



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja



ESCUELA DE SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN

TEMA:

**“EL MODELADO DE NEGOCIO DE LA
INGENIERÍA WEB”**

*Memoria de Tesis previa a la obtención
del Título de Ingeniero en Sistemas
Informáticos y Computación.*

AUTOR: Luis Javier Capa León

DIRECTOR: Ing. Manuel Sucunuta España

CODIRECTOR: Ing. Armando Cabrera Silva

Loja – Ecuador
2008



CERTIFICACIÓN

Ingenieros

Manuel Sucunuta España

Armando Cabrera Silva

DIRECTOR DE TESIS y CODIRECTOR DE TESIS

C E R T I F I C A N:

Que el presente trabajo de investigación, previo a la obtención del título de INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN, ha sido dirigido, supervisado y revisado en todas sus partes, por lo mismo, cumple con los requisitos legales exigidos por la Universidad Técnica Particular de Loja, quedando autorizada su presentación.

Loja, 11 de junio del 2009

Ing. Manuel Sucunuta España

Ing. Armando Cabrera Silva



AUTORÍA

El presente proyecto de tesis previa a la obtención del Título de Ingeniero en Sistemas Informáticos y Computación; sus conceptos, análisis, conclusiones y recomendaciones emitidas, es de absoluta responsabilidad del autor.

Se indica además que la información de otros autores empleada en este trabajo está debidamente especificada en fuentes de referencia y apartados bibliográficos.

Luis Javier Capa León



CESIÓN DE DERECHOS

Yo, Luis Javier Capa León, declaro ser el autor del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos y acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja, que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través o con el apoyo financiero académico o institucional (operativo) de la Universidad”.

Luis Javier Capa León



AGRADECIMIENTO

Esta tesis representa la culminación de una etapa muy enriquecedora y el inicio de un camino aún más exigente. En toda la experiencia universitaria y la conclusión del trabajo de Tesis, existen personas que se merecen las gracias porque sin su valioso aporte no hubiera sido posible la culminación de este trabajo y también hay quienes las merecen por haber dejado su huella en mi camino.

A mis padres, les agradezco su apoyo, su guía y su confianza en la realización de mis sueños. Soy afortunado por contar siempre con su amor, comprensión y ejemplo. Esta tesis es suya. A Krmita, Daniel, Manuel, Diego y Galo por ser unos amigos increíbles y con quienes he compartido muchos momentos que siempre llevaré en mi corazón. Ustedes han enriquecido mi vida con su cariño y su alegría. Gracias por recordarme que hay personas valiosas en el mundo y gracias por estar en el mío

A mi director y co-director de Tesis, Ing. Manuel Sucunuta e Ing. Armando Cabrera respectivamente, quienes con su esfuerzo y dedicación. Sus conocimientos, sus orientaciones, su manera de trabajar, su persistencia, su paciencia y su motivación han sido fundamentales para la culminación de este trabajo. Ellos han inculcado en mí un sentido de seriedad, responsabilidad y rigor académico sin los cuales no podría tener una formación completa como profesional. A su manera, ha sido capaz de ganarse mi lealtad y admiración, así como sentirme en deuda con ellos por todo lo recibido durante el periodo de tiempo que ha durado esta tesis.

A todas las personas que han dejado huella en mi vida. Ustedes han sido fuente de alegría y muchos de ustedes fueron de gran ayuda en los días y noches en los que fue redactada esta tesis. A todos Uds., gracias desde el fondo de mi corazón...



DEDICATORIA

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mi madre Rosario.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Luis.

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mis familiares.

A mis hermanos Patricio y Danny por ser el ejemplo de unos hermanos mayores, de los cuales aprendí mucho en momentos difíciles; y a todos aquellos que participaron directa o indirectamente en la elaboración de esta Tesis. ¡Gracias a ustedes!

A mis maestros.

Por su gran apoyo y motivación para la culminación de mis estudios profesionales, por su tiempo compartido y por impulsar el desarrollo de mi formación.

A mis amigos.

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora, seguimos siendo amigos: Daniel, Manuel, Diego, Galo y Arabel por haberme ayudado a realizar este trabajo.



A la Universidad Técnica Particular de Loja y en especial a la Escuela de Ciencias de la Computación por permitirme ser parte de una generación de triunfadores y gente productiva para el país.

Javier Capa León



ÍNDICE GENERAL

CESIÓN DE DERECHOS	4
AGRADECIMIENTO	5
DEDICATORIA	6
OBJETIVOS	11
CAPÍTULO I	15
1 Una introducción a la Ingeniería Web	15
1.1 Evolución de la Web	15
1.2 Categorías de Aplicaciones Web.....	16
1.3 Características de las Aplicaciones Web.....	18
1.3.1 Características relacionadas al producto	20
1.3.2 Características relacionadas al uso.....	20
1.3.3 Características relacionadas al desarrollo	21
CAPÍTULO II	23
2 El Proceso de Ingeniería Web	23
2.1 Ingeniería de Software	23
2.2 Ingeniería de Requisitos para las Aplicaciones Web.....	25
2.3 Modelado de Aplicaciones Web:.....	28
2.3.1 Modelos específicos en la Ingeniería Web	29
2.3.2 Modelo de requerimientos	32
2.3.3 Modelo de Contenidos	34
2.3.4 Modelado de Hipertexto	36
2.3.5 Modelado de Adaptación	41
2.4 Arquitecturas de las Aplicaciones Web.....	43
2.4.1 Componentes genéricos de la arquitectura de una Aplicación Web	48
2.4.2 Arquitecturas por capas	49
2.5 Pruebas en Aplicaciones Web	51
2.6 Desarrollo de Aplicaciones Web.....	54
2.6.1 Gestión de requerimientos en el desarrollo de una Aplicación Web.....	56
CAPÍTULO III	60
3 Metodologías de desarrollo de Aplicaciones Web	60
3.1 UWAT+.....	60
3.1.1 Proceso de Desarrollo.....	60
3.2 WebML: Web Modeling Language.....	61
3.3 OOHDM: Object-Oriented Hypermedia Design Method	63
3.3.1 Proceso de desarrollo	63
3.4 UWE: UML based-web Engineering	64
3.4.1 Proceso de desarrollo	65
3.5 OO-H: Object-Oriented Hypermedia Method.....	66
3.5.1 Proceso de desarrollo	67
3.6 WSDM: Web Site Design Method.....	68
3.6.1 Proceso de desarrollo	68
3.7 AWE: Agil Web Engineering	69
3.7.1 El ciclo de vida del proceso AWE	70
3.8 Análisis de las Metodologías de Desarrollo Web	78
4 Métricas en las Aplicaciones Web	82
4.1 Evaluación de una Aplicación Web.....	82
4.2 La evaluación de aplicaciones web en la actualidad	83
4.2.1 Criterios de Evaluación o Heurísticas	84
4.3 Métricas Web Automatizables	86
4.3.1 Cantidad de enlaces rotos en un sitio.....	86
4.3.2 Cantidad total de enlaces en un sitio.....	86
4.3.3 Porcentaje de enlaces rotos de un sitio.....	87
4.3.4 Cantidad de nodos web muertos en un sitio.....	87
4.3.5 Cantidad de imágenes que referencian a la propiedad ALT.	87



4.3.6	Cantidad total de imágenes en un sitio.....	88
4.3.7	Porcentaje de presencia de la propiedad ALT.....	88
4.3.8	Cantidad total de páginas de un sitio.....	88
4.3.9	Promedio de enlaces por página.	88
4.3.10	Tamaño de una página.	88
4.3.11	Páginas de acceso rápido.....	89
4.3.12	Mantenimiento del color de los enlaces.....	89
4.3.13	Cantidad máxima de marcos o frames.	89
4.4	Metodología de Evaluación de Calidad de Sitios Web.....	89
4.4.1	Fases de Web-Site QEM.....	90
4.4.2	Herramientas de automatización de métricas.....	92
5	Caso de Estudio.....	97
5.1	Evaluación del Sitio Web de la UTPL.	97
5.2	Evaluación de las métricas propuestas.	114
6	CONCLUSIONES.....	122
7	RECOMENDACIONES.....	124
8	BIBLIOGRAFIA.....	125
9	DIRECCIONES ELECTRÓNICAS.....	129
ANEXO 1	131
ANEXO 2	145
ANEXO 3	157
ANEXO 4	173
ANEXO 5	186
ANEXO 6	197



ÍNDICE DE FIGURAS

Figura 1: Categorías de Aplicaciones Web	17
Figura 2: Dimensiones para la categorización de las Aplicación web.....	19
Figura 3: Dimensiones para la categorización de las Aplicación web.....	30
Figura 4: Diagrama de casos de uso del sistema de revisión	33
Figura 5: Ejemplo de diagrama de actividades para el proceso de envío y aceptación de papers	34
Figura 6: Diagrama de clases del sistema de revisión de papers.....	35
Figura 7: Diagrama de máquinas de estado para el sistema	36
Figura 8: Modelo de hipertexto para el sistema de revisión	37
Figura 9: Modelo de hipertexto para el sistema de revisión	41
Figura 10: Adaptación dinámica de una página en el modelo de presentación	43
Figura 11: Factores que influyen en el desarrollo de una arquitectura	47
Figura 12: Componentes genéricos de una arquitectura para una Aplicación Web	49
Figura 13: Arquitectura de dos capas para una Aplicación Web.....	50
Figura 14: Arquitectura multicapa para una Aplicación Web.....	51
Figura 15: Procesos vs. Métodos en el desarrollo de un proyecto software	54
Figura 16: Ilustración del desarrollo iterativo de software	55
Figura 17: Esquema del proceso de desarrollo UWAT+	61
Figura 18: Proceso de desarrollo WebML	62
Figura 19: Proceso de desarrollo OOHDM	64
Figura 20: Proceso de desarrollo OOHDM	66
Figura 21: Proceso de desarrollo OO-H	67
Figura 22: Proceso de desarrollo WSDM	69
Figura 23: Ciclo de vida del proceso AWE	71
Figura 24: Cronología de las principales metodologías de desarrollo de aplicaciones web ..	75
Figura 25: Principales módulos que intervienen en Web-Site QEM.....	91
Figura 26: Principales módulos que intervienen en Web-Site QEM.....	94
Figura 27: LinkScan 12.0 Pantalla principal	95
Figura 28: SiteSweeper 2.0. Pantalla principal.....	96
Figura 29: Resultados obtenidos sobre el sitio web de la UTPL en una escala sobre 100. 113	
Figura 30: Representación de la documentación existente.....	114
Figura 31: Opciones de LinkScan para presentar informes	115
Figura 32: Ejemplo errores encontrados, causa y solución.....	116
Figura 33: Gráfica de documentos y sus errores externos en los sitios web de las 3 Universidades	120
Figura 34: Gráfica de enlaces totales internos, rotos y sospechosos en los sitios web de las 3 Universidades	121

ÍNDICE DE TABLAS

Tabla 1: Comparación entre las metodologías de desarrollo de aplicaciones web	75
Tabla 2: Comparación entre requerimientos de necesitan las metodologías de desarrollo. 78	
Tabla 3: Comparación entre los requisitos de las distintas metodologías de desarrollo de aplicaciones web.....	81
Tabla 4: Cuadro comparativos principales características de herramientas para automatización de métricas	96
Tabla 5: Comparación de documentos y sus errores externos en los sitios web de las 3 Universidades	119
Tabla 6: Comparación de enlaces totales, rotos y sospechosos en los sitios web de las 3 Universidades	120
Tabla 7: Comparación de enlaces totales internos, rotos y sospechosos en los sitios web de las 3 Universidades.....	121



OBJETIVOS

Objetivo general

- ✓ Realizar un análisis sobre el proceso de Ingeniería Web en la actualidad, sus métodos y técnicas que marcan la pauta para el Análisis, Diseño, Construcción e Implementación de una aplicación web.

Objetivos específicos

- ✓ Estudiar y analizar métodos, técnicas y herramientas para realizar el modelado de negocio en el desarrollo de sistemas web.
- ✓ Presentación y análisis de propuestas existentes.
- ✓ Definir los conceptos fundamentales aplicados a la Ingeniería Web como variante de la Ingeniería de Software.
- ✓ Establecer plantillas de evaluación para una aplicación web con los parámetros mínimos que demanda el desarrollo de las mismas.
- ✓ Evaluar el sitio web de la Universidad Técnica Particular de Loja y emitir un informe con los conocimientos previos que brinde la investigación realizada.

Resultados Esperados

- ✓ Elaboración de un documento que sirva como base para entender el proceso de Ingeniería Web como variante de la Ingeniería del Software.
- ✓ Aplicación de las encuestas realizadas para determinar el proceso aplicado en la construcción del portal de la UTPL y elaboración del informe cuantitativo sobre su estado actual, en base a categorías y parámetros que se detallan a los largo de este trabajo.



Estrategias

- ✓ Para el presente trabajo se ha efectuado una investigación bibliográfica para poder determinar el estado del arte. Además se accedió a foros, revistas, congresos, workshops, direcciones web, libros, memorias de tesis, etc. de interés y relacionados con el tema tratado.
- ✓ Elaboración de las plantillas de trabajo para la evaluación del sitio web de la UTPL.

Actividades

- ✓ Entrevistas con los encargados del sitio web de la UTPL para obtener la información necesaria de cómo se efectuó el proceso de Análisis, Diseño, Construcción e Implementación



INTRODUCCIÓN

El crecimiento acelerado que ha tenido la Web en estos últimos años está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas de donde proviene, lo que ha provocado que todas las actividades de las personas tiendan a depender de la Web. Internet pasó de ser una forma de diversión a un ente de desarrollo constante.

Con tanta información empezando a crecer y a estar disponible, crearon un desafío para los ingenieros de software, a raíz de esto se crearon enfoques disciplinados, sistemáticos y metodologías donde tuvieron en cuenta aspectos específicos de este nuevo medio. Uno de los aspectos más explotados, en el desarrollo de sitios web, es sin duda alguna el diseño gráfico y la organización estructural del contenido. Para ello, expertos en el tema utilizan herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de las aplicaciones web.

Ahora para garantizar el buen funcionamiento y mantenimiento de los sitios web, este debe contar con ciertos atributos y características que en conjunto forman un concepto muy importante, para alcanzar el éxito en cualquier organización, herramienta, y todo aquello que se pueda considerar como servicio. Dicho concepto es la calidad, que con atributos como, usabilidad, navegabilidad, seguridad, entre otros, hace posible por un lado la eficiencia del artefacto web y por ende la satisfacción del usuario final.

En el presente trabajo se abordan diferentes conceptos que luego de una investigación y adaptación, pueden ser tomados como referencias. El capítulo 1 presenta algunos aspectos claves dentro de la evolución de la web, categorías y características que son afines a un mismo tipo de aplicación, y que en mayor o menor grado, determinan su usabilidad.



El capítulo 2 señala los detalles propios del proceso de ingeniería web como tal, los diferentes tipos de modelado que se utilizan, los procesos que se llevan a cabo, tipos de pruebas, los pasos en el desarrollo, todo desde una perspectiva orientada a la web tomando como base la ingeniería de software.

En el capítulo 3 se hace una introducción a las metodologías más utilizadas en el proceso de desarrollo de una aplicación web, así como comparativa entre todas y ciertas pautas para determinar la mejor, al momento de aplicarse a un proyecto real; para ello son necesarias ciertas consideraciones que son abordadas dentro del capítulo.

El capítulo 4 comprende todo lo relacionado con la usabilidad de una aplicación web, recomendaciones, consideraciones, aspectos que ayudan a mejorar un sitio web desde el diseño. El capítulo 5 presenta aspectos de seguridad en la web y algunos de los ataques más comunes que sufren ciertas aplicaciones de acuerdo al ámbito donde se desenvuelven. También se presentan un conjunto de métricas para las aplicaciones web en el capítulo 6 que se basan en la usabilidad para el usuario final y los parámetros que deben tomarse en cuenta para evaluar un sitio.

Finalmente, el capítulo 7 comprende un informe sobre la evaluación realizada al portal de la Universidad Técnica Particular de Loja, en base a plantillas adaptadas de Swebok¹ con lineamientos básicos que deben ser considerados al momento de desarrollar una aplicación web.

¹Swebok, Software Engineering Body of Knowledge, disponible en: <http://www.swebok.org/>



CAPÍTULO I

1 Una introducción a la Ingeniería Web

1.1 *Evolución de la Web*

Hoy en día cuando navegamos en Internet a menudo nos hemos encontrado con una cantidad de Sitios Web que han llegado para cambiar la forma de ver y hacer las cosas; para mostrarnos que se puede ir más allá del aspecto informativo. Todo se debe a los avances tecnológicos y al empuje que muchas compañías han puesto en aprovechar todas estas potencialidades e implementar un valor agregado a sus clientes.

Ha pasado mucho tiempo desde aquel día en que Tim Berners-Lee creó la WWW o simplemente Web, desde entonces muchas personas comparten conocimientos a través de una extensa red de computadores interconectados desde diversos lugares del mundo.

En los primeros años pocas personas tenían acceso al contenido de la red; documentos generalmente confidenciales por sus características de investigación o seguridad nacional. Con el paso del tiempo se observó el gran potencial que podría tener este grandioso invento y comenzó un gran intercambio de información con variedad de temáticas, y con ello apareció la necesidad de compartir dicha información de manera más sencilla, entonces se creó el navegador Mosaic el cual facilitaba la visualización a través de hipertexto.

La evolución que ha tenido la Web ha sido continua; llegaron las imágenes a acompañar los textos, aparecieron las primeras animaciones, las primeras herramientas interactivas y la evolución continuó. Durante estos años la Web ha sido siempre un sitio informativo, un sitio de lectura en donde pocas personas escribían y los demás nos dedicábamos a leer. Según muchos autores esto es lo que podría llamarse la Web 1.0 (claro está si gustan de las versiones), pero todo ha cambiado como veremos más adelante, la Web ha evolucionado.



La Web está sufriendo una evolución importante y marcada que merece mucha atención especialmente por la gran cantidad de contenidos y de aplicaciones realmente útiles y sorprendentes, como por ejemplo los servicios que prestan las redes sociales (facebook, twitter, wordpress, flickr, plurk, etc.). Por otra parte hoy en día sin requerir gran tecnología, conocimiento técnico o esfuerzo significativo, podemos sentarnos una tarde frente a nuestro computador y escribir, el tema es lo de menos, un equipo de fútbol, biología marina, viajes por el mundo, cualquier tema es válido. Para el usuario final todo se ha hecho más fácil simplemente, sabe que quiere hacer y dispone de herramientas sencillas para hacerlo. [EscalonaMJ, 2002].

Visualizando la evolución para las personas que estamos del otro lado, ésta es significativamente grande y en ocasiones confusa; pasamos de páginas estáticas hechas con HTML, Java Script e imágenes al uso de CMS y estándares como XHTML, CSS, P3P por nombrar unas cuantas tecnologías, métodos o formas de ver las cosas. Todo enfocado al “Usuario Final” y a su interacción y producción. La Web crece a medida que los usuarios colaboran usando para ello la Web como una plataforma. Esta evolución tecnológica es lo que muchos llaman ahora la Web 2.0; no es una versión, sino un concepto y una forma de ver las cosas, lo verdaderamente importante que se presenta hoy en día es quizá la revolución social gracias a las nuevas técnicas y herramientas para la Comunicación y Colaboración. [Ceballos].

1.2 Categorías de Aplicaciones Web

Las aplicaciones web tienen diferentes grados de complejidad. El desarrollo de aplicaciones web puede iniciarse desde categorías bajas e irse expandiendo hasta incrementar su grado de complejidad. Las nuevas categorías de aplicación web representan un grado mayor de dificultad pero ello no significa que reemplacen en su totalidad a las antiguas generaciones. Cada una de estas categorías tiene sus propios campos específicos. En consecuencia, la complejidad de las aplicaciones web puede fácilmente ser atribuida a algunas categorías de una vez. [Kappel, 2003]

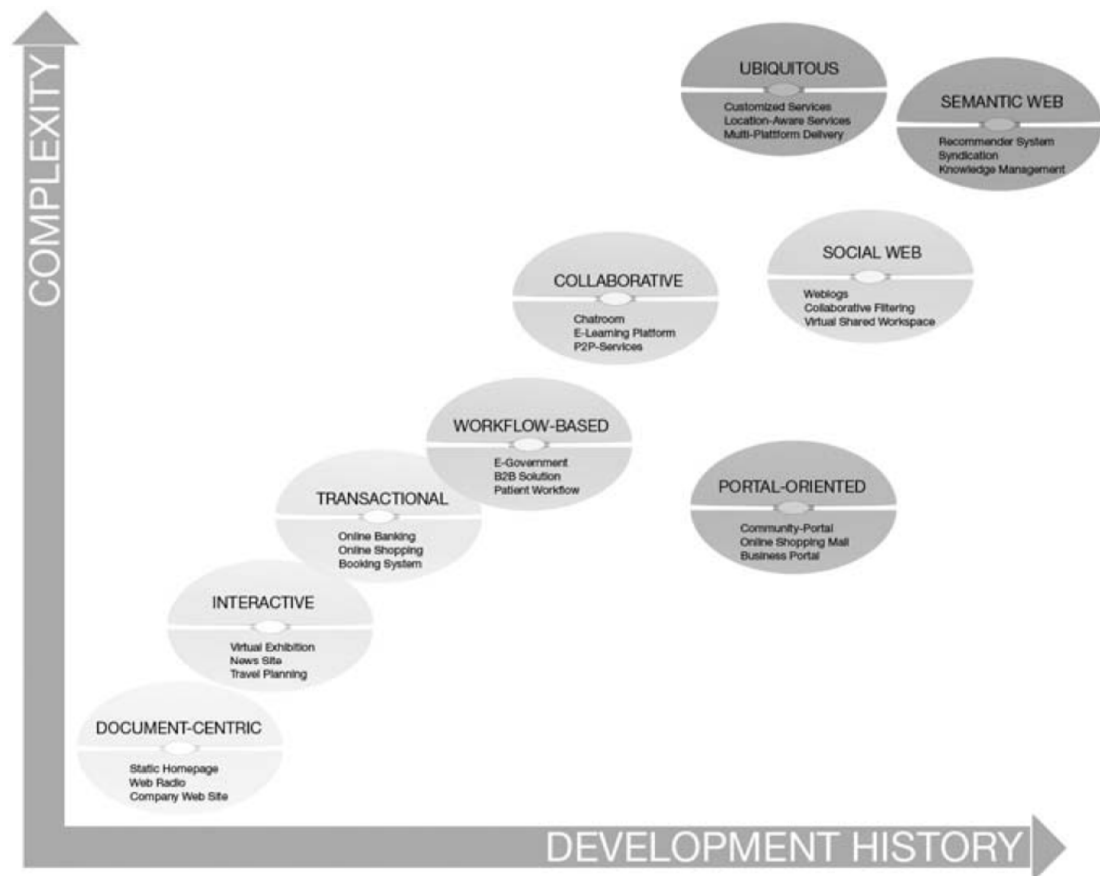


Figura 1: Categorías de Aplicaciones Web²

A continuación se describen los campos que se representan en el esquema anterior:

- ✓ Centrada en documentos (Document Centric): las páginas son almacenadas en un servidor web, de donde son enviadas al cliente en respuesta a las peticiones. Son usualmente actualizadas de forma manual mediante el uso de herramientas. Riesgo de inconsistencias e información desactualizada. Mayor simplicidad y estabilidad con tiempos de respuesta muy cortos.

² Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003, página 5.



- ✓ Interactiva (Interactive): ofrecen nuevas formas, aspectos de mejora en la interfaz y selecciones. Las páginas y los enlaces a otras son generados de forma dinámica.
- ✓ Transaccional (Transactional): fueron creadas para brindar mayor interactividad al usuario, permitiéndole acciones no sólo de lectura sino de actualización de los contenidos del sitio.
- ✓ Basada en flujo de trabajo (Workflow-based): permite la gestión de flujos de trabajo entre compañías, autoridades públicas y usuarios privados. La complejidad de los sistemas que se ofrecen, la autonomía de las compañías participantes y la necesidad de flujos de trabajos flexibles y robustos son los principales desafíos.
- ✓ Colaborativa (Collaborative): son utilizadas específicamente para fines colaborativos. Soportan la información compartida y workspaces para editar, generar y gestionar la información que se tiene compartida.
- ✓ Web Social (Social Web): son sitios donde los usuarios comparten su identidad a un grupo reducido de personas con los mismos intereses.
- ✓ Orientada a portales (Portal-Oriented): ofrecen un acceso a fuentes de información y/o servicios heterogéneos separados.
- ✓ Ubicua (Ubiquitous): proveen servicios adaptados para cualquier dispositivo en cualquier lugar y momento.
- ✓ Web Semántica (Semantic Web): el objetivo de la web semántica es brindar información no únicamente para el entendimiento humano sino para los propios sistemas que la manipulan. Lo que facilitaría la gestión del conocimiento en la web y su reutilización. [Kappel, 2003]

1.3 Características de las Aplicaciones Web

Las aplicaciones web difieren de las tradicionales en algunos campos. Estas características no están presentes en las aplicaciones tradicionales y dependen mucho del tipo de aplicación web que se esté desarrollando, por ejemplo una aplicación web transaccional demandará un mayor enfoque en sus detalles. Por estas características tan particulares, muchos conceptos, métodos, técnicas y herramientas de la ingeniería de software como las



pruebas de usuario, diagramas de navegabilidad entre otras, han tenido que ser adaptados a las necesidades de la ingeniería web o han resultado totalmente inadecuadas. La figura 2 nos ofrece una revisión de estas características agrupadas en tres dimensiones: producto, uso y desarrollo con su respectiva evolución. [Kappel, 2003]

Estas dimensiones están basadas en el estándar ISO/IEC 9126-1 para la evaluación de las características de calidad del software.

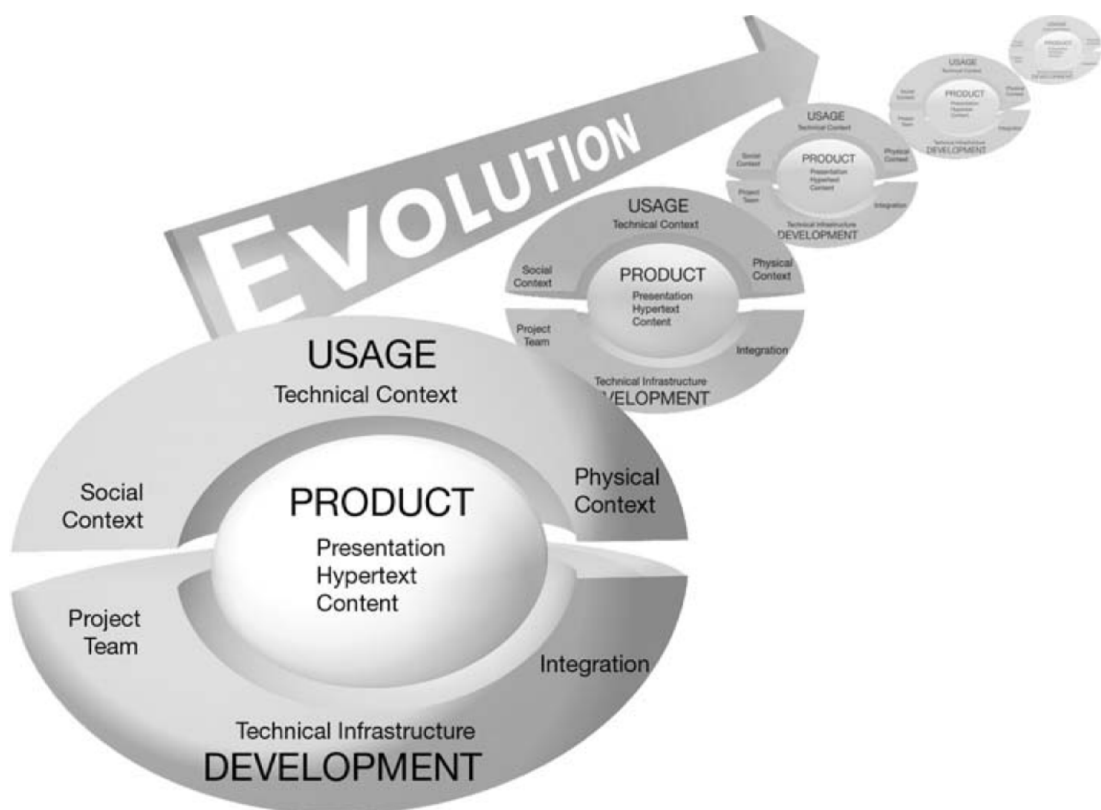


Figura 2: Dimensiones para la categorización de las Aplicación web³

³ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 8.



1.3.1 Características relacionadas al producto

Constituyen la mayor parte en la construcción de una nueva aplicación web. Aquí se abarca el contenido, el modelo de navegación y el modelo de presentación o interfaz de usuario, de acuerdo a lo que se señala en The Discipline of Systematic Development of Web Applications.

- ✓ Contenido: generar contenido, tenerlo disponible, integrarlo y actualizarlo es igualmente importante como lo es el desarrollo de la aplicación en sí. Los programadores no sólo deben actuar como constructores de la aplicación sino también como autores de la información.
- ✓ Modelo de navegación: entre las características propias de las aplicaciones web está la no linealidad de los documentos de hipertexto. Existen muchos modelos de hipertexto y una web como tal, define un modelo simple por su cuenta. Los elementos básicos de un modelo de hipertexto o navegación son los nodos, enlaces y la fuente. Un nodo es una unidad de información única e identificable. Un enlace es la ruta desde un nodo hacia otro. En la web, estas rutas son siempre unidireccionales y su significado no está claramente definido. Finalmente, la fuente es el espacio dentro del contenido de un nodo que es el origen o destino de un enlace.
- ✓ Modelo de presentación: dos características de las aplicaciones web a nivel de presentación son la estética y el ser intuitivas para el usuario. La estética es importante porque captura la atención del usuario y al ser de fácil uso, tendrá una utilización recurrente y cumplirá los objetivos para la que inicialmente fue diseñada.

1.3.2 Características relacionadas al uso

Comparada con aplicaciones tradicionales, el uso de aplicaciones web es muy heterogéneo. Los usuarios varían en número, tiempos y lugares de acceso que no pueden ser predichos, tienen diferentes componentes hardware y/o software [Kappel, 2003]. El uso de las aplicaciones es caracterizado por la necesidad del continuo cambio en las situaciones de



uso de los usuarios, o también llamados contextos. Estos contextos pueden ser divididos en tres grupos: contexto social, contexto técnico y contexto natural [Kappel, 2003].

- ✓ Contexto Social: (Usuarios). Se refiere a aspectos específicos de los usuarios tales como espontaneidad y multiculturalidad. Un usuario puede visitar una aplicación web cuando él lo desee y regresar en cualquier momento. Desde que la competencia en internet se ha visto incrementada, un usuario únicamente utilizará una aplicación web si ésta representa una ventaja inmediata sobre sus competidoras. La gran variedad de grupos de posibles usuarios hace difícil la tarea de seleccionar un grupo representativo para el análisis de requerimientos.
- ✓ Contexto Técnico: (Red y Dispositivos). Comprende lo relacionado a la conexión de red y su calidad en el servicio. Abarca los diferentes tipos de hardware y software que los usuarios utilizan para acceder a la aplicación web. Un usuario puede personalizar a su manera el estilo de navegación, sus accesos, plugins adicionales. El gran número de navegadores disponibles en el mercado también es un desafío que las aplicaciones web deben enfrentar, cada uno tiene sus propias funcionalidades y restricciones.
- ✓ Contexto Natural: (Lugar y Tiempo). Incluye aspectos de localización y tiempos de acceso. Maneja términos como globalidad y disponibilidad. La globalidad de una aplicación web demanda medidas de seguridad más elevadas que impidan el acceso deliberado o accidental de los usuarios a las áreas críticas del sistema. La disponibilidad involucra una estabilidad en las aplicaciones web, las cuáles se vuelven usables inmediatamente luego de su liberación obligando a que las exigencias de calidad en el producto desarrollado sean seguras y óptimas. [Escalona, 2003].

1.3.3 Características relacionadas al desarrollo

El desarrollo de aplicaciones web se caracteriza por el uso de los recursos necesarios como un equipo de desarrollo, la infraestructura técnica



necesaria, el proceso de desarrollo como tal y la integración de todas las soluciones disponibles en el momento: [Kappel, 2003]

- ✓ Equipo de desarrollo: se ve altamente influenciado por factores como un equipo de trabajo multidisciplinario y generalmente más joven que otro equipo de desarrollo de software tradicional. Estos factores y los métodos de la llamada comunidad de desarrollo contribuyen a una forma completamente nueva de colaboración organizada de los diferentes grupos de desarrollo.
- ✓ Infraestructura técnica: la falta de homogeneidad e inmadurez de los componentes usados son características importantes en la infraestructura técnica de las aplicaciones web. El desarrollo de aplicaciones web depende de dos factores externos como lo son el servidor y el navegador. Mientras que el servidor puede ser configurado para que tenga el comportamiento deseado por el equipo de desarrollo, no sucede lo mismo con los navegadores que poseen preferencias individuales y los conocidos plugins.
- ✓ Proceso de desarrollo: se ve influenciado por la flexibilidad y el paralelismo. Flexibilidad para adaptarse a nuevas condiciones de cambio y paralela dado que las aplicaciones web son desarrolladas a la par por subgrupos que son estructurados de acuerdo a los componentes autónomos de la aplicación, y no de acuerdo a la experiencia laboral como ocurre en las aplicaciones tradicionales.
- ✓ Integración: una característica importante de muchas aplicaciones web es la necesidad de integración interna y externa no solamente en aspectos técnicos sino también de contenido. Tanto en integraciones internas como externas con servicios o sin ellos, el nuevo sistema debe recopilar las viejas prestaciones y mejorarlas.



CAPÍTULO II

2 El Proceso de Ingeniería Web

2.1 Ingeniería de Software

IEEE define a la Ingeniería del Software como: “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software” [Swebok, 2004]. La ingeniería de software es una disciplina que concierne a todos los aspectos de la producción del software.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería del software. Los objetivos de la Ingeniería del Software son:

- ✓ Mejorar la calidad de los productos de software
- ✓ Aumentar la productividad y trabajo de los ingenieros del software.
- ✓ Facilitar el control del proceso de desarrollo de software.
- ✓ Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- ✓ Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

La ingeniería de software se relaciona con varias disciplinas como Gerencia, Matemáticas, Gestión de proyectos, Gestión de la calidad, Ingeniería de sistemas, Ergonomía del software. Abarca un conjunto de áreas del conocimiento que son la base fundamental para el desarrollo de un proyecto software; estas áreas son: Requerimientos, Diseño, Construcción, Pruebas, Mantenimiento, Gestión de la configuración, Gestión de la ingeniería, Procesos de ingeniería, Herramientas y métodos de la ingeniería,



Calidad del software; a continuación se detallan las más importantes [Swebok, 2004]:

- ✓ **Requerimientos de software:** un requerimiento se define como una exigencia que debe ser cumplida para dar solución a un problema del mundo real. Contiene subareas como: especificación de requerimientos, análisis, validación, clasificación, negociación, etc.
- ✓ **Diseño del software:** es el proceso de definir la arquitectura, componentes, interfaces y otras características relativas al sistema como tal. Fundamentos, claves en el diseño, estructura y calidad son algunas de las subareas que comprende el diseño de software.
- ✓ **Construcción del software:** se refiere a la creación detallada de un sistema software a través de la combinación de codificación, verificación, pruebas de unidad, pruebas de integración y depuración.
- ✓ **Pruebas del software:** consiste en la verificación dinámica del comportamiento de un software ante un conjunto limitado de casos de prueba. Contiene áreas como: niveles de pruebas, técnicas de pruebas, etc.
- ✓ **Mantenimiento del software:** las actividades de mantenimiento comienzan teóricamente cuando el producto final es liberado, pero en la práctica empieza desde etapas mucho más tempranas, debido a los cambios en las necesidades del usuario a las que la aplicación debe adaptarse.
- ✓ **Gestión de configuración del software:** es una disciplina que identifica la configuración del software en distintos puntos de su desarrollo con la finalidad de garantizar la integridad y trazabilidad que se puedan ver comprometidas cuando se presentan cambios en la configuración del software.



