipos de usuarios. **Noticias del Dia**

**Noticias**  
**Avisos y Circulares**  
**Lista de Trámites**  
**Publicaciones**  
**Unidades**

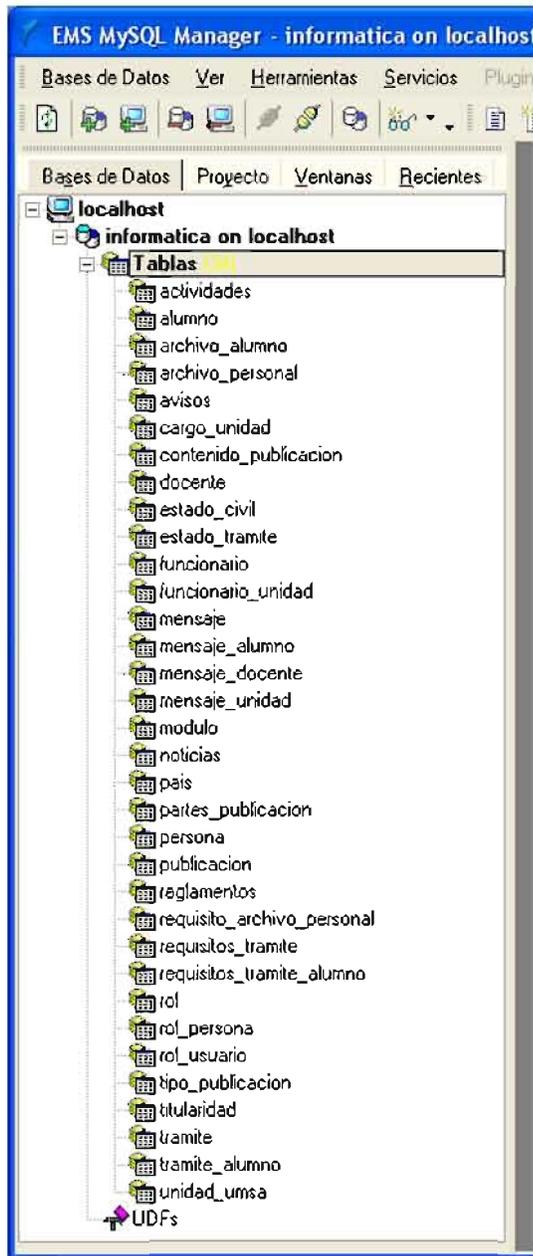
### 3.6.1. Modelo de Datos Relacional

Fig. 3.24. Modelo Relacional del Dominio del Problema



### 3.6.2. Modelo de Datos en MySql

Fig. 3.25. Base de Datos en MySql



### 3.6.3. Interfaces del Sistema

#### 3.6.3.1. Interfaz Principal de acceso a la aplicación

Fig. 3.26. Interfaz Principal de acceso a la aplicación



#### 3.6.3.2. Interfaz de Ingreso a la parte administrativa de la aplicación

Fig. 3.27. Interfaz de ingreso a la parte administrativa de la aplicación



### 3.6.3.2. Interfaz de Ingreso de usuarios registrados

Fig. 3.28. Interfaz de Ingreso de usuarios registrados

Los campos marcados con ! son obligatorios

¿Olvidó su contraseña?

### 3.6.3.3. Interfaz de Roles de usuario

Fig. 3.29. Interfaz de Roles de usuario

Ud. Está Aquí: Principal >> Roles de Usuario

**Roles de Usuario**

No.	Rol
1	<b>ADMINISTRADOR</b>
	<b>Descripción</b> Tiene acceso a todos los módulos de la aplicación, puede hacer alta, bajas y modificaciones
	<b>Módulos</b> - Avisos y Circulares - Cargos de la Unidad - Calendario de Actividades - Noticias - Trámites
2	<b>DOCENTE</b>
	<b>Descripción</b> Permite que los docentes accedan a la aplicación
	<b>Módulos</b> - Módulo 1 - Noticias
3	<b>ALUMNO</b>

Asignación de Módulos

### 3.6.3.4. Interfaz de Módulos del sistema

Fig. 3.30. Interfaz de Módulos del sistema

The screenshot shows the 'Módulos del Sistema' interface. At the top, it displays the UMSA logo and 'CARRERA DE INFORMÁTICA'. The breadcrumb trail is 'Ud. Está Aquí: Principal >> Módulos del Sistema'. On the left sidebar, there are navigation options: 'Roles de Usuario', 'Módulos del Sistema' (selected), 'Unidades', 'Buscar Usuario', and 'Salir'. The main content area is titled 'Módulos del Sistema' and contains a table with the following data:

No.	Módulo		
1	<b>Avisos y Circulares</b>		
	<b>Descripción</b> Avisos y circulares de la carrera de Informatica		
	<b>Etiqueta</b> Avisos y Circulares		
	<b>Archivo</b> ../Avisos_Admin/Principal.php		
2	<b>Cargos de la Unidad</b>		
	<b>Descripción</b> Cargos del organigrama de la unidad		
	<b>Etiqueta</b> Cargos de la Unidad		
	<b>Archivo</b> ../Cargos_Admin/Principal.php		
3	<b>Calendario de Actividades</b>		
	<b>Descripción</b> Calendario de Actividades de la carrera de Informatica		
	<b>Etiqueta</b> Calendario de Actividades		
	<b>Archivo</b> ../Calendario_Admin/Principal_Calendario.php		
4	<b>Modulo 1</b>		

### 3.6.3.5. Interfaz de Unidades del sistema

Fig. 3.31. Interfaz de Unidades del sistema

The screenshot shows the 'Unidades' interface. At the top, it displays the UMSA logo and 'CARRERA DE INFORMÁTICA'. The breadcrumb trail is 'Ud. Está Aquí: Principal >> Unidades'. On the left sidebar, there are navigation options: 'Roles de Usuario', 'Módulos del Sistema', 'Unidades' (selected), 'Buscar Usuario', and 'Salir'. The main content area is titled 'Unidades' and contains a table with the following data:

No.	DIVISION		
1	<b>Dirección de Carrera</b>		
2	Laboratorio Superior de Informatica		
3	Sub Dirección Academica		

Below the table, there are four buttons: 'Adicionar Nuevo Registro', 'Modificar Datos del Registro', 'Eliminar Registro', and 'Detalles del Registro'. At the bottom of the page, there is a footer with 'Mapa del Sitio', 'Contactos', and copyright information: '©2006 La Paz - Bolivia Av. Villazon N°1995 Monoblock Central Tel. 591-2352232'.

