

ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE LATACUNGA

CARRERA DE SISTEMAS E INFORMÁTICA

**DISEÑO DEL PORTAL ACADÉMICO DE LA ESCUELA
POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA (ESPE-L), CON
UNA METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A
OBJETOS (OOHDM)**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

**CHANGO CANDO MILTON RODRIGO
IZURIETA NARANJO EFRÉN LEONARDO**

Latacunga, Noviembre del 2006

CERTIFICACIÓN

Los suscritos Ing. Raúl Rosero e Ing. Fabián Montaluisa certifican que el presente trabajo teórico – práctico, fue desarrollado íntegramente por los señores: Chango Cando Milton Rodrigo e Izurieta Naranjo Efrén Leonardo, bajo nuestra supervisión.

Ing. Raúl Rosero
DIRECTOR DE TESIS

Ing. Fabián Montaluisa
CODIRECTOR DE TESIS

AGRADECIMIENTO

Nuestro principal agradecimiento es a Dios por su infinito amor incondicional quien nos ha bendecido siempre, a nuestros padres por su amor, su sacrificio que cada día nos regalaron, a nuestros familiares por su apoyo incondicional en todo momento, a los amigos por el buen tiempo compartido en el transcurso de la vida estudiantil.

A la Escuela Politécnica del Ejército Sede Latacunga en especial a la Carrera de Sistemas e Informática y a sus docentes, por la oportunidad que nos brindaron para culminar nuestros estudios.

A nuestro director, codirector y a la Unidad de Tecnologías de Información y Comunicación por su ayuda durante el desarrollo de este proyecto.

DIOS LES BENDIGA A TODOS USTEDES

DEDICATORIA

Este proyecto le entrego en ofrenda a DIOS, lo dedico a mis padres y familiares.

Leo

Este trabajo lo dedico a DIOS, a mis padres, a mi esposa, a mi hija, a mis hermanos y a toda mi familia, que han sido mi ayuda y mi inspiración para concluir este proyecto.

Milton

PRESENTACIÓN

Los portales Web tienen cada vez más complejidad añadida y, asimismo, están creciendo paulatinamente. Presenciamos en nuestros días una demanda creciente de portales Web con funcionalidad de aplicación de software. Además, un portal Web debe satisfacer necesidades tan diversas que pueden ir desde la distribución y presentación de documentos hiperenlazados cuyo contenido puede estar compuesto por diversos medios (texto, sonido, imagen, video), hasta la automatización de procesos de negocios más o menos complejos. A pesar de este crecimiento en la demanda y complejidad, según observamos, modelos de proceso de producto bien definidos que promuevan un ciclo de vida de desarrollo repetible y eficiente, y modelos de proceso y metodologías de evaluación que promuevan la comprensión, el control y la mejora de la calidad de portales Web, no están acompañando este rápido proceso.

Más bien las prácticas actuales para desarrollar portales Web son realizados en forma tradicional, y el aseguramiento y el control de la calidad son por lo general, procesos poco tenidos en cuenta. Esto motiva un cambio de paradigma en el seguimiento de estos procesos y prácticas para minimizar la crisis en los desarrollos en la Web y por lo tanto, propender a la calidad y a una efectiva evolución de los portales Web. En esta dirección, el empleo sistemático y disciplinado de procesos, métodos, modelos, y técnicas de Ingeniería Web para el desarrollo, el mantenimiento, y el aseguramiento de la calidad de los portales Web, principalmente en los proyectos de mediana o gran escala, promoverá el cambio de paradigma de estrategias tradicionales hacia estrategias más objetivas y justificables.

Para el diseño y desarrollo de este portal Web, utilizaremos la metodología de diseño hipermmedia orientado a objetos (OOHDM), la cual realiza un aporte de

ingeniería para sentar las bases y construir las estructuras de diseño, para el uso del portal Web en una forma adecuada.

En un enfoque sistemático, disciplinado y cuantitativo que se adecue a la evaluación, comparación y análisis de la calidad de aplicaciones centradas en la Web.

El desarrollo del portal académico de la ESPE-L, ayudará a satisfacer las necesidades actuales y futuras, que cada día van apareciendo en este mundo evolutivo, tanto en información como en tecnología. Con la aplicación de la metodología y técnicas se logrará un portal Web de interés y calidad que alcancen las expectativas que tiene la comunidad politécnica.

Ofreciendo una mayor comodidad en la interacción de la interfaz con el usuario, así como también dando un mayor realce y competitividad en la presentación hacia el mundo globalizado.

CONTENIDO

CAPÍTULO I.....	- 1 -
I.- INGENIERÍA WEB Y USABILIDAD.....	- 1 -
1.1.- INGENIERÍA WEB.....	- 1 -
1.1.1.- INTRODUCCIÓN	- 1 -
1.1.2.- DEFINICIÓN.....	- 2 -
1.1.3.- ESTÁNDARES Y TECNOLOGÍAS	- 3 -
1.1.4.- GESTIÓN DEL CONOCIMIENTO EN LA WEB.....	- 18 -
1.1.5.- ARQUITECTURA DE LA INFORMACIÓN WEB	- 19 -
1.1.6.- MÉTRICAS WEB	- 22 -
1.2.- INGENIERÍA DE LA USABILIDAD	- 26 -
1.2.1.- GENERALIDADES.....	- 26 -
1.2.2.- MODELO DE PROCESO PARA LA INGENIERÍA DE LA USABILIDAD	- 34 -
1.3.- MULTIMEDIA WEB Y DISEÑO DE INTERFACES	- 40 -
1.3.1.- INTRODUCCIÓN A LA MULTIMEDIA WEB.....	- 40 -
1.3.2.- CONCEPTOS DE LOS ELEMENTOS MULTIMEDIALES.....	- 40 -
1.3.3.- MULTIMEDIA DISTRIBUIDA	- 42 -
1.3.4.- MÉTODO DE DESARROLLO DE SISTEMAS MULTIMEDIA EN LA WEB	- 42 -
1.3.5.- DISEÑO DE INTERFACES DE USUARIO	- 48 -
CAPÍTULO II.....	- 50 -
II.- METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A OBJETOS (OOHDM).....	- 50 -
2.1.- INTRODUCCIÓN.....	- 50 -
2.2.- ANÁLISIS	- 52 -
2.2.1.- LEVANTAMIENTO DE REQUISITOS	- 52 -
2.3.- DISEÑO	- 84 -
2.3.1.- DISEÑO CONCEPTUAL.....	- 84 -

2.3.2.-	DISEÑO NAVEGACIONAL	- 110 -
2.3.3.-	DISEÑO DE INTERFAZ ABSTRACTA	- 157 -
2.4.-	IMPLEMENTACIÓN	- 164 -
2.4.1.-	MAPEAMIENTO DE LOS OBJETOS	- 164 -
2.4.2.-	MAPEAMIENTO DE LOS RELACIONAMIENTOS	- 165 -
2.4.3.-	TABLAS DE MAPEAMIENTO DE LOS OBJETOS DE IMPLEMENTACIÓN ..	- 165 -

CAPÍTULO III..... - 167 -

III.- ANÁLISIS Y DISEÑO DE LA ARQUITECTURA DE LA INFORMACIÓN DE LA ESPE-L..... - 167 -

3.1.-	ESTUDIO DE LA ARQUITECTURA DE LA INFORMACIÓN DE LA ESPE-L	- 167 -
3.1.1.-	INFORMACIÓN INSTITUCIONAL	- 167 -
3.1.2.-	SERVICIOS WEB	- 167 -
3.1.3.-	EVENTOS O NOTICIAS	- 170 -
3.1.4.-	COMENTARIOS O SUGERENCIAS	- 170 -
3.2.-	ANÁLISIS	- 171 -
3.2.1.-	ESPECIFICACIÓN DE REQUISITOS (ERS).....	- 171 -
3.2.2.-	LEVANTAMIENTO DE REQUISITOS	- 172 -
3.3.-	DISEÑO	- 173 -
3.3.1.-	DISEÑO CONCEPTUAL.....	- 173 -
3.3.2.-	DISEÑO DE NAVEGACIÓN	- 174 -
3.3.3.-	DISEÑO ABSTRACTO DE INTERFAZ DE USUARIO	- 175 -

CAPÍTULO IV - 176 -

IV.- DESARROLLO DEL PORTAL ACADÉMICO DE LA ESPE-L..... - 176 -

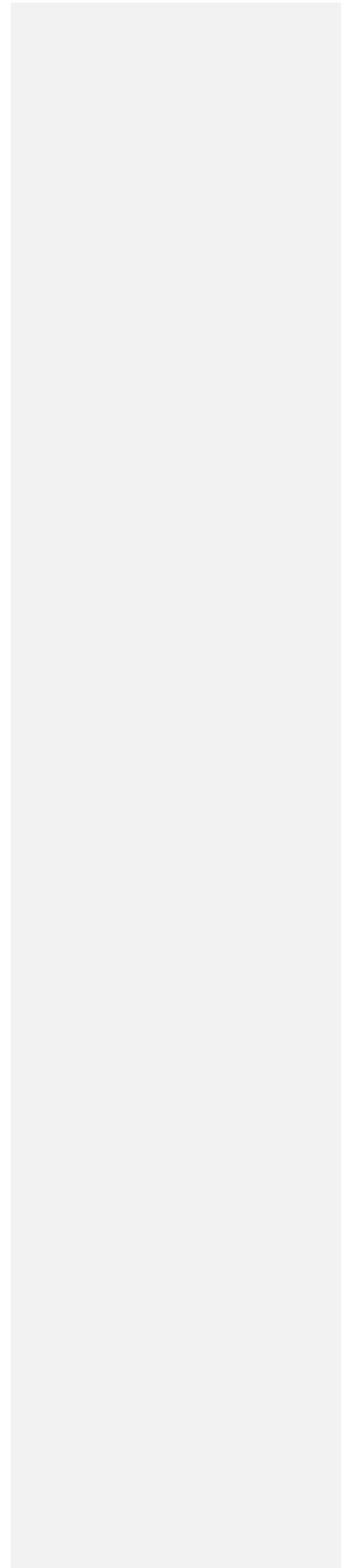
4.1.-	DESARROLLO DEL PORTAL	- 176 -
4.1.1.-	CONFIGURACIONES EN LOS SERVIDORES.....	- 176 -
4.1.2.-	DESARROLLO Y PROGRAMACIÓN DE WEB ESPEL	- 177 -
4.2.-	PRUEBAS Y EVALUACIÓN	- 178 -
4.2.1.-	DESARROLLO DE UN PLAN DE PRUEBAS.....	- 178 -
4.2.2.-	EJECUCIÓN DEL PLAN DE PRUEBAS.....	- 200 -
4.3.-	PUESTA EN MARCHA.....	- 201 -
4.3.1.-	DESARROLLO DE UN PLAN DE LANZAMIENTO	- 201 -

4.3.2.-	DESARROLLO DE UN PLAN DE COMUNICACIONES	- 202 -
4.3.3.-	EL SITIO COMO APOYO DE LA INSTITUCIÓN.....	- 204 -
4.3.4.-	ENLACES DESDE OTROS SITIOS	- 205 -
4.4.-	EVALUACIÓN	- 206 -
4.4.1.-	MÉTRICAS DE EVALUACIÓN DE DESEMPEÑO INTERNAS Y EXTERNAS.....	- 206 -
4.4.2.-	LA IMPORTANCIA DEL ARCHIVO «LOG».....	- 207 -
4.4.3.-	PRESENCIA DEL SITIO EN BUSCADORES.....	- 207 -
 CAPÍTULO V		- 209 -
V.- CONCLUSIONES Y RECOMENDACIONES		- 209 -
 5.1.- CONCLUSIONES		- 209 -
5.2.- RECOMENDACIONES.....		- 211 -

LISTADO DE TABLAS

TABLA I.1 LOS ESTÁNDARES DEL SOFTWARE	- 3 -
TABLA I.2 FASES Y ETAPAS DE LA METODOLOGÍA. DOCUMENTACIÓN RELACIONADA ..	- 44 -
TABLA II.1 CONTROL DE VALIDACIÓN DE LOS UID'S	- 83 -
TABLA II.2 MAPEAMIENTO DE LOS OBJETOS CONCEPTUALES PARA OBJETOS NAVEGACIONALES	- 108 -
TABLA II.3 MAPEAMIENTO DE LOS OBJETOS CONCEPTUALES PARA OBJETOS DE INTERFAZ	- 109 -
TABLA II.4 MAPEAMIENTO DE LOS OBJETOS CONCEPTUALES PARA OBJETOS DE IMPLEMENTACIÓN	- 109 -
TABLA II.5 CARTA DE ENLACE (ORIENTA)	- 124 -
TABLA II.6 CARTA DE CONTEXTO O GRUPO DE CONTEXTO	- 142 -
TABLA II.7 CARTA DE CONTEXTO	- 144 -
TABLA II.8 CARTA DE CONTEXTO O GRUPO DE CONTEXTO	- 144 -
TABLA II.9 CARTA DE ESTRUCTURA DE ACCESO PROFESORES	- 148 -
TABLA II.10 CARTA DE ESTRUCTURA DE ACCESO PROFESORES	- 149 -
TABLA II.11 CARTA DA ESTRUCTURA DE ACCESO PROFESORES : ÁREAS	- 149 -
TABLA II.12 MAPEAMIENTO DE LOS OBJETOS NAVEGACIONALES PARA OBJETOS CONCEPTUALES	- 155 -
TABLA II.13 MAPEAMIENTO DE LOS OBJETOS NAVEGACIONALES PARA OBJETOS DE INTERFAZ	- 156 -
TABLA II.14 MAPEAMIENTO DE LOS OBJETOS NAVEGACIONALES PARA OBJETOS DE IMPLEMENTACIÓN	- 156 -
TABLA II.15 MAPEAMIENTO DE LOS OBJETOS DE INTERFAZ PARA OBJETOS CONCEPTUALES	- 162 -
TABLA II.16 MAPEAMIENTO DE LOS OBJETOS DE INTERFAZ PARA OBJETOS NAVEGACIONALES	- 163 -
TABLA II.17 MAPEAMIENTO DE LOS OBJETOS DE INTERFAZ PARA OBJETOS DE IMPLEMENTACIÓN	- 163 -

TABLA II.18 MAPEAMIENTO DE LOS OBJETOS DE IMPLEMENTACIÓN PARA OBJETOS CONCEPTUALES	- 166 -
TABLA II.19 MAPEAMIENTO DE LOS OBJETOS DE IMPLEMENTACIÓN PARA OBJETOS NAVEGACIONALES	- 166 -
TABLA II.20 MAPEAMIENTO DE LOS OBJETOS DE IMPLEMENTACIÓN PARA OBJETOS DE INTERFAZ.....	- 166 -



LISTADO DE FIGURAS

FIGURA I.1 FASES DE LA ARQUITECTURA DE LA INFORMACIÓN WEB	- 20 -
FIGURA I.2 "PIRÁMIDE DE NECESIDADES" PARA CONSTRUIR UN SITIO WEB	- 20 -
FIGURA I.3 DIVISIÓN DE LA INFORMACIÓN EN UNIDADES LÓGICAS	- 21 -
FIGURA I.4 JERARQUÍA DE LA INFORMACIÓN	- 21 -
FIGURA I.5 ESTRUCTURAR LAS UNIDADES LÓGICAS DE INFORMACIÓN	- 22 -
FIGURA I.6 MODELO DE PROCESO PARA LA INGENIERÍA DE LA USABILIDAD.....	- 35 -
FIGURA I.7 FASES DE LA METODOLOGÍA PARA EL DISEÑO DE PÁGINAS WEB BASADAS EN MULTIMEDIA.....	- 43 -
FIGURA II.1 ESPECIFICACIÓN DE UN ESCENARIO	- 54 -
FIGURA II.2 ESPECIFICACIÓN DE UN CASO DE USO	- 56 -
FIGURA II.3 EJEMPLO DE ÍTEM DE INFORMACIÓN	- 58 -
FIGURA II.4 EJEMPLO DE ESTRUCTURA	- 59 -
FIGURA II.5 EJEMPLO DE INFORMACIÓN OPCIONAL	- 60 -
FIGURA II.6 EJEMPLO DE ENTRADA DE USUARIO	- 61 -
FIGURA II.7 EJEMPLO DE ENTRADA DE USUARIO ENUMERADA.....	- 62 -
FIGURA II.8 EJEMPLO DE SALIDA DEL SISTEMA.....	- 63 -
FIGURA II.9 EJEMPLO DE TEXTO	- 64 -
FIGURA II.10 EJEMPLO DE ESTADO DA INTERACCIÓN	- 64 -
FIGURA II.11 ESTADO INICIAL DE INTERACCIÓN	- 65 -
FIGURA II.12 EJEMPLO DE ESTADO INICIAL DA INTERACCIÓN	- 65 -
FIGURA II.13 ESTADOS ALTERNATIVOS DE INTERACCIÓN.....	- 65 -
FIGURA II.14 EJEMPLO DE ESTADOS ALTERNATIVOS DE INTERACCIÓN	- 66 -
FIGURA II.15 EJEMPLO DE SUB-ESTADOS DE UN ESTADO DE INTERACCIÓN	- 66 -
FIGURA II.16 EJEMPLO DE SUB-ESTADOS DE UN ESTADO DE INTERACCIÓN	- 67 -
FIGURA II.17 LLAMADA DE OTRO UID.....	- 67 -
FIGURA II.18 EJEMPLO DE LLAMADA DE OTRO UID.....	- 68 -
FIGURA II.19 LLAMADA A PARTIR DE OTRO UID.....	- 69 -
FIGURA II.20 EJEMPLO DE LLAMADA A PARTIR DE OTRO UID.....	- 69 -

FIGURA II.21 ESCOGER UN PROFESOR A PARTIR DE UNA ÁREA DE INVESTIGACIÓN: UID.....	- 73 -
FIGURA II.22 ESCOGER UN PROFESOR A PARTIR DE UN PROYECTO DE INVESTIGACIÓN: UID.....	- 74 -
FIGURA II.23 ESCOGER UN PROFESOR A PARTIR DE UN PARÁMETRO: UID PARAMETRIZADO	- 74 -
FIGURA II.24 SELECCIÓN DE UN CD DE MÚSICA: UID	- 75 -
FIGURA II.25 COMPRAR LOS CDs SELECCIONADOS: UID	- 76 -
FIGURA II.26 CONFIRMACIÓN DEL REGISTRO DE UN MIEMBRO DEL COMITÉ CON TRATAMIENTO DE ERRORES: UID	- 78 -
FIGURA II.27 RELACIONAMIENTO DE INCLUSIÓN ENTRE UIDS	- 80 -
FIGURA II.28 RELACIONAMIENTO DE EXTENSIÓN ENTRE UIDS	- 81 -
FIGURA II.29 RELACIONAMIENTO DE PRECEDENCIA ENTRE UIDS.....	- 81 -
FIGURA II.30 DIAGRAMA DE RELACIONAMIENTOS ENTRE UIDS SIMPLIFICADO	- 82 -
FIGURA II.31 DIAGRAMA DE RELACIONAMIENTOS ENTRE EL UID INICIAL	- 83 -
FIGURA II.32 REPRESENTACIÓN DE UNA CLASE	- 85 -
FIGURA II.33 REPRESENTACIÓN DE CLASE PROFESOR.....	- 85 -
FIGURA II.34 REPRESENTACIÓN DE CLASE ABSTRACTA PERSONA.....	- 86 -
FIGURA II.35 REPRESENTACIONES SIMPLIFICADAS DE UNA CLASE	- 86 -
FIGURA II.36 ATRIBUTO CON MÚLTIPLES PERSPECTIVAS CON PERSPECTIVA <i>DEFAULT</i>	- 88 -
FIGURA II.37 ATRIBUTO CON MÚLTIPLES PERSPECTIVAS SIN PERSPECTIVA <i>DEFAULT</i> ..	- 89 -
FIGURA II.38 FORMAS DE REPRESENTACIÓN DE GENERALIZACIÓN/ESPECIALIZACIÓN	- 93 -
FIGURA II.39 EJEMPLO DE GENERALIZACIÓN/ESPECIALIZACIÓN CON DISCRIMINADOR ..	- 93 -
FIGURA II.40 EJEMPLO DE GENERALIZACIÓN/ESPECIALIZACIÓN CON RESTRICCIONES ..	- 94 -
FIGURA II.41 HERENCIA DE UN ATRIBUTO CON MÚLTIPLES PERSPECTIVAS CON PERSPECTIVA <i>DEFAULT</i>	- 95 -
FIGURA II.42 HERENCIA DE UN ATRIBUTO CON MÚLTIPLES PERSPECTIVAS SIN PERSPECTIVA <i>DEFAULT</i>	- 95 -
FIGURA II.43 RELACIONAMIENTO UNARIO.....	- 97 -
FIGURA II.44 RELACIONAMIENTO BINARIO	- 97 -
FIGURA II.45 RELACIONAMIENTO TERNARIO	- 97 -
FIGURA II.46 RELACIONAMIENTO 1-N.....	- 98 -

FIGURA II.47 EJEMPLO DE NOMBRE DEL PAPEL.....	- 99 -
FIGURA II.48 CLASE DE RELACIÓN.....	- 99 -
FIGURA II.49 AGREGACIÓN.....	- 100 -
FIGURA II.50 COMPOSICIÓN	- 100 -
FIGURA II.51 DIAGRAMA DE OBJETOS	- 102 -
FIGURA II.52 CLASE DEFINIDA EN OTRO SUBSISTEMA	- 103 -
FIGURA II.53 DIAGRAMA DE SUBSISTEMAS	- 103 -
FIGURA II.54 ESQUEMA CONCEPTUAL	- 103 -
FIGURA II.55 REPRESENTACIÓN DE UN NODO.....	- 112 -
FIGURA II.56 REPRESENTACIÓN DEL NODO PROFESOR.....	- 112 -
FIGURA II.57 REPRESENTACIÓN DEL NODO ABSTRACTO PERSONA.....	- 113 -
FIGURA II.58 REPRESENTACIÓN DEL NODO DISCIPLINA ESTUDIANTE.....	- 113 -
FIGURA II.59 REPRESENTACIONES SIMPLIFICADAS DE UN NODO.....	- 113 -
FIGURA II.60 ATRIBUTOS MAPEADOS DEL ATRIBUTO CON MÚLTIPLES PERSPECTIVAS.....	- 115 -
FIGURA II.61 ANCLAS E ÍNDICE COMO ATRIBUTO.....	- 117 -
FIGURA II.62 FORMAS DE REPRESENTACIÓN DE GENERALIZACIÓN/ESPECIALIZACIÓN.....	- 119 -
FIGURA II.63 EJEMPLO DE GENERALIZACIÓN/ESPECIALIZACIÓN CON DISCRIMINADOR E RESTRICCIONES.....	- 120 -
FIGURA II.64 ENLACE BINARIO	- 121 -
FIGURA II.65 ENLACE UNARIO.....	- 121 -
FIGURA II.66 ENLACE BIDIRECCIONAL.....	- 122 -
FIGURA II.67 ENLACE 1-N	- 122 -
FIGURA II.68 ENLACE CON ORDENACIÓN	- 123 -
FIGURA II.69 EJEMPLO DE NOMBRE DEL PAPEL EN UN ENLACE	- 124 -
FIGURA II.70 AGREGACIÓN DE NODOS.....	- 125 -
FIGURA II.71 COMPOSICIÓN DE NODOS	- 125 -
FIGURA II.72 DIAGRAMA DE OBJETOS	- 127 -
FIGURA II.73 ESQUEMA NAVEGACIONAL	- 127 -
FIGURA II.74 REPRESENTACIÓN DE UN CONTEXTO	- 130 -
FIGURA II.75 EJEMPLO DE UN CONTEXTO.....	- 131 -
FIGURA II.76 CONTEXTO CON ELEMENTOS DE VARIAS CLASES	- 132 -
FIGURA II.77 CONTEXTO DE INSTANCIAS.....	- 132 -

FIGURA II.78 CONTEXTO DE NAVEGACIÓN POR CONSULTA.....	- 133 -
FIGURA II.79 GRUPO DE CONTEXTO	- 134 -
FIGURA II.80 EJEMPLO DE GRUPO DE CONTEXTO.....	- 135 -
FIGURA II.81 PASO DE PARÁMETROS PARA UN CONTEXTO Y GRUPO DE CONTEXTO..	- 135 -
FIGURA II.82 CONTEXTO DE NAVEGACIÓN DINÁMICO	- 137 -
FIGURA II.83 CONTEXTO DE CREAR.....	- 138 -
FIGURA II.84 GENERALIZACIÓN DE CONTEXTOS.....	- 140 -
FIGURA II.85 ESTRUCTURA DE ACCESO PROFESORES	- 145 -
FIGURA II.86 ESTRUCTURA DE ACCESO CON MÚLTIPLES CRITERIOS DE ORDENACIÓN. -	145
-	
FIGURA II.87 ÍNDICE CON UN CONTEXTO COMO DESTINO	- 146 -
FIGURA II.88 ÍNDICE CON UN CONTEXTO COMO DESTINO	- 146 -
FIGURA II.89 ÍNDICE CON VARIOS CONTEXTOS COMO DESTINO.....	- 147 -
FIGURA II.90 ESTRUCTURA DE ACCESO JERÁRQUICA	- 147 -
FIGURA II.91 CLASE EN CONTEXTO PROFESOR POR ÁREA.....	- 151 -
FIGURA II.92 ESQUEMA DE CONTEXTOS DE NAVEGACIÓN	- 152 -
FIGURA II.93 ADV PERSONA.....	- 158 -
FIGURA II.94 DIAGRAMA DE CONFIGURACIÓN.....	- 160 -
FIGURA II.95 ADVCHART LABORATORIO.....	- 161 -

CAPÍTULO I

I.- INGENIERÍA WEB Y USABILIDAD

1.1.- INGENIERÍA WEB

1.1.1.- INTRODUCCIÓN

Pocos pueden discutir que Internet y la World Wide Web están cambiando nuestras vidas. Cada día es más común que tareas tales como la lectura del periódico, la compra de libros o discos, operaciones bancarias, reserva de hoteles, compra de billetes de avión o tren, entre otras muchas, las realicemos conectados con nuestro ordenador a Internet. Es así que, durante la última década hemos asistido al crecimiento vertiginoso del desarrollo y uso de aplicaciones y sistemas Web cada vez más complejos y sofisticados.

Desafortunadamente, dicha complejidad no parece estar acompañada de los mecanismos adecuados que garanticen la calidad de unos sistemas de los que cada día tenemos mayor dependencia a nivel social, funcional y económico. Esta carencia de calidad ha venido generando una preocupación creciente entre la comunidad científica y técnica involucrada en el desarrollo Web. Así pues, en los últimos años surgen varias iniciativas con el objetivo de poner cierto orden dentro de la maraña que estamos creando y en la que nos movemos habitualmente.

En 1998, Roger Pressman moderó una mesa redonda virtual con representantes la ingeniería software tradicional y del desarrollo software basado exclusivamente en Internet. El debate principalmente se centró en discutir si valía la pena aplicar un proceso de ingeniería a las aplicaciones con base en Internet, o

qué características tenían éstas que justificaran el no utilizarlo. La conclusión general fue que aplicar un proceso de ingeniería nunca es una mala idea pero que éste debería adaptarse a los requerimientos de cambio continuo y rapidez siempre presentes en el proceso de desarrollo *Web*. De iniciativas como ésta y de otras como la organización de congresos y talleres especializados en el desarrollo para la *Web*, surge el nacimiento de una nueva disciplina denominada *Ingeniería Web*. En este capítulo se pretende dar una visión general de lo que es la *Ingeniería Web*.

1.1.2.- DEFINICIÓN

Murugesan¹ promotor inicial del establecimiento de la *Ingeniería Web* como nueva disciplina, da la siguiente definición:

“Como el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas *Web* de alta calidad”.

Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar que tipo de proceso es más adecuado en función de las características del mismo.

¹ S. Murugesan, “Web Engineering: A New Discipline for Development of Web-Based Systems.” 2001

1.1.3.- ESTÁNDARES Y TECNOLOGÍAS

1.1.3.1.- Ingeniería de Software Convencional

1.1.3.1.1.- Estándares

Tabla I.1 Los Estándares del Software

CÓDIGO	TÍTULO
IEEE J-STD-016	Procesos de ciclo de vida del software. Desarrollo de software. Acuerdos cliente/proveedor. Estándar provisional de EIA/IEEE.
IEEE 1016-1	Práctica recomendada para descripciones de diseño de software
IEEE 1042	Guía para la gestión de configuraciones del software
IEEE 1074	Estándar de IEEE para el desarrollo de procesos del ciclo de vida del software
IEEE 1209:1993	IEEE 1209:1993 Prácticas recomendadas para la evaluación y selección de herramientas CASE
IEEE 830-1993	Práctica recomendada para la especificación de requisitos de software
IEEE 982.1-1988	Diccionario de medidas estándar para la producción de software fiable
ISO/IEC 14102	Tecnologías de la información - Líneas de orientación para la evaluación y selección de herramientas CASE
ISO/IEC 12207	Tecnologías de la información - Procesos del ciclo de vida del software
ISO/IEC TR 9294	Tecnologías de la información - Líneas de orientación para la gestión de la documentación del software.
ISO/IEC 15504-1	Tecnologías de la Información. Evaluación de Procesos de Software Parte 1
ISO/IEC 2382	Tecnologías de la información – Vocabulario
ISO 8402	Gestión de la calidad y garantía de la calidad - vocabulario
ISO 9000-3 EN 29000-3	Normas de gestión de la calidad y garantía de la calidad: Parte 3: Orientaciones para la aplicación de la Norma ISO 9001 al desarrollo, suministro y mantenimiento del software

1.1.3.1.2.- Tecnologías

Definimos tecnología de software como un conjunto integrado de notaciones, herramientas y métodos, basados en unos sólidos fundamentos, que permiten el desarrollo de un producto software en un contexto organizativo dado.

A continuación se mencionan algunas herramientas como: DOORS para la gestión de requisitos, Rational Rose para el análisis y diseño de software; entornos de programación como Visual Studio y JBuilder; herramientas de pruebas y validación como J Test y C++ Test; y herramientas para la gestión de plazos y costes como Artemis Project View y Artemis Track View.

1) Herramientas para análisis

Estas herramientas ayudan a los especialistas en sistemas a documentar un sistema existente, ya sea manual o automatizado. También sirve para determinar los requerimientos de una nueva aplicación. Incluye:

- Herramientas para recolección de datos: capturan detalles que describen sistemas y procedimientos en uso. Documentan procesos y actividades de decisión, se utilizan para apoyar la tarea de identificar requerimientos.

- Herramientas para diagramación: crean representaciones gráficas de sistemas y actividades. Apoyan el dibujo y revisión de diagramas de flujos de datos e iconos asociados con el análisis estructurado. Incluyen programas para representación en diagramas de flujo.

- Herramientas para el diccionario: registran y mantienen descripciones de los elementos del sistema, como grupo de datos, procesos, alimentos de datos, etc. Frecuentemente proporcionan la capacidad de examinar las descripciones del sistema, para decidir si son incompletas o inconsistentes.

2) Herramientas para diseño

Apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos deseados durante las actividades de análisis.

Incluye:

- Herramienta de especificación: apoyan el proceso de formular las características, como por ejemplo deben tener una aplicación como entradas, salidas, procesamientos específicos de control.

- Herramienta para presentación: se utilizan para describir la posición de datos, mensajes, y encabezados sobre las pantallas de las terminales, informes y otros medios de entradas y salidas.

3) Herramientas de tipo back-end

Su finalidad es ayudar al analista a formular la lógica del programa, los algoritmos de procesamiento y la descripción física de datos. También ayudan a la intersección con los dispositivos (para entrada y salida). Estas actividades convierten los diseños lógicos del software en un código de programación; este es que da existencia a la aplicación.

4) Herramientas para el desarrollo

Ayudan al analista a trasladar los diseños en aplicaciones funcionales. Incluye:

- Herramientas para ingeniería Software: apoyan el proceso de formular diseños de software, incluyendo procesamientos y controles.

- Generadores de códigos: producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas

- Herramientas para pruebas: apoyan la fase evaluación de un sistema. Incluyen facilidades para examinar la correcta operación del sistema.

5) Herramientas integrales

Proporcionan un ambiente que automatiza tareas claves a lo largo del proceso de desarrollo. Estas herramientas facilitan el diseño, administración y mantenimiento del código. Brinda un ambiente eficiente para crear, almacenar, manipular y documentar sistemas.

1.1.3.2.- Ingeniería de Software orientado al Web

1.1.3.2.1.- Estándares

Los Estándares Web están pensados para ser una base común, un cimiento para la World Wide Web de modo que los navegadores y demás software entiendan el mismo vocabulario básico.

La W3C y otras organizaciones estándar han establecido tecnologías para crear e interpretar el contenido Web. Los estándares actuales son:

1) Lenguajes Estructurales

XHTML (eXtensible HyperText Markup Language) 1.0

XHTML 1.1

XML (eXtensible Markup Language) 1.0

2) Lenguajes de Presentación

CSS (Cascading Style Sheets) Nivel 1

CSS Nivel 2

CSS Nivel 3

3) Modelo de Objetos

DOM (Document Object Model) Nivel 1 (Núcleo)

DOM Nivel 2

4) Lenguajes de Scripting

ECMAScript 262 (versión estándar de JavaScript)

5) Lenguajes Adicionales de Presentación (Formato del documento)

MathML (Mathematical Markup Language) 1.01

MathML 2.0

SVG (Scalable Vector Graphics) 1.0

1.1.3.2.2.- Tecnologías

Esta sección trata los diferentes lenguajes y tecnologías, que son usadas hoy en día en la Web, para generar contenidos para navegadores Web.

1) Bases de Datos

Tecnologías de bases de datos: Oracle, Informix, Mysql, SQLServer, Sybase entre otras.

2) Html

Son las siglas de (Hyper Text Markup Language), es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Internet Explorer o Netscape).

3) Hojas de estilo en cascada CSS

Una hoja de estilo es un conjunto de instrucciones que definen la apariencia de diversos elementos de un documento HTML (también XML). En otras palabras una hoja de estilo nos permite indicar por ejemplo el tamaño de la fuente, color y estilo de cierto párrafo, mediante la definición de estilos y aplicación de los mismos.

Las hojas de estilo en cascada están muy relacionadas con el DOM (document object model) que define como representar mediante objetos los elementos de un documento HTML.

Un hecho importante más relacionado con las hojas de estilo es que a través de ellas se define la implementación estándar del DHTML o HTML dinámico, ya que las hojas de estilo son el puente del HTML con el DOM y los lenguajes de programación Web como Javascript.

4) JavaScript

Netscape en un esfuerzo por extender la funcionalidad de su navegador (browser), desarrollo un lenguaje de programación que se puede colocar dentro de archivos HTML. Originalmente fue llamado LiveScript, pero después fue renombrado a JavaScript con la idea de capitalizar la fama de Java, lenguaje desarrollado por Sun Microsystems.

JavaScript fue diseñado para ser un lenguaje de elaboración de scripts que pudieran incrustarse en archivos HTML. No es compilado, sino que, en vez de ello, es interpretado por el navegador. A diferencia de Java, que primero es convertido a código de byte fácil de interpretar, JavaScript es leído por el navegador como código fuente. Esto facilita el aprendizaje de JavaScript mediante ejemplos, debido a que se puede ver la manera en que otros usan JavaScript en sus páginas.

5) Modelo de Objetos de Documento (DOM)

El Modelo de Objetos de Documento o DOM, es la interfaz que permite acceder y manipular, mediante la programación, los contenidos de una página Web (o documento). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las propiedades de dichos objetos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiéndote crear contenido dinámico.

El DOM proporciona también una interfaz para trabajar con eventos, permitiendo capturar y responder a las acciones del usuario o del navegador.

6) Interfaz CGI

El cliente (navegador) Web accede a páginas estáticas. Para que el usuario pueda interactuar con ella y que se puedan construir páginas en función de sus necesidades, se puede optar por dos alternativas:

- Programas ejecutados en el cliente, como JavaScript.
- Aplicaciones ejecutadas en el servidor y que devuelven páginas HTML, gráficos, etc. como pueden ser los formularios (páginas) o los contadores (imágenes).

La interfaz CGI es un mecanismo que permite la invocación de una aplicación ejecutada en el servidor desde una página HTML que se está viendo en un cliente. Tradicionalmente la aplicación devolverá una página HTML al cliente o recogerá datos que el cliente le envió.

7) Perl

El lenguaje Perl (Practical Extraction and Report Lenguaje) nació en entornos UNIX para la realización de tareas administrativas. Es muy potente manipulando ficheros y cadenas de texto, ya que en este entorno la mayoría de ficheros de

configuración del sistema, de las aplicaciones, etc. son archivos de texto, y a menudo de un tamaño considerable

Es un lenguaje interpretado, con una sintaxis similar a C. En la actualidad, El intérprete de Perl es gratuito, y está portado a multitud de plataformas.

Debido a estas características, se asocia, sobre todo, a la programación en entornos Web usando la interfaz CGI (Common Gateway Interface), ya que en estos entornos se conectan plataformas heterogéneas, tanto software como hardware, y es más fácil codificar una aplicación en un lenguaje que pueda ser ejecutado en cualquier máquina sin tener que ser recompilado, por lo que Perl sale ganando al poder ejecutarse en distintas máquinas con la única condición de que tenga instalado su intérprete correspondiente.

8) Python

Es un lenguaje interpretado, orientado a objetos de propósito general. Python permite mantener de forma sencilla interacción con el sistema operativo, y resulta muy adecuado para manipular archivos de texto. Esta característica hace que muchas distribuciones de GNU/Linux utilicen Python para sus herramientas de configuración. También es ampliamente utilizado en la Web Python es multiplataforma y podemos descargar el intérprete para cualquier máquina.

9) Php

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor y permite crear páginas dinámicas HTML. Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear tus aplicaciones para Internet.

Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server).

10) ActiveX

Es una tecnología introducida por Microsoft para ejecutar códigos de programas en páginas WWW y la pretensión de ser una alternativa o competencia para Java. ActiveX es una definición general para diferentes componentes de software. Todos esos componentes se basan en el llamado Component Object Model (COM). ActiveX no es sin embargo un estándar en Internet, sino el intento de aprovechar algunas cualidades del sistema operativo MS-Windows para páginas WWW.

11) Macromedia Studio

Incluye las siguientes herramientas de desarrollo, potentes y fáciles de usar, y compatibles con una variada gama de tecnologías y estándares Web, que permiten crear aplicaciones dinámicas de Internet:

- Dreamweaver es la mejor elección para los profesionales que se dedican a crear y administrar sitios Web basados en HTML.
- Flash es la solución más potente para desarrollar contenido y aplicaciones dinámicas de Internet para varias plataformas y varios dispositivos mediante Macromedia Flash Player.
- Fireworks es un entorno de producción potente y flexible para crear, editar y animar gráficos Web.
- FreeHand es una aplicación de dibujo vectorial que permite crear ilustraciones de alta calidad para material impreso y para Web.
- ColdFusion es un entorno de scripting de servidor rápido que permite crear aplicaciones dinámicas de Internet.

12) Microsoft .NET

Es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser

suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

Esta es la plataforma .NET, y a los servicios antes comentados se les denomina servicios Web.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), Microsoft ha publicado el denominado kit de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio.NET, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.

Podemos destacar aplicaciones desarrolladas por Microsoft tales como Windows.NET, Hailstorm, Visual Studio.NET, MSN.NET, Office.NET, y los nuevos servidores para empresas de Microsoft (SQL Server.NET, Exchange.NET, etc.)

