



**Universidad Autónoma del Estado de Hidalgo**

**Instituto de Ciencias Básicas e Ingeniería**

**Centro de Investigaciones en Tecnologías de Investigación y Sistemas**

**PIENSSA**

**“Herramienta para Generar y Documentar un Portal Web Corporativo”**

**Tesis**

**Para Obtener el Título de:**

*Licenciado en Sistemas Computacionales*

**Presentan:**

*Ana Laura López Angeles*

*Héctor Hernández Barrera*

*Juan Francisco Lastiri Monárrez*

**Directora de Tesis:**

*Dra. Aurora Pérez Rojas*

Pachuca de Soto, Hidalgo, Diciembre de 2005

## **Con todo mi agradecimiento...**

### ***A Dios...***

Por concederme la dicha de seguir disfrutando  
de mis triunfos...

### ***A Josefa y Mauricio, mis padres...***

Por darme la vida y enseñarme a valorar lo que tengo,  
por su apoyo incondicional, confianza y cariño.  
Esto es por Ustedes, lo logramos...

### ***A Ana Lilia y Miriam, mis hermanas...***

Saben que cuentan conmigo, gracias por  
escucharme y brindarme sus consejos.  
Las quiero mucho...

### ***A Jesús Leo...***

Una gran persona que me da su amor, amistad,  
comprensión.  
Gracias por ayudarme a concluir esto.  
Te Amo...

***Ana Laura López Angeles***

## **Con todo mi agradecimiento...**

### ***A Dios...***

Por dejarme llegar hasta esta instancia de la vida y dejarme compartir con todos los que quiero un logro más en mi vida.

### ***A mis padres Héctor Manuel y Socorro...***

Por que no solo me dieron la vida, sino también me supieron guiar con sus consejos, sabiéndome inculcar sus principios y valores y formarme como la persona que ahora soy.

### ***A mis hermanos Sandra y Omar...***

Por apoyarme y darme el ánimo que necesitaba cuando no tenía un objetivo en la vida y donde más que ser un ejemplo yo soy quien mas he aprendido de ustedes.

A Sandra por haber dado la luz más hermosa que ha iluminado mi vida, Alan.

### ***A Maricela...***

Por ser la ilusión que alimenta mi espíritu de seguir luchando en esta vida, por saber que estas ahí y que nunca me dejaras caer.

Por ser mi gran Amor...

***Héctor Hernández Barrera***

## **Con todo mi agradecimiento...**

### ***A mi esposa Oti...***

Por ser el motor impulsor de todos mis proyectos y respaldar todas mis decisiones, buenas y malas, soportando horas de ausencia al haber emprendido este proyecto de carrera.

### ***A mi hijo Alex...***

Por ser mi motivo de vivir y de ser mejor en todos los aspectos que puedan servirle de ejemplo, además de brindarme el amor más puro y desinteresado que puede haber.

### ***A mis padres, Eliezer y Bertha...***

Por el apoyo que me han brindado siempre, cada uno a su manera, pero siempre valioso para mí.

### ***A mis hermanos Pepe y Lupita...***

Por que se que siempre contaré con ellos en cualquier circunstancia que la vida me depare.

### ***A todos aquellos...***

Quienes directa o indirectamente me han permitido crecer y me han obligado a forjarme como la persona que ahora soy.

### ***A Dios...***

Por que sin el nada es posible.

***Juan Francisco Lastiri Monárrez***

**Con todo nuestro agradecimiento...*****A la Dra. Aurora Pérez Rojas...***

Por su infinita paciencia al asesorarnos y guiarnos adecuadamente en la elaboración de este trabajo y por todas las buenas cosas que hemos aprendido de ella como profesional, pero sobre todo como ser humano.

***A nuestros Sinodales...***

Por su amables y acertados consejos para enriquecer nuestros conocimientos.

***Ana Laura López Angeles***

***Héctor Hernández Barrera***

***Juan Francisco Lastiri Monárrez***

# Índice

<b>Introducción.</b>	vi
Antecedentes.	ix
Planteamiento del problema.	x
Objetivos.	xi
Justificación.	xiii
<b>Capítulo I. Metodologías, Lenguajes y Servidores</b>	<b>1</b>
1.1. <u>Metodologías</u>	2
1.1.1. Metodologías Estructuradas	2
1.1.1.1. <i>Metodología RNA</i>	2
1.1.1.2. <i>Metodología HDM</i>	3
1.1.1.3. <i>Metodología RMM</i>	3
1.1.1.4. <i>Metodología ERMIA</i>	4
1.1.1.5. <i>Metodología Lowe-Hall's Engineering Approach</i>	4
1.1.2. Metodologías Orientadas a Objetos	5
1.1.2.1. <i>Metodología WSDM</i>	5
1.1.2.2. <i>Metodología SOHDM</i>	6
1.1.2.3. <i>Metodología UWE</i>	6
1.1.2.4. <i>Metodología W2000</i>	7
1.1.2.5. <i>Metodología UWA</i>	8
1.1.2.6. <i>Metodología NDT</i>	9
1.1.2.7. <i>Metodología DESING-DRIVEN REQUERIMENTS ELICITATION</i>	10
1.1.2.8. <i>Metodología OOHDM</i>	10
1.1.2.9. <i>Metodología HFPM</i>	13
1.1.2.10. <i>Metodología MacWeb Approach</i>	14
1.1.2.11. <i>Metodología EORM</i>	14
1.1.2.12. <i>Metodología OO/PATERN APPROACH</i>	14
1.1.2.13. <i>Metodología AHDM</i>	15
1.1.2.14. <i>Metodología SHDT</i>	16
1.1.2.15. <i>Metodología IHDM</i>	16
1.1.2.16. <i>Metodología RUP</i>	17
1.1.3. Comparación de Metodologías Estructuradas y Orientadas a Objetos	21
1.2. <u>Lenguajes de Programación Web</u>	23

1.2.1.	Lenguaje PHP	24
1.2.2.	Lenguaje JAVA	26
1.2.3.	JDBC	27
1.2.4.	Java Servlets	28
1.2.5.	XML	28
1.2.6.	XSL	29
1.2.7.	CGI	30
1.2.8.	Lenguaje ASP	31
1.2.9.	Lenguaje JSP	33
1.2.10.	Lenguaje SSI	33
1.2.11.	Lenguaje HTML	33
1.3.	<u>Servidores</u>	34
1.3.1.	Servidor Apache	34
1.3.2.	Servidor IIS	34
1.3.3.	Servidor Iplanet	34
 <b>Capítulo II. Portales y Calidad Web</b>		 36
2.1.	<u>Portales Web</u>	37
2.1.1.	Artefactos Web como Software	38
2.1.2.	Estudio Previo	39
2.1.3.	Necesidad de Modelos de Proceso de Desarrollo de Artefactos Web	42
2.1.4.	Tipos de Estructuras Web	44
2.1.5.	Primeros Diseños	47
2.1.6.	Fases Operativas de un Sistema de Información Centrado en la Web	48
2.1.7.	Diseño Modular	48
2.1.8.	Características de Portales Web	51
2.1.9.	Programación de la Web	53
2.1.10.	Relación Diseño-Programación	55
2.1.11.	Construcción "Local" de Sitio Web	55
2.1.12.	Cotización de un Proyecto Web	57
2.2.	<u>Calidad Web</u>	58
2.2.1.	Los Primeros Modelos de Calidad del Producto	60
2.2.2.	Otros Modelos o Marcos Conceptuales de Descomposición	63
2.2.2.1.	<i>Modelos de Calidad de Productos en los Estándares ISO 9126 e IEEE 1061</i>	63
2.2.2.2.	<i>Modelo de Calidad conforme al Estándar ISO 9126-14</i>	65
2.2.2.3.	<i>El Modelo MECA.</i>	67

2.2.3.	Seleccionando el Enfoque de Modelo de Calidad	68
2.2.4.	Comparación de los Modelos de Calidad.	69
2.2.5.	Características de la Calidad Web	71
2.2.6.	Atributos que Determinan la Calidad de los Sitios Web.	75
2.2.7.	Otros Criterios o Atributos de la Calidad de los Sitios Web.	77
<b>Capítulo III. Tecnologías</b>		<b>79</b>
3.1.	<u>Herramientas Case para el Modelado de Aplicaciones</u>	80
3.1.1.	Erwin 3.0	80
3.1.2.	ER/Estudio 2.5	84
3.1.3.	PowerDesigner 6.1	87
3.1.4.	System Architect 4.0	89
3.1.5.	Rational Rose	93
3.2.	<u>Editores de Páginas Web</u>	97
3.2.1.	Med	97
3.2.2.	Macromedia Dreamweaver	97
3.2.3.	Frontpage	99
3.3.	<u>Sistemas de Gestión de Base de Datos</u>	100
3.3.1.	MySQL	100
3.3.2.	SQL Server 7.0	101
3.3.3.	Oracle	104
<b>Capítulo IV. Aplicación PIENSSA</b>		<b>108</b>
4.1.	<u>Descripción General</u>	109
4.2.	<u>Acerca de Aplicación</u>	110
4.2.1.	Diseño PIENSSA	111
4.2.1.1.	Diagrama de los Casos de Uso	111
4.2.1.2.	Descripción de los Casos de Uso	112
4.2.1.3.	Diagrama de Clases	113
4.2.1.4.	Diagramas de Actividades	114
4.2.1.4.1.	<i>Registro del Título</i>	114
4.2.1.4.2.	<i>Registro de Usuarios</i>	115
4.2.1.5.	Diagrama de Implementación	116
4.2.1.6.	Diseño de la Base de Datos	117
4.2.1.6.1.	<i>Modelo Conceptual</i>	117
4.2.1.6.2.	<i>Modelo Físico</i>	118

4.2.2.	Desarrollo de PIENSSA	119
4.2.2.1.	Creación de las Bases de Datos	119
4.2.2.2.	Programación de la Aplicación PIENSSA	120
4.3.	Funcionamiento de PIENSSA	120
4.3.1.	Mapa del Sitio	121
4.3.2.	Acceso al sitio PIENSSA	122
4.3.3.	Ingreso a PIENSSA	122
4.3.4.	Registro de Usuarios	123
4.3.5.	Menú de Trabajo	123
4.3.5.1.	<i>Generar un Portal Web Corporativo</i>	123
4.3.5.2.	<i>Ver un Portal Web Corporativo creado con anterioridad</i>	124
4.3.5.2.1.	Carga de Portales	124
4.3.6.	Diseño de la Estructura del Portal	125
4.3.6.1.	<i>Registro de Logotipos</i>	126
4.3.6.2.	<i>Registro de Título</i>	127
4.3.6.3.	<i>Registro de la Barra de Herramientas</i>	128
4.3.6.4.	<i>Registro de las Opciones de Menú</i>	129
4.3.6.5.	<i>Configuración del Área de Despliegue</i>	131
4.3.6.6.	<i>Registro de Avisos</i>	132
4.3.6.7.	<i>Registro de Noticias</i>	133
4.3.6.8.	<i>Registro de Anuncios</i>	134
4.3.7.	Menú de Administración de PIENSSA	135
4.3.7.1.	<i>Registro de Ocupaciones</i>	136
4.3.7.2.	<i>Registro de Formas de Enterarse</i>	136
4.3.7.3.	<i>Registro de Países</i>	137
4.3.7.4.	<i>Registro de Colores</i>	137
4.3.7.5.	<i>Registro de Fuentes</i>	138
4.3.8.	Contacte con Nosotros	138
4.3.9.	Datos de Creación	139
4.3.10.	Modificación de la Ruta de Destino	140
4.3.11.	Generación del Portal Web Corporativo y su Documentación.	140
	<b>Conclusiones.</b>	142
	<b>Glosario.</b>	146

<b>Bibliografía.</b>	159
<b>Referencias Electrónicas.</b>	164
<b>Anexos.</b>	167
Anexo A: Archivo PortalWeb.htm	168
Anexo B: Archivo Documenta.txt	171
Anexo C: Archivo VisiondelNegocio.txt	182

# PIENSSA



## *Introducción*

El presente trabajo de tesis detalla el proceso de diseño, desarrollo y funcionamiento de PIENSSA, una aplicación Web creada con la finalidad de permitir a los usuarios de la misma generar Portales Web Corporativos de manera fácil, rápida y dinámica, considerando aspectos de calidad en el desarrollo de los mismos y generando la documentación técnica que permita continuar su desarrollo de manera sistemática y normalizada, basándose en la metodología del Proceso Unificado de Rational (RUP).

Para entender lo anterior, debemos comenzar diciendo que el crecimiento de Internet (y las así llamadas Intranets y Extranets) y, en general el crecimiento de la Web (World Wide Web) está implicando un fuerte impacto y cambio paradigmático en el manejo de la información en sectores tan diversos como: comerciales, educativos, industriales, financieros, de entretenimiento, gubernamentales, etc. y porqué no decirlo también, poco a poco está cambiando en nuestras vidas personales el modo de ver y relacionarse con el mundo (en un modelo globalizado) [72].

Hablando un poco sobre la evolución de esta tecnología, se puede decir que Internet nace a mediados de la década de los setenta, bajo los auspicios de DARPA, la Agencia de Proyectos Avanzados para la Defensa de Estados Unidos, quien inició un programa de investigación de técnicas y tecnologías para unir diversas redes de conmutación de paquetes, permitiendo así a los ordenadores conectados a estas redes comunicarse entre sí de forma fácil y transparente.

De estos proyectos nació el protocolo de comunicación de datos IP o Internet Protocol, que permitía a ordenadores diversos comunicarse a través de una red formada por la interconexión de diversas redes.

A mediados de los ochenta, la Fundación Nacional para la Ciencia norteamericana, creó la red **NSFNET**, que se convirtió en el backbone (el troncal) de Internet junto con otras redes similares creadas por la NASA (**NSINet**) y el **U.S. DoE** (Department of Energy) con la **ESNET**. En Europa, la mayoría de países disponían de backbones nacionales (**NORDUNET**, **RedIRIS**, **SWITCH**, etc.) y de una serie de iniciativas europeas como **EARN y RARE**; en esta época, aparecen los primeros proveedores de acceso a Internet privados que ofrecen acceso pagado a Internet.

La WWW (World Wide Web) se ha convertido junto con el correo electrónico, en el principal caballo de batalla de Internet: ésta ha dejado de ser una inmensa biblioteca de páginas estáticas para convertirse en un servicio que permite acceder a multitud de prestaciones y funciones, así como, a infinidad de servicios, programas, tiendas, etc.

En estos momentos disponer de una dirección de correo electrónico, de acceso a la web, etc., ha dejado de ser una novedad para convertirse en algo normal en muchos países del mundo. Por eso las

empresas, instituciones, administraciones y demás están migrando rápidamente todos sus servicios, aplicaciones, tiendas, etc., a un entorno web que permita a sus clientes y usuarios acceder a todo ello por Internet. A pesar del ligero descenso experimentado en el ritmo de crecimiento, Internet está destinado a convertirse en una suerte de servicio universal de comunicaciones, permitiendo una comunicación universal.

## Antecedentes.

En 1989, mientras trabajaba en el CERN (Centro Europeo de Investigación Nuclear), Tim Berners-Lee empezó a diseñar un sistema para hacer accesible fácilmente la información del CERN. Dicho sistema empleaba el hipertexto para estructurar una red de enlaces entre los documentos. Una vez obtenida la aprobación para continuar el proyecto, nació el primer navegador web, llamado WorldWideWeb (sin espacios) [72].

A partir de esta época, gracias a la amplia disponibilidad de implementaciones de la *suite* de protocolos TCP/IP (formada por todos los protocolos de Internet y no sólo por TCP e IP), algunas de las cuales eran ya de código libre, Internet empezó lo que posteriormente se convertiría en una de sus características fundamentales, un ritmo de crecimiento exponencial, hasta que a mediados del 2002 empieza a descender ligeramente el ritmo de crecimiento.

A mediados de los noventa, se inició el *boom* de Internet; en esa época, el número de proveedores de acceso privado se disparó, permitiendo a millones de personas acceder a Internet, que a partir de ese momento ya se empezó a conocer como la Red, desbancado a las demás redes de comunicación existentes (**Compuserve**, **FidoNet/BBS**, etc.). El punto de inflexión vino marcado por la aparición de implementaciones de TCP/IP gratuitas (incluso de implementaciones que formaban parte del sistema operativo) así como por la popularización y abaratamiento de medios de acceso cada vez más rápidos (módems de mayor velocidad, **RDSI**, **ADSL**, cable, satélite), a medida que se conectaban más usuarios los costos se reducían, aparecían más proveedores e Internet se hacía más atractivo y económico.

En 1992 el sistema ya se había extendido fuera del CERN. El número de servidores “estables” había aumentado, alcanzando la sorprendente cifra de veintiséis.

En 1993 la web ya era merecedora de un espacio en el *New York Times*. Éste es el año del lanzamiento de **Mosaic**, un navegador para **X-Window/Unix** que con el tiempo se convertiría en Netscape y fue un factor clave de popularización de la web.

En 1994 se fundó el WWW Consortium, motor de desarrollo de los estándares predominantes en la web [43]. A partir de ese momento, el crecimiento ya fue constante, convirtiéndose hacia finales de los noventa en el servicio insignia de Internet y dando lugar al crecimiento imparable de los servicios en línea que estamos experimentado actualmente.

## **Planteamiento del problema.**

La existencia de diversas y variadas aplicaciones informáticas que permiten a los usuarios de Internet generar software para la web sin conocimientos previos, ha provocado la proliferación de páginas que no cumplen en muchos casos con requisitos mínimos de calidad ni permiten darles un seguimiento adecuado ya que no proporcionan la documentación técnica que soporte su creación y tampoco facilitan su mantenimiento.

Pero, si hablamos de portales web el problema es más grave ya que para crearlos los usuarios generan páginas sin ton ni son, es decir, sin basarse en ninguna metodología web existente, lo cual provoca que en la mayoría de los casos el resultado no sea lo esperado, ni en apariencia, ni en funcionalidad y efectividad.

La falta de aplicaciones de uso común, de fácil acceso y de sencilla utilización para generar Portales Web Corporativos u orientados a las empresas, basándose en una metodología web e incluyendo su documentación técnica, es el motivo que nos impulsa a desarrollar la aplicación "PIENSSA", la cual pretende subsanar esta deficiencia y aunque esta primera versión cubre aspectos básicos, significa un buen comienzo.

# Objetivos.

## 1. General.

- Diseñar, desarrollar e implementar una Aplicación de Software que permita generar un "Portal Web Corporativo (PWC)", incluyendo su documentación como parte del producto.

Dicha herramienta servirá como una guía practica para que cada usuario pueda diseñar un Portal Web Corporativo, tomando en cuenta los productos y/o servicios que ofrecen dentro de su compañía o negocio.

Los usuarios deberán registrarse para hacer uso de dicha herramienta web, la información de cada usuario se almacenará en el Servidor Web donde la aplicación resida y en el mismo equipo se encontrará el servidor de la Base de Datos

La aplicación tendrá un estándar de 1024x768 píxeles con configuración predeterminada a color y será hospedada en un Servidor Web que nosotros administraremos (al menos en lo que respecta a PIENSSA).

La herramienta será un sitio en Internet que permita utilizar la aplicación de manera gratuita, dicha aplicación no se podrá descargar; el código que generará será HTML y/o DHTML y esto es lo que proveerá al usuario, así como la documentación de cómo fue creado el sitio; se mostrará de inicio una plantilla estándar, la cual podrá ser modificada para crear el Portal Web Corporativo; las noticias podrán estar en constante rotación.

El sitio o aplicación tendrá su archivo Help en línea y la seguridad de esta herramienta dependerá de los perfiles de administrador y de usuario existentes.

## 2. Específicos.

- Utilizar los Estándares de Calidad requeridos por la Ingeniería del Software para el Desarrollo de Portales Web.

Para el desarrollo de esta herramienta se considerarán las normas ISO 9126, ISO 14598, IEEE 1061, Modelo de Calidad para Web.

- Analizar y seleccionar la Metodología adecuada para Desarrollar la aplicación y el Portal Web Corporativo.

Considerando de manera inicial la metodología RUP con la herramienta UML para el desarrollo de este proyecto.

- Crear un "Portal Web Corporativo" con calidad, a partir de la herramienta de software desarrollada, individualizada para cada usuario.

PIENSSA únicamente se encargara del portal Web Corporativo en su página de inicio: el portal como Producto Final deberá cubrir aspectos mínimos necesarios de tipo empresarial. Una vez creado el portal, el usuario podrá modificarlo, la aplicación deberá permitir dar continuidad al desarrollo de PWC para que con otras herramientas sea terminado si el usuario lo desea.

- Facilitar el acceso a las plantillas RUP, como complemento de la documentación del Portal.

Parte del producto final obtenido al utilizar PIENSSA serán algunas plantillas RUP que permitan ser la base de la documentación técnica para la continuación del proyecto por parte del usuario.

## Justificación.

Inicialmente la web era simplemente una colección de **páginas estáticas**, documentos, etc., que podían consultarse o descargarse: el siguiente paso en su evolución fue la inclusión de un método para confeccionar **páginas dinámicas** que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición) [37], dicho método fue conocido como **CGI** (*common gateway interfase*) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores web los soportan debido a su sencillez, además, proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos [72].

El esquema de funcionamiento de los CGI tenía un punto débil: cada vez que recibíamos una petición, el servidor web lanzaba un proceso que ejecutaba el programa CGI [71]. Como la mayoría de CGI estaban escritos en lenguaje interpretado (Perl, Python, etc.), o en lenguaje que requería *run-time environment* (VisualBasic, Java, etc.), implicando una gran carga para la máquina del servidor, además, si la web tenía muchos accesos al CGI, esto suponía problemas graves.

Por ello se empiezan a desarrollar alternativas a los CGI para solucionar este grave problema de rendimiento, las soluciones se generan principalmente por dos vías; por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, evitando instanciar y ejecutar multitud de programas, la otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) el cual incluye las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de este momento, se vive una explosión del número de arquitecturas y lenguajes de programación que permiten desarrollar aplicaciones web, todas ellas siguen alguna de las dos vías ya mencionadas; de ellas, las más útiles y que más se utilizan son las que permiten mezclar los dos sistemas, es decir, un lenguaje de programación integrado que permita al servidor interpretar comandos que se "incrusten" en las páginas HTML y otro sistema de ejecución de programas más enlazado con el servidor que no presente los problemas de rendimiento de los CGI.

Al diseñar y desarrollar aplicaciones web se cumple el viejo dicho de "bueno, bonito y barato", interpretado ahora como "funcional, estético y navegable", es difícil encontrar sitios web que cumplan los tres requisitos del diseño [72].

Un sitio web funcional debe cumplir los objetivos prácticos para los que fue construido; un sitio estético es aquel cuya estética es particularmente bonita y un sitio navegable permite al usuario emplear poco esfuerzo en llegar a la información que desea encontrar.

Generalmente se tendrá que elegir entre dos de los tres requisitos, dejando el tercero un poco más descuidado, creando así unas páginas muy bonitas, con animaciones en Flash y un diseño a base de capas y a la vez el sitio web resultante puede ser navegado por el usuario en unos pocos clicks de ratón, siendo de esta forma muy difícil que cumpla los requisitos funcionales exigidos.

Se puede crear un sitio web totalmente funcional y operativo donde los procesos de intercambio con las bases de datos sean fácilmente implementables y además sea muy fácil de navegar, pero entonces, seguramente se estará muy limitado a la hora de expresar en sus páginas las inquietudes estéticas y de diseño.

Por esta razón, para el caso de sitios web destinados a comercio o banca electrónica, o en aquellos que deban soportar complejos procesos de intercambio con bases de datos el orden a seguir será: funcional, navegable y estético. El objetivo principal de estos sitios será facilitar las consultas preliminares al usuario y las transacciones de datos posteriores, sin posibilidad de fallos de ningún tipo.

A estas páginas accede todo tipo de público, que no necesariamente debe tener grandes conocimientos informáticos ni de Internet, por lo cual debe facilitar lo máximo posible su navegación por las diferentes páginas del sitio, como regla general debe acceder a la información deseada con tres clicks de ratón como máximo.

Con estas limitaciones es a veces difícil crear un diseño vanguardista o innovador, ya que además se debe emplear en estos casos colores corporativos de la empresa cliente, las animaciones DHTML y Flash no están bien vistas por estos, etc., una vez cumplidos los objetivos de funcionalidad y navegabilidad, se debe ajustar el diseño lo más posible al estándar del tipo de aplicación desarrollada, criterios como estos atribuyen la calidad del producto.

Si el sitio web pertenece a una empresa que desea ofrecer un catálogo de sus productos, pero sin posibilidad de compra on-line ó a una organización (universidad, organismo oficial, etc.) con miras a ofrecer la posibilidad al navegante de poder consultar documentos de cualquier tipo, entonces la guía a seguir es la navegabilidad.

Es aquí donde el usuario desea encontrar lo antes posible el producto o documento que busca, estando generalmente el aspecto de programación web limitado a una simple consulta a una o más bases de datos; se debe diseñar el sitio web de tal forma que desde el principio el usuario identifique en cada

momento dónde se encuentra, cómo volver al inicio y cómo filtrar los datos para acceder rápidamente a la información que busca, y si las páginas tienen como objetivo el que el visitante pueda imprimirlas para su consulta off-line, deberán poseer un diseño que contemple este aspecto, estructurando su contenido de forma documental.

En estos casos y siempre que se cumpla esta primera condición, se dispone de más libertad en el diseño, pudiendo ofrecer una imagen más innovadora, aunque siempre dentro de los estándares que impongan la lógica de este tipo de sitios y el propio cliente.

Por último si el sitio web es destinados para dar a conocer una nueva empresa en general, un grupo musical o de teatro, una asociación de cualquier tipo, etc., deberán ser ante todo creativos en su diseño y estética, con el objetivo de captar desde el principio la atención del visitante y mantenerlo enganchado a las páginas y que explore todas ellas; este será pues el objetivo principal a conseguir, seguido de la navegabilidad, para que identifique siempre en qué lugar del sitio se encuentra.

Todo esto es lo que hoy en día consideran muchos de los diseñadores de portales web ya que cada día las diferentes organizaciones y empresas buscan poner en Internet sus productos y/o servicios para que puedan ser consultados por miles de navegadores.

Particularmente, en estos días se asiste a un crecimiento paulatino de sitios Web con funcionalidad de comercio electrónico y hasta vale mencionar, que se han realizado experimentos de cómo personas "aisladas" pueden proveerse de bienes y servicios, e intentan valerse y sobrevivir (con algún éxito), itan sólo con una computadora conectada a Internet ante sus manos! lo cierto es que una amplia gama de aplicaciones distribuidas cada vez más complejas está emergiendo en el entorno de la Web. El indiscutido incremento de popularidad y aceptación de las aplicaciones centradas en la Web tiene su origen en sus propias características y naturaleza: provee un modelo de información que soporta enlazar todo tipo de contenido y medios; permite a los usuarios un fácil acceso a información a partir de una representación ubicua de la misma, a diferencia de un documento en papel, tiene el valor agregado de la funcionalidad de software y, entre otros aspectos, permite a los usuarios (no necesariamente profesionales del área de informática) crear y enlazar fácilmente sus propios documentos por medio de herramientas ampliamente disponibles [72].

En la actualidad, la mayoría de los portales en Internet no cumplen con los requisitos mínimos de calidad que permite interactuar adecuadamente con ellos y muchos menos garantizan la integridad de su contenido y el objetivo para el cual fueron creados.

Regularmente, el desarrollo de este tipo de software de aplicación se realiza sin apearse a metodologías adecuadas ni confiables, lo que implica graves complicaciones desde el comienzo de su creación.

Por ello se pretende crear una aplicación de software que cumpla con los requisitos de ser: funcional, eficiente y eficaz; esta aplicación debe proporcionar una mejora social utilizando Tecnologías de Información de punta que brinden satisfacción y beneficio a los usuarios de Internet, además de crear un Portal Web atendiendo a las características como indican las Normas Internacionales

# Capítulo I



## *Metodologías, Lenguajes y Servidores*

Este capítulo trata de las diferentes metodologías, lenguajes de programación y servidores de Internet que existen para el diseño, desarrollo y funcionamiento de aplicaciones Web.

Los conceptos aquí explicados nos permitirán definir un panorama real del ciclo de vida en el desarrollo de PIENSSA.

## **1.1. Metodologías**

Existen en la actualidad, tecnologías que permiten un rápido desarrollo de aplicaciones poco reusables y difíciles de mantener, tales como las metodologías que permiten obtener aplicaciones mediante un proceso de desarrollo en capas, aprovechando al máximo la potencia de la programación orientada a objetos.

Una metodología de ingeniería de software es un proceso para desarrollar software de forma organizada, empleando una colección de técnicas y consideraciones de notaciones predefinidas, por lo que la misma se puede presentar como una serie de pasos, técnicas y notaciones asociadas a cada etapa construyendo un modelo que corresponde al dominio de una aplicación, al cual se le añaden detalles de implementación durante el diseño del sistema [71].

Las metodologías pretenden evitar:

- Desarrollo de producto erróneo.
- Inferioridad de cualidades.
- Superación del tiempo establecido.

A continuación se describirán un conjunto de metodologías Estructuradas y Orientadas a Objetos encontradas para el análisis y diseño de sitios Web

### **1.1.1. Metodologías Estructuradas**

#### **1.1.1.1. Metodología RNA**

RNA (Relationship-Navegational Analysis<sup>1</sup>), plantea una secuencia de pasos centrándose fundamentalmente en el flujo de trabajo de análisis; el proceso de trabajo que presenta RNA se basa en la realización de las siguientes fases:

- Análisis del entorno o análisis de audiencia.

- Análisis de Elementos de interés.
- Análisis de relaciones y del conocimiento.
- Análisis de la navegación.
- Implementación del análisis.

La propuesta de RNA es quizás una de las que más ha resaltado la necesidad de trabajar con la especificación de requisitos, incluyendo tareas como el análisis del entorno y de los elementos de interés, además resulta interesante, pues plantea la necesidad de analizar los requisitos conceptuales de manera independiente a los navegacionales.

#### **1.1.1.2. Metodología HDM**

HDM (Hypertext Design Model<sup>2</sup>) esta metodología se extiende del modelo entidad-relación, una aplicación se especifica dentro HDM mediante su estructura (hiperbase), su navegación (esquema de acceso) y su presentación (esquema de presentación), permite una descripción de manera global de la aplicación tanto a nivel navegacional como a nivel de presentación; esta metodología es adecuada para las aplicaciones en la que la regularidad, organización, modularidad y consistencia son factores importantes, permitiendo la construcción de aplicaciones de documentación técnica, educación, sistemas de auditoria para grandes aplicaciones de entidades corporativas [74].

#### **1.1.1.3. Metodología RMM**

RMM (Relationship Management Methodology <sup>3</sup>) es una metodología encamida al diseño estructurado y construcción de sistemas hipermedia, para facilitar el diseño de sitios Web y su relación con bases de datos y su integración con grandes sistemas de información, permite obtener una separación de datos, estructura e interfaz de usuario, se centra en las fases de diseño, no aborda todo el ciclo de vida del proyecto como la dirección del proyecto, estudios de viabilidad, análisis de requisitos, planificación, evaluación, mantenimiento. Este modelo proporciona un lenguaje que permite describir el modelo de navegación así como los objetos de información.

Comprende las siguientes fases [74]:

- Diseño E-R.
- Diseño de slices.
- Diseño de navegación.

---

<sup>1</sup> Análisis Relacional-Navegacional

<sup>2</sup> Modelo del Diseño de Hipertexto

- Diseño del protocolo de conversión.
- Diseño de la interfaz de usuario.
- Comportamiento en tiempo de ejecución.
- Construcción.
- Evaluación.

#### **1.1.1.4. Metodología ERMIA**

ERMIA (Entity-Relationship Modeling of Information Artifacts<sup>4</sup>) no se trata de una metodología sino más bien de una extensión del modelo Entidad-Relación para conseguir un diagrama de modelado de interacción entre el usuario y los artefactos de información, las principales características de este modelo son:

- Proporciona soporte para evaluación de costos particularmente en lo que se refiere a métricas para buscar o modificar estructuras.
- Toma en cuenta la estructura interna de las entidades permitiendo que el proceso del modelado sea independiente de los detalles de implementación, cada estructura interna tiene requisitos de tiempo y memoria.
- Permite la distinción entre la información conceptual y su representación. Se pide aplicar a las fases iniciales del diseño permitiendo evaluar interfaces, comparar modelos mentales y por lo tanto reflejar la descripción conceptual como aspectos de percepción del sistema [74].

#### **1.1.1.5. Metodología Lowe-Hall's Engineering Approach**

Metodología (Metodología Lowe-Hall's Engineering Approach<sup>5</sup>) que trata de cubrir todo el ciclo de desarrollo, esta metodología consta de las siguientes fases [74]:

- Fase análisis del dominio: se fijan los tipos de información, estructura, tipos de usuarios a los que se dirige la aplicación (responsabilidades y papel que desempeñan), aplicación actual en uso (si la hay), objetivos del proyecto, limitaciones de la tecnología usada y estudio de factibilidad de costos.
- Fase de selección del modelo del producto: aquí se establecen aspectos como la información esencial de la aplicación, la escritura, el comportamiento y funcionalidad de la aplicación,

---

<sup>3</sup> Metodología de la Administración de Relaciones

<sup>4</sup> Modelado de la Entidad-Relación de los artefactos de información

<sup>5</sup> Metodología de Aproximación a la Ingeniería Lowe-Hall's

además de que propone la elaboración de tres modelos, uno basado en el lenguaje de programación, o en base a pantallas y otro centrado en la información.

- Fase de selección del modelo de proceso: consiste en seleccionar las estrategias adecuadas para solucionar la integración global del sistema.
- Fase de plan de proyecto: consiste en organizar, coordinar y guiar las acciones del equipo de desarrollo, planificar cada uno de los requisitos para la construcción del sitio.
- Fase de mantenimiento: trata de planificar la corrección de errores o la decisión de incorporar nuevas funcionalidades.
- Fase de documentación: consiste en verificar que toda actividad realizada y que intervino en la construcción del sistema debe encontrarse en documentos que permitan su posterior consulta.

## **1.1.2. Metodologías Orientadas a Objetos**

### **1.1.2.1. Metodología WSDM**

WSDM (Web Site Design Method<sup>6</sup>) es una propuesta para el desarrollo de sitios web, en la que el sistema se define en base a los grupos de usuarios, su proceso de desarrollo se divide en cuatro fases:

- Modelo de usuario.
- Diseño conceptual.
- Diseño de la implementación.
- Implementación.

La fase de modelo de usuario intenta detectar los perfiles de usuarios para los cuales se construye la aplicación. En esta fase se realizan dos tareas:

- Clasificación de usuarios: en este paso se deben identificar y clasificar a los usuarios que van a hacer uso del sistema, para ello, WSDM propone el estudio del entorno de la organización donde se vaya a implantar el sistema y los procesos que se vayan a generar, describiendo las relaciones entre usuarios y actividades que realizan estos usuarios, para la representación gráfica de estas relaciones WSDM propone una especie de mapas de conceptos de roles y actividades.

---

<sup>6</sup> Método de Diseño de Sitio Web

- Descripción de los grupos de usuarios: en esta segunda etapa se describen con más detalles los grupos de usuarios detectados en la etapa anterior. Para ello, se debe elaborar un diccionario de datos, en principio con formato libre, en el que indican los requisitos de almacenamiento de información, requisitos funcionales y de seguridad para cada grupo de usuarios.

El resto de las fases del proceso de WSDM se hacen en base a la clasificación de usuarios que se realiza en esta primera etapa.

### **1.1.2.2. Metodología SOHDM**

SOHM (Scenario-based Object-Oriented Hypermedia Design Methodology<sup>7</sup>) presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema, proponiendo el uso de escenarios.

El proceso de definición de requisitos parte de la realización de un diagrama de contexto tal y como se propone en diagramas de flujos de datos (DFD) de Yourdon, en este diagrama de contexto se identifican las entidades externas que se comunican con el sistema, así como los eventos que provocan esa comunicación. La lista de eventos es una tabla que indica en qué eventos puede participar cada entidad, por cada evento diferente SOHDM propone elaborar un escenario; éstos son representados gráficamente mediante los denominados SACs (Scenario Activity Chart), cada escenario describe el proceso de interacción entre el usuario y el sistema cuando se produce un evento determinado especificando el flujo de actividades, los objetos involucrados y las transacciones realizadas.

SOHDM propone un proceso para conseguir a partir de estos escenarios el modelo conceptual del sistema que es representado mediante un diagrama de clases, el proceso de SOHDM continua reagrupando estas clases para conseguir un modelo de clases navegacionales del sistema.

### **1.1.2.3. Metodología UWE**

UWE (UML-Based Web Engineering<sup>8</sup>) es una propuesta metodológica basada en el Proceso Unificado y UML (Lenguaje de Modelado Unificado<sup>9</sup>) para el desarrollo de aplicaciones web. UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas (adaptativas).

---

<sup>7</sup> Metodología de Diseño de Hipermedia orientada a Objetos basados en Escenarios.

<sup>8</sup> Ingeniería Web basada en el Lenguaje de Modelado Unificado

<sup>9</sup> Lenguaje de Modelado Unificado

Esta metodología distingue entre la tarea de obtener requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz. UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales. Los requisitos funcionales tratados por UWE son:

- Requisitos relacionados con el contenido.
- Requisitos relacionados con la estructura.
- Requisitos relacionados con la presentación.
- Requisitos relacionados con la adaptación.
- Requisitos relacionados con los usuarios.

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un sistema web las entrevistas, los cuestionarios y los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos; para la validación propone walk-throughs, auditorias y prototipos.

#### **1.1.2.4. Metodología W2000**

W2000 supone una propuesta que amplía la notación de UML con conceptos para modelar elementos de multimedia heredados de la propuesta HDM (Hypermedia Design Model<sup>10</sup>). El proceso de desarrollo de W2000 se divide en tres etapas:

- Análisis de requisitos.
- Diseño de hipermedia.
- Diseño funcional.

La especificación de requisitos en W2000 se divide en dos subactividades:

- Análisis de requisitos funcionales.
- Análisis de requisitos navegacionales.

La especificación de requisitos comienza haciendo un estudio de los diferentes roles de usuario que van a interactuar con el sistema. Cada actor potencialmente distinto tendrá su modelo de requisitos de navegación y de requisitos funcionales; el modelo de requisitos funcionales es representado como un modelo de casos de uso tal y como se propone en UML, en él se representa la funcionalidad principal asociada a cada rol y las interacciones que se producen entre el sistema y cada rol. El segundo modelo consiste en otro diagrama de casos de uso pero que no representa funcionalidad sino posibilidades de navegación de cada actor, la representación gráfica es realizada con una extensión de UML.

---

<sup>10</sup> Modelo de Diseño de Hipermedia

### **1.1.2.5. Metodología UWA**

UWA (Ubiquitous Web Applications<sup>11</sup>) ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA, sin embargo, W2000 ha sido incluida en UWA sólo en la fase de diseño hipermedia, siendo ambas propuestas diferentes en la fase de definición de requisitos.

El proceso de captura de requisitos en UWA comienza definiendo los diferentes roles de usuario que pueden interactuar con el sistema, los objetivos globales de los sistemas y la relación entre éstos; el proceso continúa haciendo un refinamiento de esos objetivos globales, concretándolos en subobjetivos. Estos subobjetivos son estudiados y refinados para detectar conflictos entre ellos, de esta forma, se concretizan aún más dividiéndolos en requisitos.

Los requisitos son clasificados en varios tipos: de contenido, de estructura de contenido, de acceso, de navegación, de presentación, de operaciones de usuario y de operaciones del sistema.; de esta forma, los requisitos se van refinando hasta que solo pertenezcan a uno de estos grupos y finalmente los requerimientos son asignados a artefactos de diseño o a reglas de customización. Para definir los objetivos, UWA propone una notación propia, basada en una plantilla, la definición de los actores y la relación con los objetivos se hace usando un diagrama basado en casos de uso.

Por último, para definir y refinar los subobjetivos y los requisitos, utilizan una notación gráfica propia que denominan Grafo de Refinamiento de Objetivos, el refinamiento de este grafo permite ir representando la relación entre los requisitos y hacer un seguimiento para validar la consecución de los objetivos del sistema, una vez que los requisitos son detectados, hacen uso de XML para definirlos de una manera formal.

---

<sup>11</sup> Aplicaciones Web Obicuas

### **1.1.2.6. Metodología NDT**

NDT (Navigational Development Techniques<sup>12</sup>) es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones web, el flujo de especificación de requisitos de NDT comienza con la fase de captura de requisitos y estudio del entorno; para ello, plantea el uso de técnicas como las entrevistas o el brainstorming y JAD, tras esta fase, se propone la definición de los objetivos del sistema.

En base a estos objetivos, el proceso continúa definiendo los requisitos que el sistema debe cumplir para cubrir los objetivos marcados, NDT clasifica los requisitos en:

- Requisitos de almacenamiento de información.
- Requisitos de actores.
- Requisitos funcionales.

Requisitos de interacción, representados mediante:

- Frases, que recogen cómo se va a recuperar la información del sistema utilizando un lenguaje especial denominado BNL (Bounded Natural Language).
- Prototipos de visualización, que representan la navegación del sistema, la visualización de los datos y la interacción con el usuario.

Para los requisitos no funcionales se dice que todo el proceso de definición y captura de requisitos y objetivos que propone NDT se basa principalmente en plantillas o patrones, pero también hace uso de otras técnicas de definición de requisitos como son los casos de uso y los glosarios. La propuesta ofrece una plantilla para cada tipo de requisito, lo que permite describir los requisitos y objetivos de una forma estructurada; algunos de los campos de los patrones son cerrados, es decir, solo pueden tomar valores predeterminados, estos campos permiten que en el resto del proceso del ciclo de vida de NDT se puedan conseguir resultados de forma sistemática, el flujo de trabajo de especificación de requisitos termina proponiendo la revisión del catálogo de requisitos y el desarrollo de una matriz de trazabilidad que permite evaluar si todos los objetivos han sido cubiertos en la especificación.

La propuesta viene acompañada de una herramienta case, NDT-Tool, que facilita el cumplimiento de los patrones y que permite automatizar el proceso de consecución de resultados.

