

**REQUERIMIENTOS NO FUNCIONALES PARA
APLICACIONES WEB**

Silvana del Valle, Rojo

Director/es:
Alejandro Oliveros

**“Trabajo Integrador presentado para obtener el grado de
especialista en Ingeniería de Software”**

Facultad de Informática – Universidad Nacional de La Plata

Abril, 2012

Índice

RESUMEN.....	6
ABSTRACT	7
1. INTRODUCCIÓN	8
1.1 OBJETIVO.....	8
1.2 ESTADO DEL ARTE.....	8
1.3 ESTRUCTURA DEL TRABAJO DE INTEGRACIÓN	9
2. REQUERIMIENTOS NO FUNCIONES Y LAS APLICACIONES WEB.	11
2.1 REQUERIMIENTOS NO FUNCIONALES ¿QUÉ SON?	11
2.2 CONCEPTOS A APLICAR EN ESTE TRABAJO.....	24
3. TÉCNICAS UTILIZADAS EN LA INGENIERÍA DE REQUERIMIENTOS	29
4. METODOLOGÍAS DE DESARROLLO DE APLICACIONES WEB Y LOS REQUERIMIENTOS	37
4.1 APLICACIONES WEB	37
4.2 METODOLOGÍAS DE DESARROLLO WEB Y LOS REQUERIMIENTOS NO FUNCIONALES	38
4.2.1 WSDM: WEB SITE DESIGN METHOD.....	42
4.2.2 HFPM: HYPERMEDIA FLEXIBLE PROCESS MODELING.....	46
4.2.3 UWE: UML-BASED WEB ENGINEERING.....	51
4.2.4 WEBML: WEB MODELING LANGUAGE.....	55
4.2.5 NTD: NAVIGATIONAL DEVELOPMENT TECHNIQUES	59
4.2.6 DDDP: DESIGN-DRIVEN REQUIREMENTS ELICITATION	67
5. ESTUDIO COMPARATIVO TRATAMIENTO DE REQUERIMIENTOS NO FUNCIONALES EN LAS METODOLOGÍAS DE DESARROLLO DE APLICACIONES WEB.	71
5.1 TIPOS DE REQUERIMIENTOS NO FUNCIONALES EN CADA ENFOQUE.....	71
5.2 TÉCNICAS DE ELICITACIÓN, ESPECIFICACIÓN, VALIDACIÓN Y ADMINISTRACIÓN PARA TRATAMIENTO DE REQUERIMIENTOS NO FUNCIONALES.	82

5.3	CICLO DE VIDA Y LOS REQUERIMIENTOS NO FUNCIONALES.....	84
6.	CONCLUSIONES	87
7.	BIBLIOGRAFÍA.....	90

Índice de Figuras

FIGURA 2.1: CLASIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES – MAMANI.....	16
FIGURA 2.2: CLASIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES – SOMMERVILLE.	17
FIGURA 2.3: ÁRBOL DE CARACTERÍSTICAS DE LA CALIDAD DEL SOFTWARE – BOEHM	17
FIGURA 2.4: UNA TAXONOMÍA DE REQUERIMIENTOS NO FUNCIONALES- CYSNEIROS.....	18
FIGURA 4.1: METODOLOGÍAS PARA APLICACIONES WEB Y LOS REQUERIMIENTOS.....	41
FIGURA 4.2: DIAGRAMA DE LAS FASES DE WSDM.....	43
FIGURA 4.3: DIAGRAMA DE LAS FASES DE HFPM	47
FIGURA 4.4: DIAGRAMA DE PROCESO DE UWE	52
FIGURA 4.5: DIAGRAMA DE PROCESO DE WEBML.....	56
FIGURA 4.6: DIAGRAMA DE PROCESO DE NDT	61
FIGURA 4.7: DIAGRAMA DE PROCESO DE DDDP.....	68
FIGURA 5.1: REQUERIMIENTOS NO FUNCIONALES Y METODOLOGÍAS DE DESARROLLO WEB - PARCIAL.....	79
FIGURA 5.2: REQUERIMIENTOS NO FUNCIONALES Y METODOLOGÍAS DE DESARROLLO WEB.....	81

Índice de Tablas

TABLA 2.1: EJEMPLOS DE DISCREPANCIAS EN ESQUEMAS DE CLASIFICACIÓN DE RNF	23
TABLA 2.2: REQUERIMIENTOS NO FUNCIONALES COMO REQUERIMIENTOS DE CALIDAD	27
TABLA 4.1: REQUERIMIENTOS TRATADOS POR CADA METODOLOGÍA	40
TABLA 5.1: REQUERIMIENTOS NO FUNCIONALES DE CALIDAD Y LAS METODOLOGÍAS WEB.....	72
TABLA 5.2: REQUERIMIENTOS NO FUNCIONALES – RESTRICCIONES Y LAS METODOLOGÍAS WEB.....	74
TABLA 5.3: REQUERIMIENTOS IDENTIFICADOS POR CADA METODOLOGÍA	77
TABLA 5.4: REQUERIMIENTOS NO FUNCIONALES IDENTIFICADOS Y TRATADOS POR CADA METODOLOGÍA DE DESARROLLO WEB - PARCIAL	78
TABLA 5.5: REQUERIMIENTOS NO FUNCIONALES IDENTIFICADOS Y TRATADOS POR CADA METODOLOGÍA DE DESARROLLO WEB	80
TABLA 5.6: TÉCNICAS DE INGENIERÍA DE REQUERIMIENTOS Y HERRAMIENTAS CASE APLICADAS A RNF	84
TABLA 5.7: LOS REQUERIMIENTOS NO FUNCIONALES Y EL CICLO DE VIDA	85
TABLA 5.8: INFLUENCIA DE LAS RESTRICCIONES DE PROCESO EN EL CICLO DE VIDA	87

Resumen

Los requerimientos no funcionales de los sistemas software no son un todo homogéneo, hay una falta de consenso para responder preguntas tales como: ¿qué son? ¿cómo se clasifican dentro del contexto de desarrollo de software?, y ¿cómo se clasifican en el desarrollo de aplicaciones Web?

Este trabajo presenta el estado de arte del concepto de Requerimientos No Funcionales dentro de la literatura existente en la Ingeniería de Requerimientos y establece como bases conceptuales que los Requerimientos No Funcionales son *requerimientos de calidad y son restricciones*.