- Define un Common Language Runtime (CLR) y un IL (Intermediate Language) al que todos los lenguajes .NET compilan.
 - Idea similar a la máquina virtual de Java y a los bytecodes generados por el compilador de Java, respectivamente.
- Lenguajes
Visual Basic .NET, Visual C++ .NET, Visual C# .NET, Visual J# .NET, etc.
- Tecnologías
 - ADO.NET, ASP.NET, COM+: similares en concepto a JDBC, JSP y EJB, respectivamente.
 - Son una mejora de sus versiones anteriores (ADO, ASP, COM, etc.).
 - APIs para XML.

13) ADO.NET

ADO.NET es una mejora evolutiva de Microsoft ActiveX Data Objects (ADO) que aporta interoperabilidad entre plataformas y un acceso a datos escalable.

Puesto que utiliza XML (Lenguaje de marcado extensible), ADO.NET garantiza la transferencia eficaz de los datos a cualquier aplicación que se ejecute en cualquier plataforma.

14) ASP.NET

ASP.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

15) Modelo de objeto de componente (COM+)

COM+ es modelo de componentes de Microsoft ya que permite interconectar componentes software y aplicaciones de una manera "simple", segura y homogénea. COM+ es la nueva versión de COM que Microsoft incluye en .NET.

En .NET la utilización de COM+ es muy simple ya que todos los objetos son componentes COM+ por defecto. Por otra parte, utilizar dentro de nuestras aplicaciones .NET MS Word, PowerPoint o Corel Draw es muy fácil ya que la integración entre .NET y estas herramientas es perfecta.

16) Java 2 Enterprise Edition (J2EE)

J2EE define un estándar para el desarrollo de aplicaciones empresariales multicapa. J2EE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchos de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

La arquitectura de J2EE se puede dividir en cinco partes:

- El lenguaje de programación java.
- El modelo de programación del cliente.
- La infraestructura de la capa de middleware.
- La API de negocios para los programadores.
- La API no visible para los programadores.

La mayor parte de las abstracciones de las APIs corresponden a interfaces y clases abstractas.

17) Java DataBase Connectivity (JDBC)

JDBC (Conectividad de Base de Datos) es una interfaz que provee comunicación con bases de datos. Consiste de un conjunto de clases e interfaces escritas en Java, que proveen una API (Interfaz de Programación de Aplicación) estándar para desarrolladores de herramientas de base de datos, permitiendo independizar la aplicación de la base de datos que utiliza.

La API JDBC es la interfaz natural a las abstracciones y conceptos básicos de SQL (Lenguaje de Consultas Simple): permite crear conexiones, ejecutar sentencias SQL y manipular los resultados obtenidos.

18) Java Server Pages (JSP)

JSP (Páginas de Servidor Java) provee a los desarrolladores de Web de un entorno de desarrollo para crear contenidos dinámicos en el servidor usando plantillas HTML y XML (Lenguaje de Marcado Extensible), en código Java, encapsulando la lógica que genera el contenido de las páginas. Cuando se ejecuta una página JSP es traducida a una clase de Java, la cual es compilada para obtener un servlet. Esta fase de traducción y compilación ocurre solamente cuando el archivo JSP es llamado la primera vez, o después de que ocurran cambios.

JSP y XML tienen una interesante relación. Así como pueden generarse páginas HTML dinámicas a partir de una fuente en JSP, pueden generarse dinámicamente en forma análoga documentos XML.

19) JavaBeans Enterprise (EJB)

La especificación de JavaBeans Enterprise define una arquitectura para un sistema transaccional de objetos distribuidos basado en componentes. La especificación manda un modelo de programación; es decir, convenciones o protocolos y un conjunto de clases e interfaces que crean el API EJB. Este modelo de programación proporciona a los desarrolladores de Beans y a los vendedores de servidores EJB un conjunto de contratos que definen una plataforma de desarrollo común. El objetivo de estos contratos es asegurar la portabilidad a través de los vendedores y el soporte de un rico conjunto de funcionalidades.

Los Beans Enterprise son componentes de software que se ejecutan en un entorno especial llamado un **contenedor EJB**. El contenedor contiene y maneja un Bean enterprise de igual forma que el Servidor Web Java contiene un servlet.

Un Bean Enterprise no puede funcionar fuera de un contenedor EJB. El contenedor EJB controla cada aspecto del bean enterprise en tiempo de ejecución

incluyendo accesos remotos al bean, seguridad, persistencia, transacciones, concurrencia, y accesos a un almacén de recursos.

20) Servlets

Un servlet es una clase Java, embebida dentro del Web server, y utilizada para extender la capacidad del servidor. La API de servlets provee clases e interfaces para responder a cualquier tipo de requerimientos; en particular, para las aplicaciones que corren en servidores Web, la API define clases de servlet específicas para requerimientos HTTP.

No necesitan ser ejecutados como nuevos procesos dado que corren directamente en el Web server. Viven entre sesiones y se puede decir que son persistentes, no es necesario crear un servlet por cada requerimiento realizado desde el cliente, sino que corren dentro de éste múltiples hilos.

Los servlets son programas Java que proveen la funcionalidad de generar dinámicamente contenidos Web. Pueden ser ejecutados a través de una línea de comando. A diferencia de los applets, no poseen restricciones en cuanto a seguridad. Tienen las propiedades de cualquier aplicación Java y pueden acceder a los archivos del servidor para escribir y leer, cargar clases, cambiar propiedades del sistema, etc. Del mismo modo que las aplicaciones de programas Java, los servlets están restringidos por los permisos del sistema.

21) eXtensible Markup Language (XML)

La familia XML es un conjunto de especificaciones que conforman el estándar que define las características de un mecanismo independiente de plataformas desarrollado para compartir datos.

Se puede considerar a XML como un formato de transferencia de datos multi-plataforma. Es un subconjunto de SGML (Lenguaje de Marcado Generalizado Standard) y uno de sus objetivos es permitir que SGML genérico pueda ser

servido, recibido y procesado en la Web de la misma manera que actualmente es posible con HTML.

XML ha sido diseñado de tal manera que sea fácil de implementar. No ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

XML hace uso de etiquetas (únicamente para delimitar datos) y atributos, y deja la interpretación de los datos a la aplicación que los utiliza. Por esta razón se van formando lenguajes a partir del XML, y desde este punto de vista XML es un metalenguaje.

El conjunto de reglas o convenciones que impone la especificación XML permite diseñar formatos de texto para los datos estructurados, haciendo que se almacenen de manera no ambigua, independiente de la plataforma y que en el momento de la recuperación se pueda verificar si la estructura es la correcta.

Para comprobar que los documentos estén bien formados se utiliza un DTD (Definición de Tipo de Documento). Se trata de una definición de los elementos que pueden incluirse en el documento XML, la relación entre ellos, sus atributos, posibles valores, etc. Es una definición de la gramática del documento, es decir, cuando se procesa cualquier información formateada mediante XML, el primer paso es comprobar si está bien formada, y luego, si incluye o referencia a un DTD, comprobar que sigue sus reglas gramaticales.

22) eXtensible Stylesheet Language (XSL)

XSL (Lenguaje de Hojas de Estilo Extensible) es una especificación desarrollada dentro del W3C (World Wide Web Consortium) para aplicar formato a los documentos XML de forma estandarizada.

Aunque se ha establecido un modo para que puedan usarse hojas de estilo CSS (Hojas de Estilo en Cascada) dentro de documentos XML, es lógico pensar

que para aprovechar las características del nuevo lenguaje hace falta tener un estándar paralelo y similar asociado a él.

Según el W3C, XSL es "un lenguaje para transformar documentos XML", así como un vocabulario XML para especificar semántica de formateo de documentos. Además del aspecto que ya incluía CSS referente a la presentación y estilo de los elementos del documento, añade una pequeña sintaxis de lenguaje de comandos para poder procesar los documentos XML de forma más cómoda, la XSL permite añadir lógica de procesamiento a la hoja de estilo.

La idea es asociar al documento XML con una hoja de estilo y a partir de esto visualizar el documento XML en cualquier plataforma: PalmPC, PC, Internet Explorer, Netscape, etc. y con el aspecto (colores, fuentes, etc.) que se quiera utilizar.

La gran ventaja de utilizar XML y XSL es que los datos y la presentación de estos quedan en dos archivos diferentes.

1.1.4.- GESTIÓN DEL CONOCIMIENTO EN LA WEB

Es el proceso sistemático de detectar, seleccionar, organizar, filtrar, presentar y usar la información por parte de los participantes de la organización, con el objeto de explotar cooperativamente los recursos de conocimiento basados en el capital intelectual propio de las organizaciones, orientados a potenciar las competencias organizacionales y la generación de valor.

La Gestión del Conocimiento es, en definitiva, la gestión de los activos intangibles que generan valor para la organización. La mayoría de estos intangibles tienen que ver con procesos relacionados de una u otra forma con la captación, estructuración y transmisión de conocimiento. Por lo tanto, la Gestión del Conocimiento tiene en el aprendizaje organizacional su principal herramienta.

La diferencia entre dato, información y conocimiento. Una primera aproximación podría ser la siguiente: los datos están localizados en el mundo y el conocimiento está localizado en agentes (personas, organizaciones), mientras que la información adopta un papel mediador entre ambos conceptos.

Hay que reconocer que, en realidad, lo que fluye entre agentes distintos nunca es conocimiento como tal, sino datos (información). Es posible aproximar el conocimiento de dos agentes que comparten los mismos datos, pero debido a sus experiencias anteriores y a las diferencias en el modo de procesar los datos (modelos mentales, modelos organizacionales), nunca tendrán las mismas tendencias para la acción, ni estados idénticos de conocimiento. Sólo podemos conseguir aproximaciones, ya que el contexto interno y externo de un agente siempre es diferente a otro. Esto es así, porque el conocimiento es información puesta dentro de un contexto (experiencia).

En definitiva, los datos, una vez asociados a un objeto y estructurados se convierten en información. La información asociada a un contexto y a una experiencia se convierte en conocimiento. El conocimiento asociado a una persona y a una serie de habilidades personales se convierte en sabiduría, y finalmente el conocimiento asociado a una organización y a una serie de capacidades organizativas se convierte en Capital Intelectual.

1.1.5.- ARQUITECTURA DE LA INFORMACIÓN WEB

Arquitectura de la Información (AI) se refiere al diseño, organización, etiquetado, navegación y sistemas de búsqueda que ayudan a los usuarios a encontrar y gestionar la información de manera efectiva.

Necesidades mínimas de organización y funcionamiento de un sitio Web:

- Arquitectura Funcional

- Arquitectura Estructural
- Arquitectura Navegacional

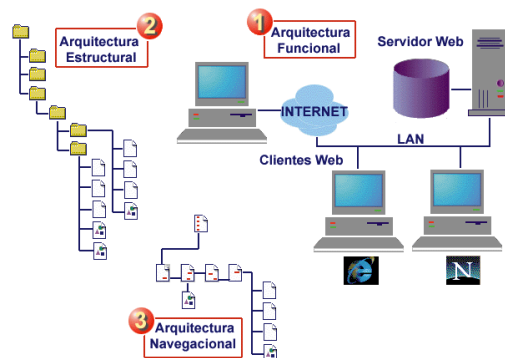


Figura I.1 Fases de la Arquitectura de la información Web



Figura I.2 “Pirámide de necesidades” para construir un sitio Web

1.1.5.1.- Organización de la información

El objetivo de la organización de la información es diseñar una serie de estructuras y esquemas de acceso e interrelación de contenidos para ayudar a los usuarios a encontrar respuestas a sus necesidades de información, con los siguientes pasos:

1.1.5.1.1.- Dividir en unidades lógicas (los trozos de información)

Trozo de información.- Concepto flexible acorde con el sentido común, organización lógica y conveniencia del usuario. Naturaleza del contenido debería sugerir la mejor manera de subdividir y organizar la información. Como por ejemplo: La información en trozos cortos para poder leer en pantalla ó pueden ser más largos para facilitar impresión.



Figura I.3 División de la información en unidades lógicas

1.1.5.1.2.- Establecer jerarquía de importancia y generalidad entre esas unidades

Determinar las estructuras básicas de navegación.



Figura I.4 Jerarquía de la información

1.1.5.1.3.- Usar esa jerarquía para **ESTRUCTURAR** las relaciones entre los trozos de información

Los usuarios construyen modelos mentales acerca de la organización de la información, y luego usan esos modelos para encontrar cosas que no han visto aun.

El éxito de un sitio depende de cuanto coincida la organización de la información con el modelo formado en la mente del usuario.

Una organización lógica de la información permite a los usuarios hacer predicciones exitosas acerca de donde encontrar las cosas que busca.

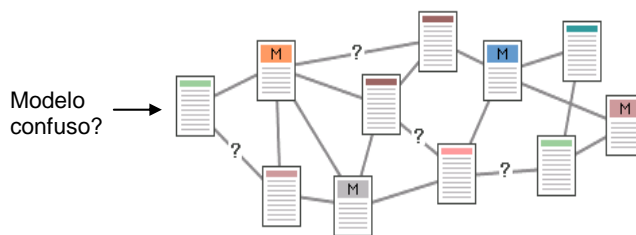


Figura I.5 Estructurar las unidades lógicas de información

1.1.5.1.4.- Estudiar, probar, y analizar el éxito (o fracaso) estético y funcional del sitio

Mediante pruebas y seguimientos se comprobará el éxito o fracaso del sitio.

1.1.6.- MÉTRICAS WEB

1.1.6.1.- Definición

Una métrica (medida) es un valor numérico o nominal asignado a características o atributos de un ente computado a partir de un conjunto de datos observables y consistentes con la intuición. Una métrica puede ser directa o indirecta, interna o externa, objetiva o subjetiva.

1.1.6.2.- Clasificación de métricas

Métrica Directa: Es el resultado de una correspondencia o mapeo directo entre un atributo de un ente (del dominio empírico) y el valor (del dominio numérico), es decir, va desde un atributo a un número y sirve como referencia para describir y evaluar aspectos o situaciones del mundo empírico. Los atributos se miden generalmente por medio de métricas directas.

Métrica Indirecta: Es el resultado de una correspondencia o mapeo entre relaciones de dos o más atributos, o de un atributo compuesto (del dominio empírico) y el nuevo valor (del dominio numérico), y que sirve como referencia para describir y evaluar características y/ o subcaracterísticas y/ o atributos compuestos de un ente del mundo empírico. Las características y subcaracterísticas se miden, generalmente, por medio de métricas indirectas también denominadas métricas derivadas.

Métrica Interna: Es un valor numérico o nominal del atributo que siempre involucra al ente en sí, ya sea obtenido por una métrica directa o indirecta.

Métrica Externa: Es el valor resultante del atributo al aplicar una métrica indirecta y que siempre involucra al ente y su comportamiento con el entorno.

Métrica Objetiva: Es el valor resultante del atributo de un ente, comprobable, independiente del juicio y subjetividad humana. Sin embargo existen grados de objetividad.

Métrica Subjetiva: Es un valor numérico que siempre involucra al juicio humano por medio de heurísticas o criterios de preferencia directa.

1.1.6.3.- Descripción de algunas métricas

Métricas de éxito

La métrica más simple para medir la usabilidad de un sitio es la tarifa de éxito al realizar una tarea representativa: registre el porcentaje de usuarios que logró finalizar la tarea encomendada.

Es una métrica firme, pero constituye una estadística reveladora. Asignando peso 1 a la tarea finalizada, 0.5 a la tarea a medio terminar y 0 a la no terminada.

$$\text{Éxito} = (\text{N}^\circ \text{tareasterminadas}) + (\text{N}^\circ \text{ tareas a medias} * 0.5) * 100 / \text{N}^\circ \text{ total de tareas}$$

Satisfacción del usuario

Medida generalmente por las respuestas de un usuario a un cuestionario con declaraciones.

Métricas sobre el contenido del sitio. Lectura

- Tiempo en recordar la estructura de un sitio: segundos que tarda el usuario en realizar un mapa del sitio.
- Memoria del conocimiento: porcentaje basado en el número de respuestas correctas en un formulario sobre el contenido relleno por el usuario en relación al número total de preguntas.
 $(\text{N}^\circ \text{respuestas correctas} - \text{N}^\circ \text{respuestas incorrectas}) * 100 / \text{N}^\circ \text{total preguntas}$

Métricas de Contenido

- Legibilidad de texto.

Índice de legibilidad: $0.4 * (\text{n}^\circ \text{palabras} / \text{n}^\circ \text{líneas} + \text{porcentaje de palabras de 3 o más sílabas})$

Tiempos de respuesta

- 0.1 segundos es el límite para hacer sentir al usuario que la respuesta es inmediata.
- 1 segundo es el límite para que el usuario piense que no hay interrupción.
- 10 segundos es el límite máximo para mantener la atención del usuario.

1.2.- INGENIERÍA DE LA USABILIDAD

1.2.1.- GENERALIDADES

Los antecedentes de usabilidad vienen de la ingeniería y del diseño de interfaces, y el diseño Web ha tomado de allí elementos para definirla. Podríamos decir que se llama usabilidad al nivel de facilidad de uso que posee un sitio Web, lo “amigable” que resulta para el visitante. La usabilidad comprende, además, el estudio del comportamiento del usuario al navegar un sitio Web.

La usabilidad no se limita al diseño de las páginas que comprenden el sitio, sino también al diseño de los contenidos y a la simplicidad y economía de recursos en la elaboración de un proyecto para Internet. El disponer de últimas tecnologías no siempre garantiza la satisfacción del visitante y la fidelidad del mismo.

Existe una estadística reveladora en relación a la actualización del software, a las versiones de navegadores, plugins, etc. que indica que pasan 2 años hasta que el 100% de los usuarios han incorporado la última versión de los programas, algo así como un 1% de actualización semanal. Debemos tener en cuenta esto al pensar si usamos o no tal o cual versión de plugin, javascript, etc.

En la mayoría de los casos quienes desarrollan sitios en Internet están a la vanguardia en el proceso de actualización, y poseen instaladas las últimas versiones de navegadores y filtros, pero el “usuario común”, al que destinamos nuestro desarrollo pasará mucho tiempo hasta que incorpore esta nueva versión.

1.2.1.1.- Definiciones

La usabilidad es una característica de alto nivel que se puede medir mediante el cálculo a partir de métricas directas e indirectas y representa la capacidad o potencialidad del producto para ser utilizado, comprendido y operado por los

usuarios, además de ser atractivo. Incluye a subcaracterísticas como comprensibilidad, operabilidad y comunicatividad entre otras como estética y estilo que hacen que el producto software sea agradable de usar.

Podemos definir también la usabilidad como la medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

1.2.1.1.1.- Efectividad

Precisión y la plenitud con las que los usuarios alcanzan los objetivos especificados. A esta idea van asociadas la facilidad de aprendizaje (en la medida en que este sea lo más amplio y profundo posible), la tasa de errores del sistema y la facilidad del sistema para ser recordado (que no se olviden las funcionalidades ni sus procedimientos).

1.2.1.1.2.- Eficiencia

Recursos empleados en relación con la precisión y plenitud con que los usuarios alcanzan los objetivos especificados. A esta idea van asociadas la facilidad de aprendizaje (en tanto que supone un coste en tiempo; igualmente, si se requiere un acceso continuo a los mecanismos de ayuda del sistema), la tasa de errores del sistema y la facilidad del sistema para ser recordado (una asimilación inapropiada puede traducirse en errores de usuario).

1.2.1.1.3.- Satisfacción

Ausencia de incomodidad y la actitud positiva en el uso del producto. Se trata, pues, de un factor subjetivo.

1.2.1.2.- Situación actual

El sitio Web de una organización es una entrada a su información, productos y servicios. Como tal, debe idealmente ser una reflexión de las necesidades de los clientes que sirve.

Desgraciadamente, el diseño y el desarrollo del sitio Web son conducidos a menudo por la tecnología o por la estructura de organización u objetivos de negocio, más que por lo que el usuario necesita. En años recientes sin embargo, los dueños y diseñadores del sitio Web han comenzado gradualmente a reconocer la importancia de la usabilidad.

Dado el índice de crecimiento único de la naturaleza de la Web, está claro que la usabilidad está llegando a ser cada vez más importante para que los diseñadores del Web site la traten. Hasta cierto punto, los usuarios del Web site cuentan con la satisfacción inmediata.

1.2.1.3.- Errores más comunes de usabilidad

El 90% de los sitios Web presentan fallos de usabilidad. Los fallos más comunes son los siguientes:

- Indefinición del sitio (¿Qué es esto?)
Es bueno informar claramente que es lo que se puede encontrar aquí y cuales son las ventajas de utilizar esta página.
- Indefinición de las funciones a realizar (¿Dónde está lo que busca?)
Hay que indicar al usuario la ruta claramente de principio a fin. Si los caminos son complejos hay que simplificarlos.
- Enlaces perdidos
Los enlaces perdidos sólo demuestran un mal producto.

- Tiempo de respuesta
Se suele producir cuando la programación es redundante, servidores poco potentes o exceso de seguridad. No son justificables.
- Exceso de peso
Se resuelve quitando imágenes que no sean necesarias y dando el formato a la página utilizando los estándares de HTML.

1.2.1.4.- Principios o reglas de usabilidad

Según Jakob Nielsen² existen 4 reglas básicas sobre la usabilidad en la Web, a saber:

- En Internet el usuario es el que manda.
- En Internet la calidad se basa en la rapidez y la fiabilidad, cuenta que la página sea más rápida que bonita, fiable que moderna, sencilla que compleja.
- Seguridad, la confianza es algo que cuesta mucho ganar y se pierde con un mal enlace.
- Poner las conclusiones al principio, el usuario se sentirá más cómodo si ve las metas al principio. De esta forma no tendrá que buscar lo que necesita y perderá menos tiempo en completar su tarea. Si completa su tarea en menos tiempo se sentirá cómodo y quizás se dedique a explorar el sitio o quizás se lo recomiende a un amigo.

1.2.1.5.- Factores que influyen en la usabilidad

A continuación se tratan los factores más importantes que influyen en la usabilidad de un sitio.

² Emprendedor de la Ingeniería Web y Usabilidad: <http://www.useit.com>

1.2.1.5.1.- Diseño de Página

1) Espacio de pantalla

- Las páginas Web deben estar presididas por un contenido que despierte la atención del usuario.
- Como norma el contenido de interés para el usuario debe ocupar al menos la mitad del diseño de una página y, preferiblemente, hasta un 80%. La navegación debe mantenerse por debajo del 20% del espacio de las páginas de destino.
- Desde el punto de vista de la usabilidad sería mejor eliminar todos los elementos innecesarios tal como la publicidad, en caso de que tenga que haber publicidad considerarla dentro del espacio de navegación.

2) Vinculación

Los vínculos constituyen la parte más importante del hipertexto: conectan las páginas y permiten a los usuarios ir a nuevos sitios de la Web, entre los cuales se resalta los más importantes:

- Vínculos de navegación estructural: botones de las páginas de inicio y vínculos a páginas subordinadas a la actual.
- Vínculos asociativos del contenido de página: palabras subrayadas que señalan páginas con más información relacionada con la del texto.

3) Títulos de vínculos y directrices de los mismos

- Los títulos deben tener como máximo 60-80 caracteres.
- Colorear los vínculos, utilizar colores estándar. Generalmente los vínculos de páginas que el usuario no ha visto son azules y las que ya ha visitado aparecen en rojo o morado.

4) Hojas de Estilo

- Utilizarlas para separar presentación y contenido.

- Utilizar una sola hoja de estilos para todas las páginas de su sitio.

5) Imprimir

- Proporcionar versiones imprimibles de cualquier documento largo.
- La recomendación es generar dos tipos de versiones de todos los documentos extensos: una versión optimizada para su visualización en línea y otra optimizada para la impresión.
- El archivo imprimible debe estar en Postscript o PDF y debe existir siempre un vínculo con una versión HTML para su lectura en línea.
- Todo archivo que sirva para la impresión debe aceptar los dos formatos internacionales comunes: A4 y carta.

1.2.1.5.2.- Diseño de Contenido

1) Escribir para la Web

- Las dos directrices principales para escribir en la Web son:
 - Escribir para poder encontrar las cosas.
 - Utilizar hipertexto para dividir la información extensa en múltiples páginas.
- Que los textos sean breves.
- Lenguaje claro.

2) Títulos de página

- Títulos que tengan de 2 a 6 palabras, con no más de 60-80 caracteres.
- Páginas diferentes necesitan títulos diferentes.
- Optimizar los títulos para una búsqueda rápida.

3) Legibilidad

Reglas básicas que aseguran la legibilidad en un sitio Web:

- Utilizar colores de alto contraste entre el texto y el fondo, lo mejor es texto negro sobre fondo blanco.
- Fondos de colores claros o patrones de formato muy sutiles.
- Fuentes suficientemente grandes para buena lectura.
- Texto estático y preferiblemente alineado a la izquierda.
- Evitar el uso de mayúsculas en el texto.

4) Multimedia

Requiere una buena disciplina de diseño, el uso no restringido de multimedia confunde a los usuarios y dificulta la comprensión de la información.

5) Imágenes y fotografías

- Reducir al mínimo los gráficos de la página.
- Las páginas Web de alto nivel deben reducir el número de ilustraciones pero una página biográfica ha de llevar una fotografía del personaje.

6) Animación

Reducir el uso de la animación ya que distrae y molesta a los usuarios.

7) Vídeo

El uso del video debe minimizarse debido a las restricciones del ancho de banda.

8) Audio

En términos de usabilidad, el sonido de buena calidad mejora la experiencia del usuario por lo que es necesario invertir en sonido de calidad profesional.

1.2.1.5.3.- Diseño del sitio

Desde el punto de vista de la usabilidad el diseño de sitio es más importante que el diseño de página.

El diseño de los sitios debe orientarse ante todo a la simplicidad, con pocas distracciones y con una arquitectura de la información muy clara con herramientas de navegación adecuadas.

1) La página de inicio

- Es el estandarte del sitio, debe ser diseñada de forma distinta al resto de páginas.
- Las páginas interiores y la página de inicio aunque diferentes deben tener el mismo estilo.
- Debe ofrecer tres cosas:
 - directorio de las principales áreas de contenido del sitio (esquema de navegación del sitio).
 - noticias o promociones especiales.
 - opción de búsqueda.

2) Navegación

Las interfaces de navegación tienen que ayudar a los usuarios a:
¿Dónde estoy? La ubicación actual debe mostrarse a dos niveles diferentes: en relación con la Web como un todo y en relación con la estructura del sitio.

3) Estructura de un sitio

Una arquitectura deficiente de la información siempre llevará a una deficiente usabilidad.

4) Opciones de búsqueda

La búsqueda debe resultar de fácil acceso desde cada una de las páginas del sitio.

1.2.2.- MODELO DE PROCESO PARA LA INGENIERÍA DE LA USABILIDAD

El Modelo de Proceso de la Ingeniería de la Usabilidad (MPIU) especifica una metodología que guía al equipo de desarrollo de aplicaciones interactivas con altos niveles de usabilidad.

Conseguir que las aplicaciones cumplan con todos los principios que permiten etiquetar a las mismas como “usables” es aún un proceso más laborioso si no se realiza siguiendo un disciplinado y riguroso procedimiento.

El MPIU tiene sus cimientos por una parte en la Ingeniería del Software (IS) y por otra en la disciplina de la Interacción Persona-Ordenador, la cual contribuye entre otras con toda una sólida base de conocimiento y un conjunto de técnicas y experiencias conocidas para el diseño de interfaces centrada en el usuario.

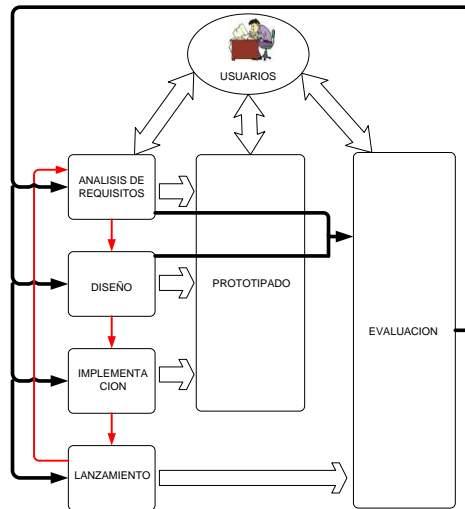


Figura I.6 Modelo de proceso para la Ingeniería de la Usabilidad³

A continuación se describen cada una de las fases del proceso de la Ingeniería de la Usabilidad:

1.2.2.1.- Análisis de Requisitos

Para que una aplicación finalice con éxito depende de manera crítica de esta fase y aunque este aspecto es ampliamente compartido por los desarrolladores de software lo cierto es que habitualmente esta fase no se realiza con el rigor que merece.

En el paradigma Web los requisitos se centran en los grandes temas el estudio de la audiencia y las necesidades de los usuarios.

³ Fuente: http://griho.udl.es/publicacions/2003/IU_sitios_Web.pdf

1.2.2.2.- Diseño

Durante la fase de análisis y especificación de los requisitos el objetivo es documentar lo que el sitio debe hacer.

En la fase de diseño es importante determinar como será el sitio estructuralmente y gráficamente. En esta parte presentamos dos aspectos importantes en el diseño de un sitio Web, el análisis de tareas y la arquitectura de la información.

1.2.2.3.- Prototipado

1.2.2.3.1.- Maquetas

Las maquetas son básicamente una sola página de representaciones estáticas del espacio de diseño. Se usan para mejorar el diseño y facilitar la comunicación entre los diseñadores, usuarios e implicados, consideraremos tres tipos:

1) Bocetos

Los bocetos son maquetas rápidas, pequeñas para desarrollar un amplio espectro de ideas de diseño. Se usan típicamente en la primera etapa del diseño, muchas veces de que se haya acabado la fase de análisis de requisitos. La clave de los bocetos es su velocidad.

2) Maqueta de papel

Son representaciones de una mayor cualidad que los bocetos. Permite explorar el diseño de la página, aspectos estéticos. Permite una comprensión más realista de los límites del tamaño de la página, del espacio de diseño, identificar posibles dificultades en el proceso de diseño y comenzar la evaluación con los usuarios.

3) Maqueta digital

El prototipo digital constituye un paso más en la elaboración de prototipos el cual es realizado con herramientas de diseño gráfico, siendo, por tanto más costosos y elaborados. Permiten afinar aspectos importantes del diseño cómo son los colores, los contrastes, las fuentes tipográficas, etc. que el prototipo de papel no proporciona.

4) Guión

Un guión indica la manera de convertir las ideas creativas del equipo en una presentación visual, en este caso las páginas. Un guión no muestra todas las páginas o elementos, pero si la mayoría de categorías de páginas. Un guión puede ser usado también para la reproducción de video, representar cuadros estáticos o para gráficos.

1.2.2.4.- Implementación

Dos aspectos primordiales deben considerarse, uno es el puramente tecnológico y otro es el de los contenidos.

El primero incluye los lenguajes de programación a utilizar, las hojas de estilos, e incluso el seguimiento de los estándares para disponer de páginas accesibles.

En cuanto a los contenidos debe tenerse en cuenta, es que escribir para la Web es totalmente distinto a hacerlo para otro medio, puesto que el texto se ha de redactar de tal manera que soporte las tareas y los objetivos del usuario y deberá adaptarse a la audiencia prevista. El proceso de escritura deberá, por tanto, estar orientado en función de los parámetros de lectura fácil, legibilidad, escaneabilidad (los navegantes no leen sino que escanean la información en busca de un contenido), la paginación, los enlaces y los principales hitos de cada página. Esta

conducta a la hora de leer condicionara la manera en qué deberemos escribir para la Web.

Utilizaremos el modelo de escritura en pirámide invertida, que consiste en empezar cada página por la conclusión. El usuario debe poder discernir al primer vistazo si aquello que hay dentro le interesa lo suficiente cómo para continuar leyendo.

1.2.2.5.- Lanzamiento

Todo lo realizado anteriormente entra en escena definitivamente en esta fase, es el momento en que todo el trabajo se ha desarrollado y se pondrá a disposición del usuario final.

1.2.2.6.- Evaluación

A continuación se describen algunos métodos de evaluación:

1.2.2.6.1.- Evaluación heurística

La evaluación heurística fue desarrollada por Nielsen y Molich⁴. La evaluación heurística consiste en analizar la conformidad de la interfaz con unos principios reconocidos de usabilidad (la “heurística”) mediante la inspección de varios evaluadores expertos.

⁴ Nielsen, J. and Molich, Evaluación Heurística de interfaces de usuario (1990).

1.2.2.6.2.- Recorrido de la usabilidad plural

Este método es creado por Bias ⁵ y fue desarrollado en los laboratorios IBM, comparte algunas características con los recorridos tradicionales pero tiene algunas características propias que lo definen.

Este método se realiza con tres tipos de participantes, usuarios representativos, desarrolladores y expertos en usabilidad, que conforman todos los actores implicados en el producto.

1.2.2.6.3.- Análisis de logs

Cada visitante que accede a nuestro sitio Web deja un rastro físico de su visita, estos rastros quedan almacenados en el servidor para poder ser consultados, filtrados y recopilados con la intención de obtener información de lo que está sucediendo.

El formato típico de un log es el de una cadena de texto que almacena la información sobre las peticiones que realiza el usuario en el servidor. Por tanto cuando un navegante accede a nuestra página de inicio, se escribirá una línea por cada elemento que solicite, es decir una línea al solicitar la página de inicio, otra para una imagen y así sucesivamente para cada elemento de la página solicitada.

Por tanto efectuando un análisis de logs correcto dispondremos de una información muy valiosa que nos permitirá desde detectar problema de usabilidad hasta determinar perfiles de usuario.

⁵ Bias, R and Rietmeyer, P.B. Usability Support Inside and Out. interactions ACM Press (1995)

1.3.- MULTIMEDIA WEB Y DISEÑO DE INTERFACES

1.3.1.- INTRODUCCIÓN A LA MULTIMEDIA WEB

La Multimedia se compone, de combinaciones entrelazadas de elementos de texto, gráficos, sonido y vídeo. Cuando se permite a un usuario, controlar ciertos elementos y cuándo éstos deben presentarse, se denomina Multimedia Interactiva. Cuando se proporciona una estructura de elementos ligados a través de los cuales el usuario puede navegar, entonces Multimedia Interactiva se convierte en Hipermedia.

Por lo anteriormente definido, el Web, es por naturaleza un Sistema Hipermedia, las características particulares de la tecnología Web hacen de Internet una tecnología excelente para el uso de elementos multimedios y además facilita acceso remoto a la información sin las limitaciones espacio temporales propios de otros sistemas de estas características, favoreciendo el incremento de la interactividad e integración y la actualización dinámica de los contenidos.

La mayoría de las páginas Web, se realiza de forma estática, esto hace suponer que no existe conocimiento de elementos multimedios o necesidad de utilizarlos apropiadamente, que permitan hacer que las páginas Web sean mejoradas y que se conviertan en interactivas y dinámicas.

1.3.2.- CONCEPTOS DE LOS ELEMENTOS MULTIMEDIALES

En Internet se obtiene grandes experiencias con las páginas Web, debido principalmente a los elementos multimedia. Los elementos Multimedia dentro de las páginas Web, que más se utilizan son: Hipertexto, Conectores, Scripts, Applets, Base de Datos, Gráficos y Animaciones (Tecnología de Gráficos), Sonido

y Vídeo (Tecnología de Audio y Vídeo), Programas de autor (Flash, Director, Toolbook y otros).

1.3.2.1.- Tecnología de Gráficos

La tecnología de gráficos se puede dividir en tres grupos: Las imágenes, las animaciones y la Realidad Virtual.

1.3.2.2.- Tecnología de Audio y Vídeo

1.3.2.2.1.- Sonido

Eguiluz (2000)⁶, clasifica el sonido en distintos tipos:

- a) Música, que se divide en Música de Fuente (la que se oiría si una escena estuviera ocurriendo) y Música de Fondo (apoya, comenta o refuerza el contenido).
- b) Efectos de sonido.
- c) Sonido ambiente.
- d) Voz, que podríamos dividir en Diálogo (que refleja lo que se supone están diciendo los personajes) y Voz en off o narración, que es una voz que aclara la imagen para el espectador.

1.3.2.2.2.- Vídeo

La información de vídeo, es provista en una serie de imágenes ó "cuadros" y el efecto del movimiento es llevado a cabo a través de cambios pequeños y continuos en los cuadros. Debido a que la velocidad de estas imágenes es de 30

⁶ Eguiluz, A; 2000: *Sonido*. Disponible: http://www2.eside.deusto.es/asignaturas/tim/TIM_Sonido.html

cuadros por segundo, los cambios continuos entre cuadros darán la sensación al ojo humano de movimiento natural. La velocidad actual de Internet, condiciona que los vídeos que pueden utilizarse tengan tamaño reducido. Difícilmente pueden emplearse videos de más de 320x240 píxeles de resolución, y generalmente predominan los de 160x120. Con estos tamaños y con elevadas compresiones pueden generarse incluso vídeos que se transmiten en tiempo real, especialmente para conexiones a 64 Kbps o preferiblemente a 128 Kbps.

1.3.3.- MULTIMEDIA DISTRIBUIDA

Las aplicaciones multimedia distribuidas pueden programarse utilizando tecnologías Java de objetos distribuidos (como RMI). Una aplicación multimedia de alta calidad y distribuida debe ser capaz de ofrecer una cierta calidad de salida, es decir, ofrecer una imagen a la salida cada cierto tiempo. La corrección en el procesamiento temporal de las imágenes depende no sólo de la gestión local del procesamiento sino también de la gestión llevada a cabo por los protocolos de red.

En este estudio se deberá estudiar la estructura de tecnologías Java para la integración eficiente de protocolos de red que permitan garantizar los plazos de tiempo que una aplicación debe cumplir en su procesamiento de datos multimedia.

1.3.4.- MÉTODO DE DESARROLLO DE SISTEMAS MULTIMEDIA EN LA WEB

La presente metodología describirá los principales pasos para el desarrollo de páginas Web utilizando multimedia, los cuales comprenden:

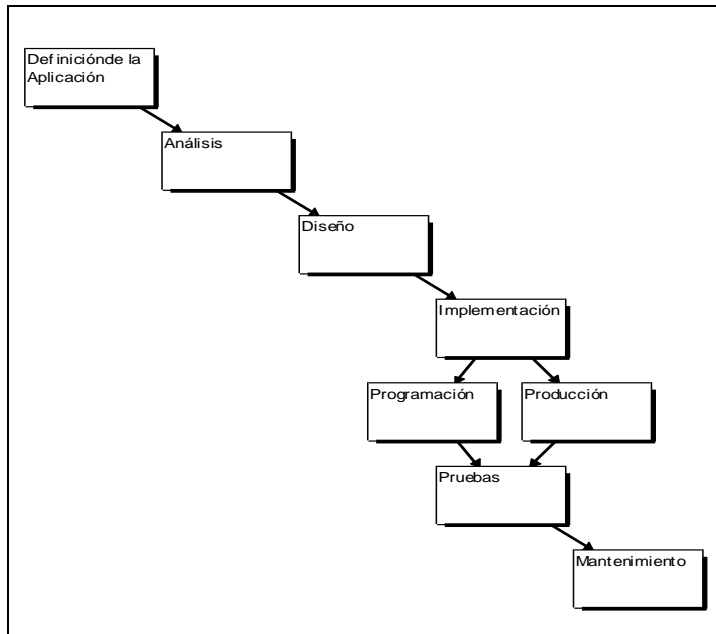


Figura I.7 Fases de la Metodología para el Diseño de Páginas Web Basadas en Multimedia

La metodología comprenderá varias fases como se observa en la [Figura I.7](#), cada fase estará compuesta de etapas y esta a su vez de pasos de ser necesario.

Se empieza con la definición de la aplicación, método por medio del cual se define la visión del sitio Web, es el punto de partida para desarrollar cualquier página Web. En la fase de análisis, se procede a realizar un estudio del objetivo de la publicación de la página y de su contenido, se realiza un estudio de viabilidad y un análisis económico para el desarrollo de la página llegando a fijar una fecha límite de entrega. De aquí se pasa a la fase de diseño, en la cual se obtiene un mapa de navegación y se define la interfase de usuario (boceto de páginas) para finalmente implementar cada una de las páginas y los elementos a ser utilizados en ellas.