---

<sup>12</sup> Técnicas de Desarrollo Navegacional

### **1.1.2.7. Metodología *DESING-DRIVEN REQUERIMENTS ELICITATION***

Metodología (DESING-DRIVEN REQUERIMENTS ELICITATION<sup>13</sup>) es parte del proceso del manejo de diseño que proponen Lowe y Eklund para el desarrollo de aplicaciones en el entorno Web, consiste en realizar la captura, definición y validación de requisitos durante el proceso de diseño, esto hace necesario que las actividades de diseño sean realizadas de modo que los requerimientos pueden ser tratados y administrados.

El proceso se basa en el uso de prototipos para ayudar al cliente en la exploración de las posibles soluciones y de los problemas que tienen que ser resueltos, los usuarios o clientes definen sus requerimientos basándose en la observación o trabajo con estos prototipos.

Es un proceso iterativo que consiste en reducir la incertidumbre del cliente. El ciclo tiene tres fases:

- Evaluación.
- Especificación.
- Construcción.

Este proceso fue definido sobre la base de un exhaustivo análisis de "las mejores prácticas" en el desarrollo de aplicaciones comerciales para el entorno Web. Esta metodología trata a todos los requisitos de la misma manera; estos requisitos son: de contenido, de protocolo de interfaces, de estructura navegacional, de "ver y sentir", de representación interna de datos, de versionamiento, de control de cambios, de seguridad, de gestión de contenido, de acceso de control, de eficiencia, de monitoreo del usuario, de soporte de funcionalidad, de adaptación del sistema, de identificación del usuario y sus derechos de acceso, etc. En las tablas comparativas se usa la abreviatura DDDP para Design-Driven Development Process (los autores no usan esta forma abreviada).

### **1.1.2.8. Metodología *OOHDM***

OOHDM (Object Oriented Hypermedia Design Model<sup>14</sup>) es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones de la web. En sus comienzos no contemplaba la fase de captura y definición de requisitos, pero actualmente propone el uso de Diagramas de Interacción con el Usuario (UIDs). Esta propuesta parte de los casos de uso que considera una técnica muy difundida, ampliamente aceptada y fácilmente entendible por los usuarios y clientes no expertos, pero que resulta ambigua para el equipo de desarrollo en fases posteriores del ciclo de vida. Igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos web, teniendo un claro y

---

<sup>13</sup> Diseño del Manejo de Necesidades.

amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema.

Partiendo de estas dos premisas, OOHDM propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos los analistas elaboran los UIDs. Estos UIDs son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz. El proceso de transformación de un caso de uso a un UIDs es descrito detalladamente en la propuesta y se basa en detectar la interacción necesaria para la realización del caso de uso. OOHDM centra el desarrollo de un sistema de información web entorno del modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema, pero los casos de uso resultan demasiado ambiguos para ello. Así, propone refinar el proceso de desarrollo descrito en UML de forma que de los casos de uso se generen los UIDs que concreten más la definición de los requisitos para, a partir de ellos, la descripción de la representación de los SACs. En algunos de los primeros trabajos OOHDM propone la descripción de escenarios en forma textual y gráfica para cada tipo de usuario como etapa previa al diseño de la navegación.

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación [71].

### **a) Diseño Conceptual**

Durante esta actividad se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. En las aplicaciones hipermedia convencionales, cuyos componentes de hipermedia no son modificados durante la ejecución, se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intenten ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos.

En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

---

<sup>14</sup> Modelo de Diseño de Hipermedia Orientado a Objetos

Se usa notación similar a UML (Unified Modeling Language<sup>15</sup>) y tarjetas de clases y relaciones similares a las tarjetas CRC (Class-Responsibility-Collaboration<sup>16</sup>). El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos y las relaciones que son usadas para construir enlaces.

## **b) Diseño Navegacional**

La primera generación de aplicaciones *web* fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermedia. En OOHDM, la navegación es considerada un paso crítico en el diseño aplicaciones; un modelo navegacional es construido como una *vista* sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual. El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales. En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos.

La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos y otros contextos navegacionales (contextos anidados). Pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros. Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.

Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un particular contexto.

## **c) Diseño de Interfaz Abstracta**

Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz, esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué

---

<sup>15</sup> Lenguaje de Modelado Unificado

<sup>16</sup> Clase-Responsabilidad-Colaboración

transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas; una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones *web*) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar. En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia.

El modelo de interfaz ADVs (Abstract Data View<sup>17</sup>) [4] especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación.

#### **d) Implementación**

En esta fase, el diseñador debe implementar el diseño; hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación. Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema, debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno *web*, el diseñador debe decidir además qué información debe ser almacenada.

#### **1.1.2.9. Metodología HFPM**

La propuesta de HFPM (Hypermedia Flexible Process Modeling<sup>18</sup>) describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software.

HFPM propone las siguientes fases:

- Descripción breve del problema.
- Descripción de los requisitos funcionales mediante casos de uso.
- Modelar la interfaz de usuario.
- Modelar los requisitos no funcionales.

---

<sup>17</sup> Vista de Datos Abstracta

<sup>18</sup> Modelado de Proceso Flexible de Hipermedia

### **1.1.2.10. Metodología MacWeb Approach<sup>19</sup>**

Es un entorno de diseño y desarrollo que soporta la evolución de la especificación, un crecimiento incremental de la estructura, prototipos y desarrollo de la aplicación, proporciona un modelo orientado a objetos flexible en el cual se encapsula conocimiento y permite que un medio de almacenamiento clásico se comporte como un mecanismo de script, además de proporcionar ayuda a los diseñadores para crear redes semánticas y proporciona un conjunto de primitivas.

Comprende las siguientes fases [74]:

- Obtención de conceptos.
- Generación de material.
- Modelo navegacional.
- Organización y estructura.
- Inteface abstracta.
- Modelo de implementación.
- Reorganización y Reestructura.
- Test, Evaluación.

### **1.1.2.11. Metodología EORM**

EORM (Enhanced Object-Relationship Model<sup>20</sup>) propone tres fases para desarrollar un hiper-documento:

- Análisis Orientado a Objetos.
- Diseño Conceptual.
- Construcción.

### **1.1.2.12. Metodología OO/PATERN APPROACH**

Metodología (OO/Patern Approach<sup>21</sup>) Se basa en la utilización de patrones para permitir la reutilización de la navegación, interacción y estructura de las interfaces y el diseñador cree una estructura confortable y sin sobrecarga de conocimiento. Los patrones de diseño para dar soporte a la construcción de la estructura del sistema serian los siguientes [74]:

- Patones de diseño navegacional: sugiere como organizar la navegación de manera que resulte clara y entendible para el usuario.

---

<sup>19</sup> Aproximación a la Web de Macintosh

<sup>20</sup> Modelo Mejorado de Objeto Relacionados.

<sup>21</sup> Aproximación de Patrones orientados a Objetos.

- Patrones de interfaz: son guías para el diseño de GUI centrándose en como presentar la información que no puede ser mostrada de manera integral en toda la pantalla, es recomendable mostrar únicamente un subconjunto de la información permitiendo al usuario obtener el resto a través de objetos o botones.
- Patrones separadores de información-interacción: permiten al diseñador diferenciar controles y contenidos en las interfaces, esto se soluciona a través de la separación de los canales de entrada-salida, otro patrón es agrupar todos los tipos de controles en diferentes partes de la pantalla.
- Patrones de anticipación de comportamiento: proporciona al usuario como entender la respuesta obtenida al hacer uso de los objetos de la interfaz, informándole del estado de la interacción. Tiene como objetivo que el usuario se desespere y conozca el estado del sistema y active la misma opción varias veces.
- Patrones de noticias: contemplan la necesidad de comunicar a los usuarios nueva información del sitio si la hay.

Las fases que implementan son:

- Análisis de casos de uso.
- Diseño conceptual.
- Diseño de colaboración.
- Definición de clases.
- Diseño navegacional basado en patrones.
- Implementación.

### **1.1.2.13. Metodología AHDM**

AHDM (Adaptative Hypermedia Systems Development<sup>22</sup>) presenta un planteamiento que se centra en la concentración de sistemas hipermedia adaptativos, mediante un enfoque orientado a objetos tomando como base el desarrollo de sistemas clásicos: análisis, diseño, implementación y test.

Incluye aspectos adicionales como planificación, análisis de riesgos y costos; se basa en la utilización de modelos, notaciones y técnicas como UML ya utilizadas en la ingeniería del software, como casos de uso o diagramas de estados; realiza una separación entre navegación, contenido y presentación.

---

<sup>22</sup> Desarrollo de Sistemas Adaptado a Hipermedia

Las fases que propone son [74]:

- Estudio de viabilidad.
- Análisis de usuarios.
- Análisis de requisitos.
- Análisis de estrategia.
- Diseño de modelo de usuario.
- Diseño del modelo conceptual.
- Diseño navegacional.
- Diseño de interface abstracta.
- Diseño del modelo de dialogo.
- Diseño de implementación.
- Diseño de test de calidad.
- Mantenimiento.

#### **1.1.2.14. Metodología SHDT**

SHDT (Structured Hypermedia Design: Technique / W3DT: Word Wide Web Design Technique <sup>23</sup>) permite modelar la información estructurada y la no estructurada de los sitios Web, esta metodología se ayuda en una herramienta de diseño Web llamada W3DT que permite la representación gráfica de la metodología y genera páginas HTML y scripts CGI's en la fase de diseño.

Comprende las siguientes fases [74]:

- Estructura de la información.
- Diseño navegacional.
- Diseño de la interfaz.

#### **1.1.2.15. Metodología IHDM**

IHDM (Indexed Driven Hypermedia Design Methodology <sup>24</sup>) se enfoca en sistemas de hipermedia que están centradas en el contenido, se ajusta en aplicaciones que ofrecen servicios de marketing o de información, utiliza nodos índices como entradas a los contenidos hipermedia y soporta dos tipos de enlaces: estructurales y de referencia [74].

---

<sup>23</sup> Diseño Estructurado de Hipermedia: Técnica / W3DT: Técnica de Diseño Word Wide Web

<sup>24</sup> Metodología de Diseño de Hipermedia Manejando Indices

### **1.1.2.16. Metodología RUP**

El RUP (Rational Unified Process<sup>25</sup>) es un proceso lógico y disciplinado que nos permite asignar tareas y responsabilidades en una organización de desarrollo, para producir software de alta calidad que resuelva las necesidades de los usuarios en tiempos (plazo) y presupuestos fiables.

#### Un proceso de desarrollo de software debe de:

- ❑ Definir los pasos que guían los líderes y quién es responsable de ellos.
- ❑ Ayudar a la administración del proyecto en la planeación y medición del progreso.
- ❑ Ayudar a controlar el proyecto y reducir la confusión.
- ❑ Reducir el Riesgo
- ❑ Hacer el desarrollo de software predecible, repetible y medible.

Su objetivo es asegurar la producción de un software de alta calidad que introduzca las necesidades de usuarios finales tomando en cuenta un programa de trabajo predecible.

La Fig. 1 muestra las dos dimensiones que constituyen a la Metodología RUP, en la que:

- El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida de los procesos.
- El eje vertical representa las disciplinas, donde las actividades grupales actúan lógicamente por naturaleza.

La primera dimensión representa el aspecto dinámico del proceso y este es expresado en términos de fases e iteraciones.

La segunda dimensión representa el aspecto estático del proceso: como es descrito en términos de los componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

---

<sup>25</sup> Proceso Unificado de Rational.

**Fig. 1** Arquitectura RUP.

A continuación se explican con más detalle cada una de las fases y disciplinas de las que esta constituida la arquitectura RUP.

**A. Fases.**

El RUP tiene diversas fases vitales las cuales se describen a continuación:

**a) Inicio.** Realiza un estudio del proyecto para definir los alcances, limitaciones, costo, tiempo, preparación del ambiente para el proyecto y mostrar una arquitectura candidata del proyecto.

**b) Elaboración.** Es la base del diseño y del esfuerzo de construcción; la estabilidad de la arquitectura se evalúa a través de uno o más prototipos. Crea un caso de desarrollo, crea plantillas, pautas y las herramientas de creación.

**c) Construcción.** Clarifica los requisitos restantes y termina el desarrollo del sistema basado en la arquitectura base, es en el proceso de fabricación, donde el énfasis se pone en recursos de manejo y operaciones, que controlan para optimizar costo, horario y calidad.

**d) Transición.** Es asegurarse de que el software está disponible para sus usuarios del extremo, esta fase puede atravesar varias iteraciones, e incluye la prueba del producto en la preparación para el lanzamiento y la fabricación de los ajustes de menor importancia basados en la regeneración de usuario. A este punto en el ciclo de vida, debe centrarse principalmente en la multa que temple el producto, configuración, instalación y las ediciones de la utilidad.

## **B. Disciplinas.**

Una disciplina muestra todas las actividades que se pueden hacer para producir un set de artefactos particular.

### **a) Modelado de negocios.** Sus propósitos son:

- Entender la estructura y la dinámica de la organización en donde el sistema es realizado.
- Entender los problemas actuales en la organización e identificar sus potenciales a mejorar.
- Asegurar que los clientes, usuarios finales y desarrolladores tengan un conocimiento común de la organización.
- Derivar los requerimientos necesarios del sistema para dar soporte a la organización.

Para alcanzar estas metas, la disciplina de modelado de negocios describe como desarrollar una visión de la nueva organización y en base a esta visión definir los procesos, roles y responsabilidades de dicha organización en un modelo de caso de uso de un negocio y un modelo del objetivo de negocios.

### **b) Requerimientos.** Sus propósitos son:

- Establecer y mantener un acuerdo con los clientes.
- Proveer desarrolladores de sistemas con un mejor entendimiento de los requerimientos.
- Definir los límites del sistema
- Proveer una base de planeación técnica.
- Proveer una base de estimación de costo y tiempo de desarrollo de sistema.
- Definir la interfaz de usuario para el sistema.

### **c) Análisis y diseño.** Sus propósitos son:

- Transformar los requerimientos en un diseño.
- Envolver una arquitectura robusta para el sistema.
- Adaptar el diseño que marque la implementación del ambiente.

### **d) Desarrollo.** Sus propósitos son:

- Definir la organización del código, en términos de subsistemas.
- Implementar clases y objetos en términos de componentes.
- Evaluar el desarrollo de componentes como unidades.

- Integrar los resultados producidos por individuos que implementan y ejecutan el sistema.

**e) Prueba.** Sus propósitos son:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente advertir de la calidad del software.
- Proveer la validación del diseño y la especificación de requerimientos.
- Validar el software como fue diseñado.
- Validar que los requerimientos han sido implementados apropiadamente.

**f) Implantación.** Sus propósitos son:

- Describe las actividades asociadas asegurando que el producto de software este disponible para los usuarios.

**g) Medio Ambiente.** Sus propósitos son:

- Se concentra en las actividades necesarias para configurar el proceso de un proyecto. Describe las actividades requeridas para desarrollar las guías para el soporte del proyecto.

**h) Administración del proyecto.** Sus propósitos son:

- Hacer el trabajo fácil al proveer algún contexto del manejo del proyecto.
- Proveer un marco de trabajo para manejar el proyecto intensivo de software.
- Proveer un marco de trabajo para manejar riesgos.

**i) Configuración y administración del cambio.** Sus propósitos son:

- Un sistema CM es esencial para controlar los numerosos artefactos de mucha gente que trabajan en un proyecto común. El control ayuda a prevenir constantemente confusiones y asegura que los resultados de los artefactos no serán conflictivos a pesar de ya tener algunos problemas.

El RUP muestra como se puede aplicar las mejores prácticas de la ingeniería de software y como se pueden usar las herramientas para automatizar el proceso de ingeniería de software.

### 1.1.3. Comparación de Metodologías Estructuradas y Orientadas a Objetos

La comparación de métodos de desarrollo de sistemas de software es una tarea difícil [30], el foco de cada metodología puede ser diferente, algunas tratan de concentrarse en varios aspectos del proceso de desarrollo, otras tratan de detallar en profundidad algún aspecto en particular. En la Tabla 1 se presenta una comparación de distintas metodologías [30], teniendo en cuenta los pasos que componen el proceso, la técnica de modelado, la representación gráfica, la notación elegida para los modelos y la herramienta CASE de soporte proporcionada para el desarrollo. Las metodologías comparadas son: HDM (Hypermedia Design Method<sup>26</sup>) [11], RMM (Relationship Management Methodology<sup>27</sup>) [41], EORM (Enhanced Object Relationship Methodology<sup>28</sup>) [5], OOHDM, SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology<sup>29</sup>) [16], WSDM (Web Site Design Method<sup>30</sup>) [31] y RUP (para Web) [24].

Cabe aclarar que las metodologías que se comparan en la Tabla 1 y Tabla 2 son las más comunes, es por ello que las demás metodologías no se mencionan en dichas tablas.

---

<sup>26</sup> Método de Diseño Hipermedia

<sup>27</sup> Metodología de Administración de Relaciones

<sup>28</sup> Metodología de Relaciones de Objetos Mejorada

<sup>29</sup> Methodology Metodología de Diseño Hipermedia orientada a objetos y basada en escenarios

<sup>30</sup> Método de Diseño de Sitios Web

**Tabla 1:** Comparación de metodologías.

	<b>I. Proceso</b>	<b>Técnica de modelado</b>	<b>Representación gráfica</b>	<b>Notación</b>	<b>Herramienta de soporte</b>
<b>HDM</b>	1.Desarrollo a largo plazo 2.Desarrollo a corto plazo	E-R	1.-2.Diagrama E-R	1.E-R	
<b>RMM</b>	1.Diseño E-R 2.Diseño <i>Slice</i> 3.Diseño de navegación 4.Diseño de protocolo de conversión 5.Diseño de UI 6.Diseño de comportamiento en tiempo de ejecución 7.Prueba y construcción	E-R	1.Diagrama E-R 2.Diagrama <i>Slice</i> 3.Diagrama RMDM	1.E-R 2.3.Propio	<i>RMCase</i>
<b>EORM</b>	1.Clases del entorno de desarrollo 2.Composición del entorno de desarrollo 3.Entorno de desarrollo de UI	OO	1.Diagrama de clases 2.Diseño GUI	1.OMT	<i>ONTOS Studio</i>
<b>OOHDM</b>	1.Diseño conceptual 2.Diseño navegacional 3.Diseño abstracto de la UI 4.Implementación	OO	1.Diagrama de clases 2.Diagrama navegacional, clase + contexto 3.Diagrama de configuración de ADV + Diagrama ADV	1.OMT/ UML 2.Propio 3.ADV	<i>OOHDM-Web</i>
<b>SOHDM</b>	1.Análisis del dominio 2.Modelo en OO 3.Diseño de la vista 4.Diseño navegacional 5.Diseño implementación 6.Construcción	Escenarios Vistas-OO	1.Diagramas de escenarios de actividad 2.Diagrama de estructura de clase 3.Vista OO 4.Esquema de enlace navegacional 5.Esquema de páginas	1.-5.Propio	
<b>WSDM</b>	1.Modelado del usuario 2.Diseño conceptual 2.1.Modelo objetos 2.2.Diseño navegacional 3.Diseño implementación 4.Implementación	E-R/ OO	1.Diagrama de E-R o clase 2.Capas de navegación	1.E-R/ OMT 2.Propio	
<b>RUP</b>	1.Manejo de proyecto 2.Captura de requerimientos 3.Análisis 4.Diseño 5.Implementación 6.Prueba 7.Desarrollo 8.Configuración y manejo de cambios	OO	2.-5.Diagramas UML	UML	<i>Rational Rose</i>

En la Tabla 2 se presenta un segundo estudio comparativo de la misma fuente [30], que relaciona los conceptos de diseño de los tres niveles típicos de diseño *web*: conceptual, estructural y visible. La mayoría de estos métodos realizan una clara separación entre el análisis del dominio, la especificación de la estructura navegacional y el diseño de la interfaz de usuario.

**Tabla 2:** Comparación de conceptos de diseño de las metodologías de desarrollo *web*

	<b>HDM</b>	<b>RMM</b>	<b>EORM</b>	<b>OOHDM</b>	<b>SOHDM</b>	<b>WSDM</b>	<b>RUP</b>
<b>Nivel Conceptual</b>	Entidad Colección Perspectiva Relaciones	entidad relación	clases relación-OO -generalizada -definida por el usuario	clases perspectiva relación-OO	escenarios: -evento -actividad flujo de actividad	objeto perspectiva relación	<i>case</i> relación-OO
<b>Nivel Estructural</b>	Enlace: -estructural -aplicación -perspectiva  componente nodo  Estructuras de acceso: -enlace colección -enlace índice -visita guiada	enlace: -unidireccional -bidireccional  <i>Slices</i>  primitivas de acceso: -agrupar (menú) -índice -visita guiada -visita guiada indexada	enlace: -simple -navegacional -nodo a nodo -tramo a nodo -estructural -conjunto -lista	enlace  clase navegacional  contexto navegacional  estructuras de acceso: -índice -visita guiada	enlace navegacional  vista-OO: -base -asociación -colaboración  ASN: -agrupar -índice -visita guiada	enlace  componente -navegación -información -externo  camino navegacional	enlace enlace dirigido redirigir construir enviar  página <i>web</i> -página del cliente -página del servidor
<b>Nivel Visible</b>	Ranura Marco	<i>Slices</i>	ADV  en contexto	componente UI:  -elección -texto de entrada de búsqueda -botón -imagen -barra de desplazamiento - ancla HTML -otros		conjunto de marcos formulario objetivo elemento de selección elemento de entrada elemento de área de texto	

## 1.2. Lenguajes de Programación Web

Son varios los lenguajes de programación que se adaptan a las necesidades de los sitios web, presentando cada uno de ellos sus ventajas e inconvenientes, todas ellos se ejecutan en el servidor web, generando código HTML puro, por lo que el visitante recibe en su navegador páginas sin ningún tipo de código, solamente el HTML y JavaScript generado.

Los principales lenguajes de programación son:

### 1.2.1. Lenguaje PHP

PHP (Hypertext Preprocessor<sup>31</sup>) es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor [56] [55].

El lenguaje PHP no es lo mismo que un script escrito en otro lenguaje de programación como Perl o C, en vez de escribir un programa con muchos comandos para crear una salida en HTML, se escribe el código HTML con cierto código PHP incluido en el mismo, que producirá cierta salida, este código se incluye entre etiquetas especiales de comienzo y final que permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor; si se tiene un script en servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido, el servidor Web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

El usar PHP es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales, aunque el desarrollo de PHP está concentrado en la programación de scripts en el lado del servidor, se puede utilizar para muchas otras cosas.

#### a) Lo que se puede hacer con PHP.

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Existen tres campos en los que se usan scripts escritos en PHP.

- **Scripts del lado del servidor.** Este es el campo más tradicional y el principal foco de trabajo, se necesitan tres cosas para que esto funcione: el intérprete PHP (CGI ó módulo), un servidor Web y un navegador. Es necesario correr el servidor Web con PHP instalado, el resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor Web.
- **Scripts en la línea de comandos.** Puede crear un script PHP y correrlo sin ningún servidor Web o navegador, solo se necesita el intérprete PHP para usarlo de esta manera, este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en \*nix o Linux) o el

Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.

- **Escribir aplicaciones de interfaz gráfica.** Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si se conoce bien PHP y si se requiere utilizar algunas características avanzadas en programas clientes, se puede utilizar PHP-GTK para escribir dichos programas, también es posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más.; soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros; tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

Con PHP se tiene la libertad de elegir el sistema operativo y el servidor, también se tiene la posibilidad de usar programación procedimental o programación orientada a objetos, aunque no todas las características estándar de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas bibliotecas y aplicaciones grandes (incluyendo la biblioteca PEAR) están escritas íntegramente usando programación orientada a objetos.

Con PHP no hay limitaciones a resultados en HTML, entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) sobre la marcha. También puede presentar otros resultados, como XHTML y archivos XML. PHP puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos, escribir un interfaz vía Web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

- Adabas D
- InterBase
- PostgreSQL
- FilePro (read-only)
- Direct MS-SQL

---

<sup>31</sup> Preprocesador de Hipertexto

- Sybase
- IBM DB2
- MySQL
- Informix
- ODBC

PHP cuenta con una extensión DBX de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros; PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en Web, puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

PHP tiene unas características muy útiles para el procesamiento de texto, desde expresiones regulares POSIX extendidas o tipo Perl hasta procesadores de documentos XML. Para procesar y acceder a documentos XML, soporta los estándares SAX y DOM, puede utilizar la extensión XSLT para transformar documentos XML.

Si se usa PHP en el campo del comercio electrónico, se encontrarán muy útiles las funciones Cybercash, CyberMUT, VeriSign Payflow Pro y C CVS para programas de pago.

Cuenta con otras extensiones como: las funciones del motor de búsquedas mnoGoSearch, funciones para pasarelas de IRgfc, utilidades de compresión (gzip, bz2), conversión de calendarios, traducción, etc.

### **1.2.2. Lenguaje JAVA**

*Java* es un lenguaje de programación orientado a objetos desarrollado por la compañía Sun Microsystems [64, 29], está construido a partir de lenguajes orientados a objetos anteriores, como C++, pero no pretende ser compatible con ellos sino ir mucho más lejos, añadiendo nuevas características como recolección de basura, programación multihilos y manejo de memoria a cargo del lenguaje. *Java* fue diseñado para que la ejecución de código a través de la red fuera segura, para lo cual fue necesario deshacerse de herramientas de C tales como los punteros, también se han eliminado aspectos que

demonstraron ser mejores en la teoría que en la práctica, tales como sobrecarga de operadores, que por cierto todavía está en discusión y herencia múltiple [71].

La portabilidad fue otra de las claves para el desarrollo de *Java*, para lograr que las aplicaciones se escriban una sola vez sin la necesidad de modificarlas para que corran en diferentes plataformas, esta independencia se alcanza tanto a nivel de código fuente (similar a C++) como a nivel de código binario. La solución adoptada fue compilar el código fuente para generar un código intermedio (*bytecodes*) igual para cualquier plataforma.

La JVM (Máquina Virtual de *Java*), donde reside el intérprete *Java*, sólo tiene que interpretarlos.

### 1.2.3. JDBC

JDBC (Java-Conectividad de Base de Datos<sup>32</sup>) [71] es una interfaz que provee comunicación con bases de datos, consiste de un conjunto de clases e interfaces escritas en *Java* [71], 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. Conceptualmente es similar a ODBC (Conectividad de Base de Datos Abierta), pero ésta no es apropiada para usar directamente desde *Java* porque usa una interfaz en C y una traducción literal de C a *Java* no es deseable.

JDBC soporta dos modelos de acceso a base de datos: modelo de dos capas y modelo de tres capas [71]; en el primer caso, la aplicación *Java* se comunica directamente con la base de datos mediante un controlador JDBC específico para cada DBMS (Sistema de Administración de Base de Datos) que se desee manipular; en el segundo caso, los comandos son enviados a una capa intermedia de servicios, encargado de reenviar las sentencias SQL a la base de datos.

Existe un controlador, llamado puente JDBC-ODBC, que implementa las operaciones de JDBC traduciéndolas en operaciones ODBC, con lo cual se provee acceso a cualquier base de datos cuyo controlador ODBC se encuentre disponible.

---

<sup>32</sup> Java-Conectividad de Base de Datos

#### 1.2.4. Java Servlets

Un *servlet* es una clase *Java* [30], embebida dentro del *web server* y utilizada para extender la capacidad del servidor [71]. 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* [11] 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. Son cargados la primera vez que son usados y permanecen en memoria para satisfacer futuros requerimientos. Tiene un método *init*, donde el programador puede inicializar el estado del *servlet* y un método *destroy* para administrar los recursos que son mantenidos por el *servlet*.

#### 1.2.5. XML

La familia XML (Lenguaje Extensible Markup<sup>33</sup>) 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 [71]. 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

---

<sup>33</sup> Lenguaje de Marcación Extendida

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.

### **1.2.6. XSL**

XSL<sup>34</sup> es una especificación desarrollada dentro del W3C (*World Wide Web Consortium*) para aplicar formato a los documentos XML de forma estandarizada [71].

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. En esencia, XSL son dos lenguajes: uno de transformación y otro de formateo. El lenguaje de transformación permite transformar un documento XML en otro con diferente formato, como HTML o texto plano, o bien en otro documento XML. El lenguaje de formateo no es más que un vocabulario XML para especificar objetos de formateo (FO).

Al igual que con HTML, se pueden especificar las hojas de estilo, CSS o XSL, dentro del propio documento XML o haciendo referencia a ellas en forma externa. Esto es muy útil para mover datos de una representación XML a otra representación, basada en correo electrónico, intercambio de datos electrónicos, intercambio de metadatos y alguna aplicación que necesite convertir datos entre diferentes representaciones de XML a otro tipo de representación.

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

---

<sup>34</sup> Lenguaje Extendido de Hojas de Estilo

Existen tres opciones para transformar un documento XML a otro formato, como se describen en, tal como HTML, utilizando hojas de estilo XSL:

- *En el cliente:* los documentos XML y las hojas de estilo son enviados al cliente (*Web Browser*), el cual se encarga de transformar los documentos basándose en la especificación de las hojas de estilo y luego de la transformación se presentan al usuario en el explorador.
- *En el servidor:* el servidor es el encargado de aplicar el XSL al documento XML para transformarlo en algún otro formato (generalmente HTML) y envía el documento transformado al cliente.
- *Ni en el servidor, ni en el cliente:* una tercera opción consiste en transformar el documento original XML en algún otro formato (usualmente HTML) antes de que el documento sea ubicado en el servidor.

### **1.2.7. CGI**

Como su propio nombre indica (Common Gateway Interface ó Interfaz Común de Comunicación<sup>35</sup>), esta tecnología no es en sí un lenguaje de programación, si no un conjunto de reglas que deben cumplir diferentes programas para poder comunicarse correctamente entre sí. Cumpliendo estas reglas, dos aplicaciones cualesquiera, escritas cada una en un lenguaje diferente, pueden intercambiarse datos sin problemas. Por este motivo, se puede escribir script CGI en lenguajes distintos, siendo los más usados Perl y C.

Fueron los primeros script funcionales que se incluyeron en páginas web y se usaron principalmente para la interacción con el usuario por medio de formularios.

Mediante un script CGI una página web puede establecer una conexión a una base de datos, con lo que se puede leer los registros de la misma, añadir nuevos o modificar los existentes. También se puede con esta tecnología crear contadores de visitas, carritos de compra, etc.

Normalmente los script CGI se guardan en una carpeta especial del servidor web, carpeta que cuelga directamente del directorio raíz y que se suele llamar CGI o CGI-Bin y para poder ejecutarlos se necesitan permisos especiales del Administrador del servicio web.

Actualmente está cada vez más en desuso, aunque se han lanzado nuevas versiones de que Perl intenta corregir carencias anteriores e impulsar de nuevo su uso.

---

<sup>35</sup> Interfaz Común de Comunicación

### 1.2.8. Lenguaje ASP

La tecnología **ASP** (Active Page Server ó Página Activa en el Servidor <sup>36</sup>) no es en sí mismo un lenguaje de programación, si no más bien un marco sobre el que se construyen aplicaciones basadas en Internet, apoyándose para ello en el lenguaje HTML, en lenguajes de script conocidos (generalmente **VBScript**, pero también JavaScript –Jscript para Microsoft-, Perl, etc.), en motores de bases de datos y en el lenguaje de consulta SQL [59].

Es una tecnología más avanzada que CGI, aunque precisa para su uso de un servidor Microsoft (Internet Information Server), o de otro tipo de servidor en el que se hayan instalado las extensiones necesarias, aunque normalmente los servidores Apache e IPlanet no se suelen llevar bien con esta tecnología.

Entre sus ventajas están el tener como base un lenguaje de fácil aprendizaje, el permitir acceso a bases de datos de forma estándar y el manejar correctamente variables de sesión y de aplicación y entre sus defectos el que a veces se queda corto en sus funcionalidades y que precisa máquinas concretas.

Debido a esta conjunción de tecnologías diferentes es más acertado considerar una aplicación ASP en su conjunto que una serie de páginas web aisladas.

Las principales características de la tecnología ASP y de sus páginas asociadas son:

- Es gratuita. Viene implementada por defecto en los servidores web PWS (Windows 95-98-Millennium) e IIS (Windows NT-2000-XP). Además también es posible implementarla en otros servidores, aunque hace falta la instalación de las extensiones adecuadas.
- Se puede mezclar código ASP con código HTML y código de script en cliente, sin tener que compilarlos por separado. Esto facilita la construcción de páginas ASP, siendo posible escribirlas en un simple editor de textos (Notepad, Word, etc.).
- El código ASP se ejecuta en el servidor web, produciendo como salida HTML puro (junto con código de script en el cliente, si así se precisa). Por este motivo, su resultado es interpretado correctamente por cualquier navegador web estándar, aunque se utilice VBScript, compatible tan sólo con Internet Explorer y los navegadores en él basados (aunque es posible ejecutar código VBScript en Netscape Navigator usando el plug-in ScriptActive, de la casa NetCompass).
- Permite usar componentes escritos en otros lenguajes de programación, como Visual Basic, Delphi, etc., que pueden ser llamados desde los propios guiones ASP.

---

<sup>36</sup> Página Activa en el Servidor

- Se ha portado a la plataforma Java, lo que hace que pueda ser utilizado en casi cualquier sistema operativo.
- Permite acceder a las bases de datos de una forma rápida y sencilla, siendo éste uno de sus principales usos.
- También permite el uso de componentes OLE, acceso a ficheros, a login del sistema, enviar correo, etc.
- Tiene persistencia de variables en memoria entre distintas visualizaciones de página, permitiendo con ello manejar sesiones de usuario, con lo que se resuelve uno de los mayores problemas de la programación web, derivado de la no orientación a conexión del protocolo HTTP.

#### **a) Herramientas necesarias para trabajar con ASP**

Para escribir código ASP sólo hace falta un editor de textos cualquiera, desde el sencillo Notepad hasta editores de texto más complejos, como Word. También se puede desarrollar en editores web comunes, como Allaire HomeSite, Macromedia Dreamweaver, etc. Para usos avanzados existen aplicaciones especialmente concebidas para trabajar con ASP, como Visual Interdev, de Microsoft.

En cambio, para poder visualizar páginas ASP hace falta disponer de un servidor web que implemente la tecnología necesaria. Para ello se puede usar dos servidores de Microsoft:

- Personal Web Server, PWS: Es en realidad una especie de servidor web "de juguete", pero que permite una instalación fácil y una perfecta visualización de páginas ASP y viene de forma gratuita con Windows 95-98-Millennium, por lo que es una solución ideal para trabajar "en local".
- Internet Information Server, IIS: Este es ya un servidor web en toda regla, que no sólo permite implementar servicios de páginas web basados en el protocolo HTTP, si no que también suministra servicios de correo, noticias, servidor seguro, etc. Su instalación y configuración son mucho más complejas que las del PWS, siendo su uso indicado en servidores web públicos, accesibles desde Internet o desde una Intranet.

## Lenguaje JSP

La tecnología JSP<sup>37</sup> fue introducida por la empresa Sun, en estrecha colaboración con Netscape, buscando extender las funcionalidades del lenguaje Java (por ella creado) a las aplicaciones basadas en la web.

Desde su aparición su uso ha ido creciendo fuertemente, siendo actualmente una de las tecnologías más usadas y demandadas para aplicaciones web.

JSP está basada en el lenguaje Java. Incluyendo instrucciones Java en las páginas, estas se ejecutan en el servidor web, devolviendo al navegador cliente código HTML puro. Permite el acceso a casi cualquier base de datos y, al estar basadas en Java, son multiplataforma, multiusuario y contienen código seguro.

JSP se complementa con los Servlets y con los EJB (Enterprise Java Bean<sup>38</sup>), que son módulos escritos en Java que pueden ser llamados desde máquinas remotas, lo que transforma el lenguaje Java en la web en un lenguaje modular.

Sólo decir que para su uso se precisan servidores web que acepten esta tecnología. Inicialmente la implementaron los servidores específicos de Sun y Netscape, pero ya la aceptan correctamente Apache, Resin, etc., e incluso los servidores de Microsoft poseen extensiones para soportarla.

### 1.2.9. Lenguaje SSI

SSI (Server Side Incluye<sup>39</sup>) es un lenguaje de comandos específicos que se ejecutan en el servidor. Se puede apreciar una página web de este tipo por que su extensión es shtml. Es soportado por la mayoría de los servidores web.

### 1.2.10. Lenguaje HTML

HTML (Lenguaje de Marcado de Hipertexto) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces 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

---

<sup>37</sup> Servidor de Páginas Java

<sup>38</sup> Java Bean Empresarial

dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape) [48].

Si una página es correctamente escrita debe de ser vista por cualquier navegador.

Esto se puede realizar mediante la utilización de la codificación HTML y sus estándares los cuales son delimitados por el World Wide Web Consortium (W3.org).

### **1.3. Servidores**

En cuanto a los servidores web, los más usados son los siguientes:

#### **1.3.1. Servidor Apache**

Es por ahora el servidor web más usado y está basado en el sistema operativo UNIX y orientado principalmente a él. Es un servidor web desarrollado por la comunidad de programadores UNIX, es de código abierto y lo más importante, gratuito en sus versiones estándares.

Desde su aparición ha ido sufriendo cambios para adaptarse a las necesidades de cada momento y actualmente soporta casi todas las tecnologías, incluyendo JSP, aunque si no posee extensiones adecuadas no soporta ASP.

#### **1.3.2. Servidor IIS**

Internet Information Server es el servidor web de la casa Microsoft, especialmente diseñado para soportar sus tecnologías propietarias, como ASP y extensiones de Front Page. También posee extensiones para PHP, SSI, etc.

Es un servidor fácilmente configurable, aunque menos seguro que Apache. Se puede encontrar en los CDs de Windows NT Server, ya que está especialmente pensado para trabajar con ese sistema operativo, aunque también se puede encontrar una versión reducida del mismo en los CDs de instalación de Windows 98 o Millennium, que permite su instalación y uso en estos sistemas.

Aunque empezó desde cero, cada vez es más usado y está comiéndole terreno día a día a los servidores Apache.

#### **1.3.3. Servidor Iplanet**

Es este un servidor web creado por la colaboración de las empresas Sun Microsystems y Netscape. Esta última disponía de una familia de servidores web propios, Netscape Enterprise, que han adaptado al

---

<sup>39</sup> Servidor Lateral Incógnito

uso del lenguaje Java de Sun, creando lo que se conoce como servidor IPlanet. Especialmente indicados para aplicaciones web basadas en JSP, Servlets y EJB.

Está comenzando ahora su expansión, aunque por ahora está muy limitado a usos concretos, debido a su falta de compatibilidad con sistemas anteriores.

También se pueden encontrar otros servidores web varios, como los de la empresa IBM, Resin, etc., pero son menos usados.

Hay otra serie de servidores necesarios para muchas aplicaciones web, como pueden ser los **servidores de bases de datos**, que van a condicionar el tipo de ellas que se pueden manejar en la aplicación (Oracle, MySQL, Informix, etc.), los **servidores seguros**, que implementan manejo de claves cifradas, certificados digitales, Secure Socked Layer (peticiones https), etc. y los **servidores de aplicaciones**.

# Capítulo II



## *Portales y Calidad Web*

Este capítulo trata de los diferentes aspectos que se consideran necesarios para el desarrollo de portales Web y de las diferentes normas de calidad establecidas para este fin.

PIENSSA tiene como finalidad ofrecer Portales Web Corporativos que incluyan su documentación técnica y consideren aspectos mínimos de calidad para poder ser atractivos a los usuarios de dicha aplicación.

## **2.1. Portales Web**

Un sitio web no es sólo un lugar virtual en el que se le expone al visitante imágenes, textos y animaciones. Si se elabora así, el portal se convierte en un elemento pasivo y se desaprovechan muchas de las posibilidades que brinda Internet. El gran objetivo de cualquier sitio web es interesar al visitante y hacerle participar para que se sienta miembro de la Comunidad Virtual generada alrededor de esa web y el tema que trata, para de esa forma fidelizarlo. Y eso es algo que sólo se consigue cuando se logra que el internauta interactúe con la web.

En un principio la construcción de un sitio web era una tarea más o menos simple. Constaba de unas cuantas páginas enlazadas entre sí de forma sencilla y cada una de ellas estaba formada por un código HTML básico, unas pocas imágenes y algunos elementos más.

*Un portal de Internet es un sitio web dinámico que recibe muchos visitantes diariamente [60].*

Los portales otorgan un vasto contenido y aplicaciones que dan funcionalidad a los usuarios y los invitan a regresar con frecuencia, siempre y cuando se aplique en ellos adecuadamente: capital, buenas ideas, conocimientos técnicos y tecnología.

La implementación de Portales Corporativos para organizaciones empresariales ofrece valor añadido a sus clientes, empleados y socios desde un único punto de acceso, con el fin de maximizar su gestión de negocio.

Un Portal bien estructurado facilita:

- Un entorno único que optimiza la búsqueda eficaz de recursos.
- Una reducción en los costos de inversión y mantenimiento de la web.
- Una rápida innovación en la adopción de cambios (tan importante en el mercado actual).
- Un acceso eficiente a la información que proporcione su empresa a clientes, empleados y socios.
- Una reducción de los costos de administración y formación.

Esto ha hecho que aquellos, inicialmente simples, sitios web se conviertan en verdaderos monstruos de código, formados por multitud de páginas repletas de tablas, formularios y códigos de lenguajes de programación de todo tipo. Se ha pasado de lo simple a lo complejo, proceso que continúa imparablemente.

Al cambiar el concepto de los sitios web también ha cambiado paralelamente la forma de planificarlos y de estructurarlos, siendo necesario ahora un complejo estudio de los mismos, con objeto de crear un ente complejo y funcional, bien estructurado, capaz de facilitar a los usuarios finales una navegación clara y una operatividad perfectamente definida. Con ello se ha pasado del diseñador web solitario, que construye páginas artesanalmente, a un completo y complejo equipo de desarrollo, formado por numerosos técnicos que siguen un plan perfectamente estudiado de diseño y programación. Esto es lo que requiere la sociedad actual y esto es lo que hay que darle.

Este proceso constructivo es aplicable en mayor o menor grado a la planificación y puesta en funcionamiento de cualquier sitio web; en unos casos habrá que añadir algún paso adicional y en otros bastará con realizar sólo unos pocos, pero el esquema general será válido para todos.