### 3.6.3.7. Otras Interfaces del sistema

Otras interfaces del sistema que son importantes para la operación del mismo son:

- Calendario de actividades
- Normas y reglamentos
- Lista de avisos y circulares
- Lista de alumnos
- Tramites de los alumnos
- Archivo personal
- Publicaciones
- Cargos de la Unidad
- Noticias
- Información Institucional
- Comentarios y sugerencias

### 3.7. MÉTRICAS DE CALIDAD

Las métricas de calidad, contemplan atributos y características deseables, observables y cuantificables en el desarrollo de aplicaciones Web; La métrica de calidad Web-Sitie QEM, parte de un modelo jerárquico de calidad de producto basado en el estándar ISO 9126-1. Es decir, la calidad de producto queda definida a un alto nivel de abstracciones por las características denominadas: usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad y mantenibilidad. Esta métrica recomienda la definición de metas de evaluación y selección del perfil de usuario. Se ha identificado los perfiles de usuarios que interactúan con el sistema, y para realizar las pruebas de calidad, se toma en cuenta éstos perfiles de usuario, y los atributos del estándar ISO 9126-1.

**Tabla 5.1 Atributos generales de Calidad en Web-site QeM**

<p><b>1. Usabilidad</b></p> <p>1.1 Comprensibilidad Global del Sitio</p> <p>1.1.1 Esquema de Organización Global</p> <p>1.1.1.1 <i>Mapa del Sitio</i></p> <p>1.1.1.2 <i>Calidad en el Sistema de Etiquetado</i></p> <p>1.1.1.3 <i>Visita Guiada Orientada al Usuario</i></p> <p>1.1.1.4 <i>Mapa de Imagen (Campus/Edificio)</i></p> <p>1.2 Mecanismos de Ayuda</p> <p>1.2.1 Global de todo el sitio</p> <p>1.2.1.1 <i>Ayuda explicatoria orientada al usuario</i></p> <p>1.2.2 Directorio de Direcciones</p> <p>1.2.2.1 <i>Directorio E-mail</i></p> <p>1.2.3 Retroalimentación</p> <p>1.2.3.1 <i>Comentarios/Sugerencias</i></p> <p>1.3 Aspectos de Interfaces y Estéticos</p> <p>1.3.1.1. <i>Cohesividad al Agrupar los Objetos de</i></p>	<p><b>3. Confiabilidad</b></p> <p>3.1 Eficiencia</p> <p>3.1.1 Errores de Enlaces</p> <p>3.1.1.1 <i>Enlaces Rotos</i></p> <p>3.1.1.2 <i>Enlaces Inválidos</i></p> <p>3.1.1.3 <i>Enlaces no Implementados</i></p> <p>3.1.2 Errores o Deficiencias Varias</p> <p>3.1.2.1 <i>Deficiencias o cualidades ausentes debido a diferentes navegadores (browsers)</i></p> <p>3.1.2.2 <i>Deficiencias o resultados inesperados independientemente del browsers</i></p> <p>3.1.2.3 <i>Nodos Destinos (inesperadamente) en Construcción</i></p> <p>3.1.2.4 <i>Nodos Web Muertos (sin enlaces de retorno)</i></p>
--	--

<p><i>Control Principales</i></p> <p>1.3.2 Permanencia y Estabilidad en la Presentación de los Controles Principales</p> <p>1.3.2.1 <i>Permanencia de Controles Directos</i></p> <p>1.3.2.2 <i>Permanencia de Controles Indirectos</i></p> <p>1.3.2.3 <i>Estabilidad</i></p> <p>1.3.3 Aspectos de Estilo</p> <p>1.3.3.1 <i>Uniformidad en el Color de Enlaces</i></p> <p>1.3.3.2 <i>Uniformidad en el Estilo Global</i></p> <p>1.3.3.3 <i>Guía de Estilo Global</i></p> <p>1.3.4 Preferencia Estética</p>	
<p><b>2. Funcionalidad</b></p> <p>2.1 Aspectos de Búsqueda y Recuperación</p> <p>2.1.1 Mecanismo de Búsqueda en el Sitio Web</p> <p>2.1.1.1 <i>Búsqueda de Usuarios</i></p> <p>2.1.1.2 <i>Búsqueda Global</i></p> <p>2.1.2 Mecanismos de Recuperación</p> <p>2.1.2.1 <i>Nivel de Personalización</i></p> <p>2.1.2.2 <i>Nivel de Retroalimentación</i></p> <p>2.2 Aspectos de Navegación y Exploración</p> <p>2.2.1 Navegabilidad</p> <p>2.2.1.1 Orientación</p> <p>2.2.1.1.1 <i>Indicador de Caminos</i></p> <p>2.2.1.1.2 <i>Etiqueta de la Posición Actual</i></p> <p>2.2.1.2 <i>Promedio de Enlaces por Página</i></p> <p>2.2.2 Objetos de Control Navegacional</p> <p>2.2.2.1 Permanencia y Estabilidad en la Presentación de los Controles Contextuales</p> <p>2.2.2.1.1 <i>Permanencia de los Controles Contextuales</i></p> <p>2.2.2.1.2 <i>Estabilidad</i></p> <p>2.2.2.2 Nivel de Desplazamiento</p> <p>2.2.2.2.1 <i>Desplazamiento Vertical</i></p> <p>2.2.2.2.2 <i>Desplazamiento Horizontal</i></p> <p>2.2.3 Predicción Navegacional</p> <p>2.2.3.1 <i>Enlace con Título</i></p> <p>2.2.3.2 <i>Calidad de la frase del enlace</i></p> <p>2.3 Aspectos del Dominio orientados al usuario</p> <p>2.3.1 Relevancia de Contenido</p> <p>2.3.1.1 Información institucional</p> <p>2.3.1.1.1 <i>Historia y situación actual</i></p> <p>2.3.1.1.2 <i>Perfil del profesional</i></p> <p>2.3.1.2 Calendario anual de actividades</p> <p>2.3.1.2.1 <i>Programa operativo anual</i></p> <p>2.3.1.3 Normas y reglamentos</p> <p>2.3.1.3.1 <i>Normas docentes</i></p> <p>2.3.1.3.2 <i>Régimen estudiantil</i></p> <p>2.3.1.4 Publicaciones</p> <p>2.3.1.4.1 <i>Artículos y libros docentes</i></p> <p>2.3.1.5 Noticias de importancia</p> <p>2.3.1.5.1 <i>Noticias académicas</i></p> <p>2.3.1.5.2 <i>Noticias de la comunidad Informática</i></p> <p>2.3.1.6 Trámites</p> <p>2.3.1.6.1 <i>Listado de trámites</i></p> <p>2.3.1.6.2 <i>Trámites ordenados cronológicamente</i></p>	<p><b>4. Eficiencia</b></p> <p>4.1 Rendimiento</p> <p>4.1.1 <i>Páginas de Acceso Rápido</i></p> <p>4.2 Accesibilidad</p> <p>4.2.1 Accesibilidad de Información</p> <p>4.2.1.1 Legibilidad al desactivar la Propiedad Imagen del Browser</p> <p>4.2.1.1.1 <i>Imagen con Título</i></p> <p>4.2.1.1.2 <i>Legibilidad Global</i></p> <p>4.2.2 Accesibilidad de Ventanas</p> <p>4.2.2.1 <i>Número de Vistas considerando Marcos</i></p> <p>4.2.2.2 <i>Versión sin Marcos</i></p>