La metodología también comprende las fases de pruebas y mantenimiento, por medio de las cuales se realiza la corrección de errores y las modificaciones necesarias para mejorar ya sea el rendimiento o la interfase del sistema.

A continuación se presenta un esquema de los componentes de la metodología, se compone de seis fases, cada una de las cuales contiene una serie de etapas necesarias para su desarrollo y genera un documento específico.

Tabla I.2 Fases y Etapas de la Metodología. Documentación relacionada

FASES	ETAPAS	DOCUMENTOS
1. Definición de la Aplicación	<ul style="list-style-type: none"> • Entrevista 	DEFINICIÓN DE LA APLICACIÓN Documento de descripción del sitio Web en términos del cliente.
2. Análisis	<ul style="list-style-type: none"> • Definición de Propósito. • División del contenido en asuntos principales. • Definición del público destino. • Estudio de viabilidad. • Análisis económico. 	ANÁLISIS DEL SISTEMA Documento que contiene la definición del propósito del sitio, los temas que compondrán la página, definición del público al que se dirige la página, la viabilidad del proyecto y un presupuesto de desarrollo.
3. Diseño	<ul style="list-style-type: none"> • Definición de la estructura de navegación. • Definición de elementos de enlace. • Definición de guiones 	DISEÑO Documento con el diseño de la arquitectura de navegación y de las páginas a ser implementadas.
4. Implementación	<ul style="list-style-type: none"> • Programación. • Producción. • Instalación del servidor Web. 	PÁGINAS WEB Resultado de la traducción del documento del diseño. Documentos publicados en el Internet utilizando el lenguaje HTML.
5. Pruebas	<ul style="list-style-type: none"> • Pruebas de los elementos multimedia. • Pruebas individuales. • Pruebas de integración 	REPORTES DE ERRORES Documentos que permitirán realizar la corrección de errores encontrados en la página luego de su implementación.
6. Mantenimiento	<ul style="list-style-type: none"> • Mantenimiento 	MANTENIMIENTO

	<p>Correctivo</p> <ul style="list-style-type: none"> • Mantenimiento Adaptativo • Mantenimiento Perfectivo 	<p>Documento que permite documentar el mantenimiento de la página, desde la petición de mantenimiento pasando por la presentación de informes y finalizando en la evaluación del mantenimiento realizado.</p>
--	--	---

A continuación se describen cada una de las fases para el Desarrollo de Páginas Web Basadas en Multimedia.

1.3.4.1.- Definición de la aplicación

El primer paso que lleva a cabo un cliente para obtener un sitio en el Web consiste en describir las propiedades del sitio *desde su punto de vista*. O bien el desarrollador podría pensar en algo interesante que publicar en el Web utilizando multimedia, de esta forma llamando irresistiblemente la atención del navegante.

1.3.4.2.- Análisis

En esta fase el desarrollador es responsable de decidir si la información a ser publicada puede ser o no presentada como parte de un sitio Web, si se cuenta con el equipo humano necesario o si se tendrá que capacitar a alguien, si existe el hardware y software indispensable para la producción de los elementos multimedia, en caso de ser necesario el analista deberá presentar un presupuesto de adquisición o de alquiler de equipos de computación así como de salarios para el equipo humano y presentar un plan del proyecto a desarrollarse.

1.3.4.3.- Diseño

En esta fase se transforman las necesidades del cliente en las páginas necesarias para entregar y procesar la información que residirá en el servidor. El desarrollador es responsable de definir la estructura física de navegación, así como el formato para cada una de las páginas a presentarse y el tipo de acceso

los elementos que cada una de esta contendrá, ya sea de lectura, escritura, búsqueda o ingreso y modificación de datos.

1.3.4.4.- Implementación

En esta fase el desarrollador es responsable de programar todas y cada una de las páginas que se encuentran enunciadas en el Plan de navegación y de producir los elementos multimedia necesarios o de escogerlos de librerías o del Internet, para luego enlazarlos.

1.3.4.4.1.- Programación

Implementar una página por cada una de las Páginas Web definidas en el Mapa de Navegación, utilizando el lenguaje HTML directamente o utilizando un generador de páginas Web tomando en cuenta los guiones definidos para cada tipo de página. Las únicas páginas que no se implementan en esta etapa son las páginas Web externas, que son páginas desarrolladas por otra persona o equipo y a la cual únicamente se realiza un enlace en la página que haga falta.

1.3.4.4.2.- Producción

Producir el video, las pistas de sonido, las imágenes estáticas, las animaciones, etc. Es decir los elementos de multimedia que no sean considerados como texto. Tomar los guiones o scripts de producción para reproducir un elemento por cada guión. Obtener los elementos adicionales necesarios de alguna librería de imágenes, videos, animaciones, por ejemplo se puede buscar en el Internet ciertas Sound Tracks que son gratis, o imágenes que se encuentran en librerías gráficas disponibles para desarrolladores de páginas Web.

1.3.4.5.- Pruebas

En esta fase el desarrollador es responsable de probar todas y cada una de las páginas Web, así como de probar los enlaces para asegurarse de que esté libre de errores y de que no existan enlaces sin páginas. Las pruebas sólo pueden ser realizadas durante y después de la fase de implementación. Una de las mayores dificultades que se encara en esta fase es que el desempeño depende del equipo en donde se ejecute el browser y de la velocidad de conexión a la red o a la Internet, en caso de ser una página pública.

Si no se puede controlar la plataforma del usuario, se debe probar por completo en distintas browsers, o al menos en los dos más conocidos como son Netscape e Internet Explorer, en vista que estos navegadores proveen de plug-ins o software necesario para visualizar los principales elementos de multimedia ajenos a los estándares aceptados por la mayoría de browsers, como son el sonido: *.wav, *.midi, o el video: *.avi. Si la página contiene formatos distintos de audio, video, imágenes y animación, será necesario instalar el plug-in correspondiente. Todas las pruebas deberán ser realizadas por el equipo de trabajo en conjunto.

1.3.4.6.- Mantenimiento

“La naturaleza omnipresente del cambio mediatiza todo el trabajo del software”⁷. El cambio es inevitable en la construcción de sistemas basados en computadora; por ello, se debe desarrollar mecanismos de evaluación, control e implementación de modificaciones. El desarrollador de páginas no debe observar la fase de mantenimiento como una fase de corrección de errores.

Durante esta fase se pretende definir un plan de mantenimiento de acuerdo a las necesidades del cliente, de observaciones hechas por los usuarios de la

⁷ ROGER PRESSMAN, Ingeniería de Software Un enfoque práctico, Quinta edición, McGraw-Hill 2002

página y de necesidades de perfeccionamiento de acuerdo al avance tecnológico del medio en el que será publicada.

1.3.5.- DISEÑO DE INTERFACES DE USUARIO

La interfaz gráfica del usuario fue diseñada para darle a la gente control directo sobre sus computadoras personales. Los usuarios se han acostumbrado a contar con un cierto nivel de sofisticación en todos los diseños, incluyendo el de páginas Web. El objetivo es satisfacer las necesidades de todos los usuarios potenciales, adaptando la tecnología Web a fin de cubrir sus expectativas, sin esperar nunca que el lector se conforme con una interfaz que pone obstáculos innecesarios en su camino.

Es aquí dónde resulta absolutamente crucial investigar las necesidades y datos demográficos del público objetivo. Diseñar para una persona desconocida, cuyas necesidades no se comprende, es imposible. Elabore varios guiones para diferentes tipos de usuarios que buscan información en el sitio Web.

El diseño de la página de inicio facilitará o complicará las cosas para un usuario experimentado que busca información específica. Un menú complejo con sólo texto intimidará a un usuario principiante.

1.3.5.1.- Elaborar un menú de navegación claro

En la situación actual de la tecnología Web la mayoría de los usuarios están acostumbrados a interactuar con el diseño navegando a través de los vínculos que encuentran en las páginas. El problema principal en la interfaz de sitios Web es la carencia de una indicación clara de dónde se encuentra el usuario dentro de la organización local de la información.

Iconos claros y consistentes, temas gráficos con identidad, y una pantalla que muestre una visión de conjunto o un resumen pueden darle al usuario la confianza de que podrá encontrar lo que busca sin perder tiempo. Los usuarios deben poder regresar a la página de inicio o a cualquier otro punto importante del menú de navegación con toda facilidad. Estos vínculos básicos, que deben estar presentes en todas las páginas del sitio Web, son a menudo presentados con botones gráficos y no sólo proveen los vínculos básicos de navegación y ayudan a crear la identidad gráfica del sitio Web sino que también le indican al usuario que aún está dentro del dominio del sitio Web.

La barra de botones es útil (muchas alternativas en un espacio pequeño), confiable (siempre está ahí, al principio de cada página), y provee una identidad gráfica consistente para cada página en el sitio Web.

1.3.5.2.- Acceso directo a los usuarios

El objetivo es proporcionar la información que el usuario desea en el menor número posible de pasos y en el tiempo más corto posible. Esto requiere que se diseñe una organización eficiente para presentar la información y reducir el número de pasos entre las diferentes páginas del menú. Estudios sobre la interfaz han demostrado que los usuarios prefieren menús que presenten un mínimo de cinco a siete vínculos, y que los usuarios prefieren unas pocas alternativas con mucha información en vez de muchas capas de menús simplificados.

El diseño de la interfaz de usuario es importante ya que es la interacción del usuario con la aplicación, la forma como navegara en el sitio Web que son partes fundamentales dentro del desarrollo de sistemas hipermedios.

Tal es la necesidad de utilizar un método de diseño hipermedia orientado a objetos, que este será detallado en el capítulo siguiente.

CAPÍTULO II

II.- METODOLOGÍA DE DISEÑO HIPERMEDIA ORIENTADO A OBJETOS (OOHDM)

2.1.- INTRODUCCIÓN

En este trabajo se presenta la metodología OOHDM (Object-Oriented Hypermedia Design Method) por Gustavo Rossi⁸ y de Daniel Schwabe⁹.

OOHDM propone 5 actividades durante la construcción de una aplicación hipermedia:

Levantamiento de requisitos, diseño conceptual, diseño navegacional, diseño de interfaz abstracta e implementación. Las primeras actividades son desarrolladas iterativamente, en cuanto a la actividad de implementación generalmente es desarrollada después de estas.

En este trabajo se presenta para cada actividad de la metodología OOHDM, los conceptos y la notación que se usa para su representación.

La actividad de levantamiento de requisitos define cuales son los usuarios de la aplicación y las tareas que podrán realizar.

⁸ ROSSI, Gustavo Héctor. Un Método Orientado a Objetos para el desarrollo de Aplicaciones Hipermedia. Río de Janeiro: DI / PUC-RIO (Tesis de Doctorado). 1996

⁹ <http://www.telemidia.puc-rio.br/oohdm/oohdm.html>

El diseño conceptual analiza el dominio de la aplicación que será desarrollada, el mismo que solamente da una parte de la aplicación en esta fase.

En esta fase son presentados todos los conceptos utilizados en el diseño conceptual y la notación para cada concepto. La notación usada en el diseño conceptual esta basada en la notación UML (Unified Modeling Language)¹⁰.

El diseño navegacional mapea un esquema conceptual para la definición de la aplicación hipermedia que será desarrollada, definiendo vistas de acuerdo a cada tipo de usuario. Puede ser dividida en dos sub actividades. Definición del esquema navegacional y definición del esquema de contexto navegacional. En esta fase son presentados todos los conceptos utilizados en el diseño navegacional. Debido a la falta en UML no se tratara el esquema de contexto navegacional con UML, solamente la notación del esquema navegacional esta basada en UML.

El diseño de interfaz abstracta define los objetos de la interfaz, sus propiedades y transformaciones. En esta fase son presentados todos los conceptos utilizados en el diseño de interfaz abstracta y la notación de cada concepto.

Por último, se presenta la actividad de implementación. La implementación transforma el resultado del proyecto navegacional y de la interfaz abstracta hacia un ambiente de implementación. En esta fase son presentados los conceptos relacionados con la implementación.

¹⁰ <http://www.rational.com/uml>

2.2.- ANÁLISIS

2.2.1.- LEVANTAMIENTO DE REQUISITOS

La actividad de levantamiento de requisitos define cuales son los usuarios de la aplicación que será desarrollada y las tareas que deberán ser respaldadas. Las tareas respaldadas por la aplicación son representadas a través de escenarios, casos de uso y diagramas de interacción de usuarios (UIDs).

Esta actividad presenta las siguientes fases: identificación de actores y tareas, especificación de escenarios, especificación de casos de uso, especificación de los diagramas de interacción de usuarios (UIDs) y la validación de los casos de usos y UIDs.

En la fase de identificación de actores y tareas, el analista interactúa con el dominio de la aplicación para identificar los actores y las tareas. Esta interacción es realizada a través del análisis de documentos disponibles y entrevistas con los usuarios. El principal objetivo es capturar las necesidades de los usuarios.

En la fase de especificación de escenarios, los escenarios describen las tareas que los usuarios desean realizar en el dominio en cuestión.

En la fase de especificación de casos de uso, el analista especifica los casos de uso a partir de los escenarios.

En la fase de especificación de diagramas de interacción de usuario (UIDS), los UIDs que representan los casos de uso son especificados. Un UID representa la interacción entre el usuario y la aplicación descrita textualmente en un caso de uso.

2.2.1.1.- Identificación de actores y tareas

Los actores son aquellos que alimentan de información a la aplicación. Un actor representa un papel específico de un usuario que utiliza un sistema¹¹. O sea, un usuario puede representar varios papeles, representando varios actores.

Es importante identificar las diferentes clases de actores porque generalmente ellos no participan en dos mismos escenarios. Así, son discutidos con cada usuario solamente los escenarios relativos a las clases de actores a los cuales el pertenece.

Las clases identificadas deben ser confirmadas con los usuarios en la etapa siguiente, o sea durante la especificación de escenarios.

Un ejemplo de actor, en un dominio de universidad, es un estudiante. Un estudiante es un usuario que busca información sobre un profesor que trabaja en una determinada área de investigación.

Las tareas que una aplicación debe soportar también son identificadas en esta fase. El término de tarea significa un objetivo que un usuario desea alcanzar usando la aplicación.

Para ejemplificar la identificación de tareas, a continuación son presentadas algunas tareas para un dominio de universidad, relativas al actor estudiante.

- Búsqueda de información de un profesor a partir de su nombre.
- Búsqueda de información de un profesor a partir del nombre del área de investigación.
- Búsqueda de proyectos de investigación conducidos por un laboratorio.

¹¹ Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G. Object-Oriented Software Engineering – A Use Case Driven Approach, Addison-Wesley, 1992.

2.2.1.2.- Especificación de escenarios

Un escenario es una descripción narrativa que explica detalladamente las tareas que un usuario desea realizar en el dominio en cuestión.

Antes de iniciar la descripción de escenarios, un usuario debe identificar los actores del dominio y a que clase de actores el pertenece. Si un usuario pertenece a mas de una clase, a medida que se van especificando los escenarios, se debe enunciar a que clases de actores en los escenarios el pertenece. En este caso el deberá tener mas de un identificador, uno para cada papel de actor que el representa.

La ~~Figura II.1~~~~Figura II.4~~ muestra la descripción de un escenario donde un usuario escoge un profesor a partir del nombre del área de investigación.

Con formato: Español (alfab. internacional)

Escenario E1: Escoger un profesor a partir de una área de investigación.
Contexto: Soy un alumno de penúltimo año de curso de Ciencias de Computación. Como en el próximo año tengo que elaborar el proyecto final, me gustaría escoger un profesor orientador para dar inicio al proyecto.
Objetivo: Escoger un profesor orientador a partir del nombre de una área de investigación.
Acciones: Ingreso el nombre del área de investigación (ex. Redes de Computadores) y el sistema me retorna una o más áreas que contengan el nombre ingresado. Entonces, seleccionó el área deseada y son presentados todos los profesores que trabajan en esa área. Si uno de los profesores presentados es un orientador, selecciono al profesor y el sistema me presenta toda la información sobre el profesor.

Figura II.1 Especificación de un escenario

2.2.1.3.- Especificación de casos de uso

Un caso de uso es una manera de usar el sistema¹². Los casos de uso tratan solamente de las interacciones del usuario con el sistema y de la información visible para el usuario, no abordan en el funcionamiento interno del sistema.

Un caso de uso no representa el mismo significado que un escenario, el representa un conjunto de potentes escenarios que tratan de una misma tarea¹³. O sea, un escenario representa una instancia de un caso de uso¹⁴. Por lo tanto para especificar los casos de uso es necesario primero agrupar todos los escenarios que describen las mismas tareas. Cada grupo de escenarios es mapeado para un caso de uso.

La [Figura II.2](#) muestra el caso de uso Seleccionar un profesor a partir del nombre de una área de investigación.

¹² Rumbaugh J., Jacobson, I. and Booch, G. The Unified Software Development Process. Addison-Wesley, 1999.

¹³ Rumbaugh, J. Getting Started: Using Use Cases to Capture Requirements. Journal Object-Oriented Programming (September 1994), 8-12.

¹⁴ Jacobson, I. The Use Case Construct in Object-Oriented Software Engineering. Scenario-Based Design: Envisioning Work and Technology in System Development. John Wiley & Sons, 309-336, 1995.

<p>Caso de Uso CU1: Seleccionar un profesor a partir del nombre de una área de investigación</p> <p>Escenarios: E1/...</p> <p>Descripción:</p> <ol style="list-style-type: none">1. El usuario entra con todo o parte del nombre del área de investigación.2. El sistema retorna una lista de las áreas de investigación que se acoplen a la entrada. Si existe solamente una área de investigación con ese nombre, todos los profesores que trabajan con ella son presentados directamente.3. El usuario selecciona el área de investigación.4. El sistema retorna una lista de profesores que trabajan con aquella área de investigación. Para cada profesor es presentado su nombre y su titulación.5. Si un usuario desea, el puede acceder a mas información del profesor(caso de uso Mostrar información del profesor)6. Si el usuario no esta interesado, el puede retornar al paso 5 para obtener información sobre otro profesor que trabaja con la misma área de investigación.
--

Figura II.2 Especificación de un caso de uso

2.2.1.4.- Especificación de UIDs

Un Diagrama de interacción de Usuario o UID (del término en inglés *User Interaction Diagram*) muestra la interacción entre el usuario y el sistema descrita en el texto del caso de uso. La interacción representada describe el intercambio de información entre el sistema y el usuario, sin entrar en detalles relacionados a la interfase con el usuario.

La especificación de un diagrama de interacción de usuario se elabora utilizando la siguiente notación:

2.2.1.4.1.- Ítem de Información

Un ítem de información representa una información única (simple) que aparece durante la interacción. Un ítem de información puede estar acompañado de su dominio, siendo, en este caso, seguido por dos puntos y el nombre del dominio.

Los dominios usados no especifican la interacción entre el usuario y el sistema a través de los UIDs son definidos por el propio analista. Es importante que el analista defina los dominios que puedan ser fácilmente comprendidos por el usuario. (Ej. Número, Texto, Sonido, Imagen, Video). Si un dominio enumerado por definición, los valores deben ser especificados entre llaves y separados por comas. Generalmente el nombre del ítem de información es escrito con letras minúsculas.

Si el dominio de un ítem de información no es especificado, es asumido el dominio como Texto. No así en el caso de un ítem de información con el dominio Sonido, Imagen o Video, el nombre el ítem de información puede ser suprimido y este es, entonces, representado solamente por el símbolo “.” seguido de su dominio.

```
<ítem de información> : <dominio>  
  
<ítem de información>  
    : Sonido  
    : Imagen  
    : Video
```

La [Figura II.3](#), el estado de interacción presenta dos ítems de información que representan, respectivamente, el nombre del alumno y el número de su matrícula.

Como el dominio del ítem de información nombre del alumno no es especificado, es asumido el dominio como Texto. El dominio del ítem de información matrícula es definido como Número.

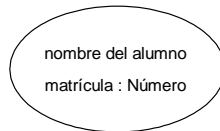


Figura II.3 Ejemplo de Ítem de Información

2.2.1.4.2.- Estructura

Una estructura representa una colección de información (ítems de información, estructuras, conjunto de ítems o conjuntos de estructuras) que están relacionados de alguna manera. Las colecciones de información de una estructura esta especificada entre paréntesis después del nombre de la estructura. El nombre de la estructura es obligatorio. Es permitido utilizar el mismo nombre para dos estructuras distintas, a fin de indicar visiones diferentes que el usuario tiene sobre la misma información. Generalmente el nombre de la estructura es escrito con la primera letra en mayúscula.

<Estructura> (<ítem de informacion₁>, <ítem de informacion₂>.. <ítem de informacion_n>)

<Estructura> ()

La [Figura II.4](#), los dos estados de interacción presentan estructuras. La primera interacción es presentada la estructura área de investigación, que presenta asociado a ella el ítem de información nombre. La segunda interacción se presenta la estructura Profesor. Esta estructura presenta varios ítems de información asociados a ella: nombre, títulos, fotos y e-mail.

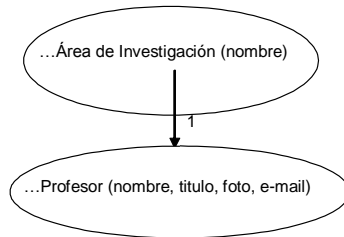


Figura II.4 Ejemplo de Estructura

2.2.1.4.3.- Conjunto

Un conjunto representa un conjunto de ítems de información o estructuras. La multiplicidad de un conjunto es representada por min..máx. en frente del ítem de información o estructura. La multiplicidad por default es 1..N y es representada solamente por paréntesis (...).

```

...<ítem de información>
... <Estructura> (<ítem de informacion1>, <ítem de inf2> .. <ítem de infn>)
... <Estructura> ( )
1..5 <ítem de inf>
0..5 <ítem de inf>
  
```

La [Figura II.4](#) presenta 2 conjuntos: conjunto de Áreas de investigación y conjunto de Profesores. Los dos conjuntos de estructuras presentan cardinalidad mínima de 1 y máxima de N.

2.2.1.4.4.- Información Opcional

La información opcional representa un ítem de información, estructura o texto opcional. La información opcional son representados por el símbolo "?".

Entretanto, estos también pueden ser representados por un conjunto con multiplicidad de 0..1. En el caso de un ingreso del usuario, una información opcional incluso puede ser representada por un rectángulo con líneas punteadas.

<ítem de información>?
 ...<ítem de información>?
 <Estructura> (<ítem de inf₁>, <ítem de inf₂> .. <ítem de inf_n>)?
 “ texto “?”
 0..1 <ítem de inf>
 <ítem de inf>

La [Figura II.5](#) presenta un ítem de información opcional foto. Este ítem de información también podría ser representado como un conjunto con cardinalidad 0..1.

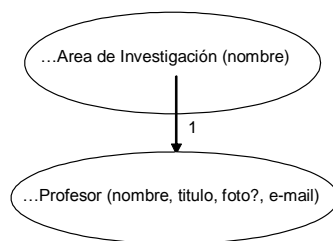
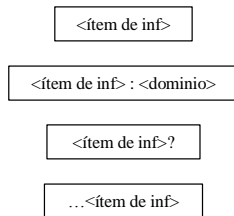


Figura II.5 Ejemplo de Información Opcional

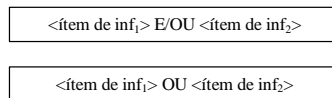
2.2.1.4.5.- Entrada de Usuario

Una entrada de usuario representa un ítem de información o estructura suministrador por el usuario. Toda información ingresada por el usuario es siempre colocada dentro de un rectángulo.



Si la entrada de dos ítems de información son dependientes entre si y por los menos uno de ellos necesita ser suministrado por el usuario, es usado el símbolo “E/OU” entre los dos ítems de información.

Si la entrada de dos ítems de información son dependientes entre si y solamente uno de ellos puede ser suministrado por el usuario, es usado el símbolo “OU” entre los dos ítems de información.



La [Figura II.6](#) presenta 2 ítems de información suministrados por el usuario: área de investigación y título. El ítem de información de área de investigación es obligatorio, en cuanto al ítem de título es opcional por eso esta acompañado del símbolo “?”. Como visto anteriormente, el ítem de título, puede ser opcional, también podría ser representado por un rectángulo con líneas punteadas.

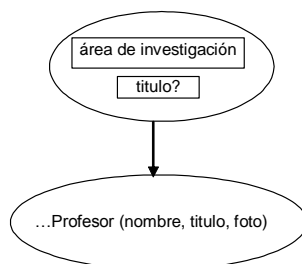


Figura II.6 Ejemplo de Entrada de Usuario

2.2.1.4.6.- Entrada de Usuario Enumerada

Una entrada de usuario enumerada representa una entrada de usuario que debe ser seleccionada a partir de opciones suministradas por el sistema. Las opciones presentadas al usuario aparecen entre corchetes separadas por comas.

Si más de una opción puede ser seleccionada, entonces la cantidad es indicada en frente del nombre del ítem de información. Cuando la interacción va seguida para una interfaz con el usuario, esas opciones serán previamente presentadas por el sistema.

<ítem de inf> [<opción₁>, <opción₂> .. <opción_N>]

<min>..<max> <ítem de info> [<opción₁>, <opción₂> .. <opción_N>]

El estado de interacción presentado en la [Figura II.7](#) presenta una entrada de usuario enumerada, donde el ítem de información suministrado por el usuario es la matrícula, que puede ser una nueva matrícula o una renovación.

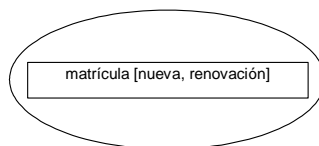


Figura II.7 Ejemplo de Entrada de Usuario Enumerada

2.2.1.4.7.- Salida de Sistema

Una salida de sistema representa un ítem de información o estructura retornado por el sistema. Toda información retornada por el sistema es colocado directamente en el estado de interacción, o donde, todo lo que aparece fuera de los rectángulos es información retornada por el sistema.

<ítem de inf> : <dominio>
 <ítem de inf>
 <ítem de inf >?
 ...<ítem de inf>
 <Estructura> (<ítem de inf>)

La [Figura II.8](#) ~~Figura II.8~~ presenta dos estados de interacción que contienen salidas del sistema. En el primer estado, el sistema retorna un conjunto de estructura área de investigación. El segundo, el sistema retorna un conjunto de estructura Profesor, para cada profesor del conjunto, es presentado con su información.

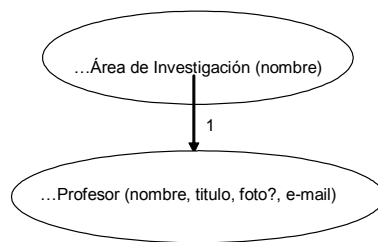


Figura II.8 Ejemplo de Salida del Sistema

2.2.1.4.8.- Texto

Representa un texto pre-definido de carácter explicativo presentado por el sistema durante la interacción.

“ <texto> “

La [Figura II.9](#) ~~Figura II.9~~, el estado de interacción presenta un texto orientado al usuario sobre el tipo de información que debe ser suministrada al sistema (“*Ingrese el titulo del profesor*”).



Figura II.9 Ejemplo de Texto

2.2.1.4.9.- Estado de interacción

Representa un estado de interacción entre el usuario y el sistema. La información suministrada por el usuario y las retornadas por el sistema son usualmente colocadas dentro del elipse. Un estado de interacción nunca puede ser vacío.

Es considerado que a partir de cualquier estado de interacción de un UID el usuario tiene la posibilidad de abortar la interacción con el sistema. Esta información no es representada gráficamente para no ensuciar el diagrama. Así, el analista precisa dejar explícito que el usuario no pueda abortar la interacción con el sistema en un determinado estado de interacción, entonces el tendrá que incluir esa información como una nota textual anexada al UID.

La [Figura II.10](#) ~~Figura II.10~~ presenta un estado de interacción entre el usuario y el sistema representado por una elipse. Ese estado de interacción significa que en un determinado momento de interacción, el sistema estará mostrando un conjunto de profesores y su información para el usuario.

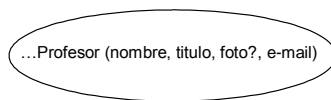


Figura II.10 Ejemplo de Estado da interacción

2.2.1.4.10.- Estado Inicial de interacción

Representa el estado inicial de interacción entre el usuario y el sistema. Esta representado por una pequeña transición sin origen.

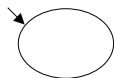


Figura II.11 Estado Inicial de interacción

La [Figura II.12](#) ~~Figura II.12~~ presenta dos estados de interacción. El estado de interacción que presenta una transición sin origen representa el estado inicial de interacción. Eso significa que la interacción entre el usuario y el sistema comienza a partir de ese estado, o sea, comienza con el usuario suministrando el título del profesor.

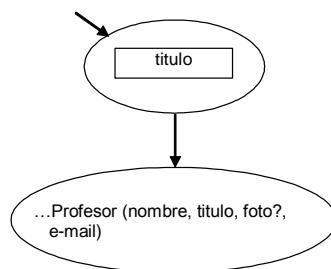


Figura II.12 Ejemplo de Estado Inicial da interacción

2.2.1.4.11.- Estados Alternativos de interacción

Esta representación es usada cuando existen dos o más salidas alternativas a partir de un estado de interacción. El estado de interacción, que se tornara punto de interacción, depende de la información suministrada por el usuario o la opción seleccionada por él.

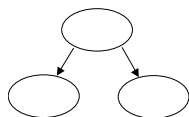


Figura II.13 Estados Alternativos de interacción

La [Figura II.14](#) ~~Figura II.14~~ presenta dos estados alternativos a partir del estado inicial de interacción. El estado que se tornara punto de interacción será determinado de acuerdo con la opción seleccionada por el usuario. Si a partir del estado inicial de interacción el usuario selecciona la opción Proyectos de investigación, entonces el estado que presenta un conjunto de Proyectos de investigación se tornara punto de interacción. Si a partir del estado inicial de interacción el usuario selecciona la opción Profesores, entonces el estado que presenta un conjunto de profesores se tornara punto de interacción.

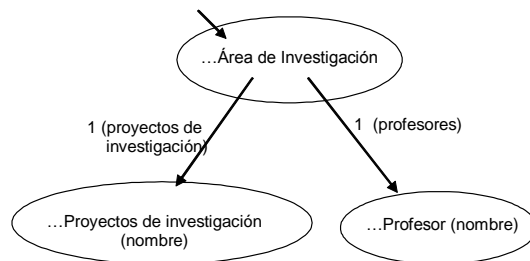


Figura II.14 Ejemplo de Estados Alternativos de interacción

2.2.1.4.12.- Sub-Estados de un Estado de interacción

Esta representación es usada cuando las partes de un estado de interacción son excluyentes. Las partes excluyentes son colocadas en sub-estados de estado de interacción. Cuando un estado de interacción presenta sub-estados, el usuario tiene que optar por el sub-estado que él vaya a seguir durante la interacción con el sistema.

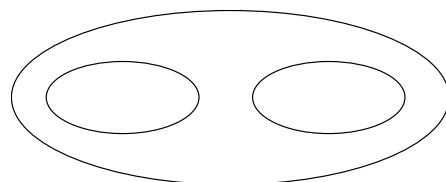


Figura II.15 Ejemplo de Sub-Estados de un Estado de interacción

La [Figura II.16](#) ~~Figura II.16~~ presenta un estado de interacción que contiene dos sub-estados, el usuario debe seguir la interacción presentada en uno de los dos estados. Si el usuario opta en seguir la interacción del sub-estado que presenta el texto "Alumno Nuevo", entonces él debe seleccionar la opción hacer matricula. Si él opta en seguir la interacción del sub-estado que presenta el texto "Alumno Antigo", entonces el debe suministrar su nombre y matricula y seleccionar la opción login.

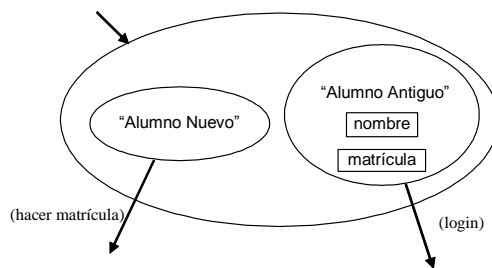


Figura II.16 Ejemplo de Sub-Estados de un Estado de interacción

2.2.1.4.13.- Llamada de Otro UID

Representa que el punto de interacción es trasferido para otro UID. Cuando el punto es transferido para otro UID, el estado inicial de otro UID se vuelve el punto de interacción. Después de ejecutar el otro UID, si no existe una transición de retorno para un nuevo estado de interacción, el punto de interacción retorna al estado de donde se hizo la llamada al UID.

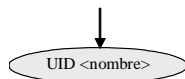


Figura II.17 Llamada de Otro UID

Si el estado de interacción que presenta la llamada del UID no posee ninguna transición de entrada, entonces ese UID puede ser llamado a partir de cualquier

estado de interacción del UID. Esa es la forma de representar los *landmarks* de navegación.

La [Figura II.18](#) ~~Figura II.48~~ presenta la llamada del UID Mostrar información de un Profesor, a partir del segundo estado de interacción, donde el sistema presenta un conjunto de profesores, el usuario puede seleccionar un profesor en la opción mostrar información del profesor. Esta opción es responsable en transferir el punto de interacción para la interacción inicial del UID que es llamado. Al final la interacción del UID, el punto de interacción retorna para el estado de interacción a partir de donde el UID fue llamado. Para que el retorno sea hecho al estado de interacción específico, es necesario incluir una transición de salida de UID llamado para ese estado de interacción. La llamada de otro UID a partir de cualquier estado de interacción del UID en transcurso también puede ser representado a través del relacionamiento que existe en el diagrama de relacionamiento entre UIDs, como será visto en la sección 2.2.1.4.20.-.

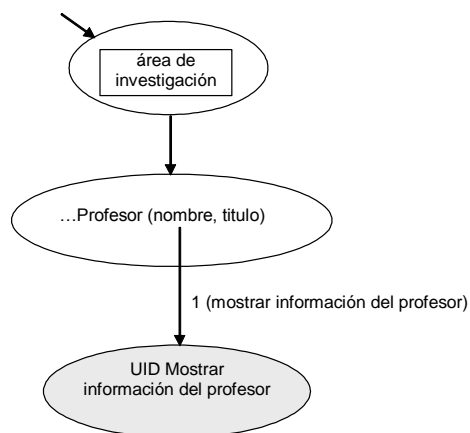


Figura II.18 Ejemplo de llamada de Otro UID

2.2.1.4.14.- Llamada a partir de Otro UID

Representa que el punto de interacción fue transferido a partir de otro UID, después de ejecutar el UID en curso, el punto de interacción retorna al estado de interacción del otro UID, a partir de donde el UID en curso fue llamado.

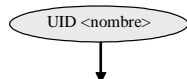


Figura II.19 Llamada a partir de Otro UID

La [Figura II.20](#) ~~Figura II.20~~ presenta parte del UID Mostrar información del profesor, que es llamado a partir del UID Selección de un profesor a partir del área de investigación.

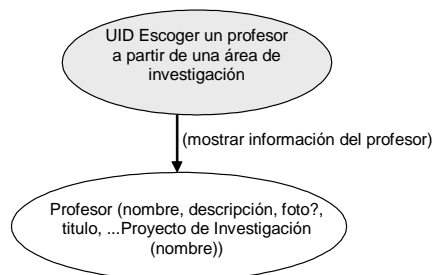


Figura II.20 Ejemplo de Llamada a partir de Otro UID

2.2.1.4.15.- Transición del Estado de interacción

Una transición representa que el estado destino puede volverse el nuevo punto de interacción después el sistema retorna la información necesaria y el usuario suministra la información requerida en el estado origen.



Para que una transición ocurra, siempre es necesario que, en lo mínimo, un elemento sea seleccionado, una opción sea seleccionada o la información sea proporcionada. El origen de una transición puede ser todo un estado, un sub-estado o, cuando uno o más elementos necesitan ser seleccionados por el

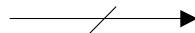
usuario, un ítem de información o estructura retornados por el sistema. También puede ocurrir de una transición tiene mas de un origen. El destino de una transición es siempre todo un estado.

Cuando el usuario selecciona alguna opción que no emite un punto de interacción, el origen y el destino de una transición pueden ser el mismo estado de interacción.

Es necesario representar explícitamente que el usuario puede retornar al estado de interacción origen de transición, entonces esta transición debe ser representada con doble flecha.



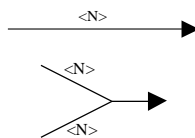
Es necesario representar explícitamente que el usuario no puede retornar al estado de interacción origen de transición, entonces la transición debe estar cortada.



Es posible asociar rótulos de las transiciones del estado de interacción. Esos rótulos pueden ser combinados y asociados a las transiciones. A continuación son presentados los posibles rótulos.

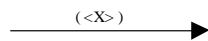
1) Transición con Selección de N Elementos

Representa que N elementos deben ser seleccionados por el usuario antes que el estado de interacción destino tome el punto de interacción. Los elementos seleccionados también pueden pertenecer a conjuntos diferentes.

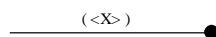


2) Transición con Selección de opción X

Representa que la opción X es seleccionada para que el estado de interacción destino se vuelva el punto de interacción. El nombre de opción es colocado entre paréntesis.

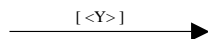


Cuando la selección de una opción no cambia el punto de interacción, o sea, el origen y el destino de transición son el mismo estado de interacción, la transición puede ser representada por una línea con un círculo relleno en la punta y con la opción asociada en esta.



3) Transición con Condición Y

Representa que el estado de interacción destino de la transición toma el punto de interacción si la condición Y es satisfecha. Una condición no puede estar asociada a la transición de entrada del estado inicial de interacción. Si más de una transición de salida de un estado de interacción presenta condiciones, estas condiciones deben ser excluyentes. La condición se expresa usando lenguaje natural y colocada entre corchetes.



2.2.1.4.16.- Pre-Condiciones y Pos-Condiciones

Es posible asociar pre-condiciones y pos-condiciones a los UIDs. Las pre-condiciones de los UIDs son similares a las pre-condiciones y pos-condiciones de los casos de uso.

En los casos de uso, una pre-condición representa lo que el sistema debe asegurar que deja satisfecho antes de ejecutar el caso de uso al iniciar. La asociación de pre-condiciones de un UID también significa que el solo puede ser ejecutado si esas condiciones fueran satisfechas.

Es importante resaltar que la pre-condición de un UID es diferente a la condición de una transición. La primera esta asociada al UID como un todo, en cuanto a la segunda esta asociada a un único estado de interacción. Las pre-condiciones de un UID son representadas dentro de un rectángulo y expresadas en lenguaje natural.

Pre-condiciones: <Y>

Las pos-condiciones de los casos de uso describen los posibles estados finales de los casos de uso. En los UIDs, la definición de las pos-condiciones describen las condiciones que podrían ser satisfechas después de ejecutar la interacción descrita en los UIDs. Las pos-condiciones de un UID también son representadas dentro de un rectángulo y expresadas en lenguaje natural.

Pos-condiciones: <Y>

2.2.1.4.17.- Notas Textuales Anexadas a los UIDs

Es posible anexar una nota textual en un UID. La nota textual sirve para especificar alguna información importante que no puede ser representada gráficamente en el UID.

2.2.1.4.18.- UIDs Parametrizados

Los UIDs que presentan la misma secuencia de interacción, pero con información distinta, pueden ser representados como un único UID parametrizado.

El UID parametrizado presenta la misma secuencia de interacción de los UIDs del cual el esta basado y la misma información que tienen en común. La información que diferencia con los UIDs originales son presentados como parámetros del UID parametrizado.