2.2 Ingeniería de Requisitos para las Aplicaciones Web

Un problema dentro de la industria del software es la falta de decisiones comunes sobre la terminología que se debe utilizar. Este apartado empieza dando una definición conjunta de requerimiento basándose tanto en el punto de vista del usuario como del desarrollador:

Un requerimiento es una especificación de lo que debería ser implementado. Son descripciones de cómo el sistema debe comportarse, es una propiedad o un atributo y son sobre todo restricciones en el proceso de desarrollo del sistema. [Torres Bosh, 2008].

Las especificaciones de los requerimientos no incluyen detalles de diseño e implementación, información sobre la planificación del proyecto, o planes de pruebas. Los proyectos como tal contemplan otra clase de requerimientos como el ambiente de desarrollo, el cronograma de actividades para la liberación de un nuevo componente.

Los requerimientos de aplicaciones al igual que los de software tradicional, comprenden 3 niveles distintos: requerimientos del negocio, del usuario y funcionales. [Kappel, 2003].

- ✓ **Requerimientos del Negocio:** Representan los objetivos de la empresa o usuario que solicita el sistema.
- ✓ **Requerimientos del Usuario:** Describen lo que el usuario de la aplicación web será capaz de realizar una vez que se haya implementado el mismo.
- ✓ **Requerimientos Funcionales:** Especifican las funcionalidades del software que los desarrolladores deben construir para que el usuario logre cumplir sus objetivos planteados.

Los desarrolladores utilizan los requerimientos funcionales y no funcionales para diseñar soluciones que implementen características de funcionamiento



y desempeño, a fin de lograr los objetivos de calidad especificada dentro de los límites que las restricciones lo permiten.

Todas las subdisciplinas que derivan de la ingeniería de requisitos, tales como: análisis, verificación, validación, etc., abarcan todas las actividades relacionadas con la obtención, evaluación y documentación de los requerimientos. La iteración es la clave para el éxito del proceso junto con un plan para múltiples ciclos de captura de requisitos, detallando y llegando a un acuerdo con el usuario cuando un requerimiento es aprobado en su totalidad. [Torres Bosh, 2008].

Se puede considerar correcto un requisito cuando éste cumpla con una serie de características como se detallan a continuación: [Kappel, 2003]

- ✓ Completo: Debe contener toda la información necesaria para que el desarrollador diseñe e implemente ese requerimiento de la mejor manera.
- ✓ Correcto: Cada requerimiento debe estar asociado a un objetivo que debe ser necesariamente satisfecho.
- ✓ Factible: El requerimiento debe estar enmarcado dentro de lo realizable en la aplicación web.
- ✓ Necesario: Debe estar basado en lo que el usuario realmente necesita hacer, no debe ser un derivado de otros requisitos.
- ✓ Priorizado: No todos los requerimientos tienen la misma importancia dentro del desarrollo de la aplicación web, se debe establecer un nivel jerárquico para dar soluciones a los mismos.
- ✓ Sin ambigüedades: El requerimiento debe estar expresado en un lenguaje natural, libre de interpretaciones variadas, claro y formal.
- ✓ Verificable: Un requerimiento debe ser verificado dentro del contexto del negocio del usuario. Debe tener un fundamento sólido por el cual se lo va a cubrir dentro de la aplicación web.
- ✓ Consistente: No deben existir conflictos entre requerimientos, cada uno demanda un nivel de especificación más profundo cuando esto se produce.



- ✓ Modificable: Si a medida que el desarrollo de la aplicación web avanza se descubren incongruencias en un requisito, éste debe tener la facilidad de ser modificable sin alterar el proyecto como tal.
- ✓ Trazable: Un requerimiento correctamente definido debe poder ser rastreado desde su origen, durante el transcurso del desarrollo y posteriormente cuando se encuentre implementado.

La Ingeniería de Requerimientos (IR) comprende actividades que son críticas para el éxito de la Ingeniería Web. Los requerimientos incorrectos, incompletos o ambiguos pueden derivar en severas dificultades en el desarrollo, incluso pueden causar la cancelación de proyectos. La IR trata con los principios, métodos y herramientas para extraer, describir, validar y gestionar los requerimientos.[Kappel, 2003] En la Ingeniería Web la IR se orienta a desafíos especiales como stakeholders no disponibles, requerimientos volátiles y constantes, ambientes operacionales impredecibles, la inexperiencia con las tecnologías web, la importancia particular de los aspectos de calidad propios de la web como usabilidad o rendimiento. No obstante, cuando se adoptan métodos de la IR en la Ingeniería Web, muchos principios importantes deben permanecer presentes: el involucramiento de stakeholders importantes y la identificación iterativa de requerimientos.

Los requerimientos juegan un papel primordial en el desarrollo de aplicaciones web. Sin embargo, los requerimientos no son siempre descritos de manera apropiada y pueden ser especificados de una manera vana, ambigua e incorrecta. Las consecuencias típicas de los requerimientos pobres son la baja aceptación de los usuarios, fallas en la planificación o arquitecturas de software inadecuadas. [Torres Bosh, 2008].

La IR trata con los principios, métodos, técnicas y herramientas para identificar, describir, validar y gestionar los requerimientos en el desarrollo de un sistema. Actualmente, numerosos métodos de IR y herramientas están disponibles. Sin embargo, estos beneficios no siempre son aplicados por los desarrolladores y la IR es, en algunos casos, gestionada de manera errónea, particularmente en la ingeniería web. Aunque la complejidad de las



aplicaciones web actuales requiere un enfoque mayor, la madurez del proceso de IR resulta ser insuficiente.

La definición de requerimientos no es un problema nuevo. En 1976 en un artículo titulado *Software Requirements: Are They Really a Problem?*, Bell Thayer enfatizan que los requerimientos no aparecen automáticamente sino que son identificados con actividades específicas de ingeniería. A mediados de 1980, Boehm estudió el costo de los defectos en los requerimientos y determinó que una corrección de defectos no descubiertos resulta hasta 200 veces más costosa que la identificación y corrección desde una etapa temprana. La colección iterativa y el refinamiento de los requerimientos son las funciones más importantes de la ingeniería de software para un cliente [Kappel, 2003].

Hay un amplio consenso sobre la importancia de los requerimientos para el desarrollo de un sistema exitoso y a través de los años han emergido varios estándares, modelos, lenguajes de descripción y herramientas. A pesar de ello, la industria del software está aún enfrentándose a grandes dificultades en cuestión de requerimientos. [Torres Bosh, 2008].

2.3 Modelado de Aplicaciones Web:

Los modelos representan un punto de partida inicial para la implementación de una aplicación web, definiendo los aspectos dinámicos y estáticos de contenido, hipertexto y los niveles de presentación. Aunque el modelo de contenidos de una aplicación web apunta a la captura de información subyacente y una aplicación lógica es similar a su correspondiente modelo de una aplicación no web, la necesidad de considerar el hipertexto es particularmente importante para las aplicaciones web. Los modelos de hipertexto representan todos los tipos de navegación posibles basados en el contenido. [Kappel, 2003] El modelo de presentación mapea las estructuras de hipertexto hacia páginas y sus correspondientes links, ambos representan la interfaz gráfica de usuario. “La inclusión de información y su contexto, tales como, usuario, tiempo, localización,



dispositivos usados y la adaptación de las aplicaciones web, derivadas de su información, ha ganado un incremento en el interés por su modelado” [Kappel, 2003].

Esto es indudablemente una consecuencia de las aplicaciones web ubicuas que se han convertido increíblemente populares. Según como se menciona en *The Discipline of Systematic Development of Web Applications*, tenemos lo siguiente: [Kappel, 2003].

2.3.1 Modelos específicos en la Ingeniería Web

Las herramientas de modelado para las aplicaciones web no son relativamente nuevas, sin embargo, los métodos para modelar aplicaciones tradicionales no son lo suficientemente expresivas para las características de las aplicaciones web. Por ejemplo, los lenguajes de modelado tradicional (tales como UML), no proveen conceptos apropiados para la especificación de hipervínculos. Esta fue una de las razones por las que modelos especiales fueron desarrollados para las aplicaciones web hace unos pocos años, con ello se permitió introducir una aplicación web en las tres dimensiones: niveles, aspectos y fases. [Kappel, 2003].

2.3.1.1 Niveles

Para modelar aplicaciones web, el contenido de los documentos, tal como conocemos es una navegación de hipertexto no lineal, que debe ser tomada en cuenta. Esta es la razón por la que se discuten tres niveles en el modelado de aplicaciones web, como se muestra en la figura 3, en contraste con los dos niveles utilizados en métodos de modelado para las aplicaciones tradicionales. Los tres niveles son: contenido (información), hipertexto (estructura de contenidos en nodos y vínculos entre esos nodos) y la presentación (interfaz de usuario). La mayoría de los métodos usados para modelar aplicaciones web siguen esta separación en tres capas. [Torres Bosh, 2008].



Una separación clara de estos tres niveles permite la reutilización y ayuda a reducir la complejidad. Por ejemplo, podemos especificar un número de diferentes estructuras de hipertexto que se ajustarán a requerimientos específicos de diferentes grupos de usuarios y dispositivos de usuarios para un contenido dado. El objetivo primordial de un modelo de contenido es la definición explícita de la estructura de información; comparado con un esquema de base de datos, el modelado de datos elimina redundancias. Esto significa que la estructura de la información permanecerá intacta, incluso si la información como tal cambia constantemente. [Kappel, 2003].

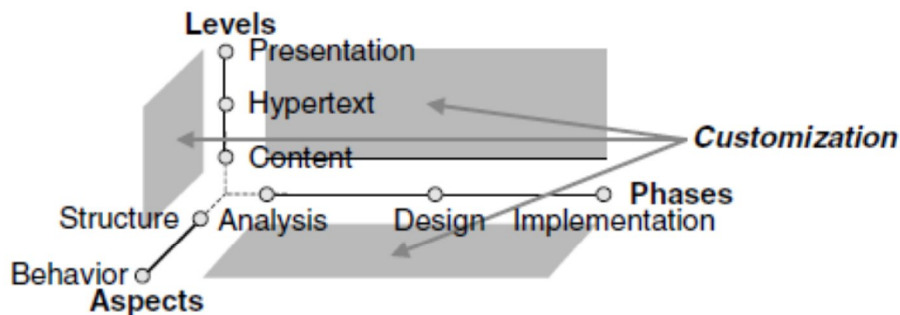


Figura 3: Dimensiones para la categorización de las Aplicación web⁴

Para diseñar una navegación eficiente, el contenido puede presentar redundancias en algunos nodos en el nivel de hipertexto. Para prevenir que los usuarios se pierdan mientras están navegando y para mantener bajo el stress cognitivo, el modelado de hipertexto debe basarse en patrones de navegación recurrentes. [Torres Bosh, 2008].

A su vez, cuando se modela el nivel de presentación, la atención se centra en una estructura uniforme de las páginas para facilitar ágilmente los efectos de reconocimiento de una aplicación web para sus usuarios.

⁴ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 62.



Aunque la apariencia visual de una aplicación web es importante, los aspectos estéticos no son el enfoque mayoritario del modelado.

2.3.1.2 Aspectos

Siguiendo con los principios orientados a objetos, la estructura y el comportamiento son modelados en cada uno de los tres niveles. La relevancia de éstos depende del tipo de aplicación web a ser implementada. Aquellas que poseen mayormente información estática disponible requieren un modelado de comportamiento menor comparadas con las aplicaciones web altamente interactivas, como por ejemplo, aplicaciones e-commerce, las cuales proveen motores de búsqueda, compra y otras funciones. Con respecto al mapeo de los diferentes niveles, es recomendable usar un formalismo de modelado uniforme para la estructura y el comportamiento, lo que permitirá derivar en una simple herramienta CASE. Naturalmente, este formalismo al modelar, tiene como frente las características específicas de cada uno de los tres niveles. [Kappel, 2003].

2.3.1.3 Fases.

No existe un consenso en lo escrito hasta la actualidad acerca de las ventajas de un modelado general para el desarrollo de aplicaciones web. En todo caso, la secuencia de pasos para modelar los niveles debería ser formulada por el encargado del proyecto. Dependiendo del tipo de aplicación web, básicamente se siguen ciertos pasos como iniciar con el modelado del contenido o de los aspectos de la presentación como tal. El desarrollo basado en modelos en la web, contradice las clásicas mejores prácticas en los proyectos web, como por ejemplo, ciclos de desarrollo cortos y el deseo de tener métodos ágiles. [Schwabe, 2001]

Los modelos además aseguran la sustentabilidad de soluciones, en contraste con las ideas del software de vida corta. Adicionalmente, la comunicación entre los desarrolladores de un equipo o mejor conocidas como cliente y desarrollador, se ve mejorada.



2.3.1.4 Customización

La inclusión del contexto de la información en el desarrollo de aplicaciones web juega un papel significativo. La customización considera el contexto, por ejemplo, preferencias de usuario, características de dispositivos o restricciones de ancho de banda, y permite adaptar la aplicación web de una manera correcta. Esto influye a todas las tres dimensiones del modelado web como son: el contenido, hipertexto y presentación, en lo relacionado a la estructura y comportamiento, y debería ser tomada en cuenta en todas las fases del desarrollo. No obstante, el manejo del contexto de la información se realiza como un modelado de dimensión independiente. [Kappel, 2003]

Dado que no existe un método de modelado completo que abarque estas tres dimensiones analizadas, en el resto de este capítulo se utiliza UML como notación con la inclusión de algunos nuevos conceptos provenientes de UML-based Web Engineering (UWE). Se sugiere usar UWE en complemento con UML.

2.3.2 Modelo de requerimientos

Varias técnicas pueden ser usadas para analizar, describir, evaluar y gestionar los requerimientos de una aplicación web. Los casos de uso son la técnica de modelado preferida para requerimientos funcionales debido a que pueden ser representados gráficamente. La funcionalidad general de una aplicación web es modelarse como un conjunto de casos de uso, lo cuales describen los requerimientos desde la perspectiva de los actores (personas u otros sistemas). [Kappel, 2003]

Adicionalmente, los casos de uso pueden ser complementados por diagramas de actividades de UML para detallar los requerimientos funcionales en mayor precisión. Una peculiaridad de los requerimientos de las aplicaciones web es la funcionalidad de navegación, que es lo que permite al usuario navegar a través de hipertexto y encontrar nodos. Todos los sistemas web tienen al menos un usuario humano, más conocido como anónimo.



Como ejemplo práctico se propone lo siguiente: Un sistema de revisión de papers para conferencias en línea, (llamado más adelante “el sistema de revisión”). Se identifican cuatro actores: usuarios del sistema de revisión, autores de los papers, miembros del comité de revisión y el sistema como tal. La figura 3 muestra un diagrama de casos de uso para el sistema de revisión el cual servirá como punto de partida para un modelado posterior.

Los casos de uso deben ser descritos en detalle. Se puede describir cada caso de uso en forma textual o mediante un diagrama de comportamiento, por ejemplo un diagrama de actividad. Éstos son principalmente usados cuando los casos de uso están basados en aplicaciones lógicas más complejas. [Kappel, 2003]

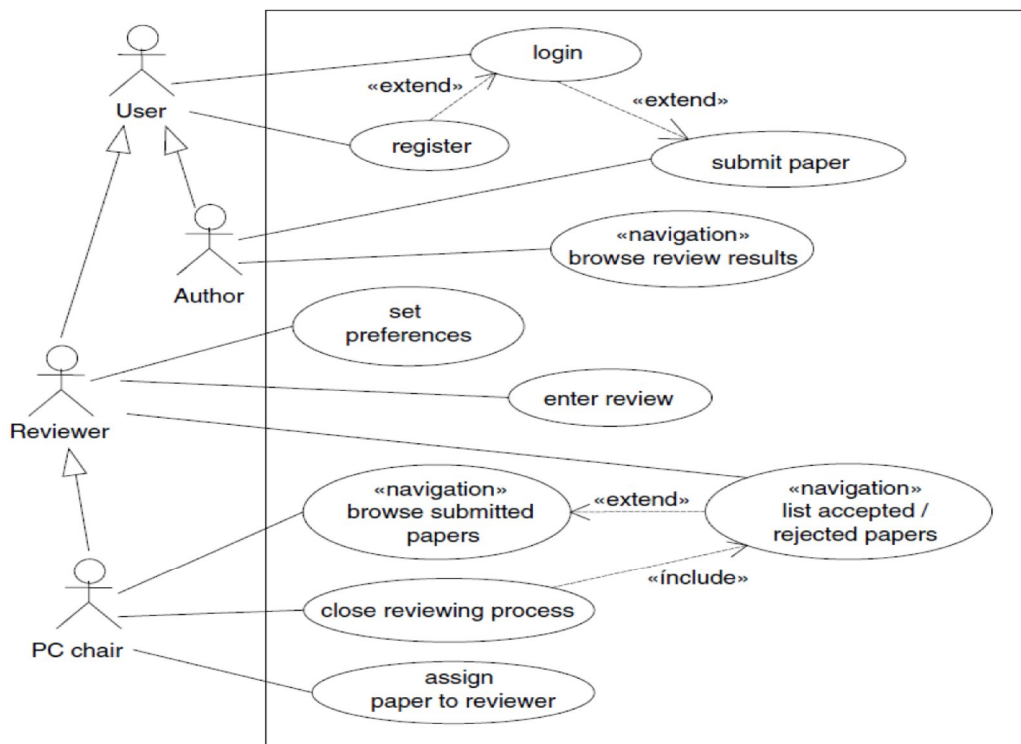


Figura 4: Diagrama de casos de uso del sistema de revisión⁵

⁵ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 65.

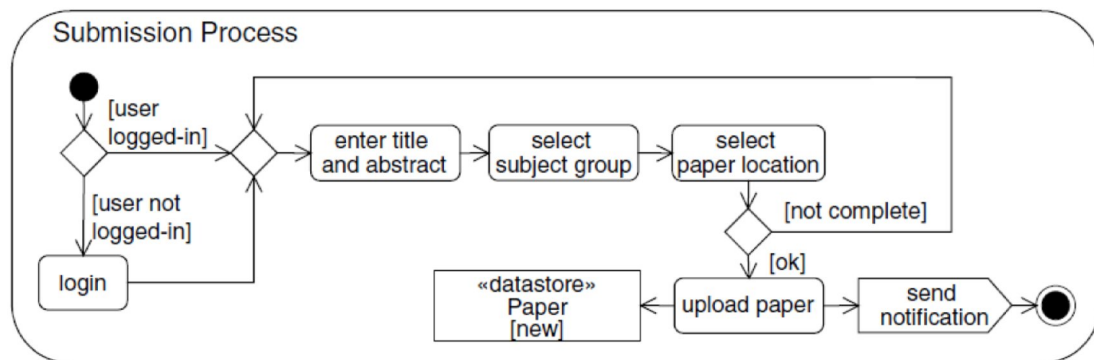


Figura 5: Ejemplo de diagrama de actividades para el proceso de envío y aceptación de papers⁶

2.3.3 Modelo de Contenidos

La información provista por una aplicación web es uno de los factores más importantes de dicha aplicación. Modelar el contenido basándose únicamente en los datos es suficiente para aplicaciones web estáticas. Las aplicaciones web complejas adicionalmente requieren un modelado de los aspectos de comportamiento. Esto significa que el modelado de contenido incluye la creación del modelo de dominio del problema con sus aspectos estáticos y dinámicos, además de características propias de las aplicaciones web tales como: [Mateu, 2004].

- Multimedia y documentos: Necesario para obtener todos los tipos de formatos multimedia diferentes durante el modelado, incluyendo las estructuras de información en las que se basa.
- Integración de datos existentes y software: Muchas aplicaciones web se construyen sobre repositorios de datos y componentes de software, los cuales no fueron creados por las aplicaciones web originalmente. El modelado de contenido tiene que satisfacer dos posibles objetivos primordiales, por ejemplo: debe cubrir los

⁶ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 65.



requerimientos de contenido de la aplicación web, lo más que pueda, e incluir estructuras de datos y componentes software.

El modelado de contenidos está orientado a transferir información y los requerimientos funcionales determinados por la IR hacia un modelo. Los caracteres de hipertexto y los requerimientos de su presentación no son considerados en esta etapa. Al modelar los contenidos se produce un esquema que comprende tanto los aspectos estructurales del contenido y dependiendo del tipo de aplicación web, los aspectos de comportamiento expresados como diagramas de estado e interacción.[Torres, Boch, 2003]

Como se mencionó al inicio, el modelo de contenidos se construye sobre los conceptos y métodos del modelado de datos o del modelado orientado a objetos, se trata de asegurar que la información existente esté libre de redundancias y sea reutilizable. La figura 6 muestra un diagrama de clases simplificado para el ejemplo del sistema de revisión.

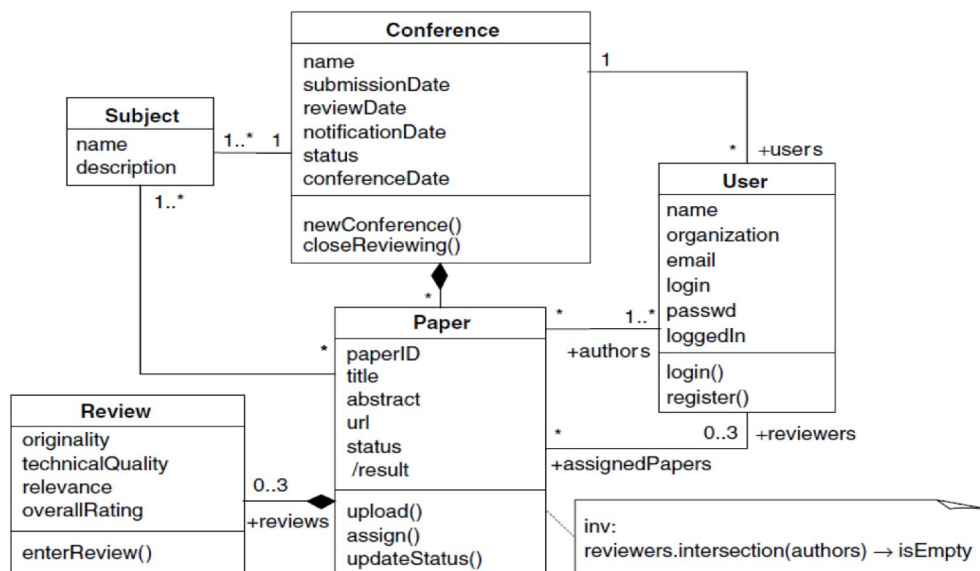


Figura 6: Diagrama de clases del sistema de revisión de papers.⁷

⁷ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 67.

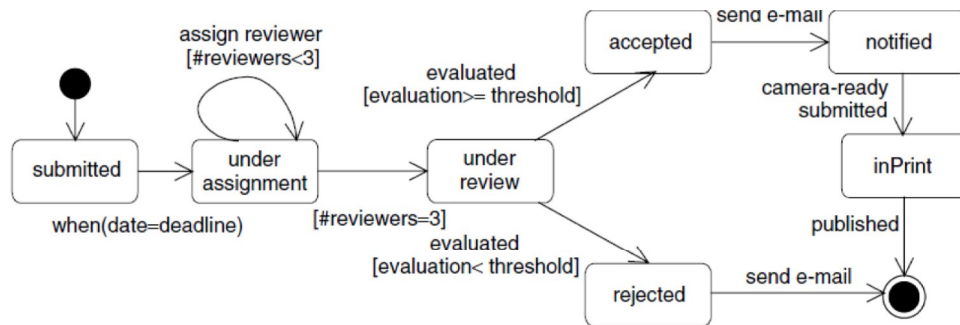


Figura 7: Diagrama de máquinas de estado para el sistema⁸

2.3.4 Modelado de Hipertexto

El objetivo del modelado de hipertexto o también conocido como modelado de navegación permite especificar la navegación a través de los contenidos de la aplicación web. El modelado de hipertexto genera dos resultados, el primero, el modelo de estructura de hipertexto con su definición, y segundo, refina el modelo de hipertexto en base a los elementos de acceso dando como resultado el modelo de acceso. El modelado de hipertexto se centra en los aspectos estructurales y en elementos de acceso. El comportamiento de navegación de una aplicación web no está normalmente representado porque provee muy poca información adicional para el desarrollador. [Kappel, 2003]

El modelado de estructuras de hipertexto está en conceptos de hipertexto como nodos (también llamados páginas o documentos) y los enlaces entre estos nodos. El punto de partida utilizado para la creación de un modelo estructurado de hipertexto es usualmente el modelo de contenido que abarca las clases y objetos que estarán disponibles como nodos en el hipertexto. Algunas veces el modelo de estructura de hipertexto es especificado como una vista en el modelo de contenido y es conocido también como la vista de navegación. Algunos métodos incluso definen

⁸ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 67.



reglas de transformación para los enlaces derivados basados en relaciones en el nivel de contenidos. Enlaces adicionales pueden añadirse por decisión en el diseño explícito. [Berzal, 2003]

Otros métodos modelan la estructura de hipertexto independientemente del modelo de contenido. Por ejemplo el OOHDM (Object-Oriented Hypermedia Design Method) ofrece una aproximación a escenarios de modelo donde la estructura de hipertexto puede construirse directamente desde los requerimientos de navegación identificados en estos escenarios. [Torres, Boch, 2008] En cualquier caso, podemos crear varios modelos de estructuras de hipertexto que definan vistas en el contenido; por ejemplo, si tomamos en cuenta los derechos de acceso de diferentes usuarios para la estructura de modelado de hipertexto, podemos obtener vistas personalizadas. En el ejemplo propuesto anteriormente del sistema de revisión, las vistas de hipertexto son requeridas por los siguientes actores: autor, comité de revisión y el sistema en sí. La figura 8 muestra el modelo de estructura de hipertexto para el ejemplo anteriormente señalado. [Kappel, 2003]

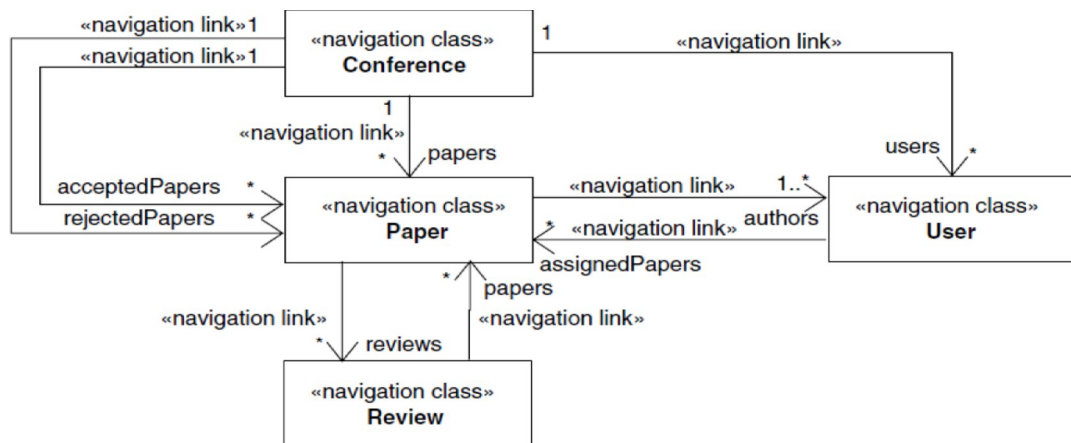


Figura 8: Modelo de hipertexto para el sistema de revisión⁹

⁹ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 69.



Los enlaces son modelados por asociaciones directas con el estereotipo de enlaces de navegación. Se han definido a lo largo del tiempo, varios tipos específicos de enlaces para refinar más aún la semántica del modelo de hipertexto; por ejemplo, el método HDM (Hypertext Design Model) especifica los siguientes tipos de enlaces: [Kappel, 2003]

- Enlaces estructurales: Conectan elementos del mismo nodo.
- Enlaces de perspectiva: Coloca varias vistas de un nodo en relación a los demás.
- Enlaces de aplicación: Coloca varias vistas de un nodo en relación a los demás dependiendo de la aplicación.
- Enlaces contextuales: Manejan información de contexto.
- Enlaces no contextuales: No tienen asociada información de contexto.

Por otro lado, WebML especifica adicionalmente los siguientes tipos de enlaces:

- Enlaces intra-páginas: Son usados cuando el origen y el destino de un enlace pertenecen a la misma página
- Enlaces entre-páginas: Son usados cuando el origen y el destino de un enlace están en diferentes páginas.

Basado en los requerimientos funcionales, el método de modelado UWE define los siguientes tipos de enlaces: [Brinck, 2002]

- Enlaces de navegación: Son usados para el desplazamiento entre nodos.
- Enlaces de proceso: Punto para regresar al nodo de inicio de un proceso.
- Enlaces externos: Punto a un nodo que no pertenece directamente a la aplicación.

El método de modelado OO-H (Object-Oriented Hypermedia) define cinco tipos de enlaces, tal como se muestran a continuación: [Torres, Boch, 2008]



- Enlaces internos: Punto de acceso a nodos que se encuentran dentro de las fronteras de un requerimiento de navegación dado.
- Enlaces transversales: Punto de acceso a nodos que comprenden otros requerimientos de navegación.
- Enlaces de requerimiento: Punto hacia una ruta inicial de navegación.
- Enlaces externos: Punto hacia nodos externos.
- Enlaces de servicio: Puntos con sus correspondientes enlaces de respuesta hacia servicios disponibles.

2.3.5 Modelado de presentación

Al igual que la ingeniería de software tradicional, el modelado de presentación comprende lo relacionado a la interfaz de usuario y lo referente a la estética de una aplicación web. En contraste con aplicaciones tradicionales, el elemento central de la presentación en las aplicaciones web es una página como unidad de visualización. [Montero, 2006]

El modelado de presentación está orientado a diseñar la estructura y comportamiento de la interfaz de usuario y asegurarse que la interacción con la aplicación web es simple e intuitiva además, se toman en cuenta las tareas de comunicación y representación. Se generan dos resultados: primero, un concepto de presentación uniforme por el modelado recurrente de los elementos en las páginas. Sería ideal mostrar composición de cada página y el diseño de los campos, textos, imágenes, formas, etc., que están incluidos, y segundo, adicionalmente a la estructura de las páginas el modelado de presentación describe los aspectos orientados al comportamiento de la interfaz de usuario. Debido a la gran variedad de opciones de navegación y el riesgo latente de perderse, se debe dar a los usuarios la guía apropiada y ayuda en el nivel de presentación.

Esto se puede lograr, por ejemplo, mostrando la ruta de navegación actual, o las páginas visitadas durante la sesión activa. No todos los métodos disponibles para modelar aplicaciones web soportan variadas tecnologías en cuanto a los conceptos de modelado de presentación, algunas manejan sus

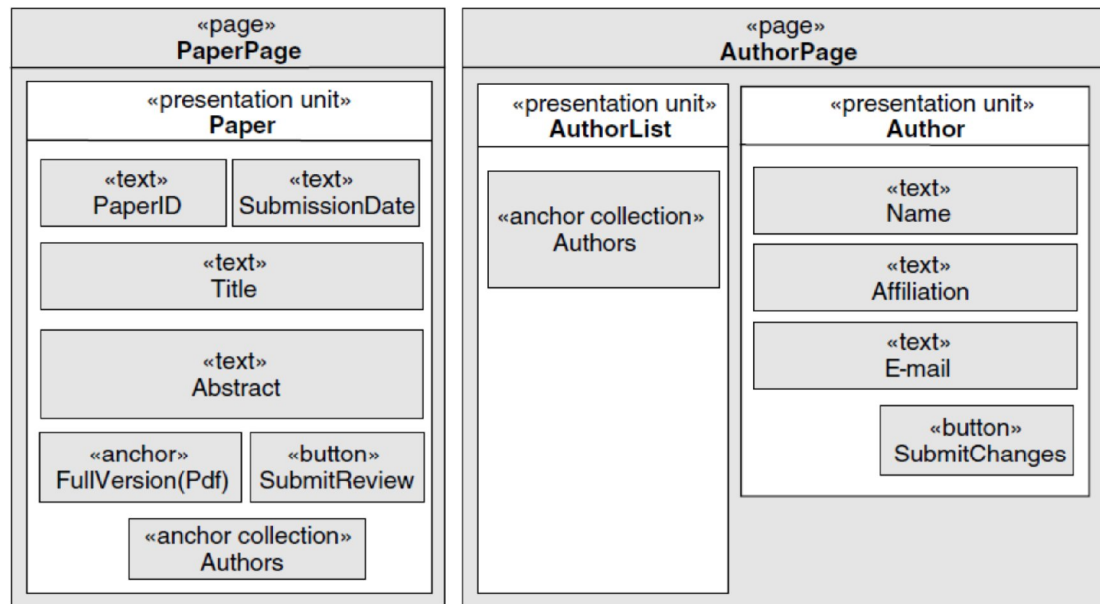


propios conceptos como hojas de estilo, lenguajes, etc. Otro factor importante para las aplicaciones web es el diseño del nivel gráfico de la interfaz de usuario. Es elaborada por un diseñador gráfico basado en algunas imágenes predefinidas y los prototipos inicialmente establecidos. Esto no lo soportan las técnicas de modelado actuales. [Montero, 2006]

Los elementos de modelado son descritos en 3 niveles jerárquicos: [Chile, 2000]

- Página de presentación: que describe una página presentada al usuario como unidad de visualización. Puede estar compuesta de diferentes unidades de presentación.
- Unidad de presentación: sirve para agrupar elementos de interfaz de usuario relacionados, representando un fragmento lógico de la página.
- Elemento de presentación: es el bloque de construcción básico del modelo de presentación. Los elementos de presentación representan un set de nodos de información y pueden incluir texto, imágenes, audio, etc.

Podemos visualizar la composición de las páginas de presentación en base a la representación por diagramas de clases UML anidadas conocidas como composición conforme lo muestra la figura 9:

Figura 9: Modelo de hipertexto para el sistema de revisión¹⁰

2.3.5 Modelado de Adaptación

Desde que las aplicaciones web ubicuas han empezado a ganar importancia, la consideración del contexto de la información y una apropiada adaptación de la aplicación es requerida tan pronto como sea posible en la fase de modelado. [Kappel, 2003]

La adaptación o Customization está orientada a representar explícitamente el contexto de la información y las adaptaciones derivadas de ésta. Dependiendo del método de modelado, los resultados no siempre son explícitos. En muchos casos, el modelado de adaptación es incorporado con los modelos de contenido, hipertexto y presentación. La adaptación debe estar enfocada desde el mantenimiento o reingeniería. Los modelos de adaptación consideran que el contexto de la información puede ser predicha al momento del modelado con lo que se asumen diferentes valores cuando la aplicación web está en ejecución. En contraste, la adaptación debido a

¹⁰ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 73.



cambios en el ambiente tecnológico u organizacional es parte de las actividades de mantenimiento o reingeniería.

La adaptación requiere examinar la situación de usabilidad de las aplicaciones web. Para ser capaces de personalizar una aplicación web, debemos modelar y gestionar las preferencias y características del usuario dentro de lo que se conoce como user profile. Por ejemplo: para adaptar una aplicación web en el campo de la computación móvil, se deben considerar, las características de los dispositivos, la localización de la información y el ancho de banda para la transmisión. Esta información es entonces representada dentro del modelo de contexto en forma de diagramas de clases. Durante la ejecución, el contexto puede cambiar o la aplicación es consumida en diferentes lugares. Esta situación es la razón principal por la que se debe adaptar la aplicación web. [Kappel, 2003]

La adaptación hacia un contexto puede ser modelada fundamentalmente de dos maneras diferentes: orientada hacia el resultado, creando varios modelos, lo que da origen a una adaptación estática; y la forma de adaptación dinámica que agrega reglas de transformación contexto dependientes para los modelos de contenido, hipertexto y presentación; esto describe las variantes a ser creadas durante la ejecución.