### **2.1.1. Artefactos Web como Software**

*"Software es un programa de computadora o un conjunto de programas de computadora, además del material relacionado como documentación, el cual se usa para realizar alguna tarea" [40].*

Un sitio Web puede en muchos casos ser considerado un artefacto de software (y hasta con complejidad añadidas). Un simple sitio Web puede ser no más que un conjunto de páginas estáticas (documentos hiperenlazados) recuperados por un usuario, a partir de un navegador, desde un servidor remoto. No obstante, en la medida que el sitio provee capacidad de recolectar información desde formularios, provee recuperación de información desde bases de datos por medio de consultas (y acceso a sistemas legados), permite generar páginas dinámicamente (consultas, búsquedas, generación de páginas personalizadas, etc.), permite realizar cómputos por medio de applets o scripts, etc. La funcionalidad del artefacto Web es comparable a la funcionalidad de programas de software [26, 37, 38].

Mientras que la publicación de un sitio con componentes estáticos orientados a la documentación debe lograr un adecuado compromiso de calidad entre organización, presentación, contenido y navegación, los componentes orientados a la aplicación (programación) deben proveer la adecuada integración y funcionalidad de la lógica agregada. Con todo, características de calidad del artefacto como confiabilidad y eficiencia, puede ser afectada por ambos componentes. Por ejemplo, un sitio estático pobremente evaluado puede contener varios enlaces rotos o inválidos, lo cual degrada la confiabilidad del

mismo desde el punto de vista del usuario o por otra parte, una falta de prueba de funcionalidad de applets o scripts, puede provocar deficiencias en la presentación dependiendo del tipo de navegador. Del mismo modo, un diseño descuidado de las páginas en consideración de la suma de los tamaños de todos sus componentes, puede degradar el rendimiento del sitio y en definitiva la eficiencia.

### **2.1.2. Estudio Previo**

Antes de crear el sitio es necesario y de una gran importancia delimitar claramente el objetivo de las páginas, con lo cual se habrá avanzado mucho en el establecimiento del esquema general que se usará en el sitio web.

No es lo mismo el diseño ni los contenidos de una página que va a servir para transmitir unos conocimientos (un manual, por ejemplo) que los de una cuyo objetivo va a ser mostrar un catálogo de productos, ni estos que los de una página cuya finalidad va a ser vender esos productos o de los de una página destinada a la banca electrónica. En todos estos casos, las herramientas necesarias, las bases de datos y los lenguajes de programación a usar, tanto en cliente como en servidor, serán también diferentes.

La siguiente pregunta a hacerse es ¿Para quién se va a crear el conjunto de páginas web? y esta pregunta a la vez se debe desglosar en dos: ¿Quién va a ser el cliente? (la persona o empresa para la que se va a crear el sitio) y ¿Quiénes van a ser los potenciales visitantes?

La primera de ellas va a marcar el contenido de las páginas que se deben crear, ya que en ellas se debe expresar el espíritu y la imagen de la empresa, organización o particular para el que se crean, conjugando esta imagen con las directrices que se sacaron de la pregunta "para qué". Es decir, si el sitio web va a ser por ejemplo una tienda on-line, con todas las tecnologías y herramientas que se necesitan para ello, debiendo adaptarse estas y el diseño de las páginas a la imagen corporativa concreta de la empresa que va a vender los productos. Esto incluye el diseño y la implementación del logotipo de la misma, el uso de los colores corporativos propios, etc.

En cuanto al segundo apartado, referente a quien van a estar dirigidas principalmente las páginas, va a depender del estudio estadístico potencial de visitantes posibles. Si el sitio a crear es de manuales informáticos para principiantes se debe crear un diseño neutro y general, apto para todo tipo de público, con menús de navegación claros y unos textos fácilmente comprensibles. Si es de manuales avanzados, se supone que los visitantes ya poseen conocimientos suficientes del tema a tratar y además una buena experiencia en navegación por Internet, por lo que se tienen más libertades a la hora de diseñar los menús y el contenido textual.

Si el sitio es una tienda virtual se debe, en este caso, usar un esquema de navegación simple e intuitivo, mostrando siempre claramente la forma de consultar los artículos que se ofrecen y el método para realizar la compra y el pago, ya que la venta será el principal objetivo. Procurando que el usuario final pueda acceder a cualquiera de los artículos en menos de 3 click's de ratón y que el formulario de introducción de datos personales y de su número de tarjeta de crédito (si así procede) sea lo más sencillo posible, a la vez ofrecer una imagen de seguridad y confianza.

El tipo de público al que irá destinada la aplicación web va a ser determinante del diseño estético de las páginas que la formen. Así, si el usuario tipo es una persona joven, los colores y las imágenes deben ser vivos, alegres, estando permitido el uso en este caso de gamas brillantes. En cambio, si el usuario tipo es una persona de edad madura los colores deben ser neutros, tal vez en tonos pastel o grises, con pocas variaciones en el colorido, o colores serios, como el azul oscuro. Para cada tipo de usuario final hay que hacer un estudio de colores y gráficos adecuado.

La última pregunta es cómo se va a plasmar todo lo anterior y su respuesta es ya el complejo proceso de la creación del sitio. Los pasos necesarios para ello van a variar mucho dependiendo de las premisas obtenidas antes.

Sea como sea, lo primero que hay que hacer es recoger toda la información posible sobre el trabajo a realizar. Si se va a diseñar una web de empresa o de comercio, se deberá repasar cuantos catálogos, folletos, revistas, imágenes, etc., se pueda sobre el cliente, ya que de ellos se sacará una visión general de sus colores corporativos, de sus logotipos, de la imagen que desean dar al público, de campañas publicitarias ya realizadas, etc., lo que nos ayudará en gran medida a diseñar un sitio web acorde con la imagen que nuestro cliente ya posee, adaptándola al medio web. También se puede sacar de esta información ideas sobre el tipo de productos que maneja, sobre el público al que están dirigidos, etc.

Una vez estudiadas estas fuentes se comenzará con el proceso creador. Para empezar se debe fijar qué estructura se dará al sitio web.

La Fig. 2 muestra una representación de aplicaciones Web considerando rangos de complejidad (estáticas versus dinámicas) y grados de orientación (orientado a documentos versus orientado a aplicaciones). Según esta figura los sitios pueden ser categorizados en [37]:

- *Sólo Estático.*
- *Sitio Estático con Formularios de Entrada.*
- *Sitio con Acceso de Datos Dinámicos.*

- *Sitio Creado Dinámicamente.*
- *Aplicación de Software basada en la Web.*

**Fig. 2:** Rangos de complejidad y niveles de orientación de las aplicaciones Web

- *Sitio Sólo Estático:* En su forma más simple, un sitio Web es una colección de páginas estáticas (documentos, o información editada y publicada en formato HTML). Desde el punto de vista de la funcionalidad, ésta es dada básicamente por los enlaces que permiten navegación, ya sean enlaces provistos por los controles principales y navegacionales de la interfase, o por los enlaces estructurales (mecanismos de acceso como un índice) y semánticos. En un sitio de este tipo, el énfasis de diseño está puesto en la organización de la estructura y el contenido, en la estética de la presentación y en las facilidades de navegación (no obstante hay otros desafíos de usabilidad, además de eficiencia, confiabilidad y mantenibilidad) [37].
- *Sitio Estático con Formularios de Entrada:* En un sitio de este tipo, el énfasis de diseño está puesto en la organización de la estructura y el contenido, en la estética de la presentación, en las facilidades de navegación, etc., proveyendo además al usuario un nivel de interacción básico implementado por medio de formularios de entrada. Esto favorece la usabilidad del sitio al permitir mecanismos de retroalimentación por parte del usuario. Atributos como *Cuestionarios,*

*Libros de Invitados o Comentarios y Sugerencias*, son de fácil implementación favoreciendo a la comunicación en línea.

- *Sitio con Acceso a Datos Generados Dinámicamente:* En un sitio de este tipo, además de las características previamente comentadas, el usuario puede acceder por medio de las páginas (del lado del cliente) a datos almacenados en bases de datos remotas, por medio de consultas y búsquedas. Los datos retornados a partir de la interacción iniciada por el usuario son generados dinámicamente (ASP, etc.) y presentados en formato de documentos HTML (DHTML o XML).
- *Sitio Creado Dinámicamente:* Surge la necesidad de diseñar este tipo de sitios Web, en el caso en que el mismo deba proveer requerimientos semejantes aunque personalizados en consideración del contenido de las páginas para cada instancia de usuario o cuando por aspectos de compatibilidad tecnológica, se requiere construir dinámicamente a sitios en conformancia con el entorno del navegador del usuario. Para ello, los documentos estáticos deben ser mudados dinámicamente aunque en el lado del cliente, no provean interactividad alguna.
- *Aplicación de Software basada en la Web:* Este tipo de artefacto puede ser el más complejo y con mayor orientación a la aplicación, como se aprecia en la Fig. 2. Este tipo de sitio Web (a veces implementado como Intranet o Extranet) puede ser un sistema de control y seguimiento de inventarios, o un sistema educación a distancia, etc., proveyendo funcionalidad que está más cercana a una implementación cliente/servidor tradicional que a un sitio Web estático. Sin embargo, todas las características potenciales de los tipos antes discutidos pueden estar incorporadas a la aplicación.

Como se observa de lo anterior, los artefactos Web se pueden tornar muy complejos. Al igual que el desarrollo de productos de software tradicional, el desarrollo y evolución de aplicaciones centradas en la Web puede ser un proceso con muchos desafíos, máxime si se emplean estrategias ad hoc.

### **2.1.3. Necesidad de Modelos de Proceso de Desarrollo de Artefactos Web**

Por otra parte, es necesario tomar conciencia que si bien la Ingeniería en la Web involucra el modelado y desarrollo de funcionalidad de software y adopta principios y prácticas establecidas de la Ingeniería de Software, hay características propias en los desarrollos y artefactos Web que deben ser debidamente puntualizados. Varios autores afirman que, en tanto los sitios Web pueden ser considerados software, los desafíos que abarca la Ingeniería en la Web no son solamente los que enfrenta la Ingeniería de Software [18, 38, 8]. Los sitios Web y su desarrollo tienen algunos aspectos que lo hacen y harán distintos del software tradicional:

- La mayoría de los sitios Web seguirán teniendo la característica de orientación a la documentación. En lo previsible no hay razón para pensar que se abandone esta característica básica surgida desde los inicios de la Web. Las páginas se generan estática o dinámicamente.
- Las aplicaciones Web continuarán focalizadas en la interfase de usuario y en su apariencia y estética (ver y apreciar), favoreciendo la creatividad visual y la incorporación de multimedia. Muchos sitios están dirigidos por objetivos de presentación, promoción de productos y servicios y metas de negocios; esto parece ir en incremento.
- La mayoría de los sitios Web seguirán siendo orientados al contenido. Esta característica se observa en muchas aplicaciones de software que ofrecen documentación y ayuda en línea; sin embargo, el alcance de la incorporación de contenido en un sitio Web es mucho más amplio. Además, el proceso de autoría es generalmente propio de desarrollos Web o CD-ROM multimedia.
- Un sitio Web puede atender una multiplicidad de perfiles de usuario y sus distintos requerimientos. El construir un sitio orientado a audiencias no siempre es una tarea sencilla.
- El medio donde corren las aplicaciones basadas en la Web es generalmente más impredecible que el medio donde se ejecutan las aplicaciones de software tradicionales. Esto puede afectar la percepción y aceptabilidad que tiene el usuario del artefacto.
- Un sitio Web implica un mayor acercamiento entre arte y ciencia que lo que puede implicar un artefacto de software.
- En general un proyecto Web de mediana y gran escala requiere mayor multiplicidad de tipos y niveles de habilidades y roles de los participantes que un proyecto de software.
- La Web representa un nuevo paradigma en la distribución de documentación y software.
- La mayoría de los desarrollos en la Web requieren estar operativos en un corto lapso de tiempo. Esto puede dificultar la aplicación del mismo nivel de rigor formal o semiformal de varias actividades del proceso que en un proyecto tradicional.

Lo anterior representa las principales características distintivas de un proyecto de desarrollo en la Web.

### 2.1.4. Tipos de Estructuras Web

La estructura de un sitio web se va a referir a la disposición entre los enlaces de las diferentes páginas que lo forman, es decir, al esquema general de disposición de las páginas entre sí y a la forma de acceso entre ellas.

Los principales tipos de estructura son:

**1) Estructura Jerárquica:** Parte de una página principal mediante la que se puede acceder a diferentes páginas secundarias, a partir de las cuales se puede acceder a las páginas terciarias y así sucesivamente. La disposición de un sitio de este tipo sigue el esquema general expresado en la Fig. 3:

**Fig. 3:** Estructura Jerárquica

Si se usa una estructura de tipo jerárquica se puede crear un menú general en la página principal, mediante el que se dará acceso a las diferentes páginas de entrada a las secciones y en cada una de estas deberá establecer otro menú desde el que el usuario pueda ir a cualquiera de las páginas que la componen. En cada una de las páginas individuales tendrá que implementar unos enlaces a las diferentes secciones principales y a la página de inicio.

**2) Estructura Lineal:** Partiendo de una página inicial se van recorriendo las demás del sitio web secuencialmente, una detrás de otra. Es análoga en su disposición a la estructura de las páginas de un libro.

Cada página posee un enlace a su anterior en la secuencia y otro a su siguiente. La Fig. 4 muestra este tipo de estructura:

**Fig. 4:** Estructura Lineal

El sistema de menús característico de este tipo de estructura sería el acceso a una página de entrada, desde la que se puede acceder únicamente a la página que le sigue en la secuencia establecida y en esta se encontrará un pequeño menú, generalmente situado en la parte inferior o superior de la misma, desde el que se puede acceder tanto a la página anterior como a la siguiente en la secuencia y así sucesivamente, hasta llegar a la última página, en la que sólo figurará un enlace a la página anterior.

Debido a sus repercusiones de diseño y navegación, este tipo de estructuras en su forma pura es raramente usado.

**3) Estructura Lineal-Jerárquica o Mixta:** como su propio nombre indica es una mezcla de las dos anteriores, en la que partiendo de una página principal o de inicio se accede a diferentes páginas de entrada a secciones, a partir de las cuales la navegación es lineal. Su representación se muestra en la Fig. 5:

**Fig. 5:** Estructura Mixta

En el caso de usar este tipo de jerarquía se deberá situar un menú en la página de inicio desde el que se pueda acceder a las páginas de entrada a cada uno de los temas a tratar y en cada una de las páginas que forman la secuencia del tema tendrá que establecer un enlace a la página anterior y otro a la siguiente. Como complemento se puede habilitar en cada una de ellas un enlace a la página que abre la secuencia y en cada una de estas, otro a la página de inicio.

**4) Estructura de Frames:** es la típica de una interfaz a base de frames y donde el usuario dispone de un menú siempre presente desde el que puede acceder en todo momento a las páginas de entrada a las diferentes secciones del sitio web, a partir de las que puede navegar bien de forma jerárquica, bien de forma lineal, bien de forma mixta (ver Fig. 6):

**Fig. 6:** Estructura de Frames

Este tipo de estructura se suele combinar con otro jerárquico o mixto que nos ayude a navegar dentro de los subtemas principales, a los que se accede a través del frame lateral.

El sistema de menús consiste generalmente en un menú situado en un frame lateral, que nunca cambia, desde el que se accede a las diferentes secciones principales del sitio, cuyas páginas iniciales se cargan en el frame central, que suele ser el de mayor tamaño. Para acceder a las diferentes subsecciones y páginas se suelen establecer los enlaces adecuados bien como submenús en el frame lateral, bien como menús individuales dentro de cada página de entrada a las secciones, dentro del frame principal.

**5) Estructura Web:** en la que se puede estructurar las diferentes páginas con libertad total. Es la que da más facilidades a los diseñadores, pero puede resultar a veces demasiado confusa para los usuarios, ya que le permite visitar el sitio sin un rumbo fijo, pudiendo desde cualquier página acceder a los contenidos de un conjunto indeterminado de otras. No es aconsejable su uso, ya que suele resultar caótica. Esta estructura se representa en la Fig. 7:

**Fig. 7:** Estructura Web

Estos son los tipos principales de estructura de un sitio web. Generalmente se suele usar un tipo mixto o un tipo frames, dependiendo siempre de la naturaleza propia del sitio.

### **2.1.5. Primeros Diseños**

Frecuentemente, llegados a este punto, los diseñadores abren su programa editor preferido y comienzan a escribir líneas de código HTML y Javascript sin freno.

Si se hace un estudio de las horas perdidas en un proyecto por adelantarse demasiado a las pautas lógicas de proceso, se verá que al final se pierden tiempo y dinero, ya que el código que se escribe generalmente se tira a la papelera de reciclaje, al no ser luego aprobado por el cliente, pues no se adapta a lo deseado, bien porque no se ha sabido plasmar sus objetivos, o porque de un día para otro ha cambiado de opinión sobre lo que desea.

Para evitar la pérdida de tiempo es conveniente siempre empezar esta fase del proceso creativo abriendo el programa gráfico preferido y plasmando en una serie de gráficos vectoriales una primera visión de la interfaz a crear, basada en la información recopilada, así como de alguna de las páginas secundarias y de los formularios necesarios.

Normalmente se deben crear 2-3 versiones de la misma, que se diferenciarán en algunos aspectos clave como la disposición de los elementos, la distribución de los colores, etc. Luego se deben presentar estos gráficos al cliente, para que vea cómo van a resultar sus páginas al final y en base a ellas diga qué le gusta de la idea y qué no, qué desea que se cambie y qué desea que se deje.

Este proceso previo al paso a HTML puede parecer innecesario, pero ahorrará mucho tiempo luego y dará una base sólida sobre la que empezar a programar. Además, si en el proceso de diseño en HTML hay implicadas varias personas, con estos gráficos tendrán una idea clara y concisa de qué es lo que deben construir.

Es en esta fase también cuando se debe crear el logotipo y todas las demás imágenes que se van a necesitar luego en las pantallas generales. En cuanto al logotipo, generalmente los clientes no poseen uno en formato vectorial, si no que darán una imagen del mismo en un folleto, tarjeta, etc., que se deberá pasar al formato conveniente. Este conviene que sea de tipo vectorial, ya que entonces puede ser modificado, ampliado y reducido con toda comodidad y debe resumir el espíritu del cliente.

Una vez oída la opinión del cliente sobre las imágenes de interfaz previas se hará, también en formato gráfico, la interfaz definitiva, con lo que se tendrá ya la base final sobre la que construir con código y además se habrá conseguido dejar claro al cliente que lo que se va a hacer es lo que él desea, lo que ha aprobado de antemano, por lo que si posteriormente desea cambios en la interfaz estos serán debidos a su propia decisión y consecuentemente, deberá ser él el que cargue con los nuevos gastos ocasionados con los cambios.

### 2.1.6. Fases Operativas de un Sistema de Información Centrado en la Web

En la primera fase, denominada **de exploración**, se abordan conceptos y requerimientos iniciales, si fuera necesario se realiza un estudio de factibilidad y se planifica preliminarmente.

La segunda fase, la **de desarrollo**, es la fase esencial de replanificación, análisis y especificación de requerimientos detallados, funcionales y no funcionales [6], coordinación y control, análisis del dominio, diseño, construcción, verificación y validación, integración, aseguramiento de la calidad, entre otros procesos (como cognitivos, estéticos, de autoría, etc.) para proyectos de Hipermedia [32, 33, 35].

La tercera fase, denominada **fase operativa** o de vida útil de los artefactos, consiste esencialmente en tareas de documentación, configuración de cambios, mantenimiento y evolución de los mismos. En cuanto a la tarea de mantenimiento al menos podemos considerar tres tipos de mantenimiento: correctivo, adaptativo y perfectivo. Además, como indican los autores [37], en los sitios Web operativos se debe introducir la idea de mantenimiento de contenido, debido a que los artefactos Web son tanto orientados a la documentación como a la aplicación de software tradicional.

### 2.1.7. Diseño Modular

Generalmente un sitio web es un conjunto grande de páginas que guardan una fuerte relación entre sí, expresando el conjunto de todas ellas la naturaleza y el objetivo del sitio.

Normalmente dentro de este amplio número de páginas a crear existirán varios subgrupos de ellas que tendrán un aspecto y funcionalidad muy parecidos. Así, todas las páginas que contengan un formulario de entrada de datos serán muy similares, lo que ocurrirá también con todas aquellas destinadas a presentar los resultados de consultas, con las que encabezen una sección determinada, etc.

De esta forma, si se tiene que construir una a una y desde el principio todas las páginas del sitio, en cuanto este esté formado por un número elevado de estas, el trabajo se hará enseguida lento, arduo y pesado. Además, el tiempo es oro, vale dinero y se debe procurar crear sitios web en el menor número de días posible.

Puesto que las cosas son así, resalta entonces la idea de la modularidad a la hora de construir las páginas. Se entiende por **diseño web modular** aquel que parte de unas plantillas o páginas modelo, que van a definir el aspecto y el diseño interno de un completo grupo de páginas similares. Además, el diseño modular conlleva la reutilización, es decir, el uso repetido de mayor número de elementos de diseño, como pueden ser las imágenes de las páginas.

Siguiendo este diseño modular lo primero que se hace es crear la página principal, aquella que va a dar entrada al sitio, la primera página importante que se va a encontrar el visitante. Esta debe ser sin duda la más elaborada y cuidada del proyecto, ya que es la tarjeta de presentación y de ella depende que el usuario siga navegando por el sitio o que por el contrario, lo abandone.

Para su elaboración se usarán todos los datos obtenidos en la primera fase (logotipos, imágenes preestablecidas, colores, etc.) siguiendo al pie de la letra el último modelo gráfico aprobado por el cliente.

De esta se pasará a las páginas que dan acceso a las subsecciones. Deben tener todas ellas un diseño similar, siendo aconsejable que tan sólo cambien en algún color o signo representativo de la sección en concreto. Aplicando el diseño modular, se creará una plantilla (template) válida para todas estas páginas de entrada a las subsecciones, debiendo mantener en ellas la línea estética de la página principal y reutilizando en las mismas los logotipos ya usados en esta. Inicialmente se repite esta página tantas veces como subsecciones se tengan y si no se tienen los textos de contenido perfectamente definidos por el cliente, se complementarán con los titulares que identifiquen la sección y con unos textos en latín, por ejemplo, que llenen las páginas y le den un aspecto similar al que tendrán al final.

Se sigue bajando de esta forma en el árbol jerárquico de la estructura de páginas, hasta llegar a las páginas individuales últimas en la cadena. Para ellas se siguen aplicando el diseño modular, de tal forma que si se van a tener varios formularios en el sitio, todos deben basarse en una plantilla padre para estos, en la que se define si van a estar encuadrados, los márgenes que van a tener, si llevan cabeceras, etc.

Por ejemplo, se puede crear la siguiente plantilla para los formularios:

título del formulario	
primer enunciado	primer campo de formulario
segundo enunciado	segundo campo de formulario
tercer enunciado	tercer campo de formulario
enviar	borrar

Se ve que en esta plantilla se definen unas columnas de bordes laterales de 10 píxeles, un título superior para cada formulario, sobre color de fondo diferente al de la tabla y una serie de celdas de separación vertical entre el título y los campos de formulario y entre estos y los botones de envío y reseteo, dejando un margen inferior adecuado. Aquí se pone la tabla con bordes para que se vea el efecto, pero las diferentes tablas irán luego sin bordes.

Una vez definida esta plantilla, todos los formularios de las diferentes páginas del sitio seguirán este esquema de construcción y lo mismo se hará con los demás grupos de páginas similares: una plantilla para las tablas de presentación de datos, otra para los mensajes informativos o de error, etc.

Ya creadas las plantillas, se procede ahora a montar todo el sitio web, creando una variante provisional del sitio completo, aunque todavía no estén personalizadas todas las páginas del mismo y una vez montado todo el sitio, siendo totalmente navegable entre sus páginas y siguiendo todas ellas un esquema de diseño y estética común, es hora de citar de nuevo al cliente y enseñarle el resultado, para obtener una nueva opinión de él.

En este momento se le pregunta al cliente qué le parece lo que van a ser sus páginas web y si cree oportuno algún cambio, ya que una vez dada su aprobación se empezará a individualizar cada una de las páginas hasta obtener el aspecto final de cada una de ellas y la del sitio completo.

Aprobado por el cliente, se completan los textos y los campos de formulario de cada una de las páginas, con la idea clara de que los diseñadores tendrán total libertad a la hora de completar las páginas que serán estáticas, sin código de programación interno que las afecte, pero que para aquellas páginas que vayan a llevar código ASP, JSP, etc., deberán antes contar con la opinión de los programadores, ya

que tal vez necesiten ellos un tratamiento especial de la página que afecte al diseño o al código HTML de la misma.

Este es el último paso de la parte de diseño de las páginas del sitio web. Una vez completadas todas, si la aplicación precisa acceso a bases de datos o efectos dinámicos pasa a manos de los programadores, que se encargarán de dar vida a la misma. Estos tienen entonces entre manos un completo trabajo que realizar, que incluye la elección de los servidores web y de bases de datos a usar (si no vienen impuestos de antemano), la selección de los lenguajes de programación a usar y la propia implementación del código en las páginas [26, 37, 38].

### **2.1.8. Características de Portales Web**

Para obtener las principales características de los portales web se realizó un estudio para analizar las características que tienen estos, tales como: color de fondo, color de letra, ubicación de logo, uso, etc. todo esto ayudó para poder elaborar un portal con características estándar.

En la siguiente tabla se muestran las características principales de algunos Portales Web.

**Tabla 3:** Características de Portales Web

Nombre	Uso	Color Fondo	Color Letra	Ubicación Logo	Otras
<a href="http://www.reduaeh.mx">www.reduaeh.mx</a> (Universidad Autónoma del Estado de Hidalgo)	Educativa	Azul, Gris	Blanco y Negro	Central-Izquierdo	<ul style="list-style-type: none"> <li>Nombre de la Institución ubicado en la Parte Superior-Centro</li> <li>Iconos: Chat, Correo, Ayuda</li> <li>Menú de Links en parte Central-Derecha</li> <li>Información general de enlaces, avisos institucionales, noticias. En la parte Inferior-Central</li> <li>Reloj ubicado debajo del nombre de la institución</li> <li>Imágenes y pensamientos ubicados en la parte Central</li> </ul>
<a href="http://www.ucol.mx">www.ucol.mx</a> (Universidad de Colima)	Educativa	Amarillo, Verde Pistache, Blanco	Blanco, Verde y Negro	Superior-Izquierda	<ul style="list-style-type: none"> <li>Links: Parte Superior (arriba del logo)</li> <li>Menú de Links: Parte Superior-Izquierda</li> <li>Anuncios Institucionales: Parte Central-Derecha</li> <li>Buscador: Arriba de las noticias</li> <li>Imágenes con animación: Ubicado en la parte Superior.</li> </ul>
<a href="http://www.ipn.mx">www.ipn.mx</a> (Instituto Politecnico Nacional)	Educativa	Rosa, Guinda, Amarillo, Gris, Blanco	Negro y Blanco	Superior-Derecha	<ul style="list-style-type: none"> <li>Reloj y Fecha: Parte Superior-Izquierda</li> <li>Nombre de la Institución: Parte Superior-Central</li> <li>Links: Parte Superior-Central (debajo del nombre de la Institución)</li> <li>Links de Servicios: Parte Central</li> <li>Links de Noticias y datos de la Institución: Parte Inferior</li> </ul>
<a href="http://www.google.com.mx">www.google.com.mx</a>	Buscador	Blanco	Azul y Negro	Superior-Central	<ul style="list-style-type: none"> <li>Diferentes Opciones de búsqueda ubicado debajo del logo</li> <li>Caja de Texto y Botones de acción de búsqueda en la parte Central</li> <li>Idiomas de búsqueda debajo de la caja de texto.</li> <li>Links en la parte derecha de la caja de texto y debajo de las opciones de tipo de idioma.</li> </ul>
<a href="http://www.lycos.es">www.lycos.es</a>	Buscador	Blanco, Amarillo, Crema	Negro y Blanco	Superior-Izquierda	<ul style="list-style-type: none"> <li>Links: Parte Superior.</li> <li>Menú de Links: Parte Izquierda</li> <li>Noticias: Parte Central</li> <li>Entretenimiento: Parte Izquierda y Parte Inferior</li> </ul>

Dadas las características de la tabla anterior se puede concluir que la mayoría de los portales web analizados cumplen con aspectos similares en cuanto a su estructura y a su diseño, el resultado de este estudio esta plasmado en la siguiente tabla:

**Tabla 4:** Elementos principales de Portales Web

<b>Elementos</b>	<b>Ubicación</b>	<b>Tipo de Página</b>
Nombre y Logo	Superior	Educativo Buscador Entretenimiento Financiero
Menú de Links	Debajo del nombre (Izquierda)	Educativo Buscador Entretenimiento Financiero
Reloj y Fecha	Superior-Izquierda	Educativo Financiero
Anuncios e Información general de la Institución	Inferior	Educativo
Imágenes Animadas	Central	Educativo
Caja de Buscador	Superior-Central (debajo del logo)	Educativo Buscador
Promociones	Izquierda-Central	Financiero
Iconos (chat, correo y buscador)	Superior-Central (debajo del logo)	Educativo Buscador Entretenimiento
Letras: Blanco y Negro		Educativo Buscador Entretenimiento Financiero
Fondo: Azul, Gris, Amarillo y Blanco		Educativo Buscador Entretenimiento Financiero

### 2.1.9. Programación de la Web

Actualmente un portal web no es sólo un conjunto de páginas con un tema común, formateadas con HTML, si no que ésta serie de páginas forman una verdadera y compleja aplicación web, ya que desde la base que da el protocolo HTTP y el lenguaje HTML se construyen páginas que llevan insertado código en distintos lenguajes de programación y que permiten acceder a bases de datos, realizar verdaderas transacciones comerciales con los visitantes, efectuar operaciones bancarias, etc.

Pero la transformación de un conjunto de páginas web estáticas en una aplicación en Internet conlleva una serie de factores a tener en cuenta, desde el punto de vista del diseño, que se puede ver afectado enormemente por el código oculto, hasta el aspecto económico, ya que la inclusión de este tipo de lenguajes acarrea un mayor tiempo de construcción del sitio web, la entrada en escena de programadores capaces de construir la aplicación y la necesidad de un soporte físico (hardware) capaz de manejar este tipo de páginas. Todo ello repercute en la dificultad y en el precio del sitio web que se debe construir.

El primer paso que se debe dar, en lo que a la programación web se refiere, es determinar qué tipo de lenguaje se utilizará para conseguir el dinamismo de datos, ya que esta decisión será la base del desarrollo posterior. Además, el lenguaje usado va a determinar el tipo de máquinas físicas que se van a necesitar. Esta interrelación lenguaje-máquinas puede tomar tres enfoques diferentes:

**1) Se tiene predeterminado el hardware a utilizar.** Este es el caso más habitual a la hora de desarrollar un sitio, ya que normalmente el cliente va a tener contratado un espacio web en un ISP específico y por lo tanto el tipo de servidores a usar será el que disponga este proveedor y será este el factor que marque el lenguaje que se tendrá que implementar en las páginas. Así por ejemplo, si la empresa alojadora dispone de servidores Windows NT Server, lo más normal es que imponga el usar ASP; si es un servidor IPlanet (Netscape-Sun), normalmente se debe trabajar con JSP y Servlets y así sucesivamente.

**2) Se tiene predeterminado el Lenguaje a usar.** En este caso lo normal es que el cliente tenga ya claro en qué lenguaje determinado o qué tecnología desea usar en su sitio, bien porque ha encargado un estudio previo en el que le han aconsejado el uso de estas tecnologías concretas o porque son éstas el medio adecuado para obtener el resultado final que busca. En este caso la principal preocupación será determinar si ese lenguaje posee todas las características necesarias para poder realizar el trabajo específico que se va a necesitar en las páginas.

**3) Se tiene Libertad total.** Este último caso, el menos común, se da cuando el cliente no dispone todavía de un espacio contratado para su sitio web y encarga que se elija uno. Entonces se puede estudiar el conjunto hardware-software que mejor se adapte a las necesidades de las páginas que se deben crear.

Sea como sea, lo importante es que todas estas consideraciones van a afectar mucho al tiempo de creación del sitio, a la funcionalidad final del mismo y al precio total de la aplicación. Es por esto que el estudio pertinente debe hacerse en una primera fase, que se corresponderá con la de diseño de los esquemas iniciales, ya que tanto el tipo de requerimientos necesarios como el presupuesto final debe ser aprobado por el cliente antes de empezar a construir nada.

### **2.1.10. Relación Diseño-Programación**

A veces es necesario el uso de diferentes lenguajes de programación en las páginas y el uso de diferentes tecnologías y servidores, lo cual puede llegar a afectar el diseño Web y el código HTML. Lo antes mencionado depende del tipo de programación usado y de las necesidades funcionales derivadas de la propia aplicación. Pero siempre se va a ver afectado el código HTML y la estructura propia de la página por la inclusión de código de programación.

La solución se basa en hacer un planteamiento totalmente diferente del diseño HTML de cada página desde el principio, para no perder tiempo y ánimos. Debe haber reuniones previas entre diseñadores y programadores, hacer un estudio a fondo de los requerimientos funcionales de cada elemento de cada página y diseñar luego éstos de acuerdo a estas premisas.

A veces esto es imposible, por no saber de antemano la longitud de los datos que se recibirán, cuántos van a ser, etc. Esto condiciona a realizar un diseño elástico, entendiendo por tal aquel que se puede adaptar lo más posible a diferentes situaciones.

Es posible construir tablas que no usen rowspan o colspan (sobre todo el primero), sustituyéndolos por filas que tengan todos los mismos números de TDs, rellenando las vacías con imágenes transparentes de 1x1 píxeles. Se puede acordar con los programadores un tamaño máximo admisible para los campos de longitud desconocida, de tal forma que mediante código en servidor se recorten aquellos datos que sobrepasen estos límites preestablecidos. Establecer un número máximo de filas aceptables en cada tabla, pidiendo al servidor de bases de datos ese número de registros en cada tanda e introduciendo en las páginas unos botones de repaginación que permitan al usuario cargar de nuevo la tabla con los registros anteriores o siguientes.

De manera más o menos compleja se debe encontrar el punto de equilibrio necesario entre diseño y programación; pero para ello es básico un estudio inicial profundo y una estrecha colaboración entre diseñadores y programadores.

Este trabajo de colaboración mutua define unas fases propias en el proceso de creación de una aplicación web, que van a ser más o menos constantes en todos los trabajos.

### **2.1.11. Construcción "Local" de Sitio Web**

Siguiendo la pauta del Diseño Modular se sugirió crear una maqueta inicial navegable, formada por plantillas iguales. Es hora de personalizar cada una de las páginas basándose en la información que ha suministrado el cliente y para ello es necesario conjugar diseño y programación, con objeto de crear

páginas lo más estables posible, que no vayan a sufrir cambios posteriores en el diseño por causa del código de servidor.

Se toma una página concreta y se reúnen los programadores con los diseñadores, estudiando el tipo de código que va a precisar, tanto desde el punto de vista del cliente (navegador) como del servidor. Qué tipo de datos va a llevar, la longitud en caracteres de cada campo, cuantas filas van a tener sus tablas, etc. y sólo cuando se esté seguro del contenido completo de ella se comienza la construcción.

Una vez creada la página con HTML y JavaScript se incluye el código de servidor. Este código de servidor requiere para ver sus resultados de el enlace con las páginas de un servidor, por lo que se deberá instalar un servidor en una de las máquinas accesibles por red y este servidor web debe ser lo más parecido posible al que al final va a alojar el sitio web, ya que diferentes servidores tienen distintas particularidades.

Normalmente los enlaces que llevan las páginas se deberán pasar de relativos (`../images/punto.gif`) a absolutos (`/images/punto.gif`). Estos enlaces absolutos parten de la carpeta raíz del sitio web, es decir, especificaremos cada link usando una ruta que parte siempre desde el mismo punto.

Al introducir el código necesario para la página (ASP, JSP, etc) se debe probar para ver cómo queda y esta prueba, lógicamente, debe hacerse accediendo a la página mediante el protocolo http, es decir, haremos peticiones directas al servidor web desde el navegador. Es como si se accediera a cualquier página de Internet, con la única diferencia que en este caso accedemos vía Intranet. Para ello debemos configurar nuestro navegador para que las peticiones a las direcciones web del servidor local no las realice a través del Proxy (si estamos en una red).

En caso de que la página se vea mal en el navegador, será necesario corregir sus fallos de diseño, adaptando siempre éste al código de servidor necesario. Y lo mejor para ello es ver qué es lo que está recibiendo del servidor. Se carga la página resultante y pulsando el botón derecho del ratón se activa "ver código fuente", seleccionando todo el código de la página y se guarda como página web (extensión .html). Editamos este fichero y se le cambian las rutas a relativas (para poder verlo en local), corrigiéndose los errores, hasta que se vea como es debido. Por último, se pasan estos cambios a la página original, comprobando que se ve bien.

Este proceso se debe realizarlo con todas y cada una de las páginas que formen el sitio web, con lo que al final se obtendrá un sitio estable.

El proceso arriba descrito puede ser fácilmente automatizado desarrollando el Portal Web desde la herramienta tecnológica Macromedia DreamWeaver MX, el cual nos permite definir un sitio web local y simular un sitio web remoto, todo desde un mismo equipo que haga las funciones de Servidor de Internet y Cliente de Internet.

Otro tema a tener en cuenta es la diferente renderización que hacen los distintos navegadores que hay en el mercado. Aunque el navegador web más usado es Internet Explorer, de Microsoft, hay muchas personas que accederán a estas páginas con alguna de las versiones de Netscape Navigator. Salvo que el cliente indique otra cosa, se deberán crear páginas compatibles con Internet Explorer 5.0 y superiores y Netscape Navigator 4x y superiores. Esto obligará a usar un código HTML perfectamente construido, ya que es la única forma de que la página se vea bien en todos los navegadores y a usar normalmente una hoja de estilos y un código JavaScript para cada uno de ellos.

El trabajo de creación de las páginas es una tarea de equipo y para que este trabajo simultáneo de diferentes personas no acarree problemas será necesario aplicar técnicas concretas.

### **2.1.12. Cotización de un Proyecto Web**

Para poder cotizar por proyecto, se solicita proporcionar la mayor información posible sobre la empresa. (Fotografías, folletos, catálogos de productos, organigramas, etc.) Con estos materiales se realiza una propuesta llamada "Demo".

La información que requerimos para la realización de un "Demo" es:

- Fotografías de la empresa, de la mejor calidad posible. Los negativos o transparencias generan un costo extra que el desarrollador tendría que absorber.
- Logotipo de la empresa, fotografiado o digitalizado, preferentemente manual de imagen corporativa, para no alterar los colores corporativos y ajustar nuestros diseños a los estándares de su empresa.
- Historia de la empresa, fundación, giro, oficinas centrales y foráneas, etc. Todo lo relacionado a su empresa.
- Diagrama de flujo, Explicación en forma de diagrama (la distribución de su página) y basándonos en él los diseñadores podrán conceptualizar el desarrollo de una forma más clara en base a sus necesidades.
- Catálogo de Productos o Servicios.

## 2.2. Calidad Web

La calidad es un concepto y una actitud que ahora, más que nunca, se pretende alcanzar y conservar para mantenerse dentro de un nivel competitivo en el ámbito profesional [68].

Se define a la calidad de un ente como al conjunto de las características de dicho ente:

- *Proceso: es el ente compuesto posiblemente de otros subprocesos y actividades, usado para producir artefactos;*
- *Artefacto: es el ente temporal o persistente que representa al producto de realizar un proceso.*
- *Recurso: es un ente requerido como entrada por un proceso para producir alguna salida especificada (recursos de un proyecto son: humanos, monetarios, materiales, tecnológicos, temporales).*

Todo esto le confiere la capacidad de satisfacer requerimientos o necesidades explícitas e implícitas, en consideración de un perfil de usuario. En el mismo sentido, podemos alternativamente describir a la calidad como a una característica de muy alto nivel, que se la puede medir mediante métricas directas e indirectas y procesos de agregación, de utilidad en el proceso de evaluación y comparación de ciertas propiedades y relaciones atribuidas a un ente.

En este sentido Fenton y Pfleeger [12], realizan una interesante observación con respecto a la idea de calidad relacionada con pocos defectos. Comentan que bajo ciertas circunstancias las medidas basadas en defectos pueden ser útiles, pero que no siempre se puede asumir que sean indicadores exactos de calidad (como percibidos por los usuarios). La limitación subyace en que los defectos descubiertos durante las actividades de evaluación, inspección o compilación de programas, pueden llegar a no producir fallas en la fase de operación. Así, un alto nivel de defectos no siempre es un indicador de baja calidad y un bajo nivel de defectos no siempre podrá significar un alto nivel de calidad.

Por otra parte, la Organización de Estándares Internacionales ISO, en trabajo conjunto con IEC, definen seis características de muy alto nivel que describen, con un mínimo de solapamiento, a la calidad del producto y que son:

- Usabilidad
- Funcionalidad
- Confiabilidad
- Eficiencia
- Portabilidad
- Mantenibilidad

*“Esas características proveen una línea base para ulteriores refinamientos y descripciones de la calidad del software” [20].*

Se puede citar el principio de Metas Borrosas de Gilb, que dice que: *“Los proyectos sin metas claras no alcanzarán sus metas claramente” [14]* y por otra parte, *“De acuerdo a muchos estudios realizados sobre la aplicación de métricas y modelos en ambientes industriales, las métricas para ser efectivas deben estar: 1) Localizadas en metas específicas; 2) Aplicarlas a todo el ciclo de vida de productos, procesos y recursos; 3) Interpretarlas sobre la base de las características y comprensión del contexto organizacional, entorno y metas”*

Otra definición de Calidad es la siguiente: *Conjunto de características inherentes de un producto, proceso o sistema que permiten satisfacer las necesidades y expectativas de los clientes y de otras partes interesadas (ISO 9000:2000) [66].*

Definiciones de calidad del software [46]:

- *“Concordancias con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. [17]*
- *“Definir a la calidad de software para un sistema es equivalente a definir una lista de atributos de calidad de software requeridos por ese sistema” [17].*
- *“Para medir atributos de calidad de software se debe identificar un conjunto apropiado de métricas” [17].*
- *“Las métricas de software miden atributos específicos de un producto de software o un proceso de desarrollo de software”, considerados como atributos de un recurso [15].*
- *“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”.*

El aseguramiento de la calidad se realizara a través de un conjunto de actividades planificadas y sistematizadas necesarias para dar la confianza de que el producto o software final satisface los requisitos de calidad [46].