La parametrización de los UIDs facilita la validación, porque durante las entrevistas con los usuarios, el analista puede utilizar solamente un UID parametrizado en ves de utilizar varios UIDs similares. Cuando es necesario, el analista debe referir las particulares de cada UID individual.

La [Figura II.21](#) ~~Figura II.21~~ y [Figura II.22](#) ~~Figura II.22~~ presentan, respectivamente, los UIDs Escoger un profesor a partir de una área de investigación y Escoger un profesor a partir de un proyecto de investigación. Esos UIDs presentan una misma secuencia de interacción y la mayoría de información en común. Entretanto, el UID escoger un profesor a partir de una área de investigación presenta una entrada de una área de investigación echa por el usuario y un conjunto de áreas de investigación que retorna el sistema, en cuanto al UID Escoger un profesor a partir de un proyecto de investigación presenta la entrada de un proyecto de investigación y un conjunto de proyectos de investigación.



Figura II.21 Escoger un profesor a partir de una área de investigación: UID

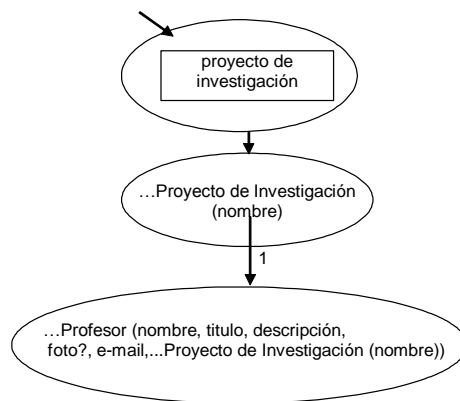


Figura II.22 Escoger un profesor a partir de un proyecto de investigación: UID

Como estos UIDs presentan una misma secuencia de información y la mayoría de la información en común, es posible definir un UID parametrizado a partir de ellos. El UID parametrizado definido es presentado en la [Figura II.23](#). Este presenta la misma secuencia y la misma información en común. Las diferentes informaciones, área de investigación y proyecto de investigación, son presentadas como parámetro.

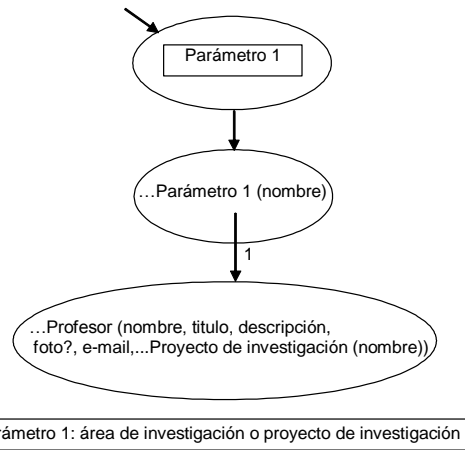


Figura II.23 Escoger un profesor a partir de un parámetro: UID Parametrizado

2.2.1.4.19.- Ejemplos de UIDs

1) Selección de un CD de música

La [Figura II.24](#) presenta el UID definido para presentar la interacción entre el usuario y el sistema durante la tarea Selección de un CD de música. En esta tarea, el usuario debe seleccionar un CD a partir de un conjunto de CDs mostrados por el sistema. Para el CD seleccionado, el sistema debe mostrar toda la información relacionada con la música, inclusive parte de la música y video.

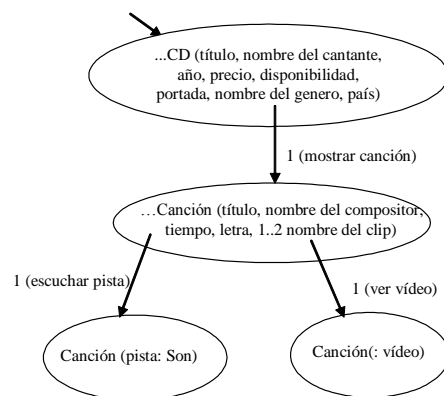


Figura II.24 Selección de un CD de música: UID

En el primer estado de interacción el UID, el sistema presenta un conjunto de CDs. Para cada CD, el sistema presenta los siguientes ítems de información: título, nombre, cantante, año, precio, disponibilidad, portada, nombre del género, país. Durante este estado de interacción, el usuario debe seleccionar un CD en la opción mostrar canción.

Para el CD seleccionado, el sistema muestra el conjunto de canciones. Para cada canción son mostrados los siguientes ítems de información: título, nombre del compositor, tiempo, letra y un conjunto de dos nombres de clip. La pista de la canción y los videos no son mostrados junto con esos ítems de información. Para el usuario escuchar una pista del cd, es necesario que el seleccione una canción y la opción escuchar pista. Para el usuario ver un video de dos videos de música, es necesario que el seleccione un nombre del clip en la opción ver video.

Si el usuario escoge la opción escuchar pista, entonces el sistema toca la pista de la canción y después termina la pista, retorna al punto de interacción para el segundo estado. Si el usuario escoge la opción ver video, entonces el sistema presenta el respectivo video y, entonces, retorna al punto de interacción para un segundo estado.

2) Comprar los CDs seleccionados

La ~~Figura II.25~~ Figura II.25 presenta el UID definido para representar la interacción entre el usuario y el sistema durante la tarea Comprar los CDs seleccionados. En esta tarea, el usuario proporciona todos los datos necesarios para realizar la compra.

Si es por un cliente antiguo, no es necesario suministrar su nombre, dirección de entrega, dirección de cobro, solo el suministra su e-mail y su clave.

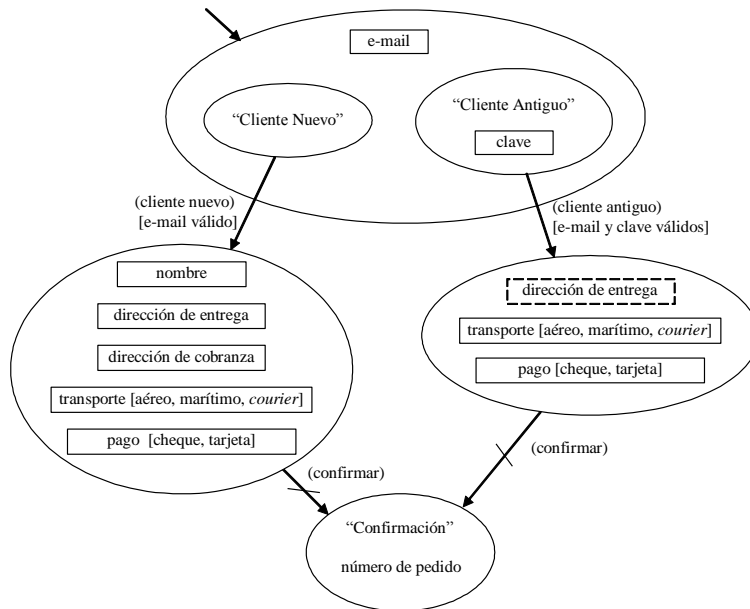


Figura II.25 Comprar los CDs seleccionados: UID

En el primer estado de interacción de este UID, el usuario debe ingresar con su e-mail y debe seguir uno de los dos sub-estados de este estado de interacción. Si el usuario sigue por el sub-estado conteniendo el texto “Cliente Nuevo”, el deberá seleccionar la opción cliente nuevo. Con la selección de esta opción, el punto de interacción pasa para el estado donde el usuario proporciona la siguiente información: nombre, dirección de entrega, dirección de cobranza, tipo de transporte (aéreo, marítimo, o courier), el tipo de pago (cheque o tarjeta). A partir de este estado, el usuario puede confirmar la compra.

Si a partir del estado inicial de interacción, el usuario sigue por el sub-estado conteniendo el texto “Cliente Antiguo”, el deberá proporcionar la clave y seleccionar la opción cliente antiguo. Con la selección de esta opción, el punto de interacción pasa para el estado donde el usuario podrá proporcionar una nueva dirección de entrega (opcional) y deberá suministrar el tipo de transporte y el tipo de pago. A partir del estado de interacción, el usuario podrá confirmar la compra.

Cuando el usuario confirma la compra, a través de la opción confirmar, el sistema retorna el numero de pedido. Como las transiciones de llegada están cortadas, no es posible retornar al estado de interacción anterior a partir de este estado de interacción.

3) Confirmación del registro de un Miembro del Comité con tratamiento de errores

La ~~Figura II.26~~ **Figura II.26** presenta el UID que representa la interacción entre el usuario y el sistema durante la tarea Confirmación del registro de un Miembro del Comité de Programas. Si el miembro del Comité del programa proporciona un login y la clave incorrectos, el deberá proporcionar estos datos nuevamente.

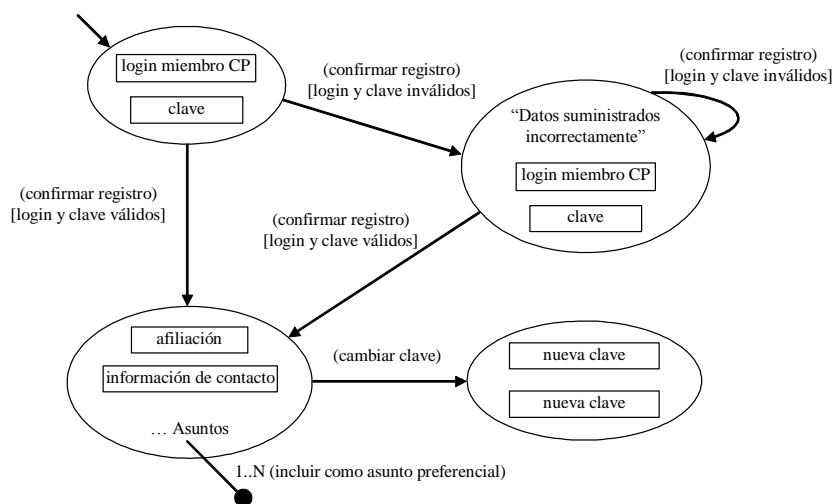


Figura II.26 Confirmación del registro de un Miembro del Comité con Tratamiento de Errores: UID

En el estado inicial de la interacción, si el usuario proporciona el login y la clave correctos y selecciona la opción confirmar registro, el punto de interacción pasa a ser el estado de interacción donde el usuario debe proporcionar su afiliación y la información de contacto.

Si el usuario proporciona el login y la clave incorrectos y selecciona la opción confirmar registro, el punto de interacción pasa a ser el estado de interacción para el tratamiento de ese error. En este estado de interacción, el sistema presenta el texto “Datos suministrados incorrectamente” y el usuario proporciona nuevamente el login y la clave. En cuanto el proporcione el login y clave inválidos, el punto de interacción continua en el mismo estado de interacción. Si, a partir de este estado de interacción para el tratamiento de ese error, el usuario suministra el login y la clave correctamente y selecciona la opción confirmar registro, el punto de interacción pasa para el estado de interacción donde el usuario debe proporcionar su afiliación y la información de contacto.

2.2.1.4.20.- Relacionamientos entre UIDs

El diagrama de relacionamiento entre UIDs presenta los relacionamientos más significativos entre los UIDs. Este diagrama sirve para solventar algunas relacionamientos que son descritos en los casos de uso, que todavía no son representados en los UIDs, por otro lado el facilitar la visualización de como los UIDs se integran para dar soporte a las tareas mayores. Los actores que están asociados a los UIDs también pueden ser especificados en diagramas de relacionamiento entre los UIDs.

En el diagrama de relacionamientos entre UIDs, identificaremos los siguientes tipos de relacionamientos:

1) Relacionamiento de inclusión (incluir)

El relacionamiento **incluir** entre dos UIDs significa que un UID se hace parte de otro UID y que la secuencia de interacción es incluida en otro UID. Este relacionamiento es usado cuando una parte común de un UID es de varios UIDs es especificado en otro UID. La interacción definida en el UID incluido no tiene sentido ser única. Durante la especificación de un UID que incluye otro UID, el relacionamiento de inclusión es representado por la llamada de otro UID.

2) Relacionamiento de extensión (extender)

El relacionamiento **extender** entre dos UIDs significa que un UID puede ser extendido por otro UID. Este relacionamiento es usado cuando un UID presenta un comportamiento alternativo u opcional. Si, durante la interacción del UID que esta siendo extendido, hay una llamada a otro UID, entonces la secuencia de interacción definida al otro UID será incluida en el. La interacción descrita en el UID que extiende al otro puede ocurrir independientemente del UID que esta siendo extendido. Durante la especificación de un UID, el relacionamiento de extensión también puede ser representado a través de una llamada al otro UID.

3) Relacionamiento de precedencia (precede)

El relacionamiento **precede** entre dos UIDs significa que un UID solo puede ser ejecutado si otro ha sido ejecutado con éxito previamente. La ejecución de un UID depende de la ejecución de otro UID. Durante la especificación de un UID, el relacionamiento de precedencia puede ser representado a través de una precondición del UID que es precedido por el otro.

La representación de estos relacionamientos es semejante a la representación usada en UML. Ellos son representados gráficamente por una dependencia estereotipada con el respectivo rótulo («incluir», «extender» y «precede»). De ser necesario, también es posible crear nuevos tipos de relacionamientos usando los estereotipos de UML.

La ~~Figura II.27~~ ~~Figura II.27~~ presenta un diagrama con el relacionamiento de inclusión entre algunos UIDs. Este diagrama representa que los UIDs Escoger un profesor a partir de una área de investigación y Escoger un profesor a partir de su nombre incluyen las mismas interacciones la interacción definida en el UID Mostrar información del profesor.

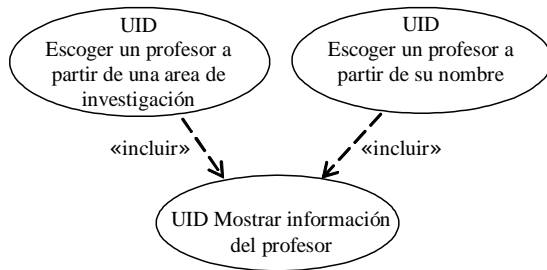


Figura II.27 Relacionamiento de inclusión entre UIDs

La [Figura II.28](#) ~~Figura II.28~~ presenta un diagrama con el relacionamiento de extensión entre dos UIDs. Este diagrama representa que el UID Comprar ítems seleccionados puede ser extendido por el UID Cambiar Clave. El UID Cambiar clave también puede ser ejecutado solo.

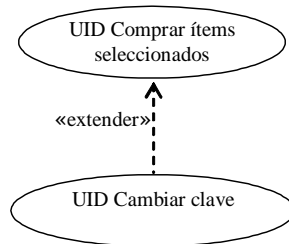


Figura II.28 Relacionamiento de extensión entre UIDs

La [Figura II.29](#) ~~Figura II.29~~ presenta un diagrama con el relacionamiento de precedencia entre dos UIDs. Este diagrama representa que la ejecución del UID Enviar un artículo debe preceder a la ejecución del UID Modificar información de un artículo.

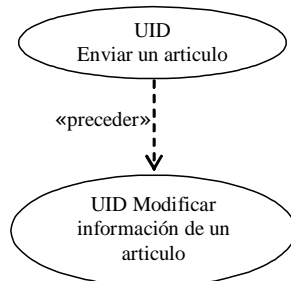


Figura II.29 Relacionamiento de Precedencia entre UIDs

Para simplificar el diagrama de relacionamiento entre UIDs, es posible agrupar los UIDs que presentan el mismo relacionamiento con otro UID en un mismo rectángulo. La [Figura II.30](#) ~~Figura II.30~~ presenta el mismo diagrama de relacionamiento entre UIDs presentado en la [Figura II.27](#) ~~Figura II.27~~. En la versión simplificada del diagrama apenas un relacionamiento es representado entre todo el grupo de UIDs y el UID relacionado al grupo. Entre tanto, este único relacionamiento representa que existe un relacionamiento entre cada UID perteneciente al grupo del UID relacionado.

Por ejemplo, el relacionamiento incluir, mostrado de ejemplo, representa que existe un relacionamiento incluir entre el UID Escoger un profesor a partir de una área de investigación y el UID Mostrar información del profesor y otro relacionamiento incluir entre el UID Escoger un profesor a partir de su nombre y el UID Mostrar información del profesor.

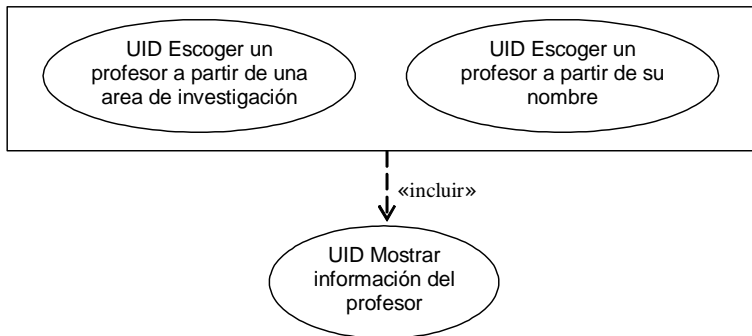


Figura II.30 Diagrama de Relacionamientos entre UIDs Simplificado

2.2.1.4.21.- Representación del UID Inicial

Algunas veces es necesario representar el UID inicial de la aplicación. Este UID representa las tareas que pueden ser llamadas directamente cuando la aplicación es iniciada. Esas tareas son representadas a través de UIDs y la llamada de las mismas es representada a través de relacionamientos incluir.

La [Figura II.31](#) ~~Figura II.34~~ presenta un ejemplo de UID inicial, donde las tareas Escoger un profesor a partir de una área de investigación y Escoger un profesor a partir del nombre pueden ser llamadas directamente.

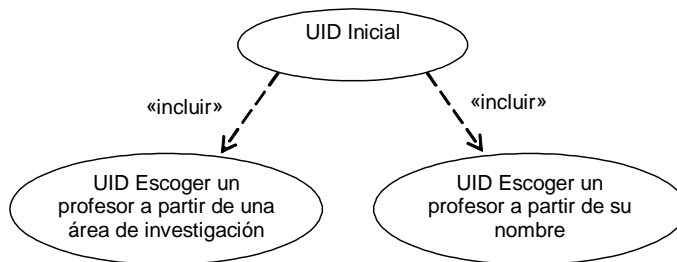


Figura II.31 Diagrama de Relacionamientos entre el UID Inicial

2.2.1.5.- Validación de los casos de uso y los UID's

La validación se la realiza con el usuario, y se realiza una tabla especificando el numero de caso de uso CU # y el numero de diagrama de interacción de usuario UID #, además se ha visto la necesidad de confrontar también con la Especificación de requisitos de software ERS REQ #, indicando si esta en estado correcto (OK) para pasar a la siguiente fase de OODHM.

Tabla II.1 Control de Validación de los UID's

Usuario	REQ #	CU #	UID #	ESTADO

2.3.- DISEÑO

2.3.1.- DISEÑO CONCEPTUAL

El diseño conceptual es una actividad responsable para el análisis del dominio de la aplicación, o sea engloba todo el universo de información relevante para la aplicación en cuestión, el mismo que apenas un subconjunto de la información venga a ser considerado posteriormente en su implementación.

El resultado de diseño conceptual de una aplicación consiste en un esquema conceptual conteniendo los objetos del dominio de la aplicación (clases, relacionamientos y subsistemas).

El esquema conceptual es un diagrama, conteniendo las clases representativas de la aplicación, los relacionamientos existentes entre estas clases y los subsistemas que agrupan las clases.

2.3.1.1.- Clases

Las clases en OOHDM poseen el mismo significado de las clases en modelo orientado a objetos. Ellas representan un conjunto de entidades que presentan las mismas características (atributos, relacionamientos y métodos).

Una clase es representada por un rectángulo dividido en 3 partes ([Figura II.32](#)). La primera parte contiene el nombre de la clase y otras propiedades generales de clase. La segunda parte contiene la lista de los atributos definidos para las instancias de clase. La tercera parte contiene la lista de las operaciones definidas para las instancias de clase.

Nombre-Clase
lista-atributos
lista-operaciones

Figura II.32 Representación de una clase

El nombre de una clase debe ser escrito en negrita, centrado y con la primera letra en mayúscula. Las propiedades de clase deben ser colocadas entre “{“ y “}“ (llaves). Las propiedades son informaciones adicionales que no son representadas explícitamente en el modelo, por ejemplo, el autor de clase, si la clase aun fue testeada, la versión de clase, etc.

Los atributos y operaciones son escritos con letra normal y justificada a la izquierda. Los atributos de la clase y las operaciones de la clase deben ser subrayados. Los nombres de las clases abstractas deben ser escritos en itálico.

En OOHDM no existen operaciones abstractas.

La [Figura II.33](#) muestra la representación de la clase Profesor, que presenta la propiedad de autor=Sandro, y la [Figura II.34](#) muestra la representación de clase abstracta Persona.

Profesor { autor=Sandro }
clase: String
altera-gratificação (): Real

Figura II.33 Representación de clase Profesor

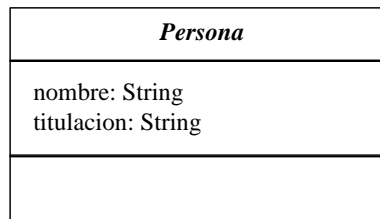


Figura II.34 Representación de clase abstracta Persona

Una clase también puede ser representada de forma simplificada, sin presentar la segunda parte (atributos) y/o la tercera parte (operaciones).

La [Figura II.35](#) muestra 3 ejemplos de representaciones simplificadas de clase Proyecto de Investigación. El primer ejemplo presenta solamente el identificador de clase. El segundo ejemplo presenta el identificador de clase y sus atributos. El tercer ejemplo presenta el identificador de clase y sus operaciones.

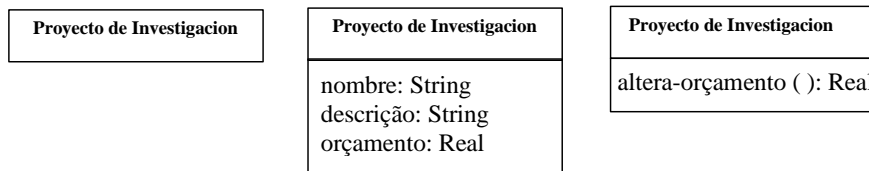


Figura II.35 Representaciones simplificadas de una clase

2.3.1.1.1.- Atributos

Los atributos de las clases representan las propiedades de los objetos. Por ejemplo, todo proyecto de investigación presenta un nombre, una descripción y un estimamiento.

Cada atributo debe ser de un tipo predefinido simple (ej., String), complejo (ej., Array) o un tipo definido por la aplicación.

En OOHD, los atributos pueden tener múltiples perspectivas, siendo que cada perspectiva representa una presentación diferente para el contenido del atributo.

Cuando un atributo posee múltiples perspectivas, ellas deben ir enumeradas entre “[” y “]” (corchetes), y cada una es representada por un identificador, seguido del carácter “:” y de un tipo.

Existen dos situaciones las cuales son usadas múltiples perspectivas. La primera situación es cuando el atributo con múltiples perspectivas es presentado en el mínimo en una misma perspectiva siempre que el objeto es mostrado. En esta situación, es necesario que una de las perspectivas sea *default*, siendo asignada con el carácter “+”.

La perspectiva *default* representa la perspectiva en la cual el atributo será un valor en todas las instancias. La perspectiva default es la única perspectiva que no precisa presentar un identificador, en este caso, recibir el identificador del atributo. Las otras perspectivas siempre precisan presentar un identificador y pueden o no aparecer las instancias.

La [Figura II.36](#) muestra dos maneras de representar el atributo descripción de clase abstracta Persona. Este atributo presenta múltiples perspectivas (Text e Image), y podrá también poseer una foto (perspectiva Image).

La [Figura II.36](#) (a) la perspectiva default del atributo descripción no presenta un identificador, recibiendo, en este caso, el identificador del propio atributo, el identificador descripción.

La [Figura II.36](#) (b) la perspectiva default del atributo descripción presenta un identificador propio, el identificador bio.

<i>Persona</i>
nombre: String titulacion: String descripcion: [Text+, foto: Image]

(a)

<i>Persona</i>
nombre: String titulacion: String descripcion: [bio: Text+, foto: Image]

(b)

Figura II.36 Atributo con múltiples perspectivas con perspectiva *default*

La segunda situación en que un atributo con múltiples perspectivas es usado es cuando este es presentado en diferentes perspectivas de acuerdo con el contexto en el cual el objeto es mostrado. En esta situación, no existe perspectiva default.

La [Figura II.37](#) muestra la representación del atributo descripción de clase Proyecto de Investigación. Este atributo posee múltiples perspectivas (TextoMarketing y TextoPesquisa), mas no posee ninguna perspectiva default.

En este ejemplo, un proyecto accedido en el contexto Proyecto Alfabético podría representar el atributo TextoMarketing, o sea, un texto con énfasis en el marketing del proyecto, en cuanto un proyecto accedido en el contexto Proyecto por Área podría presentar el atributo TextoPesquisa, o sea, un texto con énfasis en investigación del proyecto.

Proyecto de Investigacion
nombre: String descripcion: [TextoMarketing: Text, TextoPesquisa: Text] orçamento: Real

Figura II.37 Atributo con múltiples perspectivas sin perspectiva *default*

Los nombres de los atributos inician con una letra minúscula en fuente normal.

Los atributos presentan la siguiente definición:

visibilidad nombre : tipo = valor-default { propiedad }

En OOADM, la visibilidad de un atributo tiene una semántica diferente de UML, significa que un atributo puede ser visible o invisible. Así, la visibilidad puede ser privada o pública, no existiendo la visibilidad protegida, como acontece en UML.

Si el atributo no es visualizado por los usuarios de la aplicación (atributo invisible), este presenta la visibilidad privada. La visibilidad privada es representada por el carácter "-". Si el atributo es visualizado por los usuarios (atributo visible), este presenta la visibilidad pública. Esta visibilidad es representada por el carácter "+", o sea, por default la visibilidad es pública.

El **nombre** es el identificador del atributo.

Tipo es la especificación del tipo de implementación de un atributo. Cuando el atributo presenta múltiples perspectivas, estas son mostradas entre corchetes, y la perspectiva default es indicada por un signo de "+".

Valor-default es el valor atribuido al atributo cuando un objeto es creado. El valor-default es opcional, y si el no va definido, el signo de "=" también es omitido.

En el caso de un atributo con múltiples perspectivas, el valor-default es colocado al lado de cada perspectiva, precedido del signo de “=”.

Propiedades indican informaciones adicionales aplicadas a los atributos, semejante a las propiedades de las clases. La especificación de las propiedades es opcional. Cuando las propiedades no fueran indicadas, las llaves son omitidas.

A continuación son mostrados algunos ejemplos de definición de atributos:

- +nombre: String
- título: String = Conferencia Nacional
- gratificación: Real = 0 {oriunda da circular 001}
- +descripción: [Texto+, Imagen]
- +descripción: [Texto+, Imagen = desc1.bmp]
- +descripción: [TextoMarketing, TextoPesquisa]

Cuando un atributo es calculado a partir de otros atributos (ej. Atributo derivado), es colocado una barra inclinada (“/”) en frente de su nombre. Por ejemplo, si el valor total gastado en un proyecto de investigación se va a sumar del valor gastado con carteras y con material, el atributo es representado de la siguiente manera:

/total-gasto: Integer

2.3.1.1.2.- Operaciones

Cuando es necesario, las clases pueden representar operaciones, o sea, pueden presentar cálculos sobre las informaciones de la aplicación.

Los nombres de las operaciones inician con una letra minúscula y son escritos en letra normal.

Las operaciones presentan la siguiente definición:

Visibilidad nombre (lista-parámetros) : expresión-resultado { propiedad }

La visibilidad de una operación, así como la de los atributos, pueden ser pública o privada, o sea, puede ser activada o no por el usuario. También no existe la visibilidad protegida para una operación en OOHD, como existe en UML.

Si la ejecución no puede ser ejecutada por los usuarios de la aplicación, o sea, es manipulada solamente internamente, ello presenta la visibilidad privada, representada por el signo "-". Por ejemplo, una operación para mostrar un video es manipulada internamente por la aplicación. Si la operación es visualizada por los usuarios, esta presenta la visibilidad pública, y no es representada explícitamente, lo contrario de UML.

Por ejemplo, una operación para alterar el valor de gratificación de un profesor puede ser manipulada por el usuario.

El **nombre** es el identificador de la operación.

Lista-parámetros es una lista de parámetros formales que son pasados para la operación, separados por coma. Cada parámetro es especificado por la siguiente sintaxis:

nombre : tipo = valor-default, donde nombre es el nombre del parámetro formal, tipo es la especificación del tipo de parámetro y valor-default es un valor opcional atribuido al parámetro. Si el valor-default no es especificado, el signo "=" también es omitido.

expresión-resultado es la especificación del valor retornado por la operación. Si la operación no retorna ningún valor, la expresión es omitida.

Propiedades indican la información adicional aplicadas a las operaciones, también semejante a las propiedades de las clases. La especificación de las propiedades es opcional. Cuando las propiedades no fueran indicadas, las llaves son omitidas.

A continuación son mostrados algunos ejemplos de definiciones de operaciones:

Cambio-posición (posición:coordenada = (20,20))

-muestra()

altera-gratificación ():Real

2.3.1.2.- Generalización

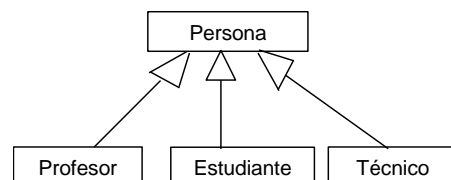
Aparte de clasificación, el concepto de generalización/especialización también es utilizado en la definición del esquema conceptual.

La generalización/especialización es utilizada para denotar que clases especializadas heredan todas las características (atributos, métodos y relacionamientos) de clases generalizadas, pudiendo agregar nuevas características.

Como ejemplo de generalización/especialización, podemos citar el relacionamiento entre Persona (clase generalizada) y Profesor (clase especializada).

La generalización/especialización es representada por una flecha conectando el elemento mas específico (subclase) al elemento mas general (superclase), así que la punta de flecha es un triangulo vacío posicionado al lado de la superclase.

La [Figura II.38](#) muestra las dos formas de representación de generalización/especialización (con diversas flechas o con una única flecha compartida).



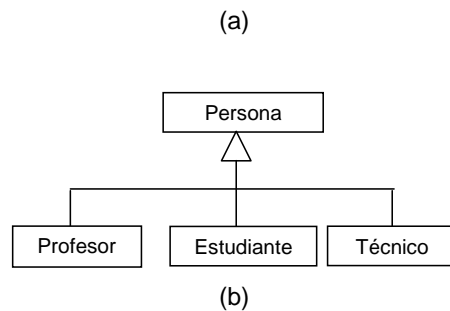


Figura II.38 Formas de representación de generalización/especialización

Cada camino de una subclase a la superclase puede poseer un discriminador.

Un discriminador representa el criterio usado en partición de las subclases. La [Figura II.39](#) muestra un ejemplo de generalización/especialización que presenta un discriminador.

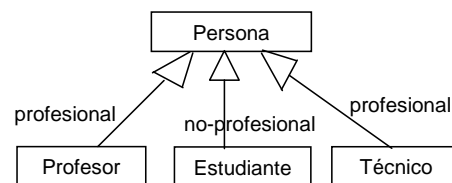


Figura II.39 Ejemplo de generalización/especialización con discriminador

Las restricciones semánticas existentes entre las subclases son definidas a través de una lista contenida entre llaves, que queda junto al triángulo compartido por todas las subclases (en el caso de representación compartida) o de una línea trazada que atraviesa las subclases involucradas.

Las siguientes restricciones entre las subclases son predefinidas:

- Sobrepuesta: una instancia puede descender de más de una subclase.
- Disjunto: una instancia no puede descender de más de una subclase, o sea, sino puede descender de una única clase.

- Completa: todas las posibles subclases precisan ser especificadas.
- Incompleta: solamente algunas subclases son especificadas. En este caso, existen otras subclases que aun no están en el modelo.

La [Figura II.40](#) muestra un ejemplo de jerarquía donde algunas restricciones son definidas. La restricción disjunta significa que una persona puede ser del tipo Profesor, Estudiante o Técnico, mas nunca de dos tipos simultáneamente. La restricción incompleta significa que además de Profesor, Estudiante y Técnico, pueden existir otros tipos de personas, pero, por el momento, solamente esos tipos fueron especificados en el modelamiento.

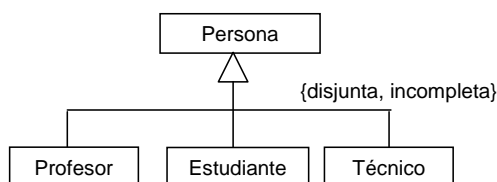


Figura II.40 Ejemplo de generalización/especialización con restricciones

La herencia de los atributos y operaciones es igual a la herencia tradicional de orientación a objetos, excepto la herencia de los atributos con múltiples perspectivas. En este caso, cuando una subclase define un atributo con el mismo nombre de un atributo con múltiples perspectivas heredado, las nuevas perspectivas del atributo son adicionadas a las perspectivas del atributo heredado.

En el caso de los atributos que presentan múltiples perspectivas, siendo una de estas default, si el nuevo atributo presenta una nueva perspectiva default (asignada por el signo "+"), la perspectiva default definida anteriormente pasa a ser una perspectiva opcional y la nueva perspectiva default pasa a ser la obligatoria.

Solamente una perspectiva default en toda la jerarquía puede no representar un identificador, recibiendo el identificador del propio atributo. La [Figura II.41](#)

II.41 muestra la herencia de un atributo con múltiples perspectivas y una perspectiva default.

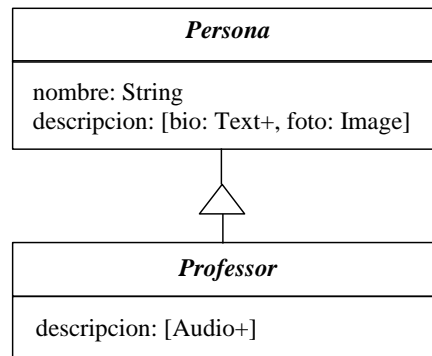


Figura II.41 Herencia de un atributo con múltiples perspectivas con perspectiva *default*

En el caso de los atributos con múltiples perspectivas y sin perspectiva default, las perspectivas del nuevo atributo serán mostradas de acuerdo con el contexto en el cual el objeto es visualizado.

La ~~Figura II.42~~ ~~Figura II.42~~ muestra la herencia de un atributo con múltiples perspectivas y sin perspectiva default.

Con formato: Fuente: (Predeterminado) Arial

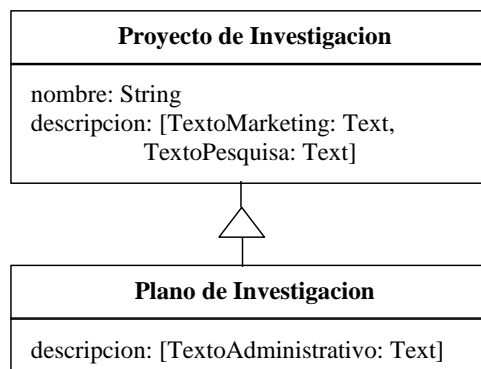


Figura II.42 Herencia de un atributo con múltiples perspectivas sin perspectiva *default*

2.3.1.3.- Relacionamientos

Los relacionamientos (denominados 'asociaciones' en UML) representan las ligaciones entre objetos. Son representados a través de una línea conectando una o más clases.

Un relacionamiento puede conectar una misma clase. En este caso, una instancia de esta clase esta relacionada con otra instancia de esta misma clase.

Por ejemplo, una disciplina puede requerir otra disciplina como pre-requisito.

Un relacionamiento puede conectar dos clases (relacionamiento binario). Por ejemplo, el relacionamiento enseñar entre la clase Profesor y Disciplina significa que un profesor enseña una disciplina ([Figura II.44](#)~~Figura II.44~~).

Un relacionamiento puede conectar tres o más clases (relacionamiento N-ario), donde una clase puede aparecer más de una vez. Un relacionamiento N-ario es representado por un rombo grande, con una línea conectando cada clase que participa del relacionamiento. En este caso, el nombre del relacionamiento es mostrado junto al rombo. Por ejemplo, el relacionamiento ternario enseñar-en entre Profesor, Disciplina y Sala de Aula ([Figura II.45](#)~~Figura II.45~~).

Los relacionamientos pueden presentar, junto a la línea que el representa, el nombre del relacionamiento. Este nombre, a su vez, puede tener un pequeño triángulo relleno de negro, indicando la dirección en el cual el nombre es leído.

La [Figura II.43](#)~~Figura II.43~~, [Figura II.44](#)~~Figura II.44~~ y [Figura II.45](#)~~Figura II.45~~ muestran, respectivamente, un relacionamiento que conecta una misma clase, un relacionamiento que conecta dos clases distintas y un relacionamiento que conecta tres clases. El relacionamiento enseñar presenta el pequeño triángulo junto a su nombre indicando que el relacionamiento también puede ser denominado de Profesor enseñar Disciplina.

En el caso de un relacionamiento unario, ese pequeño triángulo no es necesario, excepto cuando sea para representar un papel, como será visto más adelante.

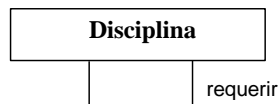


Figura II.43 Relacionamiento unario

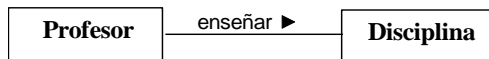


Figura II.44 Relacionamiento binario

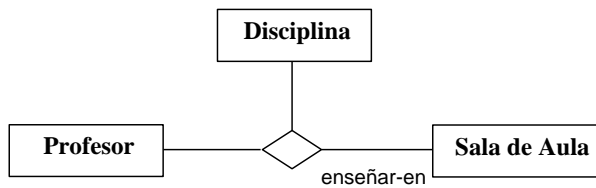


Figura II.45 Relacionamiento ternario

Cada clase que participa de un relacionamiento puede estar asociada a un papel, que describe la finalidad de sus instancias en aquel relacionamiento. Símbolos gráficos indicando las propiedades del papel pueden ser colocados junto a la clase del papel, en la línea que representa el relacionamiento en cuestión. Los papeles pueden presentar las propiedades de cardinalidad y nombre del papel.

2.3.1.3.1.- Cardinalidad

Para cada relacionamiento es necesario definir la cardinalidad de sus papeles. La cardinalidad es un subconjunto de enteros no negativos. Es representada por

una secuencia de intervalos de enteros separados por coma, donde cada intervalo representa un grupo de enteros en formato:

limite-inferior .. limite-superior, si la cardinalidad puede variar, o el limite, si la cardinalidad no puede variar. Para representar el limite superior infinito es usado el símbolo *.

A continuación se muestra varios ejemplos de cardinalidad de los papeles:

0..1 - la cardinalidad puede ser 0 o 1;

1 - la cardinalidad es igual a 1;

0..* - la cardinalidad puede variar de 0 hasta infinito;

* - la cardinalidad puede variar de 0 hasta infinito;

1..* - la cardinalidad puede variar de 1 hasta infinito;

1..6 - la cardinalidad puede variar de 1 hasta 6;

1..3, 7..10, 15,19..* - la cardinalidad puede ser 1, 2, 3, 7, 8, 9, 10, 15, y a partir de 19 hasta infinito.

La [Figura II.46](#) muestra el relacionamiento Profesor enseñar Disciplina, que tiene cardinalidad 1-hacia-N (1 hasta infinito), significando que un profesor enseña de 1 a N disciplinas y que una disciplina es enseñada por exactamente 1 profesor.

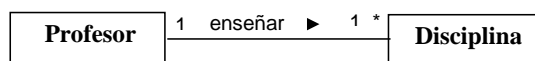


Figura II.46 Relacionamiento 1-N

2.3.1.3.2.- Nombre del Papel

Cuando la clase asume un papel en el relacionamiento, ese papel es representado por un nombre, colocado junto al de la conexión entre la clase y la relación. La [Figura II.47](#) muestra un ejemplo donde una disciplina presenta el papel pre-requisito.