La mayoría de las metodologías de modelado existentes tratan con el modelado de adaptación, definiendo reglas o un filtro para cada punto en la aplicación web donde la adaptación se aplica.

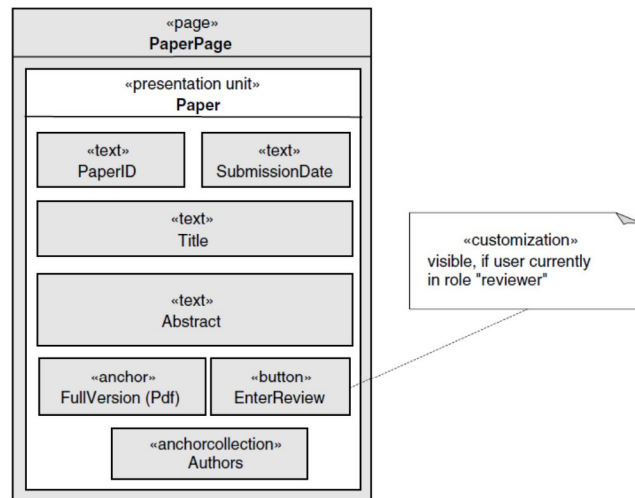


Figura 10: Adaptación dinámica de una página en el modelo de presentación¹¹

2.4 Arquitecturas de las Aplicaciones Web

A pesar del gran interés que ha surgido últimamente por la arquitectura del software, resulta un tanto complicado que los investigadores lleguen a un concepto unificado, de qué componentes deberían ser exactamente incluidos en la definición de arquitectura.

A continuación se presentan algunos conceptos para la arquitectura de software, con orientaciones a aplicaciones basadas en red según Fielding.

Abstracción en tiempo de ejecución: Es “una abstracción de los elementos que se encuentran ejecutándose en un sistema software durante su fase de operación. Un sistema puede estar compuesto de muchos niveles de abstracción y operación, cada uno de los cuales, tiene su propia arquitectura de software”. [Fielding, 2000].

¹¹ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 77.



Una aplicación web compleja puede tener numerosos niveles de abstracción con sus respectivas arquitecturas. Una arquitectura representa una abstracción del comportamiento de la aplicación web en un nivel dado, el cual puede tener otros niveles internos y crear un subsistema dentro de la aplicación general, lo que en conjunto define la abstracción del sistema completo. Un sistema también puede comprender varias fases de operación que necesariamente funcionan bajo una arquitectura propia.

Elementos: Una arquitectura de software está definida por la configuración de los elementos que componen la arquitectura, tales como componentes, conectores y los datos; restringidos en sus relaciones para lograr un conjunto deseado de propiedades en la arquitectura. [Fielding, 2000].

Los componentes o elementos de procesamiento son aquellos que efectúan cambios sobre los datos; los datos son los que contienen la información que será usada y transformada; los conectores por su parte son aquellos que sostienen la arquitectura y mantienen sus piezas enlazadas.

Componentes: Un componente es una unidad abstracta de instrucciones de software y estado interno del mismo, que facilita una transformación de datos (como cargar en memoria desde una unidad de almacenamiento externo) a través de su interfaz. Son los aspectos más fácilmente reconocibles dentro de la arquitectura, y cuyo comportamiento puede ser influyente o no para otro componente. Dicho de otra manera, un componente es definido por el servicio que brinda a los demás componentes y por su interfaz.

Conectores: Un conector es un mecanismo abstracto que permite la comunicación, cooperación y coordinación entre componentes sin la modificación de los datos. Un conector puede estar conformado por subsistemas que realizan, en caso de ser necesario, las transformaciones requeridas para que los datos puedan ir de una interfaz a otra.



Datos: Los datos son elementos de información que son enviados y recibidos por los componentes mediante los conectores.

Configuraciones: Una configuración es la estructura de las relaciones que conforman la arquitectura y mantiene unidos los componentes. [Fielding, 2000].

Propiedades: Las propiedades son inducidas por el conjunto de restricciones dentro de una arquitectura.

La calidad de una aplicación web se ve considerablemente influenciada por la arquitectura que lleva por detrás. Aspectos de arquitectura incompletos o mal manejados hacen difícil de cumplir los requerimientos de calidad e incluso no los permiten apreciar. Un mantenimiento insuficiente y baja disponibilidad de la aplicación web son causas de una arquitectura inapropiada. Adicionalmente a los aspectos técnicos como disponibilidad de los servidores, aplicaciones usadas en los servidores o la integración con otros sistemas, las arquitecturas de las aplicaciones web deben considerar el framework organizacional sobre el cual se encuentran trabajando. El uso de arquitecturas flexibles multicapa, la consideración de contenidos multimedia y la integración de repositorios de datos existentes son desafíos en el desarrollo de arquitecturas de aplicaciones web exitosas. [Fielding, 2000]

Cuando se desarrolla aplicaciones web debemos considerar un gran número de requerimientos y restricciones, que van desde requerimientos funcionales como órdenes de producto en línea, hasta requerimientos de calidad como rendimiento y disponibilidad para la integración de otros sistemas o de repositorios de datos que la aplicación web debe tener en cuenta. Las aplicaciones web no son normalmente desarrolladas de la nada, en lugar de ello se tiende a adaptar una infraestructura existente. Además, sólo de aspectos técnicos, se pueden identificar otros criterios como la factibilidad económica de una infraestructura técnica. La arquitectura debería diseñar una mejor manera de obtener estos requerimientos. [Fielding, 2000]



Existen criterios variados sobre la definición del término arquitectura, no obstante todos ellos centralizan su visión hacia las propiedades más importantes de una arquitectura: [Kappel, 2003]

- La Arquitectura describe la estructura: consiste en la descomposición de sus componentes principales, sus interfaces y relaciones. Describe aspectos estáticos y dinámicos.
- La Arquitectura es una forma de transición del análisis a la implementación: cuando se crea una aplicación web se trata de plasmar los requerimientos funcionales en componentes del sistema con sus relaciones e interfaces.
- La Arquitectura puede ser vista de diferentes maneras: se distinguen 4 formas diferentes: desde el ámbito conceptual, desde el ámbito de ejecución, el ámbito de los procesos y la implementación.
- La Arquitectura hace a un sistema entendible: permite gestionar de manera más eficiente la complejidad de los sistemas software.
- La arquitectura representa el framework de un sistema flexible.

Los requerimientos de software y con ellos su arquitectura son susceptibles de cambios. Constantes técnicas y organizacionales cambian durante y después del desarrollo de una aplicación. Esto se debe a requerimientos mal definidos al inicio del proceso de desarrollo o a cambios de los requerimientos una vez que el sistema se ha completado. Por esta razón, los sistemas de software son conocidos como “blancos móviles”. La figura 11 muestra los factores que influyen en el desarrollo de una arquitectura [Jacobson, 1999]:

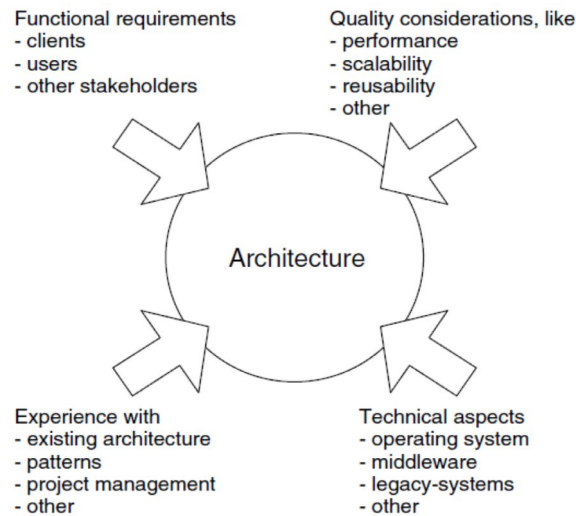


Figura 11: Factores que influyen en el desarrollo de una arquitectura¹²

La arquitectura de una aplicación es básicamente influenciada por los requerimientos funcionales y por las consideraciones de calidad. Fuera de estos requerimientos, las arquitecturas son también regidas por aspectos técnicos, el middleware, los sistemas a ser integrados, estándares usados, reglas de desarrollo y los aspectos de distribución. Por otra parte, la experiencia de los arquitectos de software juega un papel importante en la definición de la arquitectura. [Fielding, 2000] Por ser considerado un sistema como un blanco en movimiento, las arquitecturas son desarrolladas de manera iterativa, lo cual reduce el riesgo de requerimientos inseguros pero no es garantía de una buena arquitectura; dado que no es suficiente para resolver problemas de diseño específicos. Afortunadamente, los patrones de diseño han probado ser muy efectivos al brindar soporte en las decisiones de diseño.

¹² Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 88.



2.4.1 Componentes genéricos de la arquitectura de una Aplicación Web

La figura 12 muestra los componentes básicos de una arquitectura y sus relaciones. La comunicación entre estos componentes está generalmente basada en el principio de petición y respuesta. A continuación se describen brevemente algunos de estos componentes: [Kappel, 2003]

- ✓ Cliente: Generalmente un navegador controlado por un usuario para operar la aplicación web. Puede ser mejorado por plugins y applets.
- ✓ Firewall: Software, hardware o la combinación de estos dos para regular la comunicación entre redes inseguras y redes seguras mediante reglas de acceso.
- ✓ Proxy: Usualmente utilizado como sistema de almacenamiento temporal para las páginas web. Sin embargo, un proxy puede asumir otras funcionalidades.
- ✓ Servidor Web: Software, hardware o la combinación de estos dos, que soportan varios protocolos web para procesar las peticiones de los clientes.
- ✓ Servidor de Base de Datos: Mantiene los datos en una forma estructurada para el uso de la organización.
- ✓ Servidor Media: Usado para volúmenes de contenido no estructurado.
- ✓ Servidor de Gestión de Contenidos: Similar a un servidor de bases de datos, los contenidos almacenados generalmente provienen en forma de datos semi estructurados como ficheros XML.
- ✓ Servidor de Aplicación: Mantiene la funcionalidad requerida por algunas aplicaciones.
- ✓ Aplicaciones Integradas: Son aquellos sistemas antiguos que deberían ser integrados como un componente interno o externo.

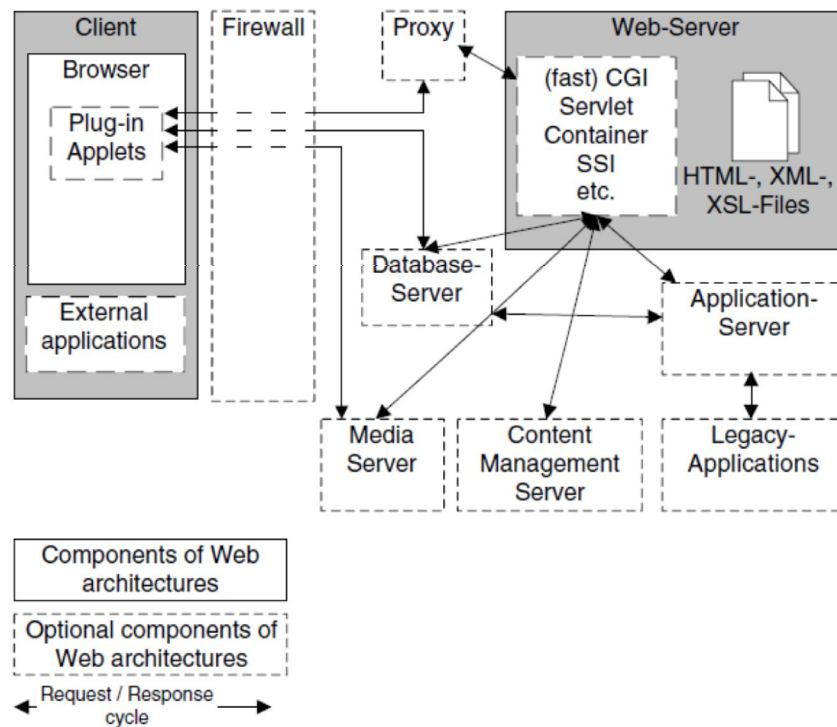


Figura 12: Componentes genéricos de una arquitectura para una Aplicación Web¹³

2.4.2 Arquitecturas por capas

2.4.2.1 Arquitecturas de dos capas

También conocidas como arquitectura cliente/servidor, utilizan un servidor web para brindar el servicio a los clientes. La arquitectura de dos capas puede adoptar diferentes formas dentro del ambiente de las aplicaciones web. La petición de un cliente está destinada directamente hacia páginas HTML estáticas sin necesidad de un procesamiento lógico en la capa de servicio, o puede acceder a una base de datos usando la aplicación lógica en el servidor web.

¹³ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 93.

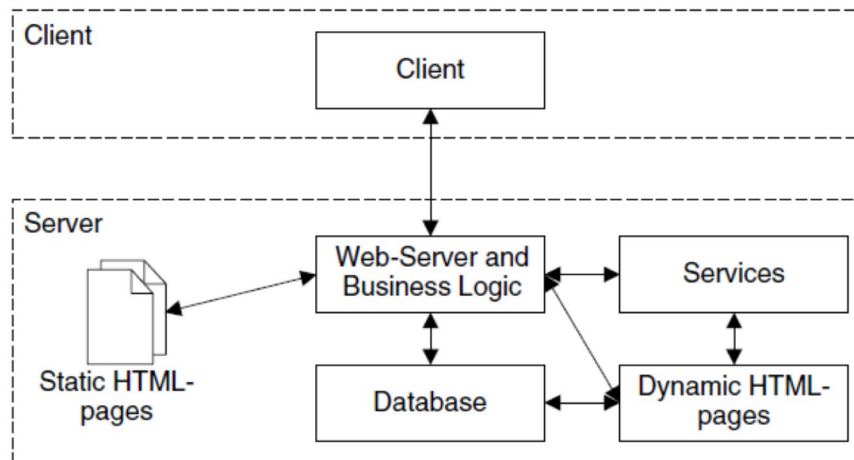


Figura 13: Arquitectura de dos capas para una Aplicación Web¹⁴

Las páginas HTML dinámicas incluyen scripts en código HTML que en algunas ocasiones pueden utilizar servicios cuando se requiere generar una respuesta. Este tipo de arquitectura es particularmente eficaz para aplicaciones web simples, en cambio una arquitectura multicapa es requerida para aplicaciones que son accedidas por un gran número de clientes concurrentes, aquellas que manejan procesos de negocios complejos, o que tienen que establecer comunicaciones para la integración con otros sistemas. [Kappel, 2003]

2.4.2.2 Arquitecturas de N Capas

Las arquitecturas de N capas nos permiten organizar una aplicación web en un número arbitrario de capas, como lo muestra la figura 14. Típicamente consisten en tres capas, la capa de datos, la capa de negocio y la capa de presentación. Mecanismos de seguridad como firewalls, proxys pueden ser integrados en el flujo de petición y respuesta de acuerdo a la demanda.

Las arquitecturas de 2 y N capas difieren principalmente en como embeben los servicios dentro del componente de aplicaciones del servidor. Servicios

¹⁴ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 94.



como la adaptación o el flujo de trabajo están sujetos al contexto del servidor de aplicaciones de tal manera que están disponibles para todas las aplicaciones web.

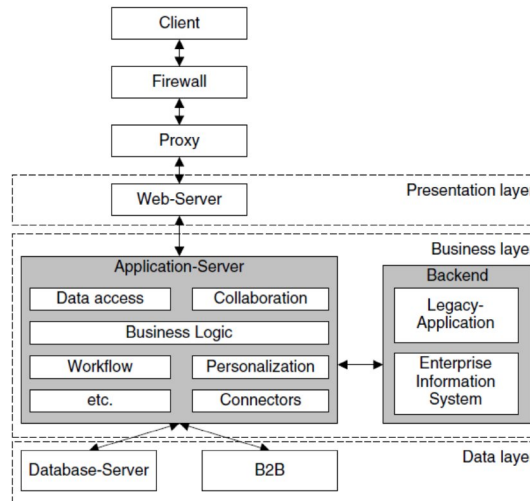


Figura 14: Arquitectura multicapa para una Aplicación Web¹⁵

Los servicios son embebidos en el servidor de aplicaciones con una interfaz definida y esta misma puede gestionar los servicios. Una ventaja adicional radica en que las aplicaciones web tienen mecanismos de balanceo de carga para el servidor y conectores, que pueden ser usados para integrar sistemas externos.

2.5 Pruebas en Aplicaciones Web

Muchos métodos y técnicas para probar las aplicaciones web han sido creadas y se encuentran actualmente disponibles. Sin embargo, no pueden ser directamente aplicadas sobre el software web. Debe ser replanteado y adaptado a estas condiciones propias de la web. El efectuar pruebas sobre aplicaciones web va más allá que el tradicional testeado del software tradicional. El proceso de ejecución de pruebas está destinado a evaluar la calidad de un producto y a mejorarlo, identificando los defectos y problemas

¹⁵ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 95.



que pudiera tener. Cada requerimiento, los objetivos planteados, las expectativas de los diferentes stakeholders; conforman la base para la elaboración del plan de pruebas. Cuando se realizan las pruebas, si un valor tiene una variación con respecto a los que son esperados, se considera como un error. [Kappel, 2003]

Los usuarios no solamente esperan sea estable sino una que sea fácil de usar, se adapte a nuevas condiciones, que sea confiable, compatible con otras aplicaciones. A pesar de ello, el proceso de prueba se realiza –sobre todo en proyectos web-, de manera limitada debido a la falta de recursos y los períodos cortos de desarrollo a los que está sometida la aplicación web.

De acuerdo a las distintas fases en el proceso de desarrollo de las que se puede someter a pruebas los resultados, se identifican niveles de pruebas que se detallan a continuación en base a The Discipline of Systematic Development of Web Applications:

- ✓ Pruebas de unidad: son efectuadas por el propio desarrollador durante la elaboración de la clase o método.
- ✓ Pruebas de integración: evalúan la interacción entre componentes distintos y validados por separado. Son efectuadas por el desarrollador, el tester o incluso ambos.
- ✓ Pruebas del sistema: comprende la evaluación completa de la aplicación web y es efectuada por un equipo especializado de pruebas.
- ✓ Pruebas de aceptación: en una prueba o conjunto de pruebas que se efectúan con la ayuda de personas que no están involucradas directamente con el desarrollo del proyecto web en un ambiente cercano al de producción, con datos y condiciones reales.
- ✓ Pruebas Beta: son versiones beta del producto final que son entregadas a un número limitado de personas con el objetivo de obtener una retroalimentación de las posibles falencias que se pudieran detectar en el trabajo diario. [Kappel, 2003]



A continuación se detallan los principales aspectos que deben someterse a prueba en una aplicación web, basados en sus características promedio. [Murray, 1999]

- ✓ Errores en el contenido: siendo la parte más importante de toda aplicación web, el contenido debe ser meticulosamente revisado de una manera periódica con la finalidad de que se encuentre actualizada toda la información disponible a los usuarios. La explotación de métodos tanto manuales como automáticos de actualización de información, debe estar acorde con la aplicación desarrollada y su complejidad.
- ✓ Enlaces rotos: se debe asegurar que la estructura de todo el hipertexto contenido en los documentos esté correctamente enlazada y que al momento que el usuario solicite uno de esos enlaces, la aplicación se redireccione hacia lo que se busca.
- ✓ Dispositivos potenciales: se refiere a los diferentes equipos en donde la aplicación web debe funcionar. No todos tiene la capacidad de representar lo que el software puede ofrecer, a menos que se realicen configuraciones por separado propias del dispositivo que a la final complican la relación del usuario con la aplicación.
- ✓ Multiculturalidad: comprende la capacidad de la aplicación para adaptarse a nuevos tipos de usuarios por su ubicación geográfica. El plan de pruebas debe contemplar este aspecto importante dentro de una aplicación web como medida de usabilidad. Se debe considerar un lenguaje universal para los usuarios en la medida de lo posible.
- ✓ Integración de componentes: la mayoría de las aplicaciones web modernas están basadas en componentes que son reutilizables y por lo tanto no necesariamente construidas por el mismo ente. Por ello, cada componente que será utilizado debe ser puesto a prueba ya que la calidad total de la aplicación dependerá de la calidad individual de sus componentes. [Kappel, 2003]



2.6 Desarrollo de Aplicaciones Web

Dentro del proceso de desarrollo de aplicaciones web existen dos términos con significados parecidos, dado que ambos, tanto los modelos de procesos como los métodos, describen como un proyecto de desarrollo de software avanza. Sin embargo, mientras que un modelo de procesos los describe de una manera general, los métodos lo hacen en una forma detallada. La mejor forma de ver esta diferencia es indagando sobre sus características comunes, dado que ambas formas definen como alcanzar objetivos específicos, y cómo éstos son influenciados por factores externos, lo que se traduce en dependencias entre fases y condiciones externas. Esto se ilustra en la figura 15:

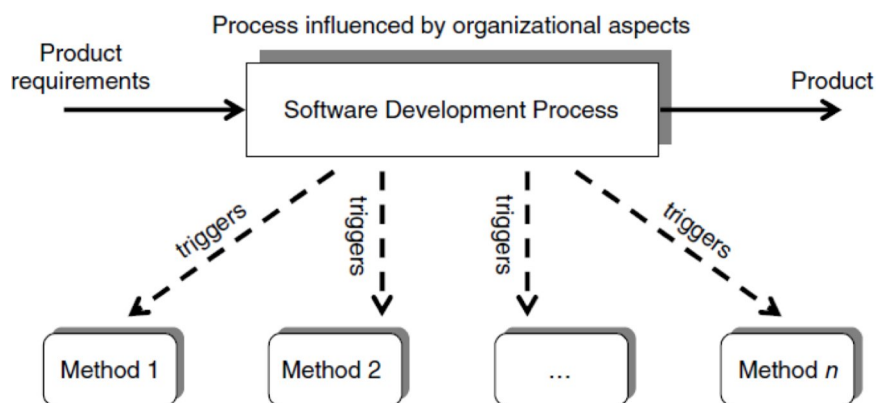


Figura 15: Procesos vs. Métodos en el desarrollo de un proyecto software¹⁶

Una propiedad importante de los procesos actuales de desarrollo de software es que son iterativos. La idea básicamente es desarrollar un sistema de software que se vaya incrementando desde un resultado inicial, luego del análisis, hasta el producto final en varias iteraciones. Una iteración es un grupo de actividades que dan origen a una liberación. Esto le permite al desarrollador tomar ventaja de lo que se fue aprendiendo

¹⁶ Web Engineering.- The Discipline of Systematic Development of Web Applications, 2003. Página 220.



durante el desarrollo de versiones previas del sistema. [Torres, Boch, 2008] El desarrollo de una aplicación web de forma iterativa, demanda un continuo desarrollo y revisiones del producto, lo que significa que la misma metodología es puesta a prueba varias veces. De esta manera, el proceso completo es dividido en una secuencia de iteraciones, como lo indica la figura 16.

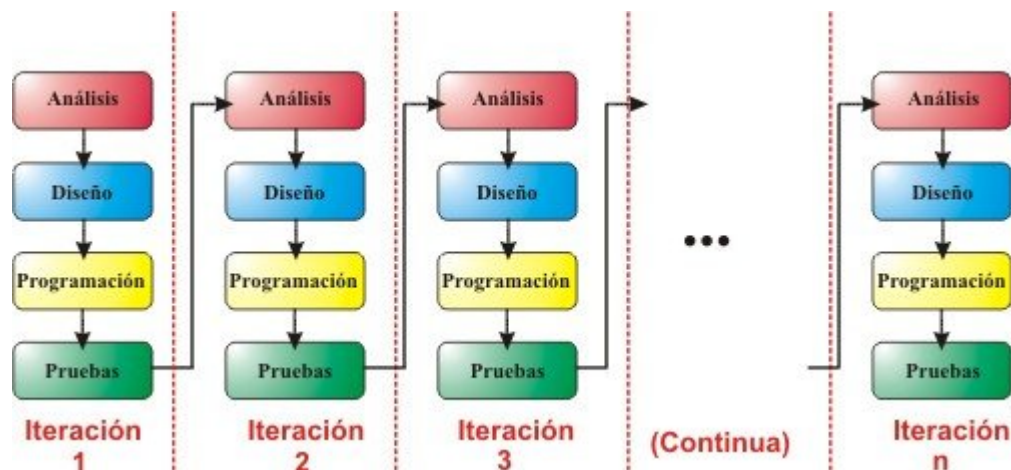


Figura 16: Ilustración del desarrollo iterativo de software

Se recomienda que un desarrollo iterativo se aplique cuando un equipo que trabajó junto, en anteriores ocasiones, sea asignado a un nuevo dominio de aplicación. El conjunto de iteraciones sucesivas le permite al equipo familiarizarse mutuamente, de manera gradual, con los requerimientos que no se conocen del nuevo dominio de la aplicación y para aprender de sus experiencias [Torres Bosh, 2008]. Vale recalcar que los procesos iterativos son populares porque incrementan el potencial de alcanzar los objetivos planteados de los clientes, quienes en ocasiones no saben definir correctamente lo que necesitan en su negocio.

Un proceso alternativo a las iteraciones es dividir el proceso en fases. Pero estas fases son erróneamente llamadas etapas de: definición de requerimientos, análisis, diseño, implementación, pruebas y mantenimiento, lo que corresponde al modelo tradicional de cascada, donde una actividad o fase tiene su predecesora. El problema con este modelo en cascada es que la gestión de riesgos se pospone para las etapas finales.



Cuando un riesgo potencial ocurre, el costo para depurarlo o eliminarlo se incrementa a medida que el proyecto avanza. Una manera óptima de resolver este inconveniente es la demarcación de hitos como puntos de sincronización dentro del sistema, para controlar el avance en el cumplimiento de los objetivos planteados.

Los conocidos procesos de desarrollo de software pueden ser agrupados dentro de dos categorías: procesos ágiles y procesos formales. Tanto los ágiles como los formales se refieren al grado o nivel de formalización de los procesos. Actualmente, sería correcto hablar de métodos ágiles y formales porque la creación de la documentación y los modelos, en forma específica, son propios de los métodos. Los procesos formales son usados especialmente cuando equipos de desarrollo grandes trabajan en aplicaciones que demandan altas prestaciones de calidad. Por otro lado, los procesos no formales son apropiados para aplicaciones pequeñas acordes con un equipo de trabajo pequeño. [Torres, Boch, 2003]

2.6.1 Gestión de requerimientos en el desarrollo de una Aplicación Web.

En este apartado se describirán las cinco actividades más importantes para el proceso de gestión de requerimientos para aplicaciones web. Estas actividades son la razón fundamental del por qué los modelos del proceso de desarrollo de aplicaciones de software tradicionales, tienen que ser adaptados a las necesidades del desarrollo de aplicaciones web, e incluso ser replanteadas en su totalidad, tal como lo mencionan Engels, Lohnman y Wagner:

2.6.1.1 Manejo de ciclos de desarrollo cortos

Un factor encontrado en varios estudios y experimentos realizados es que el desarrollo de aplicaciones web es extremadamente corto, que no excede de seis meses normalmente y su duración promedio es de menos de tres meses. Estos ciclos de desarrollo cortos son tan comunes para las WebApp que pueden ser asumidos como el primer requerimiento en el proceso de su



desarrollo. La competencia es extremadamente dura en el mundo de la Web, es un mercado internacional con un tamaño que no puede ser razonablemente estimado. Además, es necesario y demanda con urgencia tener sitios web más competentes en el mercado para asegurar el intercambio de información. [Torres, Boch, 2003]

No se puede esperar que los clientes sean leales a una misma aplicación web cuando la competencia está a un click de distancia. Se debe recordar que muchas aplicaciones web son instrumentos de marketing. Estos ciclos cortos dejan menos libertad para un proceso de desarrollo sistemático.

2.6.1.2 Gestión de Cambios en los requerimientos

Los requerimientos de una aplicación web emergen, la mayoría de las veces, únicamente durante su desarrollo. Un desarrollador en ocasiones tiene que tratar con campos desconocidos del negocio y sus requerimientos cambian dramáticamente conforme éste va adquiriendo experiencia y entendiendo el negocio en el transcurso del proyecto. Los nuevos requerimientos que resulten de esta situación deben ser incorporados rápidamente, para prevenir que tengan que ser aceptados requerimientos ambiguos o incompletos. [Kappel, 2003]

Es necesario un análisis detallado de un grupo de usuarios de una Aplicación web para definir requerimientos estables, pero esto a su vez es casi imposible por la heterogeneidad de los usuarios. Debido a que el software puede distribuirse inmediatamente tiende a convertirse, en práctica común, inferir sobre los requerimientos actuales, producto de las respuestas de los clientes hacia las partes publicadas de la aplicación.

La necesidad de experimentar con un producto software, la evolución resultante de los requerimientos y la fuerte presión de la competitividad, probablemente nunca desaparezcan y como consecuencia directa de estos requerimientos, nace la integración del cliente con el equipo de desarrollo para ser informados en detalle acerca del avance. Un escenario óptimo se tendría si el cliente y el equipo de desarrollo están en el mismo lugar, de tal



manera que ambos participen continuamente en la construcción de la aplicación web, así como en la actualización de los contenidos. [Kappel, 2003]

2.6.1.3 Liberaciones con fechas límite corregidas y contenidos flexibles

Una consecuencia indirecta del anterior requerimiento es la necesidad de utilizar un tipo especial de prototipado en el proceso de desarrollo. Específicamente, liberaciones descartables son desarrolladas para detallar y validar los requerimientos del cliente. Algunos autores han encontrado que esta forma “explorativa” de definición de la especificación de un sistema, se ha convertido en una práctica común. El cliente describe la funcionalidad básica que necesita y los prototipos son rápidamente elaborados con fines demostrativos.

Esto significa que el prototipado es el punto de enlace en la comunicación con el cliente. Aquí se producen liberaciones rápidas de aplicaciones web que pueden ser rápidamente cambiadas por nuevas versiones; los intervalos entre liberaciones son relativamente cortos (de 2 a 15 días) [Ramesh, 2002]. La planificación para la secuencia de liberaciones y su evaluación es aún más importante que la planificación de requerimientos para una liberación. Las características sencillas de la aplicación pueden ser completadas y aplazadas a una siguiente liberación, lo que normalmente no representa un problema mientras los intervalos sean cortos y no demanden costos adicionales representativos. [Kappel, 2003]

Otro punto importante es que las aplicaciones web raramente son críticas. Requerimientos de alta calidad obligan a una planificación más fuerte de las liberaciones con respecto a las fases de pruebas, etc. El soporte no es un interés particular si el software va a ser rápidamente reemplazado por nuevas versiones. Este tipo de liberaciones por fases o planificadas apropiadamente tiene otro beneficio adicional, uno de los mayores problemas en el desarrollo de Aplicación web es la falta de métricas apropiadas para cuantificar, comparar y estimar el desarrollo en tiempo y costo.



2.6.1.4 Desarrollo paralelo de diferentes liberaciones

Una agresiva competitividad conducen a ciclos de liberación cada vez más cortos. Bajo esta presión, únicamente el desarrollo en paralelo de proyectos puede favorecer la distribución de una aplicación completa a tiempo o sus liberaciones de acuerdo al cronograma establecido. Esto significa que actividades metodológicas en las fases de diseño, implementación y aseguramiento de la calidad, son trabajadas a la par en diferentes liberaciones con miembros del equipo de trabajo, desenvolviéndose en tareas similares, lo que implica que la comunicación tiende a ser el factor primordial en el equipo de trabajo. [Kappel, 2003]

2.6.1.5 Reutilización e Integración

Una de las consecuencias de las enormes presiones en cuanto al tiempo en el desarrollo de una aplicación web es que el equipo de trabajo debe tratar de reutilizar la mayoría de componente como sea posible. [Ramesh et al. 2002], esto se complica cuando la interoperabilidad e integración se realizan con sistemas o componentes que han sido desarrollados externamente o se han adquirido a terceros.

Por esta razón, el proceso de desarrollo de una aplicación web no puede ser llevado de manera separada de los procesos de desarrollo de otra aplicación web incluso de la misma organización. Si un componente reutilizable es desarrollado en un proyecto, entonces debe ser gestionado en coordinación con otros proyectos que necesitan el componente en cuestión. [Kappel, 2003]

Además, la necesidad de integrar aplicaciones web con aplicaciones existentes, u otras que aún se encuentran bajo construcción, se incrementa conforme va creciendo la inclusión de la aplicación web en los procesos de negocio del cliente. Este tipo de integración también requiere que los procesos sean coordinados. Bajo este escenario, las necesidades de desarrollo de Aplicación web y el desarrollo convencional de software confluyen.



CAPÍTULO III

3 Metodologías de desarrollo de Aplicaciones Web

3.1 UWAT+

El framework UWA (Ubiquitous Web Applications) es el resultado de un esfuerzo en conjunto de un grupo de investigadores de diferentes países que conformaron el UWA Consortium. Como su nombre lo indica, es un framework que provee una metodología para la construcción de aplicaciones web ubicuas. El aspecto ubicuo brinda flexibilidad respecto al objetivo y el contexto en el que la aplicación web será usada.

El objetivo del proyecto UWA fue la producción de un ambiente que permita cubrir el proceso de desarrollo de aplicaciones web ubicuas partiendo de los requerimientos hacia la fase de diseño. Este consorcio propone el uso de una adaptada y extendida notación de UML a través del proceso de desarrollo. [Torres, Boch, 2008]

3.1.1 Proceso de Desarrollo

UWA define seis modelos para llevar a cabo la especificación de una aplicación web:

- ✓ Modelo de requerimientos.
- ✓ Modelo Conceptual (especifica el contenido de la aplicación así como las formas de acceder).
- ✓ Modelo de organización.
- ✓ Modelo de ejecución.
- ✓ Modelo de navegación.
- ✓ Modelo de presentación.



La figura 17 muestra el proceso de desarrollo de una aplicación web siguiendo el modelo UWAT+.