2.3.1.7 Requisitos de los tramites 2.3.1.7.1 <i>Lista de los requisitos</i> 2.3.1.7.2 <i>Forma de presentación</i> 2.3.1.8 Estado de los tramites 2.3.1.8.1 <i>Ruta del trámite con fecha y hora</i> 2.3.1.8.2 <i>Porcentaje faltante</i> 2.3.1.9 Archivos personales 2.3.1.9.1 <i>Archivo personal del docente</i> 2.3.1.9.2 <i>Archivo personal del estudiante</i> 2.3.2 Servicios On-line 2.3.2.1 <i>Información institucional</i> 2.3.2.2 <i>Información calendario de actividades</i> 2.3.2.3 <i>Información de normas y reglamentos</i> 2.3.2.4 <i>Información de publicaciones</i> 2.3.2.5 <i>Información de noticias</i> 2.3.2.6 <i>Información de tramites</i>	
--	--

### 3.8. DEFINICIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN

A partir de la lista de requerimientos, para cada atributo cuantificable  $A_i$  se debe asociar y determinar una variable  $X_i$ , que tomará un valor real a partir de un proceso de medición. Además, para cada variable  $X_i$  calculada, por medio de un criterio elemental, se producirá una preferencia elemental  $IE_i$  que toma valores reales en porcentaje de 1 a 100.

La tabla 5.2 muestra valores de los criterios elementales para cada uno de los atributos de la característica de Usabilidad.

**Tabla 5.2 Resultado de las preferencias elementales de usabilidad**

Código	Atributo	Definición	Criterio Elemental	IE <sub>i</sub> (%)
1.1.1.1	Mapa del sitio	El mapa del sitio constituye la representación con componentes gráficos, que muestra la estructura o arquitectura global de la aplicación Web. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio binario, discreto y absoluto.	100
1.1.1.2.	Calidad en el sistema de etiquetado	Hace referencia a la calidad presente en el sistema de etiquetado asociado a la aplicación Web. Se pregunta si tiene calidad (1) o si no tiene calidad (0).	Es un criterio binario, discreto y absoluto.	100
1.1.1.3.	Visita guiada orientada al usuario	Se refiere a la capacidad de la aplicación de proporcionar los elementos básicos para guiar al usuario en un	Es un criterio binario, discreto y absoluto.	100

		recorrido por la aplicación y sus vínculos. Se pregunta si está disponible (1) o si no está disponible (0).		
1.1.1.4.	Mapa de la imagen	Permite conocer de manera grafica la aplicación. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio binario, discreto y absoluto.	100
1.2.1.1.	Ayuda explicatoria orientada al usuario	Se refiere a mensajes de aclaración breve del significado de un enlace o icono. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio binario, discreto y absoluto.	100
1.2.2.1.	Directorio e-mail	Hace referencia a los contactos potenciales de la aplicación Web. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio binario, discreto y absoluto.	100
1.2.3.1.	Comentarios/Sugerencias	Son las opiniones y sugerencias de los usuarios, respecto de los servicios que ofrece la aplicación Web. Se pregunta si está disponible esta opción (1) o si no está disponible la opción (0).	Es un criterio binario, discreto y absoluto.	100
1.3.1.1.	Cohesividad al agrupar los objetos de control principales	Indica si los enlaces que posee el sitio Web están agrupados adecuadamente. Se pregunta si lo están (1) o si no lo están (0).	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 la navegación. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto.	100
1.3.2.2.	Permanencia de los controles indirectos.	Es un control de referencia a la página principal. Se pregunta si está disponible (1) o si no está 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 de la aplicación Web. Un control es estable si se encuentra siempre en la misma ubicación al navegar por los nodos. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto	100
1.3.3.1.	Uniformidad en el color	Se refiere a que todos los	Es un criterio, binario,	100

	de enlaces	enlaces de la aplicación Web son de un mismo color. Se emplea la experiencia del evaluador para ver si todos los enlaces son de un mismo color (1) o si son de colores diferentes (0).	discreto, absoluto	
1.3.3.2.	Uniformidad en el estilo global.	Se refiere a que si toda la aplicación Web posee uniformidad en el color, estilo y fuente. X=0 si no posee una uniformidad de estilo, X=1 si posee una uniformidad parcial de estilo y X=2 si posee una uniformidad de estilo total.	Multinivel, discreto absoluto	100
1.3.3.3.	Guía de estilo global	Es la guía de estándares usados en el estilo y diseño de las páginas. Se emplea la experiencia del evaluador para ver si el contenido de las páginas tienen un mismo estilo (1) o si son de estilos diferentes (0).	Es un criterio, binario, discreto, absoluto	100

**Tabla 5.3 Resultados de la evaluación elemental de funcionalidad**

Código	Atributo	Definición	Criterio elemental	IEi (%)
2.1.1.1.1	Búsqueda de usuarios	Proporciona a la persona autorizada la posibilidad de hacer consultas en la base de datos, realizando búsquedas por nombre o código. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto	100
2.1.1.2.	Búsqueda global	Búsqueda global en la aplicación, posible de ser realizada por el administrador del sistema. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto	0
2.1.2.1.	Nivel de Personalización	Es la medida en que el usuario puede modificar la apariencia de las páginas de la aplicación Web, Restringida a los usuarios estudiantes y ocasionales. X=0 si no posee una uniformidad de estilo, X=1 si posee una uniformidad parcial de estilo y X=2 si posee una uniformidad de estilo total.	Multinivel-discreto absoluto	50

2.1.2.2.	Nivel de Retroalimentación	Es la reutilización de los datos que son introducidos mediante los formularios, disponibles para el personal que administra el sistema. X=0 si el nivel es bajo, X=1 si el nivel es medio y X=2 si el nivel es alto.	Multinivel-discreto absoluto	50
2.2.1.1.1	Indicador de caminos	Son indicadores de ubicación que especifican la localización del usuario mientras está navegando en el sitio. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto	100
2.2.1.1.2	Etiqueta de la posición actual	Es la constancia de que el nombre o etiqueta representa la correcta ubicación de la posición actual. Se pregunta si está disponible (1) o si no está disponible (0).	Es un criterio, binario, discreto, absoluto	100
2.2.1.2.	Promedio de enlaces por página	Proporciona una idea sobre la cantidad de puntos de partida en una página promedio que indica cómo una página está interconectada hacia nodos destino. En promedio se cuenta con 25 enlaces. X=0 si el promedio es pequeño, X=1 si el promedio es mediano y X=2 si el promedio es alto.	Multinivel- discreto absoluto	75
2.2.2.1.1	Permanencia de los controles contextuales	La disposición de los controles correspondientes a la página actual, disponibles para la utilización del usuario. Se pregunta si permanece (1) o si no permanece (0).	Es un criterio, binario, discreto, absoluto	100
2.2.2.1.2	Estabilidad	Se refiere a la capacidad contextual de permanencia en la navegación del usuario sin interrupción. X=0 si no posee estabilidad, X=1 si posee una estabilidad parcial y X=2 si posee una estabilidad total.	Multinivel- discreto absoluto	100
2.2.2.2.1	Desplazamiento vertical	Nivel de desplazamiento vertical que el visitante debe realizar para ajustar la interfase, considerando 800 x 600 como la mínima resolución. Se pregunta si es adecuada para la	Es un criterio, binario, discreto, absoluto	100