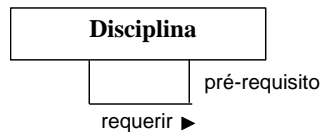


Figura II.47 Ejemplo de nombre del papel

Si fuera necesario especificar otras propiedades de la relación, esto debe ser hecho a través de un texto entre corchetes junto al final de la relación.

2.3.1.3.3.- Clases de Relaciones

Las relaciones pueden presentar propiedades de la clase, o sea, contener atributos y comportamientos. En este caso, estos son conocidos como clases de relaciones. Por ejemplo, la relación orientación entre la clase Profesor y la clase Estudiante puede tener los atributos titulación y resultado.

Una clase de relación es representada por un símbolo de clase conectado a la relación por una línea trazada. En este caso, la relación y la clase de relación forman un único elemento y, por eso, presentan un único nombre, que puede ser colocado en la línea de la relación, en la clase de relación, o en ambos. La [Figura II.48](#) muestra una relación que presenta una clase de relación.

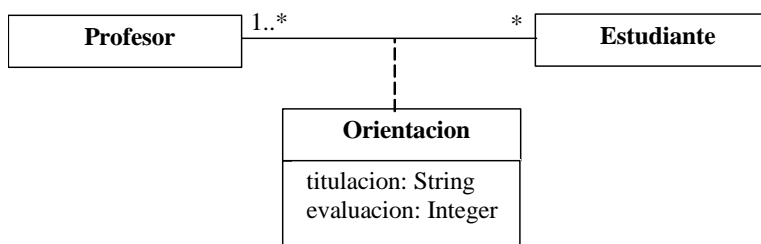


Figura II.48 Clase de relación

2.3.1.4.- Agregación y Composición

La agregación es empleada para indicar que un objeto es compuesto por otros objetos. Los objetos componentes existen independientemente de la existencia del objeto compuesto. Entretanto, los objetos componentes pueden ser parte solamente de un objeto compuesto. Como ejemplo de agregación, podemos citar que un laboratorio es compuesto por varios equipamientos.

La agregación es representada por una relación con un rombo vacío en el lado de la clase que agrega las otras. La cardinalidad del lado del objeto compuesto es siempre 1 y puede ser omitida. La [Figura II.49](#) muestra una agregación.

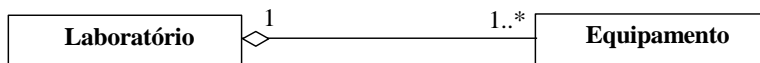


Figura II.49 Agregación

Cuando los objetos agregados dependen fuertemente del objeto agregador y solo existe mientras este existe, la agregación es del tipo composición. Así, los objetos que fueran parte del compuesto son creados después del compuesto y son destruidos junto con el compuesto. Los objetos componentes de la composición también solo pueden hacer parte de un objeto compuesto. Por ejemplo, una disciplina contiene un conjunto de material complementario, y si esta deja de ser ofertada, su material complementario también dejara de existir.

La composición es representada por un relacionamiento con un rombo relleno en el lado de la clase que agrega las otras. La cardinalidad esta junto al objeto compuesto también es siempre 1 y puede ser omitida. La [Figura II.50](#) muestra una composición.

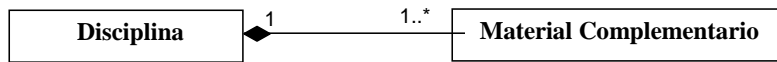


Figura II.50 Composición

2.3.1.5.- Diagrama de Objetos

Un diagrama de objetos muestra un conjunto de instancias relacionadas, de acuerdo con el diagrama de clases definido. En OOHDM, este es usado en las situaciones en que pueden existir objetos que no siguen el padrón de las clases definidas y que presentan restricciones en relación a los otros objetos del modelo.

Por ejemplo, un profesor visitante que, a pesar de no ser profesor del departamento, orienta a 3 estudiantes.

En caso que sea necesaria la representación de un objeto que no sigue las reglas de las clases definidas aun presenta una relación genérica con los otros objetos, este objeto es representado directamente en el diagrama de clases.

Por ejemplo, para evitar la creación de la clase Rector que presente solamente una instancia, es insertado el objeto Rector directamente en el diagrama de clases, pues el presenta el mismo relacionamiento con todos los profesores. La inclusión de un objeto en el diagrama de clases será mostrada en la sección 2.3.1.7.-.

La [Figura II.51](#) muestra el ejemplo de parte de un diagrama de objetos donde el objeto Profesor visitante no es instancia de ninguna clase. Cada objeto es representado por un rectángulo dividido en 2 partes.

La primera parte contiene el nombre del objeto seguido del carácter “.” y de la clase a la cual el pertenece, subrayados. En caso que el objeto no pertenezca a ninguna clase, el carácter “.” es omitido. El nombre del objeto también puede ser omitido incluyendo solamente el carácter “.” y el nombre de la clase, representando un objeto anónimo.

La segunda parte contiene la lista de los atributos del objeto y sus respectivos valores.

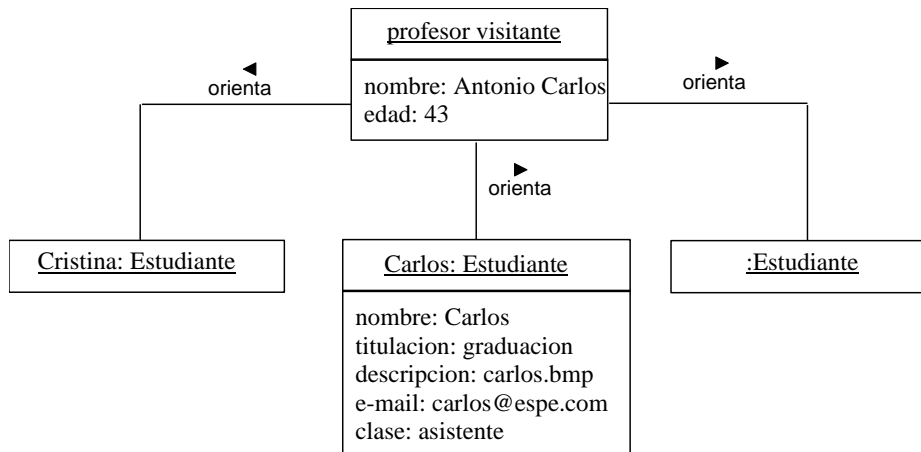


Figura II.51 Diagrama de objetos

2.3.1.6.- Subsistemas

Los subsistemas representan el agrupamiento de clases y relaciones que tratan de un mismo asunto y, juntos, son independientes del resto de la aplicación, o sea, son abstracciones de un sistema conceptual completo. En OOHDM, los subsistemas poseen semántica y la sintaxis es semejante a los *paquetes* de UML.

Se asume que cada clase presentada dentro de un subsistema está definida en esta. Cuando una clase es referenciada en un subsistema en el cual esta no está definida, esta debe ser presentada de la siguiente forma:

Nombre-Subsistema::Nombre-Clase.

La [Figura II.52](#) muestra la clase Profesor, definida en el subsistema Académico, siendo referenciada en el subsistema Contabilidad.

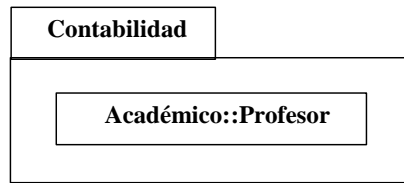


Figura II.52 Clase definida en otro subsistema

Si fuera necesario, un diagrama en el nivel de subsistemas es creado, y son mostradas las posibles referencias entre las clases de los subsistemas. En este diagrama, si el subsistema A puede referenciar clases del subsistema B, así existe una línea trazada y direccionada del subsistema A para el subsistema B con la palabra <<importa>>.

La [Figura II.53](#) muestra un diagrama de subsistemas, donde el subsistema Contabilidad referencia las clases del subsistema Académico.

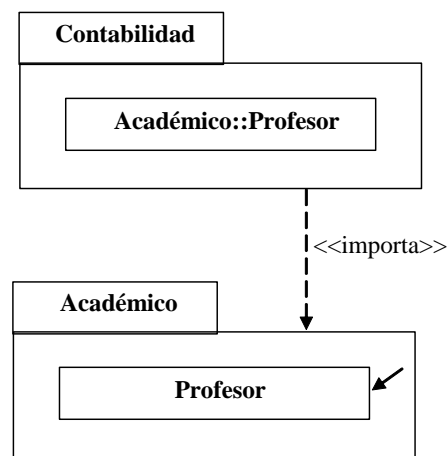


Figura II.53 Diagrama de subsistemas

Los posibles puntos de entrada del subsistema, las clases que permiten el acceso al subsistema, son representados por pequeñas flechas.

2.3.1.7.- Esquema Conceptual

El esquema conceptual es el diagrama resultante del modelo conceptual. En esta son representadas todas las clases representativas del dominio de la aplicación y sus relaciones, incluyendo generalización, composición y agregación, además de sus agrupamientos en subsistemas.

La ~~Figura II.54~~ Figura II.54 presenta un ejemplo de esquema conceptual para un dominio académico. Para evitar la creación de la clase Rector que presenta solamente una instancia, el objeto Rector es incluido directamente en el esquema conceptual.

Con formato: Fuente: 12 pto

2.3.1.8.- Tablas de Mapeamiento de los Objetos Conceptuales

Las tablas de mapeamiento representan el mapeamiento de los objetos conceptuales, específicamente las clases, atributos, relaciones y clases de relación, en objetos navegacionales, de interfaz y de implementación.

Después de terminar cada actividad (modelo navegacional, interfaz abstracta e implementación) es construida una tabla específica.

Debido que para un mismo modelo conceptual pueden ser definidas más de un modelo navegacional, y consecuentemente, más de un proyecto de interfaz abstracta e implementación, varias tablas de mapeamiento de un mismo tipo pueden ser construidas para un mismo modelo conceptual. En este caso, es necesario definir para cual aplicación la tabla esta siendo construida.

Siempre que un objeto conceptual es mapeado para otro tipo de objeto que presenta las mismas características, el objeto conceptual es considerado el objeto base y el mapeamiento es considerado directo.

En las tablas de mapeamiento, es necesario especificar solamente los objetos conceptuales que no son mapeados directamente para objetos navegacionales, de interfaz y de implementación.

Ese mapeamiento es especificado a través de una extensión del lenguaje de consulta para bases de datos¹⁵.

¹⁵ KIM, Won. Introducción a bases de datos orientados a objetos. Ed. MIT Press, 1990. 234 p.

2.3.1.8.1.- Tabla de Mapeamiento de los Objetos Conceptuales para Objetos Navegacionales

Esta tabla es construida después de la conclusión de actividades del modelamiento navegacional y presenta el mapeamiento de los objetos conceptuales para los objetos navegacionales.

La ~~Tabla II.2~~ Tabla II.2 presenta el mapeamiento del esquema conceptual de la ~~Figura II.54~~ Figura II.54 para el esquema navegacional que será presentado en la ~~Figura II.73~~ Figura II.73. Solamente las clases y atributos que no son mapeados directamente para el esquema navegacional son mostrados en la tabla.

Con formato: Fuente: 12 pto

Con formato: Fuente: 12 pto

Tabla II.2 Mapeamiento de los Objetos Conceptuales para Objetos Navegacionales

Objetos Mapeados	Regla de Mapeamiento
conceptual: Patrocinador.nombre	seleccionar Pa.nombre DE Pa: Patrocinador, Pr:Proyecto de Investigación DONDE Pa financia Pr
navegacional: ProyectoDeInvestigacion.nombrePatrocinador	
conceptual: Patrocinador	-----
navegacional: -----	
conceptual: Patrocinador financia ProyectoDeInvestigacion	-----
navegacional: -----	
conceptual: Repetición	
navegacional: DisciplinaEstudiante	mapeamiento directo
Estudiante repite DisciplinaEstudiante	mapeamiento directo
Disciplina posee DisciplinaEstudiante	mapeamiento directo
conceptual: Repetición.evaluacion	mapeamiento directo
navegacional: DisciplinaEstudiante.evaluacion	
conceptual: orientación	-----
navegacional: -----	
conceptual: Orientacion.titulacion	seleccionar O. titulación DE O: orientación, P: Profesor, E: Estudiante DONDE P orienta E
navegacional: Estudiante. titulación	
conceptual: Orientacion.evaluacion	seleccionar O.evaluacion DE O: orientación, P: Profesor, E: Estudiante DONDE P orienta E
navegacional: Estudiante.evaluacion	
conceptual: Estudiante.nombre	
navegacional: Estudiante.nombre	mapeamiento directo
DisciplinaEstudiante.nombreEstudiante	seleccionar E.nombre DE E: Estudiante, D: Disciplina DONDE E frecuenta D
conceptual: Disciplina.nombre	
navegacional: Disciplina.nombre	mapeamiento directo
DisciplinaEstudiante.nombreDisciplina	seleccionar D.nombre DE D: Disciplina, E: Estudiante DONDE E frecuenta D
conceptual: Persona.descripcion	
navegacional: Persona.descripcion	mapeamiento directo
Persona.bio	mapeamiento directo
conceptual: ProyectoDeInvestigacion.descripcion	
navegacional: ProyectoDeInvestigacionPorLaboratorio.marketing	mapeamiento directo
ProyectoDeInvestigacionPorProfesor.investigación	mapeamiento directo

2.3.1.8.2.- Tabla de Mapeamiento de los Objetos Conceptuales para Objetos de Interfaz

Esta tabla es construida después de la conclusión de la actividad del proyecto de interfaz abstracta y presenta el mapeamiento de los objetos conceptuales para los objetos de interfaz ([Tabla II.3](#)~~Tabla II.3~~).

Tabla II.3 Mapeamiento de los Objetos Conceptuales para Objetos de Interfaz

Objetos Mapeados	Regla de Mapeamiento
conceptual:	
interfaz:	

2.3.1.8.3.- Tabla de Mapeamiento de los Objetos Conceptuales para Objetos de implementación

Esta tabla es construida después de la conclusión de la actividad de implementación y presenta el mapeamiento de los objetos conceptuales para los objetos de implementación ([Tabla II.4](#)~~Tabla II.4~~).

Tabla II.4 Mapeamiento de los Objetos Conceptuales para Objetos de implementación

Objetos Mapeados	Regla de Mapeamiento
Conceptual:	
implementación:	

2.3.2.- DISEÑO NAVEGACIONAL

El diseño navegacional define la información que será presentada y la posible navegación entre ellas. El modelamiento navegacional es definido a partir de un modelo conceptual. Para un mismo modelo conceptual, pueden ser definidos uno o más modelos navegacionales. En este último caso, para un mismo dominio son definidos diferentes aplicaciones, o sea, para cada perfil de usuario es desarrollada una aplicación conteniendo parte o toda la información contenida en el esquema conceptual, dependiendo de las necesidades del usuario.

En el diseño navegacional son especificados cuales son los objetos navegacionales (nodos, enlaces, clases de contexto, estructuras de acceso, etc.) que serán vistos por el usuario y cuales son los contextos navegacionales.

En el diseño navegacional se proporcionara los siguientes modelos:

- Un esquema de clases navegacionales conteniendo la definición de nodos y enlaces.
- Un esquema de contextos identificando los contextos navegacionales y las estructuras de acceso.
- Un esquema de clases en contexto.
- Cartas especificando todos los objetos creados en el modelamiento navegacional (objetos navegacionales y contextos navegacionales).

2.3.2.1.- Esquema de Clases Navegacionales

Un esquema de clases navegacionales o esquema navegacional define un conjunto de nodos y enlaces que forman parte de una vista navegacional de la aplicación. Una aplicación puede tener uno o más esquemas navegacionales de acuerdo con las vistas existentes de la aplicación.

Los nodos y enlaces proceden de las clases y relacionamientos que componen el esquema conceptual.

El esquema navegacional es un diagrama donde los nodos son representados por rectángulos y los enlaces por líneas, así como en el esquema conceptual.

Entretanto, las líneas que representan los enlaces son direccionadas representando la posible navegación entre la información existente.

2.3.2.1.1.- Nodos

Un nodo, también denominado clase navegacional, representa un conjunto de instancias que presentan las mismas características (atributos, enlaces y métodos). Un nodo es procedente de una clase, un conjunto de clases o una clase del relacionamiento del esquema conceptual. Así como las clases del esquema conceptual, los nodos también pueden ser organizados en jerarquía de generalización/especialización.

Como ejemplo de nodo podemos citar el nodo Profesor que contiene toda la información de la clase conceptual Profesor.

Un nodo es un estereotipo UML¹⁶ de clase conceptual. Este también es representado por un rectángulo dividido en 3 partes ([Figura II.32](#)~~Figura II.32~~). La primera parte contiene el nombre de la clase y otras propiedades generales de la clase. La segunda parte contiene la lista de atributos definidos para las instancias de la clase. La tercera parte contiene la lista de operaciones definidas para las instancias de la clase. Entretanto, para diferenciar un nodo de una clase conceptual, es incluida una línea vertical en la primera parte a la derecha, como se muestra en la [Figura II.55](#)~~Figura II.55~~.

¹⁶ <http://www.rational.com/uml>

Nombre-Nodo	
lista-atributos	
lista-operaciones	

Figura II.55 Representación de un nodo

La representación de las características de los nodos es semejante a la representación de las clases conceptuales. El nombre de un nodo debe ser escrito en negrita, centrado y con la primera letra en mayúscula. Sus propiedades deben ser colocadas entre llaves "{ }". Los atributos y operaciones son escritos con letra normal y justificada a la izquierda. Los nombres de los nodos abstractos (clases navegacionales abstractas) deben ser escritos en cursiva. Así mismo como acontece en el modelamiento conceptual, en el modelamiento navegacional también no existen operaciones abstractas.

La [Figura II.56](#) muestra la representación del nodo Profesor, procedente de la clase conceptual Profesor. La [Figura II.57](#) muestra la representación del nodo abstracto Persona, procedente de la clase abstracta Persona. La [Figura II.58](#) muestra el nodo Disciplina Estudiante procedente de la clase del relacionamiento Frecuencia.

Profesor { autor=Sandro }	
clase: String	
modificar-gratificación ():	

Figura II.56 Representación del nodo Profesor

Persona	
nombre: String grado: String descripción: Imagen * biografía: Texto	

Figura II.57 Representación del nodo abstracto Persona

Disciplina Estudiante	
nombre-estudiante: String semestre: String disciplina-curso: String evaluación: Integer	

Figura II.58 Representación del nodo Disciplina Estudiante

Un nodo también puede ser representado de forma simplificada, así como las clases conceptuales. La [Figura II.59](#) muestra ejemplos de representaciones simplificadas del Proyecto de Investigación, mostrando solamente el identificador del nodo, el identificador y sus atributos, y el identificador y sus operaciones.

Proyecto de Investigación	
Proyecto de Investigación	
Proyecto de Investigación {from p: Proyecto de Investigación}	
Proyecto de Investigación nombre: String descripción: String presupuesto: Real patrocinadores: list of <pa: Patrocinador, pa.nombre where pa <i>financia</i> p>	
Proyecto de Investigación modificar-presupuesto (): Real	

Figura II.59 Representaciones simplificadas de un nodo

1) Atributos

Los atributos de los nodos representan las propiedades de sus instancias. Por ejemplo, toda la instancia del nodo Disciplina Estudiante presenta un nombre-estudiante, semestre, nombre-disciplina y evaluación.

Cada nodo puede presentar todos los atributos de la clase conceptual de la cual fue originado, solamente algunos de sus atributos (vista de la clase), o sus atributos y de las otras clases.

Por ejemplo, el nodo Proyecto de Investigación contendrá los atributos de la clase Proyecto de Investigación y también el atributo patrocinador originario del atributo nombre de la clase Patrocinador.

A pesar toma atributos de las clases conceptuales pudiendo presentar múltiples perspectivas, los atributos de los nodos solo pueden presentar una única perspectiva, y deben ser de un tipo predefinido simple, por ejemplo: String, Integer, Real, etc., complejo como el Array, o definido en la aplicación. Así, un atributo con múltiples perspectivas de una clase conceptual que posee una perspectiva por default es mapeado, en el modelamiento navegacional, para varios atributos de un nodo, uno para cada perspectiva.

La perspectiva por default es mapeada para un atributo obligatorio, semejante a los atributos comunes. Las perspectivas que no son por default son mapeadas para atributos opcionales, que son seguidos por el símbolo “ * ”.

La [Figura II.60](#) ~~Figura II.60~~ presenta el nodo abstracto Persona que contiene los atributos descripción y biografía, originados de un atributo con múltiples perspectivas y con la perspectiva default, el atributo descripción de la clase abstracta Persona. El asterisco (*) significa que un objeto puede no presentar un valor para el atributo descripción.

<i>Persona</i>	
nombre: String	
grado: String	
descripción: Imagen *	
biografía: Texto	

Figura II.60 Atributos mapeados del atributo con múltiples perspectivas

Los atributos con múltiples perspectivas y sin perspectiva default son mapeados para las clases en contexto, como será visto en la sección 0.

Los nombres de los atributos de los nodos generalmente son escritos en minúsculas y con fuente normal. Los atributos presentan la misma definición de los atributos de las clases conceptuales.

visibilidad nombre : tipo = valor _default { propiedades }

Siguiendo el padrón adoptado en el modelamiento conceptual de OOHDM, la visibilidad de un atributo puede ser privada o publica, representando, respectivamente, que un atributo es invisible o visible. La visibilidad publica es por default y no es representada, en cuanto a la visibilidad privada es representado por la señal “ - ”.

El nombre es el identificador del atributo. Tipo es la especificación del tipo de implementación de un atributo. Valor por default es el valor atribuido al atributo cuando un objeto es creado. El valor-default es opcional. Propiedades indican información adicional aplicada a los atributos y también son opcionales.

A continuación se muestran algunos ejemplos sobre definición de atributos de los nodos:

nombre: String
título: String = Conferencia Nacional
-gratificación: Real = 0 {originario del circular 001}
descripción: Texto
descripción: Imagen *

Los atributos derivados son precedidos por una barra inclinada(/). Por ejemplo, el valor total de gastos de un proyecto de investigación es representado de la siguiente manera.

/total-gastos: Integer

Las anclas e índices de un nodo también son especificados como atributos, y son, respectivamente, del tipo Ancla e Índice.

Los índices de acceso de los elementos de un contexto presentan la siguiente sintaxis.

nombre-atributo-índice : nombre-tipo-índice (parámetros-índice)

Las anclas de acceso pueden permitir el acceso de un índice o la de un elemento dentro de un contexto y presenta la siguiente sintaxis:

nombre-atributo-ancla : anchor (nombre-tipo-índice (parámetros-índice))

o

nombre-atributo-ancla : anchor (nombre-tipo-contexto (parámetros-contexto), [elemento])

Los nombres identificadores de índices y de contextos deben ser únicos, o sea, no pueden existir índices o contextos con el mismo nombre. Se adopta como práctica que: identificadores de índices y de contextos deben indicar cuales son sus elementos integrantes; identificadores de índices deben ser denominados en

plural e identificadores de contextos en singular, identificadores de índices deben ser precedidos por el prefijo Idx e identificadores de contexto por el prefijo Ctx.

Por ejemplo, Estudiantes por Profesor sería un identificador de índice, en cuanto Estudiante por Profesor sería un identificador de contexto. En este caso, los elementos del índice y del contexto serían los estudiantes de un determinado profesor.

La [Figura II.61](#) muestra la clase Profesor que presenta las anclas disciplinas y estud y el índice área-ind como sus atributos.

Profesor	
clase: String área-ind: Idx Áreas Investigadas por Profesor (self) disciplinas: anchor (Idx Disciplinas por Profesor (self)) estud: anchor (Ctx Estudiante por Profesor (self))	
modifica-gratificación (): Real	

Figura II.61 Anclas e índice como atributo

Los atributos de esta clase presentan el siguiente significado:

- área-ind y un índice del tipo área de Investigación por Profesor. Ese tipo de índice presenta las áreas de investigación de un profesor, aquel es pasado como parámetro, en el caso, el propio profesor (self).
- disciplinas y una ancla cuyo destino es un índice del tipo Disciplinas por Profesor. Ese tipo de índice presenta las disciplinas impartidas por el profesor que es pasado como parámetro, en el caso, del propio profesor (self).
- estud es una ancla cuyo destino es un elemento en el contexto Estudiante Orientado por Profesor. Este contexto es formado por los estudiantes que son orientados por el profesor que es pasado como parámetro, en el caso del propio profesor (self).

Dependiendo del contexto en el cual el nodo es presentado, el puede presentar otros atributos. Esos atributos son especificados en las clases en contexto, como será explicado adelante. Los atributos especificados en los nodos son mostrados en todos los contextos en que el nodo aparece.

2) Operaciones

Así como las clases conceptuales, los nodos o clases navegacionales también pueden presentar operaciones. Los nombres de las operaciones también inician con una letra minúscula y son escritos en fuente normal.

Las operaciones de los nodos presentan la misma definición de las operaciones de las clases conceptuales:

visibilidad nombre (lista _ parámetros) : expresión-resultado { propiedades }

Así como el modelamiento conceptual, la visibilidad de una operación puede ser privada, significando que una operación no puede ser ejecutada por el usuario, o publica, significando que una operación puede ser ejecutada por el usuario. La visibilidad privada es representada por la señal " - " y la visibilidad publica no es representada. El nombre es el identificador de la operación, lista-parámetros es una lista de los parámetros formales que son pasados por la operación. Esos parámetros son representados como el modelamiento conceptual, expresión-resultado es la especificación del valor retornado por la operación. Propiedades indican las informaciones adicionales aplicadas a las operaciones, y son opcionales.

A continuación se muestra algunos ejemplos de definiciones de operaciones:

cambiar-posición (posición : Coordenada = (20,20))

-mostrar ()

cambiar-gratificación (): Real

2.3.2.1.2.- Generalización

Así como en el esquema conceptual, la generalización/especialización es utilizada para notar que clases navegacionales especializadas (nodos especializados) heredan todas las características (atributos, métodos y relacionamientos) de clases navegacionales generalizadas (nodos generalizados), pudiendo aumentar nuevas características.

Se puede citar como ejemplo de generalización/especialización el mismo ejemplo, el relacionamiento entre Persona (clase navegacional generalizada) y Profesor (clase navegacional especializada).

La generalización/especialización es representada en el esquema navegacional de manera semejante a su representación en el esquema conceptual.

La [Figura II.62](#) muestra las dos alternativas de representación de generalización/especialización (con diversas flechas o con una flecha compartida).

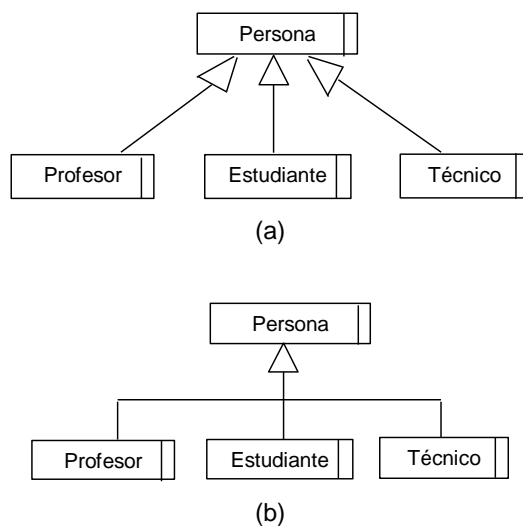


Figura II.62 Formas de representación de generalización/especialización

Cada camino de una subclase a la superclase en el esquema navegacional puede poseer un discriminador. Las subclases también pueden presentar las restricciones: sobrepuesta, disyunta, completa e incompleta.

La [Figura II.63](#) muestra una generalización que presenta un discriminador y algunas restricciones.

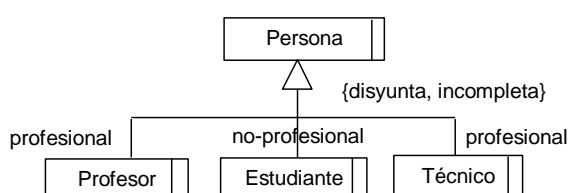


Figura II.63 Ejemplo de generalización/especialización con discriminador y restricciones

2.3.2.1.3.- Enlaces

Los enlaces, también llamados relacionamientos navegacionales, hacen las conexiones entre los objetos navegacionales (nodos y estructuras de acceso) y son, generalmente, procedentes de los relacionamientos presentes en el esquema conceptual.

Un enlace puede conectar dos instancias de nodos diferentes (enlace binario) o de un mismo nodo (enlace unario). Para el primer caso, se puede citar el ejemplo del enlace que conecta un profesor a las disciplinas por el impartido ([Figura II.64](#)).

El pequeño triángulo mostrado junto al nombre del enlace indica la dirección en la cual su nombre es leído, o sea, el enlace enseña también y puede ser denominado el Profesor enseña Disciplina. Para el segundo caso, el ejemplo de un nodo que conecta una disciplina, las disciplinas que son sus pre-requisitos ([Figura II.65](#)). En el esquema navegacional no existen enlaces conectando tres o más nodos.

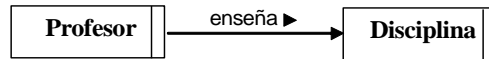


Figura II.64 Enlace binario

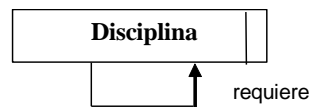


Figura II.65 Enlace unario

En el esquema navegacional, solamente los relacionamientos conceptuales que permiten la navegación entre objetos son representados como enlaces. La representación es hecha de manera semejante a la representación de los relacionamientos unarios y binarios del esquema conceptual. Entretanto, se añade una flecha que representa la navegación permitida a partir del enlace. Los enlaces unidireccionales representan que la navegación entre los objetos es permitida solamente en la dirección de la flecha. Los enlaces bidireccionales representan que la navegación entre los objetos es permitida en ambas direcciones.

Aparte de ser representados por enlaces, la navegación también debe ser representada por una ancla o índice como atributo de la clase navegacional.

En la [Figura II.65](#) el enlace requiere representa que a partir de una disciplina pueden ser accedidas las disciplinas que son requeridas como su pre-requisito, mas no pueden ser accedidas las disciplinas de la cual es pre-requisito.

La [Figura II.66](#) el enlace representa que a partir de una disciplina pueden ser accedidas las disciplinas que requiere como pre-requisito y también las disciplinas que requiere esta como pre-requisito.

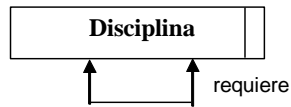


Figura II.66 Enlace bidireccional

Los enlaces, a lo contrario de los relacionamientos conceptuales, no pueden poseer atributos y métodos, o sea, no existen enlaces de asociación.

Los nodos que participan de un enlace pueden estar asociados a papeles. Los papeles de los enlaces presentan las mismas propiedades de los papeles de los relacionamientos conceptuales: cardinalidad, ordenación y nombre del papel.

1) Cardinalidad

La cardinalidad de un enlace es representada de manera semejante a la cardinalidad de un relacionamiento, o sea, a través de una secuencia de intervalos de enteros separados por comas, donde cada intervalo representa una línea de enteros en formato limite-inferior, limite-superior.

La [Figura II.67](#) muestra el enlace Profesor enseña Disciplina, que tiene cardinalidad 1..*, significando que un profesor enseña de 1 a N disciplinas y que una disciplina es enseñada por exactamente 1 profesor.

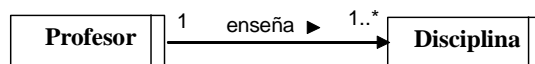


Figura II.67 Enlace 1-N

El atributo de la clase navegacional que representa un enlace con cardinalidad 1..N debe ser del tipo Índice o una ancla para un índice, en cuanto al atributo de la clase navegacional que representa un enlace con cardinalidad 1 debe ser del tipo Ancla.

2) Ordenación

Cuando el papel de un enlace presenta una cardinalidad mayor que uno, los elementos del conjunto pueden estar ordenados o no. El default es el conjunto no esta ordenado y, en este caso, nada precisa ser representado. Si el conjunto esta ordenado, debe ser usada la palabra (ordenado). Entretanto, en el nivel navegacional, no es especificado como esta hecha la ordenación y como esta es mantenida.

Cuando el papel asociado de una clase navegacional tiene la propiedad {ordenado}, todos los contextos en los cuales los objetos de esta clase navegacional son accedidos a través de este, deben seguir el mismo criterio de ordenación. Si varios contextos son derivados de este enlace y presentan diferentes criterios de ordenación, la propiedad {ordenado} no debe ser utilizada.

La [Figura II.68](#) muestra el ejemplo de un enlace ordenado entre las clases Profesor y área de Investigación, donde los profesores pertenecientes a una área están ordenados.

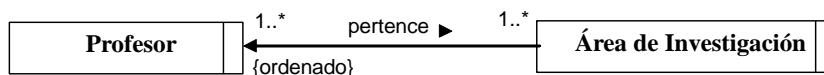


Figura II.68 Enlace con ordenación

3) Nombre del Papel

Cuando un nodo del enlace presenta un papel, el nombre de ese papel es colocado cerrando la conexión entre la clase y el relacionamiento.

La [Figura II.69](#) muestra un ejemplo de una disciplina que presenta el papel de pre-requisito.

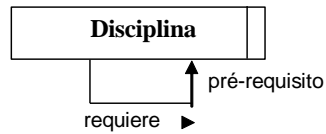


Figura II.69 Ejemplo de nombre del papel en un enlace

De manera semejante al esquema conceptual, otras propiedades son especificadas a través de un texto entre corchetes cerrados y la línea que representa el enlace

4) Cartas de Enlace

Todo enlace debe también ser especificado por una carta de enlace. La carta de enlace contiene el nombre del enlace, el destino del enlace, la semántica de navegación, comentarios pertinentes del enlace, los objetos de los cuales el enlace depende y los objetos que al enlace influyen. El destino del enlace puede ser dinámico o estático. Dinámico significa que el contenido del enlace destino es definido durante la ejecución de la aplicación. Estático significa que el contenido del enlace destino es predefinido y nunca es alterado.

La semántica de navegación puede ser origen persistente u origen no persistente. En el caso de origen persistente, el nodo origen es mantenido abierto, en cuanto al caso de origen no persistente, el nodo origen es cerrado.

Tabla II.5 Carta de enlace (orienta)

Enlace: orienta	
Destino del Enlace: estático	
semántica de navegación: origen persistente	
Comentarios:	
Depende de: relacionamiento conceptual orienta	Influencia:

2.3.2.1.4.- Agregación y Composición

La agregación y composición en el modelamiento navegacional posee la misma semántica del modelamiento conceptual. La agregación indica que un objeto es compuesto por otros objetos que no son dependientes de él, y es representado por un enlace con un rombo vacío del otro lado del nodo que agrega los otros.

La [Figura II.70](#) muestra como ejemplo de agregación un laboratorio que es compuesto por varios equipos

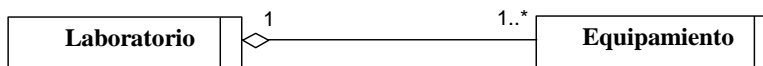


Figura II.70 Agregación de nodos

La composición indica que un objeto es compuesto por otros objetos que dependen fuertemente de este, existiendo solamente en cuanto el objeto compuesto existir. La composición es representada por un enlace con un rombo negro al lado de la clase que agrega las otras.

La [Figura II.71](#) muestra la representación de una composición, donde una disciplina es compuesta por un conjunto de materiales complementarios, y un material complementario solo existe si forma parte de una disciplina.

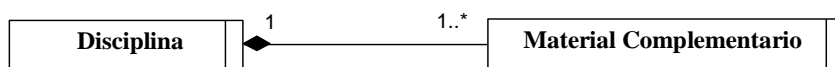


Figura II.71 Composición de nodos

Es interesante citar el modelo relacionado con la navegación en nodos compuestos. Cuando se accede a un nodo componente a partir del nodo compuesto, el nodo compuesto permanece activo, juntamente con el nodo componente, cuando se accede a un nodo componente a partir de otro nodo

componente, el nodo componente origen es desactivado y excluido del espacio navegacional, permaneciendo abierto el nuevo nodo componente y el nodo compuesto.

2.3.2.1.5.- Diagrama de Objetos

Un diagrama de objetos en el modelamiento navegacional tiene la misma función de un diagrama de objeto en el modelamiento conceptual, este muestra un conjunto de instancias relacionadas, de acuerdo con el diagrama de clases navegacionales definido.

Este también es usado en las situaciones en que pueden existir objetos que no siguen el padrón de clases definidas y que presentan restricciones en relación a otros objetos del modelo.

Cuando un objeto no sigue las reglas de las clases definidas mas presenta una relación genérica con los otros objetos, este es representado directamente en el diagrama de clases navegacional.

La [Figura II.72](#) muestra el ejemplo de parte de un diagrama de objetos navegacional semejante al diagrama de objetos conceptual de la [Figura II.51](#). Los nodos también son representados de manera semejante a los objetos conceptuales, o sea, un rectángulo dividido en dos partes, donde la primera contiene el nombre del objeto y su clase, la segunda parte contiene la lista de atributos del objeto y respectivos valores. La diferencia entre los diagramas es que el diagrama de objetos navegacionales incluye la representación de la navegación a través de líneas direccionadas.

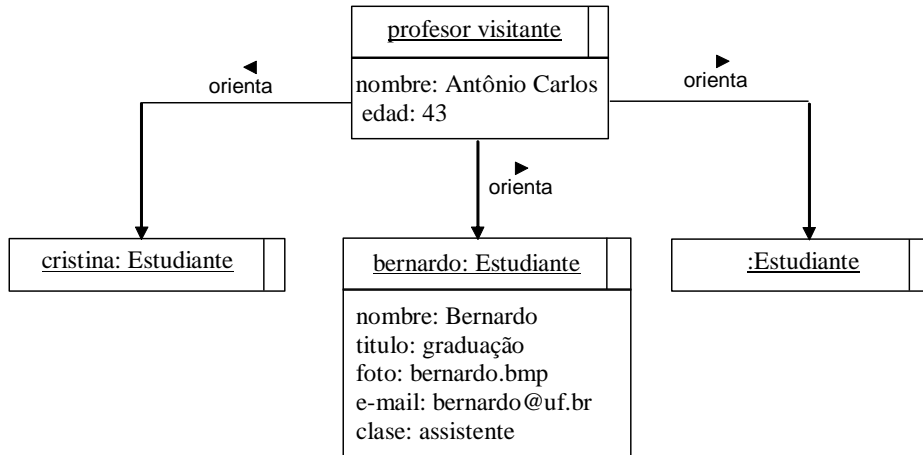


Figura II.72 Diagrama de objetos

2.3.2.1.6.- Esquema de Clases Navegacionales

La [Figura II.73](#) ~~Figura II.73~~ presenta un esquema navegacional derivado del esquema conceptual presentado en la [Figura II.54](#) ~~Figura II.54~~. Este esquema incluye el objeto rector. A pesar de las posibles navegaciones entre las instancias de las clases navegacionales serán representadas por las líneas direccionadas y también por las anclas e índices como atributos, solamente algunas anclas e índices están siendo representados.

Con formato: Fuente: 12 pto

Con formato: Fuente: 12 pto

2.3.2.2.- Esquema de Contextos de navegación

Mas allá del esquema de clases navegacionales, es necesario definir en cuales contextos será permitida la navegación entre la información y cual información será presentada.

2.3.2.2.1.- Contexto de navegación

Un contexto de navegación es un conjunto de objetos (nodos, enlaces y, recursivamente otros contextos navegacionales) que están relacionados de acuerdo con algún aspecto (ejemplo los nodos del mismo tipo que presentan un atributo con el mismo contenido; los nodos de tipos diferentes que presentan una características en común). Un contexto puede contener objetos de una única clase o puede contener objetos de varias clases.