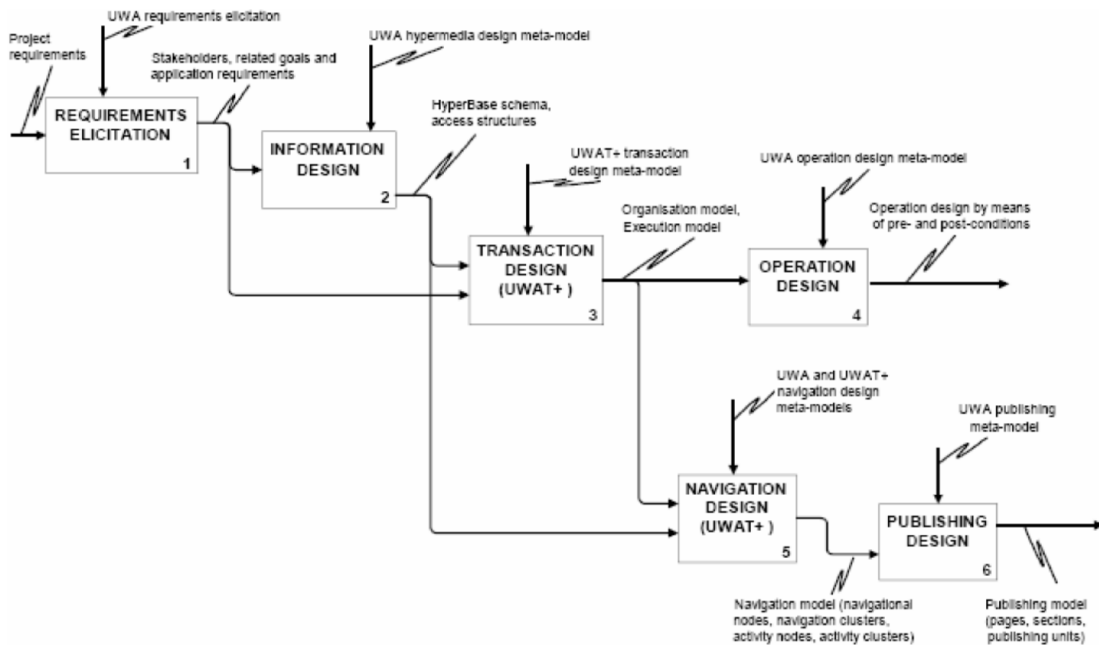


Figura 17: Esquema del proceso de desarrollo UWAT+.¹⁷

3.2 WebML: Web Modeling Language

WebML es una notación específica para sitios web complejos a nivel conceptual. [Torres, Boch, 2008] Originalmente fue concebido para dar soporte al diseño e implementación de aplicaciones web enfocadas a datos, sin embargo, como una respuesta a los nuevos desafíos que han aparecido en la web, WebML ha evolucionado continuamente para satisfacer algunos de estos retos. Como resultado de esta evolución, WebML se ha extendido a una nueva generación de aplicaciones web basadas en servicios, en flujos de trabajo, web semántica, aplicaciones enriquecidas. Por otra parte, el lenguaje de modelado WebML y su método de desarrollo están

¹⁷ A Web Engineering Approach for the Development of Business-Process Driven Web Applications, 2008, Página 92.



implementados en una herramienta web comercial llamada WebRatio Site Development Studio que automatiza la generación de la base de datos relacional y la aplicación de plantillas. Esta herramienta permite un prototipado rápido que puede ser usado para la implementación, mantenimiento y evolución de aplicaciones web modeladas. [Torres, Boch, 2008]

WebML intenta un proceso de desarrollo iterativo e incremental, como lo muestra la figura 18; en donde varias tareas son repetidas y redefinidas hasta que los resultados se asemejen a los requerimientos del negocio. Este modelo permite continuas pruebas, modificaciones o extensiones de cada versión del sistema. La figura también muestra que en este modelo, la fase de diseño conceptual ha sido extendida con dos nuevas tareas, (process design, y process distribution).

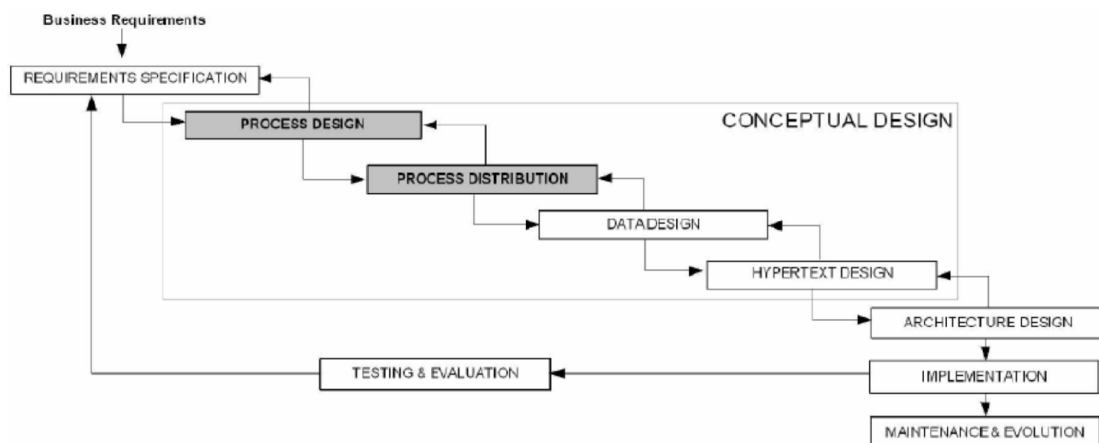


Figura 18: Proceso de desarrollo WebML¹⁸

¹⁸ A Web Engineering Approach for the Development of Business-Process Driven Web Applications, 2008, Página 96.



3.3 OOHDM: Object-Oriented Hypermedia Design Method

El conjunto de modelos propuestos por OOHDM está basado en conceptos orientados a objetos, usando conceptos como estructura de objetos, comportamiento y mecanismos de abstracción como la agregación y la generalización/especialización. Sin embargo, a pesar de ello el diseño resultante puede ser implementado por encima de cualquier plataforma convencional. El método OOHDM ha evolucionado desde sus inicios y ha sido adaptado para tratar con las necesidades de cada momento. La única herramienta disponible relacionada con OOHDM es HyperDE pero su enfoque es para SHDM (Semantic Hypermedia Design Method) que es una extensión del original para la construcción de aplicaciones semánticas. [Kappel, 2003]

3.3.1 Proceso de desarrollo

OOHDM divide el proceso de desarrollo en cuatro pasos que corresponden a:

- ✓ Modelo conceptual.
- ✓ Modelo de navegación.
- ✓ Modelo de interfaz abstracta.
- ✓ Implementación.

La primera etapa o modelo conceptual es descrita usando un modelo de diseño hipermedia orientado a objetos. Adicionalmente, un diagrama de actividades UML es incorporado para definir el flujo de control entre los procesos. En la segunda fase, la hipermedia es descrita por medio de estructuras de navegación, las cuales se construyen basándose en las responsabilidades asignadas a los diferentes tipos de usuarios. En el modelo de interfaz abstracta, se define como los usuarios recibirán los objetos a través de la interfaz. Finalmente, durante la etapa de implementación, los modelos de interfaz abstracta y de navegación son transformados en un ambiente concreto. [Torres, Boch, 2008]

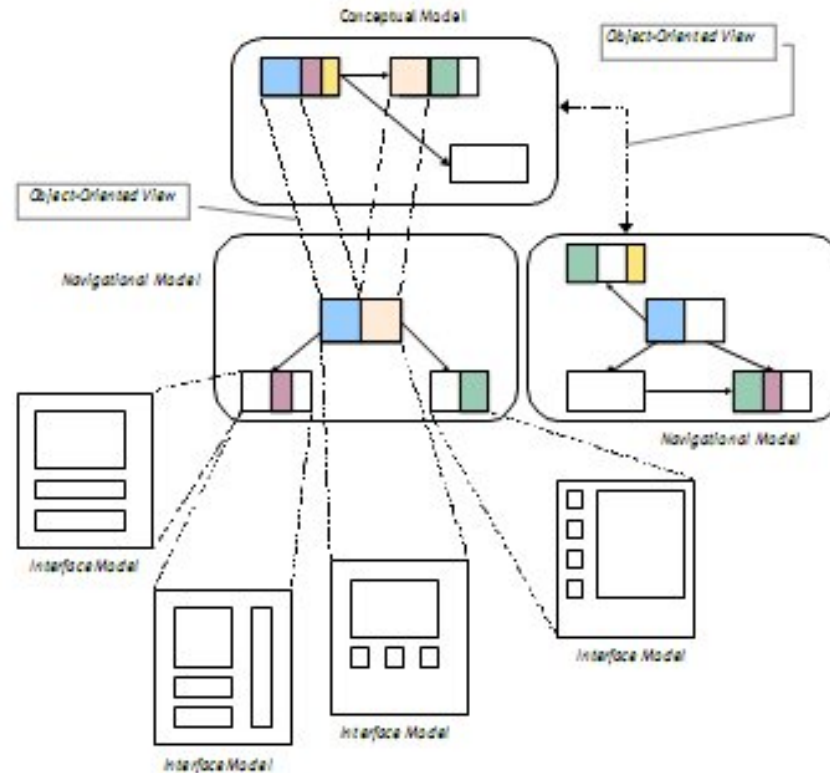


Figura 19: Proceso de desarrollo OOADM¹⁹

3.4 UWE: UML based-web Engineering

Basado en UML, utiliza la notación estándar de éste tanto como sea posible, y define un perfil para especificar las peculiaridades propias de las aplicaciones web. El mayor beneficio es que al utilizar notación UML, se puede utilizar un sin número de herramientas para producir modelos UWE. [Kappel, 2003]

¹⁹ An Object Oriented Approach to Web-Based Application Design, Daniel Schwabe, Gustavo Rossi; Departamento de Informática. PUC-RIO, Brazil.



3.4.1 Proceso de desarrollo

UWE propone tratar con el proceso de modelado de aplicaciones web con un conjunto de herramientas para especificar cuatro diferentes aspectos. Inicialmente para trabajar con los requerimientos se sugieren los casos de uso y los diagramas de actividad UWE, se especifican los usuarios y las funcionalidades que el nuevo sistema debe incluir. El modelo de contenido otorga una especificación visual del dominio de la aplicación que se complementa con el modelo de contexto. Ambos son gráficamente representados por diagramas de casos UML. La estructura de navegación permite definir los nodos, enlaces, menús y accesos primitivos como índices y guías a través del sitio. [Torres, Boch, 2008] La extensión definida por UWE involucra las fases relacionadas con los requerimientos y la estructura de navegación, no obstante lo que interesa es un modelo que permita definir las vistas de estas mismas fases para el proceso. Este nuevo modelo es llamado Modelo de flujo de procesos que es independiente de la navegación, consiguiendo así la tan deseada separación de conceptos en el diseño de aplicaciones web. Una diferencia importante entre este modelo y el de navegación es que precisamente no incluye elementos derivados del modelo conceptual. [Kappel, 2003]

El modelo de flujo de procesos es un refinamiento del modelo de procesos definido durante el nivel de análisis y representado mediante diagramas de actividad UML.

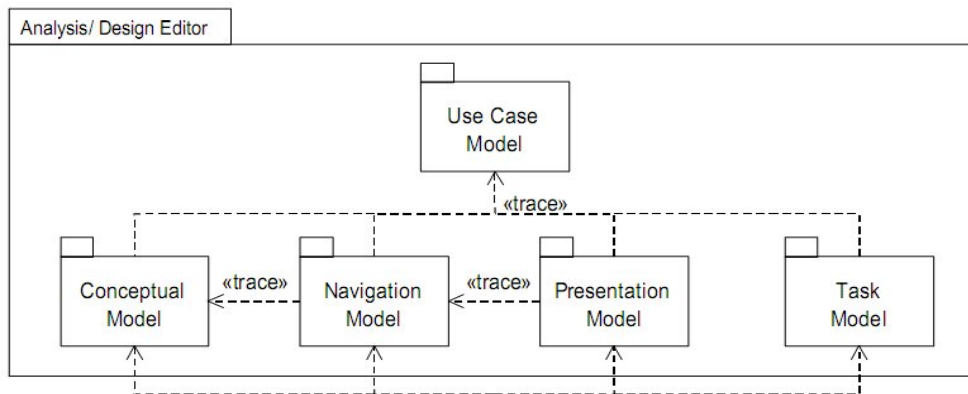


Figura 20: Proceso de desarrollo OOHDM²⁰

3.5 OO-H: Object-Oriented Hypermedia Method

Fue originalmente orientado hacia los requerimientos del usuario, está basado en objetos y parcialmente basado en estándares. Provee la notación y la semántica necesarias para el desarrollo de interfaces web personalizadas. Las fases propuestas por OO-H son soportadas por un ambiente de modelado conocido como VisualWADE. Esta herramienta provee un conjunto de compiladores de modelado que permiten generar un ejecutable para diferentes plataformas y lenguajes. [Torres, Boch, 2008]

OO-H trata con el proceso de modelado mediante el uso de tres modelos diferentes: El modelo conceptual es definido en términos de diagramas de clases UML que mantiene la información relacionada al dominio de objetos. El modelo de navegación es definido mediante diagramas de acceso que permite establecer menús, vistas, etc. El modelo de presentación es usado para definir las características de cada directorio de navegación. Estas características permiten cambiar detalles concretos como colores en los enlaces.

²⁰ The Expressive Power of UML-based Web Engineering, NORA Koch, ANDREAS Kraus Ludwig-Maximilians-Universität München. Germany.



También se permite el uso complementario de diagramas de actividad UML para representar el flujo de control de procesos con lo que OO-H mapea los elementos definidos en el modelo de procesos con elementos del modelo de navegación. [Torres, Boch, 2008]

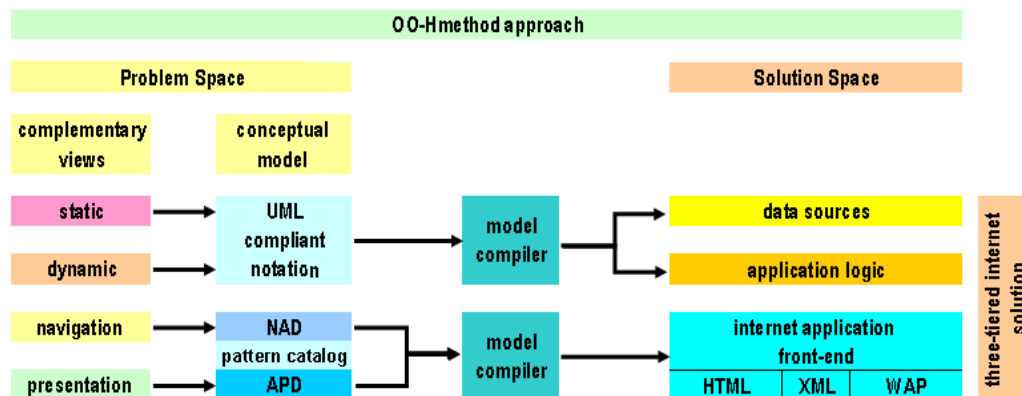


Figura 21: Proceso de desarrollo OO-H²¹

3.5.1 Proceso de desarrollo

OO-H establece cinco pasos para el desarrollo de aplicaciones web:

- ✓ El desarrollador debe definir el modelo conceptual añadiendo propiedades e incluso nuevos elementos para el diagrama de clases UML.
- ✓ El desarrollador debe redefinir el modelo de procesos establecido en el nivel de análisis utilizando casos de uso. Tomando en cuenta el modelo conceptual redefinido, es posible elaborar un diagrama de actividades más detallado en donde los enlaces entre los constructores de los procesos y el modelo conceptual estén establecidos por un identificador de estados de llamada UML.

²¹ The OO-H Project, Web Engineering Group; http://gplsi.dlsi.ua.es/iwad/oo_h_project/



- ✓ El diseñador debe aplicar un conjunto de reglas de mapeo previamente establecido para permitir la generación automática de diagramas de acceso basado en el diseño de los modelos de proceso, asegurando la trazabilidad entre ambos modelos.
- ✓ El diseñador debe realizar los cambios necesarios que pudieron resultar para el nuevo diagrama de actividades.
- ✓ Una vez que el modelo de navegación esté completo, el diseñador puede continuar el proceso definiendo aspectos de presentación del sistema. [Fielding, 2000]

3.6 WSDM: Web Site Design Method

En un principio, WSDM fue considerado como un método para el diseño de sitios web. Sin embargo el método ha ido evolucionando, permitiendo no sólo el diseño tradicional de sitios sino también el desarrollo de aplicaciones web semánticas. WSDM se ha extendido para tratar con nuevos requerimientos de modelado, lamentablemente no posee ninguna herramienta que brinde soporte tanto para el método como para los modelos definidos con la descripción de las aplicaciones web. [Torres, Boch, 2003]

3.6.1 Proceso de desarrollo

WSDM establece cinco fases, la cuales en algunos casos, están compuestas por fases más pequeñas. La primera etapa se conoce como Mission Statement Specification cuyo objetivo es la identificación del propósito del sistema web. La segunda fase conocida como Audience Modelling en donde los usuarios finales, que fueron escogidos en la fase anterior, son redefinidos en las conocidas clases de audiencia. La siguiente etapa es Conceptual Design, aquí el método especifica la información, funcionalidad y estructura de la aplicación web a un nivel conceptual. La penúltima etapa comprende Design Implementation a su vez formado por 3 subetapas llamadas: diseño de la estructura del sitio, diseño de presentación y diseño



de datos lógicos. Finalmente la fase de implementación es alcanzada con la unión del conjunto de modelos generados previamente. [Torres, Boch, 2008]

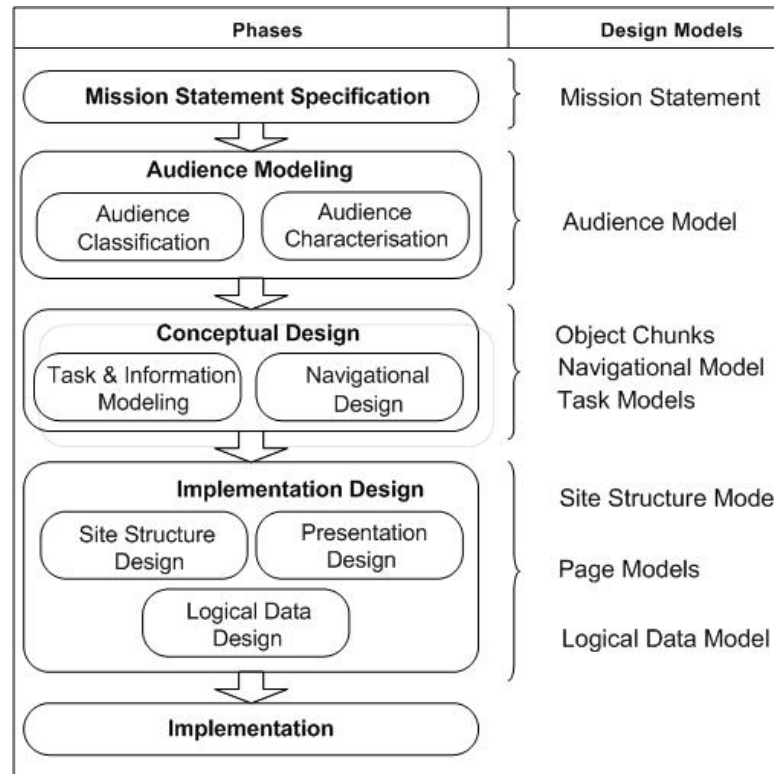


Figura 22: Proceso de desarrollo WSDM²²

3.7 AWE: Agil Web Engineering

El término ágil fue recientemente acuñado para referirse a un número de características esenciales para el desarrollo de software; entre éstas tenemos: Programación Extrema [XP], Desarrollo de software adaptativo [ASD] y Metodología para el desarrollo de sistemas dinámicos [DSDM].

²² WSDM, Web & Information System Engineering research group of the department of Computer Science of the Vrije Universiteit Brussel, <http://wsm.vub.ac.be/>



Sabemos que el éxito o fracaso de una aplicación web está directamente relacionada con el equipo de desarrollo y las organizaciones que estás detrás de éste. Un proceso de desarrollo ágil o cualquier otra metodología que se utilice, únicamente nos brindan una segunda oportunidad para afianzar el éxito o el fracaso del proyecto web. La meta principal de AWE (Agil Web Engineering) es satisfacer los objetivos del proyecto web sin dejar de lado la documentación, técnicas y herramientas que lo hacen posible.

AWE es un proceso iterativo e incremental, permite una entrega rápida y continua de versiones de software web, tiene la capacidad de manejar eficientemente los cambios en los requerimientos incluso en etapas avanzadas del desarrollo. Cada iteración durante el ciclo de desarrollo AWE es requerida para evaluar el proceso de desarrollo como tal. [McDonald, 2001]

3.7.1 El ciclo de vida del proceso AWE

Cualquier persona que tenga alguna experiencia en procesos de software tales como variantes del modelo de cascada, puede no entender inicialmente el diagrama de la figura 19; mientras que para otros que están más familiarizados con otro tipo de tecnologías, las fases de análisis del negocio, requerimientos, diseño, implementación, pruebas y evaluación resultan más confortables en cuanto a aspectos que involucran.

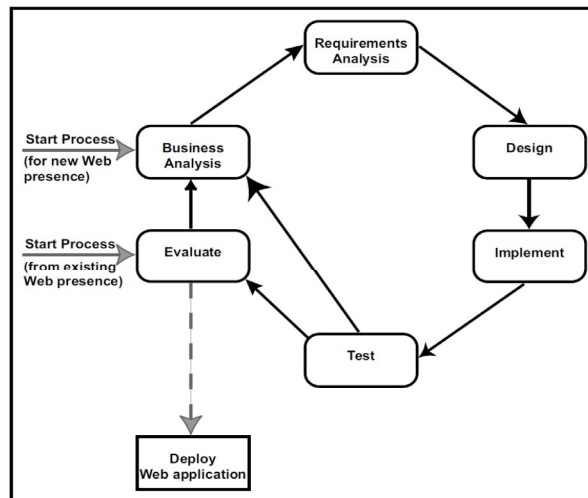


Figura 23: Ciclo de vida del proceso AWE²³

El único entregable que es requerido del proceso AWE es la aplicación web como tal. Esto no quiere decir que se pueda obtener algún beneficio de la producción inmediata de documentos, diagramas y otras notaciones. [Torres, Boch, 2008]

Tal como en la ingeniería de software, la ingeniería web deriva en entregables que comprende la integración de los datos con el software. Teniendo en cuenta el uso que el cliente final dará al sistema como tal, es de vital importancia que todos los desarrolladores, sin importar la función que estén cumpliendo dentro del proyecto se involucren activamente en el proceso de mejora de la navegabilidad del sitio web para el usuario final. [McDonald, 2001]

Las aplicaciones web deben ser altamente escalables y portables y sus componentes deben facilitar el reemplazo o actualización para asegurar un largo desempeño en el tiempo y hacer crecer los planes de negocio en los que se basa.

²³ Agile Web Engineering Process, 2001. Página 14.



3.7.1.1 Análisis del Negocio

El propósito fundamental del análisis del negocio es aclarar el panorama de los problemas que la nueva aplicación web solucionará y más importante aún que los desarrolladores entiendan lo que se debe lograr con la aplicación. Frecuentemente la fase de análisis del negocio se lleva a cabo por grupos diferentes de stakeholders o desarrolladores. Es responsabilidad de todos los miembros del equipo centrarse en descubrir los problemas reales que deben ser solucionados en cada proyecto web que se esté ejecutando. [McDonald, 2001]

Varios de los sitios web de organizaciones son desarrollados porque se vio la necesidad más no están orientados completamente a los objetivos del negocio. Se requiere mucho esfuerzo en conjunto del equipo y varias iteraciones en el ciclo de desarrollo para entender los verdaderos problemas que debe suprimir la aplicación web. Cada problema identificado representa un potencial riesgo que al igual que la ingeniería de software debe ser tratado de manera oportuna empezando por el de mayor relevancia. [McDonald, 2001]

Muchos de los proyectos empezarán el proceso AWE en la fase de análisis, pero si el equipo no logra encontrar ni llegar a un consenso sobre el espacio del problema a resolver, es necesario regresar a la fase de evaluación; que puede ser usada para mejorar el proceso de entendimiento en el grupo de trabajo y encontrar respuesta a la pregunta: ¿Se está construyendo el producto adecuado?

Cada desarrollador involucrado en una iteración AWE debería al menos ser capaz de responder si a la siguiente pregunta: ¿Entiendo los problemas que van a ser solucionados en esta iteración de AWE? Al igual que el equipo debe estar de acuerdo y responderse la misma pregunta en conjunto. Si la respuesta es no, se debe poner más atención en la fase de análisis del negocio. [McDonald, 2001]



3.7.1.2 Requerimientos

Es crucial que en el equipo, cada uno de los miembros esté de acuerdo con los problemas que se deben resolver antes de que las actividades de captura de requisitos inicien. En esta etapa se trata de definir de manera clara los requerimientos funcionales y no funcionales que normalmente existen en grandes cantidades en el desarrollo de una aplicación web y deben ser tratados para el éxito de la aplicación, especialmente aquellos que se refieren a la arquitectura incluyendo la interfaz y usabilidad. [Kappel, 2003]

AWE no sugiere el uso de técnicas específicas para soportar el proceso, no obstante si recomienda el uso de herramientas para el modelado de los requerimientos. Luego de que hayan sido definidos se debe empezar a crear un plan de pruebas que luego será ejecutado en una fase posterior.

3.7.1.3 Diseño

Involucra la comprensión, coordinación y comunicación en la mayoría de los aspectos antes de que una aplicación web pueda ser implementada. Cuando el proceso AWE es aplicado requiere comunicación entre desarrolladores de equipos separados dentro del mismo proyecto. Esto facilita la reutilización en los patrones de diseño de arquitectura lo que evitará a largo plazo problemas de portabilidad e incluso rediseño de la solución [McDonald, 2001].

Cada desarrollador involucrado en el proyecto web tiene la responsabilidad de asegurarse de que el equipo tenga una adecuada metodología en el diseño para la construcción de una aplicación web, ya sea que provenga de la experiencia o sea relativamente nueva de aplicar.

3.7.1.4 Pruebas

El objetivo de la fase de pruebas consiste en determinar si lo que se ha construido satisface los requerimientos del proyecto. Más allá de ejecutar pruebas sobre los requerimientos funcionales se debe enfocar atención sobre los requerimientos no funcionales de los cuales dependerá en mayor o menor grado el rendimiento de la aplicación web, tales como tamaño de



pantalla, ancho de banda, velocidad de procesador, memoria, etc. La elaboración del plan de pruebas debe destinarse a resolver incógnitas como: ¿Es un sistema seguro?, ¿Es un sistema escalable?, ¿Es un sistema portable? [McDonald, 2001]

Muchos desarrolladores crean sus propias pruebas para la aplicación web lo que al momento de entrar en ejecución puede dar resultados falsos u orientados a un comportamiento esperado e inducido. Incluso si los desarrolladores pruebas a conciencia y detalladamente sus componentes, resultaría mejor que todo el equipo se involucre en probar los entregables de la aplicación tanto como se pueda para simular el uso que el cliente final dará a la aplicación web con sus propios recursos. [McDonald, 2001]

3.7.1.5 Implementación

Mientras existen muchas diferencias entre las fases de diseño e implementación, ambas involucran decisiones que tiene un impacto crítico a nivel del éxito del proyecto. Todos los desarrolladores deberían colaborar y centrar su atención alrededor de la interfaz web, mediante reuniones para discutir y revisar la experiencia de usuario. Únicamente de esa manera se puede empezar el proceso para lograr la satisfacción completa del usuario final.

La figura muestra de una manera clara, la cronología de algunas de las metodologías de desarrollo y sus derivaciones:

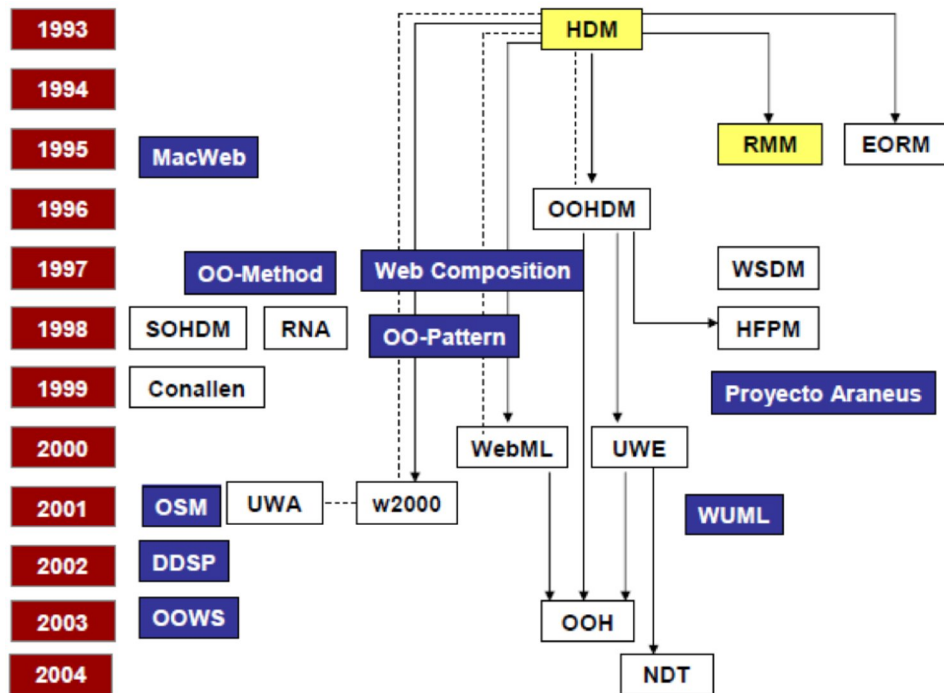


Figura 24: Cronología de las principales metodologías de desarrollo de aplicaciones web²⁴

A continuación se presenta una tabla comparativa con algunas de las metodologías descritas anteriormente y otras que son derivadas o no tan usadas en la actualidad:

Tabla 1: Comparación entre las metodologías de desarrollo de aplicaciones web

Metodología	Proceso	Técnica de Modelado	Representación Gráfica	Notación	Herramienta de Soporte
HDM	Desarrollo a largo y corto plazo	E-R	Diagrama E - R	E - R	
RMM	Diseño E-R, Diseño Slice, Diseño de navegación, Diseño de protocolo de conversión, Diseño de UI, Diseño de comportamiento en tiempo de ejecución,	E -R	Diagrama E - R, Diagrama Slice, Diagrama RMDM	E - R, Propio	RMCASE

²⁴ Metodología y Técnicas en proyectos software para la web, María José Escalona, José Mariano González, página 1.



	Prueba y construcción.				
EORM	Clases del entorno de desarrollo, Composición del entorno de desarrollo UI	OO	Diagrama de Clases, Diseño GUI	OTM	ONTOS Studio
OOHDM	Diseño conceptual, Diseño de navegación, Diseño abstracto de la UI, Implementación	OO	Diagrama de clases, Diagrama Navegacional Clase + Contexto, Diagrama de configuración ADV + diagrama ADV	OMT, UML, Propio, ADVs	OOHDM-Web
SOHDM	Análisis del dominio, Modelo en OO, Diseño de la vista, Diseño de navegación, Diseño implementación, Construcción.	Escenarios, Vistas OO	Diagramas de escenarios de actividad, Diagrama de estructura de clase, Vista OO, Esquema de enlace de navegación, Esquema de páginas.	Propio	
WSDM	Modelado del usuario, Diseño conceptual, Modelo objetos, Diseño de navegación, Diseño implementación, Implementación.	E – R, OO	Diagrama de Entidad Relación o Clase, Capas de Navegación.	E – R, OMT, Propio	
WAE	Manejo de proyecto, Captura de requerimientos, Análisis, Diseño, Implementación, Prueba, Desarrollo, Configuración y manejo de cambios.	OO	Diagramas UML	UML	Rational Rose

Ver el Anexo 3, se desarrolló una plantilla para la presentación y definición de un plan de pruebas genérico para una aplicación web.

Según M.J. Escalona y N. Koch, existen otras metodologías que también han sido implementadas, mejoradas pero por diversas circunstancias su uso es más restringido a tipos específicos de proyectos web, estas son:

SOHDM Scenario-based Object-Oriented Hypermedia Design Methodology: Esta propuesta presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema. Para ello, propone 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. En este diagrama de contexto se identifican las entidades externas que se comunican con el sistema, así como los



eventos que provocan esa comunicación [Lee, 1998]. 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.

RNA: Relationship-Navigational Analysis [Bieber, 1998] 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, Elementos de interés, Análisis del conocimiento, Análisis de la navegación, Implementación del análisis: Una vez obtenido el esquema final en el que ya se encuentran incluidos los aspectos de navegación, se pasa el esquema a un lenguaje entendible por la máquina. 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.

HFPM: Hypermedia Flexible Process Modeling La propuesta de HFPM describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software. HFPM propone un total de trece fases para las cuales se especifican a su vez una serie de tareas. Para este estudio es principalmente relevante la primera fase denominada de modelado de requisitos cuyas tareas son las siguientes [Olsina, 1998]:

- ✓ Descripción breve del problema.
- ✓ Descripción de los requisitos funcionales mediante casos de uso.
- ✓ Realizar un modelo de datos para esos casos de uso, proponiendo el uso de un modelo de clases.
- ✓ Modelar la interfaz de usuario.
- ✓ Modelar los requisitos no funcionales.

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). El proceso de desarrollo de



W2000 se divide en tres etapas: análisis de requisitos, diseño de hipermedia y diseño funcional. [McDonald, 2001]

NDT - Navigational Development Techniques NDT (Navigational Development Techniques) es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones web

Design-driven Requirements Elicitation es parte del proceso design-driven que proponen Lowe y Eklund para el desarrollo de aplicaciones en el entorno Web. La propuesta consiste en realizar la captura, definición y validación de requisitos durante el proceso de diseño [Lowe, 2002].

A nivel de requerimientos, la siguiente tabla muestra la comparaciones entre las metodologías:

Tabla 2: Comparación entre requerimientos de necesitan las metodologías de desarrollo.

Metodología	Req. Datos	Req. Interfaz	Req. navegacionales	Req. personalización	Req. transaccionales	Req. no funcionales
WSDM	X			X		X
SOHDM	X	X			X	
RNA	X	x	X		X	
HFPM	X	X	X			X
OOHDM	X	X	X			
W2000			X	X	X	
UWA	X	x	X	X	X	
NDT	X	X	X	X	X	X
DDDP	X	X	X	X	X	X

3.8 Análisis de las Metodologías de Desarrollo Web

Luego de este análisis de las principales metodologías, la metodología con mayor aceptación en proyectos web y la más óptima es la metodología de desarrollo ágil AWE, ya que la meta principal es satisfacer los objetivos del proyecto web sin dejar de lado la documentación, técnicas y herramientas para la consecución del proyecto. Permite una entrega rápida y continua de versiones y maneja eficientemente los cambios en los requerimientos.



En 2002, Scott Ambler decidió crear y dar el paso inicial en lo que sería la metodología de desarrollo ágil, que no es otra cosa que una filosofía rompedora de paradigmas, una filosofía que se basaría en eliminar de la metodología de desarrollo todo el tiempo invertido en planificar a la perfección y dar más peso esfuerzo a crear con sentido común y con objetivos claros en el desarrollo.

“El desarrollo ágil no es una metodología con pasos establecidos, es simplemente una manera de pensar y de trabajar.” [Manchón, 2009]

AWE no es un medio sino un fin, una filosofía que funciona bien con proyectos que necesitan respuestas rápidas, hacer algo y ver los cambios, no es necesario planificar hasta el último detalle ni tampoco el desarrollo de un proyecto a nivel artesanal. AWE se enfoca en las prioridades del usuario y no es ver cómo se realizarán las diferentes implementaciones para la solución web, no es una ley ni un principio y no se puede aplicar a todos los proyectos.

AWE da origen al AWEM-ESP (“AWEM Evaluation and Selection Process”), pero no sirve para todo tipo de proyectos y usuarios, tiene su base en las disciplinas de RUP y es la metodología en desarrollo actual que apunta a ser la formalidad de AWE. El modelado mediante el uso de AWEM produce artefactos como: Modelo de casos de uso, Modelo de datos físico y conceptual, Modelo Navegacional y Modelo de Presentación. [ALGHAMDI, 2009].

[Marquez, 2008] define un manifiesto sobre la metodología AWE que resume claramente los beneficios que brinda esta metodología y brinda un conjunto de normas para el desarrollo ágil.

- ✓ Crear un grupo de trabajo de pocas personas.
- ✓ Reducir los procesos de validación, tratar de encajar esta responsabilidad en el usuario final.
- ✓ Diseñar aplicaciones simples.



- ✓ Utilizar un prototipado rápido.
- ✓ Definir los requerimientos conjuntamente con el usuario.
- ✓ Simplicidad, ante todo simplicidad
- ✓ Pruebas.
- ✓ Crear pensando en componentes y modelos, ensamblar y probar.
Luego corregir.
- ✓ Medir el riesgo y el impacto.
- ✓ Esta filosofía es compatible con el mundo 2.0 en el que vivimos.
- ✓ Innovar y reinventar.

Es la metodología de desarrollo que se adapta más efectivamente al desarrollo del sitio web de la UTPL por tanto se recomienda su aplicación en este proyecto en particular, por ello se ha realizado un estudio más detallado de esta metodología.