		navegación (1) o si no es adecuada (0).		
2.2.2.2.2	Desplazamiento horizontal	Nivel de desplazamiento horizontal que el visitante debe realizar para ajustar la interfase, considerando 800 x 600 como la mínima resolución. Se pregunta si es adecuada para la navegación (1) o si no es adecuada (0).	Es un criterio, binario, discreto, absoluto	100
2.2.3.1.	Enlaces con título	Este atributo trata de explicar a través de la etiqueta asociada, los temas o contenidos que están relacionados con el enlace. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.2.3.2	Calidad de la frase del enlace	Se refiere al nemónico que expresa de manera adecuada el contenido del enlace. X=0 si no es adecuado, X=1 si es parcialmente adecuado y X=2 si es totalmente adecuado.	Multinivel- discreto absoluto	100
2.3.1.1.1	Historia y situación actual	Información contenida en el legajo institucional de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.1.2	Perfil del profesional	Información contenida en el legajo institucional de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.2.1	Programa operativo anual	Información contenida en el calendario anual de actividades de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.3.1	Normas docentes	Información contenida en el legajo de Normas y Reglamentos de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.3.2	Régimen estudiantil	Información contenida en el legajo de Normas y Reglamentos de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.4.1	Artículos y libros docentes	Información contenida en la lista de publicaciones del Docente de la Carrera de	Es un criterio binario, discreto y absoluto.	100

		Informática. Se pregunta si existe (1) o si no existe (0).		
2.3.1.5.1	Noticias académicas	Información contenida en la lista de noticias de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.5.2	Noticias de la comunidad Informática	Información contenida en la lista de noticias de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.6.1	Listado de tramites	Información contenida en la lista de trámites de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.6.2	Tramites ordenados cronologicamente	Información contenida en la lista de trámites de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.7.1	Lista de los requisitos	Información contenida en la lista de requerimiento para los trámites de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.7.2	Forma de presentación	Información contenida en la lista de requerimiento para los trámites de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.8.1	Ruta del trámite con fecha y hora	Hace referencia a un correlativo perteneciente al registro de un trámite iniciado en la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.8.2	Porcentaje faltante	Información adjunta a la gestión del trámite. Verificable a través de la observación del correlativo. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.9.1	Archivo personal del docente	Hace referencia a la lista del conjunto de documentos contenidos físicamente en el archivo personal del docente. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.1.9.2	Archivo personal del estudiante	Es el listado de los documentos contenidos físicamente en el archivo personal del estudiante. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100

2.3.2.1	Información institucional	Datos y antecedentes relacionados con la Carrera de Informática contenida en el legajo de Información Institucional. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.2.2	Información calendario de actividades	Es el cronograma de actividades aprobado por el Concejo de la Carrera de Informática a inicios de la gestión anual. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.2.3	Información de normas y reglamentos	Información contenida en el legajo Normas y reglamentos de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.2.4	Información de publicaciones	Información contenida en el listado de publicaciones de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.2.5	Información de noticias	Información contenida en el listado de noticias de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100
2.3.2.6	Información de tramites	Información contenida en el listado de trámites de la Carrera de Informática. Se pregunta si existe (1) o si no existe (0).	Es un criterio binario, discreto y absoluto.	100

**Tabla 5.4 Resultados de la evaluación de confiabilidad**

Código	Atributo	Definición	Criterio elemental	IEi
3.1.1.1	Enlaces rotos	Se mide la cantidad de enlaces que están rotos. Se utiliza la siguiente fórmula: Sea $X=X_i/X_j$ donde $X_i$ : cantidad total de enlaces rotos. $X_j$ : cantidad total de enlaces. $X= 0/25 =0$	Variable normalizada, continua absoluta.	100
3.1.1.2	Enlaces inválidos	Se mide la cantidad de enlaces que son inválidos. Se utiliza la siguiente fórmula: Sea $X=X_i/X_j$ donde $X_i$ : cantidad total de enlaces no validos. $X_j$ : cantidad total de enlaces. $X= 0/25 =0$	Variable normalizada, continua absoluta.	100
3.1.1.3	Enlaces no implementados	Se mide los enlaces no implementados en el Sitio Web. Sea $X=0$ si no existen	Multinivel, discreto, absoluto	100

		enlaces implementados, X=1 si existen algunos enlaces implementados y X=2 si todos los enlaces están implementados.		
3.1.2.1	Deficiencias o cualidades ausentes debido a diferentes navegadores.	Se mide si existe una mala presentación de la aplicación Web debido a los navegadores. Sea $X=(X_i-X_j)*(100/X_i)$ donde $X_i$ : cantidad total de navegadores conocidos. $X_j$ : cantidad total de navegadores que presentan cualidades ausentes	Multinivel, discreto, absoluto	80
3.1.2.2	Deficiencias o resultados inesperados independientemente del navegador	Se mide si existe una mala presentación del Sitio Web sin que afecte el navegador que se utiliza. Si existe buena presentación (1) Si existe una mala presentación (0).	Es un criterio de referencia de calidad directa	100
3.1.2.3	Nodos destinos en construcción	Se mide si existen páginas en construcción. Se pregunta si no existe (1) o si existe (0)	Es un criterio binario absoluto	100
3.1.2.4	Nodos de Web muertos	Se mide si existen páginas Web muertas o sin enlace de retorno. Se pregunta si no existe (1) o si existe (0)	Es un criterio binario discreto absoluto	100