Como ejemplos de contextos navegacionales podemos citar: todos los profesores del departamento, los profesores de una determinada área de investigación, las disciplinas impartidas por un profesor, los profesores y estudiantes que practican un determinado deporte, etc.

Los objetos de un contexto de navegación pueden ser parte de varios contextos. Por ejemplo, un profesor puede participar del contexto Profesor por Departamento (donde son accedidos todos los profesores) y del contexto Profesor por área (donde son accedidos todos los profesores pertenecientes a una área; ejemplo Redes de Computadores). Independientemente el contexto en la cual un objeto esta siendo accedido, este presenta todos los atributos definidos en el esquema de navegación. Entretanto, este puede presentar apariencia y atributos diferentes de acuerdo con el contexto, como será visto adelante.

1) Representación de los contextos

Un contexto es representado por un rectángulo conteniendo un identificador, los contextos generalmente son colocados dentro de un rectángulo sombreado, que representa la clase navegacional de sus elementos. El identificador clase navegacional es escrito en cursiva en la parte superior del rectángulo sombreado.

La [Figura II.74](#) ~~Figura II.74~~ presenta los contextos Profesor Alfabético, que contienen todos los profesores, y Profesor por Título Igual a Doctor, que contienen solamente los profesores que son doctores. El triángulo representado junto al contexto significa que los elementos del contexto pueden presentar varios criterios de ordenación.

Por ejemplo, puede ser especificado que los profesores pertenecientes a ese contexto pueden ser ordenados por orden alfabética o por tiempo de servicio.

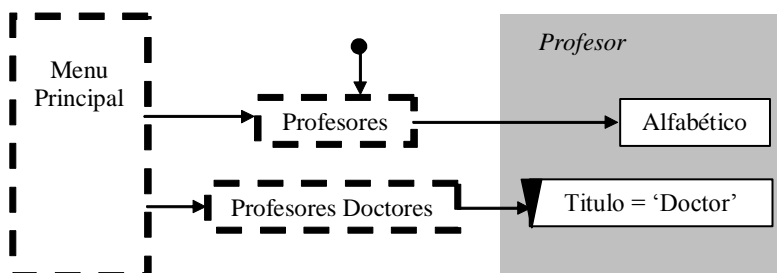


Figura II.74 Representación de un contexto

Generalmente, el acceso de los elementos del contexto es hecho a través de una estructura de acceso (índice). Las estructuras de acceso son representadas por rectángulos de líneas troceadas. Las estructuras de acceso exclusivas para el acceso a los elementos de una clase en determinado contexto generalmente poseen como identificador el nombre de la clase en plural.

Por ejemplo, la estructura de acceso para acceder a los elementos de la clase Profesor en el contexto Profesor Alfabético es denominado Profesores.

Para representar que una estructura de acceso (índice) o elemento de un contexto puede ser accedido a partir de cualquier sitio, es utilizada una flecha con un círculo negro en una de las extremidades, (implementando el “design pattern” Landmark). En la [Figura II.74](#) el índice Profesores puede ser accedido a partir de cualquier objeto de navegación de la aplicación.

La [Figura II.75](#) presenta el contexto Profesor por área Igual a Redes de Computadores. En este contexto los elementos del conjunto profesores que trabajan en la área Redes de Computadores son accedidos a partir del Profesor *pertenece-a* área de Investigación (enlace 1-para-N). El pequeño rectángulo en la parte superior izquierda del contexto representa un índice asociado al contexto.

Ese rectángulo es utilizado generalmente para simplificar la representación del esquema de contexto.



Figura II.75 Ejemplo de un contexto

2) Contextos Enumerados

Los elementos de un contexto de navegación también pueden ser escogidos arbitrariamente a partir de una o más clases (contexto enumerado). Como ejemplo de este tipo de contexto podemos citar un guión sobre las personas pertenecientes al departamento que son adeptas a los deportes radicales.

La [Figura II.76](#) muestra el contexto Deportistas Radicales que esta formado por objetos de la clase Profesor y por objetos de la clase estudiante. Los

elementos accedidos a partir de ese contexto son presentados en la carta de contexto, como será visto adelante.

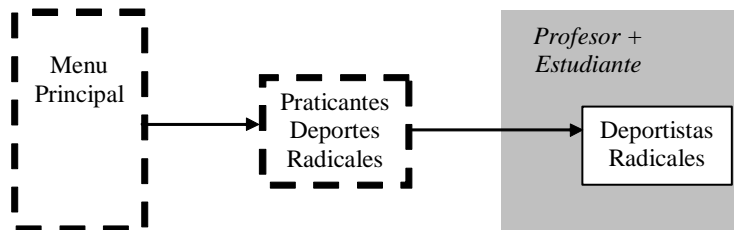


Figura II.76 Contexto con elementos de varias clases

En ciertas circunstancias, puede ser necesario presentar explícitamente todas las instancias de las clases navegacionales que son accedidas en un contexto enumerado. En este caso, el contexto es representado por un rectángulo con un identificador, conteniendo otros rectángulos con bordes redondeados, que representan las instancias de los objetos navegacionales que hacen parte del contexto. Como las instancias son presentadas en el diagrama del contexto, estas no precisamente son especificadas en la carta de contexto.

La [Figura II.77](#) ~~Figura II.77~~ presenta el contexto Tierras donde se muestra las instancias de la clase Laboratorio que están localizadas en el piso Tierras. La flecha con círculo representa que el Laboratorio de Redes puede ser accedido a partir de cualquier punto de la aplicación.

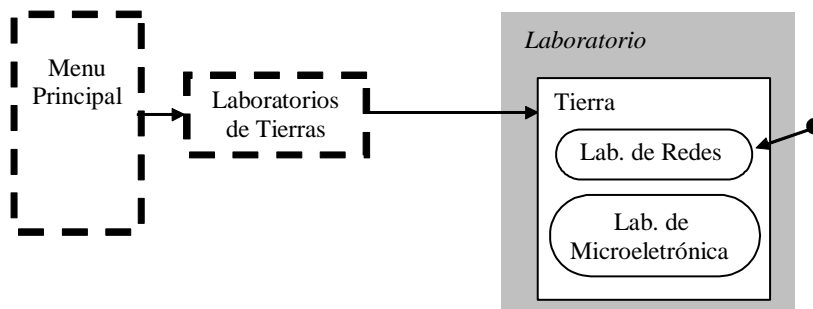


Figura II.77 Contexto de instancias

3) Contexto Por Consulta

Un contexto de navegación también puede ser formado por los elementos resultantes de una consulta realizada en tiempo de ejecución. Los parámetros de esta consulta no pueden ser previstos por el sistema antes de la ejecución de la propia consulta. Estas son definidas en tiempo de ejecución, automáticamente por el sistema o interactivamente por el usuario.

La [Figura II.78](#) ~~Figura II.78~~ presenta un contexto donde la consulta es realizada en la clase Proyecto de Investigación. Esta consulta posee como posibles parámetros un profesor, un alumno, área de investigación, del dato de inicio del proyecto o una combinación de estas. La determinación de los parámetros que sean escogidos en la definición del índice es hecha solamente en tiempo de ejecución. El índice de acceso a un contexto por consulta debe presentar una barra vertical negra en el lado derecho, indicando que la entrada es determinada durante la navegación (índice dinámico). Los nombres de los campos que forman los parámetros de la consulta son especificados entre “ < > ”.

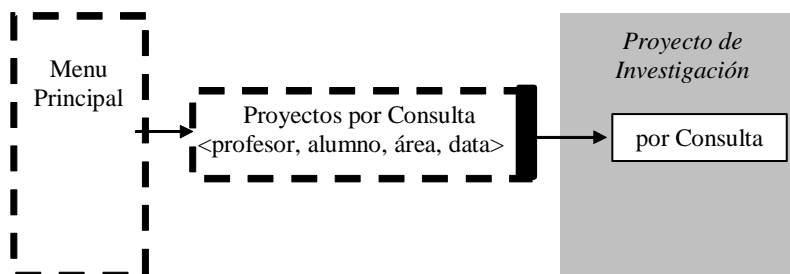


Figura II.78 Contexto de navegación por consulta

4) Grupo de Contexto

En OOHDM, a parte del concepto de contexto, existe el concepto de grupo de contexto, que es un conjunto de contextos. La ventaja de especificar un grupo de contextos es la compactación de especificación a través de una especificación

parametrizada. Esos grupos de contextos son representados de manera similar a un contexto simple, o sea, a través de un rectángulo conteniendo un identificador.

Por ejemplo, el conjunto de todos los contextos para los profesores de acuerdo con el título es representado por el grupo de contexto, Profesor por Título. Este grupo de contexto representa el conjunto de los contextos Profesor por Título Igual a Doctor, Profesor por Título Igual a Magíster, Profesor por Título Igual a Graduado etc. [Figura II.79](#) ~~Figura II.79~~ presenta el grupo de contexto Profesor por Título.

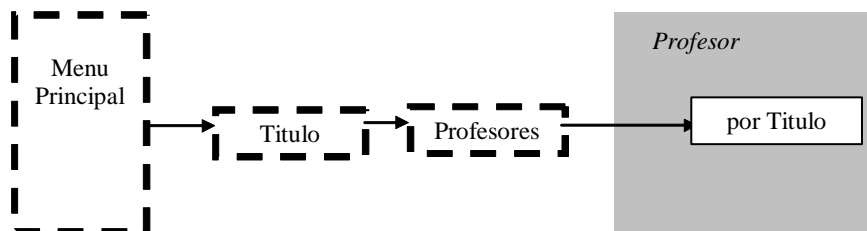


Figura II.79 Grupo de contexto

Un grupo de contextos, en este caso, es obtenido a través de la determinación de una propiedad de los objetos, Título, cuyos posibles valores determinan cada contexto que forma el grupo. Por tanto, un grupo de contextos puede ser determinado a través de una propiedad con un parámetro, y cada contexto de este grupo será determinado por la propiedad instanciada con cada valor del parámetro. En el ejemplo de la [Figura II.79](#) ~~Figura II.79~~ tenemos:

Profesor por título : $C_{tit} = \{(P \text{ pertenece a Profesores}, P.titulo=tit)\}$

La [Figura II.80](#) ~~Figura II.80~~ presenta otro ejemplo de grupo de contexto. El grupo de contexto Profesor por área representa todos los contextos derivados del enlace (1-a-N) Profesor *pertenece-a* área de Investigación, o sea, representa el conjunto de los contextos Profesor por área Igual a Redes de Computadoras, Profesor por área Igual a Ingeniería de Software, Profesor por área Igual a Inteligencia Artificial etc.

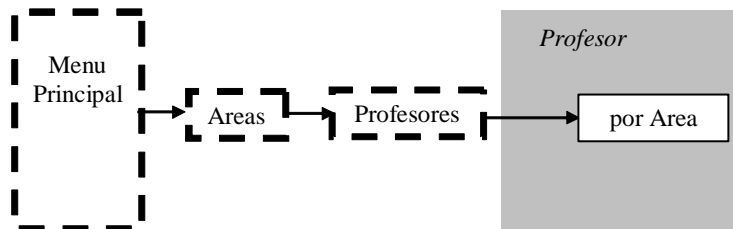


Figura II.80 Ejemplo de grupo de contexto

Este es otro ejemplo en la cual el conjunto de los contextos simples es formado a partir de un enlace 1-N. Para cada valor de origen, se forma un contexto con los N valores posibles del destino

5) Parámetro del Enlace de Acceso a un Contexto

El enlace de acceso de un contexto puede presentar un parámetro. En este caso, el parámetro puede indicar un elemento de un contexto, un elemento de un grupo de contexto o un contexto de un grupo del contexto. La [Figura II.81](#) ~~Figura II.81~~ presenta tres ejemplos de parámetros pasados por los enlaces.

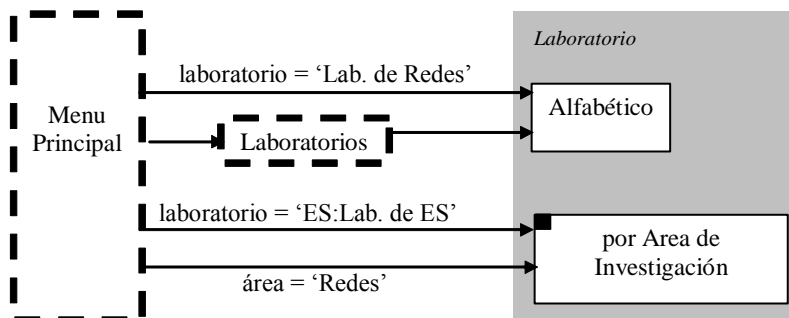


Figura II.81 Paso de parámetros para un contexto y grupo de contexto

El parámetro laboratorio = "Lab. de Redes" pasado al contexto Laboratorio Alfabético indica que el objeto "Lab. de Redes" será accedido directamente a

partir del Menú Principal. A partir de este laboratorio, los otros laboratorios pertenecientes al contexto podrían ser accedidos.

El parámetro laboratorio = "ES:Lab de ES" pasado al grupo de contexto Laboratorio por área de Investigación indica que la distancia "Lab. de ES" en el contexto Laboratorio por área = "ES" será accedido directamente a partir del Menú Principal. A partir de esta instancia, las demás instancias (los otros laboratorios) que pertenecen al contexto Laboratorio por área = "ES" podrían ser accedidos.

El parámetro área = "Redes" pasado al grupo de contexto Laboratorio por área de Investigación indica que el índice del contexto Laboratorio por área = "Redes", indicado por el pequeño rectángulo, puede ser accedido directamente del Menú Principal.

6) Contexto dinámico

Un contexto puede ser dinámico o no. Un contexto de navegación dinámico es aquel cuyos elementos son definidos o alterados en tiempo de ejecución.

La [Figura II.82](#) muestra el ejemplo de un contexto de navegación dinámico, el contexto Disciplina por Requisito, donde un conjunto de disciplinas son seleccionadas por un alumno para su posterior matrícula.

Puede ser observado la [Figura II.82](#) que la navegación entre el contexto Disciplina Alfabético y el contexto Disciplina por Requisito es representada por una línea con flecha bidireccional, significando que es permitido el retorno de la instancia de la cual la navegación fue iniciada. La flecha doble indica el retorno a la instancia de la cual la navegación fue iniciada.

Por ejemplo, el alumno recorre las disciplinas en el contexto Disciplina Alfabético, escoge una de las disciplina para su matrícula, navegando aun el contexto Disciplina por Requisito, que es compuesto por las disciplinas que el tiene seleccionado aun al momento. Posteriormente, el podría continuar analizando las disciplinas retornando el contexto Disciplina Alfabético

exactamente en lo que estaba, o sea, la última disciplina que el alumno visitó y escogió.

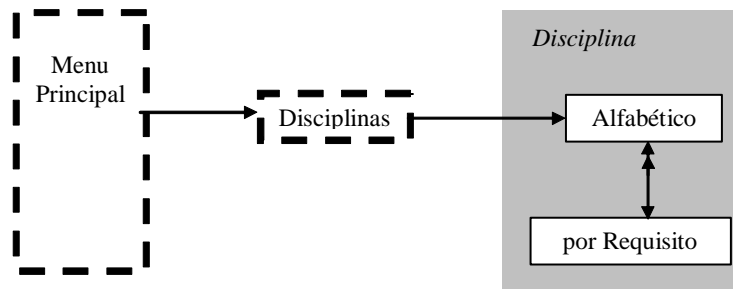


Figura II.82 Contexto de navegación dinámico

Una clase navegacional también puede presentar un contexto de crear, modificar o exclusión de sus instancias. En este caso, todos los otros contextos cuya definición es hecha a través de la especificación de una propiedad de los elementos de esa clase también son dinámicos.

Por ejemplo, si una aplicación hipermedia permite, durante su ejecución, la inclusión de un profesor en una determinada área, entonces esta presenta el contexto dinámico Crear (Figura II.83). Por la falta de nuevos elementos pueden ser creados durante la ejecución de la aplicación. Todos los otros contextos especificados por propiedades o relaciones de la clase Profesor también son considerados dinámicos, por ejemplo, el contexto Profesor por área Igual a Redes de Computadores es dinámico.

La creación, modificación o exclusión de un objeto es representada por un rectángulo con bordes redondeados y la barra vertical en el lado derecho.

En muchos casos, es interesante permitir que, en el caso que un objeto pertenezca a más de un contexto de navegación, sea posible cambiarse la navegación para que esta ocurra dentro de otro contexto a la cual el objeto pertenece. La línea troceada separando los contextos representa que no es posible navegar a partir de un elemento de un contexto hacia el otro contexto.

Cuando la línea no es presentada, es posible navegar a partir de un contexto para cualquier otro contexto.

La elipse asociada al contexto Crear indica que el acceso a ese contexto es protegido, o sea, solamente usuarios con permiso podrán acceder. En la carta de especificación del contexto son indicados los grupos de usuario que tendrán acceso al contexto así como los permisos de cada grupo.

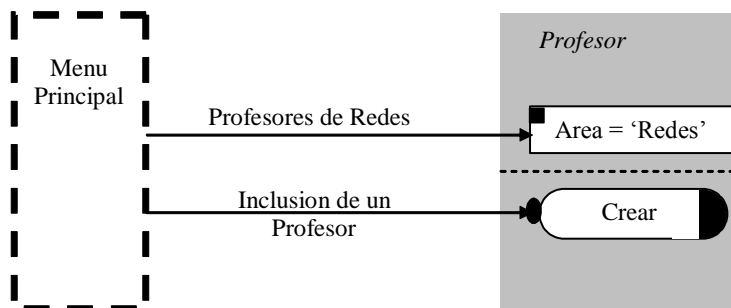


Figura II.83 Contexto de crear

Cuando un contexto permite que el usuario incluya o excluya un objeto (contexto dinámico), las operaciones de inclusión y exclusión son especificadas en la carta de contexto. Esa especificación es importante principalmente cuando esas operaciones influyen o son influenciadas por los otros objetos del contexto.

Por ejemplo, la operación de inclusión de una disciplina en una matricula es especificada en la carta de contexto de configuración de un matricula y debe considerar que dos disciplinas que presentan el mismo horario no pueden ser incluidas en una misma matricula.

7) Contexto Persistente y No Persistente

Los contextos pueden ser persistentes o no. Un contexto no persistente existe solamente durante la sesión de navegación donde este fue creado. Un contexto

persistente, después de ser creado, puede ser accedido en sesiones de navegación posteriores.

No existe representación explícita para un contexto persistente y un contexto no persistente. Pero, para un contexto no persistente será transformado en un contexto persistente la operación de salvar debe estar disponible en el contexto, siendo especificada en la respectiva carta.

8) Generalización de Contexto

Cuando el esquema navegacional presenta la generalización de clases navegacionales, el esquema de contextos también puede presentar la generalización de contextos.

La generalización de contextos es utilizada para representar que todas las subclases de una clase navegacional presentan uno o más contextos en común.

Estos contextos son incluidos como contextos de superclase y heredados por todas las subclases.

La ~~Figura II.84~~~~Figura II.84~~ presenta un ejemplo de generalización de contexto considerando parte del esquema navegacional presentado en la ~~Figura II.73~~~~Figura II.73~~. Las subclases son incluidas dentro del rectángulo representante de la superclase, que es representada por una tonalidad de gris diferente.

Con formato: Fuente: 12 pto

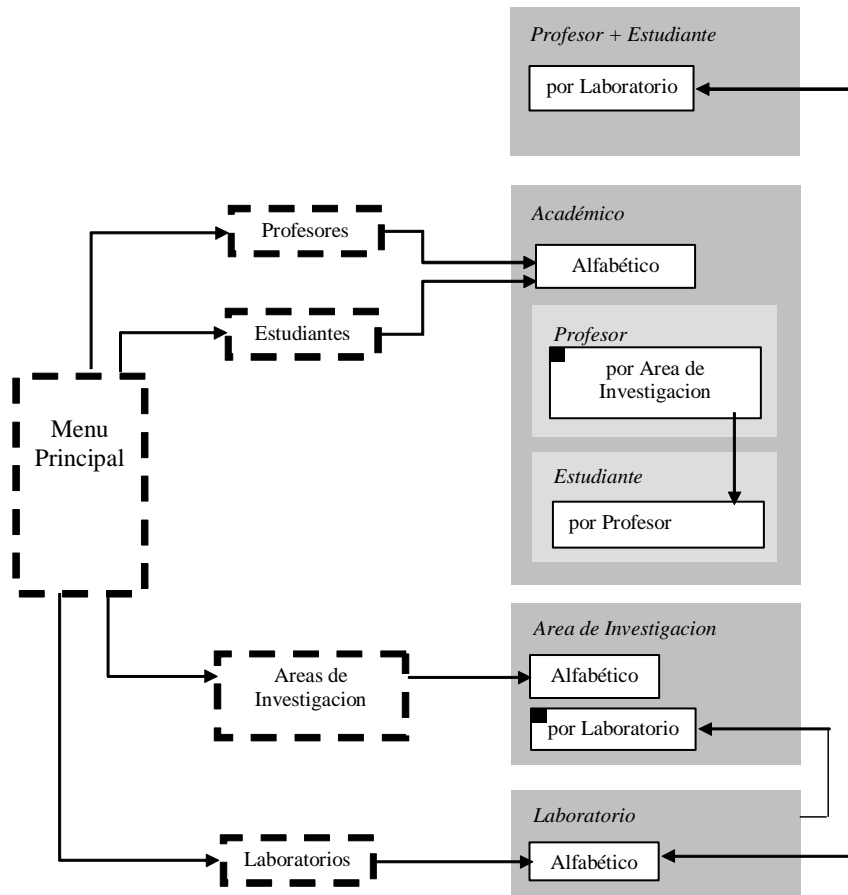


Figura II.84 Generalización de contextos

En este ejemplo, el contexto Alfabético fue generalizado porque cada subclase de Académico (Profesor y Estudiante) presenta el contexto Alfabético.

Si los elementos de dos subclases son presentados dentro de un mismo contexto, este contexto necesita ser especificado dentro de otra clase navegacional que representa la unión de dos subclases.

En este ejemplo, como el contexto Profesor + Estudiante por Laboratorio posee elementos de la clase Profesor y de la clase Estudiante, este es representado

como un contexto de la clase Profesor + Estudiante. El criterio de ordenación de los elementos del contexto es especificado en la carta del respectivo contexto.

9) Navegación en los Contextos

La navegación dentro de un contexto puede ser realizada de diversas formas:

Navegación Secuencial

Los elementos del contexto son accedidos en orden secuencial es pre-establecida. En la navegación secuencial el contexto define el primer elemento, el último elemento, y el próximo y el anterior de cada elemento.

Navegación Circular

Los elementos del contexto también son accedidos en orden secuencial y pre-establecida, pero ese orden es circular. Esta navegación difiere de la navegación secuencial por no presentar el primero y el último elemento.

Navegación Libre

Los elementos del contexto no necesitan ser accedidos en el orden secuencial, así eso también puede ocurrir pues pueden ser definidos los elementos primero, ultimo, próximo y anterior. En esta navegación un elemento puede ser accedido a partir de cualquier otro.

Navegación por Índice

Los elementos del contexto son accedidos solamente a partir del índice, o sea, después de acceder un elemento a través del índice, obligatoriamente retorna el índice, y a partir de este es accedido al próximo elemento deseado.

Combinación de navegación por Índice con Secuencial o Circular

Los elementos del contexto pueden ser accedidos tanto por el índice como por el elemento anterior y próximo.

10) Carta de un Contexto

Para cada contexto o grupo de contexto de navegación es necesario especificar una carta con las propiedades. ([Tabla II.6](#)~~Tabla II.6~~).

Tabla II.6 Carta de contexto o grupo de contexto

Contexto:
Parámetros:
Elementos:
Clases en Contexto:
Ordenación:
Navegación Interna:
Operaciones:
Usuario: Permiso:
Comentarios:

Los nombres de los campos de la carta de contexto son bastante significativos.

El campo Elementos identifica, de manera más formal, todos los elementos que fueron parte del contexto. Cuando el contexto es enumerado, los elementos deben ser especificados explícitamente.

El campo parámetros presenta los parámetros que son pasados durante la definición del contexto. El parámetro puede ser una instancia de clase o el valor de un atributo. Ese campo es utilizado solamente por los grupos de contexto. En la especificación de un contexto simple, ese campo permanece vacío.

El campo Clases en Contexto presenta la lista de las clases en contexto definidas para el contexto en cuestión. Esa lista incluye una clase en contexto para cada clase participante del contexto, si es necesario.

El campo navegación Interna define cual es el tipo de navegación permitida entre los elementos del contexto (secuencial, circular, por índice o una

combinación de estas). En el caso de un contexto que presenta la navegación secuencial o circular deber ser especificada el criterio de ordenación de los elementos.

Un mismo contexto puede tener diversos criterios de ordenación (ordenación múltiple), siendo que todos los criterios deben ser especificados y el criterio default debe ser indicado por el carácter “ + “. En determinadas aplicaciones, también puede ser necesario especificar, para algunos contextos que presentan navegación únicamente secuencial y/o circular, los puntos de entrada obligatorios del contexto, o sea, los elementos por los cuales el contexto siempre es accedido (ej. primer elemento, ultimo elemento). Todos los contextos que contienen elementos de una clase que, en el esquema navegacional, es accedida a través de un enlace que participa de un papel con la propiedad {ordenado} (sección 2.3.2.1.3.- Enlaces), deben presentar el mismo criterio de ordenación.

El campo Restricciones de Uso permite indicar los permisos de acceso para cada clase de usuarios del contexto. Todos los usuarios de un contexto, de una misma clase, poseen los mismos permisos de acceso. Los permisos pueden ser de lectura y/o escritura.

El campo operaciones presenta las operaciones de contexto, o sea, las operaciones que manipulan los elementos del contexto.

La [Tabla II.7](#) especifica la carta de contexto para el contexto Profesor por Título Igual a Doctor, presentado en la [Figura II.74](#).

Tabla II.7 Carta de contexto

Contexto: Profesor Titulo = 'Doctor'
Parámetros:
Elementos: p: Profesor where p.titulo = 'Doctor'
Clase en Contexto:
Ordenación: por p.nombre, ascendente
Navegación Interna: secuencial
Operaciones:
Usuario: alumno Permiso: lectura
Comentarios:

La [Tabla II.8](#) ~~Tabla II.8~~ especifica la carta de contexto para el grupo de contexto Profesor por área presentado en la [Figura II.80](#) ~~Figura II.80~~. Como este contexto presenta la navegación por índice, a partir de un elemento del contexto es necesario retornar el índice para acceder otro elemento del contexto.

Tabla II.8 Carta de contexto o grupo de contexto

Contexto: Profesor por área
Parámetros: a: área
Elementos: p: Profesor where p.asociado_a a
Clase en Contexto:
Ordenación: por p.nombre, ascendente
Navegación Interna: por índice (Idx Profesores por área(a))
Operaciones:
Usuario: alumnos Permiso: lectura
Comentarios:

2.3.2.2.2.- Estructuras de Acceso

Las estructuras de acceso son índices que permiten el acceso a los contextos.

Un ejemplo de estructura de acceso es el índice de acceso a los profesores ordenados por el nombre, o sea, los elementos del contexto Profesor Alfabético.

Las estructuras de acceso son representadas, como se ha dicho anteriormente, por rectángulos con líneas gruesas troceadas. La [Figura II.85](#) muestra la representación de la estructura de acceso Profesores que permite el acceso a todos los profesores.

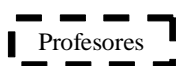


Figura II.85 Estructura de acceso Profesores

Las estructuras de acceso pueden poseer varios criterios de ordenación, pudiendo alternar entre esos criterios. Cuando varios criterios de ordenación son definidos, estos son especificados entre corchetes y separados por “;”. El criterio default es indicado por la señal “+” en la carta de estructura de acceso, como será visto mas adelante. La [Figura II.86](#) muestra la estructura de acceso Profesores que presenta múltiples criterios de ordenación y es representada por el incremento de un triángulo invertido y con relleno en negro en el lado izquierdo.



Figura II.86 Estructura de acceso con múltiples criterios de ordenación

La [Figura II.87](#) y [Figura II.88](#) muestran ejemplos de estructuras de acceso con múltiples criterios de ordenación.

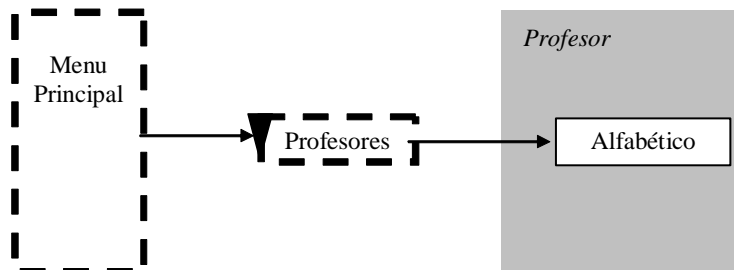


Figura II.87 Índice con un contexto como destino

La [Figura II.87](#) ~~Figura II.87~~, independientemente del criterio de ordenación presentado en la estructura de acceso, el acceso a los elementos de contexto es hecha siempre de acuerdo con el criterio de ordenación especificado en el contexto.

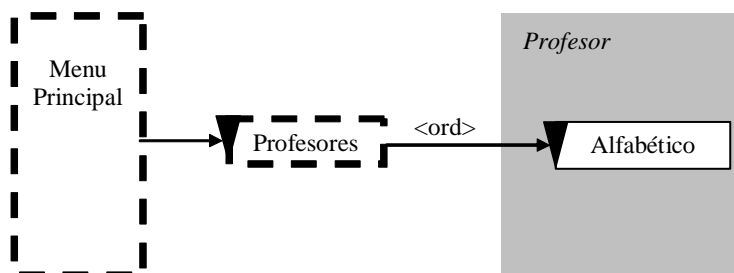


Figura II.88 Índice con un contexto como destino

La [Figura II.88](#) ~~Figura II.88~~, <ord> indica que cuando el contexto Profesor Alfabético va accedido a través del índice Profesores, este presenta el mismo criterio de ordenación del índice. En este caso, el contexto debe presentar el criterio de ordenación del índice. Si el contexto va accedido por otro camino, entonces sus elementos son presentados de acuerdo con su criterio de ordenación por default.

La [Figura II.89](#) ~~Figura II.89~~ presenta una estructura de acceso que puede tener varios contextos como destino. El contexto accedido depende del atributo

accedido en la estructura de acceso. La especificación de la cual el contexto será accedido es hecha en la carta de estructura de acceso.

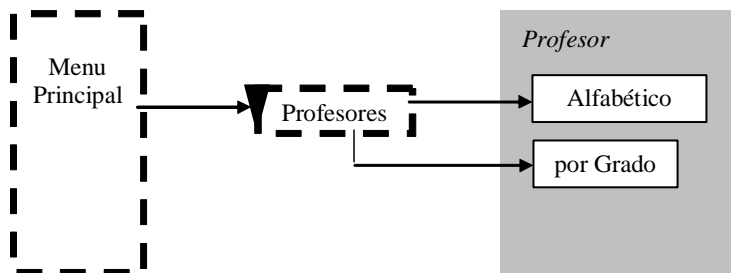


Figura II.89 Índice con varios contextos como destino

Una estructura de acceso también puede ser jerárquica. Una estructura de acceso jerárquica representa un conjunto de índices secuenciales, donde la selección en un nivel determina los elementos del próximo nivel. Esta es usada para simplificar el diagrama de contextos. El rotulo de un índice jerárquico es formado por los atributos que forman la jerarquía, separados por " : ". Los atributos usados en el refinamiento sucesivo de los índices pueden ser los nombres de las clases o los atributos de la clase. La [Figura II.90](#) muestra un índice jerárquico representando 2 estructuras de acceso, donde cada nivel depende de la selección hecha en el nivel anterior.

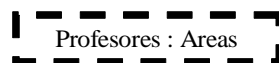


Figura II.90 Estructura de acceso jerárquica

Así como los contextos de navegación, las estructuras de acceso también son especificadas a través de cartas.

La [Tabla II.9](#) presenta la especificación de la estructura de acceso Profesores presentada en la [Figura II.74](#).

Tabla II.9 Carta de estructura de acceso Profesores

Estructura de Acceso: Profesores	
Parámetros:	
Elementos: p: Profesor	
Atributos: p.nombre p.grado	Destino: Ctx Profesor Alfabético (self) -
Ordenación: por p.nombre, ascendente	
Restricciones de Uso	
Usuario: alumnos	Permiso: lectura
Comentarios:	
Depende de: Nav_Profesor	Influencia: ADV Profesores

El campo parámetros especifica los parámetros necesarios para determinar los elementos del índice.

El campo Elementos especifica los elementos que serán mostrados en el índice.

El campo Atributo especifica los atributos de cada objeto que será mostrado en el índice. Los atributos que son usados para acceder el contexto (atributos selectores) deben presentar un destino, o sea, los elementos que serán accedidos a partir del índice. Cuando varios selectores presentan el mismo destino, estos son especificados en una lista donde los selectores son separados por comas.

El campo ordenación especifica el criterio de ordenación de los elementos en el índice. Cuando un contexto es jerárquico, deben ser especificados los criterios de ordenación para los elementos de cada nivel de jerarquía.

El campo Restricciones de Uso especifica los diferentes permisos de acceso del índice.

Los campos Depende de e Influencia presentan los elementos creados durante el modelamiento que al ser modificados influyen en este índice o que son influenciados por una modificación de este índice.

La [Tabla II.10](#)~~Tabla II.40~~ presenta la carta de estructura de acceso Profesores presentada en la [Figura II.89](#)~~Figura II.89~~.

Tabla II.10 Carta de estructura de acceso Profesores

Estructura de Acceso: Profesores	
Parámetros:	
Elementos: p: Profesor	
Atributos:	Destino:
p.nombre	Ctx Profesor Alfabético (self)
p.grado	Ctx Profesor por Grado (self.grado)
Ordenación: por p.nombre, ascendente	
Restricciones de Uso	
Usuario: alumnos	Permiso: lectura
Comentarios:	
Depende de: Nav_Profesor	Influencia: ADV Profesores

La [Tabla II.11](#)~~Tabla II.14~~ presenta la carta de estructura de acceso jerárquica presentada en la [Figura II.90](#)~~Figura II.90~~.

Tabla II.11 Carta da estructura de acceso Profesores : áreas

Estructura de Acceso: Profesores : áreas	
Nivel	
Profesores	Elementos: p: Profesor
	Atributos Destino p.nombre..... ldx áreas de Invest. por Profesor (self)
	Ordenación: por p.nombre, Ascendente
áreas de Invest. por Profesor	Parámetros: p: Profesor
	Elementos: a: área where a <i>pertenece_a</i> p
	Atributos Destino a.nombre.....Ctx áreas de Invest. por Profesor(self)
	Ordenación: por a.nombre, Ascendente
Usuarios: alumnos	Permiso: lectura
Comentarios:	

Depende de:	Influencia en:
-------------	----------------

2.3.2.2.3.- Clases en Contexto

Como los nodos pueden presentar características diferentes de acuerdo como el contexto en el cual estas aparecen, son creadas clases en contexto para definir la apariencia y las anclas de cada nodo en cada contexto a la cual el pertenece, y también otras informaciones importantes al contexto. Una clase en contexto solo es necesaria si el nodo tiene una apariencia diferente y anclas distintas en diferentes contextos.

También son especificados, en las clases en contextos, los atributos con múltiples perspectivas que no presentan perspectiva default, definidos en el esquema conceptual. Cada perspectiva de un atributo que presenta múltiples perspectivas debe ser mapeada para un atributo en una clase en contexto.

Las clases en contexto son representadas por una jerarquía que conecta la clase en contexto a la clase navegacional (enlace) que esta siendo por esta accedida. La clase en contexto es representada por un rectángulo dividido en cuatro partes. La primera parte contiene su identificador. La segunda parte contiene los atributos que están siendo adicionados a la clase. La tercera parte contiene las operaciones que también están siendo adicionados a la clase. La cuarta parte especifica los contextos en los cuales la clase en contexto participa.

La [Figura II.91](#) muestra la representación grafica de la clase en contexto Profesor por área. Como esta clase en contexto no presenta ninguna operación, el comportamiento que contienen las operaciones esta vacío.

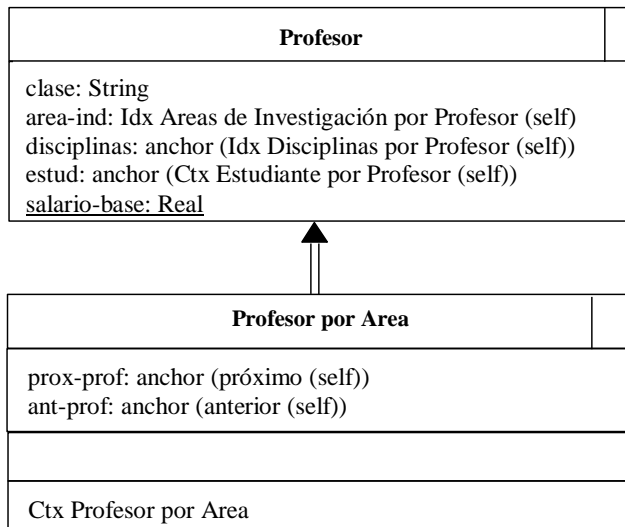


Figura II.91 Clase en contexto Profesor por área

Después de la definición del esquema de contextos navegacionales es necesario definir también los diagramas de las clases en contexto especificadas.

2.3.2.2.4.- Esquema de Contextos de navegación

La [Figura II.92](#) ~~Figura II.92~~ presenta parte de un esquema de contextos de navegación correspondiente al esquema de navegación presentado en la [Figura II.73](#) ~~Figura II.73~~. En el son presentados algunos contextos de navegación y estructuras de acceso.

Con formato: Fuente: 12 pto

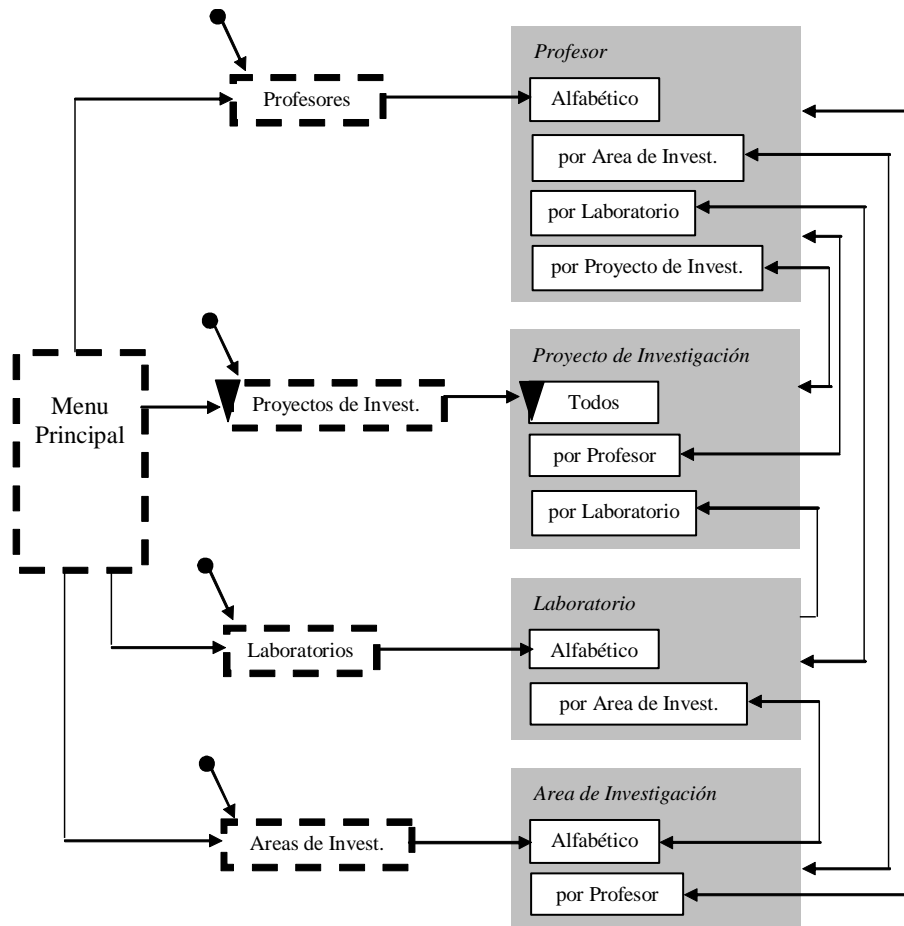


Figura II.92 Esquema de contextos de navegación

Obsérvese que, en este ejemplo, son identificados los landmarks, los puntos ha acceder a partir de cualquier objeto: los índices de Profesores, Proyectos de Investigación, Laboratorios y áreas de Investigación.