En la siguiente tabla, se muestra una recopilación de las técnicas utilizadas por las diferentes metodologías de desarrollo web, algunas son consideradas como estándares otras son tratamientos empíricos que se le dan a cada proyecto dependiendo de su enfoque.



Tabla 3: Comparación entre los requisitos de las distintas metodologías de desarrollo de aplicaciones web

		WSDM	SOHDM	RNA	HFPM	OOHDM	UWE	W2000	UWA	NDT	DDDP
Captura	Entrevistas	✓		✓			✓			✓	✓
	JAD									✓	
	Brainstorming									✓	
	Concept Mapping	Rol- Actividad									
	Casos de uso					✓					
	Cuestionario/Checklist						✓				
	Prototipos										✓
	Otra técnicas		DFD								
Definición	Lenguaje natural	✓		✓	✓						
	Glosarios				✓		✓			✓	
	Patrones/plantillas								✓	✓	
	Escenarios		SAC				✓				
	Casos de uso				✓	✓	✓	✓	✓	✓	
	Lenguaje formal								XML		
	Sketches de interfaz				✓						
	Prototipos										✓
	Otras técnicas		Lista evento				UIDs		Grafo requisitos	Frases en BNL	
Validación	Reviews/Walk-throughs						✓			✓	
	Auditorías						✓				
	Matriz trazabilidad									✓	
	Prototipos				✓		✓				✓
	Otras técnicas								Grafo requisitos		



CAPITULO IV

4 Métricas en las Aplicaciones Web

4.1 Evaluación de una Aplicación Web

Muchas características y atributos de calidad de artefactos Web tales como usabilidad, navegabilidad, seguridad, características de búsquedas y recuperación de información, rendimiento, accesibilidad, entre otros, deben ser tenidas en cuenta durante el proceso de desarrollo a modo de garantizar la calidad en la satisfacción de los requerimientos que el usuario solicita.

La meta principal en el desarrollo de aplicaciones web es producir artefactos de calidad, los que deben estar basados en un conjunto de características y atributos deseados, observables y sobre todo cuantificables. Se debe asegurar los mecanismos para construir productos que cumplan estas características y atributos a partir del planteo de un conjunto de requerimientos de calidad. Por medio de la evaluación de la calidad de artefactos web, se puede comprender el grado de cumplimiento de un conjunto de características y atributos con respecto a los requerimientos de calidad establecidos inicialmente entre el cliente y el equipo de desarrollo. [Olsina, 1999]

Para lograr este cometido, se debe introducir dentro del modelo de ciclo de vida de desarrollo de artefactos web, actividades, metodologías y procedimientos para el aseguramiento y control de calidad. El objetivo podría consistir en evaluar el estado actual de un artefacto en la fase operativa, o comparar la calidad de dos o más artefactos semejantes [Olsina, 1999].

Para determinar los criterios de medición básicos e implementarlos y de esta manera obtener indicadores parciales y globales, se requiere una metodología integral. Es decir, un conjunto bien definido y cooperante de



estrategias, métodos, modelos y herramientas que, aplicados a las distintas actividades del proceso, produzca los indicadores o resultados finales para poder conocer cuantitativamente la calidad de una aplicación web.

El proceso de evaluación de una aplicación web resulta un tanto complicado, ya que hay un sin número de consideraciones que se deben analizar, por ejemplo: se debe evaluar la aplicación desde el contexto de la organización que necesita el software, se debe evaluar desde la perspectiva del usuario final y sus requerimientos, o se obtiene alguna ventaja al aplicar un modelo formal de evaluación. [Olsina, 1999]

La evaluación de sistemas de software tiene más de 30 años de estar en continua evolución, la mayoría de las técnicas y prácticas actuales para desarrollar software y evaluar en la Web son propias de cada proyecto y pocas veces funcionan en otros semejantes. El mayor énfasis puesto en la evaluación y monitoreo del uso de artefactos y recursos en la web por parte de los usuarios, se centra en los registros de la aplicación, estados de los enlaces, tipos de usuarios, tiempos de carga, uso de servidores, etc. Esto no es suficiente, se requiere una visión más amplia en la concepción de la calidad de un artefacto web, conforme a las necesidades de los usuarios. [Olsina, 1999]

A continuación se detallan algunos aspectos a considerar en la evaluación de un sitio web, sin importar el origen del proyecto y los objetivos que deben cumplirse. Estos parámetros posteriormente serán medidos más adelante por la herramienta seleccionada, LinkScan para determinar sus valores en algunos sitios web de referencia que se ha tomado.

4.2 La evaluación de aplicaciones web en la actualidad

En conclusión, ¿Qué se está utilizando actualmente para la evaluación web? ¿Se utilizan heurísticas? ¿La evaluación se realiza en base a plantillas pre-diseñadas para un conjunto de aplicaciones generales? ¿En qué se basa



la evaluación web? Hasta la actualidad no existe una metodología formal, completa que dicte los pasos necesarios para llevar a cabo esta tarea. Todas las evaluaciones de calidad que se realizan son inherentes a su propio proyecto. A mediados de 1990 en la Universidad de Milán, investigadores propusieron una metodología de evaluación orientada al diseño [Garzotto, 2007], que se centraba en el análisis del artefacto, empleando criterios generales de usabilidad pero no métricas. Las dimensiones que incluye la evaluación son: [Olsina, 1999]

- ✓ Contenido: Relacionado a los objetos de información incluidos en la aplicación.
- ✓ Estructura: La organización del contenido.
- ✓ Presentación: Cómo se muestra a los lectores el contenido y las funciones de la aplicación.
- ✓ Dinámica: Cómo los usuarios se mueven e interactúan con los elementos y componentes de información
- ✓ Interacción: Cómo se usan los componentes dinámicos.

4.2.1 Criterios de Evaluación o Heurísticas

Según Garzotto, en su publicación hace referencia a los siguientes criterios:

- ✓ Riqueza: Que representa la abundancia de ítems de información y el modo de alcanzarlos.
- ✓ Facilidad: Que evalúa la accesibilidad a la información y cuán fácil de operar y comprender son las operaciones.
- ✓ Consistencia: Información precisa.
- ✓ Auto-evidencia: Cuán intuitiva es la aplicación web.
- ✓ Predictibilidad: Expresa cuán bien los usuarios se anticipan al resultado de una operación.
- ✓ Legibilidad: Expresa un sentimiento de la validez de toda la aplicación.
- ✓ Reutilización: Considera el uso de objetos y operaciones en diferentes contextos y propósitos. [Olsina, 1999]



Sin embargo, este enfoque es sólo conveniente cuando el problema de evaluación es más bien simple e intuitivo. En el caso en que se desee expresar a la aplicación con mayor cantidad de factores y atributos de más bajo nivel se vuelve difícil una evaluación justificable y precisa.

A finales de 1997 y principios de 1998, se elaboró el conocido “Cuestionario de satisfacción del usuario”, basado en estudios previos de aplicaciones de escritorio unido a opiniones de diseñadores, usuarios, administradores de sitios y sus experiencias. Todo ese conjunto de opiniones fue plasmado en un modelo de cinco categorías con un total de 70 preguntas evaluadas de 1 a 7. [Garzotto, 2007]. Lo denominaron WAMMI (Web Analysis and Measurement Inventory). Las categorías son:

1. Atracción: el grado que a los usuarios les gusta el sitio si lo hallan agradable de usar.
2. Control: el grado en que los usuarios se sienten comprometidos, ya que el sitio les permite navegar con facilidad.
3. Eficiencia: el grado con que los usuarios sienten que el sitio tiene la información que están buscando, si funciona a velocidad razonable, y si se adapta al navegador.
4. Ayuda: el grado que los usuarios sienten que el sitio les posibilita resolver sus problemas con respecto a encontrar información y explorarla.
5. Aprendizaje: el grado que los usuarios sienten que se acostumbran al sitio cuando ingresan por primera vez, y el grado que los usuarios sienten que pueden aprender otras facilidades o acceder a otra información una vez que han comenzado a usar el sitio.

El valor máximo es de 70 y un mínimo de 50. No obstante es una metodología que trae consigo algunos problemas, puesto que para la elaboración de los cuestionarios, el grupo de usuarios objetivo fue seleccionado meticulosamente mientras que en un ambiente natural no se presenta tal actividad. Durante la elaboración del cuestionario, menos del



5% del número de visitantes del sitio llenaban el cuestionario; de estos, muchos debían desecharse por incompletos, el cuestionario no podía ser llenado en un tiempo menor a 20 minutos.

4.3 Métricas Web Automatizables

En esta sección discutiremos los aspectos de automatización de métricas, y más precisamente, métricas orientadas a la Web. Algunos atributos medidos directa o indirectamente pueden ser parcial o totalmente automatizables en consideración al tipo de recolección de datos. Por lo tanto, para determinar el valor de un atributo, puede hacerse de un modo manual, o asistido total o parcialmente por un instrumento de medición o herramienta de recolección de datos y/o cómputo. [LAFUENTE, 2000].

Si bien muchos de los datos para las métricas empleadas son recolectados manual y observacionalmente, la recolección de datos automática es muchas veces el único mecanismo para obtener resultados de un modo confiable y efectivo. Este es el caso para medir atributos como enlaces rotos, páginas de acceso rápido, imagen con título, entre otros.

4.3.1 Cantidad de enlaces rotos en un sitio.

El chequeo de un enlace roto se comprueba por medio del tipo de error 404, conforme a los códigos de estado del protocolo HTTP. Asimismo, en consideración del tipo de error devuelto, se puede determinar la cantidad de enlaces que conducen a páginas no accesibles.

4.3.2 Cantidad total de enlaces en un sitio.

Realizando un seguimiento de los enlaces que conforman la estructura navegacional de un sitio Web se puede calcular de manera automática la cantidad de enlaces que posee un sitio. La herramienta Web-site MA



permite calcular con gran efectividad esta métrica, empleando procedimientos recursivos en el seguimiento de todos los enlaces que posee el sitio que se está analizando. [LAFUENTE, 2000].

4.3.3 Porcentaje de enlaces rotos de un sitio.

Mediante el empleo de métricas directas, se ha establecido la siguiente fórmula para conocer este valor:

$$\% \text{enlaces rotos} = \frac{\# \text{enlaces rotos internos} + \# \text{enlaces rotos externos}}{\# \text{total enlaces}} \times 100$$

4.3.4 Cantidad de nodos web muertos en un sitio.

Son aquellas páginas que no tienen ningún enlace de retorno. Si un visitante accediera a una página muerta desde un URL externo al sitio, no tendría posibilidad de navegar dentro del él. Sería simplemente, una página sin ningún vínculo hacia el sitio. Esto es fácil de automatizar, comprobando por cada página que se visita que no exista ninguna etiqueta HREF.

4.3.5 Cantidad de imágenes que referencian a la propiedad ALT.

En cada página web, se debe proveer texto alternativo para cada imagen u objeto gráfico dado ya que comunican información visual. La ALT en cada imagen permite incluir texto alternativo para cada una de ellas, y de esta manera, favorecer la legibilidad de la página. No obstante, para comprobar la calidad del texto alternativo, la presencia de esta propiedad sola, no la garantiza dado que el texto puede haber sido generado automáticamente con alguna herramienta como FrontPage o Adobe GoLive, o vía tecnologías como PHP o ASP. [LAFUENTE, 2000].



4.3.6 Cantidad total de imágenes en un sitio.

Se verifica la presencia de imágenes en un sitio web con tan sólo verificar su existencia a través de la propiedad IMG. Luego con realizar un simple cómputo, sumando todas las imágenes por cada una de las páginas analizadas, podemos obtener la cantidad total de imágenes de un sitio.

4.3.7 Porcentaje de presencia de la propiedad ALT.

El porcentaje se lo calcula de la siguiente manera:

$$\%PresenciaALT = \frac{\#imágenesALT}{\#total imágenes} \times 100$$

4.3.8 Cantidad total de páginas de un sitio.

Al igual que los otros atributos medibles, es posible contabilizar el número total de imágenes de un sitio web.

4.3.9 Promedio de enlaces por página.

El promedio de enlaces por páginas se obtiene dividiendo el total de los enlaces que posee el sitio sobre el total de páginas que contiene éste. [Chile, 2000]

$$PromedioEnlacesPágina = \frac{TotalEnlaces}{TotalPáginas}$$

4.3.10 Tamaño de una página.

Se mide el tamaño de una página estática, considerando todos sus componentes textuales, gráficos, tabulares, etc.



4.3.11 Páginas de acceso rápido.

El tamaño de cada página se especifica en función del tiempo de espera y de la velocidad mínima de carga establecida para el medio de transmisión dado. Se especifica un tamaño umbral aceptable, para el tamaño total de cada página, por ejemplo, el de 35,2 Kb. Una página de este tamaño requiere 20 segundos para ser bajada a una tasa de 14,400 bps. Ese es el tiempo máximo aceptable que un usuario debe esperar, sin que se ponga impaciente. [LAFUENTE, 2000].

4.3.12 Mantenimiento del color de los enlaces.

Se analizan los códigos de color para las propiedades LINK (en el caso de los enlaces) y VLINK (para los enlaces visitados) en la etiqueta BODY (tag).

4.3.13 Cantidad máxima de marcos o frames.

Verificando en cada una de las páginas analizadas si existe la etiqueta FRAME, se puede comprobar de manera automática la cantidad de frames o marcos que contiene la página (sumando la cantidad de veces en que aparece en la página). [Chile, 2000]

4.4 Metodología de Evaluación de Calidad de Sitios Web

La metodología de evaluación de calidad de sitios web o Web-Site QEM (Quality Evaluation Method); es integral, flexible y robusta y cubre la mayor parte de las actividades en el proceso de evaluación, comparación, y selección de artefactos web. [Olsina, 1999]

Con el transcurrir del tiempo, las aplicaciones web han ido creciendo en complejidad y servicios que otorga, lo que ha planteado grandes desafíos en las diferentes etapas de su construcción. Precisamente para enfrentar estos retos resulta efectiva la estrategia de modelar los procesos y productos lo



que nos permite comprender, analizar, optimizar y predecir la calidad final de los mismos. Web-QEM se basa en estos modelos para su construcción y especificación y puede ser utilizada en cualquier etapa del proceso de construcción de la aplicación web, pero para fines de este trabajo, me centraré en su uso en proyectos finalizados y/u operativos.

Web-Site QEM, incluye un conjunto de fases, actividades, productos, modelos y constructores de proceso que se detallan a continuación. Una de las metas principales de la evaluación y comparación de calidad de artefactos web, radica en comprender el grado de cumplimiento de un conjunto de características y subcaracterísticas con respecto a los requerimientos de calidad establecidos. [Garzotto, 2007].

4.4.1 Fases de Web-Site QEM

A continuación se describen las principales fases, actividades, modelos, y algunos constructores que intervienen en el proceso de evaluación de la calidad, tal como se muestra en la figura 25 [Garzotto, 2007]:

- ✓ Planificación y programación de la evaluación de calidad.
- ✓ Definición y especificación de requerimientos de calidad.
- ✓ Definición e implementación de la evaluación elemental.
- ✓ Definición e implementación de la evaluación global.
- ✓ Análisis de resultados, conclusión y documentación.
- ✓ Validación de métricas.

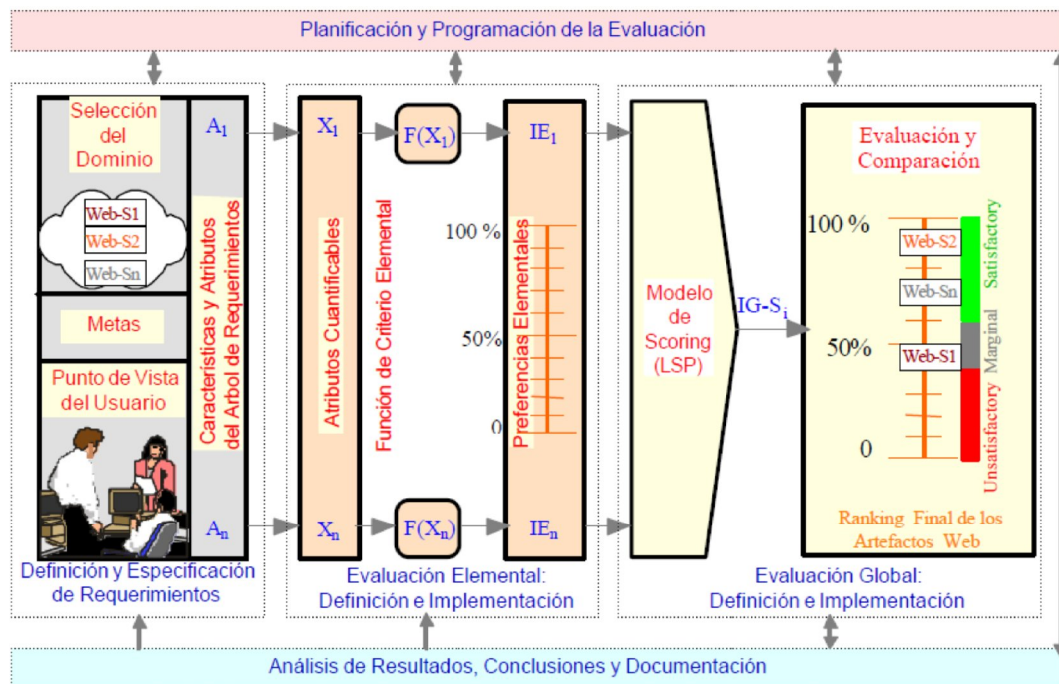


Figura 25: Principales módulos que intervienen en Web-Site QEM²⁵

La fase de Planificación y programación de la evaluación de calidad, contiene actividades y procedimientos de soporte, con el fin de establecer las principales estrategias y metas del proceso; permite seleccionar un modelo de proceso de evaluación, asignar métodos, agentes y recursos a las actividades; programar y replanificar una vez en marcha el proceso de evaluación. [Garzotto, 2007]

La fase de Definición y especificación de requerimientos de calidad, la misma trata con actividades y modelos para la determinación, análisis y especificación de los requerimientos. A partir de un proceso de medición orientado a metas, y con el fin de evaluar, comparar, analizar, y mejorar características y atributos de artefactos web, los requerimientos deben responder a necesidades y comportamientos de un usuario y dominio dados. [Garzotto, 2007]

²⁵ Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web, Luis Antonio Olsina, 1999, pág. 40.



Con respecto a la fase de Definición e implementación de la evaluación elemental, la misma trata con actividades, modelos, técnicas y herramientas para determinar métricas y criterios de evaluación para cada atributo cuantificable. Una vez definidos y consensuados los criterios para medir cada atributo, se debe ejecutar el proceso de recolección de datos, computar las métricas y preferencias elementales, y documentar los resultados. [Garzotto, 2007]

La fase de Definición e implementación de la evaluación global trata con actividades, modelos, y herramientas para determinar los criterios de agregación de las preferencias de calidad elemental a fin de producir la preferencia global, para cada sistema seleccionado. Una vez definidos y consensuados los criterios, se debe llevar a cabo el proceso de cálculo y ranquin. [Garzotto, 2007]

En la fase de Análisis de resultados, conclusiones y documentación se trata con actividades de análisis y comparación de las preferencias de calidades elementales, parciales y globales, y, asimismo, la justificación de los resultados. Por otra parte, se utilizan herramientas y mecanismos de documentación para facilitar la interpretación de los datos y su seguimiento. [Garzotto, 2007]

Por último, la Validación de las métricas es un proceso importante en la disciplina de evaluación de aplicaciones web. Podemos definirla como el proceso de asegurar que las medidas sean una caracterización numérica apropiada del atributo evaluado en el software. [Garzotto, 2007]

4.4.2 Herramientas de automatización de métricas

En la actualidad, no existen muchas herramientas que permitan la automatización de métricas para la Web, y no existen herramientas específicas que se adapten al cálculo de métricas prescriptas o preestablecidas por algún estándar o por alguna metodología. Lo que



existen son herramientas que permiten analizar sitios Web, y recolectar de ellos cierta información que el usuario selecciona previamente para luego emitir un reporte con los datos recolectados. A continuación se destaca algunas de las herramientas que existen actualmente en el mercado y que pueden ayudar al usuario, tanto evaluador como diseñador de un sitio, a analizar ciertas características y atributos del mismo, con la finalidad de incrementar la funcionalidad o la usabilidad. [Lafuente, 2000]

4.4.2.1 LinkScan™

LinkScan²⁶ en su versión 12.0, es una herramienta que permite validar, principalmente, los enlaces de un sitio Web como así también detectar archivos huérfanos (o páginas muertas), es decir, páginas que no posean ningún enlace de retorno o navegación. La figura 26 muestra una de las pantallas principales de esta herramienta en donde el usuario evaluador puede configurar varios proyectos. En cada uno de esos proyectos se puede introducir las opciones de URL, agregar una descripción para el proyecto, y además, seleccionar las opciones de análisis como chequear enlaces externos al sitio y chequear archivos huérfanos. Posee entre otras cosas, una opción que permite configurar la velocidad de análisis, indicándole la cantidad de procesos a ser lanzados en forma paralela, es decir, la cantidad de páginas a ser analizadas simultáneamente. Obviamente que cuantas más páginas sean analizadas simultáneamente el uso de la CPU va a ser mayor, pero en definitiva, la duración del análisis será menor debido a la ejecución paralela de las tareas. En el siguiente capítulo haremos uso de la herramienta para determinar la mayoría de las métricas que nos conciernen a nuestro estudio en un caso particular como es el estudio del sitio web de la UTPL.

²⁶ <http://www.elsop.com/>

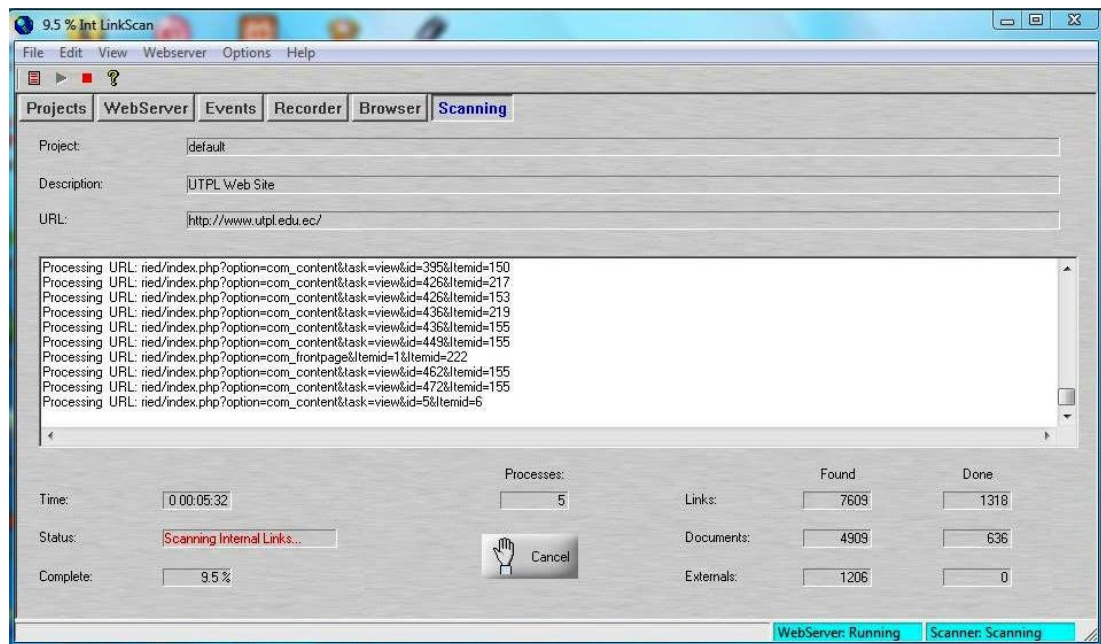


Figura 26: Principales módulos que intervienen en Web-Site QEM

4.4.2.2 WebKing™

La herramienta WebKing ²⁷ permite prevenir y detectar errores cuando una persona se encuentra desarrollando una aplicación orientada a la Web. Entre sus características más sobresalientes, permite analizar sitios Web que posean estructuras dinámicas de información, permitiendo además analizar la funcionalidad del sitio. Posee opciones para verificar los enlaces, las imágenes y cualquier otro recurso. Permite emitir reportes estadísticos sobre el análisis realizado. No es completamente una herramienta orientada a un usuario evaluador de calidad, sino más bien, orientada hacia aquel usuario dedicado a la construcción y desarrollo de sitios web complejos. La figura 27 muestra una pantalla del programa principal.

²⁷ <http://www.parasoft.com/products/>

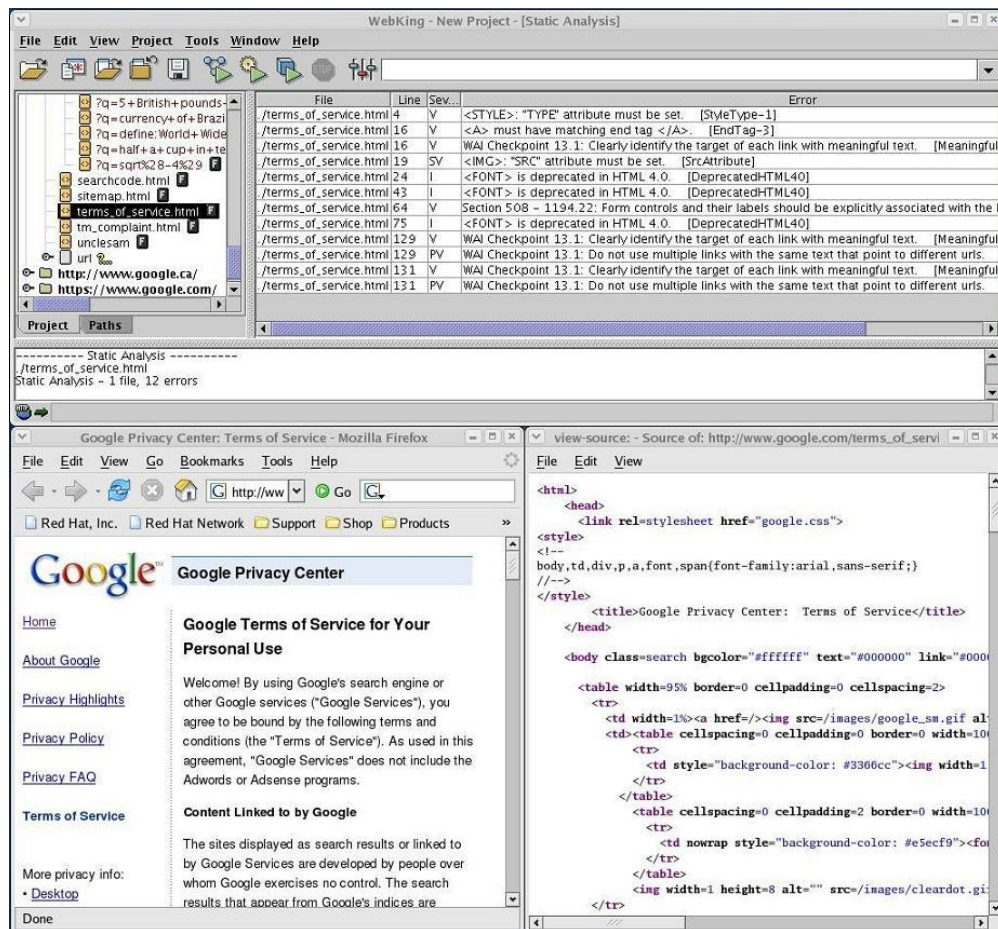


Figura 27: LinkScan 12.0 Pantalla principal

4.4.2.3 SiteSweeper™

Esta herramienta no sólo recolecta datos en línea sino que también genera informes textuales, gráficos e hipertextuales, que facilitan el ulterior análisis y documentación. La figura 1.3 muestra una de las pantallas de SiteSweeper, en donde el usuario puede elegir las opciones a evaluar de un sitio Web y personalizar el reporte final, para el caso de enlaces rotos, como se puede apreciar en la figura 28. Esta herramienta permite además analizar el tiempo de acceso a las páginas de un sitio, muestra los porcentajes de presencia de la propiedad ALT, en los documentos fuentes HTML. De igual manera, recolecta datos para la cantidad de enlaces, enlaces rotos, enlaces promedio por página, entre otros. Lamentablemente, esta herramienta fue discontinuada en su soporte y no se encuentra disponible para su uso, se puede enviar una comunicación a la empresa



que tiene los derechos pero hasta el término de esta investigación no se obtuvo respuesta de la misma.

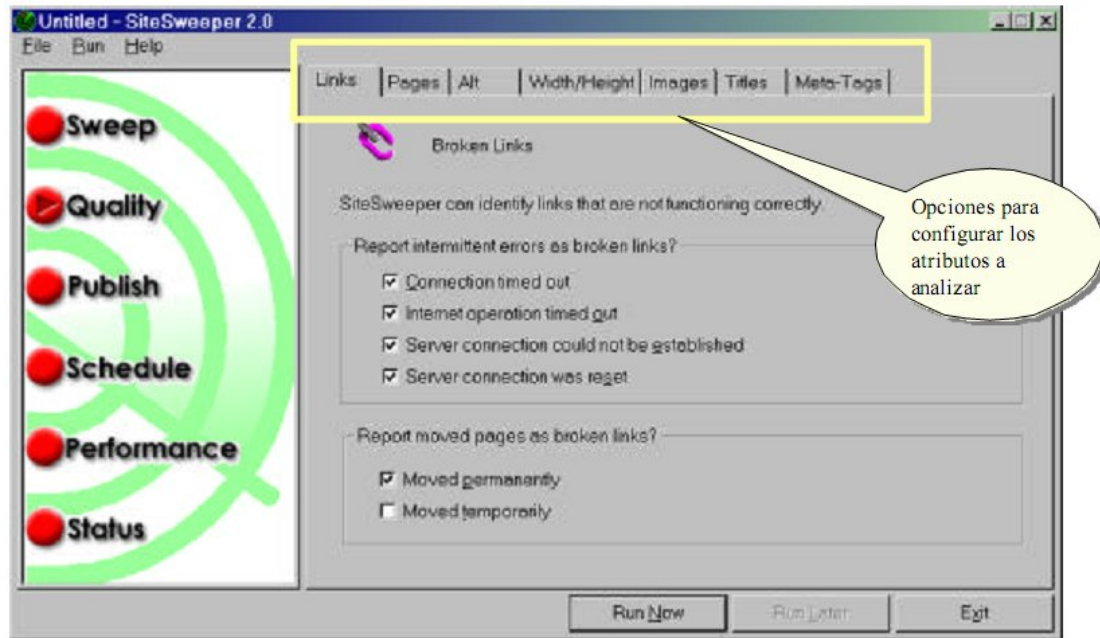


Figura 28: SiteSweeper 2.0. Pantalla principal

A continuación se indica un cuadro comparativo de las 3 herramientas analizadas anteriormente:

Tabla 4: Cuadro comparativos principales características de herramientas para automatización de métricas

Herramienta	Procesos Paralelos	Orientada al desarrollador	Orientada al usuario	Informes gráficos	Informes textuales
LinkScan	X		X		X
WebKing		X			X
SiteSweeper			X	X	



CAPÍTULO V

5 Caso de Estudio

5.1 *Evaluación del Sitio Web de la UTPL.*

Aplicando las plantillas propuestas en el Anexo 1, las cuales fueron tomadas de Swebok y adaptadas a las necesidades del sitio, se realizó la evaluación del sitio web de la Universidad Técnica Particular de Loja. Dichas plantillas permiten recabar información sobre las etapas básicas en el proceso de análisis, diseño e implementación del sitio web.

Para evaluar el proceso de desarrollo basados en los procesos de la Ingeniería del Software se utilizaron plantillas planteadas por Swebok, las mismas que se enfocan a:

1. Determinación del alcance e iniciación: la planificación, el establecimiento del proyecto y la revisión y evaluación.
2. Requerimientos de Software: requerimientos de proceso, validación de requerimientos, análisis de requerimientos y especificación de requerimientos.
3. Diseño del Software: tenemos puntos como arquitectura y estructura del software, evaluación y análisis de la calidad del diseño del software y métodos y estrategias en el diseño del software.

La evaluación de sitios web está sometida a muchos sesgos que afectan a su fiabilidad y debe realizarse en los primeros prototipos del sitio antes de su diseño definitivo. Se ha hecho la modificación en las plantillas para obtener datos de Pruebas del Software, Construcción del Software, Gestión de la Configuración del Software, Calidad del Software, y finalmente Métodos y Herramientas de la Ingeniería Web cada una adaptada para ser utilizada en el análisis del portal de la UTPL. Las plantillas presentadas a



continuación, tienen como finalidad el determinar bajo qué condiciones fue desarrollado el sitio web de la UTPL, que metodología fue aplicada, que pasos se siguieron, para ello; el cumplimiento de una actividad marcará una valoración para obtener el score final.

Los valores para la calificación en las plantillas están divididos en 25, 50, 75 y 100%, lo que indica que un requisito se ha cumplido en su totalidad 100%, medianamente 50%, en una etapa avanzada 75% y en un etapa inicial 25%. Si el requerimiento que está siendo evaluado no tiene como respaldo la documentación respectiva, se considera como que el requisito no es cumplido. Este método de evaluación proviene del hecho de que el análisis de la documentación para comprobar su estado no está contemplado dentro del presente trabajo por lo cual las plantillas de evaluación se remiten únicamente a verificar el cumplimiento de condiciones mínimas en el desarrollo del sitio web de la UTPL.