Sobre esta base se realiza un estudio comparativo de seis enfoques de desarrollo de aplicaciones Web. Se estudian los procesos de desarrollo que cada una de ellas utiliza para determinar los requerimientos no funcionales; las técnicas de la ingeniería de requerimientos que proponen para su elicitación, especificación, validación y gestión; y se establece en qué fase del ciclo de vida de desarrollo de software identifican y tratan a los requerimientos no funcionales.

Palabras claves: Ingeniería de Requerimientos, Requerimientos No Funcionales, Procesos de desarrollo, Aplicaciones Web.

Abstract

Non-functional requirements of software systems are not a homogeneous whole; there is a lack of consensus to answer questions such as: what are they? How are classified within the context of software development? And how classify it in Web Applications developing?.

This text presents the state of art of the concept of non-functional requirements in the existing literature on the Requirements Engineering and establishes the conceptual foundations of non-functional requirements as *quality requirements* and *restrictions*.

On this basis, we made a comparative study of six approaches of Web applications development. We studied the development processes that each approach uses to determine of non-functional requirements, the engineering techniques proposed for requirements elicitation, specification, validation and management, and established at what stage of software development life cycle they identify and address Non-functional requirements.

Keywords: Requirements Engineering, Non-functional Requirements, Development Process, Web Applications.

1. Introducción

1.1 Objetivo

Estudio comparativo de los diferentes tipos de Requerimientos No Funcionales y las técnicas utilizadas para su elicitación, especificación, validación y gestión por las metodologías de desarrollo de aplicaciones Web.

1.2 Estado del Arte

Un requerimiento es definido como una condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto (IEEE Standard 610.12-1990).

Las tareas de elicitación, especificación y validación de requerimientos no son sencillas, la complejidad radica en parte en cómo identificar las funcionalidades que el sistema deberá implementar para satisfacer las necesidades expresadas y esperadas de los usuarios. Otra fuente de dificultades en el desarrollo de los requerimientos la constituyen los Requerimientos No Funcionales, esto es, las restricciones que debe satisfacer el sistema en construcción.

Si bien el estado de arte de desarrollo de aplicaciones Web en la actualidad dista en aprovechar plenamente los avances en la Ingeniería del Software en los años recientes, las comunidades de Ingeniería Web y de Ingeniería de Requerimientos han dado pasos en el camino de reconocer la importancia de entender, antes de comenzar el proceso de diseño, las necesidades que deberá satisfacer una Aplicación Web. La comunidad científica realizó recientemente el “1st Workshop on The Web and Requirements Engineering (WeRE)” en el contexto de la mayor conferencia de Ingeniería de Requerimientos, esto denota fuerza y mayor relevancia por considerar la contribución de los requerimientos al desarrollo de aplicaciones Web.

Los requerimientos de los sistemas software en general y los de las aplicaciones Web en particular no son un todo homogéneo; se pueden clasificar en dos grandes tipos, requerimientos funcionales y no funcionales.

El trabajo de Escalona y Koch [1], aporta una comparación de metodologías de desarrollo web, destacando entre otros aspectos, técnicas de ingeniería de requerimientos y tipos de requerimientos que manejan en sus diferentes etapas. Respecto a los Requerimientos No Funcionales menciona cuales metodologías detectan los mismos, de hecho solo seis de las diez revisadas corresponden con ello y los agrupa a fin de resolver las diferencias entre las metodologías sobre estos requerimientos.

Cada uno de los tipos de Requerimientos No Funcionales merece ser analizado en particular. Baste mencionar la problemática específica del desarrollo de los requerimientos de seguridad o de performance.

1.3 Estructura del Trabajo de integración

El alcance del presente trabajo contempla la realización de evaluaciones comparativas de los enfoques existentes en las metodologías de desarrollo de aplicaciones Web para la obtención de los diferentes de tipos de Requerimientos No Funcionales y las técnicas utilizadas para su desarrollo y gestión.

Este trabajo integrador se concentrará en dar respuestas a las siguientes preguntas:

¿Consideran las metodologías de desarrollo Web los Requerimientos No Funcionales? ¿Cuál es el alcance de los requerimientos considerados?

¿Cuál es el concepto que esas metodologías tienen de los distintos tipos de Requerimientos No Funcionales?, ¿Cómo se comparan unos con los otros?

¿Qué técnicas proponen (si lo hacen) para manejar a los Requerimientos No Funcionales?, ¿Definen procesos específicos de Elicitación, Validación y Especificación?

¿Cómo se relacionan esos conceptos de Requerimientos No Funcionales y las técnicas propuestas para su tratamiento con las predominantes en la ingeniería de requerimientos?

¿Consideran las metodologías la gestión de Requerimientos No Funcionales?

¿Hay herramientas que soporten a las metodologías de desarrollo web?

¿Cómo es el tratamiento de los Requerimientos No Funcionales en el ciclo de vida de desarrollo propuesto por cada metodología?

En el capítulo 2, para responder a la pregunta *¿Qué son los Requerimientos No Funcionales?*, se presenta una revisión y análisis de definiciones y esquemas de clasificación existentes en la literatura de ingeniería de requerimientos. Al final de este capítulo se presentan los conceptos base sobre los tipos de Requerimientos No Funcionales a ser considerados en este trabajo.

El capítulo 3, muestra una revisión de las técnicas de captura, especificación, y validación de requerimientos, predominantes en la ingeniería de requerimientos. Las herramientas para administración de requerimientos se presentan en el capítulo 4 junto con la metodología a la que dan soporte.

En el capítulo 4, se realiza un análisis comparativo de las seis metodologías de desarrollo web, tomadas a partir del estudio realizado por Escalona y Koch [1] y señaladas por los autores como aquellas que identifican Requerimientos No Funcionales. En este capítulo se analiza el ciclo de vida completo de cada enfoque a fin de establecer cuáles tipos de Requerimientos No Funcionales son identificados en cada uno y que técnicas son empleadas para su elicitación, especificación, validación y gestión de requerimientos.

En el capítulo 5, se realiza una comparación de las metodologías estudiadas en el capítulo 4, a fin de analizar:

- a) Conceptos de Requerimientos No Funcionales son identificados en cada metodología.
- b) Técnicas de elicitación, validación y especificación que utilizan para el tratamiento de los Requerimientos No Funcionales. Técnicas que proponen (si lo hacen) para la gestión de los Requerimientos No Funcionales.
- c) Los Requerimientos No Funcionales identificados por cada metodología y las fases del ciclo de vida.

El capítulo 6 se presentan las conclusiones y se mencionan líneas de investigación o trabajos que se desprenden a partir de este estudio.