Tabla 5.5 Resultados de la evaluación elemental de eficiencia

Código	Atributo	Definición	Criterio elemental	IEi
4.1.1	Páginas de acceso rápido	Se mide el tamaño de todas las páginas de la aplicación considerando todos sus componentes, para determinar la velocidad de acceso. X=0 si la velocidad es lenta, X=1 si es media y X=2 si es rápida.	Multinivel, discreto, absoluto	100
4.2.1.1.1	Imagen con título	Mide el porcentaje de la presencia de la etiqueta <i>alt</i> de manera de incluir texto alternativo a la imagen cuando se desactiva la propiedad <i>ver imágenes del navegador</i> . X=0 si todas las imágenes no tienen título, X=1 si la mitad del total de las imágenes tienen título y X=2 si todas tienen.	Multinivel, discreto, absoluto	100
4.2.1.1.2	Legibilidad global	Representa la presencia de calidad en consideración del	Es un criterio binario discreto absoluto	100

		nivel de legibilidad global del sitio, cuando se desactiva la propiedad <i>ver imágenes del navegador</i> . X=0 si no existe calidad en la legibilidad, X=1 si existe calidad en la legibilidad.		
4.2.2.1	Número de vistas considerando marcos	Los frames o marcos organizan a una ventana en diferentes áreas o sublistas tanto de control como de contenido. Cuanto mayor es la cantidad de marcos, menor es la accesibilidad de la ventana. X=0 si existe, X=1 si no existe.	Es un criterio binario discreto absoluto	100
4.2.2.2	Versión sin marcos	Cuando se utilizan marcos, es deseable que cuente con una opción de versión sin marcos. X=1 si existe, X=0 si no existe.	Elemental binario discreto absoluto	100

### 3.9. EVALUACIÓN GLOBAL

Los valores obtenidos en la evaluación elemental en las tablas 5.2, 5.3, 5.4. y 5.5 constituyen la base principal para la evaluación global. Para obtener los valores de las características y subcaracterísticas definidos en el árbol de requerimientos, aplicando un mecanismo de agregación paso a paso, es decir aplicando una variación del método LSP (Logic Scoring of Preference). La función de agregación utilizada para obtener estos valores, se basa en la función denominada media potencia pesada:

$$IG(r) = (P^r_1IE_1 + P^r_2IE_2 + \dots + P^r_mIE_m) / r$$

Donde:

$P^r_1IE_1$ : representa el promedio de un nivel analizado en las tablas asociadas a las características de evaluación.

$m$ : representa el número de niveles analizados.

$r$ : representa el número de características.

$IG(r)$ : representa el promedio de la función total de agregación

**Tabla 5.6 Resultados de las evaluaciones parciales y globales**

<b>Código</b>	<b>Características y grados</b>	<b>Definición</b>	<b>IGi</b>
1	Usabilidad	Promedio de los niveles jerárquicos	100
	Código de grado 4	Promedio del nivel jerárquico 4	1400/14
2	Funcionalidad	La navegación por la aplicación Web	93,57
	Código de grado 4	Promedio del nivel jerárquico 4	975/12
	Código de grado 5	Promedio del nivel jerárquico 5	2300/23
3	Confiabilidad		97,14
	Código de grado 4	Promedio del nivel jerárquico 4	680/7
4	Eficiencia		100
	Código de grado 3	Promedio del nivel jerárquico 3	100/1
	Código de grado 4	Promedio del nivel jerárquico 4	200/2
	Código de grado 5	Promedio del nivel jerárquico 5	200/2

**Tabla 5.7 Características de alto nivel y evaluación global**

<b>Característica</b>	<b>IGi</b>
Usabilidad	100
Funcionalidad	93,57
Confiabilidad	97,14
Eficiencia	100
Evaluación de Calidad Global	97,67

De acuerdo los resultados obtenidos en la evaluación de la calidad global, aplicando el modelo Web site QeM, se concluye que el nivel de satisfacción de los usuarios visitantes es del 97,67%, al actualizar o navegar por el portal este valor esta dentro los márgenes de satisfacción definidos de acuerdo al modelo Web site QeM.

## CAPITULO 4

# CONCLUSIONES Y RECOMENDACIONES

### Resumen

Éste capítulo describe los resultados ya obtenidos de los objetivos planteados para el desarrollo del proyecto el cual se hizo con las metodologías seleccionadas y establecidas. Muestra el estado actual de la propuesta, las conclusiones derivadas de su desarrollo, el cumplimiento de los objetivos, las recomendaciones y los trabajos futuros

### 4.1. CONCLUSIONES

El proyecto presentado en capítulos precedentes, denominado: “Sistema de Información Basado en la Web para la Carrera de Informática de la UMSA”, cumple con los objetivos planteados, al utilizar un modelo de procesos para el análisis, diseño e implementación de una plataforma de tecnologías de la información y comunicación para la gestión de información y de servicios, que sea de fácil administración para la Carrera de Informática de la Universidad Mayor de San Andrés, y que al mismo tiempo coadyuve en la difusión de la información de carácter público, permitiendo la interacción de sus usuarios con la unidad de forma rápida y segura.

El análisis correspondiente que se hizo para la selección de la metodología a utilizar permitió conseguir un sistema de tecnología Web eficiente, que permitió aprovechar herramientas orientadas a objetos. El sistema es flexible a modificaciones futuras que permiten integrar nuevos módulos, sin afectar el funcionamiento del sistema. La aplicación posee interfaces gráficas de apoyo al usuario para el manejo del sistema de una fácil operabilidad para alcanzar resultados en la utilización de sus recursos.

### 4.2. ESTADO DE LOS OBJETIVOS

El objetivo general descrito en el apartado 1.5.1. planteaba lo siguiente: *Los canales de comunicación y difusión de información con los que cuenta la Carrera de Informática de la Universidad Mayor de San Andrés, harán uso adecuado de las Tecnologías Web, y facilitarán la información de los servicios que la Carrera presta a la comunidad informática.*

Este objetivo se encuentra cubierto con el desarrollo de la aplicación Web que de acuerdo a las pruebas realizadas en la Carrera de Informática de la Universidad Mayor de San Andrés, colabora con la difusión de la información de carácter público y permite la interacción de los usuarios haciendo uso de las tecnologías Web adecuadas, dando a conocer los servicios que el departamento presta a la comunidad universitaria de manera eficiente. Por consiguiente se puede afirmar que el objetivo general fue alcanzado con un alto grado de aceptación por el cliente.

Por otra parte los objetivos específicos planteados en el apartado 1.5.2. fueron alcanzados de la siguiente manera:

- a) Este objetivo específico pretendía: *implementar herramientas que coadyuven a la difusión de avisos, circulares y anuncios periódicos por parte de la Carrera de Informática*, el cumplimiento de este objetivo se muestra en la aplicación Web en el grupo de enlaces de interés, específicamente en el vínculo de avisos y circulares, por consiguiente se puede decir que este objetivo fue alcanzado.
- b) Este objetivo mencionaba: *difundir la información institucional de la Carrera de Informática y los servicios que presta a la comunidad universitaria*, este objetivo fue cumplido con el desarrollo de la aplicación Web. De manera específica en el grupo de información institucional abarcando los siguientes vínculos: información de la UMSA, pagina principal, objetivos, estructura orgánica, autoridades, historia y ubicación. Por lo tanto se puede afirmar que este objetivo fue alcanzado y superado en relación a su expectativa inicial.
- c) Este objetivo mencionaba: *difundir el cronograma de actividades a llevarse a cabo durante el año y mantener la constante disponibilidad de esta información a la comunidad universitaria y monitorear su cumplimiento*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en el grupo de enlaces de interés, específicamente en el vínculo de calendario de actividades. Este calendario también funciona de manera interactiva mostrando un calendario en línea con las actividades referidas a la fecha de consulta. Por lo tanto se puede afirmar que este objetivo fue alcanzado.
- d) Este objetivo específico pretendía: *mantener informada a la población universitaria interesada, dando a conocer los requisitos para su incorporación y asegurar un orden formal para la presentación de documentos*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en el grupo de enlaces de interés, específicamente

en los vínculos de avisos y circulares, trámites y el vínculo de normas y reglamentos. Por consiguiente se puede afirmar que este objetivo fue alcanzado.