Fueron realizadas las entrevistas necesarias con el personal a cargo del proyecto, Ing. Germania Rodríguez (Líder de proyecto), Ing. Alexandra González (Administradora) e Ing. Ramiro Ramírez (Programador) y los resultados son los siguientes:



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	1	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

1. GESTIÓN DE LA INGENIERÍA DEL SOFTWARE WEB:

	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Determinación del Alcance e Iniciación								
Determinación y negociación de requerimientos			X				X	Existen actas de reuniones
Análisis de Factibilidad					X		X	
Procesos para la revisión de requerimientos				X		X		Informes por medio de e-mails
Planificación del Proyecto Software								
Planeamiento de procesos					X		X	
Determinación de entregables			X			X		Correos electrónicos que posee el responsable
Estimación de Costos, esfuerzo y cronogramas		X				X		Presentaciones
Distribución de recursos			X				X	Responsables por sección, Coordinación central
Manejo de riesgos			X				X	A nivel de hardware y respaldos
Gestión de calidad			X				X	Mejora de accesibilidad
Manejo de Planes	X						X	No se ha realizado
Establecimiento del Proyecto								
Implementación de planes					X		X	
Gestión de la información de los proveedores					X		X	
Implementación de medidas a los procesos		X					X	
Monitoreo de procesos			X				X	Continuo
Control de procesos			X				X	
Reportes	X						X	Reportes informales vía email



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	2	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

Revisión y Evaluación							
Determinación de la satisfacción de los requerimientos			X			X	
Revisión y evaluación del desempeño					X	X	



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema: Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.				
	Página	3	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

2. REQUERIMIENTOS DE SOFTWARE WEB:


Actividades Básicas Requerimientos de Software	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Definición de requerimientos de software			X				X	
Requerimientos de producto y de proceso			X				X	
Requerimientos funcionales y no funcionales			X			X		Focus Group
Requerimientos cuantificables			X				X	
Requerimientos de software y requisitos del sistema			X				X	
Requerimientos de Proceso								
Modelos de proceso		X					X	
Actores de proceso			X				X	
Gestión y soporte de procesos					X		X	
Mejoramiento y calidad de proceso			X				X	Con nuevos requerimientos
Validación de requerimientos								
Revisión de requerimientos			X				X	
Prototipado				X			X	Los prototipos están en línea
Validación de modelos				X			X	
Aceptación de pruebas				X			X	
Análisis de Requerimientos								
Clasificación de requerimientos					X		X	
Modelo Conceptual			X			X		
Disgregación de la arquitectura, requerimientos					X		X	



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	4	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

Negociación de requerimientos		X				X	
Especificación de requerimientos							
Documento de definición del sistema (VIS)				X		X	
Especificación de requerimientos del sistema				X		X	
Especificación de requerimientos de software (ERS)				X		X	



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	5	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

3. DISEÑO DE SOFTWARE WEB:

Diseño del software	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Concurrencia			X				X	
Control y manejo de eventos			X				X	
Distribución de componentes			X				X	
Manejo de errores y excepciones y tolerancia a fallos			X				X	
Interacción y presentación			X			X		
Persistencia de datos			X			X		
Arquitectura y estructura del software								
Estilos arquitectónicos			X				X	
Patrones de diseño					X	X		Se diseño un patrón propio
Utilización de familia de programas y frameworks			X			X		
Evaluación y Análisis de la calidad del diseño del software								
Técnicas de análisis y evaluación de la calidad			X					Estándares web
Métodos y estrategias en el diseño del software								
Diseño orientado a objetos					X		X	
Diseño basado en componentes			X				X	
Otros métodos					X		X	



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	6	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

4. CONSTRUCCIÓN DE SOFTWARE WEB:

Gestión de la construcción	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Modelos de construcción			X				X	
Planeamiento de la construcción			X				X	
Medición de la construcción			X				X	
Diseño de construcción			X				X	
Pruebas de Construcción			X				X	
Reutilización			X				X	
Integración con los componentes				X		X		Portal funcionando



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema: Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.				
	Página	7	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

5. PRUEBAS DE SOFTWARE WEB:

Pruebas	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Pruebas de integración			X				X	
Pruebas de regresión		X					X	
Pruebas de rendimiento				X			X	
Pruebas de escalabilidad			X				X	
Pruebas de aceptación del Usuario Final				X			X	
Pruebas de instalación			X				X	
Pruebas de funcionalidad				X			X	
Pruebas de Stress		X					X	
Pruebas de recuperación		X					X	
Pruebas de configuración		X					X	
Pruebas de usabilidad			X				X	
Pruebas en tiempo real				X			X	



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	8	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

6. GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE WEB:

Software Configuration Management SCM	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Cronograma para SCM					X		X	
Implementación y selección de herramientas					X		X	
Plan de SCM					X		X	




	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	9	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

7. CALIDAD DEL SOFTWARE WEB:

	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Verificación de la calidad			X				X	
Aseguramiento de la calidad	X						X	
Revisiones y Auditorías		X					X	Únicamente revisiones




	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.		
	Página	10	Tesista	Director	Co-Director
	Versión	1.2.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

8. MÉTODOS Y HERRAMIENTAS DE LA INGENIERÍA WEB:

	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Herramientas para el modelado de software								
Herramientas para el modelado de requerimientos					X		x	
Herramientas para la trazabilidad de requerimientos					X		x	
Herramientas de diseño de software				X		X		
Herramientas de construcción de software								
Editores de programa			X				X	
Compiladores y generadores de código			X				X	
Interpretores	X						X	
Debuggers	X						X	
Herramientas para pruebas de software								
Generadores de pruebas					X		X	
Pruebas de Frameworks		X					X	
Herramientas de evaluación de pruebas					X		X	
Herramientas de gestión de pruebas					X		X	
Herramientas de análisis de rendimiento		X					X	
Herramientas de mantenimiento del software								
Herramientas de comprensión					X		X	



Herramientas de reingeniería					X		X		
	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA								
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.						
	Página	11	Tesista		Director		Co-Director		
	Versión	1.2.1	Luis Javier Capa León		Ing. Manuel Sucunuta		Ing. Armando Cabrera		

Herramientas para SCM (Software Configuration Management)								
Herramientas de detección y rastreo de problemas		X					X	Validadores web
Herramientas para la gestión de versionamiento		X					X	DotProject
Herramientas para liberación y construcción					X		X	
Herramientas para procesos de ingeniería de software								
Herramientas de modelado de procesos					X		X	
Herramientas de gestión de procesos					X		X	
Ambientes integrados CASE					X		X	
Herramientas para la gestión de la ingeniería de software								
Herramientas para rastreo y planificación de proyectos			X			X		DotProject
Herramientas para gestión de riesgos					X		X	
Herramientas de medición			X			X		Rankings, Optimizadores, Verificadores, Medidores Web
Herramientas para la calidad del software								
Herramientas de revisión y auditoría					X		X	
Herramientas de análisis estadístico			X			X		Google Analytics, Alexa, Emezeta, Webometrics, W3C



Para la ponderación de los datos, se toma como referencia el siguiente método de calificación, este cambio que se realiza desde los porcentajes nos permite tener una representación numérica del estado actual del sitio web, lo que facilitará la comprensión de los resultados. Hacia el final del capítulo los porcentajes son los que reflejan dicho estado.

Campo	Valor
No	0
25%	1
50%	2
75%	3
100%	4

Aplicando estos valores, los resultados obtenidos son los siguientes:

- ✓ Para la primera área de evaluación Gestión de la Ingeniería del Software vemos que de los 60 puntos máximos posibles de alcanzar el Sitio cuenta con 34 y únicamente existe documentación de los campos: Procesos para la revisión de requerimientos, determinación de entregables y estimación de costos y cronogramas.
- ✓ La segunda área de Requerimientos de Software indica que de los posibles 76 puntos a alcanzar, el sitio cuenta con 43. Existe documentación de los requerimientos funcionales y no funcionales y el modelo conceptual.
- ✓ En Diseño del Software el sitio web tiene una calificación de 30 sobre 52 puntos posibles alcanzables. Existe la documentación respectiva de interacción y presentación, persistencia de datos, patrones de diseño y utilización de frameworks.
- ✓ La cuarta área de Construcción del Software tiene un máximo de puntos de 28 de los cuales cuenta con 22 y con el portal funcionando como único respaldo de la integración de componentes.
- ✓ Pruebas del Software tiene 36 puntos de los 48 máximos posibles para esta área y no existe documentación alguna que respalde la ejecución de estos campos.



- ✓ Gestión de la Configuración del Software no fue realizada en su totalidad durante la construcción del sitio web de la UTPL.
- ✓ El área de Calidad de Software tiene 6 puntos de los 12 totales y no existe la documentación que respalde la ejecución de sus campos.
- ✓ La última área comprende Métodos y herramientas de la Ingeniería Web, aquí el sitio web cuenta con 29 puntos de los 100 máximos que se pueden alcanzar. Existe la documentación únicamente de la utilización de herramientas de diseño.

Para el porcentaje de cumplimiento en cada etapa, se contabilizó cada ítem en su respectiva categoría, asignando el valor detallado en la tabla anterior sea para 25, 50 75 y 100%. A continuación se muestra una conversión de puntajes de cada área todos sobre 100 puntos y un resultado final de la evaluación sobre un 100%.

Área	Total
Gestión de la Ingeniería del Software	56,67
Requerimientos de Software	56,58
Diseño del Software	57,69
Construcción del Software	78,57
Pruebas del Software	75,00
Gestión de Configuración del Software	00,00
Calidad del Software	50,00
Métodos y herramientas de la IW	29,00

Estos valores representan el total de cumplimiento de cada requisito en su respectiva categoría considerando si existe o no la documentación para respaldar el cumplimiento del campo evaluado. En el anexo 6 se detallan los estándares IEEE, ISO, etc., que se aplican en cada área evaluada. [Swebok, 2004]. Una “X” indica que el estándar se aplica en todas las áreas, una “P” señala que se aplica con prioridad para el área indicada y una “S” indica que se puede aplicar es estándar en dicha área de manera secundaria o a cierto campos.

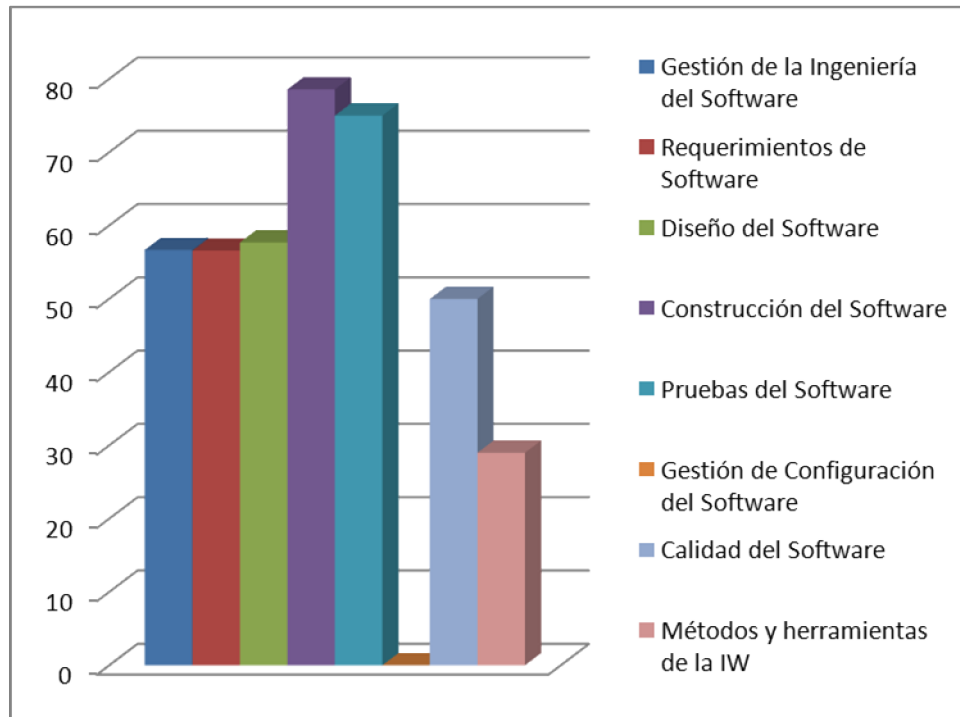


Figura 29: Resultados obtenidos sobre el sitio web de la UTPL en una escala sobre 100.

Resultado final: 50.44%. Lo que nos indica que de todas las actividades que debieron haberse realizado en el proceso de desarrollo del sitio web, se ha cumplido únicamente con el 50.44%. Las recomendaciones se encuentran al final del documento, en donde se sugieren algunas pautas sobre este análisis.

Sobre la documentación existente, a continuación se muestra una tabla y un gráfico con información de la evidencia física de la documentación.

ETAPA	SI	NO	Total
Gestión de la Ingeniería del Software Web	3	15	18
Requerimientos de Software Web	2	18	20
Diseño de Software Web	4	8	12
Construcción del Software Web	1	6	7
Pruebas del Software Web	0	12	12
Gestión de la Configuración del Software	0	3	3
Calidad del Software Web	0	3	3
Métodos y Herramientas de la IW	4	21	25

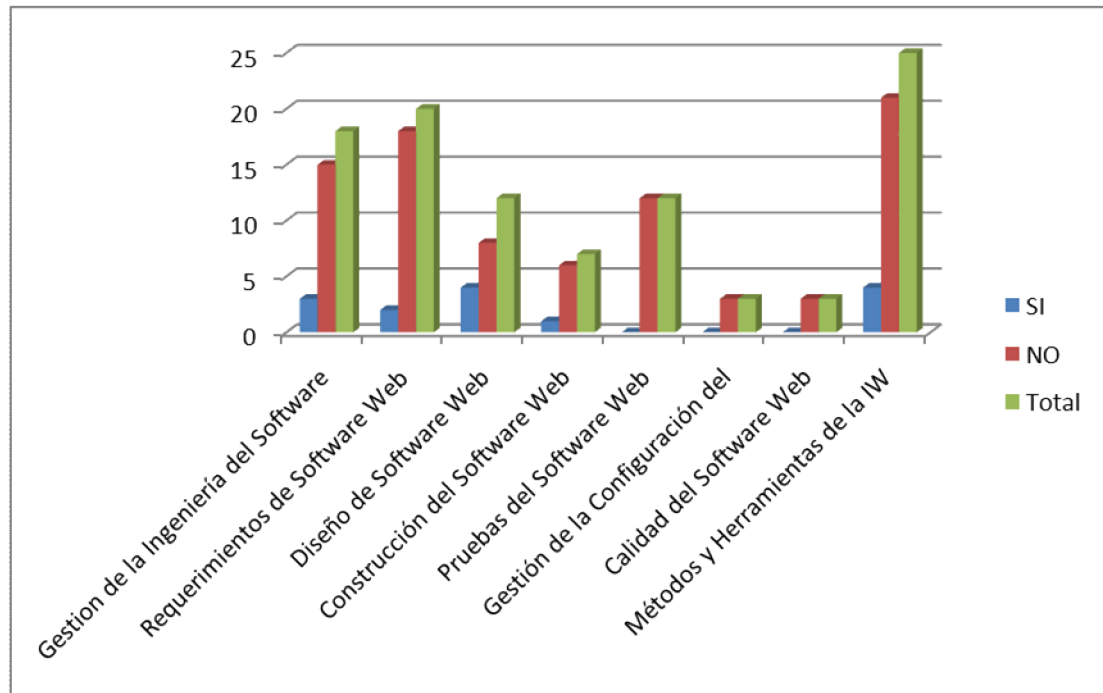


Figura 30: Representación de la documentación existente

5.2 Evaluación de las métricas propuestas.

Para el presente análisis de las métricas descritas en el capítulo anterior, hemos utilizado la herramienta LinkScan 12.0, la cual por sus prestaciones ha sido la herramienta que más se ajusta a las necesidades de evaluación del sitio. La herramienta cuenta con una licencia para usuario no comercial de 15 días sin que ello afecta las funcionalidades del programa. Evalúa los principales aspectos de la usabilidad sugeridos en la presente investigación y detallados en el anexo 6, en dónde se presentan los estándares de la IEEE en los que están basadas las plantillas que más adelante se utilizarán en la evaluación del sitio web de la UTPL. A la fecha de la evaluación, 11 de marzo de 2009, los resultados que se obtuvieron son los siguientes de los muchos posibles como también o muestra otra figura:

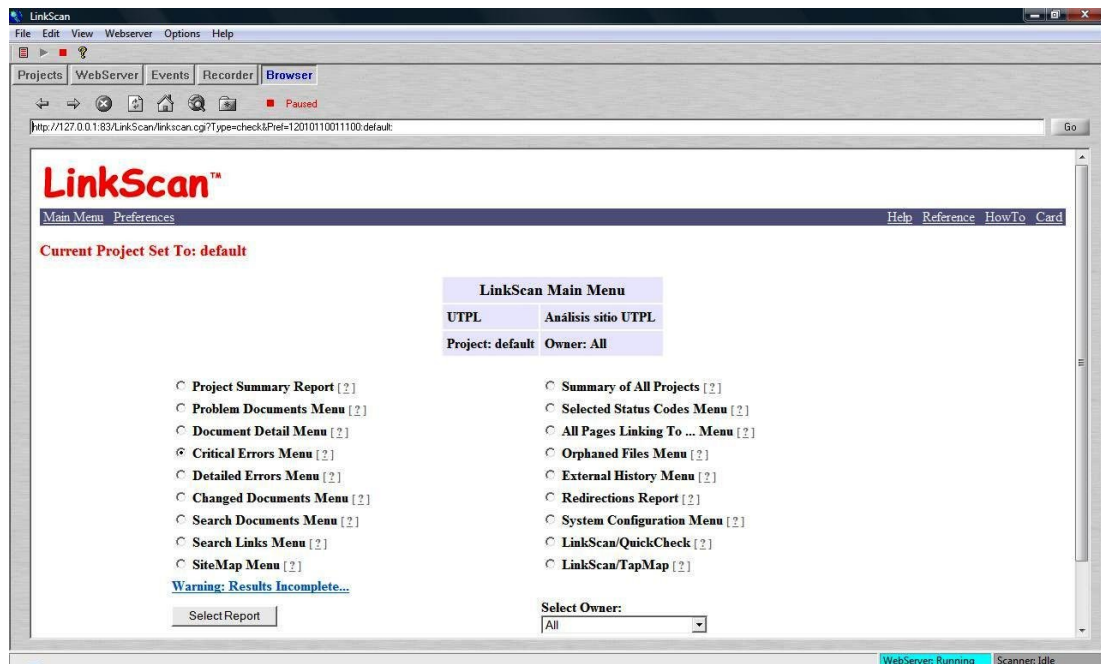


Figura 31: Opciones de LinkScan para presentar informes

Summary of Objects Tested and Errors Found			
Documents:	Total	Missing	With Errors
	60528	10519	25879
Other Files:	Total	Missing	
	48676	573	
Anchors:	Total	Broken	
	19463	22836	
Total Links:	Total	Broken	Suspect
Internal Links	2736643	132694	472761
External Links	262554	13114	239016
Unique Links:	Total	Broken	Suspect
Internal Links	304202	34032	153803
External Links	13141	1812	10650
Weighted Errors:	4.48	per document	

El nombre del proyecto para el análisis del sitio es default, se encontraron 60528 documentos entre imágenes, archivos de texto, etc., de los cuales 25879 contienen errores, 10519 están perdidos. El número total de links



internos es de 2736643 de los cuales están rotos 132694 y sospechosos de estarlo 472761. La cantidad de links externos asciende a 262554 con 13114 links rotos y 239016 links sospechosos. Los demás valores nos señalan los mismos campos ya mencionados pero a nivel de links únicos o exclusivos del sitio así como un grado de error del 4.48 por cada documento. A continuación, la imagen 7.1 muestra un ejemplo del informe de links en mal estado, así como su causa y su posible acción para corregirlo.

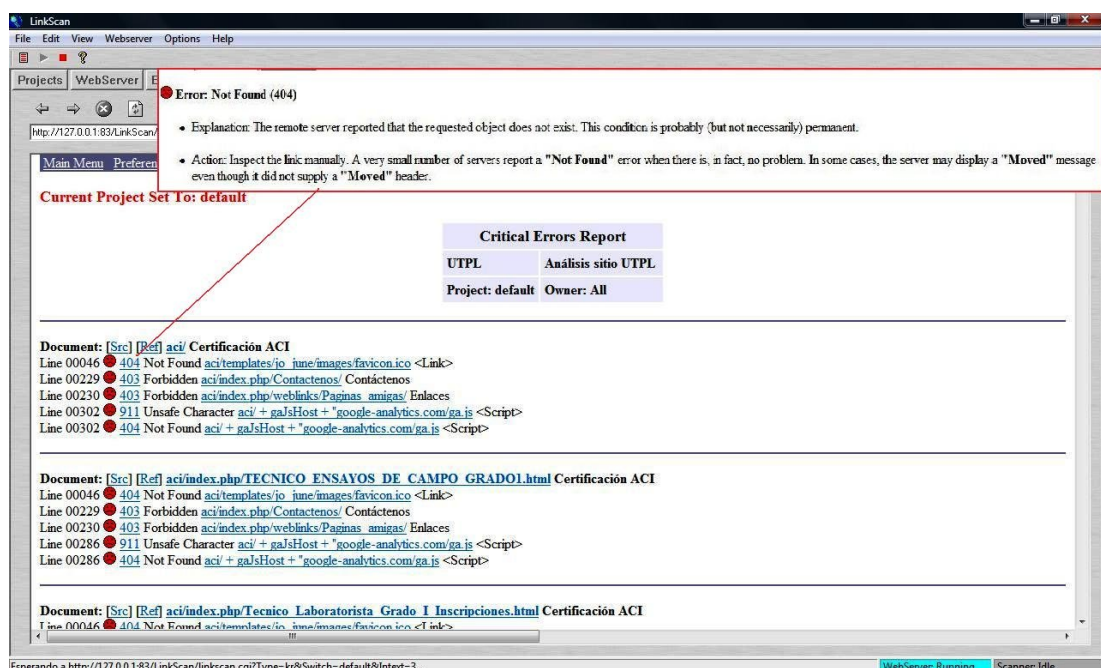


Figura 32: Ejemplo errores encontrados, causa y solución

Las siguientes tablas nos muestran estadísticas de los errores encontrados, agrupados por tipos tanto a nivel interno como externo al sitio.

Summary of Internal Errors Found by Type				
Total Links	Unique	Status	Code	Description
100	23	🤔Unknown:	0	No Status
188	22	😞Error:	18	Invalid Scheme
18832	480	😞Error:	301	Moved Permanently
6085	2986	😞Error:	400	Network/Server Error
17467	562	😞Error:	403	Forbidden



26970	6589	☹️Error:	404	Not Found
2976	114	☹️Error:	405	Method Not Allowed
56156	22836	☹️Error:	6	Bad Anchor
221	1	☹️Error:	907	Multiple Redirections
3799	442	☹️Error:	911	Unsafe Character
322	22	😞Possible Error:	12	Invalid Mailto Link
18	4	😞Possible Error:	500	Server Error
110	8	😞Possible Error:	900	No DNS Entry
116345	624	😞Possible Error:	902	Connect Error
6	6	😞Possible Error:	930	Timeout Header
63	5	😞Possible Error:	931	Timeout Body
9	5	😞Possible Error:	932	Timeout Unknown
65527	275	⚠️Warning:	904	Missing /
1794	31	⚠️Warning:	910	Location not Absolute
24	2	⚠️Warning:	914	Meta Redirect
51313	726	ℹ️Advisory:	302	Moved Temporarily
237230	152095	ℹ️Advisory:	803	Skipped - CGI Limit
2342	509	😊No Error:	11	Valid Mailto Link
2108804	96372	😊No Error:	200	URL
19942	19463	😊No Error:	5	Anchor

Summary of External Errors Found by Type				
Total Links	Unique	Status	Code	Description
12949	1741	☹️Error:	18	Invalid Scheme
54	53	☹️Error:	403	Forbidden
35	16	☹️Error:	404	Not Found
76	2	☹️Error:	991	SSL Error
2	2	😞Possible Error:	503	Service Unavailable
76566	1931	😞Possible Error:	801	Skipped - Bad Server
2417	336	😞Possible Error:	900	No DNS Entry



[159807](#) [8360](#) 😞 Possible Error: [902](#) Connect Error
[5](#) [2](#) ⚠ Warning: [401](#) Unauthorized
[2](#) [1](#) ⚠ Warning: [906](#) Contains an IP Address
[217](#) [18](#) ⚠ Warning: [908](#) Missing /
[10424](#) [679](#) 😊 No Error: [200](#) URL

Esta misma evaluación se efectuó sobre sitios de instituciones educativas como la Universidad Nacional de Loja y la Escuela Superior Politécnica del Litoral, que son instituciones educativas y se obtuvieron los siguientes resultados sobre los mismos parámetros:

Universidad Nacional de Loja

Summary of Objects Tested and Errors Found			
Documents:	Total	Missing	With Errors
	837	14	22
Other Files:	Total	Missing	
	302	0	
Anchor:	Total	Broken	
	0	3	
Total Links:	Total	Broken	Suspect
Internal Links	39730	44	3691
External Links	1650	14	766
Unique Links:	Total	Broken	Suspect
Internal Links	2102	17	286
External Links	72	7	12
Weighted Errors:	0.64	per document	

Podemos determinar un promedio de error de 0.64 por documento.



Escuela Superior Politécnica del Litoral:

Summary of Objects Tested and Errors Found			
Documents:	Total	Missing	With Errors
	36745	17208	17116
Other Files:	Total	Missing	
	7150	4	
Anchor:	Total	Broken	
	148	8	
Total Links:	Total	Broken	Suspect
Internal Links	401485	35331	1511
External Links	133284	16611	50334
Unique Links:	Total	Broken	Suspect
Internal Links	44360	17297	380
External Links	431	111	154
Weighted Errors:	1.56	per document	

Podemos determinar un promedio de error de 1.56 por documento.

A continuación se muestra una comparativa entre los resultados obtenidos por las tres evaluaciones a los sitios web antes mencionados, primero en cuanto al total de documentos en cada sitio, documentos perdidos y documentos con errores. Luego se compara en los 3 sitios evaluados la cantidad enlaces internos, la cantidad de enlaces internos rotos y sospechosos para finalmente comparar los datos obtenidos en cuanto a enlaces externos.

Cantidad de documentos, documentos perdidos y con errores:

Sitio	Documentos	Perdidos	Con errores	%Errores
UTPL	60528	10519	25879	4.48
UNL	837	14	22	0.64
ESPOL	36745	17208	17116	1.56

Tabla 5: Comparación de documentos y sus errores externos en los sitios web de las 3 Universidades

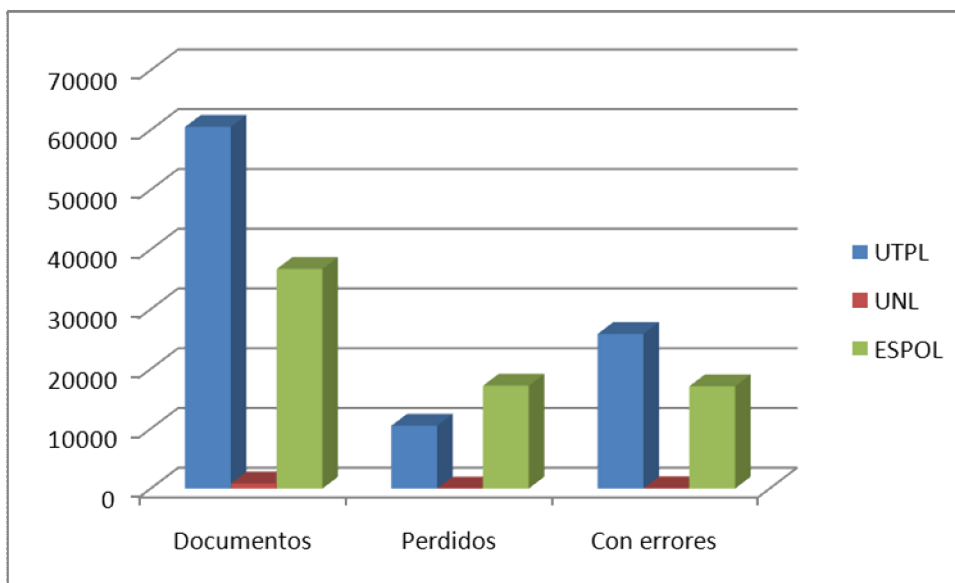


Figura 33: Gráfica de documentos y sus errores externos en los sitios web de las 3 Universidades

Cantidad de enlaces internos, rotos y sospechosos:

Sitio	Enlaces	Rotos	Sospechosos	%Mal Estado
UTPL	2736643	132694	472761	22.12
UNL	39730	44	3691	9.40
ESPOL	401485	35331	1511	9.17

Tabla 6: Comparación de enlaces totales, rotos y sospechosos en los sitios web de las 3 Universidades

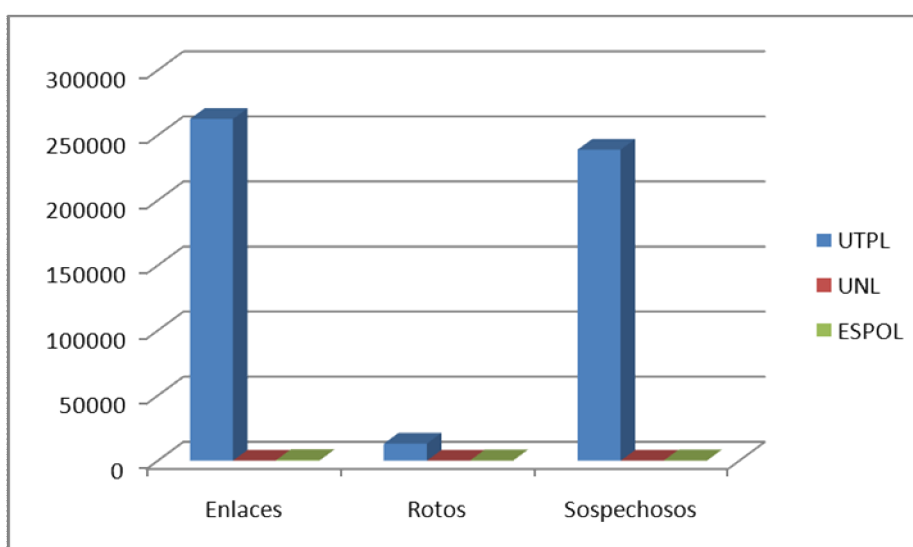


Figura 5.6: Gráfica de enlaces totales, rotos y sospechosos en los sitios web de las 3 Universidades.



Cantidad de enlaces externos, rotos y sospechosos:

Sitio	Enlaces	Rotos	Sospechosos	%Mal Estado
UTPL	262554	13114	239016	96.02
UNL	72	7	12	26.38
ESPOL	431	111	154	61.48

Tabla 7: Comparación de enlaces totales internos, rotos y sospechosos en los sitios web de las 3 Universidades

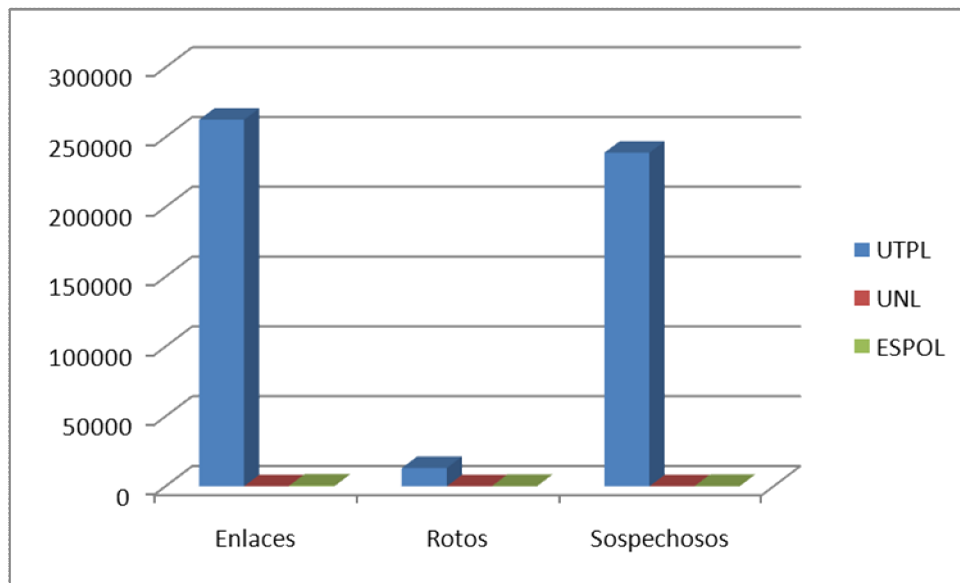


Figura 34: Gráfica de enlaces totales internos, rotos y sospechosos en los sitios web de las 3 Universidades



6 CONCLUSIONES

1. A través de la recopilación de información de diferentes fuentes, se analizaron las diferentes técnicas, métodos que se utilizan con más frecuencia en la construcción de sitios web así como de sus principales características, de los cuales la metodología AWE es la más adecuada para ser implementada en un desarrollo web independiente del contexto en el que vaya a funcionar.
2. Para evaluar un sitio web, es necesario adaptar plantillas para determinar elementos básicos que deben ser implementados en el análisis, diseño e implementación de un software web. Estas plantillas deben ser diseñadas para determinar la usabilidad del sitio.
3. Cada metodología de desarrollo de aplicaciones web tiene su propio proceso para ser llevada a cabo, pero en sí comparten los mismos pasos básicos que pretenden maximizar la relación costo beneficio al momento de implementar un proyecto de desarrollo de sitios web.
4. A través del empleo de la metodología Web-site QEM, el evaluador se adecúa, bajo un enfoque sistemático, disciplinado y cuantitativo; a la evaluación, comparación y análisis de la calidad de aplicaciones web.
5. Cuando se automatiza el proceso de medición de métricas a través de un software, el usuario evaluador o desarrollador puede determinar los errores que se encuentren debido a la gran cantidad de información que maneja el sitio y le ayuda en tareas tan tediosas como la recolección de datos de atributos de artefactos Web como así también del cálculo automático de sus valores.
6. El campo de evaluación de artefactos web es relativamente nuevo y apenas está siendo considerado. El enfoque propuesto, es esencialmente integral, flexible y robusto, y cubre la mayor parte de las actividades en el proceso de evaluación, comparación, y selección de artefactos Web aplicados, para nuestro caso, en la UTPL.



7. El sitio web de la UTPL, utiliza una metodología parcial para su desarrollo, lo cual se expresa en la evaluación de las métricas en comparación con sitios semejantes de otras universidades.
8. La evaluación realizada al sitio web de la UTPL refleja que para el análisis, diseño e implementación no se siguieron los pasos que se recomiendan en esta investigación en su mayor parte o simplemente algunos de ellos.
9. No existe la respectiva documentación que respalde las fases a las que se ha dado cumplimiento durante el desarrollo del sitio web de la UTPL, la documentación existente no es completa porque no es posible efectuar la trazabilidad entre los requerimientos y lo que realmente se ha efectuado.
10. Debido a la gran cantidad de páginas que dependen del sitio de la UTPL, su análisis y su tendencia a errores es más alto que sitios de otras universidades bajo las mismas condiciones.
11. El proceso de ingeniería web se cumple en un 50.44% en base a una metodología que permita gestionar de mejor manera tanto los resultados como la documentación obtenida al final de cada fase. Este resultado permite llegar a la conclusión de que en la construcción del sitio web se siguió una metodología de desarrollo ágil por partes, existiendo la documentación respectiva en algunas de ellas solamente.



7 RECOMENDACIONES

1. Para el análisis, diseño e implementación se deben cumplir como requisitos mínimos los que se sugieren en el anexo 1, fueron adaptados de Swebok y sirven como guía para las aplicaciones web.
2. Cuando se realiza una fase en la construcción de una aplicación web, se debe tener constancia mediante una documentación respectiva sobre lo que se está desarrollando con la finalidad de respaldar y validar lo que se ha hecho.
3. Se recomienda que la metodología que se seleccione debe estar enfocada a los objetivos que se pretenden sean satisfechos por la aplicación web y a los recursos de los cuáles se disponga.
4. Para el sitio web de la UTPL se puede realizar la documentación de algunas de las áreas propuestas en el anexo 1 para respaldar lo que se ha hecho en cada etapa.
5. Se puede considerar como un proyecto de tesis futuro, el diseño e implementación de un sistema para automatizar las tareas de evaluación de un sitio web mediante métricas como un futuro avance de esta investigación.
6. Para mejorar la evaluación de las métricas planteadas anteriormente, se debería realizar una depuración de los enlaces que la herramienta ha determinado como erróneos e implantar la solución recomendada. La mayoría de los errores corresponde a sitios que ya no se encuentran disponibles.
7. Se pueden utilizar los resultados de los errores encontrados por LinkScan™ para corregir el sitio web de la UTPL, puesto que la herramienta sugiere la acción a tomar en cada caso. Toda la información está disponible en una base de datos que puede ser explorada por el mismo programa.



8 BIBLIOGRAFIA

1. [ALGHAMDI, 2009]: Alghamdi, Abdullah S. Accessing and customizing and agile AWEM environment for a sustainable software development. College of Computer and Information Sciences, King Saud University, Saudi Arabia. 2009.
2. [Baresi, 2001]: BARESI L., Garzotto F., Paolini P. Extending UML for Modelling Web Applications. In proceedings of the 34th annual Hawaii International Conference on System Science. IEEE Computer Society. 2001
3. [Berzal, 2003]: BERZAL, Francisco; CORTIJO, Francisco; CUBERO, Juan Carlos; Aplicaciones Web, 2003.
4. [Bieber, 1998]: BIEBER M., Galnares, R., Lu, Q. (1998). Web engineering and flexible hypermedia. The Second Workshop on Adaptive Hypertext and Hypermedia Hipertext.
5. [Bozzon, 2004]: BOZZON, Alejandro; COMAI, Sara. Conceptual Modeling and Code Generation for Rich Internet Applications, 2004.
6. [Brinck, 2002]: BRINCK, Tom; GERGLE, Darren, WOOD, Scott; Usability for the web, 2002
7. [Chile, 2000]: GOBIERNO DE CHILE, Guía para el desarrollo de sitios web, disponible en: <http://www.guiaweb.gob.ec>
8. [Cueva, 2004]: CUEVA, J.M.; Métricas de usabilidad en la web; 2004.
9. [Dávila, 2003]: DAVILA, Leticia; MEJIA, Pedro; Evaluación de la Calidad de Software en Sistemas de Información en Internet; 2003.
- 10.[Eguíluz, 2008]: EGUÍLUZ, Javier; SÁNCHEZ, Miguel; Usabilidad Web en la práctica, 2008.
- 11.[Escalona, 2002]: ESCALONA, M. J.; KOCH, N, Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo, 2002.
- 12.[Escalona, 2002]: ESCALONA, M. J.; KOCH, N, The Expressive Power of UML-based Web Engineering, 2002.