2. Requerimientos no funciones y las aplicaciones Web.

2.1 Requerimientos No Funcionales ¿Qué son?

La bibliografía existente presenta una gran diversidad de enunciados y conceptos para definir Requerimientos No Funcionales, si bien el término es utilizado desde hace años, en la actualidad no hay consenso en la comunidad de la Ingeniería de para establecer “qué son” y por consiguiente tampoco hay unificación de criterios para establecer cómo se elicitán, como se especifican, ni como se validan.

La carencia de consenso se ve reflejada en la diversidad de términos utilizados en las definiciones, con significados poco claros, que dan lugar a la ambigüedad sobre su alcance o lo que representan los mismos; las discrepancias conceptuales también pueden encontrarse en las clasificaciones propuestas para Requerimientos No Funcionales, que incluyen árboles o categorías de conceptos que se muestran en forma separada en otras definiciones.

Para ilustrar lo expuesto se citarán ejemplos de definiciones y esquemas de clasificación extraídos de trabajos relacionados a esta problemática como [31] [32] y en fuentes adicionales:

- Glinz en [31] expone una discusión sobre la problemática de la noción de Requerimientos No Funcionales, basado en el análisis de trece definiciones seleccionadas de la literatura y la web.
- Chung y Leite en [32] presentan una revisión del estado de arte del tratamiento de los Requerimientos No Funcionales, dentro de la literatura del área de requerimientos, mencionan diez definiciones de diferentes autores, siete de las cuales fueron citadas en el trabajo elaborado por Glinz [31]. Además analizan cuatro esquemas de clasificación.
- Cysneiros en su tesis de doctorado [4] menciona tres esquemas de clasificación de Requerimientos No Funcionales propuestas por diferentes autores, y propone una taxonomía para los mismos.

Las definiciones y esquemas de clasificación los Requerimientos No Funcionales son presentados a continuación indicando su autor y la fuente de la cual fueron extraídas:

- a) “Describen aspectos del comportamiento de un sistema, capturando las propiedades y restricciones bajo las cuales un sistema debe operar”. Anton en [31] [32].
- b) [Son] “atributos requeridos del sistema, incluyendo portabilidad, confiabilidad, eficiencia, la ingeniería humana, capacidad de prueba, compresión y capacidad de modificación”. Davis en [31] [32].
- c) [Son] “requerimientos que no se refieren específicamente a la funcionalidad de un sistema. Se imponen restricciones sobre el producto que se está desarrollando y el proceso de desarrollo, y que especifican restricciones externas que el producto debe cumplir”. Kotonya y Sommerville en [31] [32].
- d) “... Requerimientos globales en el desarrollo o los costos operativos, performance confiabilidad, mantenibilidad, portabilidad, robustez, etc. (...) No hay una definición formal o una lista completa de Requerimientos No Funcionales”. Mylopoulos, Chung y Nixon en [31] [32].
- e) [Son las] “propiedades de comportamiento que las funciones especificadas deben tener, tales como performance, usabilidad”. Ncube en [31] [32].
- f) [Es una] “propiedad, o cualidad, que el producto debe tener, como una apariencia, o una propiedad de la velocidad o precisión”. Robertson and Robertson en [31] [32].
- g) [Es una] “descripción de una propiedad o característica que un sistema de software debe mostrar o una restricción que se debe respetar, que no sea el comportamiento observable del sistema”. Wiegers en [31] [32].
- h) “... Tipos de problemas: problemas funcionales asociados con los servicios a ser proporcionados, y los problemas no funcionales relacionados con la calidad del servicio - como seguridad, precisión, performance, etc”. Van Lamsweerde en [32].
- i) “El término "Requerimiento no-funcional" es utilizado para delinear requerimientos enfocándose en "lo bien" que el software hace algo en contraposición a los requerimientos funcionales, que se centran en el "qué" hace el software”. Paech, Kerkow en [32].

- j) “Dicho de otro forma, NFR constituyen las justificaciones de las decisiones de diseño y restringen la forma en la cual la funcionalidad requerida puede ser realizada”. Landes, Studer en [32].
- k) Los estándares de IEEE no definen este término. El estándar IEEE 610.12 distingue requerimientos de diseño, requerimientos de implementación, requerimientos de interfaz, requerimientos de performance y los requerimientos físicos. [31].
- l) Este término tampoco está definido en el estándar IEEE 830-1998. El estándar define las categorías de funcionalidad, interfaces externas, performance, atributos (portabilidad, seguridad, etc), y las limitaciones de diseño. Los requerimientos del proyecto (por ejemplo, calendario/tiempo, costo, o los requerimientos de desarrollo) están explícitamente excluidos. [31].
- m) [Un RNF es] “un NFR es un requerimiento que especifica propiedades del sistema, tales como restricciones de implementación y de ambiente, performance, dependencias de plataforma, mantenimiento, capacidad de extensión y confiabilidad. Un requerimiento que especifica restricciones físicas sobre un requerimiento funcional”. Jacobson, Booch y Rumbaugh en [31].
- n) [Un NFR es] “un requerimiento de un servicio que no tiene que ver con la funcionalidad, sino que describe atributos, restricciones, consideraciones de performance, diseño, calidad del servicio, consideraciones de ambiente, fallas y recuperación”. SCREEN Glossary en [31].
- o) [Requerimientos No Funcionales: Requerimientos que especifican los criterios que se pueden utilizar para juzgar el funcionamiento de un sistema, en lugar de comportamientos específicos]. Wikipedia en [31].

Hay dentro de la literatura existente, dos definiciones que nos interesa sumar a las descritas. Cysneriros en [7] se refiere a los Requerimientos No Funcionales como:

- p) ... [los NFR] “son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: Precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras”. Cysneiros en [7].
- q) ... [los NFR] “definen restricciones globales sobre el sistema, subsistema, sobre un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue...”

lo cual implica que estos surgen de todas partes del sistema y sus interacciones”. Cysneiros en [7].

Del análisis de las definiciones identificamos que en la mayoría de estas, se utilizan los términos *atributo*, *restricción*, *característica*, *propiedad*, *calidad*, para decir “que son” Requerimientos No Funcionales, algunas definiciones no mencionan ejemplos que aporten claridad o precisión sobre su significado, por lo que estos pueden ser interpretados como un concepto diferente en cada definición. No hay consenso sobre la terminología a usar, tampoco sobre los conceptos que cada término describe, si los analizamos en forma aislada encontramos que:

Atributo. Para Davis en b) cada requerimiento no funcional es un *atributo* del sistema; en IEEE 830-1998 en l) los *atributos* son una categoría que se subdivide en una colección de cualidades específicas y en SCREEN glossary n) el término hace referencia a *atributos* del servicio.