- e) Este objetivo mencionaba: *mantener informados a los estudiantes sobre los trámites que le afectan en la Carrera de Informática*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en el grupo de enlaces de interés, específicamente en el vínculo de tramites. Por lo tanto se puede afirmar que este objetivo ha sido alcanzado.
- f) Este objetivo específico pretendía: *mantener informados a los estudiantes sobre el estado de avance de sus trámites ante en la Carrera de Informática*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en la pagina personal del alumno, específicamente en el vínculo de tramites. Por consiguiente se puede afirmar que este objetivo fue alcanzado.
- g) Este objetivo específico pretendía: *mantener informados a los estudiantes sobre la producción intelectual y hoja de vida de los docentes*.
- h) *El estado de avance de sus trámites ante en la Carrera de Informática*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en la pagina personal del alumno, específicamente en el vínculo de tramites. Por consiguiente se puede afirmar que este objetivo fue alcanzado.
- i) Este objetivo específico pretendía: *Informar a la comunidad universitaria y dar a conocer las leyes nacionales, normativas internas y reglamentos vigentes de la UMSA con las que la Carrera de Informática trabaja y a las cuales los docentes y administrativos se deben ajustar*. Este objetivo fue cumplido con el desarrollo de la aplicación Web, en el grupo de enlaces de interés, específicamente en el vínculo de normas y reglamentos, donde se puede hacer una descarga de los principales documentos que norman las actividades de esta carrera. Por consiguiente se puede afirmar que este objetivo ha sido alcanzado.
- j) Este objetivo mencionaba: *Proponer un modelo de aplicación Web de fácil administración y actualización*. La aplicación Web es de fácil acceso y cuenta con amplia aceptación al interior de *la Carrera de Informática*, lo que la hace de fácil administración, tanto para el usuario casual, para el usuario docente y para el administrador del sistema. Por consiguiente se puede decir que este objetivo ha sido alcanzado.
- k) Este objetivo específico pretendía: *Desarrollar la aplicación Web propuesta*. La aplicación Web fue desarrollada de acuerdo a las herramientas propuestas en el

capítulo 2, al ser una aplicación con fácil accesibilidad y de usabilidad sencilla, se puede decir que este objetivo ha sido alcanzado.

- l) Finalmente este objetivo mencionaba: *Evaluar la aplicación Web*. La aplicación fue evaluada en función a la percepción de los usuarios respecto a dos parámetros: accesibilidad y usabilidad, los indicadores de eficiencia de estos parámetros son adecuados, por consiguiente se puede decir que este objetivo fue alcanzado.

#### **4.3. RECOMENDACIONES**

Considerando que existe un constante cambio tecnológico y confiando en que este proyecto permite la inclusión de nuevos módulos que responda a la automatización de los servicios, se recomienda aprovechar estas cualidades del proyecto desarrollado.

Se ha respetado un modelo de proceso evolutivo incremental con la inclusión de las herramientas necesarias para la planificación, ingeniería de requerimientos, diseño Web, código PHP, pruebas Web y medidas de usabilidad, funcionalidad, confiabilidad y eficiencia, sería pertinente utilizar otras herramientas con este mismo modelo de procesos.

En futuros desarrollos no debería descartarse la reutilización de software y la facilidad de acoplar módulos, propios de la ingeniería del software orientada a objetos.

#### **4.4. TRABAJOS FUTUROS**

Algunos trabajos que pueden emprenderse en el futuro, tomando como base el presente proyecto de grado son:

- a) Modelo de procesos utilizando metodologías ágiles como la programación extrema.
- b) Adicionar agentes de software para los procesos de consulta en línea.
- c) Crear una aplicación Web que sea capaz de gestionar los trámites de la Carrera de Informática.
- d) Para la eficiencia del anterior punto, plantear el uso del comercio electrónico para reducir la dependencia de la Caja Central de la UMSA a través de un proceso ágil con la participación de la banca local.

## BIBLIOGRAFÍA

- [Allamaraju, 2000] Allamaraju, S. et al. *Professional Java Server Programming J2EE Edition*. Wrox Press, Inc., 2000.
- [Allen y Frost, 1998] Allen, P. y Frost, S. (1998), *Component-Based Development for Enterprise Systems. Applying The SELECT Perspective™*. Cambridge University Press.
- [ALSI, 2006] Academia Latinoamericana de Seguridad Informática, 2006.
- [Baeza Yates & Rivera, 2003] Ricardo Baeza Yates & Cuahutémoc Rivera Loaiza. *Ubicuidad Y Usabilidad en la Web*. 2003.
- [Bauer & King, 2005] Christian Bauer, Gavin King “*Hibernate in Action*”, Editorial: MANNING Greenwich 2005
- [BEA, 2003] BEA Systems. *BEA WebLogic Server*. 2003. <http://www.bea.com/products/Weblogic/server/index.shtml>.
- [Booch et al., 1999] Booch, G. Rumbaugh, J. y Jacobson, I. (1999), *El Lenguaje Unificado de Modelado*. Addison Wesley Iberoamericana, Madrid.
- [Brittain & Darwin, 2004] Jason Brittain, Ian F. Darwin, *Tomcat: The Definitive Guide*, 2004.
- [Carmichael, 1994] Andy Carmichael “*Object Development Methods*”. Copyright 1994 By SIGS Books New York
- [Chiavenato, 1988] Chiavenato, Idalberto, “*Introducción a la Teoría General de la Administración*” Cuarta Edición, Santafé de Bogotá, Colombia, McGraw-Hill Interamericana S.A., 1995.
- [Cuerda y Minguillón, 2004] Cuerda García Xavier y Julia Minguillón Alfonso (2004) *Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto* 29.11.04 <http://www.loquesea.norevisanunca/CMS/admcont.htm>
- [D'Souza y Cameron, 1999] D'Souza, F. y Cameron, A. (1999), *Objects, Components, and Frameworks with UML. The Catalysism Approach*. Addison-Wesley.
- [Date, 1993] Date, C. (1993). *Introducción a los Sistemas de bases de datos*. Addison-Wesley, Wilmington, E.U.A.
- [Fernández, 2001] Daniel Fernández Lanvin “*Aplicación de Objetos Distribuidos para la construcción de Aplicaciones Web basadas en Tecnología Java*” Estudios de Doctorado Avances en informática, Curso Objetos Distribuidos: Nuevos Avances y Plataformas 2001.
- [Ferré, 2003] Ferré Grau Xavier. *Desarrollo Orientado a Objetos con UML*. Facultad de Informática – Universidad Politécnica de Madrid, España. V2.0. 2003.
- [García y Tramullas, 1996] García Marco, Javier Tramullas Saz, Jesús. *World Wide Web: Fundamentos, navegación y lenguajes de la red mundial de información*. Madrid: Ra-Ma, 1996.
- [Graham, 1991] Ian Graham, “*Object Oriented Methods*”, Addison Wesley 1991
- [Heinckiens, 1998] Heinckiens (1998), *Building Scalable Database Applications. Object-Oriented Design, Architectures, and Implementations*. Peter M. Heinckiens. Addison-Wesley, 1998.
- [Henderson-Sellers, 1996] Henderson-Sellers (1996) «The OPEN-MeNtOR Methodology». Brian Henderson-Sellers. *Object Magazin*. Noviembre, pp.56-59.
- [Hughes, 2000] Hughes, Kevin. *Guide to Cyberspace..* <http://www.kevcom.com/words/quide/quide.toc.html>
- [IEEE, 1998] IEEE-STD-830-1998: *Especificaciones de los Requisitos del Software*. Reporte de la IEEE, 1998.