2.3.2.3.- Tablas de Mapeamiento de los Objetos Navegacionales

Las tablas de mapeamiento presentan el mapeamiento de los objetos navegacionales, más específicamente nodos, enlaces, contextos, clases en contexto y estructuras de acceso, en objetos conceptuales, de interfaz y de implementación. Después de terminar el modelamiento navegacional es construida una tabla con el mapeamiento entre los objetos navegacionales y conceptuales, y después de terminar cada una de las otras actividades (proyecto de interfaz abstracta e implementación) es construida otra tabla de mapeamiento.

Así, como acontece en el modelamiento conceptual, es necesario especificar solamente los objetos navegacionales que no son mapeados directamente para los otros tipos de objetos. Ese mapeamiento también es especificado a través de una extensión del lenguaje de consulta para bases de datos¹⁷.

2.3.2.3.1.- Tabla de Mapeamiento de los Objetos Navegacionales para Objetos Conceptuales

Esta tabla es construida después de concluir la actividad del modelamiento navegacional y presenta el mapeamiento de los objetos navegacionales para los objetos conceptuales.

Con formato: Estilo Epígrafe + Justificado, Izquierda

¹⁷ KIM, Won. Introducción a bases de datos orientados a objetos. Ed. MIT Press, 1990. 234 p.

| Tabla II.12

~~Tabla II.12~~ presenta el mapeamiento del esquema navegacional presentado en la ~~Figura II.73~~ ~~Figura II.73~~ para el esquema conceptual presentado en la ~~Figura II.54~~ ~~Figura II.54~~.

Con formato: Fuente: 12 pto

Con formato: Fuente: 12 pto

Solamente los objetos que no presentan un mapeamiento directo son mostrados en la tabla. Los tipos de mapeamiento son semejantes a los presentados en la ~~Tabla II.2~~ ~~Tabla II.2~~.

Tabla II.12 Mapeamiento de los Objetos Navegacionales para Objetos Conceptuales

Objetos Mapeados	Regla de Mapeamiento
navegacional: ProyectoDeInvestigación.nombrePatrocinador conceptual: Patrocinador.nombre	seleccionar Pa.nombre DE Pa: Patrocinador, Pr: Proyecto de Investig. DONDE Pa financia Pr
navegacional: DisciplinaEstudiante conceptual: Repetición	mapeamiento directo
navegacional: Estudiante repite DisciplinaEstudiante conceptual: Repetición	mapeamiento directo
navegacional: Disciplina posee DisciplinaEstudiante conceptual: Repetición	mapeamiento directo
navegacional: DisciplinaEstudiante.evaluación conceptual: Repetición.evaluación	mapeamiento directo
navegacional: Estudiante. título conceptual: Orientación.titulo	seleccionar O. título DE O: Orientación, P: Profesor, E: Estudiante DONDE P orientación E
navegacional: Estudiante.evaluación conceptual: Orientación.evaluación	seleccionar O.evaluación DE O: Orientación, P: Profesor, E: Estudiante DONDE P orientación E
navegacional: Estudiante.nombre conceptual: Estudiante.nombre	mapeamiento directo
navegacional: DisciplinaEstudiante.nombreEstudiante conceptual: Estudiante.nombre	seleccionar E.nombre DE E: Estudiante, D: Disciplina DONDE E repite D
navegacional: Disciplina.nombre conceptual: Disciplina.nombre	mapeamiento directo
navegacional: DisciplinaEstudiante.nombreDisciplina conceptual: Disciplina.nombre	seleccionar D.nombre DE D: Disciplina, E: Estudiante DONDE E Repetición D
navegacional: Persona.descripcion conceptual: Persona.descripcion	mapeamiento directo
navegacional: Persona.bio conceptual: Persona.descripcion	mapeamiento directo
navegacional: ProyectoDeInvestigaciónPorLaboratorio.marketing conceptual: ProyectoDeInvestigación.descripcion	mapeamiento directo
navegacional: ProyectoDeInvestigaciónPorProfessor.Investigación conceptual: ProyectoDeInvestigación.descripcion	mapeamiento directo

2.3.2.3.2.- Tabla de Mapeamiento de los Objetos Navegacionales para Objetos de Interfaz

Esta tabla es construida después de concluir la actividad del proyecto de interfaz abstracta y presenta el mapeamiento de los objetos navegacionales para los objetos de interfaz ([Tabla II.13](#)~~Tabla II.13~~).

Tabla II.13 Mapeamiento de los Objetos Navegacionales para Objetos de Interfaz

Objetos Mapeados	Regla de Mapeamiento
navegacional:	
interfaz:	

2.3.2.3.3.- Tabla de Mapeamiento de los Objetos Navegacionales para Objetos de implementación

Esta tabla es construida después de concluir la actividad de implementación y presenta el mapeamiento de los objetos navegacionales para los objetos de implementación ([Tabla II.14](#)~~Tabla II.14~~).

Tabla II.14 Mapeamiento de los Objetos Navegacionales para Objetos de Implementación

Objetos Mapeados	Regla de Mapeamiento
navegacional:	
implementación:	

2.3.3.- DISEÑO DE INTERFAZ ABSTRACTA

El proyecto de interfaz abstracta define como serán los objetos de interfaz (objetos navegacionales y otros auxiliares, Ej. Barras de menús, botones de control), sus propiedades y transformaciones, además de promover la independencia de dialogo y el reuso de esos objetos.

El proyecto de interfaz es desarrollado antes de iniciar la implementación y de forma independiente del ambiente de implementación, entretanto, debe también considerarse algunas características del ambiente para que pueda ser implementado.

OOHDM utiliza Vista de Datos Abstracta (ADV, Abstract Data View) para especificar el modelo de interfaz abstracta. Complementando el proyecto de interfaz también son usados diagramas de configuración y ADVcharts.

2.3.3.1.- Vista de Datos Abstracta (ADV)

Los ADVs son objetos de interfaz usados para especificar la apariencia e interfaz de los objetos de la aplicación.

Existen otros objetos que modelan la interfaz pero que no soportan eventos externos, solamente interactúan con las estructuras de datos y de control de la aplicación. Esos objetos son llamados Objetos de Datos Abstracto (ADOs, Abstract Data Objects).

Los ADOs están asociados a los ADVs que transmiten la información de sus estados hacia afuera de la aplicación y para otros ADVs. Por ejemplo, existe un ADO Profesor que representa un objeto Profesor con su información, y ese ADO esta asociado a un ADV, el ADV Profesor, que representa su interfaz con el usuario.

Aquí, los ADVs son usados para representar los objetos de interfaz a través de la identificación abstracta de un conjunto de atributos que define el conjunto de eventos con los cuales el objeto de interfaz podrá tratar, como los eventos generados por el usuario. Cuando algunas características de los objetos no puedan ser especificadas a través de la representación de los ADVs, deben ser especificadas informalmente.

Los objetos navegacionales (nodos, enlaces, clases en contexto y estructuras de datos) actuarían como ADOs y estarán asociados los ADVs. Así, para cada objeto navegacional es necesario definir un ADV. Los ADVs relacionados con los nodos deben contener otros ADVs representando sus atributos. También son definidos ADVs para los contextos de navegación y para las clases en contexto.

La [Figura II.93](#) muestra un ejemplo de ADV, el ADV Persona.

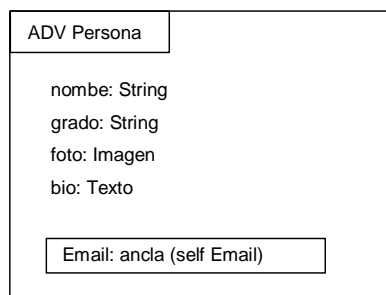


Figura II.93 ADV Persona

No es necesario especificar la interfaz de los enlaces 1-para-1, solamente las anclas. Para los enlaces 1-para-N puede ser importante especificar aspectos de interfaz, como por ejemplo, que la activación de los N objetos destino puede ser hecha a través de un menú.

Para mostrar los objetos que son perceptivos en un momento dado es usada la variable "ContextoPerceptivo". Los objetos que se tornan perceptivos, o sea, son

abiertos, son adicionados al “ContextoPerceptivo”, en cuanto a los que dejan de ser perceptivos, o sea, son cerrados, son excluidos de este.

2.3.3.2.- Diagramas de Configuración

Los diagramas de configuración son usados para representar los relacionamientos entre los objetos de interfaz y los objetos navegacionales.

Además de incluir los objetos de interfaz que son representados por los ADVs, estos diagramas especifican otras características de la interfaz abstracta de la aplicación, que son:

- Los eventos externos iniciados por el usuario y que serán manipulados por los ADVs.
- Los relacionamientos estructurales entre los ADVs. Por ejemplo, el enlace activado por determinada ancla es representado por una vinculación entre la ancla y el objeto destino; o el anidamiento entre ADVs compuestos.
- Los relacionamientos entre ADVs (que representan los objetos de interfaz) y los ADOs (que representan los objetos navegacionales).

La [Figura II.94](#) ~~Figura II.94~~ presenta un diagrama de configuración que muestra el relacionamiento entre el ADV Laboratorio y su ADO, el nodo Laboratorio. El ADV Laboratorio puede reaccionar a los eventos externos mouse doble clic y mostrar, enviando mensajes al Nodo Laboratorio. El mensaje buscar nombre retorna el nombre del laboratorio. El mensaje buscar índice retorna una lista de elementos que podrán ser seleccionados para acceder a cada objeto. El mensaje anclaSelec (Profesores) solicita al ancla Profesores que esta inicie la navegación a través del enlace asociado a esta.

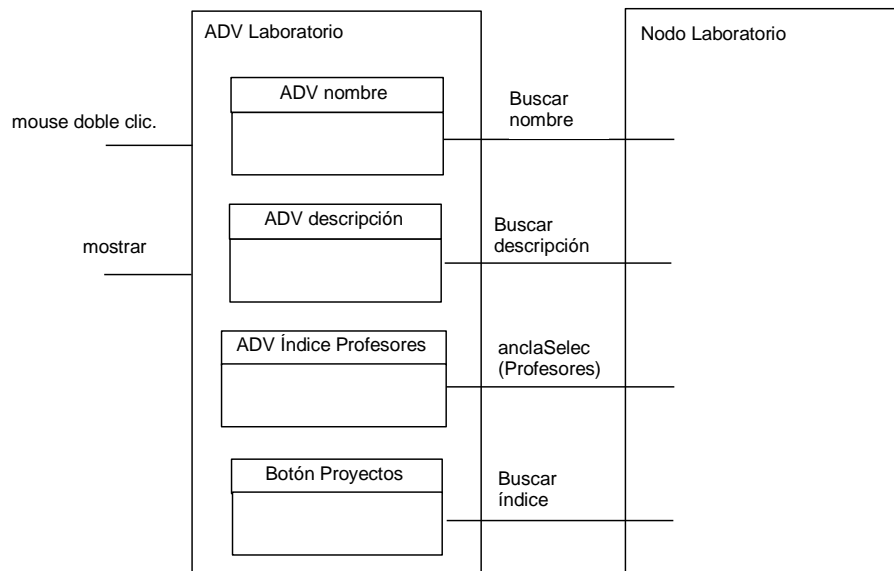


Figura II.94 Diagrama de Configuración

2.3.3.3.- ADVcharts

Los ADVcharts presentan las transformaciones ocurridas en los ADVs, responsables por cambiar en la interfaz con el usuario y en los objetos navegacionales. Cada ADVchart esta relacionado con una pantalla de la interfaz, así toda la interfaz de la aplicación es mostrada a través de ADVcharts.

Los ADVcharts pueden contener otros ADVs, estados, atributos y transiciones.

Los estados de los ADVs son conectados por transiciones y modificados por eventos externos (generados por los usuarios) o internos (generados por las transiciones).

La diferencia entre los ADVs y los estados es que los ADVs representan la estructura y comportamiento de los objetos, en cuanto a los estados representan solamente el comportamiento. Los ADVs y los estados pueden estar anidados en modo AND o XOR.

Las transiciones también presentan un evento (responsable por la activación de la transición), pre-condiciones (condiciones que deben ser satisfechas para que la transición sea activada) y pos-condiciones (acciones que serán ejecutadas con la activación de la transición).

Los ADVs son representados por rectángulos, los estados por rectángulos con bordes redondeados y las transiciones por líneas direccionadas.

La [Figura II.95](#) muestra el ADVchart Laboratorio. De acuerdo con este ADVchart, siempre que un nodo es abierto, este envía el mensaje "mostrar" al ADV asociado.

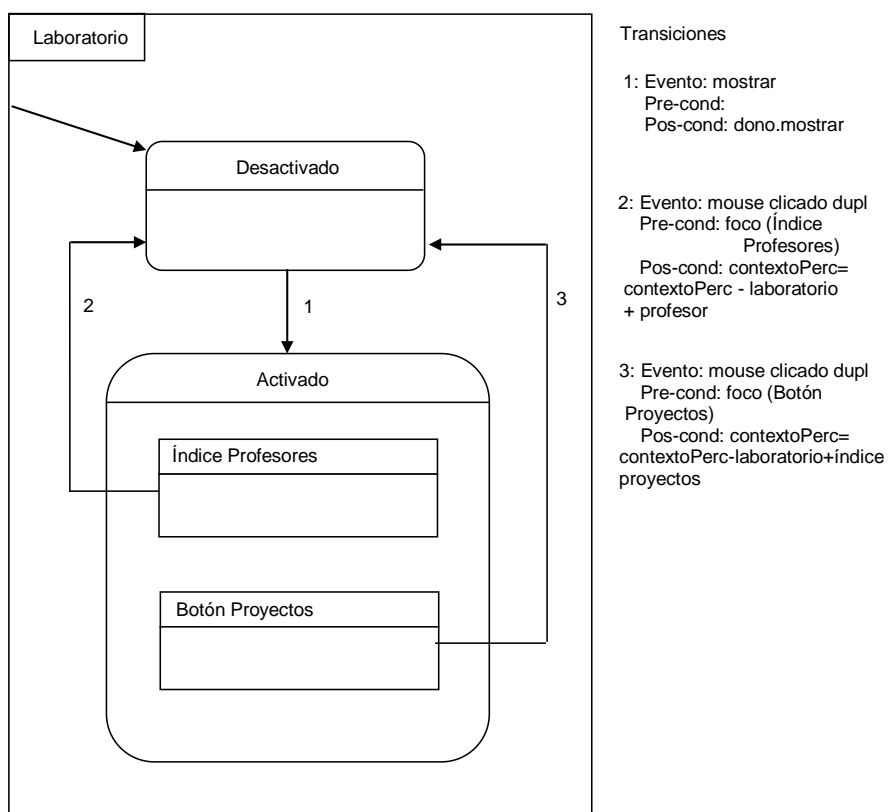


Figura II.95 ADVchart Laboratorio

Los ADVs también permiten representar la sincronización entre diferentes objetos usando el símbolo de sincronización de redes de petri para juntar las transiciones que serán sincronizadas.

2.3.3.4.- Tablas de Mapeamiento de los Objetos de Interfaz

Las tablas de mapeamiento presentan el mapeamiento de los objetos de interfaz en objetos conceptuales, navegacionales y de implementación. Después de terminar el proyecto de interfaz son construidas las tablas con el mapeamiento entre los objetos de interfaz y los objetos conceptuales y navegacionales, y después al terminar la actividad de implementación es construida la tabla con el mapeamiento entre los objetos de interfaz y los objetos de implementación.

Así, como en el modelamiento conceptual y navegacional, es necesario especificar solamente los objetos de interfaz que no son mapeados directamente para los otros tipos de objetos. Ese mapeamiento también es especificado a través de una extensión del lenguaje de consulta para bases de datos¹⁸.

2.3.3.4.1.- Tabla de Mapeamiento de los Objetos de Interfaz para Objetos Conceptuales

Esta tabla es construida después de concluir la actividad del proyecto de interfaz y presenta el mapeamiento de los objetos de interfaz para objetos conceptuales. (Tabla II.15Tabla II.15).

Tabla II.15 Mapeamiento de los Objetos de Interfaz para Objetos Conceptuales

Objetos Mapeados	Regla de Mapeamiento
interfaz:	
conceptual:	

¹⁸ KIM, Won. Introducción a bases de datos orientados a objetos. Ed. MIT Press, 1990. 234 p.

2.3.3.4.2.- Tabla de Mapeamiento de los Objetos de Interfaz para Objetos Navegacionales

Esta tabla también es construida después de concluir la actividad del proyecto de interfaz y presenta el mapeamiento de los objetos de interfaz para los objetos navegacionales ([Tabla II.16](#)~~Tabla II.16~~)

Tabla II.16 Mapeamiento de los Objetos de Interfaz para Objetos Navegacionales

Objetos Mapeados	Regla de Mapeamiento
interfaz:	
navegacional:	

2.3.3.4.3.- Tabla de Mapeamiento de los Objetos de Interfaz para Objetos de Implementación

Esta tabla es construida después de concluir la actividad de implementación y presenta el mapeamiento de los objetos de interfaz para los objetos de implementación ([Tabla II.17](#)~~Tabla II.17~~)

Tabla II.17 Mapeamiento de los Objetos de Interfaz para Objetos de Implementación

Objetos Mapeados	Regla de Mapeamiento
interfaz:	
implementación:	

2.4.- IMPLEMENTACIÓN

La etapa de implementación es la última etapa de OOHDM y es responsable por la traducción del proyecto navegacional y del proyecto de interfaz para un ambiente de implementación.

En esta etapa debe ser definido el ambiente de implementación que será utilizado y como es hecho el mapeamiento de los objetos navegacionales y de interfaz para ese ambiente.

Si la aplicación por implementar en la Web, podría utilizar el ambiente OOHDM-Web. El OOHDM-Web es un ambiente de desarrollo que permite la automatización de creación de los nodos, contextos y estructuras de acceso definidos en el modelamiento utilizando OOHDM.

Por tanto, los objetos navegacionales y sus relacionamientos resultantes del proyecto navegacional deben ser mapeados para tablas Access. A continuación son presentadas algunas reglas para el mapeamiento de las clases navegacionales para el modelo relacional.

2.4.1.- MAPEAMIENTO DE LOS OBJETOS

Las clases conceptuales presentadas en el esquema conceptual son mapeadas para tablas (Ej. relaciones) del modelo relacional. Cada clase es mapeada para una tabla diferente. Entretanto, si una clase presenta una cantidad variable de elementos de un tipo de atributo, también son creadas tablas auxiliares.

El mapeamiento de las clases para tablas sigue una regla más simple: para cada atributo del objeto, un campo es definido en la tabla. Esa regla no se aplica a los atributos multivalorados, para los cuales son definidos nuevas tablas que referencien la tabla de la clase conceptual.

2.4.2.- MAPEAMIENTO DE LOS RELACIONAMIENTOS

Los relacionamientos son mapeados con las siguientes reglas:

- Relacionamientos 1:N son mapeados para una clave foránea (FK) en la tabla que representa a la clase del lado N.
- Relacionamientos N:N son mapeados para una nueva tabla cuyos campos son claves de las tablas que representan las clases relacionadas.

2.4.3.- TABLAS DE MAPEAMIENTO DE LOS OBJETOS DE IMPLEMENTACIÓN

Las tablas de mapeamiento presentan el mapeamiento de los objetos de implementación en objetos conceptuales, navegacionales y de interfaz. Después de concluir la implementación son construidas las tablas con el mapeamiento entre los objetos de implementación y los otros objetos.

Así como en las otras actividades, es necesario especificar solamente los objetos de implementación que no son mapeados directamente para los otros tipos de objetos. Ese mapeamiento también es especificado a través de una extensión del lenguaje de consulta para bases de datos¹⁹.

2.4.3.1.- Tabla de Mapeamiento de los Objetos de implementación para Objetos Conceptuales

Esta tabla es construida después de concluir la actividad de implementación y presenta el mapeamiento de los objetos de implementación para los objetos conceptuales ([Tabla II.18](#)~~Tabla II.18~~).

¹⁹ KIM, Won. Introducción a bases de datos orientados a objetos. Ed. MIT Press, 1990. 234 p.

Tabla II.18 Mapeamiento de los Objetos de implementación para Objetos Conceptuales

Objetos Mapeados	Regla de Mapeamiento
implementación:	
conceptual:	

2.4.3.2.- Tabla de Mapeamiento de los Objetos de implementación para Objetos Navegacionales

Esta tabla también es construida después de concluir la actividad de implementación y presenta el mapeamiento de los objetos de implementación para los objetos navegacionales ([Tabla II.19](#)~~Tabla II.19~~).

Tabla II.19 Mapeamiento de los Objetos de implementación para Objetos Navegacionales

Objetos Mapeados	Regla de Mapeamiento
implementación:	
navegacional:	

2.4.3.3.- Tabla de Mapeamiento de los Objetos de Implementación para Objetos de Interfaz

Esta tabla también es construida después de concluir la actividad de implementación y presenta el mapeamiento de los objetos de implementación para los objetos de interfaz ([Tabla II.20](#)~~Tabla II.20~~).

Tabla II.20 Mapeamiento de los Objetos de Implementación para Objetos de Interfaz

Objetos Mapeados	Regla de Mapeamiento
implementación:	
interfaz:	

CAPÍTULO III

III.- ANÁLISIS Y DISEÑO DE LA ARQUITECTURA DE LA INFORMACIÓN DE LA ESPE-L

3.1.- ESTUDIO DE LA ARQUITECTURA DE LA INFORMACIÓN DE LA ESPE-L

La arquitectura de la información que se presenta ha continuación será analizada, estructurada y se la aplicara el diseño conceptual, diseño de navegación y diseño de interfaz de usuario correspondiente que luego será aplicada en la fase de implementación del portal con fundamentos de diseño de interfaces.

La información es necesaria que sea estudiada por separada de acuerdo al rol que desempeña cada usuario dentro del sistema institucional y académico.

3.1.1.- INFORMACIÓN INSTITUCIONAL

Esta información comprende la siguiente estructura:

3.1.1.1.- Campus

Dentro de esta sección corresponde ¿Quiénes Somos?, el mensaje del director, la filosofía como es la: misión, visión, principios filosóficos, los valores institucionales, la historia de la escuela, las autoridades, los convenios con

instituciones o empresas, la ubicación geográfica, el directorio de contactos en el que conste su correo electrónico y teléfono.

3.1.1.2.- Información Académica

Dentro de esta sección corresponde el proceso de admisión como es: el cronograma de admisión, las inscripciones y matrículas, los programas de financiamiento, los valores de aranceles y becas.

3.1.1.3.- Facultades e Institutos

Dentro de esta sección corresponde la información detallada de cada facultad e instituto como es: una introducción de la facultad o instituto, las especialidades que se ofertan, el pénsum académico, los docentes, la infraestructura, los servicios que prestan.

Las facultades son: Sistemas e Informática, Ciencias Administrativas. Las siguientes carreras: Electromecánica, Automotriz, Electrónica e Instrumentación. Los institutos son: Idiomas y Ciencias Básicas. Los postgrados, dentro de este se tratara una introducción, maestrías.

3.1.1.4.- Servicios

Dentro de esta sección corresponde la información de la biblioteca como es: una introducción, los servicios, la consulta bibliográfica, la biblioteca digital, el personal, las bases de datos. El policlínico, el centro de producción, los clubes, el bar estudiantil.

La ESPE-L, ofrece el servicio de FTP, con software gratuito.

Además, el mapa del sitio en el que se representa la estructura del acceso a la información que contiene el portal.

3.1.2.- SERVICIOS WEB

Para la comunidad de la ESPE-L y público en general se han creado espacios exclusivos donde los alumnos, docentes, empleados y graduados puedan acceder a los servicios electrónicos y diferentes contenidos que por el rol que tienen con la ESPE-L, son de su interés.

El acceso a los siguientes Servicios Web que ofrece el portal requiere de autenticación de usuarios.

3.1.2.1.- Alumnos y Egresados

A través de este espacio, los alumnos y egresados pueden acceder a la información y los servicios más importantes para su vida académica como:

- Modificar sus datos personales.
- Cambiar su clave de acceso.
- Consulta de notas.
- Imprimir las notas.
- Consulta de record académico.
- Imprimir el record académico.

3.1.2.2.- Docentes

Es un espacio diseñado para los docentes de la ESPE-L. A través de él, tienen a su disposición la información y servicios necesarios para su labor diaria:

- Modificar sus datos personales.

- Cambiar su clave de acceso.
- Ingreso de notas por parte de docentes.
- Imprimir las notas.
- Webmail.

3.1.2.3.- Empleados

Es un espacio diseñado para los empleados de la ESPE-L. A través de él, tienen a su disposición la información y servicios necesarios para su labor diaria:

- Modificar sus datos personales.
- Cambiar su clave de acceso.
- Webmail.

3.1.3.- EVENTOS O NOTICIAS

Se publican todos los eventos o noticias académicas, deportivas y culturales que se programan en la ESPE-L y que están a cargo de los grupos de trabajo, facultades e institutos, docentes, dependencias y estudiantes, los cuales son publicados por el administrador en forma cronológica de realización.

3.1.4.- COMENTARIOS O SUGERENCIAS

A través de este espacio los visitantes del portal pueden enviar comentarios o sugerencias, estos pueden ser presentados en el portal el administrador es el encargado de publicarlos.

3.2.- ANÁLISIS

3.2.1.- ESPECIFICACIÓN DE REQUISITOS (ERS)

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

3.2.2.- LEVANTAMIENTO DE REQUISITOS

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

3.3.- DISEÑO

3.3.1.- DISEÑO CONCEPTUAL

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

3.3.2.- DISEÑO DE NAVEGACIÓN

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

3.3.3.- DISEÑO ABSTRACTO DE INTERFAZ DE USUARIO

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

CAPÍTULO IV

IV.- DESARROLLO DEL PORTAL ACADÉMICO DE LA ESPE-L

4.1.- DESARROLLO DEL PORTAL

4.1.1.- CONFIGURACIONES EN LOS SERVIDORES

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

4.1.2.- DESARROLLO Y PROGRAMACIÓN DE WEB ESPEL

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

4.2.- PRUEBAS Y EVALUACIÓN

4.2.1.- DESARROLLO DE UN PLAN DE PRUEBAS

Una vez que el sitio se ha construido, es necesario hacerlo pasar por una serie de pruebas antes de y entrar a la fase de producción. Mediante dichas pruebas, se medirá su reacción integral frente a diversas acciones que realizarán los usuarios desde sus páginas.

Entre otros aspectos será necesario probar el desempeño computacional de la plataforma tecnológica usada; seguridad ante intentos de ataque y exactitud; corrección de su contenido y su despliegue en los diferentes programas visualizadores, entre otros aspectos.

Con el fin de probar las diferentes capacidades de un Sitio Web, es necesario dividir el trabajo en cinco áreas, que son:

- Pruebas de Interfaces y Contenidos
- Pruebas de Funcionalidades y Operación
- Pruebas de Carga
- Pruebas de Seguridad
- Pruebas de Respaldo y Recuperación

Por cada una de ellas hay actividades específicas a realizar, de las cuales se entrega un detalle a continuación.

4.2.1.1.- Pruebas de Interfaces y Contenidos

Las actividades de esta etapa consisten en hacer revisiones precisas de la forma en que se despliegan las páginas del sitio y ver si cumplen con los

Requerimientos de interfaces y contenidos, además, si cumplen con los estándares mínimos que se hayan definido como meta a ser cumplida.

Las acciones de prueba sugeridas para realizar en esta etapa son las siguientes:

4.2.1.1.1.- Verificación de Contenidos

Es una prueba básica para revisar si el Sitio Web desarrollado incluye todos los contenidos que se han especificado en los requerimientos o los que se hayan definido en el marco del plan de desarrollo. Se puede hacer en forma manual o automática, de acuerdo a las siguientes orientaciones:

1) Sistema Manual

Se refiere a hacer una revisión manual de los contenidos del Sitio Web a través de la navegación de sus páginas. Para ello se recomienda primero construir un índice de contenidos y luego verificar la existencia de cada uno de los ítems que contiene, a través de hacer un recorrido exhaustivo del sitio. Los elementos que deben probarse obligatoriamente son:

- Verificación de ortografía y redacción
- Verificación de enlaces principales
- Verificación de imágenes en páginas
- Verificación de existencia de Archivos adjuntos

2) Sistema Automático

Especialmente orientado a la verificación de enlaces rotos, lo cual se puede hacer utilizando sistemas basados en Internet o, bien, software especializado.

- Sistemas Basados en Internet: Se recomienda usar el servicio del W3C «Check Link» (<http://validator.w3.org/checklink>)
- Software: Se recomienda bajar y usar desde su computador el software gratuito Xenu (<http://home.snafu.de/tilman/xenulink.html>). De igual manera,

los actuales software de creación de sitios WEB permiten manejar en forma controlada los enlaces internos; un error común de este tipo es que una foto se vea normalmente en el computador de desarrollo, pero no en el Sitio Web, Esto ocurre porque es referida en forma absoluta desde una ubicación en un disco duro local o en red, en lugar de un directorio de imágenes del Sitio Web.

3) Sitio en Construcción

Se debe verificar que el Sitio Web no contenga espacios vacíos o que tenga el título de «en construcción». No es adecuado, bajo ningún sentido, usar espacios con dicha leyenda; en tal caso es preferible eliminar esa zona y volver a incluirla cuando exista el contenido correspondiente en el sitio.

4) Verificación de Meta Tags

Los «meta tags» son marcas en lenguaje html que van en la parte superior de cada página, a través de las cuales se entrega a los sistemas de indexación y búsqueda (como Google, Yahoo! y otros), la información mínima que se debe tener en cuenta para hacer una correcta indexación del contenido que incluye.

Los «meta tags» son elementos que obedecen a un estándar definido por el World Wide Web Consortium (<http://www.w3c.org>) por lo que su uso está regulado. Para verificar que dichas marcas cumplen con los elementos mínimos requeridos por los buscadores, existen herramientas en Internet que permiten hacer tal prueba y ofrecen recomendaciones para mejorar la información ingresada en dicha área. Se recomienda en ese sentido utilizar las aplicación existente en el Sitio Web SearchMechanics.com (<http://www.searchmechanics.com/prepare/index.htm>) que cuenta con una aplicación en idioma español para hacer dicha comprobación.

5) Verificación de Estándares

Aunque los sitios web pueden ser construidos a partir de diferentes lenguajes, todos deben cumplir ciertas normas de organización de su código fuente

(sintaxis), que permitan su visualización por software equivalente en diferentes plataformas. Dicha sintaxis está estandarizada y puede ser probada a través de herramientas públicas que están disponibles en Internet. Las dos más importantes son:

- Validación de HTML: La realiza el World Wide Web Consortium (<http://validator.w3.org>) e indica si el código usado en la página es correcto. Como resultado entrega un reporte con los eventuales errores para ayudar a su reparación.
- Validación de CSS: La realiza el World Wide Web Consortium (<http://jigsaw.w3.org/cssvalidator>) e indica si la Hoja de Cascada de Estilo (Cascade Style Sheet) cumple con la sintaxis estándar y por lo tanto podrá ser visualizada correctamente en todos los sistemas.

4.2.1.1.2.- Verificación de Interfaces

Mediante esta prueba se revisan aspectos gráficos del Sitio Web, para determinar si su despliegue en las páginas es correcto. Dentro de los elementos más importantes a ser verificados, se incluyen los siguientes:

1) Plug-ins necesarios

Cuando se utilicen elementos audiovisuales o interactivos que requieran de algún software incrustado para funcionar (plug-ins), se debe ofrecer un enlace para que los usuarios que no lo tengan instalado, puedan bajarlo y hacer el proceso de instalación. En el caso del uso de la tecnología Flash, las últimas actualizaciones del producto permiten que el software pueda ser bajado en forma automática por los programas visualizadores, si se cuenta con la codificación adecuada. Por lo anterior, es necesario hacer la prueba desde un computador que carezca de dicho software, para comprobar que efectivamente hace dicha operación.

2) Consistencia de la Diagramación

Cada una de las páginas del sitio debe tener elementos consistentes, con el fin de ofrecer al usuario una experiencia similar en cualquier área del Sitio Web; por nombrar sólo tres aspectos, lo anterior implica que los menús deben aparecer siempre en el mismo lugar; que los listados deben estar diseñados de similar manera en todo el sitio y que los colores y formas de uso de las interfaces deben ser similares a lo largo de las páginas.

3) Ancho de la Diagramación

Si la diagramación del sitio se ha realizado para un ancho determinado (por ejemplo, 800 píxeles de ancho), en esta etapa se debe probar si ello se cumple, asimismo, se debe probar en una pantalla configurada con una menor dimensión (por ejemplo 640 x 480 píxeles), cuál es el área visible del sitio y cómo afecta eso a la navegación por el mismo.

Otra prueba del mismo estilo, se refiere a usar un programa visualizador orientado sólo a texto como Lynx (<http://www.delorie.com/web/lynxview.html>), para obtener visiones alternativas de la manera en que los usuarios están accediendo a la información que se les ofrece.

En este aspecto, en caso de existir, es de interés contar con un estudio del «log» del servidor que muestre la forma en que los usuarios están accediendo a las páginas, porque de esa manera se podrá determinar hacia qué configuración de pantalla se debe atender con mayor prioridad. La norma en este aspecto es que sin importar las características técnicas que tenga el computador del usuario que accede al Sitio Web, éste siempre se vea ordenado y legible.

4) Diagramación vs. Browsers

Aunque la codificación en los lenguajes soportados por los programas visualizadores (browsers) puede apegarse a los estándares, no todos muestran de la misma manera los sitios web. Dado esto, es necesario revisar el sitio en diferentes tipos de programas, especialmente en aquellos que conforman la

minoría, por ejemplo. Es decir, las pruebas al menos deberían hacerse en Microsoft Internet Explorer (<http://www.microsoft.com/explorer>), Opera (<http://www.opera.com>) y Mozilla (<http://www.mozilla.org>), ya que con ellos se cubrirá un amplio espectro. Lo que se debe revisar en este caso es el despliegue de todos los elementos que se muestran en la pantalla, para asegurar de que aparecen en las posiciones que se les han asignado en el diseño.

5) Diagramación vs. Sistema Operativo

Tal como se explicó en el caso anterior, los diferentes sistemas operativos pueden establecer diferencias en la forma en que se muestran los sitios web. Por ello, es importante conocer cuáles son los sistemas operativos utilizados por la audiencia a la que se desea llegar y revisar el despliegue del sitio en ellos. Hay que recordar que, además de Microsoft Windows, los usuarios pueden estar visualizando el sitio desde computadores equipados con Apple Macintosh o diferentes versiones del sistema operativos Unix.

6) Imágenes Escaladas

Se debe verificar que las imágenes que aparezcan en el sitio no estén siendo mostradas en tamaño reducido artificialmente; es decir, que se tome una imagen de grandes dimensiones y por programación se muestre en un tamaño menor. El efecto de eso es que las páginas con ese tipo de imágenes serán muy pesadas y harán que el acceso a ellas sea lento. Para comprobarlo, se debe solicitar las «Propiedades» de la imagen; en la ventana que se muestra se indica el peso de la imagen, que no debería sobrepasar los 5Kb para las de tamaño pequeño (iconos y thumbnails) y los 25 Kb, para los de tamaño mediano (fotografías en noticias). Es importante considerar que, además de estas verificaciones individuales de peso de imágenes, el límite de peso para una página es de 100Kb, incluyendo todos sus elementos.

7) Imágenes Sin Atributo ALT

Para cumplir con las normas de accesibilidad es necesario que todas las imágenes que se usen en un Sitio Web, tengan una descripción utilizando el

atributo ALT (para texto alternativo) del lenguaje HTML. Para comprobarlo, basta situar el mouse sobre una imagen, para que se despliegue una leyenda en texto en una etiqueta amarilla que flota sobre la foto; si eso no ocurre, el atributo no está siendo usado y debe ser corregido e incluido.

4.2.1.2.- Pruebas de Funcionalidades y Operación

Las actividades de esta etapa se refieren a hacer chequeos completos respecto de las funcionalidades y aplicaciones que ofrece el sitio, ya sean de aplicaciones simples como formularios hasta más complejas, como consultas y modificaciones de registros en base de datos.

En este sentido, las pruebas se deben hacer sobre diferentes elementos, siendo algunos de los más importantes los siguientes:

4.2.1.2.1.- Validación de Formularios

Si el Sitio Web tiene formularios para el envío o ingreso de datos, se debe utilizar sistemas de validación del ingreso de datos para asegurar que éstos sean bien ingresados. En este aspecto, algunas de las validaciones más importantes deben ser las siguientes:

1) Campos Obligatorios

Se debe validar que en los formularios sean ingresados todos aquellos campos que sean necesarios; éstos deben ser marcados de alguna manera (usualmente con un asterisco) que permita a los usuarios entender la obligatoriedad de ingresar información en ellos; adicionalmente, debe indicarse tal condición en forma explícita.

2) Validaciones Locales

Para reducir la carga de validaciones en el servidor, se recomienda incorporar la mayor cantidad de éstas en el computador del cliente, utilizando en forma estándar el lenguaje Javascript para hacerlas.

3) Sintaxis de Ingreso

Se debe validar que, en algunos casos, los campos sean ingresados con datos válidos; el mejor ejemplo es el caso del ingreso del número de la Cédula de Identidad o RUC, cuyos números tienen un algoritmo conocido para ser validado.

4) Suscripción a Servicios

Se debe validar que cada vez que se realice la suscripción a un servicio que ofrezca el Sitio Web, se envíe un e-mail al usuario (para lo cual se debe necesariamente solicitar su dirección de correo electrónico) en el que se le informe sobre el resultado de lo realizado. Quien pruebe el sistema debe validar que el sistema esté enviando correctamente los e-mails y que dicho e-mail llegó a la dirección correspondiente; en este caso se recomienda probar con una dirección de recepción externa a la institución desde la cual se prueba.

5) Ingreso de Datos

Si se cuenta con un sistema que permita el ingreso de información hacia una base de datos, se debe revisar en la tabla de destino que efectivamente se estén enviando los datos de la manera que se ha previsto.

6) Reingreso y Corrección de Datos

Para mejorar la interacción del Sitio Web, cuando tras el ingreso y envío de los datos de un formulario (después de la validación local del formulario) el usuario presiona el botón «Back» de su programa visualizador para volver atrás y modificar algún campo, se le deben presentar todos los datos que hayan sido ingresados. De esta manera se aprovecha la información ingresada previamente, evitando la frustración del usuario por tener que escribir nuevamente el contenido completo del formulario.

7) Elementos de Interfaz

Al usar elementos del lenguaje HTML para la creación de las pantallas (input boxes, combo boxes, list boxes, radio y check buttons, etc.), se recomienda no modificar radicalmente sus atributos de despliegue (colores, formas) y comportamientos tradicionales, para lograr que el usuario sepa intuitivamente cómo usarlo y no deba aprender de nuevo su operación.

8) Multiplataforma

Se debe comprobar que los formularios funcionan en diferentes versiones de programas visualizadores (browsers), de sistemas operativos y de tipos de conexión a Internet (conmutado, banda ancha y dedicado).