Se asegura que el sistema es de calidad al ser considerado en todos los estados de su evolución, ya que si se considera al término será demasiado tarde, porque un mal proceso de desarrollo producirá como resultado un sistema falto de calidad, que en cualquier momento se tendrá que volver a desarrollar y solucionarlo generando costos no previstos.

Los aspectos que garantizan un producto final de calidad son: elección de metodología y modelos, utilización de técnicas y herramientas de análisis y diseño, programación, fase de prueba y evaluación, aspectos de documentación del software y de los cambios, procedimientos que se ocupan para ajustarse a estándares, mecanismos de medida (métricas), registros de actividades de auditoría y realización de informes [46].

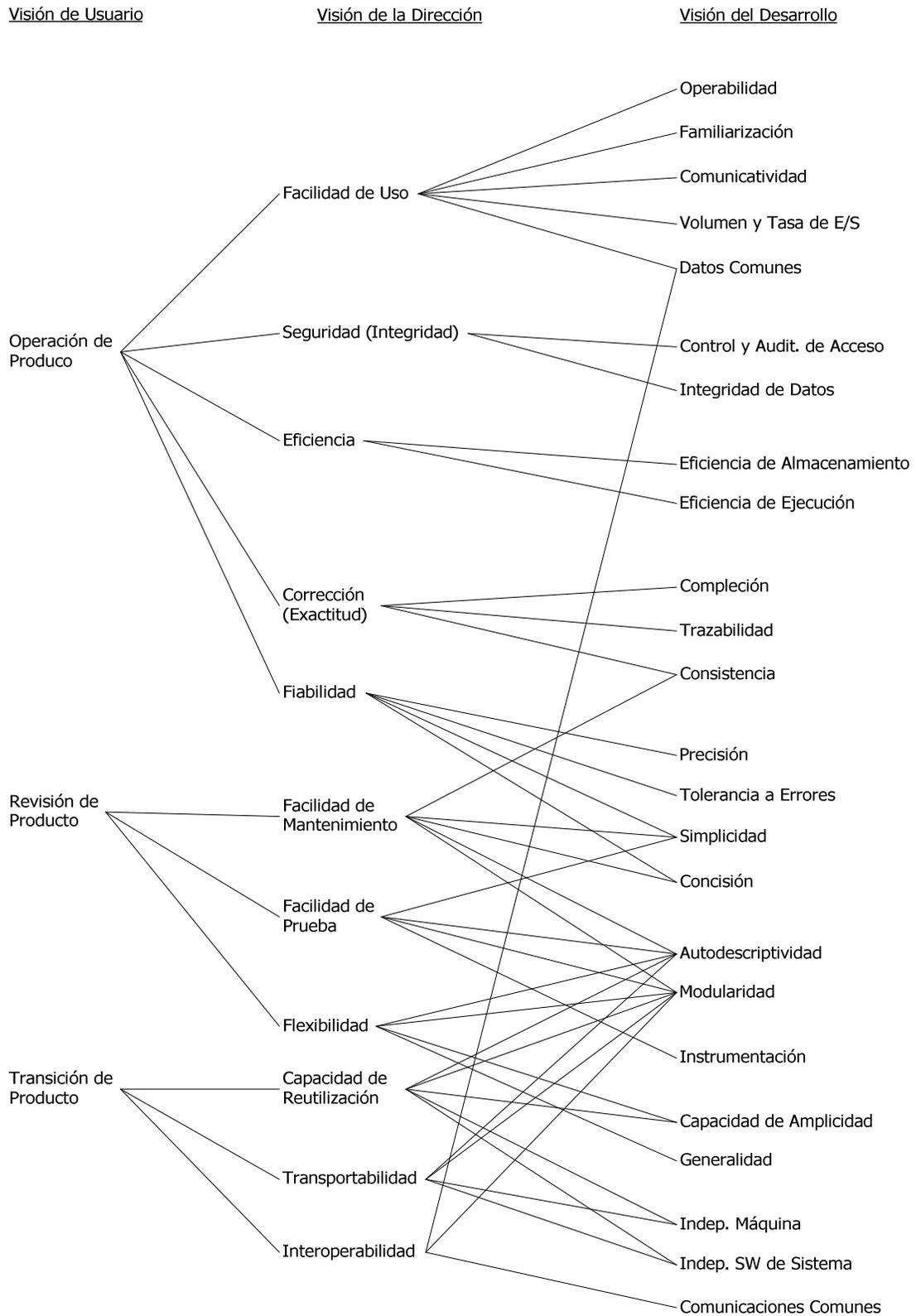
Para asegurar si un sistema es de calidad debemos de cerciorarnos que cumple con ciertas características aplicables al sitio, conociendo este conjunto de atributos permitirán comenzar su análisis. Conocer los métodos de evaluación y verificación, permitirá aplicar un enfoque sistemático, disciplinado y planificado, con la finalidad de mejorar la comprensión y la calidad de los sistemas Web.

### **2.2.1. Los Primeros Modelos de Calidad del Producto**

Desde fines de la década del 80's e inicios de la del 90's se ha puesto mucho énfasis en los conceptos de calidad de producto y satisfacción del usuario desde diversos enfoques y particularmente a la valoración y certificación de la calidad de procesos con los bien conocidos marcos conceptuales de evaluación como CMM y SPICE (entre otros), sin embargo, también es conocido que los modelos de calidad ya eran reconocidos en la comunidad científica a fines de la década del 70 como los descriptos por McCall [28] y Boehm [1].

Estos modelos describen a la calidad del producto usando un enfoque de descomposición top-down. En la Fig. 8 se puede apreciar el modelo de calidad de McCall; el modelo de McCall fue originariamente desarrollado para la Fuerza Aérea de US y se promocionó su uso para evaluar la calidad del software dentro del DoD. No obstante, otros modelos se han usado dentro del mismo departamento.

Los atributos claves son llamados en el modelo de McCall, factores de calidad (ver columna Visión de la dirección en la Fig. 8).



**Fig. 8:** Modelo de Calidad de Software de McCall [28]

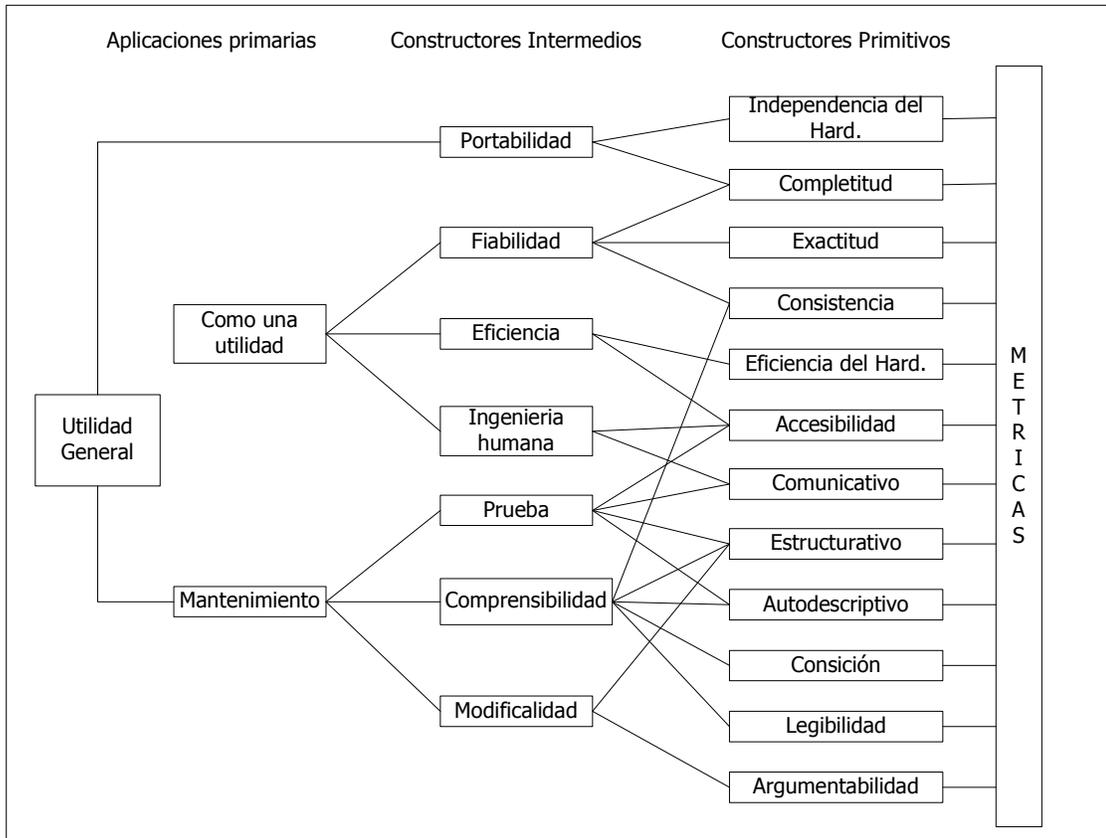
El modelo de Boehm mantiene el concepto de los tres niveles de calidad en el que los factores están compuestos por criterios y estos por métricas (preguntas).

El modelo analiza al software desde dos perspectivas: la perspectiva de la utilidad y aquella de la mantenibilidad. El modelo puntualizó que el producto de software primeramente debe ser útil; es decir, que pueda servir para un fin u objetivo, que tenga un provecho. Por tal razón, planteó las dos perspectivas: la del usuario final que recibirá los beneficios de la utilidad y la del usuario de mantenimiento que adaptará el producto de acuerdo a los requerimientos de este usuario final. De lo anterior, derivó la jerarquía de tres niveles. En el primer nivel, están las propiedades del software que pueden ser cuantificables, considerando el punto de vista del usuario final. En este caso es preciso que el programa sea portable (Portability) que se implemente en cualquier plataforma de hardware o sistema operativo; fiable (Reliability), eficiente (Efficiency), la ingeniería humana (Human engineering) que se define como la capacidad de una perfecta comprensión del software por parte del usuario. Desde la perspectiva del usuario de mantenimiento está la habilidad del programador para probar, entender y modificar el software. Se plantea los criterios de prueba (Testability), la claridad (Understandability) y la facilidad de modificación (Modificability).

Las propiedades del software se subdividen en los criterios que aparecen en el segundo nivel y que muestra las subpropiedades que facilitan su evaluación.

Finalmente las métricas se encuentran en el tercer nivel, que definen los atributos que realmente se miden en la evaluación.

En la Fig. 9 se puede apreciar la visión del modelo de calidad del producto según Boehm.



**Fig. 9:** Modelo de Calidad de Software de Boehm [28]

## 2.2.2. Otros Modelos o Marcos Conceptuales de Descomposición

En este punto es importante resaltar otros marcos conceptuales para representar factores, atributos y métricas como así también el modelo subyacente de descomposición.

### 2.2.2.1. Modelos de Calidad de Productos en los Estándares ISO 9126 e IEEE 1061

Al igual que los modelos anteriores, éste se estructura en dos niveles. En el primero están las características que se definen como las propiedades del software, que permiten describir y evaluar su calidad.

En el segundo están las subcaracterísticas definidas como las particularidades que se generan de las características y describen los métodos para evaluar los requisitos implícitos y explícitos preestablecidos por el productor y por el cliente respectivamente [52].

Según el estandar ISO 9126 (o ISO/IEC 9126), describe a la calidad del software, con mínimo solapamiento, a partir de seis características generales. "Esas características proveen una línea base para

*posteriores refinamientos y descripciones de la calidad del software'*: Funcionalidad (Functionality), Fiabilidad (Reliability), Usabilidad (Usability), Eficiencia (Efficiency), Mantenibilidad (Maintainability), Portabilidad (Portability), las subcaracterísticas y los atributos (Ver Tabla 5). Las seis características anteriores se eligieron de acuerdo a los siguientes requerimientos:

- Para que conjuntamente cubran todos los aspectos de la calidad del software que define la ISO. Para describir la calidad del producto con un mínimo de superposición. Para acercarse tanto como sea posible a la terminología establecida. Para formar un conjunto de, no mas de seis a ocho características por razones de claridad y manejo. Para identificar áreas de atributos de productos y después refinarlas. La Fig. 6 muestra el Modelo de calidad para la Calidad Interna y Externa.

**Fig. 6** Modelo de calidad para la Calidad Interna y Externa

Las características fueron mencionadas previamente y ahora son definidas en la Tabla 5:

**Tabla 5:** Características y subcaracterísticas de calidad con una descripción sintética conforme al estándar ISO/IEC 9126

Característica	II. Pregunta Central	III. Subcaracterística	Pregunta Central
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el que...?	Adecuación	¿Tiene el conjunto de funciones apropiadas para las tareas especificadas?
		Exactitud	¿Hace lo que fue acorde en forma esperada y correcta?
		Interoperabilidad	¿Interactúa con otros sistemas especificados?
		Conformidad	¿Esta de acuerdo con las leyes o normas y estándares, u otras prescripciones?
		Seguridad de Acceso	¿Previene accesos no autorizados a los datos y programas?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?	Nivel de Madurez	¿Con qué frecuencia presenta fallas por defectos o errores?
		Tolerancia a Fallas	Si suceden fallas, ¿como se comporta en cuanto a la performance especificada?
		Recuperabilidad	¿Es capaz de recuperar datos en caso de fallas?
Usabilidad	¿El software, es fácil de usar y de aprender?	Comprensibilidad	¿Es fácil de entender y reconocer la estructura y la lógica y su aplicabilidad?
		Facilidad de Aprender	¿Es fácil de aprender a usar?
		Operabilidad	¿Es fácil de operar y controlar?
Eficiencia	¿Es rápido y minimalista en cuanto a uso de recursos, bajo ciertas condiciones?	Comportamiento con respecto al Tiempo	¿Cuál es el tiempo de respuesta y performance en la ejecución de la función?
		Comportamiento con respecto a Recursos	¿Cuántos recursos y durante cuánto tiempo?
Mantenibilidad	¿Es fácil de modificar y testear?	Analisabilidad	¿Es fácil diagnosticar una falla o identificar partes a modificar?
		Modificabilidad	¿Es fácil de modificar y adaptar?
		Estabilidad	¿Hay riesgos o efectos inesperados cuando se realizan cambios?
		Testeabilidad	¿Son fáciles de validar las modificaciones?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?	Adaptabilidad	¿Es fácil de adaptar a otros entornos con lo provisto?
		Instalabilidad	¿Es fácil de instalar en el ambiente especificado?
		Conformidad	¿Adhiere a los estándares y convenciones de portabilidad?
		Reemplazabilidad	¿Es fácil de usarlo en lugar de otro software para ese ambiente?

**2.2.2.2. Modelo de Calidad conforme al Estándar ISO 9126-14**

Las subcaracterísticas pueden ser medidas por métricas internas o externas, definiendo los conceptos de calidad interna, externa y calidad en uso. Una **métrica interna** de un atributo (interno) es un valor numérico que siempre involucra al ente en sí ya sea obtenido por una métrica directa o indirecta; en cambio, una **métrica externa** es el valor resultante de aplicar una métrica indirecta y siempre involucra al ente y su comportamiento con el entorno. Por otra parte, las **métricas de calidad de uso**

miden el grado con que un producto cumplimenta las necesidades de usuarios específicos para lograr las metas con efectividad, productividad, seguridad y satisfacción.

Los documentos de trabajo o borradores 9126-2 y 9126-3 proveen un conjunto sugerido de métricas externas e internas respectivamente, a ser usado con el modelo de calidad descrito en el 9126-1. Además, en el draft 9126-4 se definen las métricas correspondientes al concepto de calidad de uso.

El estándar ISO/IEC 9126: Tecnologías de la Información – Calidad de los productos software. Consta de (ver Fig. 7): [19] [21] [22] [23].

- Calidad de Proceso o Modelo de Calidad.
- Calidad Interna o Métricas Internas.
- Calidad Externa o Métricas Externas.
- Calidad en Uso o Métricas de Calidad en Uso.

**Fig. 7:** Marco Conceptual para el Modelo de calidad especificado en ISO/IEC 9126-1

Las métricas de calidad interna y externa fueron descritas anteriormente en este documento. La Fig. 8 muestra el Modelo de calidad para la calidad en Uso, este modelo es descrito de la siguiente manera:

**Fig. 8:** Modelo de calidad para la Calidad en Uso

**a) Efectividad**

Capacidad del producto software para permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud, en un contexto de uso especificado.

**b) Productividad**

Capacidad del producto software para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.

**c) Seguridad Física**

Capacidad del producto software para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.

**d) Satisfacción**

Capacidad del producto software para satisfacer a los usuarios en un contexto de uso especificado.

**2.2.2.3. El Modelo MECA.**

Alrededor de 1991, en el Centro de Desarrollo Informático en La Habana, Cuba surgió un modelo que proponía la instrumentación de las características anteriores a través de un ciclo de vida desarrollado a partir de la norma ISO/IEC12207. Al hacerlo, las características y subcaracterísticas fueron detalladas y se presentaron los atributos en formas evaluativas. Este modelo se continuó ampliando en el Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN) en México, D.F., donde se establecieron las métricas para calcular cada atributo. El modelo incorporaba la Reusabilidad como característica y ahora se la adicionado la de Calidad en Uso.

La evaluación que realiza el modelo es más consistente porque al examinar el software a través de las etapas del ciclo de vida considera el resultado de cada una de ellas como un producto diferente, asegurando así la calidad de todos los resultados del proceso de producción de software.

Otra particularidad del modelo MECA es considerar que la documentación del software forma parte del mismo, por lo que lo trata como un producto más y se evalúa en forma separada.

Desde el punto de vista de gestión, el modelo ofrece una estructura basada en roles donde se plantea la organización de la evaluación mediante un comité dedicado exclusivamente al aseguramiento de la calidad del software.

EL modelo incorpora las características de Reusabilidad y Calidad en Uso.

El modelo presenta una escala evaluativa que permite cuantificar las medidas y a partir de los resultados obtenidos clasificar y categorizar los productos de software, posibilitando un método para su certificación.

El modelo plantea la necesidad de que se cumplan determinados atributos que denomina críticos.

### **2.2.3. Seleccionando el Enfoque de Modelo de Calidad**

Además, es importante comentar las observaciones realizadas por Fenton y Pfleeger [12], en lo que respecta a los dos posibles enfoques a adoptar al momento de modelar la calidad de software:

- *El Enfoque de Modelo Fijo de Calidad:* en la que se asume que todas las características (o factores) y demás componentes necesarios para monitorear un proyecto, son un subconjunto de aquellos publicados en los modelos bien conocidos (McCall, Boehm, ISO 9126, IEEE 1061)
- *El Enfoque "Defina su Propio Modelo de Calidad":* en la que se acepta la concepción general que la calidad está compuesta de muchas características y atributos, pero en donde no se adopta un modelo de calidad establecido. Por el contrario, el modelo se define por consenso con todas las partes que intervienen, para un proyecto y producto dado. En conjunto, se define el mecanismo de descomposición (posiblemente construido en base a un modelo existente) y se acuerdan las características, atributos (y los criterios de medición) y las relaciones entre atributos, subcaracterísticas y métricas (esta estrategia pragmática ha sido encabezada por Tom Gilb [14] y Kitchenham & Walker [25]).

### **2.2.4. Comparación de los Modelos de Calidad.**

Los seis modelos de calidad planteados anteriormente tienen algo en común: la estructura arborescente de tres niveles. El nombre de cada nivel puede variar, pero en esencia son los mismos.

Es preciso mencionar las siguientes diferencias entre el modelo MECA y los restantes:

- El modelo MECA considera que la documentación del software es parte del producto.
- El modelo MECA implementa un mayor número de métricas de calidad al expandir la evaluación a través de las etapas del ciclo de vida del software.
- El modelo MECA propone la organización de un comité para el aseguramiento de la calidad del software en la empresa o institución.
- El modelo MECA implementa la Reusabilidad y la Calidad en Uso como dos características nuevas.

**Tabla 6:** Comparación de los modelos de calidad

<b>Características Subcaracterísticas</b>	<b>McCall 1977</b>	<b>Boehm 1978</b>	<b>IEEE-1061 1992</b>	<b>ISO/IEC 9126 1991</b>	<b>MECA 1991</b>	<b>SATC 1996</b>
Fiabilidad	✓	✓	✓	✓	✓	
Usabilidad	✓	✓	✓	✓	Usabilidad	
Comprensibilidad		✓	Usabilidad	(Facilidad de aprendizaje) Usabilidad		✓
Eficiencia	✓	✓	✓	Eficiencia	Eficiencia	
Uso de los Recursos						✓
Flexibilidad	✓	✓			Portabilidad	
Reusabilidad	✓	✓	Portabilidad	(Adaptabilidad) Portabilidad	Reusabilidad	✓
Generalidad		✓				
Portabilidad	✓	✓	✓	✓	✓	
Claridad		✓			Usabilidad	Comprensibilidad
Documentación		✓			Mantenibilidad	
Elasticidad		✓			Eficiencia	
Economía		✓	(En tiempo y recursos) Eficiencia			
Mantenibilidad	✓	✓	✓	Mantenibilidad	Mantenibilidad	✓
Comprobabilidad (facilidad de prueba)	✓	✓	(Capacidad de prueba) Mantenibilidad			Corrección
Modificabilidad		✓	(Correctibilidad) Mantenibilidad			
Validez		✓				
Exactitud	✓	✓			Confiabilidad	
Corrección		(Exactitud) Confiabilidad	(Corrección) Funcionalidad			✓
Interoperabilidad	✓		Funcionalidad	Funcionalidad		
Funcionalidad			✓			
Integridad	✓	✓		(Coexistencia) Portabilidad	Funcionalidad	Completitud
Ambigüedad						✓
Completitud		(Completitud) Confiabilidad	(Completitud) Funcionalidad			✓
Trazabilidad						✓
Proporción de la Completitud		(Completitud) Confiabilidad	(Completitud) Funcionalidad			✓
Volatilidad						✓
Efectividad de la Implementación		✓	(Capacidad de prueba) Mantenibilidad	(Facilidad de pruebas) Mantenibilidad	Mantenibilidad	✓
Estructura/Arquitectura		Comprensibilidad				✓

Las etapas consideradas son:

- Estudio, análisis o *investigación preliminar*,
- Análisis y determinación de los requerimientos del software,
- Diseño general y detallado,
- Prueba de aceptación,
- Operación y mantenimiento y
- Documentación de soporte del software.

Desde el punto de vista de su estructura el modelo sigue la arborescente de tres niveles y se basa en los conceptos de la norma ISO/IEC 9126.

### **2.2.5. Características de la Calidad Web**

Esta es una recopilación de criterios (más o menos objetivos) que permiten juzgar la buena calidad de un sitio web en función de los siguientes aspectos [57]:

- a) Diseño.
- b) Navegabilidad.
- c) Contenidos.
- d) Innovación.

#### **a) Diseño**

- Legibilidad – Del texto, gráficos, del centro de atención...
- Contraste y Balance – Buena composición y colocación de elementos en función de la legibilidad, los contenidos, la navegación y factores visuales. Uso del espacio.
- Uniformidad y Variedad – Empleo de elementos de identidad y de seguimiento, suficiente variación en las secciones.
- Jerarquía – Diferenciación de secciones según su importancia.
- Claridad – Facilidad para encontrar información y no uso de elementos accesorios.
- Corrección estilística – Uso correcto de estilos en el texto.
- Iconografía adaptada al Contenido – Estilo para página infantil, de negocios, clásica, en función de los contenidos y objetivos.
- Formato - Adaptado a resolución de 800x600 ó 1024x768.
- Cromatismo – Uso correcto de combinación de colores, basados en la teoría del color.
- Calidad artística / visual – Factor no muy objetivo, basado en la experiencia y la educación del ojo, pero sustentado en gran medida en factores gestálticos: Composición, equilibrio, contraste, cromatismo, ritmo...

✓ **Se debe evitar:**

- Animaciones o imágenes con fondo mal recortado o distinto al color de fondo.
- Animaciones inservibles.
- Animaciones que no se paran.
- Borde azul alrededor de las imágenes con enlace.
- Botones de navegación con un único interés visual, especialmente cuando son muy grandes.
- Imágenes de baja calidad.
- Imágenes demasiado grandes que tardan mucho en descargarse.
- Imágenes en vista previa tan grandes como la de tamaño ampliado a la que enlazan.
- Imágenes deformadas.
- Imágenes perdidas, especialmente gráficos perdidos sin etiqueta alt.
- Imágenes que no se ajustan a la pantalla sin motivo.
- Errores estilísticos del texto - Todo en mayúsculas, todo en negrita, todo en cursiva o combinaciones de estos.
- Falta de contraste (en color, en texto, en jerarquías de información, etc.)
- Fondos repetidos o que impidan leer el texto - Combinación de colores que no permiten leer el texto (legibilidad de bajo contraste).
- Fondos complejos que distraen y hacen difícil la lectura del texto.
- Necesidad de hacer desplazamiento horizontal (800 x 600 píxeles) sin ninguna función.
- Página de entrada inútil.
- Página sin punto de atención principal.
- Página con muchos puntos de atención principal.
- Tablas con bordes (excepto de tamaño de 1 píxel).
- Tablas usadas como elementos de diseño en sí, especialmente con bordes extra gruesos.
- Texto demasiado pequeño o demasiado grande sin sentido.
- Texto que ocupa toda la página - Falta de espacio.
- Falta de uniformidad, falta de alineación.
- Uso de marcos (frames), sobretodo marcos con bordes visibles.
- Uso de noticias impertinentes.
- Complejidad innecesaria.
- "Cosas" accesorias, sin sentido o inútiles.
- Cualquier cosa que parpadee, especialmente texto.
- Muchas cosas que parpadeen, sobre todo si es al mismo tiempo.
- Líneas, objetos o textos de arco-iris parpadeantes.
- Símbolos de "En Construcción", especialmente hombrecillos trabajando y barreras amarillas y negras.

- Símbolos animados de "E-mail".
- Contadores de visitas.
- Publicidad basura.
- Gran cantidad de "premios recibidos" sin sentido en la primera página.

## **b) Navegabilidad**

- Sistema de menús o navegación útil y sencillo o de fácil comprensión.
- Ubicación del usuario – Localización de la Página actual (Directorio Dinámico) y con enlace en las secciones anteriores.
- Uso de un sistema de enlaces claro y distintivo.
- Determinación evidente de un centro de atención y jerarquías de interés que ayuden a la navegación.
- Distribución lógica de la información.
- Opciones de navegación – Atrás, inicio, siguiente...
- Compatibilidad con los principales navegadores.
- Cuando el sitio lo necesite, ayuda o documentación sobre su uso.
- Información de Contacto.
- Logotipo o similar en todas las paginas.
- Funcionamiento interno adaptado a los contenidos.

Aunque hay factores imprescindibles para cualquier sitio, parece lógico que la navegabilidad no es igual en sitios grandes que en pequeños. Podríamos considerar grandes los sitios que tengan 30 o más páginas y también los sitios con Bases de datos.

Los sitios grandes tienen que cumplir además otros factores:

- Buscador.
- Mapa del web.
- Reconocimiento del usuario cuando sea necesario por el tipo de contenidos.
- Características de navegación propias y específicas en función de los contenidos.
- Guías de usuario - 1, 2 y 3, primero esto, luego aquello, se recomienda esto.

También debemos considerar características de navegación propias de páginas diseñadas en flash:

- Precarga lógica.
- Posibilidad de evitar la "intro".
- Menú útil.

- Uso de flash justificado – No todo el sitio elaborado en flash. Ni macromedia.com ni shockwave.com están contruidos totalmente en flash.
- Posibilidad de imprimir.

✓ **Se debe evitar:**

- Entrada inservible – sin contenidos ni especificaciones.
- Difícil acceso a secciones principales desde cualquier página.
- Recomendaciones sobre cambiar la resolución del navegador, el tipo de navegador u otras especificaciones del usuario.
- Enlaces o vínculos rotos.
- Enlaces en el cuerpo de texto que distraen al lector y le llevan a páginas no útiles.
- Enlaces no subrayados o no suficientemente destacados como tal.
- Enlaces que no son claros sobre donde te llevan (Por ejemplo: haga click aquí...)
- Texto subrayado que no es un enlace.

### c) **Contenidos**

- Claridad y Concisión. - Información completa y exhaustiva.
- Jerarquía de información.
- Expresión concisa y compacta. - uso de sinopsis y resúmenes
- Orden y estructura lógica. – Planteamiento, desarrollo y conclusión si procede.
- Textos ortográficos gramaticalmente correctos, estadísticamente consistentes y con reglas de puntuación correctamente empleadas.
- Los elementos de los documentos compuestos pueden ser selectivamente seleccionados (por ejemplo para impresión, almacenamiento y correspondencia).
- Contenido adaptado al grupo objetivo específico.
- Citas y fuentes de referencia marcadas como tales, atribuidas y documentadas.
- Las imágenes grandes pueden ser vistas con anticipación ("thumbnails").
- Si procede, actualización de contenidos con la fecha de su última actualización.
- Correcta comunicación con el usuario y gestión de sus quejas.
- Relevancia e interés de los contenidos y servicios.

Requisitos no indispensables, pero que actúan muy favorablemente:

1. Integraciones y funcionalidades de base de datos.
2. Contenido multimedia. - Audio y video, uso de "streaming"
3. Listados de correos, novedades, grupos de noticias, chat, etc.
4. Contenidos adaptados al usuario discapacitado.
5. Traducción del sitio a otros idiomas.

✓ **Se debe evitar:**

- Errores ortográficos o gramaticales.
- Exceso de información.
- Falta de jerarquías lógicas.
- Páginas anticuadas.
- Errores en el proceso de código.

## **d) Innovación**

Los factores de innovación pueden no considerarse como necesarios. Se les considera más como una característica de valor añadido y por eso debe ser tomada en cuenta.

- Servicios o contenidos únicos o exclusivos.
- Soluciones avanzadas o inusuales.
- Capacidad de experimentación
- Aporte de propuestas innovadoras y eficaces - Mejora sobre convenciones estipuladas.
- Uso creativo de los recursos.
- Calidad en la investigación.
- Uso de tecnología avanzada – Java, DHTML, active X, advanced coding, XML, CSS, etc.
- Uso de herramientas de compresión - RealPlayer, Flash, Beatnik.
- Impacto multisensorial.
- Fuerte impresión emocional o intelectual.
- Formatos de WAP y WebTV
- Uso de mapas interactivos y otros medios activos.
- En comercio electrónico, soluciones y servicios novedosos.

### **2.2.6. Atributos que Determinan la Calidad de los Sitios Web.**

#### **a) Usabilidad**

Atributo de los sistemas Web que actualmente es uno de los más estudiados y que ha cobrado mucha importancia, por lo cual se ha desarrollado toda una disciplina llamada Ingeniería de la Usabilidad *"un método de diseño y solución de sistemas que toma en cuenta el factor humano"*[47]. La usabilidad busca que el diseño sea útil y no solo que sea amigable o fácil de usar, donde el sistema sea capaz de satisfacer las necesidades de los usuarios.

La ingeniería de la usabilidad plantea una metodología que guíe al equipo de desarrollo del sistema, donde la aplicación Web que se construya sea capaz de alcanzar un nivel de usabilidad, donde

tratara de ser un puente entre el equipo de desarrollo y los usuarios de manera que pueda realmente usar y le sea útil el sistema Web y planteando una evaluación del sistema a través del test de usabilidad.

### **b) Funcionalidad**

Característica de alto nivel, que se puede medir mediante métricas directas e indirectas y se representa la existencia de un conjunto de funciones y comportamientos, que deberán de satisfacer un conjunto específico de requerimientos en consideración de un dominio y perfil(es) de usuario(s) [27].

Se puede definir también como *una característica de calidad que tiene como sub-factores: la capacidad de recuperación y de búsqueda de información, los servicios de búsqueda y navegación y otros servicios extras que ofrece el sistema o aplicación.*

Se entiende como todo aquello que permite que un usuario reciba lo que requiere en el momento que lo necesita y lo decida del sistema, en base a su tipo de perfil.

### **c) Fiabilidad**

El estándar (ISO/IEC fdis 9126-1), la define como *"la capacidad del software de mantener los niveles de rendimiento cuando este es usado bajo condiciones específicas"* [50]. Se puede entender entonces que es el grado en que un sistema realiza sus tareas con la menor ocurrencia de errores, la tolerancia a fallos y que mecanismos tiene para recuperarse y validar los errores y la precisión con que realiza sus funciones.

### **d) Eficiencia**

Es una característica que representa principalmente la relación entre el grado de capacidad del artefacto y la cantidad de recursos usados bajo ciertas condiciones [27]. La eficiencia dependerá de aspectos, tales como se comporta el sistema en determinado tiempo, de que recursos dispone el sistema y la forma en que utiliza y aprovecha estos recursos [46].

Se entiende que es la capacidad del software de proveer las funciones necesarias, para producir el máximo de resultados con el mínimo de recursos, tiempo y energía, bajo condiciones específicas establecidas.

### **e) Mantenibilidad**

Atributos relacionados el esfuerzo necesario para realizar modificaciones. Se define como (estándar ISO/IEC FDIS 9126-1) *"la capacidad del software para ser modificado. Las modificaciones*

*pueden incluir correcciones, mejoras o adaptaciones del software a cambios en el ambiente, requisitos y especificaciones funcionales” [50].*

Implica que tanto es analizable el sistema, la facilidad de comprenderlo, es decir, que cantidad de recursos y esfuerzos son requeridos para localizar y corregir un error, la facilidad de adaptarse a los cambios y evolución que requerirá, sin perder su estabilidad y dejar de funcionar. Si esta característica se deja a un lado la aplicación quedara obsoleta en interfaces, modelos, estructura informativa y aspectos relacionados con software y hardware en el momento de actualizarse a la nueva tecnología [2].

### **2.2.7. Otros Criterios o Atributos de la Calidad de los Sitios Web.**

Como consecuencia de las diferentes características que presentan las aplicaciones Web, los autores incluyen diferentes criterios para determinar la calidad de los sistemas Web.

#### **a) Confiabilidad**

Es una característica de calidad que representa el grado en que un artefacto esta libre de errores y/o fallas u operable bajo ciertas condiciones especificas de uso [27]. Además de ser una características que indica el grado de confianza que tiene el usuario al hacer uso de los servicios que ofrece la aplicación Web, por ejemplo, transacciones a nivel B2C (Business-to-Consumer) y B2B (Business to Business) es donde esta confianza es indispensable ya que involucra montos momentarios y evitar una falla del sistema es importante [2].

#### **b) Portabilidad**

Conjunto de atributos que se refieren a la capacidad del software de ser transferido de un ambiente a otro entorno [68]. Para realizar esta transferencia se deberá tomar en cuenta el hardware y software, el sistema deberá de tener la capacidad de adaptación, así como de las características de instalación, capacidad de coexistencia y facilidad de sustitución [50].

#### **c) Seguridad**

Característica que debe ser inherente si la aplicación maneja datos e información importante, información que debe ser protegida y al ser enviada en la red tiene que tener mecanismos de protección para no ser modificada, leída, borrada o alterada por personas no autorizadas, las zonas de acceso exclusivo en el sitio Web son importantes, donde la verificación y validación de la autenticidad, privacidad e integridad deben de ser los parámetros básicos para cumplir con esta característica, la utilización de algoritmos de claves publicas, firmas digitales y encriptación son necesarios para realizar las transacciones en Internet [2].

**d) Time to Market**

Concepto introducido para indicarnos que si el sistema Web tiene la finalidad de realizar comercio electrónico el tiempo en que se entrega y atiende al usuario o cliente es prioridad, cumplir al usuario en tiempo y forma es prioridad y representa un aspecto que permitirá que el cliente no elija a un competidor [62, 2].

**e) Escalabilidad**

Característica que indica la capacidad del sistema para ir adaptándose a los avances tecnológicos y cumplir con los nuevos requerimientos que plantean los usuarios, además de ofrecer un rendimiento a medida que se le asignen mas recursos (memoria, procesos o equipos) [75, 2].

**f) Disponibilidad**

Se define para los sistemas Web como la capacidad de tiempo en que sitio se mantiene accesible. Donde el objetivo es que el usuario acceda al sistema en cualquier instante y las limitaciones físicas y de tiempo no se lo impidan. Algunas razones para no estar disponible el acceso al sistema pueden ser las siguientes: daños en algún elemento de la red, del host, no hay acceso a la base de datos del servidor. Otras razones para no estar disponible son por que se encuentra en mantenimiento o se han implementado actualizaciones en el momento de la solicitud.

# Capítulo III



## *Tecnologías*

En un principio la construcción de un sitio web era una tarea más o menos simple. Constaba de unas cuantas páginas enlazadas entre sí de forma sencilla y cada una de ellas estaba formada por un código HTML básico, unas pocas imágenes y poco más.

Ahora la creación de portales Web se vuelve una tarea en la que no solo toma en cuenta la apariencia que proyectan sino que también requiere que la información que se publica sea presentada de forma clara y completa para los diferentes usuarios.

El desarrollo de páginas Web se ha simplificado mucho con este tipo de programas. En favor de esto programas está la simplicidad y la facilidad de manejo de los mismos, teniendo un conocimiento mínimo sobre algunos lenguajes básicos de Internet se pueden desarrollar páginas Web sin excesivos problemas, ya que los problemas siempre se producen a la hora de crear una página Web.

Este capítulo trata de las diferentes herramientas CASE para el modelado de soluciones de software, de los editores de páginas Web y de los sistemas de gestión de bases de datos que facilitan el diseño, desarrollo e implementación de las aplicaciones Web.

### **3.1. Herramientas Case para el Modelado de Aplicaciones**

Las herramientas **CASE**, que es la Ingeniería de Software Asistida por Computadora (*Computer aided Software Engineering*) participan en el modelado de funciones de proyectos, flujos de información, entidades de datos y otra información de acuerdo a los requerimientos propuestos del sistema.

Las herramientas **CASE** ayudan a facilitar el proceso de planeación. Una vez que los datos entran en la herramienta **CASE**, la información está disponible para ser manipulada. Mucha de la información que se captura durante la fase de planeación se usará también durante las etapas de desarrollo y mantenimiento del ciclo de vida del sistema [7].

#### **3.1.1. Erwin 3.0**

##### **a) Características Generales**

Erwin es una herramienta para modelar, que ayuda a diseñar bases de datos de alto desempeño para cliente/servidor y *web/intranet*, así como aplicaciones de *data warehousing*.

La herramienta Erwin además de ayudar a diseñar modelos de datos lógicos, también construye automáticamente estructuras de datos físicos con la información del diagrama. Cuando el modelo de datos está listo para usarse, simplemente se selecciona el servidor donde se quiere construir la base de datos y se eligen las opciones de generación de esquema que se quieren incorporar. En minutos, Erwin

automáticamente construye la base de datos física, incluyendo todas las tablas, índices, procedimientos almacenados, *triggers* de integridad referencial y otros componentes necesarios para manejar exitosamente los datos usados en la organización.

## **b) Diagramas**

Los diagramas de modelos de datos en Erwin se usan para generar o actualizar bases de datos relacionales, o sea, sistemas de almacenamiento de datos ordenados en tablas. Cuando se crea un diagrama Erwin, el modelo de la información se representa por entidades (gente, lugares y cosas), atributos (hechos acerca de una entidad, tales como nombre de la persona, dirección, edad, etc.) y relaciones entre entidades.

Cada entidad corresponde a una tabla en la base de datos, con instancias de entidades que corresponden a los renglones de la tabla y atributos de entidades correspondientes a encabezados de columnas. Las relaciones, usadas por DBMS (*data base management system*) para ligar renglones de datos en tablas diferentes, están representadas como frases verbales en una línea conectando a dos entidades. Cuando se actualiza una base de datos física, Erwin automáticamente genera un *script* de definición de datos SQL, para crear tablas de bases de datos, incluyendo llaves, constraints y códigos *trigger* SQL para reforzar la integridad referencial entre tablas relacionadas.

## **c) Esquema de la base de datos**

Erwin proporciona, para diseñar un modelo de datos, propiedades específicas de DBMS y el servidor de editores que permiten definir nombres físicos y propiedades para las tablas, columnas y relaciones que genera cuando se crea el esquema de la base de datos.

Cuando Erwin crea un esquema de bases de datos, genera un *script* de cliente DDL (*data definition language*) usando la sintaxis correcta de SQL para el servidor seleccionado. Se puede ver el código que genera Erwin y, si se desea, se puede modificar antes de que se cree la base de datos.

Si el servidor soporta elementos avanzados como procedimientos almacenados y *triggers* de integridad referencial, Erwin proporciona editores de plantillas especiales y macros para guardar la información en tiempos determinados para acelerar la creación de estos objetos en el servidor. En varios servidores, se pueden hasta crear objetos de almacenamiento físico como espacios de tablas y segmentos de enrolamiento de Erwin así como especificar la ubicación y parámetros de almacenamiento para las tablas de bases de datos que Erwin genera en el servidor.

#### **d) Código que genera el lenguaje**

Erwin combina bases de datos *back-end* y desarrollo de aplicaciones *front-end* en un ambiente unificado. Tiene soporte para multiclientes, Erwin genera formas de entrada de datos en Visual Basic, DataWindows de Power Builder y PROGRESS SmartObjects del mismo modelo de datos, logrando que los desarrolladores incorporen aplicaciones altamente productivas en tres de los ambientes de desarrollo de bases de datos.

Erwin extiende el editor estándar Column Property Editor de tal forma que se pueden asignar rápidamente propiedades de columna del lado del cliente, tales como tipo de control por omisión. Despliega formato y reglas de validación de cliente para cada columna y genera formas de entrada de datos en uso y otros componentes de aplicación directamente del mismo modelo Erwin que crea la base de datos *back-end*.

Para simplificar aún más el desarrollo de aplicaciones en Visual Basic, Logic Works también ofrece DataBOT(tm), un robot de software avanzado que genera dinámicamente todo el código de acceso de datos SQL requeridos, permitiendo hasta que los programadores novatos creen rápidamente aplicaciones sofisticadas de bases de datos de alto desempeño en los ambientes actuales.