- [Instone, 1997] Instone Keith. "Site Usability Heuristics for the Web", [http://www.Webreview.com/1997/10\\_10/strategists/10\\_10\\_97\\_2.shtml](http://www.Webreview.com/1997/10_10/strategists/10_10_97_2.shtml) Webreview.com, 1997
- [iPlanet, 2003] iPlanet E-Commerce Solutions. *Product Map*. 2003. [http://www.iplanet.com/products/product\\_map/product\\_name\\_2\\_0a.html](http://www.iplanet.com/products/product_map/product_name_2_0a.html).
- [Jacobson,1995] Ivar Jacobson "*The Object Advantage: Business process Reengineering with object Tecnology*", 1995.
- [Jakarta, 2002a] The Jakarta Project. *Jakarta Tomcat*. <http://jakarta.apache.org/tomcat>
- [Jakarta, 2002b]The Jakarta Project. *Tomcat User's Guide*. <http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-ug.html>.
- [Kobrin, 1999] Kobrin, C. (1999), «UML 2001: A Standarization Odyssey». *Communications of the ACM*, Vol. 42, No. 10, pp.29-37.
- [Kumiawan,2002] Budi Kumiawan, "*Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions*", Editorial: New Riders Publishing, 2002.
- [Larman, 1999] Larman C. *UML y Patrones*. Prentice Hall, 1999.
- [Larman, 2001] C. Larman, "*Applying UML and Patterns*", Prentice-Hall, 2001.
- [Lewis, 2004] Howard M. Lewis Ship, *Tapestry in Action*, Editorial: Manning Grenwich, 2004
- [Murray & Costanzo, 1999] Murray George & Tania Costanzo. *Usability and the Web: An Overview*, <http://www.nlc-bnc.ca/9/1/p1-260-e.html> Information Technology Services, National Library of Canada. August 1999
- [Navarro,2004 ] Aspect-Oriented Requirements Engineering and Architecture Design Workshop, collocated to OOPSLA, Monday, October 25, 2004
- [Nicamer, 2006] [www.nicamer.com/texto.htm](http://www.nicamer.com/texto.htm) [Fecha de acceso: junio 2006]
- [Nielsen, 1990] Nielsen Jakob. "*Ten Usability Heuristics*", <http://www.useit.com/papers/ heuristic/heuristic list.html> Useit.com, 1990
- [Odell, 2000] Odell, J., 2000, Agents and Emergence. *Journal on Object Oriented Programming*. Febrero, pp 34-36.
- [Olsina, 1998] Olsina, L., 1998, "*Toward Improvements in Hypermedia Process Modeling*", Proceed. at the Symposium of Software Technology, 27 JALIO, Bs As., pp. 219-226
- [Olsina, 1999] Luis Antonio Olsina, "*Metodología Cuantitativa para la Evaluación y comparación de la Calidad del Sitio Web*" Tesis Doctoral. Noviembre 1999.
- [OMG, 1999] OMG (1999), *UML 2.0 Request for Information V1.0*. Analysis & Design Platform Task Force. OMG, Agosto 1999. En: <http://www.omg.org>.
- [Palos, 2002] Palos Juan Antonio. *Introducción a los Servicios Web en Java*. TheServerSide. 2002. <http://www.programacion.com/java/autor/88/>
- [Postgres, 1999] Manual de Usuario de PostgreSQL. © 1999 por el Postgres Global Development Group. <http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>
- [Pressman, 2002] Roger S. Pressman, "*Ingeniería del Software: Un Enfoque Práctico*", Quinta Edición, Editorial: Mc Graw Hill, 2002.
- [Pursnani, 2004] Pursnani Vandana, *Introducción a la Programación de Java Servlets*, 2004. Traducido por Germán Rojo Eguren <http://www.acm.org/crossroads/espanol/xrds8-2/servletsProgramming.html#authorbio#authorbio>
- [Quinteros, 2001] Quinteros Verónica, 2001. Manual de uso eficiente del lenguaje Java. <http://www.sunmicrosystems.com/AISI/vencinas/qquinteros/usingjava.html>
- [Rhyno, 2003] Rhyno, A. *The Ten Commandments of Content Management* [en línea] [usr/lib/info](http://usr/lib/info), 18 feb 2003 <<http://usr.lib.info/story/2003/2/17/82354/8716>> [Consulta: abril 2004]
- [Nielsen, 2002] Nielsen, J., *Usabilidad. Diseño de sitios Web*, Madrid, Prentice Hall, 2002.