- 13.[EscalonaMJ, 2002]: ESCALONA, M.J., Torres, J., Mejías, M. Requirements capture workflow in Global Information Systems. Proceedings of OOIS. Springer-Verlag. Montpellier, France. 2002
- 14.[Fielding, 2000]: FIELDING, Roy Thomas; Architectural Styles and the Design of Network-based Software Architectures, 2000.
- 15.[Fons, 2002]: FONS, Joan; PASTOR, Oscar; VALDERAS, Pedro; RUIZ, Marta; OOWS: Un Método de Producción de Software en Ambientes Web; 2002.
- 16.[Garzotto, 2007]: Garzotto, F., & Perrone, V. Industrial acceptability of design methods: an empirical study. Journal of Web Engineering. 2007.
- 17.[IEEE, 1990]: IEEE Standard Glossary of Software Engineering Terminology, IEEE std 610.12, 1990.
- 18.[Instone, 1997]: INSTONE, Keith; Site Usability Heuristics for the Web.
- 19.[Kappel, 2003]: KAPPEL, Gerti; BIRGIT Pröll; SIEGFRIED Reich; WERNER Retschitzegger; Web Engineering: The Discipline of Systematic Development of Web Applications, 2003.
- 20.[LAFUENTE, 2000]: LAFUENTE, Guillermo Javier, Automatizando Métricas en la Web, Universidad Nacional de Luján, 2000.
- 21.[Lee, 1998]: LEE, H., Lee, C., Yoo, C. A Scenario-based object-oriented methodology for developing hypermedia information systems. Procesings of Annual Conference on Systems Science. 1998.
- 22.[Lowe, 2002]: Lowe D., Eklund J. Client Needs and the Design Process in Web Projects . WWW2002 Web Engineering Track. 2002.
- 23.[Manchón, 2009]: Manchón, Eduardo. Desarrollo Ágil. Disponible en: http://www.alzado.org/articulo.php?id_art=602
- 24.[Mateu, 2004]: MATEU, Carlos; Desarrollo de Aplicaciones Web; 2004.
- 25.[McDonald, 2001]: McDONALD, Andrew; WELLAND, Ray; Agile Web Engineering (AWE) Process, 2001.
- 26.[Mendes, 2008]: XAVIER MENDES, Maria Emilia ; MENDES, Emilia, MOSLEY, Nile, Web Engineering: Theory and Practice of Metrics and Measurement for Web Development. Disponible en:



http://books.google.com.ec/books?id=kFfPae14AWgC&printsec=frontcover&dq=web+engineering+metrics&source=gbs_summary_r&cad=0#PPP1,M1

- 27.[Molinares, 2002]: MOLINARES, Daladier Jabba; DÍAS, José Márquez, Herramientas para Consulta y Modelado en la Web, una forma diferente del manejo de grandes volúmenes de información de los Web Sites en Internet. 2002.
- 28.[Moner David, 2002]: MONER, David; SABATÉ, Jordi; Usabilidad: El Guión Multimedia, 2002.
- 29.[Moner, 2002]: MONER CANO, David; SABATÉ ALSINA, Jordi, Usabilidad.- El guión multimedia, 2002.
- 30.[Montero, 2006]: MONTERO, Susana; DIAZ, Paloma; AEDO, Ignacio; MONTELLS, Laura; ADM: método de diseño para la generación de prototipos web rápidos a partir de modelos; 2006.
- 31.[Moreno, 2004]: MORENO, Ana; SANCHEZ, Maribel; Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico, 2004.
- 32.[Murray, 1999]: MURRAY, George, COSTANZO, Tania; Usability and the Web: An Overview; disponible en: <http://www.nlc-bnc.ca/9/1/p1-260-e.html>, 1999.
33. [Nielsen, 1990]: NIELSEN, Jacob; Ten Usability Heuristics; disponible en: http://www.useit.com/papers/heuristic/heuristic_list.html [Useit.com](http://www.useit.com), 1990
- 34.[Olsina - Rossi, 2001]: OLSINA, Luis; ROSSI, Gustavo, A Quantitative Method for Quality Evaluation of Web Sites and Applications, 2001.
- 35.[Olsina, 1998]: OLSINA, L. Building a Web-based information system applying the hypermedia flexible process modeling strategy. International Workshop on Hypermedia Development, Hypertext 1998.
- 36.[Olsina, 1999]: OLSINA, Luis Antonio; Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web, 1999.
- 37.[Schwabe, 2001]: SCHWABE, Daniel; ROSSI, Gustavo, An Object Oriented Approach to Web-Based Application Design, 2001.



- 38.[Swebok, 2004]: SWEBOK, IEEE Computer Society, Guide to the Software Engineering Body of Knowledge, 2004, disponible en: <http://www.swebok.org>
- 39.[Torres Bosh, 2008]: TORRES BOSH, Victoria; A Web Engineering Approach for the Development of Business-Process Driven Web Applications, Julio 2008.
40. [Nielsen, 2001]: NIELSEN, Jacob; Heuristic evaluation; disponible en: <http://www.useit.com/jakob/inspectbook.html>
- 41.[UWA, 2000]: UWA; Requirements Elicitation: Model, Notation and Tool Architecture. Disponible en: <http://www.uwaproject.com>



9 DIRECCIONES ELECTRÓNICAS

1. Seguridad en la Aplicaciones Web; disponible en:
<http://msdn2.microsoft.com/es-es/library/f13d73y6.aspx>
2. Seguridad: Realidad o Mito; disponible en:
<http://kodegeek.com/2006/04/seguridad-de-aplicaciones-web-mito-o.shtml>
3. Métricas del software; disponible en:
http://www.acoutin.com/esp/software_web_metrica.htm
4. Usabilidad; disponible en:
<http://iteso.mx/~carlosc/pagina/documentos/usabilidad.htm>
5. CEBALLOS, Web 2.0, ¿revolución o evolución de internet?, disponible en:
<http://www.virtual.unal.edu.co/unvPortal/articles/ArticlesViewer.do?reqCode=viewDetails&idArticle=2>




ANEXOS



ANEXO 1



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>		
	Página	1	Tesista	Director	Co-Director
	Versión	1.2.1			

1. GESTIÓN DE LA INGENIERÍA DEL SOFTWARE:


	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Determinación del Alcance e Iniciación								
Determinación y negociación de requerimientos								
Análisis de Factibilidad								
Procesos para la revisión de requerimientos								
Planificación del Proyecto Software								
Planeamiento de procesos								
Determinación de entregables								
Estimación de Costos, esfuerzo y cronogramas								
Distribución de recursos								
Manejo de riesgos								
Gestión de calidad								
Manejo de Planes								
Establecimiento del Proyecto								
Implementación de planes								
Gestión de la información de los proveedores								
Implementación de medidas a los procesos								
Monitoreo de procesos								
Control de procesos								
Reportes								



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA						
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>				
	Página	2	Tesista	Director			Co-Director
	Versión	1.2.1					

Revisión y Evaluación							
Determinación de la satisfacción de los requerimientos							
Revisión y evaluación del desempeño							




	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>		
	Página	3	Tesista	Director	Co-Director
	Versión	1.2.1			

2. REQUERIMIENTOS DE SOFTWARE:

Actividades Básicas Requerimientos de Software	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Definición de requerimientos de software								
Requerimientos de producto y de proceso								
Requerimientos funcionales y no funcionales								
Requerimientos cuantificables								
Requerimientos de software y requisitos del sistema								
Requerimientos de Proceso								
Modelos de proceso								
Actores de proceso								
Gestión y soporte de procesos								
Mejoramiento y calidad de proceso								
Validación de requerimientos								
Revisión de requerimientos								
Prototipado								
Validación de modelos								
Aceptación de pruebas								
Análisis de Requerimientos								
Clasificación de requerimientos								
Modelo Conceptual								




Disgregación de la arquitectura, requerimientos							
---	--	--	--	--	--	--	--

	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA						
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>				
	Página	4	Tesista	Director			Co-Director
	Versión	1.2.1					

Negociación de requerimientos							
Especificación de requerimientos							
Documento de definición del sistema (VIS)							
Especificación de requerimientos del sistema							
Especificación de requerimientos de software (ERS)							



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>		
	Página	5	Tesista	Director	Co-Director
	Versión	1.2.1			

3. DISEÑO DE SOFTWARE:

Diseño del software	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Concurrencia								
Control y manejo de eventos								
Distribución de componentes								
Manejo de errores y excepciones y tolerancia a fallos								
Interacción y presentación								
Persistencia de datos								
Arquitectura y estructura del software								
Estilos arquitectónicos								
Patrones de diseño								
Utilización de familia de programas y frameworks								
Evaluación y Análisis de la calidad del diseño del software								
Técnicas de análisis y evaluación de la calidad								
Métodos y estrategias en el diseño del software								
Diseño orientado a objetos								
Diseño basado en componentes								
Otros métodos								



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>		
	Página	6	Tesista	Director	Co-Director
	Versión	1.2.1			

4. CONSTRUCCIÓN DE SOFTWARE:

Gestión de la construcción	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Modelos de construcción								
Planeamiento de la construcción								
Medición de la construcción								
Diseño de construcción								
Pruebas de Construcción								
Reutilización								
Integración con los componentes								




	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Platilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de la UTPL.				
	Tema:				
	Página	7	Tesista	Director	Co-Director
Versión	1.2.1				

5. PRUEBAS DE SOFTWARE:

Pruebas	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Pruebas de integración								
Pruebas de regresión								
Pruebas de rendimiento								
Pruebas de escalabilidad								
Pruebas de aceptación del Usuario Final								
Pruebas de instalación								
Pruebas de funcionalidad								
Pruebas de Stress								
Pruebas de recuperación								
Pruebas de configuración								
Pruebas de usabilidad								
Pruebas en tiempo real								



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>		
	Página	8	Tesista	Director	Co-Director
	Versión	1.2.1			

6. GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE:

Software Configuration Management SCM	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Cronograma para SCM								
Implementación y selección de herramientas								
Plan de SCM								



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>.		
	Página	9	Tesista	Director	Co-Director
	Versión	1.2.1			

7. CALIDAD DEL SOFTWARE

	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Verificación de la calidad								
Aseguramiento de la calidad								
Revisiones y Auditorías								



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>.				
	Tema:				
	Página	10	Tesista	Director	Co-Director
	Versión	1.2.1			

8. MÉTODOS Y HERRAMIENTAS DE LA INGENIERÍA WEB:

	25%	50%	75%	100%	NO	DOC		OBSERVACIONES
						Si	No	
Herramientas para el modelado de software								
Herramientas para el modelado de requerimientos								
Herramientas para la trazabilidad de requerimientos								
Herramientas de diseño de software								
Herramientas de construcción de software								
Editores de programa								
Compiladores y generadores de código								
Interpretores								
Debuggers								
Herramientas para pruebas de software								
Generadores de pruebas								
Pruebas de Frameworks								
Herramientas de evaluación de pruebas								
Herramientas de gestión de pruebas								
Herramientas de análisis de rendimiento								
Herramientas de mantenimiento del software								
Herramientas de comprensión								




Herramientas de reingeniería									
	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA								
	Tema:		Plantilla para la obtención de datos sobre el Análisis, Diseño, Construcción e Implementación del Sitio Web de <nombre sitio>.						
	Página	11	Tesista		Director		Co-Director		
	Versión	1.2.1							
Herramientas para SCM (Software Configuration Management)									
Herramientas de detección y rastreo de problemas									
Herramientas para la gestión de versionamiento									
Herramientas para liberación y construcción									
Herramientas para procesos de ingeniería de software									
Herramientas de modelado de procesos									
Herramientas de gestión de procesos									
Ambientes integrados CASE									
Herramientas para la gestión de la ingeniería de software									
Herramientas para rastreo y planificación de proyectos									
Herramientas para gestión de riesgos									
Herramientas de medición									
Herramientas para la calidad del software									
Herramientas de revisión y auditoría									
Herramientas de análisis estadístico									




ANEXO 2



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para la obtención de datos del Sitio Web de la UTPL.		
	Página	1	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera


PREGUNTA	Si	No	NA	OBSERVACIONES
¿Se indica con claridad qué persona o empresa es responsable de los contenidos del sitio?				
Si el sitio es un sub-sitio de una Organización, ¿se indica el logo o nombre de dicha Organización?				
¿Se indica el modo de contactar con la organización o persona responsable del sitio?				
¿Se indica la persona o empresa identificada como responsable del sitio?				
¿Se indica el Copyright del sitio?				
¿Se mencionan recomendaciones o ratings de organizaciones externas?				
¿Se indican los nombres y autoridad de las personas responsables de la Organización?				
¿Se indica si la Organización tiene presencia fuera de la Web?				
¿Se indica si el sitio es patrocinado por una Organización local, nacional o internacional?				
¿Se indica la existencia y modo de obtención de materiales impresos producidos por la Organización?				
¿Se suministra una descripción completa de la naturaleza de la Organización?				
¿Se indica desde cuándo existe la Organización?				
¿Hay una lista de los empleados más significativos con indicación de sus competencias?				
¿Se indica si la Organización es auditada por una agencia independiente de control?				
¿Se suministra información financiera de la Organización?				



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA					
	Tema:		Plantilla para la obtención de datos del Sitio Web de la UTPL (Páginas secundarias).		
	Página	3	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera


PREGUNTA	Si	No	NA	OBSERVACIONES
¿Está claro qué organización es responsable del contenido de la página?				
¿Se indica con claridad el nombre del autor (departamento/centro) de la página?				
¿Se indica la autoridad de quienes suministran la información (autor/departamento/centro)?				
¿Se indica el modo de contactar con el autor (departamento/centro)?				
¿Se indica el Copyright de los materiales incluidos en la página?				
¿Ha sido aprobada oficialmente la página por quienes tienen la responsabilidad sobre el sitio?				
¿Se indica con claridad si el material de una página ha sido tomado de una fuente de información externa?				



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA					
	Tema:	Plantilla para medir la exactitud de la información del Sitio Web de la UTPL.			
	Página	4	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
¿La información está libre de errores gramaticales, ortográficos y tipográficos?				
¿Se indican las fuentes de la información fáctica para que puedan comprobarse los datos con el original?				
Si el trabajo es investigación original del autor, ¿está claramente indicado?				
Si se incluyen gráficas, cuadros o tablas, ¿están claramente tituladas y son fáciles de leer?				
¿Hay indicación de que la información ha sido revisada por un editor para comprobar su exactitud?				



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para medir la objetividad de la información del Sitio Web de la UTPL.		
	Página	5	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
¿Es evidente el punto de vista de la Organización?				
¿Es evidente el punto de vista del autor (departamento/centro)?				
¿Está la página libre de publicidad?				
¿Está clara la relación existente entre la Organización los anunciantes representados en la página?				
¿Existe una clara diferenciación entre información y publicidad en la página?				
Si hay contenidos informativos referidos a productos o servicios ajenos ¿está claro por qué se incluyen en la página?				
¿Se incluye una descripción de los objetivos de la Organización al suministrar la información (mission statement)?				
¿Existe una clara identificación del material editorial y de opinión?				
¿Existe una distinción clara entre expresiones de opinión sobre un asunto y contenidos informativos que se pretende sean objetivos?				
¿Existe una clara distinción entre contenidos informativos y de entretenimiento en la página?				
¿Existe una clara distinción entre contenidos publicitarios y de entretenimiento en la página?				



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para medir la actualización y cobertura de la información del Sitio Web de la UTPL.		
	Página	6	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
¿Se indica en la página la fecha de creación del material?				
¿Se indica en la página la fecha de publicación en el servidor del material?				
Si el material ha sido revisado, ¿se incluye en la página la fecha de la última revisión?				
Para evitar confusiones, ¿se indican todas las fechas en formatos internacionalmente conocidos (día mes año)?				
Si la página incluye información de actualidad, ¿se indica la frecuencia de actualización?				
Si la página incluye datos estadísticos, ¿se indica con claridad la fecha de recogida de datos, o se incluye un enlace a los datos originales?				

PREGUNTA	Si	No	NA	OBSERVACIONES
¿Está claro que materiales se incluyen en el sitio?				
Si la página está aún en construcción, ¿se indica la fecha estimada de finalización?				
Existe un equivalente impreso a la página o el sitio, si así fuera, ¿se indica claramente si se trata de una versión completa o qué partes del documento original se han omitido?				
¿Existe un equivalente impreso al sitio?, si así fuera, ¿se indica claramente si la versión web incluye información adicional no contenida en la versión impresa?				




	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para medir los recursos de navegación, título, enlaces, URL, mapa de sitio y buscador interno del Sitio Web de la UTPL.		
	Página	7	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
¿Indica el título para el navegador que la Organización es responsable de los contenidos del sitio?				
¿Indica el título para el navegador que se trata de la página principal?				
¿Es breve el título para el navegador?				
¿Es único para todo el sitio el título para el navegador?				
¿Indica el título para el navegador a qué sitio pertenece la página?				
¿Describe claramente el título para el navegador los contenidos de la página?				
¿Es breve el título para el navegador?				
¿Es único para todo el sitio el título para el navegador?				
¿Refleja el título para el navegador la localización de la página dentro de la jerarquía del sitio?				
¿Describe el título de la página a qué sitio corresponde la página (p.ej. mediante un logo)?				
¿Indica el título de la página que se trata de la página principal del sitio?				
¿Es breve el título de la página?				
¿Es el título de la página único para todo el sitio?				
¿Incluye la página un enlace a la página principal?				
¿Incluye la página un enlace al mapa del sitio, índice o tabla de contenidos?				
Para sitios organizados en jerarquías, ¿incluye la página un enlace a la página de nivel superior?				




en la jerarquía?				
¿Están dispuestos de forma consistente los enlaces internos en cada página?				
En el caso de enlaces que remitan a documentos en servidores externos, ¿se indica al usuario que está abandonando el sitio?				
¿Aparece el URL en el cuerpo de la página?				
¿Existe un mapa del sitio o índice en la página principal o en una página directamente enlazada desde la página principal?				
¿Incluye el mapa del sitio o índice como mínimo las principales zonas del sitio?				
¿Es el mapa del sitio o índice fácil de leer?				
¿Está el mapa del sitio o índice organizado de una manera lógica?				
¿Contienen los ítems del mapa del sitio o índice enlaces de hipertexto a los materiales referidos?				
Si el sitio contiene gran cantidad de información, ¿incluye un motor de búsqueda interno para facilitar a los usuarios la localización fácil y rápida de la información?				
¿Se obtienen con el motor de búsqueda resultados completos y apropiados?				



	UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA				
	Tema:		Plantilla para medir los contenidos no textuales del Sitio Web de la UTPL.		
	Página	9	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
Si la página incluye un gráfico como un logo o un mapa navegable, existe un texto alternativo para quienes visualizan la página en modo sólo-texto?				
Si la página incluye elementos no textuales (como archivos de audio o de vídeo) que requieran software adicional, ¿existe una indicación acerca de la necesidad de software adicional y dónde puede obtenerse?				
Si un archivo requiere software adicional para ser accedido, siempre que sea posible, ¿se suministra la misma información en otro formato que no requiera software adicional?				
Si una página requiere un navegador específico o una versión específica del navegador, ¿se especifica lo que se necesita y dónde puede ser obtenido?				
Cuando el activar un enlace conduce a la descarga de un gran archivo gráfico, de sonido o de vídeo, ¿se suministra información al usuario avisándole que esto va a ocurrir?				



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA					
	Tema:		Plantilla para medir funcionalidad de los enlaces internos y externos, facilidad para imprimir páginas y visibilidad para los spiders mediante meta tags del Sitio Web de la UTPL.		
	Página	10	Tesista	Director	Co-Director
	Versión	1.1	Luis Javier Capa León	Ing. Manuel Sucunuta	Ing. Armando Cabrera

PREGUNTA	Si	No	NA	OBSERVACIONES
¿Se ha comprobado que las páginas pueden imprimirse de forma legible?				
¿Se ha comprobado que los frames pueden imprimirse correctamente?				
Si un documento ha sido dividido en varios archivos diferentes, ¿se ha contemplado la posibilidad de que pueda imprimirse todo el documento desde un único archivo?				
¿Se ha comprobado la funcionalidad de los enlaces externos?				
¿Se comprueba regularmente la funcionalidad de los enlaces externos?				
¿Se comprueba regularmente el contenido de los enlaces externos para verificar si son apropiados para el sitio y si están actualizados?				
¿Se ha probado el sitio con usuarios tipo?				
¿Se ha probado el sitio con distintos navegadores?				
¿Se ha establecido algún mecanismo para consultar regularmente a los usuarios sobre el diseño y contenido del sitio?				
¿Se ha establecido un procedimiento regular para comprobar la funcionalidad del sitio comprobando enlaces internos y externos, formularios, imágenes, animaciones y motor de búsqueda interno?				
¿Se retira del sitio regularmente el material antiguo o no actualizado?				



¿Se indica de algún modo la inclusión de material nuevo en el sitio?				
¿Se facilita el acceso a las páginas cuya dirección haya sido cambiada?				
Cuando se realizan revisiones de fondo a una página, ¿se indica qué materiales han sido revisados?				
¿Se suministra la dirección de correo electrónico del Webmaster para informarle acerca de fallos técnicos?				
¿Se ha comprobado la descripción que se formula de cada página de cara a su aparición en los resultados de los buscadores (descriptor meta tags)?				
¿Se ha comprobado la descripción que se formula de cada página para verificar cómo será indexada en los buscadores (keyword meta tags)?				



ANEXO 3



Plantilla para el Plan de pruebas inicial

Versión [1.0.0]



Información del documento

TÍTULO: Plan de pruebas inicial
SUBTÍTULO:
VERSIÓN:
ARCHIVO:
AUTOR:
ESTADO:

Lista de cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN

Firmas y aprobaciones

ELABORADO
POR: _____
FECHA: _____ Firma: _____

REVISADO
POR: _____
FECHA: _____ Firma: _____

APROBADO
POR: _____
FECHA: _____ Firma: _____



PLAN DE PRUEBAS

Introducción

Propósito

El plan de pruebas tiene por objeto, verificar que cada componente del sistema funcione correctamente ante escenarios representativos del entorno en el que se ejecutará y validar que el software desarrollado, corresponda a los requisitos expresados por la empresa, respecto de la definición, ejecución y control de los procesos.

En este documento se reúne la información necesaria, para planear y controlar el desarrollo de las pruebas de verificación y validación de...<Nombre Sitio o Aplicación a evaluar>

El documento plan de pruebas para el proyecto... ..<Nombre Sitio o Aplicación a evaluar>, persigue los siguientes objetivos:

- Identificación de la información del proyecto: documentación generada en la etapa de análisis y los componentes de software de la aplicación que deben ser probados.
- Descripción de las estrategias de prueba a ser empleadas.
- Identificación de los recursos necesarios y suministro estimado de los esfuerzos de pruebas.
- Lista de los productos entregables del proyecto de pruebas.



Alcance

Las pruebas se enfocarán al..., que incluye los siguientes módulos:

- ✓ <Nombre módulo>
- ✓ <Nombre módulo>
- ✓ <Nombre módulo>

Además se enfocarán hacia la funcionalidad de los productos obtenidos a los cuales se incluirán los siguientes niveles de pruebas:

- Validación de documentación generada en etapa de análisis,
- Pruebas unitarias (componentes/ unidades simples).
- Pruebas de construcción (comportamiento de los componentes/ unidades al integrar datos reales).
- Pruebas del sistema (validan la función del negocio y analizan el rendimiento del sistema completo).
- Pruebas de aceptación (realizadas por el usuario final), para verificar la confiabilidad del sistema antes de ser puesto oficialmente en marcha.

Para cada una de las fases indicadas anteriormente, existirán los siguientes Tipos de Prueba:

- Funciones del negocio.
- Interfaz de usuario.
- Base de datos.



Audiencia

La audiencia involucrada en el proyecto es la siguiente:

- Desarrolladores.
- Usuarios

Referencias

Se toma como referencia la siguiente información:

- Visión.
- Casos de Uso.
- Requerimientos no funcionales.

Objetivo y factores que motivan las pruebas

Garantizar un producto final de calidad, que satisfaga las necesidades y expectativas de la empresa....; en relación...

Misión

Los principales objetivos de la iteración plan de pruebas son:

- Encontrar fallas de forma eficiente.
- Encontrar problemas significativos.
- Validar documentación generada.
- Evaluar y mitigar riesgos percibidos en cuanto a calidad.
- Verificar la conformidad a los requerimientos establecidos (funcionales y no funcionales).
- Recomendar sobre la calidad del producto.
- Brindar satisfacción a los afectados.
- Recomendar mejoras en las actividades al plan de pruebas.



Factores de motivación

Es necesario efectuar pruebas por varias razones entre ellas:

- Evaluar y mitigar riesgos del proyecto.
- Evaluar y mitigar riesgos técnicos.
- Efectuar pruebas de las especificaciones funcionales (casos de uso).
- Considerar los elementos del diseño.
- Detectar posibles fallas o defectos.

Identificación del sistema a probar

Aquí se brinda una breve descripción de los componentes que conforman el sistema y que deberán ser evaluados para controlar su nivel de calidad.

Estrategia de pruebas

La estrategia de pruebas presenta la aproximación recomendada para las pruebas de acuerdo al objetivo que persigue. Para cada tipo de prueba, se deberá suministrar una descripción de la misma y por qué está siendo implementada y ejecutada.

Las principales consideraciones para la estrategia de pruebas son las técnicas a ser usadas y el criterio de conocimiento cuando la prueba esté completa.

Adicionalmente a las consideraciones provistas para cada prueba, las pruebas sólo deberían ser ejecutadas usando bases de datos controladas, en ambientes seguros.

Referente a la documentación, conforme se van utilizando las diferentes herramientas se va generando la información relacionada.



Tipos de pruebas

Pruebas de integridad de datos

Las pruebas de integridad de datos, buscan comprobar que el acceso y manipulación de los datos generados a través del sistema.

Objetivo de la Prueba:	Asegurar que los métodos de acceso a la Base de Datos funcionan correctamente.
Técnica:	Invocar cada método y proceso de acceso a la base de datos, con datos válidos e inválidos. Inspeccionar la base de datos para asegurar que los datos han sido cargados como se pretendía, todos los eventos de bases de datos ocurren apropiadamente, o revisar los datos devueltos para asegurar que los datos correctos fueron recuperados.
Criterio de Conclusión:	

Pruebas de funcionamiento

Las pruebas de funcionamiento se basan en los casos de uso. Las metas que persiguen son: verificar la apropiada aceptación de datos, procesamiento y recuperación. Este tipo de pruebas están basadas en la verificación de la aplicación (y sus procesos internos) mediante la interacción con la aplicación a través de la interfaz gráfica y analizar los resultados obtenidos.

Objetivo de la prueba:	Asegurar la apropiada funcionalidad de las prueba de objetivos, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
------------------------	--



Técnica:	Se ejecuta cada caso de uso, usando datos válidos e inválidos, para verificar lo siguiente: <ul style="list-style-type: none">· Que los resultados esperados ocurran cuando se usen datos válidos.· Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.
Criterio de Conclusión:	

Pruebas de aceptación del usuario

Las pruebas de aceptación están diseñadas para asegurar al cliente que se construyó la aplicación estipulada; se caracterizan por tener al cliente como testigo y se ejecutan en la plataforma que van a operar.

Objetivo de la prueba:	Asegurar al cliente que la aplicación satisface sus necesidades planteadas.
Técnica:	Se ejecutan todas las opciones de la aplicación para verificar: <ul style="list-style-type: none">· Que las pantallas son fáciles de usar.· Que se realizan todas las funciones especificadas por el cliente.· Que se desplieguen los mensajes apropiados de error y precaución cuando se usan datos inválidos.
Criterio de Conclusión:	

Metodología

La metodología que guiará el desarrollo de las pruebas antes descritas, comprende:

- Bitácora de casos de prueba: Se construirá en base a la especificación detallada de los casos de uso, que contenga la siguiente información:



- ✓ Número de caso de prueba.
 - ✓ Código de escenario.
 - ✓ Código de caso de prueba.
 - ✓ Descripción del escenario.
 - ✓ Descripción del caso de prueba.
 - ✓ Confirmación o negación de la ejecución del caso de prueba.
 - ✓ Confirmación o negación de si genera error la ejecución del caso de prueba.
 - ✓ La funcionalidad sobre la que se realiza la prueba.
- Bitácora de errores: Se generará con los resultados de la ejecución de los casos de prueba, ésta contendrá la siguiente información:
- ✓ Número de error.
 - ✓ Código que vincula el caso de prueba y el escenario mediante la concatenación de sus códigos.
 - ✓ Las iniciales del probador que detecta el error.
 - ✓ Nombre del error.
 - ✓ Descripción del error.
 - ✓ Funcionalidad en la que se presenta el error.
 - ✓ Tipo de error.
 - ✓ Prioridad.
 - ✓ Desarrollador encargado de la corrección del error.
 - ✓ Estado del error.

Los errores detectados en el desarrollo de los casos de prueba, se han tipificado como:

- Defecto: Aquellos que no permiten continuar con la ejecución del sistema, o su presencia producirá resultados incorrectos en la ejecución de otras funcionalidades de la aplicación.



- Incidente: Son errores que no detienen el funcionamiento del sistema; sin embargo, producen datos ligeramente incorrectos, como por ejemplo al realizar el cálculo del saldo de una cuenta contable el valor obtenido difiere en decimales del valor correcto.
- Discrepancia: Son situaciones que pueden ser mejoradas, no afectan en el funcionamiento del sistema.

Los estados que se han definido para los errores son:

- Abierto: Significa que no se han tomado medidas correctivas para el error y que este continúa presente.
- Cerrado: En este estado se encuentran los errores que han sido debidamente corregidos y los casos de prueba que los generan se han ejecutado nuevamente, para verificarlos.
- Informe de resultados de pruebas: Este producto contendrá como su nombre lo indica, los resultados de las pruebas ejecutadas.

Herramientas a utilizar

Se utilizaran las siguientes herramientas de apoyo.

Ítem	Herramienta	Casa Vendedora	Versión
Administración de pruebas			
Control de defectos			
Pruebas manuales			



Recursos

Esta sección presenta los roles recomendados para la disciplina de pruebas en el proyecto actual. Cada recurso presenta sus responsabilidades, conocimientos y habilidades.

Recursos humanos

Esta tabla presenta los recursos que participarán en las actividades de pruebas.

Recursos Humanos		
Recurso	Cantidad recomendada (Medio Tiempo)	Responsabilidades específicas
Administrador de pruebas	1	Provee coordinación general de actividades de pruebas. Responsabilidades: <ul style="list-style-type: none">· Generar plan de pruebas.· Proveer coordinación de actividades.· Obtener y asignar recursos.
Diseñador de pruebas	1	Identifica, prioriza e implementa casos de pruebas. Responsabilidades: <ul style="list-style-type: none">· Apoyo a generar plan de pruebas.· Diseñar los casos de pruebas.



Probador	1	Ejecuta las pruebas. Responsabilidades: <ul style="list-style-type: none">· Ejecutar las pruebas.· Registrar resultados.· Documentar errores para control de cambios.
Administración de datos	1	Asegura que los datos están preparados para las pruebas.

Recursos tecnológicos

A continuación se detallan los recursos tecnológicos (hardware) del proyecto bajo pruebas.

Recursos del sistema (Hardware)	
Recurso	Nombre/Tipo
Computador	
—Red/Subset	
—Nombre de Servidor	
—Nombre de Base de Datos	

Recursos del sistema (Hardware)	
Recurso	Nombre/Tipo
Computador	
—Red/Subset	
—Nombre de Servidor	
—Nombre de Base de Datos	

Recursos del sistema (Hardware)	
---------------------------------	--



Recurso	Nombre/Tipo
Servidor	
—Red/Subset	
—Nombre de Servidor	
—Nombre de Base de Datos	

Responsabilidades

EQUIPO	FUNCIONES	COMPONENTES

Entregables

Informe de pruebas

El informe de pruebas, contendrá los resultados de la aplicación de los casos de prueba, además incluirá:

- Bitácora de casos de pruebas.
- Bitácora de errores.
- Informe de resultados pruebas.

Riesgos

Se describen los riesgos que pueden presentarse al desarrollarse las pruebas.



- Retraso en la entrega del producto.
- Tiempo elevado en la corrección de los defectos encontrados.
- Número limitado de quienes realizan las pruebas, lo cual puede provocar retrasos.

Glosario



ANEXO 4



Usabilidad en Aplicaciones Web

“En realidad, usabilidad es que algo funciona bien: significa, que una persona de capacidad y aptitudes medias (o incluso, por debajo de la media) pueda usar algo, tanto si es un sitio Web, un mando a distancia o una puerta giratoria, para lo que se supone que sirve, sin frustrarse desesperadamente mientras lo hace.” Steve Krug²⁸

La usabilidad es el factor de calidad de mayor importancia dentro de las aplicaciones web. Aquellas que tienen una usabilidad pobremente explotada o ninguna, causan reacciones de rechazo en los usuarios. Por esta razón podemos comprender que el objetivo primordial consiste en diseñar una Aplicación web en la cual los usuarios puedan ver alcanzar sus objetivos planteados de manera efectiva, eficiente y satisfactoria. [Kappel, 2003].

Para lograr este propósito hay que considerar dos elementos a la par en el desarrollo, tanto las tendencias actuales como las necesidades reales del usuario. La usabilidad no debe ser integrada al sistema de una sola vez, es un proceso que nace desde tempranas etapas en la construcción de la aplicación web y se controla en cada iteración de la misma de acuerdo a los objetivos de usabilidad planteados.

La usabilidad es la medida de calidad de sistemas de información interactivos de acuerdo con el equilibrio entre productividad, navegabilidad, accesibilidad y optimización, en función de los objetivos del negocio y del usuario. La usabilidad produce dos fenómenos que deben ser aprovechados por el usuario que requiere una aplicación web; incremento en las

²⁸ Usabilidad o como mejorar la experiencia del usuario, disponible en: <http://www.emoticon.es/descarga/usabilidad.pdf>



actividades del negocio como tal e incremento en el número de usuarios nuevos. Una Web puede tener una buena usabilidad y ser de pobre calidad pero, es prácticamente imposible, por definición, que una Web de alta calidad sea pobre en usabilidad. [Kappel, 2003].

La usabilidad es importante por las siguientes razones:

- ✓ Aumenta la productividad, reduce el coste de soporte y de aprendizaje, aumenta las ventas y reduce el tiempo y coste de desarrollo. [Eguíluz, 2008]
- ✓ No sólo son factores económicos. Los usuarios aseguran que el 60% de las veces no son capaces de encontrar la información que buscan. El resultado es una pérdida de tiempo y de productividad, una mayor frustración de los usuarios y el abandono de los sitios Web. [Eguíluz, 2008]

De hecho, la usabilidad no se limita a sistemas computacionales exclusivamente, sino que es un concepto aplicable a cualquier elemento en el cual se va a producir una interacción entre un humano y un dispositivo. En el caso de los sistemas computacionales, la usabilidad va a abarcar desde el proceso de instalación de la aplicación hasta el punto en que el sistema sea utilizado por el usuario, incluyendo también el proceso de mantenimiento. La usabilidad tiene cinco atributos definidos:

Facilidad de aprendizaje.