#### **e) Ingeniería hacia delante y de reversa**

##### **Ingeniería hacia adelante**

El desarrollo del modelo de Logic Works usa información específica en un modelo de datos Erwin para acelerar la creación y mantenimiento de soporte, migración y documentación de bases de datos relacionales. El proceso de generar una base de datos físicos de un modelo de datos lógico se llama ingeniería hacia delante (*forward-engineering*). A la Generación de un modelo lógico a partir de una base de datos física se llama ingeniería de reversa (*reverse-engineering*). Es posible llevar a cabo la ingeniería hacia delante en Erwin (que se llama operación de diagrama) mediante la generación automática de un esquema de base de datos física directamente del modelo de datos sin codificar laboriosamente las definiciones de datos necesarias en SQL y los *triggers*.

##### **Ingeniería de reverso**

La capacidad de ingeniería de reversa de Erwin, la cual puede inferir exitosamente las relaciones entre tablas, permite que se capture rápidamente la estructura de una base de datos existente y convertirla en un modelo lógico independiente del DBMS. Se puede usar el modelo de datos para análisis detallado, se define sobre tiempo y lo distribuye como parte de la documentación requerida a través del ciclo de desarrollo. Si se desea migrar la base de datos existente de una plataforma a otra, Erwin puede hacer la ingeniería de reverso de la base de datos existente, crear un modelo de datos, modificar o

agregar nuevos elementos según sea necesario y después construir la base de datos física en cualquier ambiente de servidor de los que soporta.

#### **f) Sincronización de la Base de Datos**

El manejo de cambio comprensivo de Erwin y los elementos de generación de script de alteración facilita el almacenamiento del modelo de datos y la base de datos física de manera sincronizada. Durante la sincronización, Erwin desempeña una comparación comprensiva de todos los objetos físicos y lógicos, incluyendo definiciones de tablas y columnas, llaves, índices y parámetros de almacenamiento físico, resaltando cualquier discrepancia y permitiendo que se migren las definiciones de la base de datos modelo al modelo de la base de datos.

Si se exportan cambios de un modelo a una base de datos, Erwin genera *scripts* de alteración de SQL para actualizar el esquema de la base de datos. Erwin automáticamente revisa todas las dependencias del esquema y traduce y preserva los datos existentes cuando se actualiza el esquema, facilitando los cambios de diseño migrados para activar el desarrollo, probar y producir bases de datos.

Además de la sincronización del modelo de toda la base, Erwin soporta sincronización de modelo a modelo y modelo a *script*.

#### **g) Diseño**

Erwin también integra la base de datos en el desarrollo del proceso de la aplicación. Cuando se ligan las herramientas de desarrollo, Erwin sincroniza el *back-end* de la base de datos con las formas del *frontend*.

#### **h) Web**

Logic Works Net Results Suites integra el modelado de datos Erwin con las mejores herramientas de desarrollo actuales. Lo que provee de una solución integral para diseñar aplicaciones de Web y el servidor de base de datos *back-end*.

ERwin/NetDynamics Suite combina el ambiente de desarrollo de aplicación NetDynamics Java-based, ayuda a que los desarrolladores construyan rápidamente aplicaciones de Web de alto desempeño.

#### **i) Metodología**

Soporta metodología para diagramas de Relación-Entidad, modelos de datos IDEF1 y Yourdon/DeMarco.

## **j) Reportes**

Los modelos de datos de Erwin pueden editarse, verse e imprimirse de diferentes maneras. Se incluye un *RPTwin*, que es un generador de reportes gráfico y fácil de usar y un *Report Browser built-in* con opciones predefinidas de reportes, que proveen un control completo sobre la apariencia y contenido de los reportes. Además, una sola plantilla de interfaz permite aplicar un estándar de diseño y desplegar preferencias para todos los modelos.

### **3.1.2. ER/Estudio 2.5**

#### **a) Características Generales**

ER/Studio crea y maneja diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

#### **b) Diagramas**

La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El *Explorer Navigator* facilita el trabajo hasta con los diagramas más grandes. Se usa el *browser Explorer* para encontrar y seleccionar entidades.

#### **c) Esquema de la base de datos**

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se desnormalizan los diseños físicos para su desempeño.

ER/Studio mantiene ligas entre todos los niveles de su diseño por lo tanto se pueden mezclar cambios en cualquier dirección entre ellos. ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos.

Se pueden desplegar los modelos de datos usando la notación IDEF1X o IE. ER/Studio permite tomar por omisión las opciones para todos los diagramas así como realizar cambios al momento de la ejecución.

#### **d) Código que genera el lenguaje**

Genera otros objetos de base de datos: vistas, procedimientos almacenados, defaults, reglas y tipos de datos de usuario, lo cual ayuda a la auto ordenación de tipos de objetos para eliminar errores de dependencia al construir la base de datos.

Tiene una opción para generar código fuente o para construir bases de datos. Soporte para crear bases de datos para Servidores SQL; y otra, para incluir código SQL y verificar la creación de objetos. Además de la opción para incluir encabezados de comentarios.

#### **e) Ingeniería hacia delante y de reversa**

##### **Ingeniería hacia adelante**

Cuando ya se ha diseñado la base de datos, se puede construir o generar código fuente para todo o para parte de los diseños de la base de datos. Propiamente hace la secuencia de la creación de tipos de objetos diferentes para asegurar eficiencia y construir bases de datos libres de errores.

Actualiza una base de datos del diagrama. ER/Studio permite aplicar cambios de diseño del modelo de datos directamente a la base de datos. Cuando se comparan las diferencias entre los dos, formula una estrategia de alteración inteligente que implementa el diseño de las modificaciones mientras se preserva la tabla con los datos existentes, privilegios de objetos y dependencias en la base de datos.

##### **Ingeniería de reverso**

Cuenta con ingeniería de reverso, cuando necesite iniciar un trabajo de una base de datos existente, ER/Studio puede hacer una ingeniería de reverso al esquema completo para cualquier plataforma de bases de datos. La operación de la ingeniería de reverso extrae eficientemente definiciones de objetos y construye un modelo de datos gráfico.

#### **f) Sincronización de la Base de Datos**

**Sincronización entre el diagrama físico y el lógico.** Mezcla entre cualquier par de diagramas físicos para la misma plataforma de bases de datos. Comparación lado-a-lado de las diferencias. El usuario puede decidir que diferencias mezclar o ignorar.

**Objetos reusables.** Construir atributos reusables. Aplicarlos a atributos y columnas. Propagación global de actualizaciones. Construir tipos de datos personalizables.

**Submodelado.** Crear cualquier número de subvistas personalizables sobre un diagrama físico o lógico. Cualquier objeto puede existir en cualquier número de subvistas (relaciones de muchos a muchos entre objetos y subvistas).

Crear rápidamente subvistas eligiendo un área del diagrama.

Control independiente sobre el despliegue de la subvista, incluyendo posición del objeto, colores y letras.

Utilidad de búsqueda rápida. Editores en tabla para evitar la necesidad de poner en modo cascada los diálogos.

Diferenciación de color de llaves primarias y secundarias inherentes. Sombreado de cajas de entidad.

#### **g) Diseño**

ER/Studio ayuda a prolongar la inversión que se ha hecho. Soporta el proceso de diseño iterativo inherente en el ciclo de vida de la aplicación.

#### **h) Web**

Publicación automática en el *web*. ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.

#### **i) Metodología**

Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

#### **j) Reportes**

Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que esta disponible para presentaciones profesionales.

### 3.1.3. PowerDesigner 6.1

#### a) Características generales:

Es una herramienta para crear bases de datos y aplicaciones cliente/servidor basadas o no en Web. Permite a los diseñadores de aplicaciones complejas de cliente/servidor tener una descripción general de los procesos particulares para comprender mejor a la organización.

Exporta información del modelo físico y extiende atributos al diccionario de 4GL.

Importa atributos extendidos de PowerBuilder. Soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse, Axiant y NS-DK.

#### b) Diagramas:

PowerDesigner cuenta con herramientas para la creación y control de diagramas como son: *Off-page Connector*, que representa los flujos de entradas y salidas en un proceso, *Business Rules* que define las reglas de uso para Procesos, Almacenamiento de datos, Entidades externas y Flujos de datos; y *CRUD Matrix*, que define el efecto de un proceso de datos en términos de Crear, Leer, Actualizar y Borrar operaciones (CRUD)

#### c) Ayuda:

La ayuda de Power Designer es sensible y adecuada al contexto.

#### d) Esquema de la base de datos:

**Data Architect** proporciona capacidades de modelado de datos tradicional, incluyendo diseño de bases de datos, generación, mantenimiento, ingeniería de reversa y documentación para arquitecturas de bases de datos. Permite que los diseñadores de bases de datos creen estructuras de datos flexibles, eficientes y efectivas para usar una ingeniería de aplicación de bases de datos.

También proporciona un diseño conceptual de modelo de datos, generación automática de modelo de datos, diseño de normalización física, sistema de manejo de bases de datos múltiples (DBMS) y soporte de herramientas de desarrollo y elementos de reportes con presentación y calidad.

El diseño se realiza en dos niveles:

- **Nivel conceptual:** entidades, relaciones, dominios, tipos de datos conceptuales, identificadores y reglas de negocios. Basado en la notación de Ingeniería de Información de Martin.

- **Nivel físico:** tablas, columnas, dominios, llaves primarias, llaves foráneas, llaves alternadas, índices, constraits de integridad referencial declarativa, vistas, parámetros de almacenamiento físico, reglas de negocios, *triggers* y procedimientos almacenados.

## **e) Ingeniería hacia delante y de reversa**

### **Ingeniería hacia adelante**

Cuando ya se ha diseñado la base de datos, se puede construir o generar código fuente para todo o para parte de los diseños de la base de datos. Propiamente hace la secuencia de la creación de tipos de objetos diferentes para asegurar eficiencia y construir bases de datos libres de errores.

Actualiza una base de datos del diagrama. PowerDesigner permite aplicar cambios de diseño del modelo de datos directamente a la base de datos. Cuando se comparan las diferencias entre los dos, formula una estrategia de alteración inteligente que implementa el diseño de las modificaciones mientras se preserva la tabla con los datos existentes, privilegios de objetos y dependencias en la base de datos.

### **Ingeniería de reversa**

- Visualiza estructuras de bases de datos existentes directamente vía ODBC o usando archivos de script DDL.
- Genera el modelo conceptual del modelo físico.
- Retro-documentación de bases de datos existentes.
- Re-orientación de la base de datos existente a un DBMS diferente.

## **e) Sincronización de la base de datos:**

- Soporta más de 30 DBMS, incluyendo Sybase SQL Server, Oracle, Informix, Ingress, Sybase SQL Anywhere, Microsoft SQL Server, SQLBase, Progress, Access, Paradox, FoxPro, etc.
- Creación directa de bases de datos vía ODBC o usando archivos de scripts DDL.
- Elige la generación del modelo entero, ciertos submodelos, u objetos individuales.
- Reglas de negocios definidas de usuario pueden integrarse en triggers y procedimientos almacenados.

Para el mantenimiento de la base de datos, incluye:

- Modelos de archivo.
- Comandos alterados generados para preservar datos contenidos.
- Mantenimiento vía ODBC o a través de archivos de scripts DLL.
- Elegir para generar el modelo entero, ciertos submodelos, u objetos individuales.

### **g) Diseño**

MetaWorks es un sistema diseñado para proveer los módulos gráficos de PowerDesign con la habilidad de compartir y almacenar modelos de datos en un solo punto de control, el Diccionario MetaWorks. MetaWorks se ejecuta en una PC y almacena los modelos de datos en un servidor de bases de datos, que puede ser Sybase, SQL Anywhere o cualquier otro como Oracle, Informix, DB2, MS SQL Server y CA OpenIngres. MetaWorks provee de tres funciones principales: Data Model y Submodel Extraction/Consolidation, Project (or Dictionary) Management y Environment Administration. El MetaBrowser presenta vista de árbol en una línea jerárquica de la aplicación bajo el estudio (base de datos, proyecto, modelo, objeto y submodelo), expande o colapsa vista de objetos, crea, modifica, borra o imprime objetos seleccionados, habilita comparación entre modelos del mismo tipo, en el nivel de objeto, trabaja con listas de objetos a través de cualquier proyecto, modelo o submodelo.

### **h) Web**

Genera reporte en archivo tipo HTML.

### **i) Metodología**

Soporta Modelos Funcionales y Notaciones de Diagramas de Flujo. Modelo Funcional de Objeto (OMT) Yourdon/DeMarco Gane & Sarson SSADM (Análisis de sistema estructurado y metodología de diseño, Structured System Analysis & Design Methodology)

### **j) Reportes**

Creación flexible de reportes estructurados a través de plantilla de reportes. Estructura de árbol de elementos seleccionados para facilitar la organización. Objetos drag-and-drop con estructura de árbol para facilitar los ajustes. Salva plantillas de reportes. Vista previa del reporte antes de imprimirlo. Seleccionar un lenguaje por omisión para el reporte. Dirigir la impresión o exportar a Microsoft Word, Word Perfect, PageMaker, etc.

## **3.1.4. System Architect 4.0**

### **a) Características Generales**

System Architect es una herramienta poderosa de modelado estructurado de datos, tiene la capacidad de identificar y clasificar personal para autorizar su entrada al sistema.

Los usuarios de red trabajan en un diagrama de proyecto y una llave de registro de diccionario de datos. Soporta la mayoría de los paquetes de red incluyendo Novell, 3Com, Banyan, DecNet, LAN Manager, STARLAN y otras. Define propiedades para cualquier entrada de diccionario, incluyendo definiciones, símbolos y diagramas. Construye ligas entre varios objetos del diccionario.

Especifica y define requerimientos, planes de prueba, cambio de requerimientos, objetivos de negocios, metas y más. Especifica que símbolos o grupo de símbolos son afectados.

## **b) Diagramas**

El área de dibujo de diagramas puede ser del tamaño de 64" x 60" a 50" x 150". Es posible ver los diagramas en las siguientes opciones: tamaño actual, página completa, área usada, o porcentaje de reducción.

Permite la edición de un diagrama en cualquier modo de vista, seleccionar y mover objetos individualmente o usando el ratón para obtener la porción del diagrama que se desee y cambiar el tamaño objetos individuales proporcionalmente o no proporcionalmente usando el ratón. La herramienta *Leveling Automatically* nivela diagramas y usa un mecanismo simple para cambiar la herencia en cualquier dirección.

Automáticamente crea *Decomposition Diagrams* (Diagramas descompuestos) de la herencia del *Data Flow Diagrams* (Diagramas de flujo de datos).

## **c) Esquema de la base de datos**

System Architect proporciona todos los elementos para diseñar un nuevo sistema o modificar un sistema actual.

Es posible crear modelos lógicamente normalizados y modelos de datos físicamente desnormalizados usando el conjunto de herramientas de System Architect. También se puede crear un modelo conceptual de las entidades y especificar su relación con otras. Al avanzar el proyecto, se pueden incluir llaves primarias, atributos, reglas, restricciones de integridad referencial, *triggers* personalizados y cualquier otra información que se elija para mantenerla en el modelo.

Si se diseña un nuevo sistema usando un diccionario amplio de datos es posible especificar los requerimientos de los datos antes de comenzar el modelado, mientras se está construyendo el modelo, o después de haber completado el diseño lógico.

Si se está modificando un sistema existente es posible usar la ingeniería de reverso de SA para crear un diagrama de modelo de datos físicos para el sistema actual. SA crea automáticamente un DER de un modelo de datos físico. Entonces se puede modificar el DER, creando un modelo lógico normalizado del nuevo sistema. Una vez que se ha completado el diseño lógico, se pueden generar modelos físicos. Si se planea implementar una base de datos desnormalizada, se puede documentar el proceso de

desnormalización usando diagramas *Local View* (Vista Local). SA mantiene ligas entre el modelo lógico, las vistas lógicas y el modelo físico; por lo tanto los cambios al modelo lógico se reflejan automáticamente en el modelo físico. Al final, se tienen dos modelos físicos separados: uno del sistema actual y otro del sistema propuesto. Para cualquier tipo de proyecto en el que se este trabajando, SA proporciona flexibilidad para completar el trabajo. Se pueden elegir modos de despliegue en cualquier tiempo durante el proceso de diseño: conceptual, basado en llaves, totalmente atribuido, o despliegue físico. También, una vez que se ha completado el modelo lógico, se pueden ejecutar una serie de reglas revisadas y reportes de normalización para validar la integridad del diseño. SA prueba las Formas Normales:

Primera, Segunda, Tercera y Boyce Codd.

#### **d) Código que genera el lenguaje**

Genera archivos de definición de almacenamiento de trabajo (.WKS) y sección de pantalla (.SCS) para implementaciones de COBOL. SA/PowerBuilder Link permite el intercambio de la información de diseño entre SA y Power Builder, incluyendo DDL y atributos extensos.

#### **e) Ingeniería de reversa.**

Genera y hace ingeniería de reverso en los *triggers*, reglas, defaults, tipos de datos definidos, dispositivos y bases de datos.

Elementos de la Ingeniería de reverso de SA:

- Captura datos del sistema de bases de datos más popular: Access, DB2, Informix, Oracle, SQL Server o SYBASE.
- Ingeniería de reverso de un archivo DDL, a través del manejador ODBC, o a través de la interface directa del SQL Server.
- Genera diagramas de entidad relación que incluyen definiciones para llaves, no llaves y relaciones.
- Genera diagramas de modelo de datos físicos que incluyen definiciones para tablas, columnas y constantes.
- Crea llaves primarias, índices alternados y rutas de acceso de los índices en la estructura de la base de datos.
- Importa *triggers*, procedimientos almacenados, defaults y roles de definiciones de bases de datos existentes.

- Captura, modifica y regenera esquema usando SA RDE con el Schema Generator.
- Captura, modifica y regenera pantallas y menús de aplicaciones de Windows usando el SA RDE con el SA Screen Painter

#### **f) Sincronización de la Base de Datos**

El Generador de esquemas (Schema Generator) automatiza la creación y mantenimiento de esquemas para DBMS de SQL y 4GL SA Schema Generator traduce modelos de datos (ER o diagramas físicos DM), IDEF1X y diagramas de clases orientadas a objetos en las definiciones de los esquemas para manejo de sistemas con múltiples bases de datos. La transferencia de esquemas en la DBMS se puede llevar a cabo en vivo, vía conexión ODBC, o a través del uso de un archivo. DDL intermedio.

SA Schema Generator simplifica el desarrollo y asegura la integridad de los diseños permitiendo crear y mantener esquemas de bases de datos para una amplia variedad de administradores de sistemas de bases de datos SQL y no SQL.

Cuando se genera un DBMS múltiple, los tipos de datos definidos en el modelo de datos se apean automáticamente a los tipos correctos del DBMS seleccionado.

#### **g) Diseño**

SA cuenta con la herramienta Screen Painter, con la que se pueden diseñar pantallas y menús para aplicaciones de Windows y pantallas de caracteres para aplicaciones de COBOL. Los archivos de Windows .DLG y .MNU se pueden generar automáticamente por SA, incluyendo posición, leyenda, *hot key*, orden de tablas y número de identificación para cada control o elemento de menú incluido en la aplicación original de Windows.

Pueden crearse pantallas usando controles estándar de Windows, incluyendo botones de presión, cajas de revisión, botones de opciones, cajas combo, cajas de listas, cajas de texto.

Se pueden generar archivos de diálogos (.DLG), encabezado (.H) y forma de Visual Basic (.FRM) para aplicaciones de Windows. Usa una rutina simple de captura para crear un nuevo menú de cualquier menú de aplicación de Windows. Dibujar menús usando los elementos del menú, submenú y separadores. Agregar accesos directos para elementos del menú y submenú. Los elementos del submenú y del menú se activan usando el cursor.

Los diálogos seleccionados se abren de elementos seleccionados del menú.

## **h) Web**

No tiene publicación en Web.

## **i) Metodología**

El editor de reporte SA también incluye más de 125 reportes pre-definidos, incluyendo reportes de metodología para Diagramas de Relación de Entidad, modelos de datos IDEF1, Diagramas de Flujo de Datos para Word & Mellor, Gane & Sarson y Yourdon/DeMarco.

## **j) Reportes**

El editor de reportes SA es un elemento estándar en System Architect, esta herramienta permite especificar reportes personalizados para el proyecto con un frontend fácil de usar.

El editor de reporte SA también incluye más de 125 reportes predefinidos, incluyendo reportes de metodología para Diagramas de Relación de Entidad, modelos de datos IDEF1x, Diagramas de Flujo de Datos para Ward & Mellor, Gane & Sarson y Yourdon/DeMarco.

Reportes definidos por el usuario. Los reportes pre-escritos, también pueden ser modificados con el editor de reportes GUI. Si un nuevo reporte es necesario el reporte de GUI provee de una forma fácil y rápida de crear reportes personalizables. Todos los reportes pueden incluir una imagen del diagrama relevante. El editor de reportes GUI también provee de flexibilidad agregada para controlar las letras, bordes, orientación, encabezados y más. Cuando el reporte está listo crea un archivo ASCII y/o RTF en lugar del reporte impreso.

Se usa el Editor de Reportes SA para:

- Reportar propiedades definidas por el usuario.
- Crear reportes de matrices para analizar los datos del repositorio.
- Crear algo específico, leer, actualizar, borrar (CDRU).
- Desempeñar Análisis de Afinidad.

### **3.1.5. Rational Rose**

Rational Rose es una herramienta para "modelado visual", que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software.

Rational Rose permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio.
- Captura de requisitos (parcial).
- Análisis y diseño (completo).
- Implementación (como ayuda).
- Control de cambios y gestión de configuración (parte).

#### **a) Características principales.**

- Admite como notaciones: UML, COM, OMT y Booch.
- Realiza Chequeo semántico de los modelos.
- Ingeniería "de ida y vuelta": Rose permite generar código a partir de modelos y viceversa.
- Desarrollo multiusuario.
- Integración con modelado de datos.
- Generación de documentación.
- Tiene un lenguaje de *script* para poder ampliar su funcionalidad.
- Soporta OLE.
- Disponible en múltiples plataformas.

#### **b) Versiones de Rational Rose.**

- *Modeler*: no tiene soporte para lenguajes de programación.
- *Professional*: incluye soporte sólo para un lenguaje.
- *Enterprise*: múltiples lenguajes, incluyendo C++, Visual C++ (6.0), Visual Basic, Java, CORBA, etc.).

#### **c) Diagramas.**

Los diagramas son vistas gráficas del modelo. Rose mantiene automáticamente la consistencia entre los diagramas y las especificaciones correspondientes: si se modifica el diagrama se cambia la especificación y viceversa.

##### **Tipos de diagramas soportados por Rose:**

- de Clases.
- de Casos de uso.
- de Colaboración.
- de Secuencia.
- de Componentes.
- de Estados.
- de Actividad.

- de Despliegue.

En el árbol de todo modelo aparecen cuatro **vistas**, cada una de las cuales admite determinados tipos de diagramas:

- **Caso de uso:** esta vista define la interacción entre actores y casos de uso. Diagramas principales: casos de uso, colaboración, secuencia, actividad.
- **Lógica:** esta vista define fundamentalmente las clases del sistema y sus relaciones. Diagramas principales: clases, estados.
- **Componentes:** esta vista contiene información sobre ficheros, ejecutables y librerías del sistema. Diagrama de componentes.
- **Despliegue:** esta vista muestra la asignación de procesos al hardware. Diagrama de despliegue.

La siguiente tabla muestra una comparación de las diferentes herramientas que se utilizan para el modelado de aplicaciones web, se consideraron las características, diagramas, esquema de BD, ingeniería, sincronización diseño y metodología que cada herramienta maneja.

**Tabla 7.** Herramientas Case para el Modelado de Aplicaciones

	HERRAMIENTA				
	ERWIN 3.0	ER/ESTUDIO 2.5	POWER DESIGNER 6.1	SYSTEM ARCHITECT 4.0	RATIONAL ROSE
<b>Características</b>	Diseña Bases de Datos (BD) Cliente/Servidor, Web/intranet y Aplicaciones Data Warehousing Estructura de datos físicas Construye BD Física	Diseño de BD Lógico, bidireccionales Construcción de BD Documentación y Reportes	Crea BD . Aplicaciones Cliente/Servidor Exporta información del modelo físico	Modelo estructurado de datos Soporta paquetes de red Define requerimientos, planes de prueba, objetivos de negocios y metas	Notaciones UML, COM, OMT y Booch Chequeo semantico de los modelos Ing. De Ida y Vuelta Desarrollo multiusuario Genera Documentación
<b>Diagramas</b>	BD Relacionales Entidades, atributos y relaciones entre entidades Genera Script SQL	Diagrama de clases Modelos de Despliegue	Crea y controla diagramas: Off-page connector, Business Ruels, Crud Matrix	Leveling Automatically Diagramas descompuestos y de flujo de datos	Diagramas de Clases Casos de uso Colaboración Secuencia Componente Estados
<b>Esquema de la BD.</b>	Genera Script Cliente DDL	Diseño Lógico Normalizado Notación IDEF1X o IE	Data Architect: Modelo de datos tradicional, diseño conceptual de modelo de datos, normalización física. Nivel Conceptual Nivel Físico	Crea modelos lógicamente normalizados y modelos de datos físicamente desnormalizados	
<b>Ingeniería</b>	Modelo de Datos Lógico Ing. Hacia delante: BD Física a BD Lógica Ing. Hacia atrás: Modelo de Datos Lógico a Vd. Física	Ing. Hacia delante: Código para todo o parte del diseño de la BD Ing. Hacia atrás: Extrae definiciones de objetos Construye un modelo de datos Graf	Ing. Hacia delante: Código para todo o parte del diseño de la BD Ing. Hacia atrás: Visualiza Estructuras de BD, Genera modelo conceptual del modelo físico, Retrodocumentación de BD existentes, Reorientación de BD a un DBMS diferente	Ing. Hacia atrás: En triggers, reglas, defaults, tipos de datos definidos, dispositivos y BD	
<b>Sincronización</b>	Almacenamiento del Modelado de Datos Y BD Física Sincronizados	Diagrama Físico y Lógico Objetos reusables Submodelado	Soporta mas de 30 DBMS (Oracle, SQL Server, Progress), Creación de BD vía ODBC o usando Scripts d.C.	Generador de esquemas	
<b>Diseño</b>	Herramienta de desarrollo Sincroniza Back-End de la BD con Front- End	Diseño Iterativo inherente en el ciclo de vida de la aplicación	Meta Works	Scren Painter	
<b>Diseño</b>	Herramienta de desarrollo Sincroniza Back-End de la BD con Front- End	Diseño Iterativo inherente en el ciclo de vida de la aplicación	Meta Works	Scren Painter	
<b>Metodología</b>	Diagramas R-E Modelo de Datos IDEF1 Yourdon/DeMarco	Met.Yourdon Diagrama R/E Modelos IDEF1	Soporta modelos funcionales y notación de diagramas de flujo	Relación de Entidad Modelo de datos IDEF1 Diagramas de flujo de datos para Word & Mellor, Gane & Sarson y Yourdon/DeMarco	Modelo del negocio Captura de requisitos Análisis y diseño Implementación

## **3.2. Editores de Páginas Web**

El lenguaje de marcas HTML fue diseñado exclusivamente para dar formato a los contenidos textuales presentes en un documento web, aunque posteriormente se ampliaron las funcionalidades de sus etiquetas para permitir con ellas maquetar adecuadamente las páginas.

Con este planteamiento, los únicos medios de los que disponía el diseñador web para interactuar con los visitantes de sus páginas eran los hiperenlaces y el uso de formularios, aunque con estos últimos la interacción se limitaba a poder recoger de una forma plana los datos introducidos en sus campos.

Buscando romper esta estaticidad de las páginas web diferentes empresas y desarrolladores fueron introduciendo diversas tecnologías y lenguajes de script que permitían una interacción más amplia con el usuario. La primera de ellas digna de consideración fueron los script CGI, que mediante el uso de código escrito en C, C++ o Perl y ejecutado en el servidor web permitieron recuperar los datos de los formularios de forma dinámica y actuar en tiempo real de una forma u otra dependiendo de los mismos.

### **3.2.1. Med**

MED es un editor de texto para programadores, trabaja sobre las plataformas o sistemas operativos Windows 9x / NT4.0 / 2000 y OS/2 PMS [72].

MED es conveniente para trabajar sobre cualquier archivo de texto fuente, sirve de apoyo para los lenguajes C/C++, Java, HTML, REXX, Ada, Modula2, Pascal, Fortran, X86/MC68k ensambladores, Perl, Recortador y TeX.

MED proporciona un ambiente de desarrollo universal para editar en entornos de lenguajes de programación.

### **3.2.2. Macromedia Dreamweaver**

Macromedia Dreamweaver MX es un editor visual de páginas web, lo que también se denomina WYSIWYG, el cual proporciona además una visión del código que crea el programa de forma automática a la hora de introducir los diversos elementos en dicha página [55].

Las principales características de Dreamweaver son:

**a) Soporte para las CSS (hojas de estilo):**

Dispone de un gran número de hojas de estilo predefinidas generales, por si el usuario no se quiere meter más a fondo con este tipo de archivos, las cuales dan un aspecto de "página de Macromedia" al proyecto.

También dispone de unas plantillas en las cuales se pueden obtener efectos para los enlaces, cambiar el tipo de letra, o los colores tanto del texto como de la página, con una pequeña ventana en la cual se muestra el resultado a modo de ejemplo, antes de ser aplicado. Además proporciona sugerencia sobre los parámetros para aplicar en las CSS.

**b) Manipulación de imágenes:**

Una utilidad muy interesante es que ahora desde Dreamweaver se pueden modificar las imágenes sin tener que cambiar de aplicación. Lo que se ha integrado es una pequeña paleta con herramientas de FireWorks, con acciones básicas pero muy útiles como es el redimensionado de la imagen, el poder recortar, o girar la imagen.

**c) Compatibilidad:**

Compatibilidad de las páginas generadas para HTML sean totalmente compatibles con el XHTML. Si no deseas tener esa compatibilidad puedes hacer que el programa cree el código directamente en XHTML.

**d) Comunicación con los "Sites":**

Una de las principales características de Dreamweaver MX, es la posibilidad de crear comunicación directa mediante protocolos de seguridad, es decir cifrados, mediante SSH. El envío de los datos se hace mediante cifrado, lo que le otorga un mayor nivel de seguridad. Por supuesto esto irá en detrimento de nueva velocidad, ya que el cifrado de los datos y su comunicación se hará de una forma más lenta que la comunicación de archivos mediante FTP.

Sin duda se ha potenciado mucho el trabajo con las CSS u hojas de estilo. Una tendencia existente en estos momentos en Internet. El trabajo y la creación de estilo basados en este tipo de código se ha extendido mucho en Internet hoy en día y Macromedia se ha dado cuenta de ello.

Una base de datos es una colección estructurada de datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto monto de información en una red corporativa.

Ventajas de Dreamweaver en comparación con Notepad [44].

Dreamweaver presenta grandes ventajas en relación a Notepad, como:

- No es necesario conocer la programación HTML.
- Se insertan con facilidad objetos, con tan sólo apretar un botón.
- Presenta una amplia variedad de shortcuts.
- Rápido y sencillo manejo.
- Previsualización inmediata.
- Te permite visualizar el código (dualidad).
- Manuales de ayuda

Las principales ventajas de Notepad (Editor de Texto) son:

- No cuesta.
- Independiente de la plataforma.
- Cualquier PC la visualiza.
- Control estricto de lo que se requiere.
- Código compacto sin basura

### **3.2.3. Frontpage**

#### **a) Generalidades.**

FrontPage es un programa de edición y administración de páginas Web que no requiere conocimientos de programación pero muy potente para desarrollar los sitios Web. Con los conocimientos de Microsoft Word es muy fácil editar en FrontPage [73].

FrontPage es una herramienta para crear, modificar y probar las páginas del World Wide Web y a medida que se incluyen fondos, imágenes, tablas, textos y vínculos el editor los muestra tal y como aparecerán en la red sin conocimientos previos de HTML, que es el sistema de programación en el que quedan archivadas las páginas WEB.

#### **b) Traslado a Internet.**

Con un programa FTP, también gratuito, de transmisión de archivos sólo nos queda trasladar a la red cuantas veces queramos nuestra carpeta íntegra o bien una de las subpáginas que hemos fabricado por supuesto con todas las imágenes y fondos nuevos.

En el momento de contratar el alojamiento en internet tenemos que pedir nuestro "host name", el "user ID" y la contraseña que serán necesarios para hacer la configuración del programa FTP.

La página principal hay que denominarla "index.htm"

En la siguiente tabla se puede ver una comparación que se hizo entre algunos de los editores de páginas web que existen, en dicha tabla se aprecia el tipo de editor, las características que tienen, las plataformas sobre las cuales trabaja y las ventajas que se obtienen con el uso de estos editores.

**Tabla 8.** Editores de Páginas Web

Editor	Tipo de editor	Características	Plataformas	Ventajas
<b>MED</b>	Editor de Texto	Ambiente de desarrollo universal Edita en entornos de lenguajes de programación	Windows 9x/NT 4.0/2000	No cuesta Independiente de la plataforma Cualquier PC la visualiza Control escrito de lo que se requiere Código compacto sin basura
<b>MACROMEDIA DREAMWEAVER</b>	Editor Visual	Soporte para las CSS (Cajas de estilo) Manipulación de imágenes Compatibilidad Comunicación con las "Sites"	Windows 9x/NT 4.0/2000	No es necesario conocer la programación HTML Se insertan con facilidad objetos Amplia variedad de Shortcuts Visualización código (Dualidad)
<b>FRONTPAGE</b>	Editor de texto y visual	Crea, modifica y prueba las paginas Web Programa FTP	Windows 9x/NT 4.0/2000	Programa de edición y Administración No requiere conocimientos de programación Potente para desarrollar los sitios Web

### **3.3. Sistemas de Gestión de Base de Datos**

En el siguiente apartado se tratan los sistemas de gestión de bases de datos, tales como: MySQL, SQL Server y Oracle.

#### **3.3.1. MySQL**

MySQL Database Server es la base de datos de código fuente abierto. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del

software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

Las principales características de MySQL son [61, 69]:

- MySQL es un sistema de administración de bases de datos. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.
- MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.
- MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones.

Porqué usar MySQL Server: MySQL Database Server es muy rápido, confiable y fácil de usar. MySQL Server también tiene un práctico set de características desarrollado en cercana cooperación con los usuarios. MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, MySQL Server ofrece hoy una rica variedad de funciones. Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para acceder bases de datos en Internet.

### **3.3.2. SQL Server 7.0**

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft.

SQL Server 7.0 ha dado un paso adelante definitivo en rendimiento, escalabilidad y fiabilidad, proporcionando a las empresas la oportunidad de crear soluciones inteligentes y coherentes con el mundo real.

Las principales características de SQL Server 7.0 son [69]:

#### **a) Escalabilidad:**

SQL Server 7.0 fue planificado para organizar más datos, transacciones y usuarios con facilidad.

- **Escalable desde ordenadores portátiles a un cluster de multiprocesador.** SQL Server cuenta con la capacidad de escalar las necesidades de su organización. Con SQL Server 7.0, las aplicaciones con las que trabaja la organización pueden adaptarse para organizar terabytes de datos y miles de usuarios.
- **Procesador de consultas.** SQL Server 7.0 también incluye un procesador de consultas rediseñado que proporciona un potente soporte para grandes bases de datos y consultas complejas.
- **Bloqueo dinámico de filas.** SQL Server 7.0 soporta el bloqueo dinámico de filas, lo que lo convierte en la mejor elección en sistemas de alta calidad de online transaction processing (OLTP) y Business Intelligence. El administrador de bloqueos ajusta dinámicamente los recursos que utiliza para bases de datos de gran tamaño, lo que significa que no necesitará ajustar la configuración del servidor manualmente.

#### **b) Facilidad de uso:**

Microsoft diseñó SQL Server desde un principio para facilitar a los administradores la construcción, gestión e implementación de aplicaciones de negocios. Automatiza las operaciones de base de datos estándar. Además, las nuevas y sofisticadas herramientas simplifican la gestión de las complejas operaciones de gestión.

- **Auto-gestión dinámica.** Esta característica clave de SQL Server 7.0 automatiza muchas de las tareas rutinarias. Los recursos de memoria y de bloqueo se ajustan automáticamente, el tamaño de los archivos crecen y disminuyen automáticamente y las características de auto-ajuste adaptan dinámicamente los recursos para cambiar cargas de trabajo.
- **Herramientas de perfil y ajuste.** Estas herramientas simplifican el proceso de localización y arreglo de problemas de base de datos permitiéndole capturar y reproducir la actividad del servidor. Por otra parte, un nuevo analizador de consultas gráfico hace posible a evaluar consultas más fácilmente y con detenimiento y el nuevo Index Tuning Wizard facilita el proceso de ajuste de índices.

- **Gestión de sitios múltiples.** Experimente la potencia y flexibilidad de la gestión de múltiples servidores. Con SQL Server Enterprise Manager, puede diseñar un servidor master que comunique y distribuya mensajes de tareas, alertas y eventos a servidores determinados. Además, en esta versión, puede realizar cambios entre grupos de servidores utilizando operaciones de arrastre-y-suelta y comandos únicos.
- **Amplia disposición de opciones de replicación.** SQL Server 7.0 facilita en gran medida la instalación, la administración, la monitorización y la localización de averías. El modelo de replicación hace posible que los usuarios trabajen con datos de forma libre e independiente. Con conexión o sin ella, los usuarios pueden modificar copias distribuidas de una base de datos y obtener posteriormente el trabajo combinado y sincronizado en un solo resultado uniforme. Otra característica notable para la utilización avanzada de SQL Server es la replicación por actualización, en la que los datos replicados por SQL Server 7.0 pueden modificarse en varios sitios.

### c) Internet, Intranet y Comercio:

Las innovadoras características y la integración transparente con Microsoft Windows NT® y Microsoft BackOffice® hacen de SQL Server 7.0 un factor crucial en su estrategia de Internet, intranet y comercio electrónico.

- **Búsqueda a texto completo.** SQL Server 7.0 suprime la necesidad de costosos dispositivos de ampliación para gestionar búsquedas de bases de datos a texto completo. Soporta la búsqueda lingüística y le permite crear índices especiales de palabras y frases relevantes en columnas seleccionadas de tablas seleccionadas.
- **Asistente web.** Contiene una opción de asistentes y de formateo que mejora el SQL Server Web Assistant, para que pueda generar archivos de HTML desde datos de SQL Server.
- **Replicación en Internet.** La replicación en Internet es más sencilla que nunca gracias a las suscripciones anónimas y al soporte incluido para la distribución en Internet. SQL Server 7.0 soporta programas que utilicen Active Server Pages (ASP).

### d) Sistemas fijos, móviles y distribuidos:

Microsoft diseñó SQL Server 7.0 de forma que las organizaciones pudieran dar a sus empleados y clientes la capacidad de trabajar con datos de forma fiable desde la oficina, en Internet o de viaje.

- **SQL Server Desktop.** Con SQL Server 7.0 para Microsoft Windows® 95, Microsoft Windows 98 y Microsoft Windows NT Workstation, puede contar con un sistema de gestión de bases de datos relacionales totalmente equipado (RDBMS) que presenta un funcionamiento correcto en aplicaciones móviles y de estación de trabajo. Un código único para todas las plataformas (desde un portátil con

Windows 95 a sistemas con clusters con Windows NT Server, Enterprise Edition) aporta una compatibilidad de aplicaciones total.

### 3.3.3. Oracle

Es el manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.

Es el conjunto de datos que proporciona la capacidad de almacenar y acceder a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas.

Es el mayor y más usado Sistema Manejador de Base de Datos Relacional (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Un modelo relacional posee tres grandes aspectos:

*Estructuras:* Definición de objetos que contengan datos y que son accesibles a los usuarios.

*Operaciones:* Definir acciones que manipulen datos u objetos.

*Reglas:* Leyes para gobernar la información, como y quien manipular.

Una base de datos relacional simplificada y definida como un modelo de información es estrictamente visualizable por los usuarios mediante tablas. Una tabla está compuesta por una matriz bidimensional de filas y columnas. En cualquier ocasión la información es cambiada en una base de datos relacional, cualquier información es el resultado de una consulta presentada por el usuario en el formato filas/columnas.

### **a) Estructura Física y Lógica.**

Las estructura física tales como los archivos del sistema operativo, son almacenados tangibles como son cintas magnéticas, discos y otros. A cada archivo le corresponde un espacio en el sistema operativo. Oracle requiere de varios archivos para su funcionamiento, los cuales conforman su estructura física.

A la estructura lógica le corresponde un espacio por unidad, pero sus limitaciones son independientes de las localizaciones de espacio físico.

### **b) Uso de Memoria.**

El uso de memoria en el RDBMS Oracle tiene como propósito lo siguiente:

- Almacenar los códigos de los programas para empezar a ejecutarse.
- Almacenar los datos necesarios durante la ejecución de un programa.
- Almacenar información sobre como es la transferencia entre procesos y periféricos.

### **c) Programas y Archivos que compone a Oracle.**

Un RDBMS Oracle esta compuesto por tres partes principales, que son:

1. El Kernel de Oracle
2. Las instancias del Sistema de Base de Datos.
3. Los Archivos relacionados al sistema de Base de Datos.

#### **El Kernel de Oracle:**

El Kernel es el corazón del RDBMS Oracle, el cual maneja las siguientes tareas:

- Manejar el almacenamiento y definición de los datos.
- Suministrar y limitar el acceso a los datos y la concurrencia de los usuarios.
- Permitir los backup y la recuperación de los datos.
- Interpretar el SQL y PL/SQL.

Así como el Kernel es un sistema operativo, el Kernel Oracle es cargado a la memoria al inicio de las operaciones y es usado por cada base de datos existente en el equipo.

#### **EL PGA (Program Global Area):**

Es también llamado Process Global Área, consta de datos e información de control de los procesos, asegurando el uso correcto de estos. El PGA contiene información acerca de las conexiones y los

procesos que se realizan en Oracle, su tamaño es variable en longitud, pero no es dinámico. El PGA se activa al conectarse un usuario.

### **El SGA (System Global Area):**

Se puede llamar Shared global área, se podría definir como una serie de buffers en memoria residente, a través de la cual todas las transacciones y el almacenamiento de dato fluyen. El SGA es localizado en memoria al iniciarse una instancia y desaparece al bajarla. Su tamaño no puede ser cambiado, pero si puede ser visto con el comando "SHOW SGA" en el SQL\*DBA. Su longitud esta definida por los parámetros del archivo de iniciación INIT.ORA.

Esta Compuesto por:

- Diccionario Cache
- Los Redo Log Buffers
- Los Database Buffers

- ***Diccionario Cache:***

Es un conjunto de tablas que son usadas para proveer información asociada con la base de datos, donde se definen los nombres de usuarios, privilegios, objetos, retenciones y espacios que conforman un RDBMS Oracle.