Solo la IEEE 830-1998 en l) detalla ejemplos de *atributos* (portabilidad, seguridad etc.), Davis en b) y SCREEN Glossary en n) usan este término en forma general.

Restricción. Para Kotonya y Sommerville en c) los requerimientos son *restricciones*; en otras definiciones como Anton en a), IEEE 830-1998 en l), Jacobson, Booch and Rumbaugh en m), Wiegers en g) y SCREEN glossary n), Cysneiros en q) se refieren al término como una limitación o condición.

Kotonya y Sommerville en s) ofrecen una clasificación y subclasificación de *restricciones* (propone Requerimientos de producto, de organización y externos).

Las otras definiciones si bien delimitan el término, no mencionan ejemplos específicos, la IEEE 830-1998 en l) restringe el término a *restricciones* de diseño; Jacobson, Booch y Rumbaugh en m) a *restricciones* físicas; Cysneiros en q) a restricciones globales sobre el sistema, subsistema, sobre un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue.

Anton en a), Wiegers en g) y SCREEN glossary n), usan el término de forma general.

Los términos *característica* y *propiedad* tienen un significado similar, hacen referencia a algo que el sistema debe tener. Las definiciones citadas, no son claras respecto si

consideran las *restricciones* como *propiedades*, ambos términos pueden formar parte de una misma definición como en Antón a) y Wieggers g).

Ncube en e) detalla ejemplos de *propiedades* (la performance, la usabilidad), para Robertson and Robertson en f) una *propiedad* puede ser la apariencia, velocidad o precisión; Anton en a) y Wieggers en g) se refieren al término *propiedad* como algo general y no especifican ejemplos.

Calidad. Tanto Robertson y Robertson en f), Van Lamsweerde en h), como Cysneiros en p) cuando mencionan este término, hacen referencia a un conjunto de cualidades específicas que el producto o sistema debe tener.

Robertson y Robertson en f) menciona como ejemplos la velocidad o precisión; Van Lamsweerde en h) la seguridad, precisión, performance y Cysneiros en p) a la precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance.

SCREEN glossary n) usa el término de forma general y Davis en b) si bien se refiere a los Requerimientos No Funcionales como *atributos*, se basa en el árbol de *calidad* presentado por Boehm en t) para enumerarlos.

Algunas de las definiciones citadas, no utilizan los términos descritos, sino que usan otras palabras para definir a los Requerimientos No Funcionales:

Landes, Studer en j) se refieren a estos como... “*las justificaciones de las decisiones de diseño*”...

Paech, Kerkow en i) los describen como... “*el término que es utilizado para delinear requerimientos enfocándose en “lo bien” que el software hace algo*”.

Wikipedia en o) los define como... “*los criterios que se pueden utilizar para juzgar el funcionamiento de un sistema*”.

En mi opinión, dentro del conjunto de definiciones citadas i) y o) son las más imprecisas y ambiguas. La variedad y discrepancia de terminología y conceptos utilizados en las definiciones de Requerimientos No Funcionales, deja expuesta la necesidad de una definición clara y concisa.

Clasificación de Requerimientos No Funcionales:

Las figuras 2.1, 2.2, 2.3 y 2.4 ilustran las clasificaciones de Requerimientos No Funcionales propuestas por Mamani, Sommerville, Boehm y Cysneiros.

r) Propuesta por Mamani en [4]:

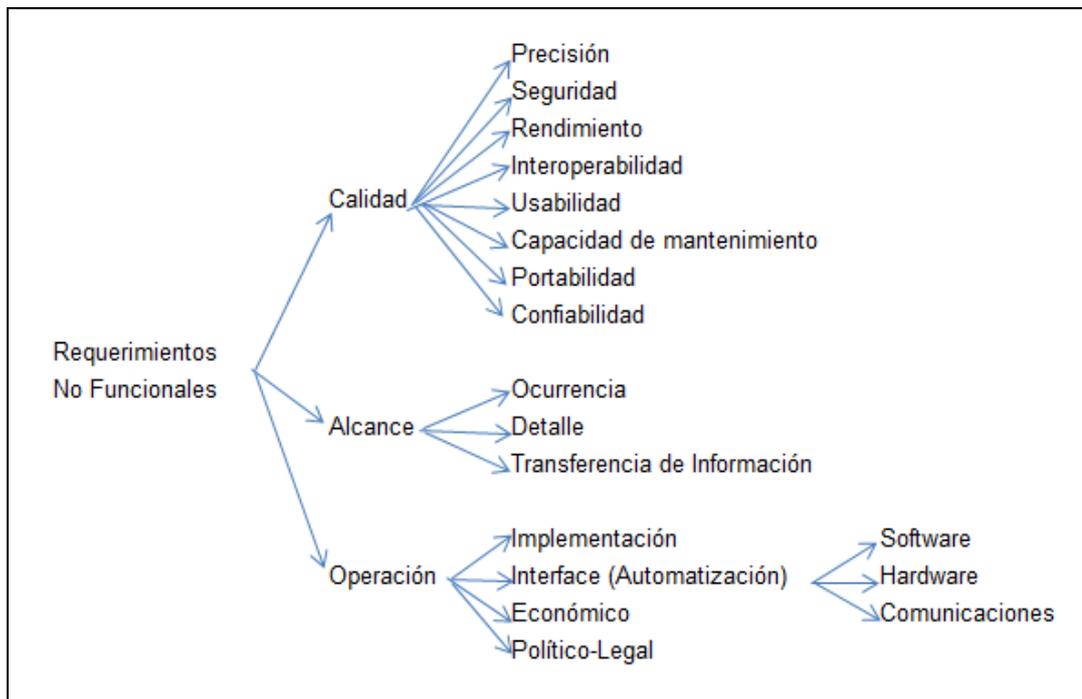


FIGURA 2.1: Clasificación de Requerimientos No Funcionales – Mamani.

s) Propuesta por Sommerville en [4]:

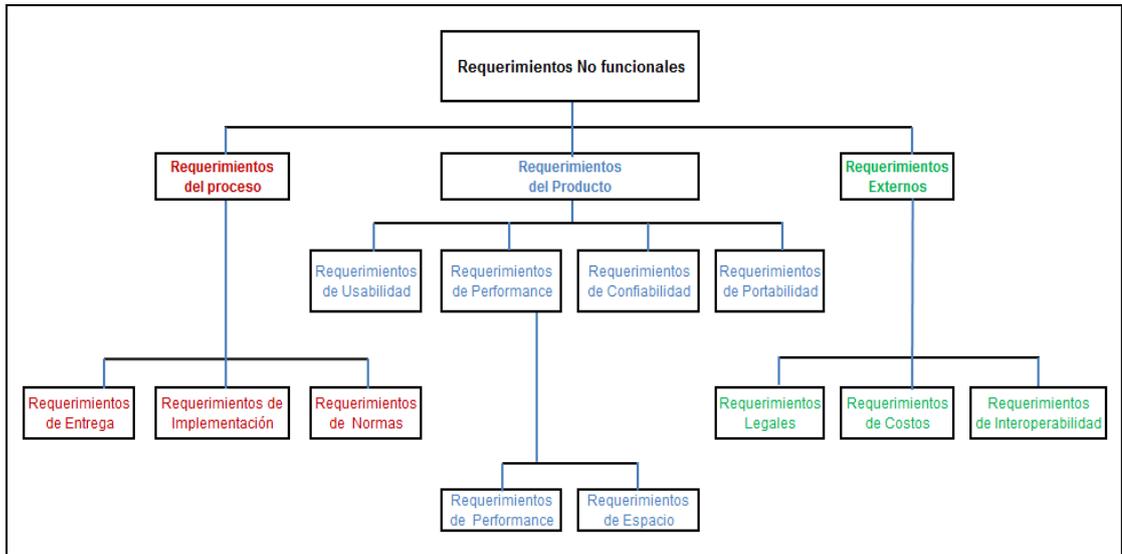


FIGURA 2.2: Clasificación de Requerimientos No Funcionales – Sommerville.

t) Árbol de características de la calidad del software propuesta por Boehm en [4], [32].

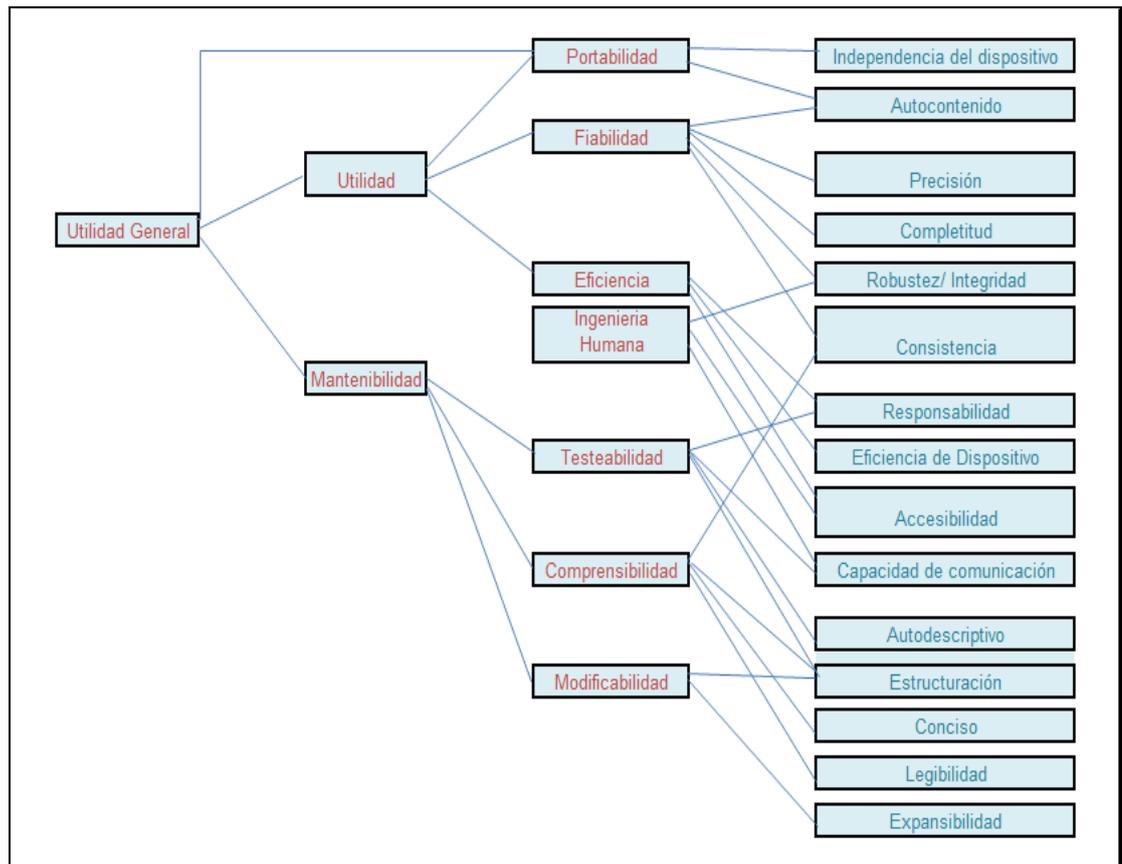


FIGURA 2.3: Árbol de Características de la Calidad del Software – Boehm

u) Taxonomía de Requerimientos No Funcionales propuesta por Cysneiros en [4]:

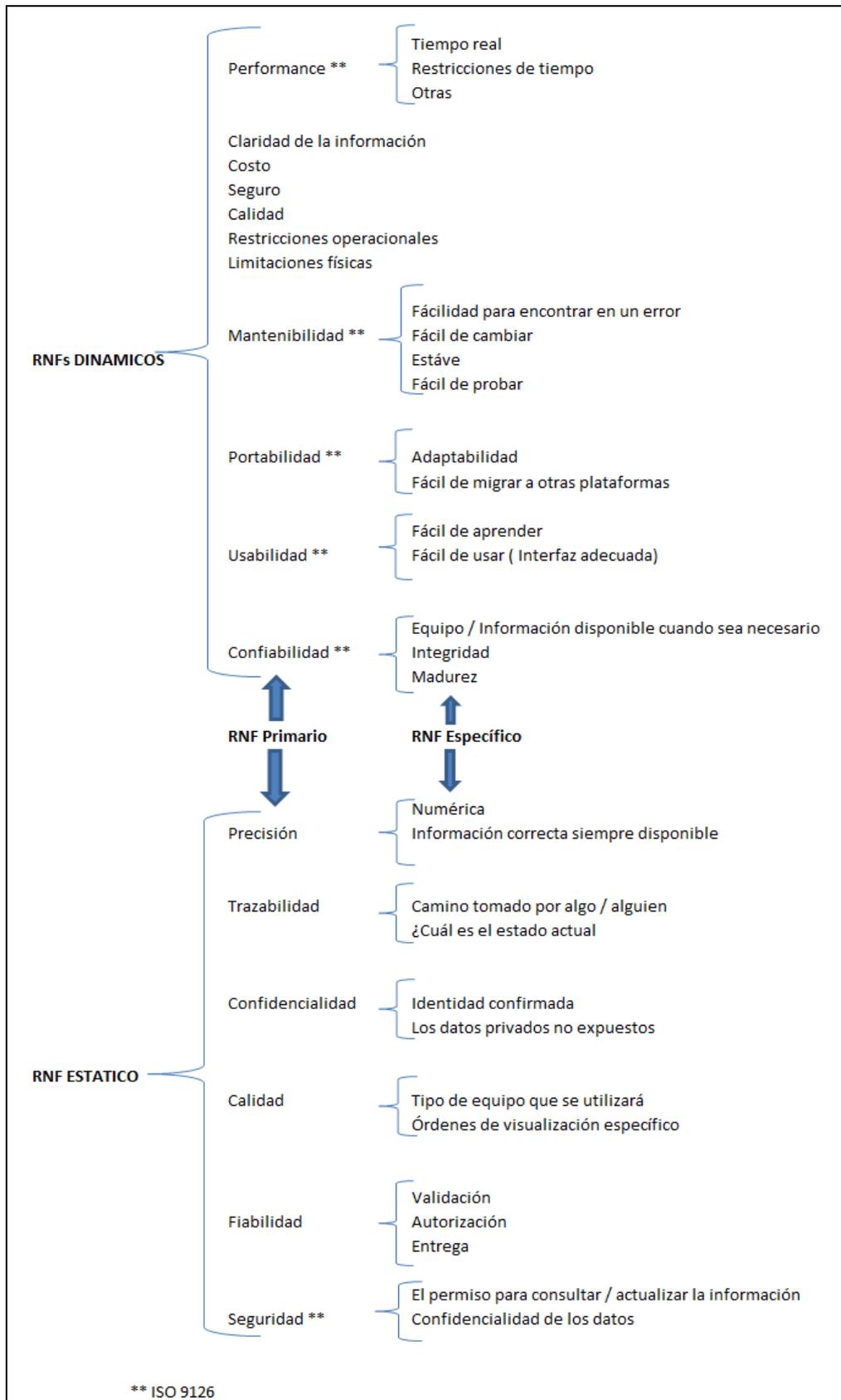


FIGURA 2.4: Una Taxonomía de Requerimientos No Funcionales- Cysneiros

v) Estándar de clasificación IEEE 810-1998 en [31]: clasifica a los Requerimientos No Funcionales en:

- *Requerimientos de Interfaz externa*: Describen los requerimientos que afecten a la:
 - Interfaz de usuario,
 - Interfaz con otros sistemas (hardware y software)
 - Interfaces de comunicaciones.
- *Requerimientos de Performance*: Detallaran los requerimientos relacionados con la carga esperada que debe soportar el sistema. (número de conexiones simultaneas, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán aquellos requerimientos que afecten a la información, por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).
- *Restricciones de Diseño*: Todo aquello que restrinja las decisiones relacionadas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.
- *Atributos del Sistema*: Los atributos de calidad (las "ilities") del sistema: Fiabilidad, mantenibilidad, portabilidad, la seguridad deberá especificar que tipos de usuario están autorizados, o no, a realizar ciertas tareas y cómo se implementarán los mecanismos de seguridad (por ejemplo, por medio de un login y una password).

w) Estándar de calidad internacional ISO/IEC 9126-1 en [34], [32]: Distingue cuatro niveles de calidad. El modelo de calidad del primer nivel propone las seis características que se detallan a continuación, cada una de ellas involucra a su vez subcaracterísticas.

- *Funcionalidad*: Un conjunto de atributos que tienen que ver con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son las que satisfacen las necesidades explícitas o implícitas.

Subcaracterísticas: Adecuación, Precisión, Interoperabilidad, Cumplimiento de la funcionalidad y Seguridad (que reúne características orientadas a la seguridad de la información y los recursos del sistema.

- *Confiabilidad*: Un conjunto de atributos que influyen en la capacidad del software para mantener su nivel de rendimiento bajo las condiciones establecidas para un período determinado de tiempo.

Subcaracterísticas: Madurez, Tolerancia a fallas, Recuperabilidad, Cumplimiento de confiabilidad.

- *Usabilidad*: Un conjunto de atributos que influyen en el esfuerzo necesario para el uso del software por usuarios determinados.

Subcaracterísticas: Comprensibilidad, Facilidad de aprendizaje, Operatividad, Atractivo, Cumplimiento de usabilidad.

- *Eficiencia*: Un conjunto de atributos que influyen en la relación entre el nivel de rendimiento o performance del software y la cantidad de recursos utilizados, bajo condiciones indicadas.

Subcaracterísticas: Tiempo de Comportamiento, Utilización de recursos, Cumplimiento de eficiencia.

- *Mantenibilidad*: Un conjunto de atributos que tienen que ver con el esfuerzo necesario para realizar las modificaciones especificadas.

Subcaracterísticas: Analizabilidad, Variabilidad, Estabilidad, Capacidad de prueba, Cumplimiento de mantenimiento.

- *Portabilidad*: Un conjunto de atributos que influyen en la capacidad del software para ser transferido de un ambiente a otro.

Subcaracterísticas: Adaptabilidad, Facilidad de instalación, Co-Existencia, Intercambiabilidad, Cumplimiento de Portabilidad.

x) Esquema introducido por Roman fue analizado en [32], este describe la siguiente clasificación de Requerimientos No Funcionales:

- *Requerimientos de Interfaz:* Que describen como el sistema va a interactuar con el usuario, el ambiente y otros sistemas.
- *Requerimientos de Performance:* Que describen las limitaciones de rendimiento del sistema que involucran:
 - Límites de tiempo / espacio: como la carga de trabajo, tiempo de respuesta, rendimiento y el espacio de almacenamiento disponible.
 - Fiabilidad que implica la disponibilidad de los componentes y la integridad de la información mantenida y suministrada.
 - Seguridad, tales como los flujos de información permisibles.
 - Supervivencia o la resistencia al fuego, las catástrofes naturales.
- *Requerimientos de Operación:* Son las limitaciones físicas (tamaño, peso), la disponibilidad de personal, las consideraciones de nivel, la accesibilidad del sistema para el mantenimiento, etc.
- *Requerimientos del ciclo de vida:* Se pueden clasificar en dos subcategorías:
 - Calidad del diseño: se mide en términos tales como mantenibilidad, capacidad para ser mejorable, la portabilidad.
 - Límites en materia de desarrollo, tales como las limitaciones de tiempo de desarrollo, disponibilidad de recursos, normas metodológicas, etc.
- *Requerimientos económicos:* los costos inmediatos y/o a largo plazo.
- *Requerimientos políticos.*

y) **FURPS+**: desarrollado en Hewlett-Packard, fue elaborado por Grady y Caswell, citado en [32]: Representa un modelo de clasificación de atributos de calidad del software, o Requerimientos No Funcionales. Es un acrónimo de las diferentes categorías en Inglés (Functionality, Usability, Reliability, Performance, Soportability)