¿Cuánto le toma al usuario típico de una comunidad aprender la manera en cómo se usan los comandos relevantes a un conjunto de tareas? Quiere decir qué tan rápido el usuario va a aprender a usar un sistema con el cual no ha tenido contacto previamente. Este punto se refiere a la consecución de tareas básicas por parte de un usuario novato.



Velocidad de desempeño.

¿Cuánto le toma a un usuario completar un grupo de tareas específicas? Una vez que el usuario ha aprendido a utilizar el sistema, se va a ponderar el lograr la velocidad con que puede completar una tarea específica.

Tasas de error por parte de los usuarios.

¿Cuántos y qué errores hace la gente al ejecutar un grupo de tareas específicas? Este apartado apunta hacia los errores cometidos por el usuario. Este atributo se refiere a aquellos errores que comete el usuario al utilizar el sistema. Una aplicación ideal evitaría que el usuario cometa errores y funcionaría de manera óptima a cualquier petición por parte del usuario. En la práctica esto difícilmente se logra. Es vital que una vez que se produzca un error, el sistema se lo haga saber rápida y claramente al usuario, le advierta sobre la severidad del mismo y le provea de algún mecanismo para recuperarse de ese error.

Retención sobre el tiempo.

¿Qué tan bien recuerdan los usuarios la manera en cómo funciona el sistema después de una hora, un día o una semana? Cuando un usuario ha utilizado un sistema tiempo atrás, y tiene la necesidad de utilizarlo de nuevo la curva de aprendizaje debe de ser significativamente menor que el caso del usuario que nunca haya utilizado dicho sistema. Esto es de primordial importancia para aplicaciones usadas intermitentemente. [Kappel, 2003]

Satisfacción subjetiva.

¿Qué tanto le gustaron a los usuarios los distintos atributos del sistema?



Este atributo se refiere a la impresión subjetiva del usuario respecto a la aplicación web.

Evaluación.

La principal actividad en el proceso de usabilidad es la evaluación. La evaluación de la usabilidad puede ayudar a determinar cuál es el nivel actual de la aplicación y si de hecho el diseño elegido realmente funciona. Los datos que se recaban mediante la observación del usuario frente a la aplicación y ver su desempeño, es información muy valiosa que ayudan en definitiva a detectar posibles falencias del sistema.

Existen diferentes técnicas para evaluar un sistema. Su uso depende de variables tales como costo, disponibilidad de tiempo, personal calificado para interpretar los datos, entre otros factores.

4.1 Usabilidad en el diseño

A continuación se detallan algunos de los parámetros básicos que se sugieren para el diseño usable de una aplicación web. Más información disponible en Usabilidad Web en la Práctica²⁹ y Usabilidad, El Guión Multimedia³⁰.

²⁹ Javier Eguíluz Pérez, Miguel Sánchez Pérez.

³⁰ David Moner Cano, Jordi Sabaté Alsina.



4.1.1 Tiempos de Respuesta

Para establecer estándares, se toman como base los tiempos de respuesta humanos, por ejemplo: una respuesta directa involucra 0.1 segundos. Hasta 3 segundos se considera tolerante, de 8 a 10 segundos se entiende que no es posible una acción. Dentro de las aplicaciones web se considera normal un tiempo de respuesta de hasta 3 segundos e incluso hasta 10 segundos se consideraría aceptable, dadas las condiciones a las que pueda estar sometida la aplicación. Bajo estos parámetros se recomienda diseñar páginas tipo slim que contienen un número limitado de elementos multimedia y, además, utilizan métodos de compresión para reducir el tamaño de éstos. [Kappel, 2003].

Como una regla general, el volumen entero de cada página dentro de una Aplicación web no debe superar los 50 Kb. Otro método para acelerar la carga de un sitio web, es especificar las dimensiones de las imágenes dentro del código HTML, para permitir al navegador presentar una página antes que las imágenes sean cargadas. Se debe tener cuidado especial en la paginación en caso de que la información resultante sea demasiado grande para presentar.

4.1.2 Colores

Una recomendación importante es el uso excesivo de colores. Se recomienda el uso de 5 como máximo, y se debe evitar en la medida de lo posible, colores destellantes o extremos. Los colores son usados para resaltar a la vista diferentes áreas de la Aplicación web que requieran especial atención por parte del usuario final. Otro punto a considerar es la tecnología que utilizan los dispositivos que proyectan la Aplicación web y como se verán los colores dentro de ellos, así como las dificultades visuales de los usuarios finales.



4.1.3 Capas de Texto

En su origen, las páginas Web eran únicamente en formato texto. Poco a poco fueron evolucionando al incluir imágenes, applets y muchos otros elementos multimedia. Pero el texto sigue siendo por el momento el principal vehículo de comunicación en la Web.

Leer el texto en una pantalla es menos eficiente que leerlo en un papel, por esta razón el texto debe ser cuidadosamente seleccionado antes de ser incorporado a la Aplicación web. Además, el background de una aplicación web no debería tener una estructura o patrón porque eso afecta seriamente la lectura de la información disponible. [Chile, 2000]

Desde que la lectura en pantalla se hizo popular también se tornó aburrida, los usuarios tratan de leer el texto de manera diagonal, es decir, buscando palabras claves. Este comportamiento debe ser entendido y aprovechado por los desarrolladores para colocar texto con frases precisas, exactas y acorde al tipo de usuario que las leerá, preferentemente que estén en su propio idioma. En cuanto a su anchura, no debe superar los 800 píxeles, y esto siendo generosos. Un tamaño de pantalla de 800x600 píxeles suele ser el habitual en los ordenadores domésticos, y ya se tiende a resoluciones de 1024x768.

4.1.4 Estructura de Páginas

Para asegurarse una fácil orientación dentro de una página, se debe evitar el scrolling horizontal, esto facilita que el usuario se pierda en el sitio. Según varias investigaciones realizadas se llegó a la conclusión de que para los usuarios el estilo de dos columnas como en los periódicos incrementa la visualización de la página. En resumen, no debería sobrecargar las páginas con la finalidad de no saturar los recursos que el usuario usa para acceder. Una página también debería permitir la impresión, y de no ser posible, o



sea difícil por diferentes causas inherentes a la aplicación web, se debe ofrecer un panel de impresión por separado. [Chile, 2000]

4.1.5 Estructura de Navegación

La navegación dentro de un sitio web representa un gran porcentaje dentro de la usabilidad y su éxito. El perderse en una aplicación web debido a una navegación no lineal debe ser evitada a toda costa, para ello es necesario un modelo mental de navegación tan rápido como sea posible, que facilite al usuario el memorizar el sitio. Los ingredientes para una navegación eficiente son: una estructura básica, lógica y clara con retroalimentación constante, información clara acerca del contenido y disponibilidad de lo siguiente que será mostrado. Hay que recordar que sin importar el tipo de navegador que el usuario disponga, siempre ofrece elementos de navegación independientes del contexto que pueden causar comportamientos indeseados o volver ineficiente a la Aplicación web.

4.1.6 Los enlaces

La regla que rige a los enlaces sugiere que éstos deben ser embebidos dentro del texto donde se localicen, por ejemplo: La Constitución del Ecuador [señala los siguientes artículos](#)... tiene menos atractivo que colocarlo así: Encuentre los Artículos de la Constitución del Ecuador [aquí](#)... Cuando se realiza este tipo de cambios en los enlaces, se brinda una idea de a dónde llegar sin la necesidad de leer todo el párrafo completo. [Chile, 2000]

En ningún caso se ha de saturar la página con enlaces. Cuando un visitante llega a la página, si lo primero que ve son numerosos enlaces distintos no sabrá a dónde ir primero, aún estando muy bien organizados.



Hay que señalar que nunca conviene cambiar el color de los enlaces; aunque está permitido, se evitará confundir al usuario si no se cambian los colores por defecto de los enlaces, puesto que el usuario está acostumbrado a ellos y asocia cada color a su significado: los tonos azules con los sitios no visitados y los rojos o morados con los que sí ha visto anteriormente.

4.1.7 Los gráficos de la Aplicación Web

Las imágenes de una página Web se han convertido en una parte fundamental de ella. Su uso permite dotar a la página de una mayor vistosidad, publicidad y en general, permiten un mayor grado de comunicación del mensaje que se quiere transmitir. Pero su uso indiscriminado también es perjudicial. Existen recomendaciones para el correcto uso de los gráficos que harán que una página Web sea más usable. El principal problema que acarrea el uso de imágenes es, debido a su tamaño, los efectos que pueden tener en la velocidad de carga de una página. Aunque se espera que con el tiempo la mayoría de usuarios tenga un acceso a internet con banda ancha, la realidad es bien distinta y la gran mayoría de internautas siguen navegando con velocidades de 56Kbps. [Eguíluz, 2008]

Conviene usar imágenes pequeñas. Esto resulta tan útil a efectos de velocidad como de estética. Si se está mostrando una imagen que tiene que verse con gran detalle, como un cuadro o una fotografía que se quiere vender, se puede emplear una imagen que ocupe toda la pantalla. Pero si es un logotipo o la imagen de un producto, no debería superar un tercio de la pantalla, tanto en altura como en anchura. Para la mayoría de las imágenes, con que sean reconocibles, ya tienen el tamaño suficiente. En cuanto al tamaño de la propia imagen, hay que mantenerlo en torno a 20 KB o 50KB como máximo. Si se utilizan imágenes de mayor tamaño es preferible utilizar una imagen en miniatura que sea un enlace a la imagen



de tamaño mayor, indicando a su lado su tamaño para que el usuario tenga una idea de lo que va a descargar. [Chile, 2000]

4.2 Métodos de usabilidad en la Ingeniería Web

A continuación se describen algunas de las actividades de ingeniería de usabilidad enfocadas específicamente en el desarrollo de aplicaciones web:

4.2.1 Análisis de requerimientos

Un análisis de requerimientos comienza con la formulación de una visión, describiendo las funciones básicas y características de rendimiento, casos de uso esenciales y los objetivos que deben cumplirse [Kappel, 2003].

Uno de los más importantes resultados de la fase de análisis es una clara definición de los objetivos de usabilidad de manera cualitativa y cuantitativa [Kappel, 2003].

Los usuarios pueden participar directa o indirectamente en el análisis de requerimientos. La participación indirecta es más fácil de llevar a cabo que la directa para una aplicación web anónima, es decir aquella que no tiene un grupo de usuarios definidos o específicos. La participación directa se basa en grupos o entrevistas con usuarios reales. Desde el punto de vista de la ingeniería de usabilidad, el análisis de requerimientos se puede resumir como se muestra:

- ✓ Perfiles de usuario.
- ✓ Análisis de tareas, escenarios y modelo de casos de uso.
- ✓ Especificación de las propiedades de la plataforma.
- ✓ Objetivos de usabilidad cualitativos y cuantitativos.



4.2.2 Diseño

El diseñador toma los resultados del análisis de requerimientos y procede a elaborar un modelo conceptual de la interfaz de usuario. Aquí define la estructura básica de la aplicación web permitiendo al desarrollador ver a través de la Aplicación web. El modelo se centra en aspectos básicos de representación, principios de navegación e interacción.

Los métodos de participación del usuario que se pueden utilizar en esta fase son: storyboarding, paper mock-ups y card sorting enfocados a grupos de usuarios semejantes a los grupos reales a los que la aplicación web estará destinada. Un storyboard representa secuencias, un paper mock-ups es una vista preliminar diseñada por el desarrollador de la interfaz, y card sorting sirve para determinar la mejor manera de que los contenidos encajen en la estructura de la Aplicación web. [Kappel, 2003]

Una vez que el modelo está elaborado es evaluado por el experto en usabilidad para determinar si existen debilidades potenciales en etapas tempranas del desarrollo como secuencias inconsistentes, estructura de navegación confusa, etc. Una vez que el modelo conceptual fue establecido, el diseñador de la interfaz y el desarrollador e implementador pueden construir juntos un diseño más detallado.

La participación del usuario en la fase de diseño se realiza en cualquiera de las dos etapas que se mencionan:

- ✓ Prototipado: se utiliza para mostrar como luciría el sistema una vez terminado el desarrollo.
- ✓ Pruebas de usabilidad: estas pruebas pueden estar ligadas a un contexto o tareas reales. Para las aplicaciones web con un grupo de usuarios finales conocido, una prueba de usabilidad convencional es



suficiente mientras que una prueba de usabilidad remota debe ser usada para aplicaciones con grupos de usuarios no conocidos.

4.2.3 Implementación

Aquí es donde quizás el experto en usabilidad juega su papel más importante para el éxito o fracaso de la Aplicación web. El experto en usabilidad debe revisar la consistencia de la aplicación, observar la directivas o estándares utilizados y documentados, y principalmente el desarrollo de estrategias si es que los requerimientos han cambio a lo largo del ciclo de vida del proyecto.

Es importante invitar a participar al mismo grupo de usuarios que participó en fases anteriores y preguntarles si el sistema hace lo que esperan que haga o pedirles su opinión sobre el mismo. Desde el punto de vista de la ingeniería de usabilidad, la documentación acerca del éxito de los objetivos de usabilidad definidos en el análisis de requerimientos, es el resultado más relevante dentro de la fase de implementación.

4.3 Tendencias de usabilidad en la Ingeniería Web

4.3.1 Patrones de diseño

Un patrón es una herramienta conceptual que captura la naturaleza de una solución exitosa para un problema de arquitectura recurrente. [Kappel, 2003]. Un conjunto de patrones es conocido como lenguaje de patrones y generalmente es publicado en un catálogo. Un patrón simple debería ser descrito en 3 partes: un problema u objetivo, la solución y el contexto.

El concepto de patrón más ampliamente utilizado en el desarrollo de aplicaciones web es el patrón de diseño, que se utiliza particularmente en el paradigma orientado a objetos. En este contexto, un patrón de diseño es una descripción de las clases y de los objetos que trabajan conjuntamente



para resolver un problema concreto [Gamma, 1998]. Estos patrones muestran una solución a un problema, que se ha obtenido a partir de su uso en otras aplicaciones diferentes. [Eguíluz, 2008]

Sin embargo, la idea de los patrones arquitectónicos puede también utilizarse con independencia de la evaluación de la arquitectura, ya que proporcionan soluciones de diseño para ciertos requisitos. Se espera que la consideración de estos requisitos de usabilidad desde el primer momento del desarrollo y más tarde en el diseño por medio de los patrones arquitectónicos, proporcione mejoras en la usabilidad del sistema final.

Hay que tener en cuenta que la usabilidad de la aplicación web final ha de ser validada y medida una vez que ésta se haya construido y esté operativa. Por tanto, tendremos que esperar hasta que estos resultados se hayan aplicado en proyectos reales para obtener los datos empíricos que puedan verificar adecuadamente los beneficios, ahora intuitivos, del uso de los patrones arquitectónicos con respecto a la usabilidad de los sistemas.

4.3.2 Accesibilidad

La necesidad de la accesibilidad surgió del hecho de que cada día existen más y más personas con diferentes tipos de discapacidades que demandan el uso de la web como medio de información y comunicación. Estudios señalan que aproximadamente el 20% de la población mundial tiene algún grado de discapacidad en uno o más de sus órganos sensoriales y el más frecuente aún es el problema visual. [Eguíluz, 2008]

Ver el Anexo 2, se muestra un plantilla de evaluación para sitios web con los principales aspectos a ser tomados en cuenta.



ANEXO 5



Seguridad en Aplicaciones Web

La seguridad de las aplicaciones web siempre ha estado en entredicho, aunque ciertos ataques se han vuelto más comunes que otros (a medida que la plataforma sigue madurando). Es mucho más fácil explotarlas, no porque sean más inseguras que los programas normales de escritorio, sino porque el acceso a estas es mucho más grande (Internet) por un número mayor de personas. Si a eso le sumamos el hecho de el "síndrome de el beta" al cual compañías como Google ya nos tienen acostumbrados y que los ciclos de desarrollo están gobernados por lo que el departamento de mercadeo y no QA entonces se podrá dar cuenta que el escenario es bien complejo.

El problema de la seguridad de las aplicaciones web es una consecuencia de cómo se escribe el software, de qué técnicas son utilizadas. Por ejemplo, la complejidad de la plataforma es un factor importante; pero aún más es saber cuándo utilizar una herramienta en vez de la otra. El uso de estándares de programación y reglas básicas dentro de una corporación ayudan a reducir, en una gran medida, los riesgos de aplicaciones que pueden ser potencialmente inseguros a ataques no enfocados al código, usando por ejemplo ingeniería social.

En pocas palabras no es solamente un problema de código, firewalls o de si usemos o no un software para revisar el código de manera automática. Todos esos factores son importantes, pero los problemas de fondo como son la complejidad de las aplicaciones (en su uso, instalación y número de funcionalidad) y como son construidas, creo que son aún más importantes en sus repercusiones.



5.1 Consideraciones sobre seguridad

Existen algunos puntos que deben considerarse como estándares en la seguridad de aplicación web, cuyo objetivo fundamental es proporcionar mecanismos para que durante el proceso de transferencia de información no interfieran agentes que vulneren la integridad de la información. Estos puntos clave son: [Kappel, 2003]

- ✓ Confidencialidad: La información que viaja entre dos entes no puede ser interferida por un tercero y aprovechar su contenido. Se sugiere el uso de la encriptación o canales dedicados como una VPN. (Virtual Private Network).
- ✓ Integridad: La información no debe ser modificada por ningún ente durante el proceso de transferencia.
- ✓ Non-repudiation: Se verifica que tanto el remitente como el receptor de la información son exactamente quienes envían o reciben el mensaje, respectivamente.
- ✓ Autenticación: Se verifica la identidad del usuario que desea acceder a la información desde cualquier origen usando mecanismos diversos.
- ✓ Autorización: Se controla las actividades que el usuario está habilitado a realizar de acuerdo a sus privilegios en el sistema.
- ✓ Disponibilidad: Hacer que el sitio esté a disposición de los usuarios el mayor tiempo posible y de ser necesario un reajuste de funcionalidades, tratar de minimizar los costos tanto como sea posible.
- ✓ Privacidad: Resguardar los datos de las operaciones que se realicen con información importante para los clientes como números de cuentas, claves de acceso, correos electrónicos, etc.



Los diferentes aspectos de seguridad son ocasionados por la naturaleza compleja de las aplicaciones web que están disponibles en internet y los usuarios acceden a ellas por medio de canales públicos de comunicación sin las seguridades necesarias para ello.

5.2 Encriptaciones, firmas digitales y certificados

La encriptación implica el uso de funciones matemáticas sobre texto plano para transformarlo en texto encriptado. La decriptación se basa en el mismo principio pero aplicado a la inversa sobre el texto codificado. La gran mayoría de los algoritmos de encriptación se basan en claves secretas para encriptar o decriptar que, si no son conocidas, es imposible aplicar estos procesos a la información. El criptoanálisis es la ciencia que se enfoca en los esfuerzos por determinar el sentido de la información que se ha encriptado. [Kappel, 2003]

Las firmas digitales tienen la agradable característica de permitir verificar la autenticidad del documento firmado, además de permitir verificar la identidad del firmante. Si se cambia un carácter del mensaje original, ya no se podrá verificar la firma del mensaje. Las firmas digitales se basan en la criptografía de clave pública. Este tipo de sistemas criptográficos utiliza dos claves, una pública que es de conocimiento general o se la puede obtener fácilmente, y una privada o inherente al receptor del mensaje.

Un certificado es un documento digital mediante el cual, un tercero confiable, garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública. Si bien existen variados formatos para certificados digitales, los más comúnmente empleados se rigen por el estándar UIT-T X.509. El certificado contiene usualmente el nombre de la entidad certificada, número de serie, fecha de expiración, una copia de la clave pública del titular del certificado y la firma digital de la autoridad emisora



del certificado, de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación.

5.3 Seguridad Cliente / Servidor

5.3.1 Seguridad Punto a Punto

Dentro de la seguridad punto a punto, las opciones más recomendadas al momento de aplicarla son SSL y TLS. SSL actualmente se encuentra en su versión 3.0 y establece una mejor comunicación entre un cliente y el servidor utilizando algoritmos avanzados de encriptación.

Con SSL, se establece una comunicación dedicada entre el servidor y el cliente donde se construye un canal dedicado y encriptado para la comunicación. Las sesiones de seguridad SSL en servidores son muy comunes para aplicaciones de comercio electrónico. Los usuarios pueden revisar su conexión SSL directamente desde sus navegadores, y así obtener información del tipo de algoritmo que está siendo usado en la negociación. Si el proveedor del certificado no es confiable, ya sea porque el certificado expiró o ha sido removido, el usuario recibe una advertencia.

5.3.2 Seguridad End-to-End

Las transacciones en línea, en algunas ocasiones pueden involucrar más de dos entes. La seguridad End-to-End es de crucial importancia para las interacciones de los servicios web, en donde se realizan operaciones que demandan altos niveles de seguridad por medio de controles que el propio nivel de seguridad exija, tales como claves cifradas, varias maneras de autenticación, etc.



5.3.3 Autenticación y Autorización

5.3.3.1 Autenticación

La autenticación se refiere a verificar la identidad del usuario, es decir, a someterlo a un proceso de validación de su identidad para afirmar que el usuario es quien dice ser. El mecanismo más utilizado para este propósito es de login/password. El usuario provee su nombre y contraseña para autenticarse. Se recomienda no transmitir la información de inicio de sesión en texto plano, sino más bien utilizar conexiones SSL.

Otra forma de autenticar a los usuarios es utilizando certificados digitales, pero este método ya no es ampliamente usado en aplicaciones B2C y ha tenido un repunte en aplicaciones B2B o en la autenticación de usuarios en una intranet corporativa.

5.3.3.2 Autorización

Las políticas de autorización, por su parte, especifican lo que los usuarios autenticados son capaces de hacer. Uno de los métodos de control para este fin es el DAC (Discretionary Access Control), que trabaja de acuerdo a los privilegios o prohibiciones asignadas a los usuarios. Las listas de control de acceso son el mejor ejemplo de un sistema DAC; permite o restringe el acceso a recursos, información, tanto a un usuario como a un grupo de usuarios. Si el número de usuarios de una aplicación web tiende potencialmente a crecer en grandes cantidades, la mejor alternativa es utilizar un sistema de control de acceso basado en roles o RBAC para mejorar la escalabilidad y reducir el esfuerzo de administración de la aplicación web. Los derechos o prohibiciones son asignados a los roles que cumple cada usuario en lugar de a los recursos e información.



5.3.4 Sistemas de pago electrónico

Para el comercio electrónico, son requeridos sistemas de pago seguros. Actualmente, las tarjetas de crédito son el sistema de pago más ampliamente utilizado para realizar transacciones en internet. Los usuarios envían su nombre, su número de tarjeta y fecha de expiración al vendedor en línea o a un servicio de pagos en línea a través de canales seguros SSL. Como el vendedor no tiene la tarjeta de crédito del usuario en sus manos, es necesaria una validación basada en diferentes aspectos de la tarjeta como la dirección del cliente, los últimos dígitos de la tarjeta, etc.

Otra forma de validación son los números que son impresos de manera única en cada tarjeta. Cuando se ha validado la identidad del usuario de estos sistemas, su número de tarjeta de crédito es encriptado y enviado al vendedor donde la información de pago es firmada digitalmente y reenviada al banco para efectuar la transacción, entonces el mensaje es descriptado y se procede con el cargo a la tarjeta de crédito.

De esta manera, la información de pago no llega en texto plano ni al vendedor ni al banco. Existen otros métodos de pago o de transferencia electrónica de dinero que demandan niveles elevados de seguridad, como los sistemas de pago por servicios, en donde es necesario una cuenta de correo electrónico y un contrato con la empresa que brinda el servicio de pagos como PayPal.

5.4 Problemas en la seguridad de los clientes

Si la información personal es intercambiada durante operaciones en línea, los usuarios deben estar seguros que los respectivos proveedores del servicio les garanticen que sus datos son manejados de una manera confiable. La privacidad de los datos de los usuarios no puede ser solamente intervenida durante el proceso de transferencia de la misma con



el proveedor, sino que también ha tenido un gran crecimiento las nuevas técnicas como phishing y web Spoofing, que son tratadas en un apartado más adelante. [Kappel, 2003]

5.4.1 Resguardando la información

Cuando se utilizan sesiones SSL, los usuarios tienen la tranquilidad de una transferencia segura de su información, pero no pueden determinar cómo esta información es manejada luego de que se completa la transferencia. De esta manera, los proveedores deben establecer vínculos de relación confiable con los usuarios antes de que éstos empiecen a transferir información.

La plataforma para preferencias privadas o P3P, propuesta por el W3C³¹, brinda un estándar para la declaración de políticas de protección de información en un formato XML legible para las máquinas, y que se puede mostrar a los usuarios finales como políticas de seguridad. Los clientes igualmente pueden declarar sus preferencias de privacidad utilizando agentes de P3P que vienen incorporados en la mayoría de los navegadores que utilizamos hoy en día. Si no se detectan conflictos de privacidad, la aplicación web es mostrada al usuario, caso contrario, se obtienen datos del problema detectado y se muestra un mensaje en formato legible para el usuario, dando una explicación del porqué del error. Estas preferencias en el cliente sólo deben ser definidas una vez y automáticamente son verificadas antes de acceder a una aplicación web.

³¹ W3C, disponible en: <http://www.w3.org/P3P/>



5.4.2 Phishing and Web Spoofing

El término Phishing es una abreviación para el proceso de captura de información privada. Estos ataques son efectuados con la intención clara de obtener información personal como números de seguro social, números de tarjetas de crédito que pueden ser usadas para suplantación de identidad o fraudes comerciales. Este tipo de ataque afecta especialmente a clientes de organizaciones financieras. El método más usado es enviar correos electrónicos suplantando la identidad de un representante de la organización, pidiendo información al usuario para diferentes actualizaciones de servicios; de esta forma, el usuario ingresa su información y se efectúa el daño. [Kappel, 2003]

En cambio, el término Web Spoofing se refiere a un conjunto de técnicas que pretenden simular la presencia de una web en una organización con la finalidad de engañar a los clientes a través de sitios web convincentes, proveedores de correo semejantes y URLs reales.

Tanto los ataques de Phishing como de Web Spoofing no sólo causan daños a los usuarios finales sino que también implican un perjuicio para los proveedores de servicios. Esfuerzos administrativos y financieros están siendo destinados día a día para contrarrestar los efectos de estos ataques y advertir al usuario tan pronto como sea posible. Actualmente la transferencia se está realizando mediante el uso de tarjetas inteligentes en lugar de mecanismos PIN/TAN que utilizan las tarjetas de crédito hoy en día

5.4.3 Seguridad de escritorio

Otros factores que influyen en la seguridad son las amenazas causadas por virus y gusanos. Es responsabilidad de cada usuario el contrarrestar estas



amenazas, mantenimiento sus sistemas actualizados, navegando por sitios seguros, utilizando antivirus o firewalls, etc. A continuación se detalla una lista de las amenazas más recurrentes:

- ✓ Adware y Spyware: Adware es una categoría de programas utilizados para la promoción no autorizada de contenidos ya sea en sitios web o embebidos dentro de programas de escritorio. Spyware por su parte está orientado a obtener información personal mediante el monitoreo de las actividades del usuario en el equipo y el escaneo del sistema. Una vez realizado esto, la información es remitida a un sistema remoto.
- ✓ Dialers: Son programas que establecen conexiones a internet a costos sumamente elevados. Se instalan sin el consentimiento del usuario y las conexiones siempre son ocultas. Se distribuyen de la misma manera que un spyware o Adware vía correo electrónico. Con el incremento del uso de conexiones DSL, los dialers están decreciendo en su uso.
- ✓ Accesos remotos y puertas traseras: Son programas que vulneran la seguridad de un equipo, en base a los puntos vulnerables del sistema para permitir un control remoto del equipo.
- ✓ Virus, Gusanos y Troyanos: Los virus son programas o código que ataca los archivos ejecutables o documentos que contienen macros. Los gusanos son programas que replican copias de sí mismos lo que produce un consumo innecesario de recursos del equipo. Los troyanos no se auto reproducen a sí mismos, están ocultos tras otros programas que aparentan ser útiles para el usuario. El daño causado varía desde destrucción de información importante hasta permitiendo el acceso, no autorizado, a recursos computacionales



otorgando un acceso remoto hacia el equipo donde se encuentra alojado el troyano. [Kappel, 2003]



ANEXO 6

ESTÁNDARES DE LA IEEE UTILIZADOS POR SWEBOK PARA LA
ELABORACIÓN DE LAS PLANTILLAS UTILIZADAS EN ESTA INVESTIGACIÓN



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
IEEE Std 610.12-1990 (R2002)	IEEE Standard Glossary of Software Engineering Terminology	This standard is a glossary of software engineering terminology.	X	X	X	X	X	X	X	X	X	X
IEEE Std 730-2002	IEEE Standard for Software Quality Assurance Plans	This standard specifies the format and content of software quality assurance plans.						S	S			P
IEEE Std 828-1998	IEEE Standard for Software Configuration Management Plans	This standard specifies the content of a software configuration management plan along with requirements for specific activities.						P	S			
IEEE Std 829-1998	IEEE Standard for Software Test Documentation	This standard describes the form and content of a basic set of documentation for planning, executing, and reporting software testing.			S	P						S
IEEE Std 830-1998	IEEE Recommended Practice for Software Requirements Specifications	This document recommends the content and characteristics of a software requirements specification. Sample outlines are provided.	P									
IEEE Std 982.1-1988	IEEE Standard Dictionary of Measures to Produce Reliable Software	This standard provides a set of measures for evaluating the reliability of a software product and for obtaining early forecasts of the reliability of a product under development.		S	S	S			S			P



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
IEEE Std 1008-1987 (R2003)	IEEE Standard for Software Unit Testing	This standard describes a sound approach to software unit testing and the concepts and assumptions on which it is based. It also provides guidance and resource information.			S	P				S		S
IEEE Std 1012-1998 and 1012a-1998	IEEE Standard for Software Verification and Validation	This standard describes software verification and validation processes that are used to determine if software products of an activity meet the requirements of the activity and to determine if software satisfies the user's needs for the intended usage. The scope includes analysis, evaluation, review, inspection, assessment, and testing of both products and processes.										P
IEEE Std 1016-1998	IEEE Recommended Practice for Software Design Descriptions	This document recommends content and organization of a software design description.		P								
IEEE Std 1028-1997 (R2002)	IEEE Standard for Software Reviews	This standard defines five types of software reviews and procedures for their execution. Review types include management reviews, technical reviews, inspections, walk-throughs, and audits.	S	S	S			S	S			P
IEEE Std 1044-1993 (R2002)	IEEE Standard for Software Anomalies Classification	This standard provides a uniform approach to the classification of anomalies found in software and its documentation. It includes helpful lists of anomaly classifications and related data.				S			S	S		P
IEEE Std 1045-1992 (R2002)	IEEE Standard for Software Productivity Metrics	This standard provides a consistent terminology for software productivity measures and defines a consistent way to measure the elements that go into computing software productivity.							P	S		
IEEE Std 1058-1998	IEEE Standard for Software Project Management Plans	This standard describes the format and contents of a software project management plan.							P			



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
IEEE Std 1465-1998 // ISO/IEC 12119	IEEE Standard, Adoption of International Standard ISO/IEC 12119:1994(E), Information Technology — Software packages — Quality requirements and testing	This standard describes quality requirements specifically suitable for software packages and guidance on testing the package against those requirements.	S									P
IEEE Std 1471-2000	IEEE Recommended Practice for Architectural Description of Software Intensive Systems	This document recommends a conceptual framework and content for the architectural description of software-intensive systems.	S	S								P
IEEE Std 1490-1998	IEEE Guide — Adoption of PMI Standard — A Guide to the Project Management Body of Knowledge	This document is the IEEE adoption of a Project Management Body of Knowledge defined by the Project Management Institute. It identifies and describes generally accepted knowledge regarding project management.							P			
IEEE Std 1517-1999	IEEE Standard for Information Technology — Software Life Cycle Processes — Reuse Processes	This standard provides life cycle processes for systematic software reuse. The processes are suitable for use with IEEE/EIA 12207.			S					P		



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
IEEE Std 2001-2002	IEEE Recommended Practice for the Internet — Web Site Engineering, Web Site Management, and Web Site Life Cycle	This document recommends practices for engineering World Wide Web pages for use in intranet and extranet environments.										P
ISO 9001:2000	Quality Management Systems — Requirements	This standard specifies the requirements for an organizational quality management system aiming to provide products meeting requirements and enhance customer satisfaction.								S		P
ISO/IEC 9126-1:2001	Software Engineering — Product Quality — Part 1: Quality Model	This standard provides a model for software product quality covering internal quality, external quality, and quality in use. The model is in the form of a taxonomy of defined characteristics which software may exhibit.	P	S	S	S						
IEEE/EIA 12207.0-1996 // ISO/IEC 12207:1995	Industry Implementation of International Standard ISO/IEC 12207:1995, Standard for Information Technology — Software Life Cycle Processes	This standard provides a framework of processes used across the entire life cycle of software.	X	X	X	X	X	X	X	P	X	X



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
IEEE/EIA 12207.2-1997	Industry Implementation of International Standard ISO/IEC 12207:1995, Standard for Information Technology — Software Life Cycle Processes — Implementation Considerations	This document provides additional guidance for the implementation of the life cycle processes of IEEE/EIA 12207.0.	X	X	X	X	X	X	X	P	X	X
IEEE Std 14143.1-2000 // ISO/IEC 14143-1:1998	IEEE Adoption of ISO/IEC 14143-1:1998 — Information Technology— Software Measurement — Functional Size Measurement — Part 1: Definition of Concepts	This standard describes the fundamental concepts of a class of measures collectively known as functional size.	P						S			S
ISO/IEC TR 14471:1999	Information technology — Software engineering — Guidelines for the adoption of CASE tools	This document provides guidance in establishing processes and activities that may be applied in the adoption of CASE technology.									P	



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
ISO/IEC 15026:1998	Information Technology — System and Software Integrity Levels	This International Standard introduces the concepts of software integrity levels and software integrity requirements. It defines the concepts associated with integrity levels, defines the processes for determining integrity levels and software integrity requirements, and places requirements on each process.	S	S								P
ISO/IEC TR 15271:1998	Information technology — Guide for ISO/IEC 12207 (Software Life Cycle Processes)	This document is a guide to the use of ISO/IEC 12207.								P		
ISO/IEC 15288:2002	Systems Engineering — System Life Cycle Processes	This standard provides a framework of processes used across the entire life cycle of human-made systems.								P		
ISO/IEC TR 15504 (9 parts) and Draft IS 15504 (5 parts)	Software Engineering — Process Assessment	This technical report (now being revised as a standard) provides requirements on methods for performing process assessment as a basis for process improvement or capability determination.								P		
ISO/IEC 15939:2002	Software Engineering — Software Measurement Process	This standard provides a life cycle process for software measurement. The process is suitable for use with IEEE/EIA 12207.							S	P		S
ISO/IEC 19761:2003	Software engineering — COSMIC-FFP — A functional size measurement method	This standard describes the COSMIC-FFP Functional Size Measurement Method, a functional size measurement method conforming to the requirements of ISO/IEC 14143-1.	P				S		S			S



Standard Number	Standard Name	Description	Software Requirements	Software Design	Software Construction	Software Testing	Software Maintenance	Software Configuration Management	Software Engineering Management	Software Engineering Process	Software Engineering Tools and Methods	Software Quality
ISO/IEC 20968:2002	Software engineering — Mk II Function Point Analysis — Counting Practices Manual	This standard describes Mk II Function Point Analysis, a functional size measurement method conforming to the requirements of ISO/IEC 14143-1.	P				S		S			S
ISO/IEC 90003	Software and Systems Engineering — Guidelines for the Application of ISO 9001:2000 to Computer Software	This standard provides guidance for organizations in the application of ISO 9001:2000 to the acquisition, supply, development, operation, and maintenance of computer software.								S		P