- ***Redo Log Buffers:***

Es un espacio reservado en memoria que contiene los cambios realizados a los bloques de datos, permitiendo la reconstrucción de la base de datos en caso de ocurrir un accidente.

- ***Database Buffers:***

Es un espacio reservado en memoria para las operaciones normales de la base de datos, el cual depende del tamaño especificado en el archivo de inicialización (INIT.ORA).

En la siguiente tabla se muestran algunos de los Sistemas de Gestión de Base de Datos, en la cual se muestran sus diferentes características, plataformas y ventajas.

**Tabla 9.** Sistema de Gestión de Base de Datos

<b>SGBD</b>	<b>Características</b>	<b>Plataformas</b>	<b>Ventajas</b>
<b>MYSQL</b>	Sistema de administración de BD Sistema de administración relacional de BD Software de fuente abierta	Windows 9x/NT 4.0/2000	Rápido, confiable y fácil de usar Practico Set de Características Manejo de grandes BD Conectividad, velocidad y seguridad
<b>SQL SERVER 7.0</b>	Escalabilidad Facilidad de uso Internet, Intranet y Comercio Sistemas fijos, móviles y distribuidos	Windows 9x/NT 4.0/2000	Rendimiento Escalabilidad Fiabilidad Soluciones inteligentes y coherentes con el mundo real
<b>ORACLE</b>	Manejador de BD Relacional Estructura física y lógica Uso de Memoria El kernel de Oracle Programa de Área Global Sistema de Área Global	Windows 9x/NT 4.0/2000 UNIX	Corre en PC, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo 17 idiomas  Corre automáticamente en mas de 80 arquitecturas de Hardware y Software Los códigos internos son iguales en todas las plataformas de sistemas operativos

# Capítulo IV



## *Aplicación PIENSSA*

Este capítulo trata del motivo principal de este trabajo de tesis: "La aplicación Web PIENSSA, una herramienta para generar y documentar un Portal Web Corporativo". En el se explica a detalle qué es, cómo fue creada, cual es su funcionamiento y qué obtenemos como producto final al hacer uso de ella.

#### **4.1. Descripción General**

PIENSSA es una aplicación Web que permite crear un "Portal Web Corporativo" con calidad, utilizando una metodología adecuada y generando su documentación. PIENSSA es un software que permite optimizar y agilizar los procesos de diseño, desarrollo y mantenimiento del "Portal Web Corporativo" generado a partir de él, aprovechando las características que el lenguaje de marcación de hipertextos ofrece.

PIENSSA es una Herramienta Metodológica para Análisis y Diseño de Sistemas Web, basada en la metodología RUP (Proceso Unificado del Racional) y apoyada de la notación UML (Lenguaje Unificado de Modelado). La tendencia de esta aplicación es poder hacer uso de recursos de hipermedia en versiones posteriores.

PIENSSA permite generar más de un "Portal Web Corporativo" por usuario y entrega como producto final el código en HTML del portal que se ha generado, la documentación técnica y plantillas RUP generadas dinámicamente cada vez que el "Portal Web Corporativo" es creado.

El software y la documentación técnica arriba mencionados son entregados en formato ASCII para su total portabilidad y el tamaño de los archivos que genera lo permiten por lo pequeño que estos resultan.

PIENSSA no requiere del uso de recursos tecnológicos adicionales para el usuario ya que solo es necesario contar con una conexión a Internet para poder hacer uso de ella y el servidor de Internet donde se encuentre instalado se hará cargo del resto.

PIENSSA sigue un estándar de diseño en sus pantallas y en el registro de su información, buscando agilizar y familiarizar al usuario con su procedimiento de trabajo. PIENSSA basa la creación del "Portal Web Corporativo" en principios básicos de calidad como son:

- Los elementos tienen una distribución estándar de acuerdo a las características de Portales Corporativos visitados y analizados.
- No contiene gif's animados ni letreros tipo marquesinas.
- No coloca páginas con letreros "En construcción".
- Llega a cualquier tema seleccionado en no más de 3 click's del Mouse.

- El software fuente generado contiene una instrucción por cada línea.
- El software fuente y la documentación técnica generados son totalmente portables al estar en formato ASCII.
- Los botones de la Barra de Herramientas tienen el tamaño estándar de 32 por 32 píxeles.
- Por defecto no contiene marcos o se sugieren de grosor del número 1 en caso de ser necesarios.
- Entre otros...

## **4.2. Acerca de la Aplicación**

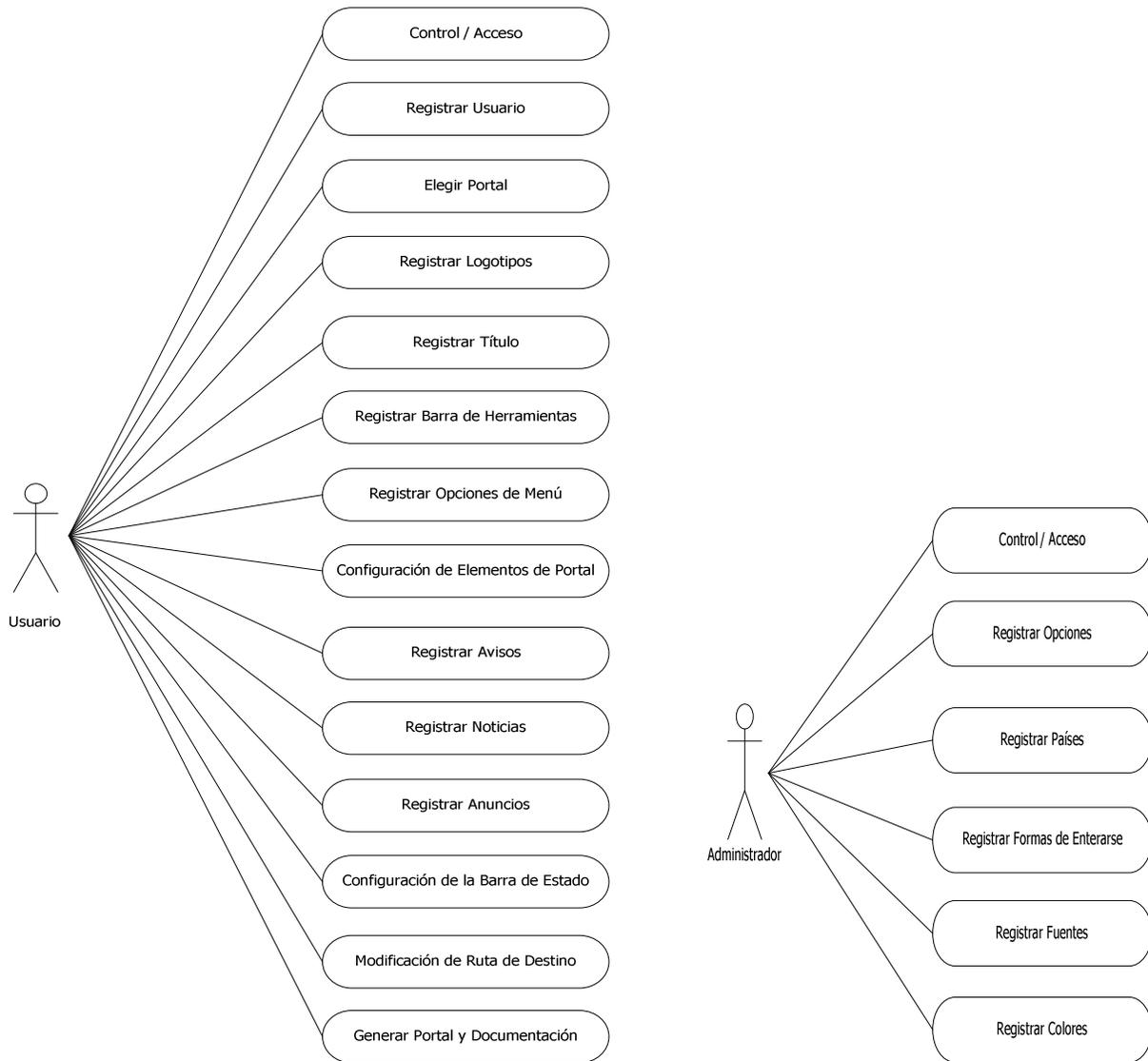
PIENSSA ha sido desarrollada utilizando las siguientes herramientas tecnológicas:

- **Software**
  - Sistema operativo, Microsoft Windows, basado en tecnología NT.
  - Servidor de Internet, Microsoft Internet Information Server (IIS 5.0).
  - Herramienta Case de Diseño de Base de Datos, Power Designer DataArchitec 6.0.
  - Editor de páginas y Sitios Web, Macromedia DreamWeaver MX 6.0.
  - Sistema Manejador de Bases de Datos, Microsoft SQL Server 7.0.
  - Lenguajes de Programación, Microsoft ASP, JAVA Script y HTML.
- **Metodología**
  - Proceso Racional Unificado (RUP).
- **Estándares de Calidad:**
  - ISO 9126.
  - ISO 14598.
  - IEEE 1061.

### 4.2.1. Diseño de PIENSSA

#### 4.2.1.1. Diagrama de Casos de Uso

La Fig. 9 muestra los Diagramas de Casos de Uso, estos diagramas representan las actividades y las pantallas a las cuales pueden tener acceso el Usuario y el Administrador.



**Fig. 9:** Diagramas de Casos de Uso

### 4.2.1.2. Descripción de los Casos de Uso

La Tabla 10 y la Tabla 11 representan la Descripción de dos de los Casos de Uso, el del Registro de Usuario y el de la Generación del Portal y su Documentación, cada tabla tiene el nombre del Caso de Uso, los Actores que intervienen, el Propósito del caso de Uso, el Resumen de las actividades que se realizan, el Tipo y las Referencias Cruzadas.

**Tabla 10.** Registrar Usuario

Caso de Uso:	Registrar Usuario.
Actores:	Usuario (Iniciador).
Propósito:	Dar de alta un Usuario para su posterior ingreso al sistema
Resumen:	<ul style="list-style-type: none"> <li>• El Usuario ingresa su información.</li> <li>• El sistema agrega al Usuario a la Base de Datos.</li> <li>• El sistema agrega al Cuenta y la Contraseña del Usuario a la Base de Datos.</li> <li>• El sistema envía los datos de la Cuenta y de la Contraseña del Usuario al Correo Electrónico especificado por el Usuario.</li> </ul>
Tipo:	
Referencias Cruzadas:	Control / Acceso

**Tabla 11.** Generar Portal y su Documentación

Caso de Uso:	Generar Portal y su Documentación.
Actores:	Usuario (Iniciador).
Propósito:	Generar el Portal Web Corporativo, las páginas que lo complementan y la documentación técnica del mismo.
Resumen:	<ul style="list-style-type: none"> <li>• El sistema genera el archivo PortalWeb.htm.</li> <li>• El sistema genera el archivo de Avisos: Avisos.htm.</li> <li>• El sistema genera el archivo de Noticias: Noticias.htm.</li> <li>• El sistema genera el archivo de Anuncios: Anuncios.htm.</li> <li>• El sistema genera la documentación técnica del Portal Web Corporativo en Documentación.txt</li> <li>• El sistema genera las Plantillas RUP en los archivos: ReglasdeNegocio.txt y VisiondelNegocio.txt</li> </ul>
Tipo:	
Referencias Cruzadas:	Control / Acceso

### **4.2.1.3. Diagrama de Clases**

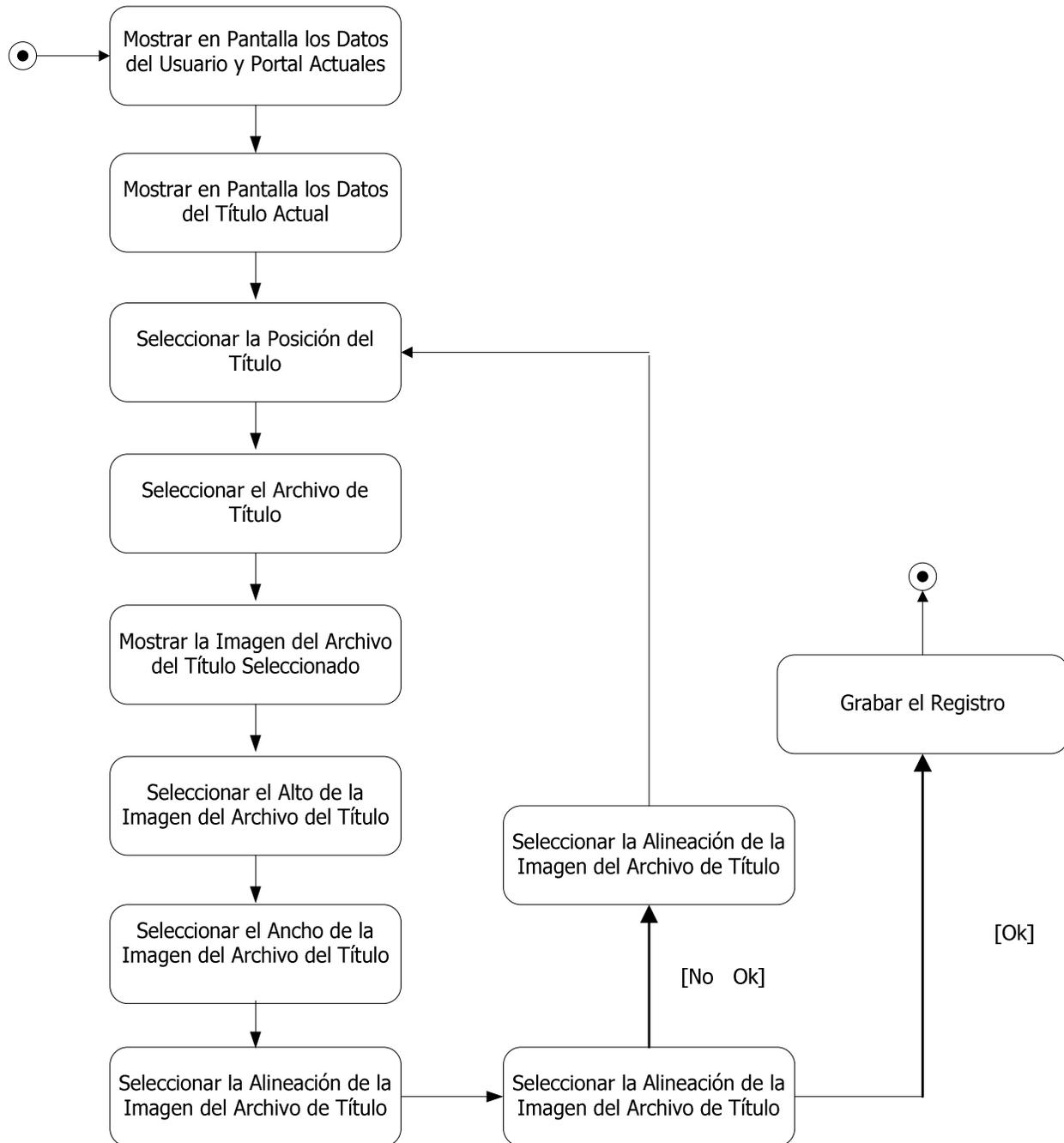
La Fig. 10 muestra el Diagrama de Clases generado con la ayuda de la Metodología RUP, en el cual se pueden apreciar las Clases con sus atributos y métodos que componen a PIENSSA.

**Fig. 10:** Diagrama de Clases

### 4.2.1.4. Diagramas de Actividades

#### 4.2.1.4.1. Registro del Título

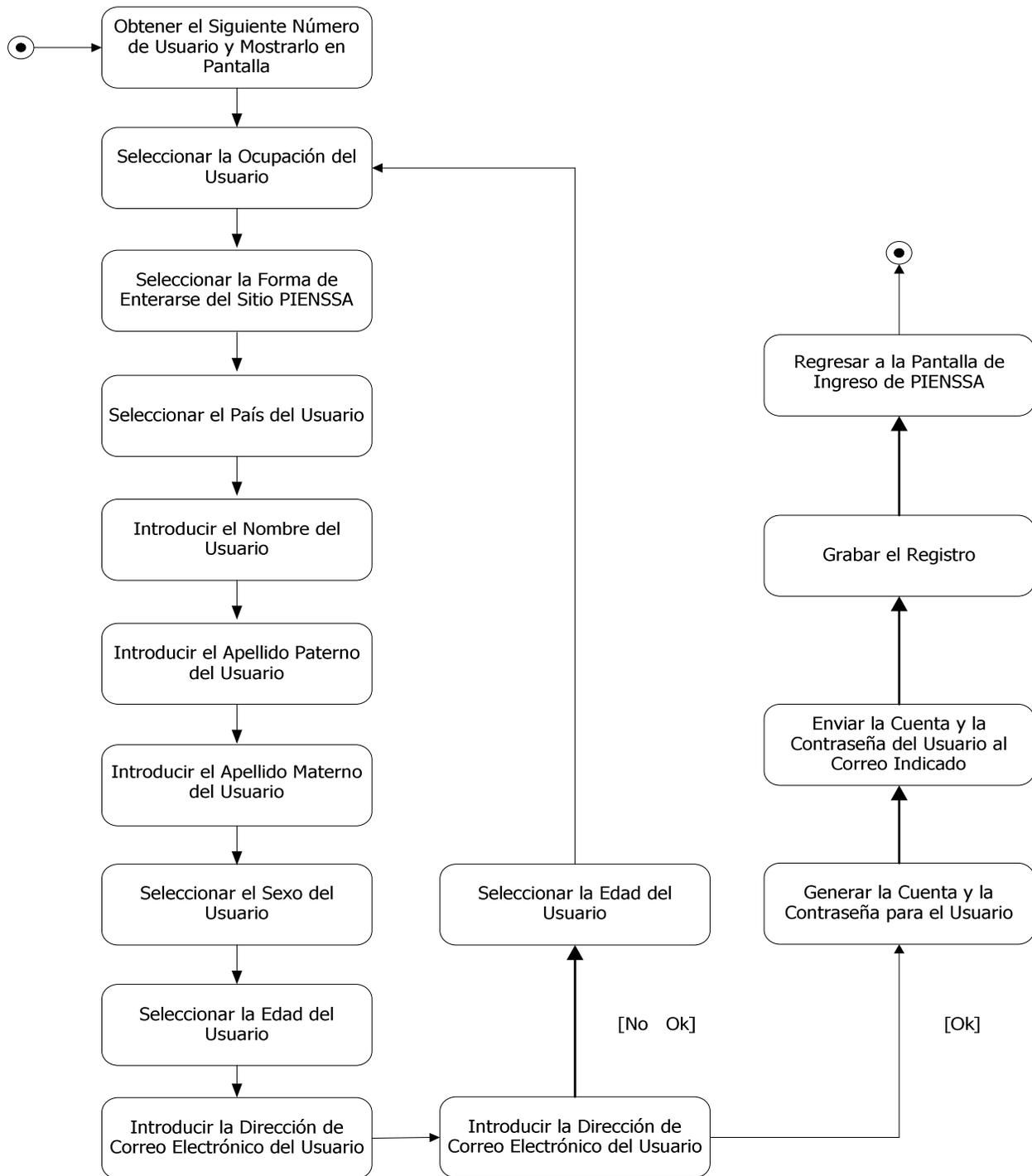
La Fig. 11 muestra el Diagrama de Actividades correspondiente al Registro del Título, en él se puede apreciar el flujo que se tiene en todo el proceso de las actividades apoyadas en la Herramienta PIENSSA.



**Fig. 11:** Diagrama de Actividades del Registro de Título

**4.2.1.4.2. Registro de Usuarios**

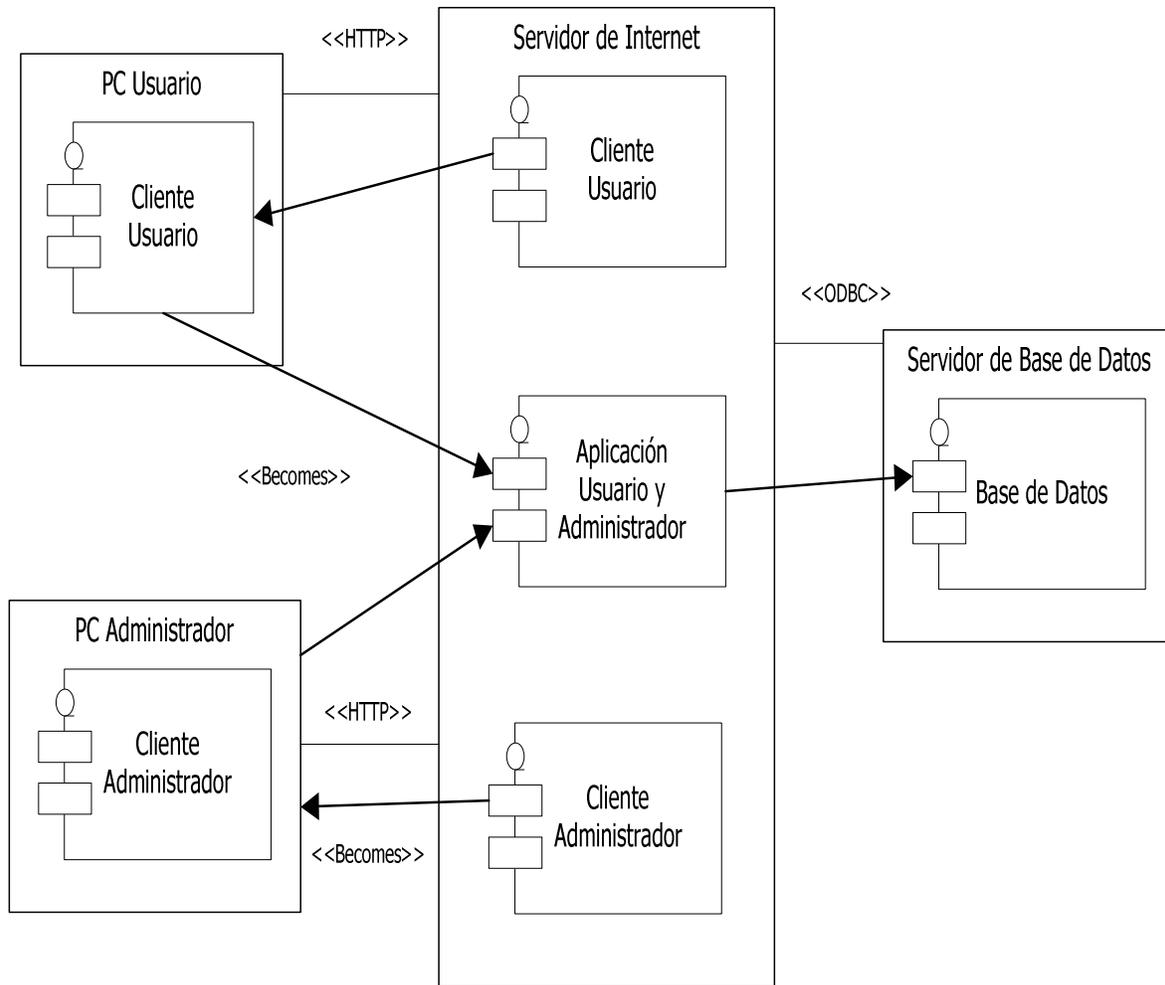
La Fig. 12 muestra el Diagrama de Actividades correspondiente al Registro de Usuarios, en él se puede apreciar el flujo que se tiene en todo el proceso de las actividades con la ayuda de la Herramienta PIENSSA.



**Fig. 12:** Diagrama de Actividades del Registro de Usuarios

### 4.2.1.5. Diagrama de Implementación

La Fig. 13 representa el Diagrama de Implementación de la Herramienta PIENSSA.



**Fig. 13:** Diagrama de Implementación

#### **4.2.1.6. Diseño de la Base de Datos**

El diseño de la Base de Datos se realizó en dos etapas:

##### **4.2.1.6.1. *Modelo Conceptual***

Creado desde Power Designer DataArchitec 6.0, basándose en las clases obtenidas en los casos de uso de la notación UML de la metodología RUP y factible para cualquier Sistema Manejador de Bases de Datos.

La Fig. 14 muestra el Modelo Conceptual de la herramienta PIENSSA

De este modelo se genera, de manera automática, el Modelo Físico, orientado específicamente al SMBD Microsoft SQLServer 7.0.

**Fig. 14:** Modelo Conceptual de PIENSSA

#### **4.2.1.6.2.        *Modelo Físico***

Este modelo cuenta ya con características únicas y propias del SMD Microsoft SQLServer 7.0, con el cual trabaja PIENSSA.

**Fig. 15:** Modelo Físico de PIENSSA

## **4.2.2. Desarrollo de PIENSSA**

### **4.2.2.1. Creación de la Base de Datos**

El archivo de tipo Script que se genera automáticamente desde Power Designer DataArchitec 6.0 cuenta con las instrucciones necesarias para generar de manera automática la Base de Datos en Microsoft SQLServer 7.0.

En la Fig. 16 se muestra el Script generado.

**Fig. 16:** Archivo Script generado en DataArchitec 6.0

Para lograr la creación de la base de datos es necesario ingresar al Administrador Corporativo de Microsoft SQL Server 7.0 y desde el Analizador de Consultas, ejecutar el archivo Script antes mencionado. Esto permitirá la creación automática de la Base de Datos en este SMBD, como se muestra en la Fig. 17.

**Fig. 17:** Creación de la Base de Datos desde Microsoft SQL Server 7.0

### **4.2.2.2. Programación de la Aplicación PIENSSA**

Para programar la aplicación PIENSSA y configurar el servidor de Internet se utilizó el Editor de Páginas y Sitios Web, Macromedia DreamWeaver MX 6.0.

Con este editor se programaron los módulos de PIENSSA en lenguaje HTML, ASP y Java Script y se conjuntaron con los complementos como las imágenes y el Manual de Ayuda en línea en formato Mht. Ver Fig. 18.

**Fig. 18:** Desarrollo de PIENSSA en Macromedia DreamWeaver MX 6.0

### **4.3. Funcionamiento de PIENSSA**

PIENSSA es una aplicación Web instalada en un Servidor de Internet y los usuarios se conectan a la página del sitio para poder generar su "Portal Web Corporativo". Todo el procedimiento que se sigue para la creación de un portal con PIENSSA se puede apreciar en la Fig. 19.

PIENSSA almacena en el servidor de Internet los datos de cada usuario que ingresa al sitio además de las características de cada uno de los "Portales Web Corporativos" que generen.

El "Portal Web Corporativo" como tal, la documentación técnica y las plantillas RUP creadas dinámicamente son almacenados en la computadora del Usuario en formato ASCII.

**Fig. 19:** Procedimiento para la generación de un portal a través de PIENSSA

Cuando un usuario desea modificar un "Portal Web Corporativo" generado con anterioridad, es necesario indicarle a PIENSSA cual de los portales disponibles para este usuario desea cargar para su edición.

#### **4.3.1. Mapa del Sitio**

La Fig. 20 muestra el Mapa del Sitio PIENSSA, el cual ilustra los módulos y la relación entre ellos.

**Fig. 20:** Mapa del Sitio

### 4.3.2. Acceso al sitio PIENSSA

Para utilizar la aplicación **PIENSSA** se debe ingresar al Sitio PIENSSA del Servidor de Internet donde este se encuentre instalado, ya sea de forma local o de manera remota. Si la aplicación se encuentra instalada como un sitio local, se debe escribir en la barra de navegación la siguiente URL:

http://localhost/PIENSSA

Con lo cual se ingresa al sitio PIENSSA y en donde se puede hacer uso de esta herramienta para crear un "Portal Web Corporativo" con calidad.

### 4.3.3. Ingreso a PIENSSA

Una vez en la página principal de **PIENSSA**, se introduce la Cuenta y la Contraseña de usuario que previamente ha sido asignada de manera automática por PIENSSA.

En la Fig. 21 se muestra la pantalla que permite ingresar a PIENSSA

En el espacio de Cuenta, se introduce el nombre de Cuenta o Usuario que ha sido asignado y enviado a la cuenta de correo electrónico al momento de registrarse como usuario de PIENSSA.

En el espacio de Contraseña se escribe la clave de acceso que le fue enviada al correo electrónico al momento del registro.

#### **Fig. 21:** Ingreso a PIENSSA

El botón **Entrar** permite ingresar al **Menú** de la aplicación y comenzar a generar el "Portal Web Corporativo".

Si aún no se cuenta con estos datos, se da click en "*regístrese aquí*" para ingresar los datos personales y así poder obtener la Cuenta y Contraseña de Ingreso a PIENSSA, los cuales serán enviados a la cuenta de correo electrónico que se proporcione.

#### 4.3.4. Registro de Usuarios

En esta pantalla los usuarios de PIENSSA deben registrar sus datos personales para poder obtener su Cuenta de Usuario y Contraseña de Ingreso. Estas son generadas y enviadas de manera automática a la dirección de correo electrónico especificada por el usuario en esta pantalla.

Esta pantalla se muestra en la Fig. 22 en la cual:

- **Usuario** indica el número de Usuario actual de PIENSSA. Este dato no es modificable.
- **Ocupación, ¿Cómo se entero de este sitio?, País de origen, Sexo y Edad** para especificar estos campos se da click en la flecha del combo y se selecciona de la lista el valor deseado para este campo.

**Fig. 22:** Registro de Usuarios

- **Nombre**, los **Apellidos Paterno y Materno** y la dirección de **Correo Electrónico** (este último dato es muy importante ya que la Cuenta y la Contraseña de Usuario serán enviadas a la dirección que aquí se indique) son obligatorios.
- **Enviar** este botón se presiona al terminar de introducir la información.
- **Restablecer** en caso de que se desee comenzar la introducción de información desde el principio se utiliza este botón.

#### 4.3.5. Menú de Trabajo

PIENSSA ofrece 2 alternativas para trabajar con un "Portal Web Corporativo":

##### **4.3.5.1. Generar un Portal Web Corporativo**

Al elegir esta alternativa PIENSSA permite crear un "Portal Web Corporativo", donde el Usuario puede especificar todas las características del mismo, comenzando desde cero.

#### **4.3.5.2. Ver un Portal Web Corporativo creado con anterioridad**

Todas las características de cada "Portal Web Corporativo" generado por PIENSSA pueden ser almacenadas, esta opción del menú permite recobrar la configuración de algún "Portal Web Corporativo" creado y almacenado con anterioridad para su posible edición. Al hacer click en esta opción lleva a la pantalla de "Carga de Portales".

La pantalla que muestra las dos alternativas de Menú de Trabajo son apreciadas en la Fig. 23.

En cualquiera de los casos anteriores, las características del Portal pueden ser modificadas y almacenadas para su posterior utilización y/o edición.

**Fig. 23:** Menú de Trabajo de PIENSSA

##### **4.3.5.2.1. Carga de Portales**

Desde esta pantalla se puede elegir cualquiera de los portales generados con anterioridad desde **PIENSSA** por el usuario actual.

Para lograr esto basta con hacer click en el número del "Portal Web Corporativo" que se desee recuperar para su edición y presionar el botón **Enviar**.

El botón **Restablecer** elimina la selección, permitiendo repetir esta acción.

**Fig. 24:** Carga de Portales

### 4.3.6. Diseño de la Estructura del Portal

La Fig. 25 muestra la pantalla donde se puede elegir las características a configurar para generar el "Portal Web Corporativo" y su documentación.

En la parte inferior izquierda de la pantalla se muestra en todo momento el nombre del Usuario actual, el número de Portal que se está generando o editando y la ruta donde se almacenara el resultado de generar el "Portal Web Corporativo".

Los elementos que conforman la estructura del "Portal Web Corporativo" son:

1. Logotipo Izquierdo.
2. Título
3. Logotipo Derecho.
4. Barra de Herramientas.
5. Menú.
6. Área de Despliegue.
7. Avisos.
8. Noticias.
9. Anuncios.
10. Barra de Estado.
11. Cabecera.
12. Cuerpo.
13. Lateral.
14. Pantalla Completa.

**Fig. 25:** Estructura del Portal Web Corporativo

Para configurar cualquiera de estos elementos, basta con hacer click con el Mouse en cualquiera de los vínculos indicados.

Desde esta pantalla es posible enviar comentarios y/o sugerencias al Administrador de PIENSSA acerca del funcionamiento del mismo.

- **Modificar Ruta** permite modificar la ruta actual donde se almacenarán tanto el "Portal Web Corporativo" como la documentación generada.

- **Nuevo Portal si** se desea cambiar de Portal, ya sea para iniciar uno nuevo o para cargar alguno creado con anterioridad, se presiona este botón.

Para obtener ayuda acerca del funcionamiento de PIENSSA, es necesario hacer click en el Enlace de **Ayuda** ubicado en la esquina inferior derecha de esta pantalla.

#### **4.3.6.1. Registro de Logotipos**

Cada "Portal Web Corporativo" puede tener 2 logotipos los cuales pueden ubicarse a la izquierda, al centro o a la derecha del área de Cabecera. Desde esta pantalla se registran tanto el logotipo izquierdo como el logotipo derecho del Portal.

La parte superior de esta pantalla, Fig. 26, muestra los datos del Usuario y del Portal actuales, además de la información que de los logotipos ya pudiese estar configurada.

- **Logotipo** indica el logotipo que se está configurando. Logotipo 1 por defecto se considera el de la izquierda y logotipo 2 es considerado por defecto el de la derecha.

**Fig. 26:** Registro de Logotipos

- **Posición** permite ubicar el logotipo en cuestión, teniendo en cuenta que dependiendo de la ubicación que se de al Logotipo actual, el Título y el Logotipo restante se reubicarán de manera automática, siempre dentro del área de Cabecera.

- **Archivo de Logotipo** hace referencia al archivo de imagen que se mostrará como el Logotipo actual. Se puede especificar directamente o se puede utilizar el botón Examinar para seleccionar dicho archivo de imagen. Una vez seleccionado el archivo, la imagen de muestra se mostrará en pantalla.

- **Alto y Ancho de la Imagen** determinan el tamaño que tendrá la imagen del Logotipo en el espacio que dentro de la Cabecera tiene reservado.

- **Alineación de la Imagen** permite elegir la posición de la imagen con respecto al espacio reservado para esta dentro de la cabecera. (Línea de base, Superior, Medio Absoluta, etc.).

Para almacenar las características del Logotipo actual es necesario hacer click en el botón **Enviar**.

El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, haga click en el botón **Estructura del Portal**.

#### **4.3.6.2. Registro de Titulo**

El Título es la imagen principal dentro del área de encabezados del Portal y se da de alta siguiendo las mismas consideraciones que se siguen al registrar los logotipos.

Cada "Portal Web Corporativo" tendrá un Título que podrá ubicarse a la izquierda, al centro o a la derecha del área de Cabecera.

La Fig. 27 muestra la pantalla que permite configurar el Título de Portal Web Corporativo, en la cual:

- La parte superior de esta pantalla muestra los datos del Usuario y del Portal actuales, además de la información que del Título ya pudiese estar configurada.
- **Posición** permite ubicar el Título, teniendo en cuenta que dependiendo de la ubicación que se dé a este, los Logotipos se reubicarán de manera automática, siempre dentro del área de Cabecera.

**Fig. 27:** Registro de Título

- **Archivo de Título** hace referencia al archivo de imagen que se mostrará para el Título. Puede especificarse directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo de imagen. Una vez seleccionado el archivo, la imagen de muestra se mostrará en pantalla.
- **Alto y Ancho de la Imagen** determinan el tamaño que tendrá la imagen del Título en el espacio que dentro de la Cabecera tiene reservado.
- **Alineación de la Imagen** permite elegir la posición de la imagen con respecto al espacio reservado para esta dentro de la cabecera. (Línea de base, Superior, Medio Absoluta, etc.).

Para almacenar las características del Título es necesario hacer click en el botón **Enviar**. Con el botón **Restablecer** se puede borrar la captura actual y si se desea regresar a la pantalla de la estructura del Portal, hay que presionar el botón **Estructura del Portal**.

#### **4.3.6.3. Registro de la Barra de Herramientas**

La Barra de Herramientas estará compuesta por Botones que colocados de izquierda a derecha ofrecerán funcionalidad específica al "Portal Web Corporativo" generado por **PIENSSA**. Cada Botón tendrá una imagen relacionada y enlazará el archivo o URL que se indique para que cumpla con la función que le sea asignada. Es posible colocar hasta 20 Botones en la Barra de Herramientas, aunque por cuestiones técnicas de calidad no se recomiendan más de 10.

La parte superior de esta pantalla, Fig. 28, muestra los datos del Usuario y del Portal actuales.

- **Herramienta** hace referencia al Botón que actualmente se está configurando. Se dan de alta en orden ascendente.
- **Etiqueta** indica el texto de título para cada Botón de la Barra.
- **Color de la Fuente** permite seleccionar del catálogo de colores el que se desee asignar a la Etiqueta del Botón de la Barra.

**Fig. 28:** Registro de Barra de Herramientas

- **Posición** especifica la Posición que de izquierda a derecha tiene el Botón dentro de la Barra. Por defecto, la posición de cada Botón coincide con el orden en que se da de alta.
- **Archivo de Imagen** hace referencia al archivo de imagen que se mostrará para el Botón. Se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo de imagen.
- **Enlace** especifica el Archivo o URL al cual se enlazará el control cuando este Botón de la Barra de Herramientas sea seleccionado. Se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo o URL.

Para almacenar las características de cada Botón es necesario hacer click en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, es necesario hacer click en el botón **Estructura del Portal**.

Para buscar una Herramienta previamente guardada se tiene que escribir el **Id** de la misma en el cuadro **Herramienta** y presionar el botón **Buscar**, el cual mostrará la información del registro para su consulta y/o posible modificación. Si el registro solicitado no existe, PIENSSA mostrará un mensaje de "REGISTRO NO EXISTE". Para eliminar el registro solicitado hay que presionar el botón **Eliminar** y este se borrará de la Base de datos.

#### **4.3.6.4. Registro de las Opciones de Menú**

Las opciones que conformarán el Menú Principal del "Portal Web Corporativo" se registran en esta pantalla, la cual muestra en la parte superior, información acerca del Usuario y del Portal actuales. **Menú** hace referencia a la Opción del Menú que actualmente se está configurando. Se dan de alta en orden ascendente. Esta pantalla se puede apreciar en la Fig. 29, en la que:

- **Etiqueta** indica el texto que se mostrará en pantalla por cada opción de Menú.
- **Formato del Párrafo** da forma al mismo, permitiendo seleccionar Encabezado 1, Encabezado 2, Preformateado, etc.).
- **Fuente** permite seleccionar del catálogo la fuente disponible para cada Opción del Menú.

- **Tamaño de la Fuente** permite seleccionar el tamaño de la misma, desde 1 hasta -7.
- **Color de la Fuente** permite seleccionar del catálogo de colores el que se desee asignar a la Etiqueta de la Opción de Menú.
- **Estilo de la Fuente** ofrece la alternativa de elegir entre normal, negrita, cursiva o combinación de estilos.

**Fig. 29:** Registro de Opciones de Menú

- **Alineación de la Etiqueta** como su nombre lo dice alinea la etiqueta como se le indique (izquierda, centro, derecha o justificada).
- **Posición Vertical de la Etiqueta** especifica la Posición que de arriba hacia abajo tiene la Opción de Menú dentro de este. Por defecto, la posición de cada Opción coincide con el orden en que se da de alta.
- **Enlace** especifica el Archivo o URL al cual se enlazará el control cuando esta Opción de Menú sea seleccionada. Se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo o URL.

Para almacenar las características de cada Opción de Menú se tiene que hacer click en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, hay que presionar el botón **Estructura del Portal**.

Para buscar una Opción del Menú previamente guardada, hay que escribir el **Id** de la misma en el cuadro **Menú** y presionar el botón **Buscar**, el cual mostrará la información del registro para su consulta y/o posible modificación.

Si el registro solicitado no existe, PIENSSA mostrará un mensaje de "REGISTRO NO EXISTE". Para eliminar el registro solicitado se tiene que presionar el botón **Eliminar** y este se borrará de la Base de datos.

#### **4.3.6.5. Configuración del Área de Despliegue**

En esta pantalla, Fig. 30, se puede configurar el fondo y los bordes de cada uno de los Elementos que conforman el "Portal Web Corporativo". La parte superior de la pantalla muestra información referente al usuario y al portal en cuestión.

- **Elemento** permite seleccionar el Elemento del Portal que se desea configurar.
- **Archivo de Imagen de Fondo** hace referencia al archivo de imagen que se mostrará para el fondo de este Elemento. Se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo de imagen.
- **Alineación de la Imagen de Fondo** como su nombre lo dice alinea la Imagen previamente especificada (izquierda, centro, derecha).

**Fig. 30:** Configuración de Área de Despliegue

- **Color de Fondo** permite seleccionar del catálogo de colores el que se desee asignar al fondo del Elemento, el cual será visible siempre y cuando no se especifique una Imagen para el mismo.

- **Grosor del Borde** determina el grosor para la línea de borde del Elemento. Los valores permitidos van de 0 a 10, aunque por cuestiones de calidad se sugiere no colocar borde o si es necesario no debe ser mayor de 1.

- **Color del Borde** permite seleccionar del catálogo de colores el que se desee asignar al Borde del Elemento, el cual será visible siempre y cuando el valor para el Borde sea igual o mayor que 1.

Para almacenar las características de configuración del Fondo de cada Elemento, hay que hacer click en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, se tiene que hacer click en el botón Estructura del Portal.

#### **4.3.6.6. Registro de Avisos**

PIENSSA ofrece la capacidad de incluir en el "Portal Web Corporativo" que genera, un apartado de Avisos, donde se debe especificar la etiqueta del Aviso y el archivo o URL que contiene el cuerpo o contenido del mismo. La pantalla, Fig. 31, muestra en la parte superior, información referente al Usuario y Portal actuales.

- **Aviso** hace referencia al Aviso que actualmente se está configurando. Se dan de alta en orden ascendente y de forma automática.
- **Descripción** permite escribir el resumen del Aviso que se está dando de alta. Esta caja de texto es deslizable. Es decir, se puede escribir una Descripción de hasta 250 caracteres, aunque por cuestiones de calidad se sugiere que esta no sea mayor a 50 caracteres.

**Fig. 31:** Registro de Avisos

- **Enlace** especifica el Archivo o URL al cual se enlazará el control cuando este Aviso sea seleccionado. Este se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo o URL.