- *Functionality*: Conjunto de características, capacidades, la generalidad, Seguridad.
- *Usability*: Factores humanos, Estética, Coherencia, Documentación
- *Reliability*: Frecuencia/Severidad de la falla, la recuperabilidad, la previsibilidad, La precisión, tiempo medio entre fallas.
- *Performance*: Velocidad, Eficiencia, Consumo de recursos, Rendimiento, Tiempo de respuesta.
- *Supportability*: Capacidad de prueba, Extensibilidad, Adaptabilidad, Mantenimiento, Compatibilidad, Capacidad de configuración y Capacidad de servicio, facilidad de instalación, Localizable, Portabilidad.

El signo “+”: Son requerimientos adicionales que regularmente son restricciones.

Un análisis de los siete esquemas de clasificación presentados permite ver que son inconsistentes unos con otros, en la cantidad de niveles que tiene cada estructura, y en los conceptos agrupados en cada categoría o característica. Por ejemplo, en algunos casos la *performance* es un atributo o subcaracterística, en otros casos es una categoría es si misma que contiene a un subconjunto de atributos o subcaracterísticas y hay esquemas en los cuales no es mencionado como tal. A continuación la tabla 2.1 muestra un resumen de observaciones similares para otros ejemplos como son: *usabilidad*, *seguridad*, *mantenimiento*.

Referencia	Autor de Esquema de Clasificación	Atributos de calidad			
		Performance	Usabilidad	Seguridad	Mantenimiento
r)	Mamani	-	Atributo (Depende de la Categoría Calidad)	Atributo (Depende de la Categoría Calidad)	Atributo (Depende de la Categoría Calidad)
s)	Kotonya y Sommerville	Clasificación (RNF de Producto)	Clasificación (RNF de Producto)	-	-
t)	Boehm	-	-	-	Categoría (Agrupa otras características)
u)	Cysneiros	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)
v)	IEEE 810	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de Requerimientos de Interfaz Externa)	Característica (Depende de Atributos de sistema)	Característica (Depende de Atributos de sistema)
w)	ISO/IEC 9126	Subcaracterística (Depende de una característica)	Característica de Calidad	Subcaracterística (Depende de una característica)	Característica de Calidad
x)	Roman	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de Requerimientos de Interfaz)	Atributo (Depende de Requerimientos de Performance)	Atributo (Depende de Requerimientos de Ciclo de Vida)
y)	FURPS	Característica (Agrupa Subcaracterísticas)	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de una Categoría)	Atributo (Depende de una Categoría)

TABLA 2.1: Ejemplos de discrepancias en esquemas de clasificación de RNF.

Otro punto importante a destacar es que algunos esquemas plantean una clasificación basados en el concepto de calidad y atributos de calidad, mientras que otros, consideran atributos de calidad y aportan otras nuevas formas de agrupación, por ejemplo, Roman en x) presenta las categorías: *Requerimientos del ciclo de vida*, *Requerimientos económicos* y *Requerimientos políticos*, Antoya y Sommerville en s) consideran *Requerimientos Legales* dentro de una agrupación de *Requerimientos externos*. Cada esquema de clasificación está ligado al concepto o definición de Requerimientos No Funcionales de su autor.

Quizás no hay que seguir buscando respuestas a la pregunta *¿Qué son los Requerimientos No Funcionales?* y generar definiciones nuevas. Quizás se debería poner foco en trabajar sobre la unificación de criterios de la terminología en las definiciones existentes y recién a partir de ahí, analizar las diferentes categorías de clasificación propuestas a fin de armar un esquema de referencia que agrupe los conceptos y tipos de Requerimientos No Funcionales.

2.2 Conceptos a aplicar en este trabajo

De lo expuesto en la sección 2.1 surgen los siguientes interrogantes:

¿Cómo se aplican estos conceptos en enfoques metodológicos de desarrollo de aplicaciones web?

¿Cuál o cuáles tipos de Requerimientos No Funcionales son significativos en el desarrollo de aplicaciones web?

El crecimiento en el uso aplicaciones web de los últimos años, impulsó el surgimiento de metodologías para el desarrollo de aplicaciones web. Para Escalona y Koch [1], la mayoría de estos enfoques están orientados a actividades de diseño dentro de su ciclo de vida, contemplando en menor escala, o no incluyendo, tareas de ingeniería de requerimientos, testeo y administración de la calidad.

Tanto en los enfoques de desarrollo tradicional (no web) como en el desarrollo de aplicaciones web, los requerimientos juegan un rol relevante, las consecuencias de partir de requerimientos especificados en forma ambigua, vaga o incorrecta, pueden reflejarse en modelos de diseño y arquitecturas de software inadecuados, demoras en la planificación por la aparición de gran cantidad de fallas o defectos que deben ser corregidos, hasta la no aceptación de los usuarios.

Actualmente la Ingeniería de Requerimientos proporciona numerosas técnicas y herramientas para identificar, describir, validar y gestionar requerimientos, sin embargo, estos no son aplicados muy a menudo por los ingenieros de software. En particular en el desarrollo de aplicaciones web las que se construyen en forma ad-hoc, a pesar de que por la

complejidad de las aplicaciones web de hoy se requiere un enfoque más sistemático, la madurez del proceso de Ingeniería de Requerimientos parece ser insuficiente, específicamente con el tratamiento de los Requerimientos No Funcionales.