9) Botones de Interacción

Si se cuenta con botones interactivos que permiten imprimir, enviar una página a un amigo, etc. Se debe validar que estén realizando correctamente la acción indicada.

4.2.1.2.2.- Sistemas de Búsqueda

Si se cuenta con ellos, se debe validar que efectivamente permitan encontrar documentos existentes en el sitio; en este sentido se deben ingresar documentos específicos y luego buscarlos de manera de asegurarse que la funcionalidad está operando adecuadamente. Si el sistema de búsqueda tiene una versión de «búsqueda avanzada», se debe asegurar de que las opciones ofrecidas encuentren los documentos de la manera en que se ofrezca. El formulario para hacer la búsqueda debe ser intuitivo, evitándose el lenguaje técnico y específico que impida entender su funcionamiento entre usuarios con menores conocimientos de los temas abordados en la institución.

4.2.1.2.3.- Sistemas de Feedback

Si se cuenta con sistemas de envío de preguntas o reclamos (al estilo de los indicados para la Oficina de Informaciones, Reclamos y Sugerencias, OIRS), se debe asegurar de que se está completando el ciclo de vida de la consulta. En este sentido se debe validar que el sitio realiza la consulta y que ésta es recibida por el funcionario encargado de atenderla. De otra manera, la funcionalidad podría operar computacionalmente pero no en términos de tramitación.

4.2.1.2.4.- Sistemas de Compra

Si se cuenta con sistemas de pago en línea, se debe revisar cuidadosamente el flujo de trabajo de la aplicación y asegurarse de que en cada uno de los pasos se está asegurando la calidad y seguridad de la transacción.

4.2.1.2.5.- Administración del Error 404

Cuando se ingresa una dirección equivocada, el software del servidor web muestra una pantalla de error anunciando el número de código del problema (Error 404). No obstante, dicho software puede ser configurado para que muestre una página diferente, en la que se explique a los usuarios las probables razones del error. Es importante incluir, en dicha página, un enlace al Mapa del Sitio y un Buscador, de tal manera que el usuario tenga más herramientas para resolver la inexistencia del contenido que buscaba. Se recomienda, además, que el Administrador de Sistemas de la institución entregue un reporte semanal basado en los «logs» del servidor, que permita ver qué es lo que más buscan los usuarios y de qué manera el Sitio Web les está respondiendo sus consultas.

4.2.1.3.- Pruebas de Carga

La carga de trabajo se refiere a la capacidad máxima que tiene un servidor web (hardware y software), para atender a un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación.

Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas.

Es importante considerar que si el servidor está en las dependencias de un tercero que entrega el servicio de alojamiento del Sitio Web (hosting), se le debe solicitar a dicho proveedor un informe en que dé a conocer las características de carga de la solución de hardware y software sobre la cual funciona el Sitio Web de la institución.

Hay diversos software en el mercado que están orientados a este tipo de simulaciones, todos los cuales ofrecen características similares. Entre los datos más relevantes que es posible obtener se cuenta:

- Tiempo de acceso de los usuarios a los datos
- Volumen de datos y ancho de banda utilizado
- Archivos solicitados y tiempos usados en transferencia de datos
- Tiempo de espera de los usuarios tras hacer un clic
- Tiempo de respuesta a clicks de usuarios
- Niveles de error existentes tras clicks de usuarios

Como se puede apreciar del listado anterior, los reportes que se obtienen a través de esta vía se refieren a tiempos de acceso que tienen los usuarios que acceden al Sitio Web y la degradación que ocurre en los servicios cuando aumenta el volumen de visitantes concurrentes.

Con los resultados obtenidos con pruebas de este tipo se debe hacer una revisión acuciosa de los sistemas, con el fin de hacer las optimizaciones que aparezcan como necesarias. Asimismo, se debe tener en cuenta que será normal la existencia de situaciones excepcionales que harán que los servicios no funcionen adecuadamente.

4.2.1.4.- Pruebas de Seguridad

Las actividades que se pueden realizar para hacer las pruebas de seguridad son diversas y se orientan a varios ámbitos, como se describe a continuación. Los temas a tratar son los siguientes:

- Manejo de DNS
- Protección de Estructura Interna del Sitio Web
- Manejo de Privacidad
- Canales seguros
- Mecanismos de Control de Acceso
- Protección de Programas
- Hosting vs. Sitio Propio
- Roles Mínimos a asegurar

A continuación se entrega información para cada uno de ellos.

4.2.1.4.1.- Manejo de DNS

Un aspecto que se debe cuidar es el de utilizar un nombre de dominio adecuado y relacionado con la identidad y misión de la institución. No obstante, gracias a la forma de operación del Domain Name Service (DNS o Servicio de Nombre de Dominio) es posible asignar más de un nombre de dominio a un mismo Sitio Web.

De esta manera es posible incorporar otras denominaciones, bajo el dominio u otro, que permitan generar «alias» adicionales para el sitio y así permitir utilizar las denominaciones más coloquiales con la cual la institución es conocida por los ciudadanos. No obstante, sin importar cuántos alias tenga un sitio, se recomienda que todos los dominios sean redirigidos para que la primera pantalla, en cualquier caso, corresponda a la portada «oficial» del Sitio Web.

4.2.1.4.2.- Protección de la Estructura Interna del Sitio Web

Uno de los mecanismos que permite proteger la estructura interna del sitio (especialmente para casos de intentos de ataques externos y/o intentos de violación de confidencialidad), es disminuir la cantidad de información contenida en las URL que se muestran en el programa visualizador. Esto es importante respecto de directorios y nombres de programas, pero especialmente en lo que se refiere a la entrega de parámetros de sesión, datos de usuario u otro mecanismo de transferencia de información entre páginas y/o secciones de código.

Se recomienda que los mecanismos de traspaso de información entre páginas sea a nivel de objetos del servidor, asociados a la sesión, sin que la interacción con el lado cliente deba hacerse responsable de la transferencia de datos y/o información entre sesiones de ejecución del servidor.

De igual forma, se recomienda evitar que el acceso a elementos del servidor web esté asociado a «direccionamientos relativos por sesión» o asociados al UserId o SessionId; esto se debe a que mediante simples pasos se puede

conocer «token» de sesión y gracias a eso simular que es el mismo usuario que regresa al sitio. Para evitar el problema se recomienda incorporar protecciones de dirección relativas a la Dirección IP de origen.

Otro método de protección de estructura interna consiste en deshabilitar (excepto para casos excepcionales, como repositorios públicos de archivos) la navegación sobre directorios mediante el servidor web. Esta protección se hace para todos los directorios desde la configuración del software del servidor web.

Otra forma de hacerlo consiste en incorporar un archivo por omisión del servidor web en todos los directorios, aun cuando no sea directamente referenciado por otras páginas para que se muestre su contenido cuando un usuario intente revisar el contenido existente en el directorio. En el caso de habilitar la navegación sobre directorios, se debe evitar el acceso a ciertos directorios protegidos.

Junto con estas protecciones de lectura, se recomienda realizar periódicamente una revisión de los esquemas de permisos otorgados a directorios y archivos. Las permanentes actualizaciones del software de un servidor web, generalmente provocan un problema de control del acceso a nivel de archivos, lo cual requiere procedimientos claros de supervisión.

4.2.1.4.3.- Manejo de Privacidad

Mantener la privacidad de los usuarios debe ser un objetivo permanente del sitio. Para ello se requiere de contar con una Política de Privacidad formal y explícita en el sitio y, además, deben existir mecanismos de seguridad concretos para proteger los datos de sus usuarios.

Entre estos, se debe contar con protecciones físicas y lógicas sobre dicha información.

En el caso de disponer de arquitecturas multicapas reales, se recomienda proteger la información de clientes en servidores físicos distintos de almacenamiento de datos, incluyendo interfaces idealmente separadas de consulta de datos. Además, incorporar mecanismos de encriptación de los datos para información sensible.

Se recomienda que la información, si es almacenada para efectos de que los usuarios la recuperen desde el Sitio Web, sea encriptada con claves administradas por ellos mismos (por ejemplo, su clave de autenticación frente al sitio).

Una decisión de arquitectura que disminuye el riesgo de robo de información de clientes o cuentas de acceso, consiste en evitar que desde la Base de Datos sea posible generar parejas UserId/Clave que permitan autenticarse frente al sitio. La forma de hacerlo es incorporar mecanismos que almacenen un valor de índice de la clave en la Base de Datos, en vez de almacenar la clave propiamente tal.

Gracias a esto, cuando un cliente se autentica frente al sitio, la comparación de claves se realiza sobre los valores de índice y se evita conocer directamente esa información.

Finalmente, se recomienda un control particular de todos los procesos de respaldo, recargas, manejo de medios removibles y generación de copias de información, por ser mecanismos internos de fugas o compromiso de confidencialidad de la información.

4.2.1.4.4.- Canales Seguros

Es importante incorporar mecanismos de encriptación del canal de comunicaciones (como el protocolo Secure Socket Layer o SSL), para la transferencia de información privada entre los usuarios y el Sitio Web, a través de la red Internet. Hacerlo no requiere de programación adicional a las

funcionalidades de interacción, y asegura la protección de toda la información intercambiada entre el servidor y los usuarios.

Desde un punto de vista de desempeño, si bien el inicio («hand shaking») del proceso de establecimiento del canal SSL puede significar un pequeño retardo en la conexión inicial, posteriormente no provoca un aumento significativo del ancho de banda utilizado en la conexión, ni tampoco obliga a un aumento significativo de recursos del servidor.

Es importante considerar que la seguridad asignada a un Sitio Web debe ser correspondiente con la inversión y la importancia de los datos almacenados, siendo estas capacidades obligatorias para el caso de los sitios transaccionales.

4.2.1.4.5.- Mecanismos de Control de Acceso

Otro aspecto que genera mejoras en la protección de la privacidad de los usuarios y de la información contenida en el Sitio Web, es la incorporación de mecanismos modernos de generación de claves y autenticación, como los que se plantean a continuación.

1) Firma Electrónica Simple y Avanzada

Es un sistema que identifica al usuario cuando realiza trámites a través de Internet o redes cerradas. Existe una ley y el correspondiente reglamento que la regula y empresas que las ofrecen en el mercado (más información en <http://www.entidadacreditadora.cl>). Ambas firmas constituyen las bases legales para que ciudadanos y empresas puedan identificarse virtualmente y de esa manera enviar comunicación y hacer negocios de manera más segura y confiable.

Se trata de un mecanismo de complejidad media, aunque incluye funcionalidades de seguridad y criptografía. No obstante, la incorporación de este mecanismo en forma única dependerá del control absoluto que se tenga de la comunidad de usuarios de la solución. Para comunidades abiertas es preferible

establecer dos mecanismos de autenticación: uno fuerte, mediante Firma Electrónica (usando certificados digitales) y otro, mediante autenticación de Usuario y Clave. Por otro lado, la Firma Electrónica Simple podría ser usada para las comunicaciones oficiales enviadas por la institución a sus usuarios. El uso de la Firma Electrónica debe definirse al momento de determinar la arquitectura de solución del Sitio Web.

2) Autenticación con par Usuario y Clave

Si se emplea esta solución, debe existir un procedimiento concreto para los casos en que un usuario pierda o no recuerde su clave. Se recomienda ofrecer mecanismos de «regeneración de clave», mediante la entrega de respuestas a preguntas predefinidas por los usuarios, en lugar de hacer el «envío de la clave por e-mail». En el caso de contar con mecanismos de Ayuda, se debe ofrecer apoyo para la regeneración de las claves, sin que el personal de la institución tenga acceso a la información de seguridad del cliente. Se debe evitar el uso de mecanismos de autenticación administrados por terceros, en caso de que puedan comprometer la confidencialidad o la suplantación del usuario.

3) Sistemas de Hardware para Autenticación

Para sistemas de seguridad que requieren una autenticación absoluta del usuario, es preferible considerar mecanismos de autenticación fuerte. Para ello, se deben incorporar mecanismos que incluyan elementos de hardware que deben estar en posición del usuario, tales como tarjetas u otros similares (security token) que permiten el acceso a las áreas de autenticación. Allí el usuario debe ingresar su identificación de Usuario (security challenge response) y se le genera una clave de sesión que al ser digitada en pantalla, le permite acceder al sistema. Dicha clave cambia frecuentemente para aumentar la seguridad de acceso.

4.2.1.4.6.- Protección de Programas

Es fundamental proteger los códigos y programas internos del servidor web, particularmente evitando la transferencia de parámetros o información a través de

la dirección de acceso a las páginas (por ejemplo, al usar el método GET para la entrega de parámetros), los cuales son mecanismos frecuentes de «hacking» o robo de información.

De igual forma, se debe evitar la lectura de ejecutables desde los directorios del servidor, controlando los permisos adecuados de acceso a éstos, con el fin de evitar desensamblaje y/o ingeniería reversa para analizar accesos internos.

En cuanto a los «scripts» ubicados en el lado del cliente, en caso de ser críticos, se recomienda utilizar compactadores de código y eliminar documentación de apoyo que permita su fácil comprensión a partir de la lectura del código.

Es importante que estas medidas sean incluidas junto a las acciones de seguridad informática normales de la institución.

4.2.1.4.7.- Hosting Externo vs. Sitio Propio

Sin entrar en profundidad en cuanto al detalle de los elementos a considerar para esta decisión, la principal recomendación es hacer una evaluación objetiva basada en los siguientes aspectos:

- Evaluar las reales capacidades disponibles para la operación permanente del sitio, desde un punto de vista técnico.
- Evaluar los requerimientos de control y seguridad necesarios.
- Evaluar el nivel de soporte efectivo que el personal técnico del servicio puede realizar sobre los servidores.

Con estos parámetros se debe definir la mejor opción, no sólo desde el punto de vista del interés de las áreas técnicas, sino que mediante una evaluación de impacto global de la decisión asociada.

La amplia oferta disponible permite realizar combinaciones de servicios e infraestructura de muy diversos tipos, lo cual facilita configurar una solución óptima en términos del costo-beneficio asociado (por ejemplo, hosting compartido, dedicado, collocation, housing, red administrada, monitoreo de seguridad, administración de seguridad perimetral, control de aplicaciones, fulfillment, etc.).

En caso de que se decida externalizar esta área, es importante contar con altos estándares de parte del proveedor en todo lo referido a tiempo de desempeño («uptime»), respaldos y recuperación, actualizaciones de software, etc.

4.2.1.4.8.- Roles Mínimos a Asegurar

Un último aspecto considerado en esta área de recomendaciones, consiste en definir los diversos roles profesionales dentro de la definición y diseño de un Sitio Web para una institución.

Desde un punto de vista exclusivamente técnico, es fundamental considerar al menos los siguientes roles, identificando tanto sus responsabilidades como el personal más competente que pueda cumplirlos.

Si bien más de uno de estos roles funcionales puede ser desarrollado simultáneamente por una persona o área de la organización, es importante que dichas áreas sean cubiertas no sólo durante la puesta en marcha de la solución sino también durante su etapa de producción.

- Arquitecto: encargado de hacer las configuraciones de trabajo de los servidores y aplicaciones.
- Administrador de Aplicaciones: encargado del funcionamiento del software operativo.
- Administrador de Control de Calidad: encargado del cumplimiento de las políticas de calidad.

- Administrador de Seguridad: encargado de hacer generar y hacer cumplir las directivas de seguridad.
- Administrador de Operaciones: encargado de los aspectos operativos relacionados con el hardware.
- Administrador de Contenidos: encargado de las informaciones contenidas en el Sitio Web.
- Administrador de HelpDesk y Soporte: encargado de dar soporte a usuarios sobre las funcionalidades del Sitio Web.
- Administrador de Contingencias: encargado de enfrentar en primera línea los problemas que se generen en la operación.
- Auditor / Contralor: encargado de llevar registro de las operaciones realizadas, con el fin de apoyar la revisión de procedimientos.

Finalmente, aunque los roles del área Informática pueden estar muy claros, es necesario que se entienda que la operación del Sitio Web es una tarea conjunta en la que participan funcionarios de diversas áreas de la institución.

4.2.1.5.- Pruebas de Respaldo y Recuperación

Respaldo la información de un Sitio Web se refiere a copiar el contenido completo del sistema (datos, programación, imágenes, etc.) a un medio que sea confiable, que esté en un lugar seguro y que permita la recuperación de manera rápida y eficiente.

En este sentido, hay que preocuparse no sólo de probarla confiabilidad del sistema al momento de respaldar, sino también para la acción de recuperar y volver a instalar lo respaldado.

4.2.1.6.- Registro y Control de Pruebas y Errores

Para que una prueba sea válida, debe ser lo más documentada posible, con el fin de que, quien deba efectuar la corrección, pueda replicar el error para analizarlo y luego proceder a tomar medidas correctivas. Para ello se recomienda llevar una planilla de cálculo (Ver Anexos) en que se vayan anotando por columna los siguientes datos:

4.2.1.6.1.- Detección del Error

Para ser anotado por quien prueba.

- 1) Módulo: Indica la sección en la que se produce el error.
- 2) URL: dirección de la página donde ocurrió el error.
- 3) Acción: Indicar la secuencia de pasos que siguió para que ocurra el error.
- 4) Lo que hace o dice: es la explicación más detallada posible del error, en particular señalando la secuencia de pasos seguida hasta dar con el error.
- 5) Lo que debe hacer o decir: se debe indicar lo que se espera que debería ocurrir cuando se hace la acción que se ha descrito.
- 6) Encontrado por: nombre de quien prueba.
- 7) Fecha: fecha en la que se hace la anotación.
- 8) Reproducible: indicar si el error se repite al hacer nuevamente la prueba.
- 9) Clasificación: permite definir el grado de complejidad del error, al señalar si afecta el funcionamiento del sitio (caso extremo) o sólo su presentación.

4.2.1.6.2.- Diagnóstico del Error

Para ser anotado por quien corrige.

- 1) Causa: motivo por el cual se produce el error.

- 2) Efectos laterales: indicar en qué otros módulos la existencia de este error puede estar causando impacto negativo; muchas veces errores diversos tienen una causa común, por lo que al reparar ésta se arreglan los demás.

4.2.1.6.3.- Corrección del Error

Para ser anotado por quien corrige.

- 1) Descripción: acción realizada para hacer la reparación del error.
- 2) Archivos intervenidos: archivos en los que se hicieron modificaciones o, al menos, el principal de ellos.
- 3) Corregido por: persona que hizo la corrección.
- 4) Fecha corrección: fecha en que quedó reparado el error.

4.2.1.6.4.- Comprobación de la Corrección

Para ser anotado por quien revisa la corrección realizada.

- 1) Revisor: Nombre de quien revisa si el error fue efectivamente reparado.
- 2) Fecha: fecha en que se realiza la revisión.
- 3) Reparado: indicar si está reparado o no. Si no lo está, se debe copiar la línea de error en blanco en una nueva planilla, con el fin de solicitar nuevamente el proceso de corrección.

4.2.2.- EJECUCIÓN DEL PLAN DE PRUEBAS

Esta información no puede estar publicada, ya que es información reservada. Si usted desea puede acercarse a la Unidad de Tecnologías de Información y Comunicación de la ESPE-L, para conocer sobre esta información.

4.3.- PUESTA EN MARCHA

4.3.1.- DESARROLLO DE UN PLAN DE LANZAMIENTO

Para hacer el lanzamiento de un nuevo Sitio Web es obligatorio que el nuevo sitio haya cumplido adecuadamente las pruebas antes descritas, con el fin de que todos los contenidos prometidos estén incorporados y las funcionalidades realicen todo aquello que se describe respecto de ellas.

Si hay contenidos o funcionalidades descritas que no pueden estar disponibles para el momento en que se desea hacer el lanzamiento del Sitio Web, es preferible eliminarlos en ese momento e incorporarlos cuando estén listos, en lugar de dejarlos en el sitio y que den una mala imagen sobre el mismo.

4.3.1.1.- Lista de Chequeo Previa

Para llegar al lanzamiento del sitio, se recomienda asegurarse del cumplimiento de las siguientes actividades como mínimo:

- 1) Cumplir Listas de Chequeo: el sitio debe haber cumplido adecuadamente las pruebas indicadas en este capítulo, antes de hacer su lanzamiento.
- 2) Dominio Distintivo: se debe contar con un nombre de dominio que sea reconocible y se asocie a la institución, de tal manera que sea fácil relacionarlo con la actividad o el nombre de la misma. Es importante en este aspecto que se recuerde la obligación de inscribir el Sitio Web dentro del dominio relacionado a la actividad que desempeña.
- 3) URL Simple: la dirección de acceso de la primera página del sitio debe ser simple, de tal manera que sea fácil comunicarla. Idealmente no se debe mencionar el nombre del archivo de inicio (que corresponde a su página

inicial o portada), si sólo con el nombre del dominio se puede acceder a ella.

- 4) Chequear Disponibilidad: si el dominio es nuevo y recién se está levantando un Sitio Web en él, antes de lanzarlo se debe verificar que el sitio se ve desde diferentes lugares, para asegurar su disponibilidad para diferentes públicos.
- 5) Respaldo Administrativo: muchas veces los sitios generarán necesidades de interacción entre los usuarios de un servicio y los funcionarios del mismo, lo cual podrá estar resuelto a través de comunicaciones generadas por medio del Sitio Web. Si este es el caso, antes del lanzamiento se debe incorporar en el flujo de trabajo.

4.3.2.- DESARROLLO DE UN PLAN DE COMUNICACIONES

Una vez que se han hecho las comprobaciones descritas en el título anterior, se está en condiciones técnicas de lanzar el sitio. Lo que viene a continuación es realizar la presentación e incorporar el Sitio Web a las actividades de difusión de la institución.

Para hacerlo, se debe contemplar que dicho plan debe tener componentes online y offline, tal como se indica a continuación:

4.3.2.1.- Actividades Online

Dado que estamos presentando un medio de comunicación tecnológico, es importante cubrir adecuadamente esta área a través de las siguientes actividades:

- 1) Registrar el Sitio en Buscadores: es la actividad mediante la cual el Sitio Web comienza a formar parte de todos los directorios y buscadores de Internet. Si bien hay empresas que ofrecen esta actividad como un servicio,

esto puede ser realizado por cualquier encargado del sitio, sin necesidad de tener conocimientos técnicos avanzados. A continuación cuatro lugares «claves» donde inscribir el sitio:

Google - <http://www.google.com/intl/es/addurl.html>

Yahoo - <http://e1.docs.yahoo.com/info/sugerir.html>

Dmoz.org - <http://www.dmoz.org/World/Espa%1ol/add.html>

- 2) Generar Enlaces con Otros Sitios: varios de los algoritmos que usan los sitios de búsqueda y los directorios para incluir un sitio y mostrarlo en los primeros lugares de un directorio, revisan la cantidad de enlaces «desde sitios importantes» que llegan al sitio. Para ello, el administrador del sitio debe propiciar los enlaces hacia el sitio y conseguir que siempre haya nuevos. Para ver cuántos llegan desde otros sitios web, se puede usar en www.google.com y www.altavista.com la instrucción «link».

Ejemplo - link:<http://www.premioweb.cl/> para ver cuántos enlaces referencian el sitio del Premio Web.

- 3) Ofrecer Elementos de Fidelización: se refiere a ofrecerle a los usuarios motivos diversos para volver al sitio; puede ser un boletín de noticias en el que se envíen enlaces con contenidos de interés; fondos de pantalla; información útil de áreas relevantes, etc. Cada institución puede buscar dentro de sus contenidos, aquellos que son los más buscados por sus usuarios y ofrecerlos de manera atractiva para garantizar que siempre estén accediendo y regresando al sitio.

4.3.2.2.- Actividades Offline

Se refiere a todas las actividades que se realizan fuera del ambiente Internet, con el fin de consolidar también en este mundo la «marca Internet» de la institución. Incluye las siguientes acciones:

- 1) Imagen Corporativa: la dirección del Sitio Web de la institución debe incorporarse en la imagen corporativa de la institución para que todo documento de la institución la incluya (desde informes internos, hasta tarjetas de visita). De esta manera, se logrará una unidad muy concreta en términos comunicacionales y se dejará diseminada esa dirección en todos lugares, permitiendo difundirlo y hacerlo conocido entre quienes deseen ponerse en contacto o revisar información provista por la institución.

- 2) Actividades de Prensa: en el lanzamiento del Sitio Web se debe ofrecer un elemento tecnológico atractivo y no sólo confiar en que la aparición del Sitio Web sea la noticia. Por lo anterior, se debe definir cuál de las funcionalidades del sitio podrá ser destacada, para transformarla en la noticia que convoque a los medios.

4.3.3.- EL SITIO COMO APOYO DE LA INSTITUCIÓN

Una vez que el Sitio Web está operando normalmente, la tarea a realizar es incorporar el Sitio Web en el plan de comunicaciones, es decir, no sólo utilizar el web para hacer difusión, sino que también comenzar a incorporar la dirección web en cualquier comunicación que se haga y, más aún, hacer que el Sitio Web forme parte de las actividades.

Un ejemplo virtuoso de esto lo han dado muchas instituciones que insertan publicidad en los medios de comunicación para diferentes actividades y

normalmente lo hacen en un aviso de pequeño formato, en el caso de prensa escrita, en el que indican que toda la información está disponible en el Sitio Web de la institución.

Esto genera dos efectos: Baja los costos de la publicación del aviso, Permite anunciar de manera simple y efectiva la existencia de un Sitio Web activo y útil.

4.3.4.- ENLACES DESDE OTROS SITIOS

Es importante que el Sitio Web de la institución esté enlazado desde otros sitios web reconocidos de la Internet (buscadores, directorios, instituciones reconocidas), para mejorar su posición relativa y aparecer más arriba en las páginas de resultados de los sistemas de búsqueda de Internet.

Para conseguirlo, es muy relevante que como parte del plan de puesta en marcha del sitio, se asegure la existencia de los enlaces, especialmente desde y hacia los Sitios Web sectoriales a los que pertenezca la institución.

4.4.- EVALUACIÓN

4.4.1.- MÉTRICAS DE EVALUACIÓN DE DESEMPEÑO INTERNAS Y EXTERNAS

Una de las características interesantes que tiene un Sitio Web, es que ofrece información permanente de las actividades que están ocurriendo en su interior, lo que permite tener siempre cifras que ayudan a evaluar la gestión.

No obstante, para entender dichas cifras es necesario primero establecer ciertas definiciones, que tienen que ver con la terminología que se emplea en Internet para describir los fenómenos que se registran. En este sentido, hay tres conceptos importantes:

- 1) Hits: se refiere a cada elemento que pasa desde el servidor del sitio al computador del usuario; una página puede tener muchos hits, ya que se cuenta uno por cada elemento que la compone. En términos reales, esta información no tiene valor.
- 2) Páginas Vistas o Visitadas: se refiere a la cantidad de páginas que han sido solicitadas por los usuarios al Sitio Web; su uso más importante tiene que ver con la capacidad de establecer rankings internos en el sitio, respecto de los contenidos y funcionalidades más vistos y usados.
- 3) Sesiones de Usuario: se refiere al número de personas que ha visitado el Sitio Web, independiente de cuántas páginas hayan visto o solicitado durante su visita; esta métrica es la única que puede entregar información real en torno a la audiencia de un sitio.

4.4.2.- LA IMPORTANCIA DEL ARCHIVO «LOG»

Con las definiciones anteriores ya especificadas, hay que indicar que el servidor web en que está alojado el sitio va generando un archivo de texto donde se registra línea por línea cada operación que realiza un usuario en el Sitio Web.

Los datos que incorpora este archivo pueden ser definidos según las necesidades que existan en el Sitio Web. Entre los más importantes se encuentran la fecha; hora; Número IP de origen; páginas visitadas; páginas desde las cuales llegaron al sitio; palabras que usaron el buscador y países de origen de los visitantes, entre otros.

Este archivo aumenta de tamaño a medida que hay más visitantes, a razón de 1 Mb por cada 10 mil hits; por lo anterior, es muy importante coordinar su extracción y análisis, con el fin de obtener información que ayude a la gestión y a entender mejor la forma en que los usuarios están empleando el sitio.

Para hacer el análisis del «log» del servidor existe una gran cantidad de herramientas. Para acceder a ellas se recomienda visitar [http://www.uu.se/Software/ Analyzers /Access-analyzers.html](http://www.uu.se/Software/Analyzers/Access-analyzers.html)

4.4.3.- PRESENCIA DEL SITIO EN BUSCADORES

Otra métrica relevante para saber el grado de efectividad que está teniendo el Sitio Web, consiste en revisar periódicamente su presencia a través de los buscadores de Internet más populares.

La recomendación es hacerlo a través de los buscadores más populares, como son Google, Altavista y Yahoo!, puesto que ellos son los que concentran el mayor

tráfico y, por lo tanto, que el Sitio Web aparezca en ellos garantizará que los usuarios que estén buscando la institución la podrán encontrar.

Adicional a ellos, se recomienda que el sitio sea indexado en directorios como el DMOZ.org, puesto que se trata de un proyecto realizado a nivel mundial, cuyo contenido es ofrecido gratuitamente y actualmente es ocupado por los directorios más importantes. Por lo tanto, al estar en DMOZ.org se garantizará la aparición en los demás sitios.

Es importante que la búsqueda que se haga del sitio en los buscadores y directorios con el fin de ver cómo está indexado, se haga a través de dos criterios:

- 1) Buscar por el Nombre: cuando se busca por el nombre de la institución, normalmente el Sitio Web debería aparecer en los primeros lugares de la primera página. Si no es así, hay un trabajo fuerte que hacer para mejorar los meta tags.

- 2) Buscar por los Temas: cuando se busca por los temas que maneja la institución, es menos claro que su Sitio Web aparezca en los primeros lugares. Para mejorar ese posicionamiento es necesario refinar los meta tags y la forma de desplegar la información en el sitio, con el fin de ir aumentando la posibilidad de que siga ascendiendo en las listas de los buscadores.

CAPÍTULO V

V.- CONCLUSIONES Y RECOMENDACIONES

5.1.- CONCLUSIONES

- La metodología OOHDM, muestra que a través de este método podemos construir aplicaciones hipermedias complejas y evolutivas, dando esto sin duda a una de las metodologías que mas aceptación ha tenido y sigue teniendo en el desarrollo de aplicaciones hipermedias.
- OOHDM es una metodología basada en el diseño, que ofrece una serie de ideas. La primera de ellas es que hace una separación clara entre lo conceptual, lo navegacional y lo visual. Esta independencia hace que el mantenimiento de la aplicación sea mucho más sencillo. Además, es una de las metodologías que hace un estudio profundo en los aspectos de interfaz ya que esto es un punto crítico en cualquiera de las aplicaciones web que se desarrollan actualmente.
- La metodología OOHDM emplea un ciclo de vida iterativo incremental el cual permite depurar cada modelo en las diferentes etapas hasta llegar a obtener el mejor resultado.
- Sin embargo, y a pesar de esto, OOHDM presenta algunas deficiencias. OOHDM ha dejado fuera de su ámbito un aspecto esencial que es el

tratamiento de la funcionalidad del sistema. Esto es qué se puede hacer en la aplicación y en qué momento de la navegación o de la interfaz se puede hacer, es algo que no trata y que lo deja como tarea de implementación.

- Además, OOHDM no ofrece ningún mecanismo para trabajar con múltiples actores. Por ejemplo, imaginemos que la interfaz y la navegación de la aplicación varía sustancialmente dependiendo de quién se conecte a la aplicación. El diagrama navegacional, los contextos navegacionales y los ADVs resultarían muy complejos para representar esta variabilidad.
- OOHDM todavía es una metodología de proyecto joven; además, el proyecto de hipermedia es un campo relativamente nuevo en los aspectos de ingeniería de software relacionados con esta área, esto en la actualidad recientemente se está considerando en la literatura de software.
- En la etapa de Pruebas es muy importante realizar un Plan Estratégico de Pruebas, clasificando tipos de Pruebas con un alto grado de relevancia, e incluso ciertos ítems a ser probados, para que una prueba sea válida, debe ser lo más documentada posible, con el fin de que, quien deba efectuar la corrección, pueda replicar el error para analizarlo y luego proceder a tomar medidas correctivas.
- Luego que se han realizado las comprobaciones respectivas al Portal Web, existe una necesidad de realizar un Plan Estratégico de Lanzamiento del Portal en los diferentes medios de comunicación e incorporar el Portal Web a las diferentes actividades de la Institución.

5.2.- RECOMENDACIONES

- Realizar un análisis minucioso en el levantamiento de información, requerimientos de la aplicación, siendo fundamental durante el desarrollo de la aplicación, ya que OOHDM marca como pauta principal los Diagramas de interacción de usuario UID's.
- OOHDM no cuenta con una herramienta CASE que permita automatizar toda la documentación, por lo que es muy necesario implementar una herramienta de este tipo.
- En cada parte de las etapas de OOHDM realizar validaciones y pruebas de acuerdo a cada etapa.
- Elaborar una metodología para crear y desarrollar aplicaciones web analizando las metodologías de hipermedia, usabilidad y multimedia existentes aplicando normas estandarizadas de UML, abstrayendo lo mejor que tiene cada metodología, con un estudio profundo y tratar de obtener una metodología que cumpla todos los requerimientos del caso durante todo el desarrollo, mantenimiento y documentación de una aplicación web y con soporte de una herramienta Case.
- Para los canales seguros de Internet como los certificados SSL que estos sean firmados por una entidad emisora debidamente certificada para luego incorporar estos certificados en el servidor Web.

- Revisar periódicamente los archivos logs del servidor web, para chequear cuales son las paginas del Portal que son mas visitadas, se puede realizar este mantenimiento con la herramienta webanalyzer.

REFERENCIAS BIBLIOGRÁFICAS

- **Pressman, 2002**

Pressman Roger S: Ingeniería del Software, Quinta edición, McGraw-Hill, España, 2002

- **Vaughan, 2002**

Vaughan Tay: Multimedia manual de referencia, Quinta edición, McGraw-Hill, España, 2002

- **Nielsen, 2000**

Nielsen Jacob: Usabilidad, Diseño de sitios Web, Pearson Educación, Madrid, 2000

DIRECCIONES WEB

- **Ingeniería Web y Usabilidad**

http://www.htmlweb.net/usabilidad/usabilidad_1/usabilidad_1_3.html

<http://www.useit.com>

<http://www.w3c.org>

<http://iteso.mx/~carlosc/pagina/documentos/usabilidad.htm>

http://griho.udl.es/publicacions/2003/IU_sitios_Web.pdf

<http://www.aldeamac.com/basic/web/web001.html>

<http://www.dsic.upv.es/~ideas2002/tutoriale.html>

<http://www.dsic.upv.es/~ideas2002/tutoriale.html>

<http://www.mundofree.com/xider/otras/metrica.html>

http://www2.eside.deusto.es/asignaturas/tim/TIM_Sonido.html

- **OOHDM**

<http://www.tecweb.inf.puc-rio.br/oohtm/space/start>

<http://server2.tecweb.inf.puc-rio.br:8000/projects/hyperde/trac.cgi/wiki>

<http://www.telemidia.puc-rio.br/oohtm/>

<http://www.rspa.com/spi/index.html>

- **Archivos sobre OOHDM enviados por sus autores Daniel Schwabe y Gustavo Rossi**

Archivo	Descripción
Notação_OOHDM.doc	Notación actualizada sobre OOHDM enviado por el Dr. Daniel Schwabe
oohtm.zip	Metodología OOHDM enviado por el Dr. Gustavo Rossi

ANEXOS

Anexo

A

**PLANTILLA DE PRUEBAS Y SOFTWARE
PARA PRUEBAS**

Información en CD

Anexo
B

MANUAL DE ADMINISTRADOR

Información en CD

Anexo
C

MANUAL DE USUARIO

Información en CD

Anexo
D

GLOSARIO

HTML.- Son las siglas de (Hyper Text Markup Language). Lenguaje de marcado sencillo utilizado para crear documentos de hipertexto compatibles con varias plataformas. Los archivos HTML son archivos de texto ASCII sencillos con códigos incrustados (indicados por etiquetas de marcado) que indican formato y vínculos de hipertexto.

XML.- Son las siglas de eXtensible Markup Language. Lenguaje de metamarcado que proporciona un formato para describir datos estructurados. Esto facilita declaraciones más precisas del contenido y resultados de la búsqueda más significativos en múltiples plataformas. Además, XML permitirá una nueva generación de aplicaciones de presentación y manipulación de datos basadas en Web.

CSS.- Son las siglas de Cascading Style Sheets, hojas de estilo en cascada. La hoja de estilos es un conjunto de instrucciones que definen la apariencia de diversos elementos de un documento HTML (también XML).

PHP.- Son las siglas de Hypertext Preprocessor es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor.

OOHDM.- Son las siglas de Object-Oriented Hypermedia Design Method. Es la metodología de diseño de hipermedia orientado a objetos.

MULTIMEDIA.- La Multimedia se compone, de combinaciones entrelazadas de elementos de texto, gráficos, sonido y vídeo. Cuando se permite a un usuario, controlar ciertos elementos y cuándo éstos deben presentarse, se denomina Multimedia Interactiva.

HIPERMEDIA.- Cuando se proporciona una estructura de elementos ligados a través de los cuales el usuario puede navegar, entonces la Multimedia Interactiva se convierte en Hipermedia.

UML.- Son las siglas de Lenguaje Modeling Unified, Lenguaje de Modelado Unificado.

UID.- Son las siglas de User Interaction Diagram, Diagrama de interacción de Usuario.

ITEM.- Es un elemento o clase de información.

CTX.- Siglas de los Contextos, identificadores de contexto navegacional.

ADV.- Siglas de Abstract Data View, Vista de Datos Abstracta.

ADO.- Objetos de Datos Abstracto (ADOs, Abstract Data Objects).

GUI.- Interfaz Gráfica de Usuario

FreeTDS.- Siglas de TDS (Tabular Data Stream) que es un protocolo que utilizan varios motores de bases de datos para la transferencia de datos.

SSL.- Siglas de Secure Socket Layer o SSL. Estándar abierto propuesto para establecer un canal de comunicaciones seguro que impida la interceptación de información crítica, como el número de las tarjetas de crédito. Principalmente, permite las transacciones financieras electrónicas seguras en el World Wide Web, aunque está diseñado para funcionar también en otros servicios Internet.

BDD.- Siglas de Bases de datos

W3C.- Siglas del World Wide Web Consorcio, ente que regula los estándares de Internet.

DNS.- Siglas de Domain Name Service, Servicio de Nombre de Dominio. Base de datos jerárquica y distribuida que contiene asignaciones de nombres de dominio

DNS para varios tipos de datos, como direcciones IP. DNS permite la búsqueda de equipos y servicios mediante nombres descriptivos y el descubrimiento de otra información almacenada en la base de datos.

URL.- Siglas de Universal Resource Locator, Localizador de recursos universal. Dirección que identifica de forma única una ubicación en Internet.

FTP.- Siglas de File Transfer Protocol, Protocolo de transferencias de archivos. Miembro del conjunto de protocolos TCP/IP que se utiliza para copiar archivos entre dos equipos en Internet. Ambos equipos deben admitir sus funciones FTP correspondientes: uno debe ser un cliente FTP y el otro un servidor FTP.

SITIO WEB.- Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada Home Page, con un nombre de dominio y dirección en Internet específicos.

PORTAL.- El término "portal" se refiere a un conjunto de Sitios Web, que se usa como punto de partida para visitar otros sitios; los portales generalmente buscan formar comunidad en la localidad geográfica donde operan.

Latacunga, Noviembre del 2006

Sr. Chango Cando Milton Rodrigo
C.I. 0502591340

Sr. Izurieta Naranjo Efrén Leonardo
C.I. 1803263332

Ing. José Luis Carrillo Medina
COORDINADOR DE LA CARRERA DE SISTEMAS E INFORMÁTICA

Dr. Rodrigo Vaca Corrales
SECRETARIO ACADÉMICO