Para almacenar las características de cada Aviso es necesario hacer click en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, hay que presionar el botón **Estructura del Portal**.

Para buscar un Aviso previamente guardado, hay que escribir **Id** del mismo en el cuadro **Aviso** y presionar el botón **Buscar**, el cual le mostrará la información del registro para su consulta y/o posible modificación.

Si el registro solicitado no existe, PIENSSA mostrará un mensaje de "REGISTRO NO EXISTE". Para eliminar el registro solicitado hay que presionar el botón **Eliminar** y este se borrará de la Base de datos.

#### **4.3.6.7. Registro de Noticias**

PIENSSA ofrece la capacidad de incluir en el "Portal Web Corporativo" que genera, un apartado de Noticias, donde se debe especificar la etiqueta de la Noticia y el archivo o URL que contiene el cuerpo o contenido de la misma.

La pantalla, ver Fig. 32, muestra en la parte superior, información referente al Usuario y Portal actuales.

- **Noticia** hace referencia a la Noticia que actualmente se está configurando. Se dan de alta en orden ascendente.

**Fig. 32:** Registro de Noticias

- **Descripción** permite escribir el resumen de la Noticia que se está dando de alta. Esta caja de texto es deslizable. Es decir, se puede escribir una Descripción de hasta 250 caracteres, aunque por cuestiones de calidad se sugiere que esta no sea mayor a 50 caracteres.

- **Enlace** especifica el Archivo o URL al cual se enlazará el control cuando esta Noticia sea seleccionada. Se puede especificar directamente o se puede utilizar el botón **Examinar** para seleccionar dicho archivo o URL.

Para almacenar las características de cada Noticia se tiene que hacer en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, hay que presionar el botón **Estructura del Portal**.

Para buscar una Noticia previamente guardada, hay que escribir el **Id** de la misma en el cuadro **Noticia** y presionar el botón **Buscar**, el cual mostrará la información del registro para su consulta y/o posible modificación.

Si el registro solicitado no existe, PIENSSA mostrará un mensaje de "REGISTRO NO EXISTE". Para eliminar el registro solicitado hay que presionar el botón **Eliminar** y este se borrará de la Base de datos.

#### **4.3.6.8. Registro de Anuncios**

PIENSSA ofrece la capacidad de incluir en el "Portal Web Corporativo" que genera, un apartado de Anuncios, donde se debe especificar la etiqueta del Anuncio y el archivo o URL que contiene el cuerpo o contenido del mismo. En la Fig. 33 se muestra la pantalla en la cual, en la parte superior, muestra información referente al Usuario y Portal actual.

- **Anuncio** hace referencia al Anuncio que actualmente se está configurando. Se dan de alta en orden ascendente.

- **Descripción** permite escribir el resumen del Anuncio que se está dando de alta. Esta caja de texto es deslizable. Es decir, se puede escribir una Descripción de hasta 250 caracteres, aunque por cuestiones de calidad se sugiere que esta no sea mayor a 50 caracteres.

- **Enlace** especifica el Archivo o URL al cual se enlazará el control cuando este Anuncio sea seleccionado. Puede especificarlo directamente o puede utilizar el botón **Examinar** para seleccionar dicho archivo o URL.

Para almacenar las características de cada Anuncio hay que hacer click en el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si desea regresar a la pantalla de la Estructura del Portal, se presiona el botón **Estructura del Portal**.

**Fig. 33:** Registro de Anuncios

Para buscar un Anuncio previamente guardado, hay que escribir el **Id** del mismo en el cuadro **Anuncio** y presionar el botón **Buscar**, el cual le mostrará la información del registro para su consulta y/o posible modificación.

Si el registro solicitado no existe, PIENSSA mostrará un mensaje de "REGISTRO NO EXISTE". Para eliminar el registro solicitado se presiona el botón **Eliminar** y este se borrará de la Base de datos.

### 4.3.7. Menú de Administración de PIENSSA

Las tareas administrativas se realizan ingresando a las pantallas permitidas para el administrador de **PIENSSA**. Para esto el administrador cuenta con una Cuenta y una Contraseña autorizadas para tal fin. Desde este menú se mantienen a punto los diferentes catálogos que permiten la óptima funcionalidad del sitio **PIENSSA**, como se muestra en la Fig. 34.

- **Botón de Ocupaciones** para ingresar al Catálogo de Ocupaciones.
- **Botón de Formas de Enterarse** para ingresar al Catálogo de Formas de Enterarse.
- **Botón de Países** para ingresar al Catálogo de Países.
- **Botón de Colores** para ingresar al Catálogo de Colores.

**Fig. 34:** Menú de Administración de PIENSSA

- **Botón de Fuentes** para ingresar al Catálogo de Fuentes.

Aunque se considera que algunos Catálogos son fijos se ofrece la posibilidad de realizar alguna modificación o de acrecentar alguno o todos ellos.

#### **4.3.7.1. Registro de Ocupaciones**

Aquí se registra el catálogo de las Ocupaciones, el cual es útil al momento de registrar los datos personales de los usuarios de **PIENSSA**, ver Fig. 35.

- **Ocupación** es el Identificador consecutivo del Catálogo de Ocupaciones.
- **Descripción** especifica la Ocupación que se está dando de alta.

Para almacenar los datos en este Catálogo, es necesario hacer click en el botón **Insertar registro**. El botón **Restablecer** permite borrar la captura actual.

**Fig. 35:** Registro de Ocupaciones

#### **4.3.7.2. Registro de Formas de Enterarse**

Aquí se registra el catálogo de las Formas de Enterarse, el cual es útil al momento de registrar los datos personales de los usuarios de **PIENSSA**. Su representación gráfica se muestra en la Fig. 36.

- **Forma de Enterarse** es el Identificador consecutivo del Catálogo de Formas de Enterarse.
- **Descripción** especifica la Forma de Enterarse que se está dando de alta. Para almacenar los datos en este Catálogo, se presiona el botón **Insertar registro**. El botón **Restablecer** permite borrar la captura actual.

**Fig. 36:** Registro de Formas de Enterarse

#### **4.3.7.3. Registro de Países**

Aquí se registra el catálogo de Países, el cual es útil al momento de registrar los datos personales de los usuarios de **PIENSSA**. Esta pantalla es representada en la Fig. 37.

- **País** es el Identificador consecutivo del Catálogo de Países.
- **Nombre** especifica el País que se está dando de alta.

Para almacenar los datos en este Catálogo, se presiona el botón **Insertar registro**. El botón **Restablecer** borrará la captura actual.

**Fig. 37:** Registro de Países

#### **4.3.7.4. Registro de Colores**

Aquí se registra el catálogo de Fuentes, el cual es útil al momento de configurar algunos elementos del "Portal Web Corporativo" creado por **PIENSSA**, ver Fig. 38.

- **Color** es el Identificador consecutivo del Catálogo de Colores.
- **Descripción** especifica la descripción del Color que se está dando de alta.
- **Código** especifica el valor hexadecimal del Color que se está dando de alta.

**Fig. 38:** Registro de Colores

Para almacenar los datos en este Catálogo, hay que hacer click en el botón **Insertar registro**. El botón **Restablecer** borrará la captura actual.

#### **4.3.7.5. Registro de Fuentes**

Aquí se registra el catálogo de Fuentes, el cual es útil al momento de configurar algunos elementos del "Portal Web Corporativo" creado por **PIENSSA**, ver Fig. 39.

- **Fuente** es el Identificador consecutivo del Catálogo de Fuentes.
- **Nombre** especifica el nombre de la Fuente que se está dando de alta.

Para almacenar los datos en este Catálogo, hay que presionar el botón **Insertar registro**. Utilice el botón **Restablecer** permite borrar la captura actual.

**Fig. 39:** Registro de Fuentes

#### **4.3.8. Contacte con Nosotros**

PIENSSA ofrece a sus usuarios la posibilidad de enviar correos electrónicos para comentar fallas y/o aciertos de la aplicación o para sugerir mejoras a la misma.

Esto se logra haciendo click en el enlace "Para cualquier comentario o sugerencia, **Contáctenos**" de la pantalla donde se muestra la **Estructura del Portal**, como se aprecia en la Fig. 40.

**Fig. 40:** Contacte con Nosotros

Lo anterior le lleva al Administrador de Correo Electrónico que tenga instalado en su computadora, donde solo tendrá que escribir el comentario y/o la sugerencia y enviarlo, ya que automáticamente aparece la cuenta de Correo Electrónico del Administrador de PIENSSA.

#### **4.3.9. Datos de Creación**

Los Datos de Creación se refieren a la fecha y la hora en que el "Portal Web Corporativo" es generado por PIENSSA. La representación gráfica de esta pantalla se puede apreciar en la Fig. 41.

Esta pantalla ofrece la posibilidad de observarlos y alinearlos dentro de la Barra de Estado del Portal generado.

La pantalla muestra en la parte superior, información referente al Usuario y Portal actuales.

- **Fecha** muestra la Fecha de creación del "Portal Web Corporativo". Por defecto toma la fecha del Sistema Operativo.
- **Hora** muestra la Hora de creación del "Portal Web Corporativo". Por defecto toma la hora del Sistema Operativo.
- **Alineación** permite indicar donde se mostrarán la Fecha y la Hora de Creación del "Portal Web Corporativo" dentro de la Barra de Estado del mismo (izquierda, centro ó derecha).

Para almacenar las características de Fecha y Hora, se puede hacer presionando el botón **Enviar**. El botón **Restablecer** permite borrar la captura actual y si se desea regresar a la pantalla de la Estructura del Portal, hay que presionar el botón Estructura del Portal.

**Fig. 41:** Datos de Creación

### 4.3.10. Modificación de la Ruta de Destino

Antes de generar el "Portal Web Corporativo" con PIENSSA, es conveniente verificar la Ruta de Destino de los archivos que se obtendrán. Esta se muestra en la parte inferior izquierda de la pantalla de la **Estructura del Portal**.

Si se desea modificar dicha ruta, desde la pantalla **Estructura del Portal** hay que presionar el botón **Modificar Ruta**, con lo cual se mostrará en la pantalla de la Fig. 42, lo siguiente:

Aquí se muestra la **Ruta Actual** de Destino para el "Portal Web Corporativo" y su Documentación. Para cambiar esta ruta hay que escribir o seleccionar desde el cuadro de **Nueva Ruta para almacenar el Portal**, y el cambio se verá reflejado en la pantalla de **Estructura del Portal**.

**Fig. 42:** Modificar Ruta

### 4.3.11. Generación del Portal Web Corporativo y su Documentación.

En todo momento **PIENSSA** permite generar El "Portal Web Corporativo" y su Documentación de manera dinámica. Es decir, con las características que en ese instante tiene el "Portal Web Corporativo" que se ha configurando.

Esto se logra haciendo click en el Botón **Generar Portal y Documentación** de la pantalla donde se muestra la **Estructura del Portal**, ver Fig. 43.

Lo que se muestra en pantalla es ya el "Portal Web Corporativo" generado por PIENSSA, de acuerdo a las especificaciones dadas por el usuario.

**Fig. 43:** Producto Final

Se obtienen como producto final los siguientes elementos generados de manera dinámica cada vez que se ejecuta este proceso, los cuales se incluyen en la sección de **Anexos**:

- El Código Fuente en HTML del "**Portal Web Corporativo**" y las páginas adicionales necesarias para su funcionamiento:
  - PortalWeb.Htm
  - DespliegueWeb.Htm
  - MenuWeb.Htm
  - AvisosWeb.Htm

- NoticiasWeb.Htm
- AnunciosWeb.Htm
  
- La Documentación Técnica que avala y explica los elementos que conforman el "Portal Web Corporativo":
  - Documenta.Txt
  
- Las plantillas RUP sugeridas y relacionadas con el "Portal Web Corporativo" recién generado:
  - VisiondelNegocio.Txt
  - ReglasdelNegocio.Txt

# PIENSSA



## *Conclusiones*

Después de elaborar el presente trabajo de tesis, podemos concluir que el desarrollar un trabajo con calidad no es tarea fácil ni de todos los días y mucho menos cuando su destino final es la Internet como medio de difusión masivo.

La sociedad de nuestros días emplea la información para reducir costos y producir bienes que consumimos y en general para mejorar nuestra calidad de vida. Gracias a los sistemas de comunicaciones y a las redes de computadoras hoy es posible el intercambio rápido de informaciones residenciales en millones de páginas Web e innumerables bases de datos accesibles a través del Web esparcidas por todo el mundo.

Dada la importancia que ha adquirido Internet en los últimos años y que en materia de comercialización se ha llegado a considerar como uno de los medios más importantes y vanguardistas, hemos comprobado que la necesidad de pequeñas empresas de entrar al Web es parte fundamental para su desarrollo.

En la actualidad, México es un país donde la cultura del uso de metodologías para desarrollar algún producto o proceso no está muy difundida. Las empresas de desarrollo de software comienzan a mirar hacia el lado de las metodologías y el uso de herramientas de software relacionadas con estas, porque están convencidas de que es la única manera de obtener y asegurar la calidad de sus productos.

Observamos que las metodologías analizadas coinciden en muchos aspectos y aunque manejan algunos conceptos diferentes, la intención final es la misma: Optimizar tiempo y esfuerzo para obtener un producto de calidad y con calidad.

Desarrollar un proyecto que implica la aplicación de una Metodología, normas de Calidad y experiencia técnica, requiere de un gran esfuerzo y de una adecuada planeación, además de mucha disciplina para cumplir el objetivo del mismo. El análisis y diseño son las etapas más importantes del proceso de desarrollo. Una buena elección de la metodología ayudará al acrecentamiento de los logros en estas etapas. Por ende, generar un Portal Web Corporativo con calidad requiere además mucha creatividad y amplios conocimientos de la empresa hacia la cual está orientada.

Los aspectos que conforman las Páginas Web son muchos y muy variados y van desde la apariencia hasta los detalles de la programación, pasando por el paradigma de su funcionamiento.

Revisando Portales dentro de la Internet hemos podido observar que por lo regular son las grandes empresas (nacionales y transnacionales), las que cuidan aspectos de calidad en sus portales y al

hablar de calidad nos referimos a aspectos que van desde el utilizar una distribución uniforme de sus elementos, hasta el considerar aspectos de velocidad, portabilidad y costo.

Actualmente contamos con alternativas de desarrollo para páginas de Internet y aplicaciones Web. Existen diversos sistemas operativos, lenguajes de programación y manejadores de bases de datos que nos permiten elegir el más adecuado de acuerdo a nuestras posibilidades económicas, tecnológicas y de recursos humanos.

La experiencia en el uso del lenguaje UML demuestra que usarlo como una herramienta de apoyo a lo largo de las diferentes etapas en el desarrollo de un proyecto de software, resulta benéfico en muchos y muy variados aspectos.

En la etapa de elaboración, mencionada como la segunda de las fases del desarrollo del RUP, el UML es muy útil al permitir generar diagramas de caso de uso que a su vez proporciona una idea clara de la dimensión del proyecto.

La mayoría de los programas que tienen este fin proporcionan las herramientas para desarrollar las Páginas Web, pero no limitan ni restringen los aspectos de calidad y mucho menos consideran el seguimiento de una metodología adecuada. World Wide Web nos permite utilizar una gran variedad de tecnologías para hacer uso de los servicios existentes.

Pero si decimos que crear dicho Portal no es tarea fácil, mucho menos fácil es crear una aplicación que nos permita hacer esto de manera automática y en tiempo real ya que cada día nos acercamos más a una era electrónica en la cual debemos poner mayor énfasis en la calidad de los servicios que se brinden.

Basándonos en la planeación, análisis y desarrollo; logramos el objetivo de crear un sitio dinámico y amigable que permite al usuario desde la comodidad de su oficina proveerse de los servicios que nuestra aplicación le ofrece, que en este caso es la realización de portales Web con una mejor estructura, mayor funcionalidad y calidad.

La aplicación PIENSSA, producto final del presente trabajo de tesis, es un primer esfuerzo por desarrollar una aplicación que permita generar Portales Web Corporativos con calidad, donde hemos ocupado la metodología RUP con las herramientas UML necesarias para su análisis y diseño, los lenguajes de programación ASP, JAVA Script y HTML para su desarrollo, Microsoft SQL Server como manejador de bases de datos, Microsoft Windows como Sistema Operativo de plataforma y Microsoft Internet

Information Server como Servidor de Internet. Además nos hemos basado en estándares de calidad internacionales para el desarrollo de este tipo de herramientas de software.

PIENSSA resultó ser una aplicación Web pequeña (menos de 1 Mb de código) y relativamente sencilla de programar. La complejidad de PIENSSA radica en la dificultad que implicó diseñar el procedimiento para obtener el Portal Web Corporativo y su documentación. Difícil también ha resultado, definir los parámetros de calidad sugeridos para mantener un estándar sin perder dicha calidad, permitiendo a la vez la creatividad de los usuarios.

Las posibilidades de crecimiento de PIENSSA son muchas y dependen del ingenio de nosotros, los autores, para lograr un mejor y más funcional diseño en las nuevas versiones de esta aplicación.

La perspectiva de una estructura inicial, donde cada uno de los elementos que conforman el Portal Web Corporativo ocupen una posición predeterminada, puede ser modificada para dar paso a una nueva versión donde los elementos puedan ser arrastrados y autoajustados a los espacios disponibles en pantalla.

Ofrecer como parte del producto final, la documentación técnica que avala el diseño y desarrollo del Portal Web Corporativo es uno de los objetivos de PIENSSA y en esta primera versión se logra mediante la inclusión de documento técnico que explica línea por línea el funcionamiento del Portal Web Corporativo generado automáticamente, además de las plantillas de Visión del Negocio y Reglas del Negocio, ambas del modelo del Proceso Unificado de Rational (RUP). Existen más plantillas de dicho modelo que pueden ser incluidas en PIENSSA, pero serán consideradas para versiones posteriores.

Los objetivos específicos de esta tesis se cumplieron y deja el tema abierto para abundar en él y enriquecer cada vez más el proceso de desarrollo de software con el estudio de diferentes metodologías y herramientas utilizadas en este, ya que Internet interactúa con millones de usuarios, es de vital importancia ofrecer la máxima calidad posible en los servicios que ahí se ofrecen.

# PIENSSA



## *Glosario*

**A**

**Adabas D:** Proporciona una base de datos profesional que implementa completamente el modelo relacional, incluyendo el soporte para dominios, claves primarias, vistas de uniones actualizables (updatable join views), integridad referencial, triggers y procedimientos de la base de datos.

**Applet:** Miniprograma en lenguaje de programación Java integrado en una página web

**ASCII:** American Standard Code of Information Interchange: Código normalizado estadounidense para el intercambio de la información. Código que permite definir caracteres (números, letras o símbolos en la computadora, conformado por un byte) alfanuméricos; se lo usa para lograr compatibilidad entre diversos procesadores (conjunto de circuitos lógicos que procesa las instrucciones básicas de una computadora) de texto. Se pronuncia "aski".

**B**

**Base de Datos:** Conjunto de datos que se encuentran almacenados bajo un mismo modelo lógico.

**B2B:** Business to Business. Negocio a Negocio. Es la relación comercial establecida, por medio de la red, entre dos empresas o compañías

**Brainstorming:** Tormenta de ideas. Es un método para generar ideas, consiste en una reunión de un grupo de personas en la que cada integrante expone soluciones sobre el tema de análisis, de forma espontánea y desinhibida, sin someterse a ninguna crítica por parte de los demás asistentes.

**Bytecodes:** Especie de pseudocódigo producido por la Máquina Virtual Java, tienen la particularidad de ser iguales para cualquier tipo de arquitectura de hardware.

**C**

**CASE:** Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Ordenador. Son instrumentos o sistemas automatizados que brindan soporte a las actividades de producción de software. Se pueden clasificar en: 1) *Upper CASE*: herramientas que soportan las actividades (de *alto nivel*) de análisis de requerimientos y diseño. Por ejemplo, las herramientas de modelado diagrama de entidad relación y diagramas UML. 2) *Lower CASE*: herramientas que soportan las actividades (de *bajo nivel*) de programación, la depuración de programas y las pruebas.

**Checklists:** Listas de comprobación.

**Clases:** Tipos de declaraciones abstractas. Especificaciones de datos definidos por el usuario.

**Cliente-Servidor:** Sistema de organización de interconexión de computadoras que sirve para el funcionamiento de Internet, así como otros tantos sistemas de redes. Se basa en la separación de las computadoras miembros en dos categorías: los que actúan como servidores (oferentes de información), y las que lo hacen como clientes (receptores de información).

**CM:** Abreviatura de Cable MODEM (dispositivo que permite el acceso de datos a alta velocidad utilizando la red de Televisión por Cable).

**CMM: Modelo de Capacidad y Madurez o CMM** (Capability Maturity Model), es un método de definir y gestionar los procesos a realizar por una organización. Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute). El SEI es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMM" es una marca registrada del SEI. Su homólogo en Europa es la Fundación ESI (European Software Institute), con sede en Bilbao (España).

**Cookies:** Son pequeños archivos con datos que algunos sitios Web depositan en forma automática en las computadoras de los visitantes. Lo hacen con el objetivo de almacenar allí información sobre las personas y sus preferencias. Por ejemplo, la primera vez que un navegante visita un site y completa un formulario con sus datos, el sistema le envía una cookie y le asigna una identificación. La siguiente vez que el mismo visitante retorne a ese sitio Web, se le pedirá automáticamente su cookie, y a través de ella se lo podrá reconocer.

**CORBA:** Common Object Request Broker Architecture (CORBA) es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos. CORBA fue definido y está controlado por el *Object Management Group* (OMG) que define las APIs, el protocolo de comunicaciones y los mecanismos necesarios para permitir la interoperabilidad entre diferentes aplicaciones escritas en diferentes lenguajes y ejecutadas en diferentes plataformas, lo que es fundamental en computación distribuida.

**Cron:** Es un administrador regular de procesos en segundo plano ("demonio") que ejecuta programas a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero *crontab*.

**CSS:** Hojas de Estilo

**D**

**DBMS:** Conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos.

**DDL:** Lenguaje de Definición de Datos

**Demo:** Es un tipo de licencia de uso de software y la descripción de las condiciones del mismo: demostrativo. Es una versión restringida; se puede apreciar el funcionamiento de los diferentes comandos (no siempre todos), y por lo general no tiene fecha de expiración. Suelen facilitar la ejercitación, pero impiden el guardado del resultado.

**Diagrama de Flujo:** Representación gráfica para representar paso a paso la solución de un problema.

**DoD:** Department of Defense, Departamento de Defensa de los Estados Unidos, una de sus agencias - DARPA- fue la responsable de la invención de la Internet.

**DOM:** Document Object Model (*Modelo de Objetos de Documento*) es una forma de representar documentos estructurados (tales como una página web HTML o un documento XML) que es independiente de cualquier lenguaje orientado a objetos. Su finalidad es definir el conjunto de objetos que pueden componer documentos HTML (páginas web) o XML, así como las estructuras que se definen dentro de él, sus propiedades y sus métodos, independientemente del lenguaje de programación utilizado, con el fin de evitar problemas de compatibilidad entre navegadores. El custodio del DOM es el *World Wide Web Consortium* (W3C).

**DTD:** Definición de Tipo de Documento:

**E**

**Extranet:** Parte de una intranet de acceso disponible a clientes y otros usuarios ajenos a la compañía

**F**

**FilePro (read-only):** Es una marca registrada de fP Technologies, Inc. Mas información sobre filePro puede encontrarse en <http://www.fptech.com/>.

**Frame:** Cuadro, marco. Instrucciones en el lenguaje HTML (utilizado para diseñar las páginas Web); una forma de dividir la pantalla del navegante en varias zonas, cada una con autonomía de movimiento. Por ejemplo, se puede dividir una pantalla de modo que haya un frame vertical que ocupe el lado izquierdo de la pantalla durante toda la navegación, que contenga el menú de un sitio Web. Los frames son un agregado al HTML estándar inventado por la empresa Netscape y luego adoptados como norma.

**Frame-relay:** Tecnología de transporte de datos por paquetes utilizada muy comúnmente en las conexiones por líneas dedicadas.

## H

**Hipermedia:** Combinación de hipertexto y multimedia. Uno de los grandes atractivos de la Web.

**Host name:** Nombre de un host. Denominación otorgada por el administrador a una computadora. El hostname es parte de la dirección electrónica de esa computadora, y debe ser único para cada máquina conectada a Internet.

## I

**IDEF1:** Integrated Definition Methods. Métodos de Integración Definidos, Cubren una gama de aplicaciones de la función que modela a la información, a la simulación, al análisis y al diseño orientado al objeto y a la adquisición de conocimiento.

**IEC:** International Electrotechnical Committee. Comité Electrotécnico Internacional, organismo internacional que se ocupa de normalizaciones y seguridad.

**IEEE:** Institute of Electrical and Electronics Engineers. Importante asociación de técnicos y profesionales, con sede en los Estados Unidos. Fue fundada en 1884 y en 1998 tenía aproximadamente 320.000 miembros en 147 países. Favorece la investigación en campos diversos, como la tecnología aeroespacial, la computación, las comunicaciones y la tecnología biomédica. Promueve la estandarización de normas.

**Informix:** Gestor de base de datos creado por Informix Software Inc. Incluye un RDBMS basado en SQL, un lenguaje de cuarta generación y juegos de herramientas para la inclusión del SQL en programas de aplicación.

**Ingeniería Web:** Proceso para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.

**Internet:** La red de computadoras más extendida del planeta, que conecta y comunica a más de 50 millones de personas. Nació a fines de los años sesenta como ARPANet, y se convirtió en un revolucionario medio de comunicación. Su estructura técnica se basa en millones de computadoras que ofrecen todo tipo de información. Estas computadoras, encendidas las 24 horas, se llaman servidores y están interconectadas entre sí, en todo el mundo, a través de diferentes mecanismos de líneas dedicadas. Sin importar de qué tipo de computadoras se trate, para comunicarse utilizan el protocolo TCP/IP. Las computadoras que utilizan las personas para conectarse y consultar los datos de los servidores se llaman clientes, y acceden en general a través en un tipo de conexión llamado dial-in, utilizando un módem y una línea telefónica.

**Intranet:** red de redes de una empresa. Su aspecto es similar al de las páginas de Internet

**IP:** Internet Protocol. Protocolo de Internet definido en el RFC 791. Confirma la base del estándar de comunicaciones de Internet. El IP provee un método para fragmentar (deshacer en pequeños paquetes) y rutear (llevar desde el origen al destino) la información. Es inseguro, ya que no verifica que todos los fragmentos del mensaje lleguen a su destino sin perderse en el

**ISO:** International Organization for Standardization. Fundada en 1946, es una federación internacional que unifica normas en unos cien países. Una de ellas es la norma OSI (Open Systems Interconnection: Interconexión de Sistemas Abiertos. Norma universal para protocolos de comunicación), modelo de referencia universal para protocolos de comunicación.

**Items:** Párrafo. Objeto, elemento de una base de datos.

## K

**Kernel:** Núcleo o parte esencial de un sistema operativo (programa que administra los demás programas en una computadora). Provee los servicios básicos del resto del sistema.

## L

**LDAP:** Lineal Data Access Protocol, Protocolo línea de acceso a datos.-

**Lenguaje de Programación:** Sistema de escritura para la descripción precisa de algoritmos (conjunto de reglas bien definidas para la resolución de un problema. Un programa de software es la transcripción, en lenguaje de programación, de un algoritmo) o programas informáticos.

**M**

**Marketing:** Publicidad de negocios. Es el conjunto de actividades que la empresa informativa desarrolla para satisfacer necesidades del mercado con el fin de hacer lo que le conviene al lector (Ver Media-Publicidad).

**Método:** Modo ordenado de proceder para llegar a un resultado o fin determinado, esp. para descubrir la verdad y sistematizar los conocimientos.

**Metodología:** Se refiere a los métodos de investigación en una ciencia. Aun cuando el término puede ser aplicado a las artes cuando es necesario efectuar una observación o análisis más riguroso o explicar una forma interpretar la obra de arte.

**Métricas:** Medidas

**Modelo Entidad-Relación:** Utilizado para construir un modelo de datos conceptual (representación de la estructura y contenido de una base de datos que es independiente del Sistema Manejador de Bases de Datos)

**O**

**ODBC:** Open DataBase Connectivity, un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de *ODBC* es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (*DBMS* por sus siglas en Ingles) almacene los datos, *ODBC* logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el *DBMS*, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el *DBMS* entienda. Para que esto funcione tanto la aplicación como el *DBMS* deben ser compatibles con *ODBC*, esto es que la aplicación debe ser capaz de producir comandos *ODBC* y el *DBMS* debe ser capaz de responder a ellos. desde la versión 2.0 el estándar soporta SAG y SQL.

**OLE:** Object Linking and Embedding. Tecnología Microsoft que permite crear documentos mediante incorporación de elementos creados utilizando diferentes tipos de software.

**OLTP:** Procesamiento de transferencia en línea.

**On-line:** En línea, conectado. Estado en que se encuentra una computadora cuando se conecta directamente con la red a través de un dispositivo, por ejemplo, un módem (modulador-demodulador. Dispositivo periférico que conecta la computadora a la línea telefónica).

**P**

**Página Web:** Page o Webpage. Unidad que muestra información en la Web. Una página puede tener cualquier longitud, si bien equivale por lo general a la cantidad de texto que ocupan dos pantallas y media. Las páginas se diseñan en un lenguaje llamado HTML, y contienen enlaces a otros documentos. Un conjunto de páginas relacionadas componen un site.

**Password:** Clave o contraseña. Palabra utilizada para validar el acceso de

**Píxel:** combinación de "picture" y "element". Elemento gráfico mínimo con el que se componen las imágenes en la pantalla de una computadora.

**Portal:** Sitio web que sirve de punto de partida para navegar por Internet. Los portales ofrecen una gran diversidad de servicios: listado de sitios web, buscador, noticias, e-mail, información meteorológica, chat, newsgroups (grupos de discusión) y comercio electrónico. En muchos casos el usuario puede personalizar la presentación del portal. Algunos de los más conocidos son Altavista, Yahoo!, Netscape y Microsoft.

**PostgreSQL:** Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

**Protocolo:** Lenguaje que utilizan dos computadoras para comunicarse entre sí.

**Protocolo COM:** Es un dominio de Internet genérico que forma parte del sistema de dominios de Internet. El dominio .net es uno de los dominios originales de Internet, fue establecido en enero de 1985 y actualmente es manejado por la compañía VeriSign.

Los dominios .com fueron creados para utilización comercial, sin embargo, actualmente no existen restricciones particulares para el registro de nuevas direcciones. En los años 1990, .com se convirtió en el dominio más frecuentemente utilizado para sitios web, especialmente los de uso comercial, dando nombre a las empresas punto com. La introducción del dominio genérico .biz, restringido a negocios, no ha tenido mayor impacto en la popularidad de .com.

**Protocolo FTP:** File Transfer Protocol: Protocolo de Transferencia de Archivos. Sirve para enviar y recibir archivos de Internet.

**Protocolo HTTP:** Hypertext Transfer Protocol. Protocolo de transferencia de hipertextos. Es un protocolo que permite transferir información en archivos de texto, gráficos, de video, de audio y otros recursos multimedia.

**Protocolo IMAP:** Internet Message Access Protocol. Protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet. Una vez configurada la cuenta IMAP, puede especificar las carpetas que desea mostrar y las que desean ocultar, esta característica lo hace diferente del protocolo POP. Por defecto utiliza el puerto 143.

**Protocolo LDAP:** *Lightweight Directory Access Protocol*, en sí es un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP puede considerarse una base de datos (aunque su sistema de almacenamiento puede ser otro diferente) al que pueden realizarse consultas. Habitualmente, almacena la información de login (usuario y password) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados...). En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.

**Protocolo NNTP:** Network News Transport Protocol (NNTP), o protocolo de transferencia de noticias. Es el Protocolo de red utilizado por el Usenet internet service. Es un Protocolo de red basado en tiras de textos enviados sobre canales TCP de 7 bit ASCII . Es usado para subir y bajar así como para transferir artículos entre servidores.

**Protocolo POP3:** Post Office Protocol 3. Protocolo 3 de Correo. Es un protocolo estándar para recibir e-mail.

**Protocolo SNMP: Simple Network Management Protocol (SNMP),** protocolo simple de gestión de redes, es el protocolo de gestión de red más importante y usado en la actualidad. Forma parte del conjunto de protocolos TCP/IP y esta definido en la capa de aplicación del mismo. SNMP busca la sencillez y es por ello que en la capa de transporte esta soportado por el protocolo UDP (caracterizado por su rapidez).

**Protocolo TCP/IP:** Transfer Control Protocol / Internet Protocol. Es el protocolo que se utiliza en Internet.

**PWC:** Portal Web Corporativo

## R

**RDBMS:** Es un Sistema Administrador de Bases de Datos Relacionales. RDBMS viene del acrónimo en inglés *Relational Data Base Manager System*. Los RDBMS proporcionan el ambiente adecuado para gestionar una base de datos.

**RTF:** Rich Text Format, lenguaje de descripción desarrollado por Microsoft para intercambiar información entre programas multiplataforma de edición de texto. El RTF es un pobre formato estandarizado con diversas incompatibilidades incluso entre distintas aplicaciones de Microsoft, y rara vez se usa para guardar documentación.

## S

**SACs:** Sequence Analysis and Consulting Service. Análisis de Secuencia y Servicio de Consulta

**SAX:** Simple API for XML. Es un evento basado en la interfaz para el procesamiento de documentos XML

**Script:** Guión. Serie secuencial de instrucciones que permite realizar tareas sencillas y repetitivas, generalmente son interpretadas en tiempo de ejecución, aunque hay sistemas que permiten compilar los guiones. Algunos de los sistemas de Scripts son verdaderos lenguajes de programación.

**Servidor Proxy:** Proxy Server (intermediario, mediador). Utilizado en relación a Internet, hace referencia a un servidor que media entre el usuario (su computadora) y otro servidor de la Red. El Proxy Server puede hacer, por ejemplo, un pedido de información para un cliente en lugar de que el cliente lo haga directamente (método usado para salir de un firewall). También pueden actuar como traductores de formato de archivos (por ejemplo, convertir toda imagen GIF que pase por ellos en un BMP, o traducir del inglés al castellano, o convertir los attachments), o como cachés (almacenando en un directorio los archivos de mayor uso reciente, para entregarlos ante una nueva solicitud sin necesidad de que el usuario los busque por toda Internet), verificar la seguridad (virus, accesos permitidos, etc.), entre otras muchas tareas.

**Servlet:** son objetos que corren dentro del contexto de un servidor de aplicaciones (ej: Tomcat) y extienden su funcionalidad. La palabra *servlet* deriva de otra anterior, *applet*, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un *servlet* es un programa que se ejecuta en un servidor web. El uso más común de los *servlets* es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

**Set:** Secure Electronic Transactions, Transacciones Electrónicas Seguras. Un estándar para pagos electrónicos encriptados que está siendo desarrollado por Mastercard, Visa y otras empresas. Superior y más complejo que el SSL.

**SPICE:** Simulation Program Integrated Circuits Especially (Programa integrado Especializado en la Simulación de Circuitos). Fue desarrollado por la Universidad de California, Berkeley en 1975 por Donald Pederson. Es un estándar internacional cuyo objetivo es simular circuitos electrónicos analógicos compuestos por resistencias, condensadores, diodos, transistores, etc. Para ello hay que describir los componentes, describir el circuito y luego elegir el tipo de simulación (temporal, en frecuencia, en continua, paramétrico, Monte Carlo...).

**SQL:** Structured Query Language. Lenguaje de programación que se utiliza para recuperar y actualizar la información contenida en una base de datos. Fue desarrollado en los años 70 por IBM. Se ha convertido en un estándar ISO y ANSI.

**SSI:** Las Server Side Includes son comentarios especiales dentro de paginas HTML que, cuando son leídas por el servidor realizan acciones especiales. Estas acciones van desde incluir "trozos" de HTML hasta ejecutar un programa e incluir el resultado dentro de la página.

**Streaming:** Transferencia Continua. Sistema de envío continuo de información, que permite por ejemplo, ver un video a medida que se lo está bajando de la Red.

**Sybase:** Sybase ofrece a las empresas la infraestructura completa necesaria para lograr integrar sus distintas tecnologías. Gracias a su innovador portal empresarial (Enterprise Portal, EP), sus soluciones móviles y sin cables, sus sistemas de gestión de bases de datos de alto rendimiento y sus productos de integración pioneros, Sybase es uno de los proveedores de software independientes más importantes del mundo.

## T

**Test:** Pruebas o exámenes con ayuda de un cuestionario o escala que mide determinadas funciones, generalmente cognitivas

**Thumbnails:** Miniatura de imagen. Imagen que es una representación en escala reducida de otra. Se usan generalmente para crear vistas de galerías de imágenes.

**Triggers:** Disparadores. Automatizar acciones que usualmente se ejecutan antes o después de una consulta, de tal manera que al llevar a cabo la consulta se "dispare" la ejecución de determinadas acciones, sin intervención humana.

## U

**UI:** User Interface. Componente del GRID que posee los programas necesarios para que el usuario pueda enviar trabajos al grid, controlar su estado y recibir su salida.

**UIDs:** *User ID* (ID de usuario). Es el medio por el cual se identifica a un usuario en las distintas partes de un sistema RHL. Los UIDs son numéricos, aunque hay nombres en formato entendible por personas que se almacenan en el fichero */etc/passwd*.

**UNIX:** Sistema operativo diseñado por los Laboratorios Bell y refinado en Berkley entre otros lugares, que soporta operaciones multiusuario, multitasking y soporta a los estándares abiertos. Ampliamente difundido en la Internet, es utilizado para ejecutar en los servidores.

**URL:** Localizador Universal de Recursos /Identificador Universal de Recursos. Sistema unificado de identificación de recursos en la red. Es el modo estándar de proporcionar la dirección de cualquier recurso en Internet.

## W

**W3C:** World Wide Web Consortium. Organización que desarrolla estándares para guiar el desarrollo y expansión de la Web. Organizado por el CERN y el MIT y apadrinado por varias empresas. Su Website es <http://www.w3.org>. Ver Sistemas Abiertos.

**WAP:** Wireless Application Protocol. Norma internacional para aplicaciones que utilizan la comunicación inalámbrica, por ejemplo el acceso a Internet desde un teléfono celular.

**Warehousing:** Es el centro de la arquitectura para los sistemas de información en la década de los '90. Soporta el procesamiento informático al proveer una plataforma sólida, a partir de los datos históricos para hacer el análisis. Facilita la integración de sistemas de aplicación no integrados. Organiza y almacena los datos que se necesitan para el procesamiento analítico, informático sobre una amplia perspectiva de tiempo.

**Web Browser:** Navegador de Web Programas que permiten navegar por el WWW. Los más populares son Mosaic y Netscape.

**WebTV:** Tecnología en la que convergen la televisión y la World Wide Web. Un televisor permite recorrer las páginas web usando un navegador y un control remoto. La señal llega a través de un módem conectado a una línea telefónica.

**World Wide Web o W3 o WWW:** Conjunto de servidores que proveen información organizada en sitios, cada uno con cierta cantidad de páginas relacionadas. La Web es una forma novedosa de organizar toda la información existente en Internet a través de un mecanismo de acceso común de fácil uso, con la ayuda del hipertexto y multimedia. El hipertexto permite una gran flexibilidad en la organización de la información, al vincular textos disponibles en todo el mundo. La multimedia aporta color, sonido y movimiento a esta experiencia, haciendo versátil y rico el contenido. El contenido de la Web se escribe en lenguaje HTML y puede utilizarse intuitivamente mediante un programa llamado navegador. Se convirtió en el servicio más popular de Internet y se emplea cotidianamente para los usos más diversos: desde leer un diario hasta participar de un juego grupal.

## **X**

**XSLT:** XSL Transformaciones es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página web, directamente en un dispositivo de presentación, como el monitor de un usuario. Actualmente, XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

# PIENSSA



## *Bibliografía*

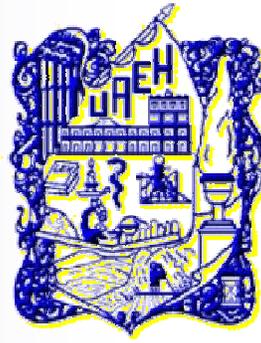
- [1] Boehm, B.; Brown, J.R.; Kaspar, J.R., et al. 1978, "*Characteristics of Software Quality*", TRW Series of Software Technology.
- [2] Catherin Bidar F., Jorge Mujica A., Dr. Marcelo Visconti Z., Web Testing, Aspectos Teóricos y Practicos, Universidad Federico Santa Maria, Departamento de Informática, Programa de Magíster en Informática, Chile, 2002.
- [3] CCS Test Suite, W3C Core Styles. <http://www.w3.org/Style/CSS/>, Marzo de 2001.
- [4] D. Cowan and C. Lucena. Abstract Data Views: An Interface Specification Concept to Enhance Design for Reuse. IEEE Transactions on Software Engineering. Vol. 21, No. 3, March 1995.
- [5] D. Lange. An object-oriented design approach for developing hypermedia information systems. Journal of Organizational Computing and Electronic Commerce, 6(3),269-293. 1996.
- [6] Davis, A., 1993, "*Software Requirements, Objects, Functions and States*", Prentice Hall
- [7] Departamento de Innovación. Subdirección de Sistemas - DCAA – UNAM
- [8] Deshpande, Y., Hansen, S; Murugesan, S.; 1999, "Web Engineering: beyond CS, IS, and SE", Proceed. of ICSE '99 (Int'l Conference on Software Engineering) Web Engineering Workshop, Los Angeles, US, pp 10-16.
- [9] E. Gamma *et al.* Design Patterns: Elements of reusable object-oriented software, Addison-Wesley, 1995.
- [10] Eva.Arderiu, Javier.Conde. Objectivity/DB and JAVA, [http://wwwinfo.cern.ch/asd/cernlib/rd45/objy\\_Java\\_info.htm#Servlets](http://wwwinfo.cern.ch/asd/cernlib/rd45/objy_Java_info.htm#Servlets), Marzo de 2001.
- [11] F. Garzotto; L. Mainetti and P. Paolini. Hypermedia design analysis. Communications of the ACM, 8(38), 74-86. 1995.
- [12] Fenton, N.E., 1991, "*Software Metrics a Rigorous Approach*", Chapman & Hall, London. [Goldberg et al 95] Goldberg, A.; Rubin, K., 1995, "*Succeeding with Objects: decision frameworks for project management*", Addison-Wesley.