- [Robertson, 2002] Robertson, J., *How to evaluate a content management system* [en línea]. Step Two, 23 enero 2002 <[http://www.steptwo.com.au/papers/kmc\\_evaluate/index.html](http://www.steptwo.com.au/papers/kmc_evaluate/index.html)> [Consulta: marzo-abril 2006]
- [Robertson, 2003] Robertson, J., *So, what is a content management system?* [en línea]. Step Two, 3 junio 2003 <[http://www.steptwo.com.au/papers/kmc\\_what/index.html](http://www.steptwo.com.au/papers/kmc_what/index.html)> [Consulta: marzo-abril 2006].
- [Rossi et al 97] Rossi, G., Schwabe, D.; Garrido, A., 1997, " *Design reuse in Hypermedia Application Development*", Proceedings of ACM Hypertext 97, Southampton, England pp. 57-66.
- [Schawe y Rossi, 1995] Schawe y Rossi (1995), «The Object-Oriented Hypermedia Design Model». *Communications ACM*, 58(8), pp. 45-46.
- [Schwabe y Rossi, 1997] Daniel Schwabe y Gustavo Rossi: *The Object-Oriented Hypermedia Design Model (OOHDM)*. <http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf>
- [Schwabe, Rosi y Barbosa, 1996] Schwabe D., G. Rosi y S. Barbosa. Systematic Hypermedia Application Design with OOHDM. *Proceedings of the ACM International Conference on Hipertext (Hypertext'96)*. Washington DC, March, 16-20, 1996. <http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf>
- [Shaw,1995] Shaw. M., y D. Garlan, "Formulations and Formalisms in Software Architecture". *Volume 1000-Lecture Notes in Computer Science*, Springer Verlag, 1995.
- [Silva y Mercerat, 1999] Silva Darío Andrés y Silvia Mercerat. *Construyendo aplicaciones Web con una metodología de diseño orientada a objetos*. [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)
- [Sun, 2001] Sun Microsystems. *A New Model for Enterprise Applications*. 2001. <http://java.sun.com/j2ee/overview2.html>.
- [Sun, 2002a] Sun Microsystems. *Hierarchy For Package javax.servlet*. <http://java.sun.com/j2ee/tutorial/api/javax/servlet/package-tree.html>.
- [Sun, 2002b] Sun Microsystems. *Java™ Servlet Technology: Implementations and Specifications*. 2002. <http://java.sun.com/products/servlet/download.html>.
- [Sun, 2002c] Sun Microsystems. *Interface javax.servlet.Servlet*. <http://java.sun.com/products/servlet/2.1/api/javax.servlet.Servlet.html>.
- [Sutherland, 1999] Sutherland (1999), *Business and the Evolution of the Internet*. Accedido el 13 de marzo de 2000 en: <http://www.onemind.com/hpi/917622885/index.html>
- [Szypersky, 1999] Szypersky, C. (1999), *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley.
- [TomCat, 2003] Página oficial de apache TomCat, 2003. <http://tomcat.apache.org/>
- [Vegas, 2002] Vegas Jesús. *Tecnologías Web*. 2002. Material disponible en: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>
- [Vujacich & Rodríguez, 2005] Vujacich G. & J.R. Rodríguez. *Sistemas Administradores de Contenidos*. Publicación: Septiembre 2005. Licencia FDL. (GNU Free Documentation License. Version 1.2, November 2002) de la Free Software Foundation: <http://www.fsf.org/licenses/licenses/fdl.txt>
- [W3C, 2002] W3C. *Architecture of the World Wide Web, Volume One*. <http://www.w3.org/TR/Webarch/>
- [Wiebe, 1999] Wiebe, J. (1999), *Poly-wha? Understanding Business Object Terminology and Acronyms*. Accedido el 13 de marzo de 2000 en: <http://www.onemind.com/hpi/919200664/index.html>.

## NORMAS

- [Universidad, 1998] Universidad, “Reglamento del Régimen Académico – Docente de la Universidad Boliviana”. Universidad Boliviana, X Congreso Nacional de Universidades, 1998.
- [Universidad, 1996a] “Normas y Reglamentos Universitarios de Personal Docente”, Universidad Mayor de San Andrés, Vicerrectorado, 1996.
- [Universidad, 1996b] “Normas y Reglamentos Universitarios de Personal Docente Tomo I”, Universidad Mayor de San Andrés, Vicerrectorado, 1996.
- [Universidad, 1996c] “Manual de Procedimientos y Trámites Administrativos de Personal Docente”, Universidad Mayor de San Andrés, Vicerrectorado, 1996.
- [Hacienda, 2001] “Normas Básicas del Sistema de Administración de Personal” Ministerio de Hacienda Servicio Nacional de Administración de personal, La Paz- Bolivia – 2001



## ANEXO A GLOSARIO

**Apache** - servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.)

**Blog (*Web log*)** - Diario en formato Web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

**CMF (*Content Management Framework*)** - Entorno a programación de aplicaciones enfocado al desarrollo de CMS.

**CMS (*Content Management System*)** - Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como *Web Content Management (WCM)* sistema de gestión de contenido de Webs.

**GPL (*General Public License*)** - Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

**CSS (*Cascading Style Sheets*)** - Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llaman "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en una Web.

**HTML (*HyperText Markup Language*)** - Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en Webs.

**HTTP (*HyperText Transfer Protocol*)** - Protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML)

**LAMP (*Linux, Apache, MySQL y PHP, Perl o Python*)** - Arquitectura formada por el sistema operativo Linux, el servidor Web Apache, la base de datos MySQL y uno o más de los lenguajes de programación PHP, Perl o Python.

**LCMS (*Learning Content Management System*)** - Software para la gestión automatizada de cursos en línea, que incluye gestión de usuarios, de resultados y de recursos. Es un sistema de gestión de cursos con las capacidades de un CMS y por lo tanto de gestionar también los contenidos de los recursos.

**LGPL (*Lesser General Public License*)** - Licencia que permite el uso y modificación de librerías de código para desarrollar software libre o propietario. Antes conocida como *Library GPL*.

**LMS (*Learning Management System*)** - se diferencia de los LCMS en que no hay gestión de los contenidos, sino simplemente administración del curso, pero acostumbra a utilizarse como sinónimo. También conocido como *Course Management System (CMS)* o *Virtual Learning Environment (VLE)*

**Metadatos** - datos sobre los datos. Información que describe el contenido de los datos. Por ejemplo de un documento serían metadatos, entre de otros, su título, el nombre del autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.

**MySQL** - base de datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

**Open source** - Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como *free software*.

**Perl** - lenguaje de programación de alto nivel que hereda de diversos lenguajes, muy utilizado para el desarrollo de Webs dinámicas.

**PHP (*PHP Hypertext Preprocessor*)** - Lenguaje de programación para el desarrollo de Webs dinámicas, con sintaxis parecida a la C. Originalmente se conocía como *Personal Hombre Page tools*, herramientas para páginas personales (en Internet).

**Python** - lenguaje interpretado de alto nivel orientado a objetos.

**URL (*Uniform Resource Locator*)** - Dirección de un recurso en la Web. Tiene el formato protocolo://máquina.domini:port/ruta/recurso. Por ejemplo <http://www.uoc.edu/dt/20396/index.html> donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

**WAI (*Web Accessibility Initiative*)** - Iniciativa del Consorcio de la *World Wide Web* para asegurar que las Webs están diseñadas pensando en el acceso de personas con discapacidades.

**Web** - sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (*el Web, un Web*) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (*la Web, una Web*) se refiere a una página Web concreta dentro del sitio Web.

**WebDAV (*Web-based Distributed Authoring and Versioning*)** - es una extensión del protocolo HTTP que permite a los usuarios editar y administrar archivos de forma colaborativa en servidores Web remotos (definición <http://www.Webdav.org/de>)

**Web log** - Ver 'Blog'.

**WYSIWYG (*What You See Is What You Get*)** - Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.



## ANEXO B ORGANIGRAMA CARRERA INFORMÁTICA