- [13] Garzotto, F.; Mainetti, L.; Paolini, P., 1995, "Hypermedia Design, Analysis, and Evaluation Issues", CACM 38,8 (Ago-95); pp 74-86.
- [14] Gilb, T., 1988, "*Principles of Software Engineering Management*", AddisonWesley.
- [15] Grady, R.B., 1994, "*Successfully Applying Software Metrics*", IEEE Computer, 27 (9), pp. 18-25.
- [16] H. Lee; C. Lee and C. Yoo. A scenario-based object-oriented methodology for developing hypermedia information systems. En: Proceedings of 31st Annual Conference on Systems Science, Eds. Sprague R. 1998.
- [17] IEEE Std 1061-1992, "*IEEE Standard for a Software Quality Metrics Methodology*", IEEE Computer Society Press
- [18] IEEE Web Publishing Guide, <http://www.ieee.org/web/developers/style/>
- [19] ISO/IEC 9126-1 International Standard (Draft), 1998 "*Information technology – Software product evaluation – Part1: Quality Model*".
- [20] ISO/IEC 9126-1991 International Standard, "Information technology – Software product evaluation – Quality characteristics and guidelines for their use".
- [21] ISO/IEC 9126-2 International Standard, 1999 (Draft) "*Information technology – Software product evaluation – Part1: External Metrics*".
- [22] ISO/IEC 9126-3 International Standard, 1999 (Draft) "*Information technology – Software product evaluation – Part1: Internal Metrics*".
- [23] ISO/IEC 9126-4 International Standard, 1999 (Draft) "*Information technology – Software product evaluation – Part 1: Quality in use Metrics*".
- [24] J. Conallen. Building Web application with UML. Addison Wesley. 1999.
- [25] Kitchenham, B.A., Walker, J.G. 1989, "*A Quantitative Approach to Monitoring Software Development*" Software Engineering Journal, 4(1), pp. 2-13.

- [26] Lowe, D., Hall, W.; 1999, "Hypermedia and the Web: an Engineering Approach", John Wiley & Sons
- [27] MAG. Luis Antonio Olsina, Ingeniería del Software en la Web. Metodología Cuantitativa para la Evaluación de la Calidad de los Sitios Web, Facultad de Ciencias Exactas, Universidad Nacional de la Plata, Argentina, Noviembre 1999.
- [28] McCall, J.A; Richards, P.K.; Walters, G.F.; 1977, "*Factors in Software Quality*", RADC TR-77-369.
- [29] Michael H. Kay, About Saxon, <http://users.iclway.co.uk/mhkay/saxon/instant.html>, Marzo de 2001.
- [30] N. Koch. Comparing Development Methods for Web Applications. Ludwig-Maximilians-University Munich, Institute of Computer Science Oettingenstr. 67, 80538 München, Germany. 2000.
- [31] O. De Troyer and C. Leune. WSDM: A user-centered design method for Web sites. En: Proceedings of the 7th International World Wide Web Conference. 1997.
- [32] Olsina, L., "*View of a Process Model to Develop Hypermedia*" (in Spanish), Proceed. of the IV Congress of the SCCC (Computer Science Chilean Society), Valdivia, Chile, 1996, pp. 219-228.
- [33] Olsina, L., 1997, "*Applying the Flexible Process Model to build Hypermedia Products*". Hypertext and Hypermedia: Tools, Products, Methods (HHTPM 97), Paris, France, Ed. Hermes, Vol I N. 1-2-3 pp. 211-221.
- [34] Olsina, L., 1998, "*Toward Improvements in Hypermedia Process Modeling*", Proceed. at the Symposium of Software Technology, 27 JAIIO, Bs As., pp. 219-226
- [35] Olsina, L., 1998, "*Functional View of the Hypermedia Process Model*". The Fifth International Workshop on Engineering Hypertext Functionality at ICSE '98 (Int'l Conference on Software Engineering) , Kyoto, Japan, (Disponible electrónicamente en <http://www.ics.uci.edu/~kanderso/htf5/papers/>).

- [36] Olsina, L.; Rossi, G.; 1999, "*Towards Web-site Quantitative Evaluation: defining Quality Characteristics and Attributes*", Proceedings of IV Int'l WebNet Conference, World Conference on the WWW and Internet, Hawaii, USA, Vol 1-2, pp.834-839.
- [37] Powell, T.; Jones, D; Cutts, D., 1998, "Web Site Engineering: Beyond Web Page Design", Prentice Hall PTR.
- [38] Pressman, R., 1998, "Can Internet-based Applications be Engineered", IEEE Software (Sep/Oct 98).
- [39] Rosenfeld, L., Morville, P., 1998, "Information Architecture for the WWW", O`Reilly.
- [40] Sommerville, I., 1992, "Software Engineering", Addison-Wesley, 4th Ed.
- [41] T. Isakowitz; E. Stohr and P. Balasubramanian. A methodology for the design of structured hypermedia applications. Communications of the ACM, 8(38), 34-44. 1995.
- [42] Thüring, M.; Hannemann, J.; Haake, J., 1995, "Hypermedia and Cognition: Designing for Comprehension", Comm. ACM 38, 8 (Aug 95) pp. 57-66
- [43] WWW Consortium, 1999, W3C Working Draft, "WAI Accessibility Guidelines: Page Authoring", W3C, 1999, W3C Working Draft, "WAI Accessibility Guidelines: Page Authoring", <http://www.w3c.org/TR/WD-WAI-PAGEAUTH/>

# PIENSSA

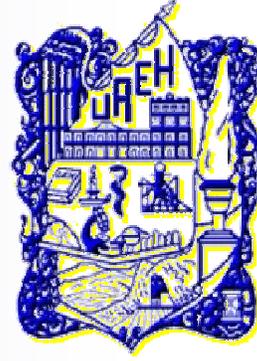


## *Referencias Electrónicas*

- [44] <http://depa.pquim.unam.mx/Curso/dreamnote.html>
- [45] <http://depa.pquim.unam.mx/Curso/quehtml.html>
  - [46] [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)
  - [47] <http://iteso.mx/~carlosc/pagina/documentos/usabilidad.htm#estructura>
  - [48] <http://mailweb.udlap.mx/~progra/Ayuda/Guias/tutorial/conceptos/quejava.html>
  - [49] <http://satlab.mor.itesm.mx:8080/biditec/tesis/CueMati2003MelissaBlanquetoEstrada.pdf>
  - [50] <http://tejo.usal.es/inftec/2003/DPTOIA-IT-2003-002.pdf>
  - [51] <http://www.accedis.com/queesusabilidad/index.asp>
  - [52] <http://www.anip.org.ar/normas/ieee.html>
  - [53] <http://www.dlsi.ua.es/webe02/articulos/5.pdf>
  - [54] <http://www.elsitiomaestro.com>
  - [55] <http://www.faq-mac.com/mt/archives/008257.php>
  - [56] <http://www.faq-web.com/mt/archives/008257.php>
  - [57] <http://www.gomezvalverde.com/raul/documentos/Calidad%20web.pdf>
  - [58] [http://www.htmlpoint.com/javascript/corso/js\\_02.htm](http://www.htmlpoint.com/javascript/corso/js_02.htm)
  - [59] [http://www.htmlweb.net/asp/curso\\_webmaster/cap\\_1/cap\\_1\\_1.html](http://www.htmlweb.net/asp/curso_webmaster/cap_1/cap_1_1.html)
  - [60] [http://www.htmlweb.net/disenio/estructura/creacion\\_1.html](http://www.htmlweb.net/disenio/estructura/creacion_1.html)
  - [61] <http://www.ihlo.com.uk/sqlnotes.html>

- [62] <http://www.lawebdelprogramador.com/apuntes/normas.html>
- [63] <http://www.mailxmail.com/curso/informatica/phpprogramacionweb/capitulo7.htm>
- [64] <http://www.mailxmail.com/informatica/capitulo7.htm>
- [65] <http://www.microsoft.com/spain/sql/70/productinfo/features.asp>
- [66] <http://www.procomer.com/software/Presentaciones%20Congreso/CertificacionISO031102.pdf>
- [67] <http://www.pst.informatik.uni-muenchen.de/personen/kochn/ideas03-escalona-koch.pdf>
- [68] <http://www.software.net.mx/portal/esp/somos.htm> (La Asociación Mexicana para la Calidad de Ingeniería de Software (AMCIS)).
- [69] <http://www.software-shop.com/Productos/MySQL/mysql.html>
- [70] [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)
- [71] [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)
- [72] [http://www.uoc.edu/masters/softwarelibre/cat/materials/Desarrollo\\_web.pdf](http://www.uoc.edu/masters/softwarelibre/cat/materials/Desarrollo_web.pdf)
- [73] <http://www.utopia-planitia.de/>
- [74] <http://www.virtual.unlar.edu.ar/catedras-virtuales/seminarios-de-act-tecnologica-3/2003art-disc-x-alumn/014r.pdf>
- [75] <http://www.w3.org/hypertext/WWW/Theproject.html>

# PIENSSA



## *Anexos*

## Anexo A

Este anexo muestra el archivo PortalWeb.Htm, generado por PIENSSA, el cual constituye la base del Portal Web Corporativo obtenido de manera dinámica y en tiempo real, cada vez que el usuario lo solicite.

### Archivo PortalWeb.Htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>PORTAL WEB CORPORATIVO</title>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>
</head>
<body bgcolor='#FFFFFF' background= 'no' align = 'Left' border = '0' bordercolor= '#FFFF00' link='#000000'
vlink='#0000FF' alink='#FF00FF' >
<table width='100%' border='0' cellspacing='0' bordercolor='#FF0000'>
<tr>
<td>
<table width='100%' cellspacing='0' align = 'Left' border = '3' bordercolor= '#FF0000' background=
'C:\PIENSSA\CUBA\Imagenes\SAGAR.BMP' bgcolor= '#FFFF00'>
<tr>
<td>
<table cellspacing='0' width = '230' height = '120' align = 'Left' border = '0' bordercolor= '#FFFF00' background=
'no' bgcolor= '#FFFFFF'>
<tr>
<td>
<div align='center'>
<img src=' C:\PIENSSA\imagenes\logo inteligente.GIF' width= '184' height= '48' align= 'Base Line'>
<div>
</td>
</tr>
</table>
</td>
<td>
<div align='center'>
<table cellspacing='0' width = '450' height = '110' align = 'Left' border = '3' bordercolor= '#FF00FF' background=
'C:\PIENSSA\CUBA\Imagenes\ALIANZ~1.BMP' bgcolor= '#FFFF00'>
<tr>
<td>
<div align='center'>
<img src=' C:\PIENSSA\imagenes\Titulo Inteligente.GIF' width= '225' height= '55' align= 'Base Line'>
<div>
</td>
</tr>
</table>
</td>
<td>
<table cellspacing='0' width = '230' height = '120' align = 'Left' border = '0' bordercolor= '#FFFF00' background=
'no' bgcolor= '#FFFFFF'>
<tr>
<td>
<div align='center'>
<img src=' C:\PIENSSA\imagenes\Logotipo Computadora.JPG' width= '138' height= '120' align= 'Base Line'>
</div>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

```

</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<table width='100%' cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor=
'#CCCC99'>
<tr>
<td>
<a href='C:\Inetpub\wwwroot\misitio\search.htm' target='_blank'><img src='C:\PIENSSA\imagenes\Buscar.JPG'
width='32' height='32' border='0' align='top'></a>
<a href='C:\Inetpub\wwwroot\misitio\Imprimir.htm' target='_blank'><img
src='C:\PIENSSA\imagenes\Impresora.JPG' width='32' height='32' border='0' align='top'></a>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<table cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor= '#CCCC99'>
<tr>
<td>
<table cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor= '#CCCC99'>
<tr>
<td width='200' height='400'><iframe name='FraMenu' src='MenuWeb.htm' width='100%' height='100%'
caption='MENU'></iframe> <div align='center'></div>
</tr>
</table>
</td>
<td>
<table width='100%' cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor=
'#FFFFFF'>
<tr>
<td width='600' height='400'>
<iframe name='FraDespliegue' src='DespliegueWeb.htm' width='100%' height='100%'></iframe> <div
align='center'>
</td>
</tr>
</table>
</td>
<td>
<table cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor= '#CCCC99'>
<tr>
<td>
<table width='100%' cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor=
'#CCCC99'>
<tr>
<td width='200' height='120'><iframe name='FraAvisos' src='AvisosWeb.htm' width='100%'
height='100%'></iframe> <div align='center'></div>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<table width='100%' cellspacing='0' align = 'Left' border = '0' bordercolor= '#FFFF00' background= 'no' bgcolor=
'#CCCC99'>
<tr>

```



## Anexo B

Este anexo muestra el archivo Documenta.Txt, generado dinámicamente por PIENSSA y en el cual se explica de manera detallada, la función que realiza cada una de las líneas de código de los archivos que integran el Portal Web Corporativo que el usuario ha creado utilizando esta aplicación.

### Archivo Documenta.Txt

```
DESARROLLO DE PORTALES WEB CORPORATIVOS CON CALIDAD
PIENSSA
DOCUMENTACION TECNICA DEL PORTAL WEB CORPORATIVO
=====
```

#### DATOS DE CREACION

```
-----
Nombre del Archivo: PortalWeb.Htm
Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'
Título del Portal: PORTAL WEB CORPORATIVO
Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'
Fecha y Hora: 30/08/2005 12:14:57 a.m.
```

#### CUERPO <body>

```
-----
Color de Fondo (bgcolor): #FFFFFF
Imagen de Fondo (background): no
Alineación (align): Left
Borde (border): 0
Color del Borde (bordercolor): #FFFF00
Color del Enlace (link): #000000
Color del Enlace Visitado (vlink): #0000FF
Color del Enlace Activo (alink): #FF00FF
```

#### TABLA PRINCIPAL <table>

```
-----
Ancho (Ancho (width)): 100%
Borde (border): 0
Espaciado de la Celda (cellspacing): 0
Color del Borde (bordercolor): #FF0000
```

#### TABLA DE CABECERA <table>

```
-----
Ancho (width): 100%
Espaciado de la Celda (cellspacing): 0
Alineación (align): Left
Borde (border): 3
Color del Borde (bordercolor): #FF0000
Imagen de Fondo (background): C:\PIENSSA\CUBA\Imágenes\SAGAR.BMP
Color de Fondo (bgcolor): #FFFF00
```

#### ELEMENTOS DE LA CABECERA

##### LOGOTIPO #1

```
-----
Tabla de Logotipo1
.....
Espaciado de la Celda (cellspacing): 0
Ancho (width):
Alto (height):
```

Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #FFFFFF

Archivo de Logotipo1

.....  
 Alineación (align): center  
 Imagen (img scr): C:\PIENSSA\imagenes\logo inteligente.GIF  
 Ancho (width): 0  
 Alto (height): 0  
 Alineación (align): Base Line

TITULO

-----

Tabla de Título

.....  
 Alineación (align): center  
 Espaciado de la Celda (cellspacing): 0  
 Ancho (width):  
 Alto (height):

Tabla de Título

.....  
 Alineación (align): Left  
 Borde (border): 3  
 Color del Borde (bordercolor): #FF00FF  
 Imagen de Fondo (background): C:\PIENSSA\CUBA\Imagenes\ALIANZ~1.BMP  
 Color de Fondo (bgcolor): #FFFF00

Archivo de Título

.....  
 Alineación (align): center  
 Imagen (img scr): C:\PIENSSA\imagenes\Titulo Inteligente.GIF  
 Ancho (width): 0  
 Alto (height): 0  
 Alineación (align): Base Line

LOGOTIPO #2

-----

Tabla de Logotipo2

.....  
 Espaciado de la Celda (cellspacing): 0  
 Ancho (width):  
 Alto (height):  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #FFFFFF

Archivo de Logotipo2

.....  
 Alineación (align): center  
 Imagen (img scr): C:\PIENSSA\imagenes\Logotipo Computadora.JPG  
 Ancho (width): 0  
 Alto (height): 0  
 Alineación (align): Base Line

BARRA DE HERRAMIENTAS

-----  
 Tabla de la Barra de Herramientas <table>

.....  
 Ancho (width): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0  
 Color del Borde (bordercolor): #CCCCCC

Tabla de la Barra de Herramientas <table>

.....  
 Ancho (width): 100%  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

Elementos de la Barra de Herramientas

.....  
 Herramienta: 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\search.htm  
 Destino (target): \_blank  
 Archivo (img scr): C:\PIENSSA\imagenes\Buscar.JPG  
 Ancho (width): 32  
 Alto (height): 32  
 Borde (border): 0  
 Alineación (align):top  
  
 Herramienta: 3  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Imprimir.htm  
 Destino (target): \_blank  
 Archivo (img scr): C:\PIENSSA\imagenes\Impresora.JPG  
 Ancho (width): 32  
 Alto (height): 32  
 Borde (border): 0  
 Alineación (align):top

TABLA DE ELEMENTOS CENTRALES <table>

-----  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

MENU DE OPCIONES

-----  
 Tabla de Menú <table>

.....  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

ARCHIVO GENERADO PARA EL MENU DE OPCIONES

-----  
 Nombre del Archivo: C:\MenuWeb.htm  
 Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'  
 Título del Archivo: Menú del Portal Web Corporativo  
 Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'  
 Fecha: 06/05/2005  
 Hora: 06:15:52 p.m.

CUERPO <BODY>

-----  
 Color de Fondo (bgcolor): #CCCC99  
 Imagen de Fondo (background): no  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Color del Enlace (link): #000000  
 Color del Enlace Visitado (vlink): #0000FF  
 Color del Enlace Activo (alink): #FF00FF

Tabla de Menú <table>

-----  
 Ancho (width): 100%  
 Alto (height): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0

División <div>

-----  
 Alineación (align): center  
 font size:= 4  
 face: Arial, Helvetica, sans-serif  
 strong: MENU

Opciones del Menú

-----  
 Menú: 1  
 Alineación (align): left  
 Fuente (face): Arial, Helvetica, sans-serif  
 Tamaño (size): 4  
 Color (color): Amarillo  
 Formato del Párrafo: 3  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Default.htm  
 Destino (target): FraDespliegue  
 Etiqueta = Objetivos

Menú: 2  
 Alineación (align): left  
 Fuente (face): Arial, Helvetica, sans-serif  
 Tamaño (size): 1  
 Color (color): Amarillo  
 Formato del Párrafo: 6  
 Referencia (href): enlace del 2  
 Destino (target): FraDespliegue  
 Etiqueta = otro dos modificado

Menú: 3  
 Alineación (align): left  
 Fuente (face): Arial, Helvetica, sans-serif  
 Tamaño (size): 4  
 Color (color): Amarillo

Formato del Párrafo: 3  
Referencia (href): C:\Inetpub\wwwroot\misitio\news.htm  
Destino (target): FraDespliegue  
Etiqueta = Novedades

Menú: 4  
Alineación (align): left  
Fuente (face): Arial, Helvetica, sans-serif  
Tamaño (size): 4  
Color (color): Amarillo  
Formato del Párrafo: 3  
Referencia (href): C:\Inetpub\wwwroot\misitio\products.htm  
Destino (target): FraDespliegue  
Etiqueta = Productos

Menú: 5  
Alineación (align): left  
Fuente (face): Arial, Helvetica, sans-serif  
Tamaño (size): 4  
Color (color): Amarillo  
Formato del Párrafo: 3  
Referencia (href): C:\Inetpub\wwwroot\misitio\services.htm  
Destino (target): FraDespliegue  
Etiqueta = Servicios

Menú: 6  
Alineación (align): left  
Fuente (face): Arial, Helvetica, sans-serif  
Tamaño (size): 4  
Color (color): Amarillo  
Formato del Párrafo: 3  
Referencia (href): C:\Inetpub\wwwroot\misitio\Default.htm  
Destino (target): FraDespliegue  
Etiqueta = Contáctenos

Menú: 7  
Alineación (align): center  
Fuente (face): Arial, Helvetica, sans-serif  
Tamaño (size): 5  
Color (color): Verde  
Estilo: strong, em  
Formato del Párrafo: 4  
Referencia (href): www.yahoo.com.mx  
Destino (target): FraDespliegue  
Etiqueta = Visitenos

División de la Tabla <td>

.....  
Ancho (width): 200  
Alto (height): 400  
Nombre del Marco Interno (iframe name): FraMenu  
Pantalla (scr): MenuWeb.htm  
Ancho (width): 100%  
Alto (height): 100%  
Título (caption): = MENU  
Alineación (align): center

AREA DE DESPLIEGUE

-----

Tabla de Area de Despliegue <table>

.....  
 Ancho (width): 100%  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #FFFFFF

ARCHIVO GENERADO PARA EL AREA DE DESPLIEGUE

-----  
 Nombre del Archivo: C:\DespliegueWeb.htm  
 Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'  
 Título del Archivo: Area de Despliegue del Portal Web Corporativo  
 Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'  
 Fecha: 06/05/2005  
 Hora: 06:15:52 p.m.

CUERPO <body>

-----  
 Color de Fondo (bgcolor): #FFFFFF  
 Imagen de Fondo (background): no  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Color del Enlace (link): #000000  
 Color del Enlace Visitado (vlink): #0000FF  
 Color del Enlace Activo (alink): #FF00FF

Tabla de Area de Despliegue

.....  
 table Ancho (width): 100%  
 Alto (height): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0  
  
 División de la Tabla <td>  
 .....  
 Ancho (width): 600  
 Alto (height): 400  
 Nombre del Marco Interno (iframe name): FraDespliegue  
 Pantalla (scr): DespliegueWeb.htm  
 Ancho (width): 100%  
 Alto (height): 100%

Tabla de Elementos Centrales

.....  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

AVISOS

-----  
 Tabla de Avisos <table>  
 .....  
 Ancho (width): 100%

Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

ARCHIVO GENERADO PARA LOS AVISOS

Nombre del Archivo: C:\AvisosWeb.htm  
 Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'  
 Título del Archivo: Avisos del Portal Web Corporativo  
 Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'  
 Fecha: 06/05/2005  
 Hora: 06:15:52 p.m.

CUERPO <body>

.....  
 Color de Fondo (bgcolor): #CCCC99  
 Imagen de Fondo (background): no  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Color del Enlace (link): #000000  
 Color del Enlace Visitado (vlink): #0000FF  
 Color del Enlace Activo (alink):#FF00FF

Tabla de Avisos <table>

.....  
 Ancho (width): 100%  
 Alto (height): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0

División de la Tabla de Avisos <div>

.....  
 Alineación (align): center  
 Tamaño de la Fuente (size): 4  
 Fuente (face): Arial, Helvetica, sans-serif  
 strong: AVISOS

Lista de Avisos <UL>

.....  
 Aviso: 1  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Av01.htm  
 Destino (target): FraDespliegue  
 Descripción = Aviso #1  
  
 Aviso: 3  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Av03.htm  
 Destino (target): FraDespliegue  
 Descripción = Aviso #3  
  
 Aviso: 7  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2

Referencia (href): enlace7  
 Destino (target): FraDespliegue  
 Descripción = siete

Aviso: 8  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio2\archive.htm  
 Destino (target): FraDespliegue  
 Descripción = Ochito

Aviso: 9  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): D:\Inetpub\wwwroot\prueba.htm  
 Destino (target): FraDespliegue  
 Descripción = Aviso # 9 Modificado

División de la Tabla <div>

.....  
 Ancho (width): 200  
 Alto (height): 120  
 Nombre del Marco Interno (iframe name): FraAvisos  
 Pantalla (scr): AvisosWeb.htm  
 Ancho (width): 100%  
 Alto (height): 100%

NOTICIAS

-----

Tabla de Noticias <Table>

.....  
 Ancho (width): 100%  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

ARCHIVO GENERADO PARA LAS NOTICIAS

-----

Nombre del Archivo: C:\NoticiasWeb.htm  
 Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'  
 Título del Archivo: Noticias del Portal Web Corporativo  
 Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'  
 Fecha: 06/05/2005  
 Hora: 06:15:52 p.m.

CUERPO <body>

.....  
 Color de Fondo (bgcolor): #CCCC99  
 Imagen de Fondo (background): no  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Color del Enlace (link): #000000  
 Color del Enlace Visitado (vlink): #0000FF  
 Color del Enlace Activo (alink): #FF00FF

Tabla de Noticias <table>

.....  
 Ancho (width): 100%  
 Alto (height): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0

División de la Tabla de Noticias <div>

.....  
 Alineación (align): center  
 Tamaño de la Fuente (size): 4  
 Fuente (face): Arial, Helvetica, sans-serif  
 strong: NOTICIAS

Lista de Noticias (UL)

.....  
 Noticia: 1  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Nt01.htm  
 Destino (target): FraDespliegue  
 Descripción = Noticia #1

Noticia: 2  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): EnlaceNoticia 2  
 Destino (target): FraDespliegue  
 Descripción = Noticia 2 de Ejemplo

Noticia: 3  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\Nt03.htm  
 Destino (target): FraDespliegue  
 Descripción = Noticia #3

División de la Tabla <td>

.....  
 Ancho (width): 200  
 Alto (height): 120  
 Nombre del Marco Interno (iframe name): FraNoticias  
 Pantalla (scr): NoticiasWeb.htm  
 Ancho (width): 100%  
 Alto (height): 100%  
 Alineación (align): center

ANUNCIOS

-----

Tabla de Anuncios <table>

.....  
 Ancho (width): 100%  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99

ARCHIVO GENERADO PARA LOS ANUNCIOS

-----

Nombre del Archivo: C:\AnunciosWeb.htm  
 Tipo de Archivo: !DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN'  
 Título del Archivo: Anuncios del Portal Web Corporativo  
 Contenido: meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'  
 Fecha: 06/05/2005  
 Hora: 06:15:52 p.m.

CUERPO <body>

.....  
 Color de Fondo (bgcolor): #CCCC99  
 Imagen de Fondo (background): no  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Color del Enlace (link): #000000  
 Color del Enlace Visitado (vlink): #0000FF  
 Color del Enlace Activo (alink): #FF00FF

Tabla de Anuncios <table>

.....  
 Ancho (width): 100%  
 Alto (height): 100%  
 Borde (border): 0  
 Espaciado de la Celda (cellspacing): 0

División de la Tabla de Anuncios <div>

.....  
 Alineación (align): center  
 Tamaño de la Fuente (size): 4  
 Fuente (face): Arial, Helvetica, sans-serif  
 strong: ANUNCIOS

Lista de Anuncios <UL>

.....  
 Anuncio: Anuncio #1  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\An01.htm  
 Destino (target): FraDespliegue  
 Descripción: Anuncio #1  
  
 Anuncio: Anuncio #3  
 Alineación (align): left  
 Tamaño de la Fuente (size): 2  
 Referencia (href): C:\Inetpub\wwwroot\misitio\An03.htm  
 Destino (target): FraDespliegue  
 Descripción: Anuncio #3

División de la Tabla de Anuncios <td>

.....  
 Ancho (width): 200  
 Alto (height): 120  
 Nombre del Marco Interno (iframe name): FraAnuncios  
 Pantalla (scr): AnunciosWeb.htm  
 Ancho (width): 100%  
 Alto (height): 100%  
 Alineación (align): center

BARRA DE ESTADO

-----

Tabla de Barra de Estado <table>

.....  
 Ancho (width): 100%  
 Espaciado de la Celda (cellspacing): 0  
 Alineación (align): Left  
 Borde (border): 0  
 Color del Borde (bordercolor): #FFFF00  
 Imagen de Fondo (background): no  
 Color de Fondo (bgcolor): #CCCC99  
 Alineación (align): right  
 strong  
 Color de la Fuente (color): #000000  
 Tamaño (size): 2  
 Fuente (face): Arial, Helvetica, sans-serif

.....  
 Documento generado por PIENSSA. (30/08/2005 12:14:59 a.m.)  
 .....

## Anexo C

Este anexo muestra una de las 2 plantillas de la metodología RUP generadas dinámicamente por PIENSSA, cuyo objetivo es promorcionar una base documental del Portal Web Corporativo creado por el usuario y al mismo tiempo le ofrece la posibilidad de complementar la información del mismo.

### Archivo VisiondelNegocio.Txt

-----  
 <Nombre de la Compañía>  
 -----

PORTAL WEB CORPORATIVO

Visión del Negocio

Versión 1.0

Nombre del Documento: VisiondelNegocio.Txt

REVISION DE LA HISTORIA

Fecha:30/08/2005 12:14:59 a.m.

Version 1.0

Descripción: Archivo que contiene la Visión del Negocio del Portal Web Corporativo creado por PIENSSA

Autor: PIENSSA (Sistema Prototipo para el Desarrollo de Portales Web Corporativos con Calidad

#### TABLA DE CONTENIDOS

<b>1</b>	<b>Introducción</b>	
1.1	Propósito	
1.2	Alcance	
1.3	Definición, Siglas y Abreviaturas	
1.4	Referencias	
1.5	Revisión	
<b>2</b>	<b>Posicionando</b>	
2.1	Oportunidades del Negocio	
2.2	Declaración del Problema	
2.3	Declaración de la Posición del Producto	
<b>3</b>	<b>Stakeholder y Descripciones del Cliente</b>	
3.1	Mercados Demográficos	
3.2	Resumen de Stakeholder	
3.3	Resumen del Usuario	
3.4	Entorno del Usuario	
3.5	Perfiles del Stakeholder	
3.5.1	<Nombre del Stakeholder>	
3.6	Perfiles del Cliente	
3.6.1	<Nombre del Cliente>	
3.7	Clave del Stakeholder o Necesidades del Cliente	
3.8	Alternativas y Competencias	
<b>4</b>	<b>Objetivos del Modelado del Negocio</b>	
4.1	<Un Objetivo>	
4.2	<Otro Objetivo>	
<b>5</b>	<b>Limitaciones</b>	

**6 Rangos de Calidad**

**7 Anterioridad y Prioridad**

**8 Otros Requisitos**

- 8.1 Estándares Aplicables
- 8.2 Requisitos del Sistema
- 8.3 Requisitos de Presentación
- 8.4 Requisitos del Entorno

Apéndice 1 - Atributos de los Objetivos

VISION DEL NEGOCIO

**1 Introduccion**

[La introducción de la Visión del Negocio proporciona una apreciación global completa del documento. Esto debe incluirse en el propósito, alcance, definiciones, siglas, abreviaturas, referencias y revisión de la Visión del Negocio.]

- 1.1 Propósito  
[Especificar el propósito del documento de la Visión del Negocio.]
- 1.2 Alcance  
[Una descripción breve del alcance de este documento de la Visión del Negocio; que proyecto(s) asociado con y nada sino que es afectado o influenciado por este documento.]
- 1.3 Definición, Siglas y Abreviaturas  
[Este apartado proporciona las definiciones de todos los términos, siglas, y abreviaciones que requiere interpretar la Visión del Negocio propiamente. Esta información puede ser proporcionada por referencia al Glosario del proyecto.]
- 1.4 Referencias  
[Este apartado puede proporcionar una lista completa de todos los documentos que hacen referencia a otra parte de la Visión del Negocio. Identificado cada documento por título, número del informe (si es aplicable), fecha, y organización publicada. Especificando las fuentes para las cuales las referencias pueden ser obtenidas. Esta información puede ser proporcionada por la referencia del apéndice u otro documento.]
- 1.5 Revisión  
[Este apartado describe lo que el resto de la Visión del Negocio contiene y explica cómo el documento es organizado.]

**2 Posicionando**

- 2.1 Oportunidades del Negocio  
[Brevemente describa la oportunidad del negocio siendo encontrado por este proyecto.]
- 2.2 Declaración del Problema  
[Proporcione una declaración resumiendo el problema a resolver por este proyecto. Puede usarse el formato siguiente.]  
  - El Problema de: [descripción del problema]
  - Afecta: [quien es el stakeholders afectado por el problema]
  - El Impacto del cual es: [cual es el impacto del problema]
  - Una Solución exitosa sería: [enliste algunos beneficios de la clave de una solución exitosa]
- 2.3 Declaración de la Posición del Producto  
[Proporcione un declaración resumiendo, al nivel más alto, la única

posición que el producto piensa obtener en el mercado. Puede usarse el siguiente formato.]

Para: [cliente asignado]  
 Quien: [declaración de la necesidad u oportunidad]  
 El (nombre del producto): Es un [categoría del producto]  
 Que: [declaración de la clave beneficiaria; que es, cual es la razón de competencia a comprar]  
 Al contrario de: [alternativa competitiva primaria]  
 Nuestro Producto: [declaración de diferenciación primaria]

[Una declaración de la posición de producto comunica el intento de la aplicación y la importancia del proyecto a todo el personal interesado.]

### 3 Stakeholder y Descripciones del Cliente

[Para proporcionar los productos y servicios que satisfacen sus stakeholders' y las necesidades reales de usuarios eficazmente, es necesario identificar e involucrar todo el stakeholders como parte del Modelado del proceso del Negocio. Usted también debe identificar a los usuarios del sistema y asegurar que la comunidad del stakeholder los representa adecuadamente. Esta sección proporciona un perfil de los stakeholders y los usuarios involucrados en el proyecto, y los problemas importantes que ellos perciben para ser dirigidos por la solución propuesta. No describe sus demandas específicas o requisitos como éstos se captura en un artefacto de demandas de stakeholder separado. En cambio, proporciona el fondo y justificación de porque los requisitos son necesarios.]

#### 3.1 Mercados Demográficos

[Resuma el mercado demográfico importante que motiva sus decisiones del producto. Describa y posicione segmentos del mercado asignado. Estime el tamaño del mercado y crecimiento usando el número de usuarios potenciales o la cantidad de dinero que sus clientes gastan intentando satisfacer necesidades que su producto o el perfeccionamiento cumplirá. Revise las tendencias de las industrias mayores y tecnologías. Conteste estas preguntas estratégicas

¿Cuál es la reputación de su organización en los mercados?

¿Qué le gustaría ser?

¿Cómo hace este producto o servicio soportar sus metas?]

#### 3.2 Resumen de Stakeholder

[Hay varios stakeholders con un interés en el desarrollo y no todos ellos son usuarios terminales. Presente una lista resumida de éstos stakeholders del no-usuario. (Los usuarios se resumen en sección 3.3.)]

Nombre: [Nombre del tipo de stakeholder.]

Descripción: [Breve descripción del stakeholder.]

Responsabilidades: [Resuma las responsabilidades importantes de los stakeholder con respecto al sistema a desarrollarse; es decir, su interés como un stakeholder. Por ejemplo, este stakeholder:

- asegura que el sistema será mantenible
- asegura que habrá una demanda del mercado para los rasgos del producto
- monitorea el progreso del producto
- aprobaciones encontradas
- y tan adelante]

#### 3.3 Resumen del Usuario

[Presente una lista resumiendo a todos los usuarios identificados.]

Nombre: [Nombre del tipo de usuario.]

Descripción: [Breve descripción de lo que ellos representan con respecto al sistema.]

Responsabilidades: [Liste las claves de las responsabilidades de los usuarios con respecto al sistema a desarrollarse, por ejemplo:

- detalles de captura
- reportes de producciones
- trabajo de coordinaciones
- y algo así]

Stakeholder: [si el usuario no es representante directo, identifique cual stakeholder es responsable para representar el interés del usuario.]

### 3.4

#### Entorno del Usuario

[Detalle el ambiente del funcionamiento del usuario asignado. Aquí está algunas sugerencias:

¿Numero de personas involucradas para completar la tarea? ¿Este es cambiante?

¿Cuánto tiempo es un ciclo de la tarea? ¿La cantidad de tiempo gastado en cada actividad? ¿Este es cambiante?

Cualquier limitación del entorno: ¿móvil, campo, en-vuelo, y así sucesivamente?

¿Qué plataformas de los sistemas están en uso hoy? ¿Plataformas futuras?

¿Qué otras aplicaciones están en uso? ¿Necesita integrar su aplicación con ellos?

De donde extrae del Modelo del Negocio podría ser incluido para perfilar la tarea y los papeles involucrados y así sucesivamente.]

### 3.5

#### Perfiles del Stakeholder

[Describa aquí cada stakeholder en el sistema rellenando la tabla siguiente para cada stakeholder. Recuerde que los tipos del stakeholder pueden ser tan divergentes como los usuarios, secciones, y los diseñadores técnicos. Un perfil completo cubriría los temas siguientes para cada tipo de stakeholder.]

##### 3.5.1 <Nombre del Stakeholder>

Representante: [¿Quién es el stakeholder representante del proyecto? (Esto es opcional si tiene documentación en otra parte.) Lo que queremos aquí es nombres.]

Tipo: [Califique la experiencia del stakeholder, técnico de fondo, y el grado de sofisticación- esto es guru, negocios, experto, usuario casual, y así sucesivamente.]

Responsabilidades: [Liste las responsabilidades importantes del stakeholder con respecto al sistema que se desarrolla- esto es, su interés como stakeholder.]

Criterio de Exito: [¿Cómo definen el stakeholder el éxito? ¿Cómo se premian el stakeholder?]

Desenvolvimiento: [¿Cómo se desenvuelve el stakeholder en el proyecto? Relate, donde es posible, a las reglas del Rational Unified Process-esto es, Requerimientos, Requisitos Críticos y así sucesivamente.]

Liberación: [¿Hay cualquier liberación adicional requerido por el stakeholder? Éstos podrían ser liberaciones del proyecto o rendimientos del sistema en desarrollo.]

Comentarios y Problemas: [Aquí van los problemas que interfieren con éxito y cualquier otra información pertinente.]

### 3.6

#### Perfiles del Cliente

[Describa cada uso único del sistema aquí rellenando la tabla siguiente

para cada tipo de cliente. Un completo de los temas por cada tipo de usuario.]

3.6.1 <Nombre del Cliente>

Representante: [¿Quién es el usuario representante del proyecto? (Esto es opcional si tiene documentación en otra parte.) Esto se refiere a menudo al Stakeholder que representa el juego de usuarios, por ejemplo,

Stakeholder: Stakeholder1.]

Descripción: [Breve descripción del tipo de cliente.]

Tipo: [Califique la experiencia del cliente, técnica de fondo, y el grado de sofisticación- esto es, guru, usuario casual, y así sucesivamente]

Responsabilidades: [Liste las responsabilidades importantes del usuario con respecto al sistema que se desarrolla- esto es, captura detalles del cliente, produzca reportes, trabajos coordinados, y así sucesivamente.]

Criterio de Éxito: [¿Cómo definen el cliente el éxito? ¿Cómo se premian el cliente?]

Desenvolvimiento: [¿Cómo se desenvuelve el cliente en el proyecto? Relate, donde es posible, a las reglas del Rational Unified Process-esto es, Requerimientos, Requisitos Críticos y así sucesivamente.]

Liberación: [¿Hay cualquier liberación producida por el cliente, si es así, para quien?]

Comentarios y Problemas: [Aquí van los problemas que interfieren con éxito y cualquier otra información. Estos incluyen tendencias que hacen el trabajo del cliente más fácil o más difícil.]

3.7 Clave del Stakeholder o Necesidades del Cliente

[Liste los problemas importantes existentes con soluciones como percibió el stakeholder. Clasifique las ediciones siguientes para cada problema

¿Cuáles son las razones para este problema?

¿Cómo es resuelto ahora?

¿Qué soluciones hacen la necesidad del usuario?]

[Es importante entender la importancia relativa que el stakeholder pone para resolver cada problema. Alineando y las técnicas de la votación cumulativas indican problemas como los que deben resolverse opuestos a problemas que les gustaría dirigir.

Rellene la siguiente tabla-si usa Rational RequisitePro para capturar las Necesidades, éste podría ser un extracto o podría informar de esa herramienta.]

Necesidad: Transmite mensajes

Prioridad:

Preocupaciones:

Solución Actual:

Solución Propuesta:

3.8 Alternativas y Competencias

[Identifique como percibe alternativas el stakeholder. Éstos pueden incluir compra el producto de un competidor, construye una solución del homegrown o simplemente mantiene el estatus. Liste cualquier opción competitiva conocida que exista pueda ponerse disponible. Incluya las fuerzas mayores y debilidades de cada competidor como es percibido por el stakeholder.]

**4 Objetivos del Modelado del Negocio**

- 4.1 <Un Objetivo>
- 4.2 <Otro Objetivo>

**5 Limitaciones**

[Anotar alguna limitación asignada, limitaciones externas u otras dependientes.]

**6 Rangos de Calidad**

[Defina los rangos de calidad para la referencia, robustez, falta de tolerancia, utilidad, y las características similares que no se capturan en los objetivos.]

**7 Anterioridad y Prioridad**

[Defina la prioridad de los diferentes objetivos.]

**8 Otros Requisitos**

[A un nivel alto, lista las normas aplicables, hardware o requisitos de la plataforma, requisitos de la actuación, y los requisitos del entorno.]

- 8.1 Estándares Aplicables  
[Liste todos los estándares que el negocio debe obedecer. Éstos pueden incluir legal y regulador (FDA, UCC) los estándares de comunicaciones (TCP/IP, ISDN), estándares de complacencia de plataforma (Windows, UNIX, y así sucesivamente), y calidad y estándares de seguridad (UL, ISO, CMM)
- 8.2 Requisitos del Sistema  
[Defina cualquier requisito del sistema necesario para apoyar a la aplicación. Éstos pueden incluir los sistemas operativos del host apoyados y plataformas de la red, configuraciones, memoria, peripherals, y software adicional.]
- 8.3 Requisitos de Presentación  
[Use esta sección para detallar los requisitos de representación. Los problemas de la representación pueden incluir tales artículos como factores de carga de usuario, bandwidth o capacidad de comunicación, throughput, exactitud, y fiabilidad o la contestación cronometra bajo una variedad de condiciones cargadas.]
- 8.4 Requisitos del Entorno  
[Detalle los requisitos del entorno como necesidad. Para el hardware-sistema basado, los problemas del entorno incluyen temperatura, ajuste, humedad, radiación, y así sucesivamente. Para las aplicaciones del software, los factores del entorno incluyen las condiciones de uso, entorno del usuario, disponibilidad del recurso, emisión de mantenimiento, error de manejo y recuperación.]

Apéndice 1 - Atributos de los Objetivos

[Los objetivos son dados para evaluar el uso de atributos, rastree, priorice, y maneja los artículos del producto propuestos para la aplicación. Liste y brevemente describa los atributos que usted ha escogido. Vea el Artefacto: Plan de Dirección para un juego de atributos del rasgo sugeridos.]

.....  
Documento generado por PIENSSA. (30/08/2005 12:14:59 a.m.)  
.....