Durante el proceso de elicitación, los Requerimientos No Funcionales podrían aparecer junto a los requerimientos funcionales, por ejemplo en forma de:

- Políticas de negocio o externas al negocio (económicas, culturales, legales entre otras).
- Presupuesto disponible para el desarrollo del software.
- Fecha de disponibilidad del software importante para el negocio.
- Paradigmas de programación, aspectos de distribución, tecnología de autorización.
- Características del proceso de desarrollo o ciclo de vida a ser aplicado en el desarrollo del software (ágil, secuencial, iterativo, nivel de especificación del de proceso).
- Conocimiento y experiencia del equipo de desarrollo (relacionados al proceso de desarrollo, lenguajes de modelado y programación).
- Infraestructura existente en el entorno de implementación (hardware, software, u otros sistemas).
- Límites en tiempos de respuesta, rendimiento y el espacio de almacenamiento disponible.
- Disponibilidad de los componentes.
- Tipos de usuarios autorizados, o no, a realizar ciertas tareas y cómo se implementarán los mecanismos de seguridad, entre otros;

La lista anterior muestra que no es una tarea sencilla para un ingeniero de software identificarlos, evaluarlos y tipificarlos, si consideramos la falta de consenso en las definiciones y categorías expuestas en la sección anterior, más aún si el enfoque metodológico que se está utilizando para de desarrollo de la aplicación web carece de lineamientos, técnicas y/ o herramientas específicas de soporte a la etapa de ingeniería de requerimientos.

Para Cysneiros en [7] [30] los enfoques de desarrollo de software que tratan sobre Requerimientos No Funcionales, pueden clasificarse como “Orientados al Proceso” y “Orientados al Producto”.

Los orientados a productos, se enfocan en establecer el grado de cumplimiento del software con los Requerimientos No Funcionales detectados, mientras que los orientados al proceso, están enfocados en el proceso de desarrollo de software, ayudan a los ingenieros de software a buscar alternativas para cumplir los Requerimientos No Funcionales a medida que se desarrolla el software, lo cual permitirá además justificar decisiones de diseño.

A continuación, para asentar las bases conceptuales a aplicar en el desarrollo de este trabajo, se parte de las definiciones expuestas por Cysneiros en [7], descritas en la sección anterior en los ítems p) y q), que considera a los Requerimientos No Funcionales como requerimientos de calidad y como restricciones.

Requerimientos No Funcionales como requerimientos de calidad:

Para Cysneiros en p) los Requerimientos No Funcionales “*son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras*”...

Dentro de los tipos de requerimientos que consideran importantes para las aplicaciones web, Kappel, Pröll, Reich y Retschitzegger en [5]; mencionan a los *requerimientos de calidad*: como aquellos que describen el nivel de calidad del servicio y especifican propiedades relevantes que debe reunir el sistema en términos de las características del estándar de calidad internacional ISO/IEC 9126 [34], y aunque Kappel, Pröll, Reich y Retschitzegger en [5] hacen mención de la existencia de investigaciones que pretenden extender el modelo básico de ISO/IEC 9126 para las aplicaciones Web, presentan a la usabilidad, la performance y la seguridad como los aspectos críticos para las aplicaciones web.

La tabla 2.2 muestra los conceptos expuestos, con la finalidad de poner el marco de atributos de calidad comparables para aplicar en este trabajo.

	Cysneiros [7]	Kappel [5]	ISO/IEC 9126-1 [34]
Características o atributos de calidad	Precisión (*)	Usabilidad	Usabilidad
	Usabilidad	Performance (**)	Eficiencia
	Seguridad (*)	Seguridad (*)	Funcionalidad
	Rendimiento (**)		Confibilidad
	Confiabilidad		Mantenibilidad
	Performance (**)		Portabilidad

* Subcaracterística de la Funcionalidad

** Subcaracterística de la Eficiencia

TABLA 2.2: Requerimientos No Funcionales como requerimientos de Calidad.

En este trabajo se considera los Requerimientos No Funcionales como requerimientos de calidad y para ellos se contemplarán las seis características del estándar de calidad internacional ISO/IEC 9126 [34], descrito en la sección 2.1., ya que este, incluye los atributos mencionados en las definiciones de Cysneiros en [7] y Kappel en [5] y aporta dos características no consideradas por ellos como lo son: la Mantenibilidad y la Portabilidad.

Requerimientos No Funcionales como restricciones:

Para Cysneiros en q) *los Requerimientos No Funcionales son las restricciones globales sobre el sistema, un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue...*

De acuerdo a lo presentado en la sección 2.1, no hay ejemplos específicos expresados por Cysneiros o unicidad de criterio en la bibliografía existente de otros autores, para definir a las “restricciones globales del sistema”, las “restricciones sobre un requerimiento funcional”, las “restricciones de proceso de desarrollo” y las “restricciones del proceso de despliegue”; este trabajo se enfocará solamente en las “*restricciones del proceso de desarrollo*” para evaluar el nivel de completitud y de especificación de los procesos para desarrollar aplicaciones web, dado que la incompletitud de los mismos hace que se descuiden aspectos relevantes para el tratamiento de los Requerimientos No Funcionales.

Se proponen a continuación las restricciones de proceso de desarrollo a considerar en este trabajo:

Alcance del proceso de desarrollo: relacionada a las fases del ciclo de vida que presenta el proceso, el cual puede ser

Parcial: significa, que el proceso de desarrollo no contempla y describe fases para todo el ciclo de vida de desarrollo.

Completo: significa, que el proceso de desarrollo contempla y describe las fases para todo el ciclo de vida de desarrollo.

Nivel de detalle de la especificación del proceso: relacionada a la profundidad del detalle de la especificación del proceso, las actividades, las técnicas utilizadas en cada fase y los productos generados en las mismas. El cual puede ser:

Parcial: Significa, que el proceso de desarrollo describe solamente las actividades a ser aplicadas en cada fase, pero no describe técnicas a utilizar ni ofrece detalles o templates de los productos que deben ser producidos en cada fase.

Completo: Proceso especificado: significa, que el proceso de desarrollo describe las actividades, las técnicas a ser aplicadas y los productos que deben ser producidos en cada fase.

Los conceptos presentados en esta sección serán aplicados en los capítulos 4 y 5.

3. Técnicas utilizadas en la Ingeniería de Requerimientos

La Ingeniería de Requerimientos “es el proceso sistemático de desarrollar requerimientos a través de un proceso cooperativo e iterativo de analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación y chequear la precisión de la comprensión obtenida” [35].

La ingeniería de requerimientos comienza con la *elicitación* que es “el proceso de adquisición de todo el conocimiento relevante necesario para producir un modelo de requerimientos del dominio del problema” [35]. La esencia de la elicitación es la comprensión del dominio del problema, los ingenieros de software adquieren el conocimiento aplicando diferentes técnicas de elicitación a partir de los expertos de dominio o de diferentes fuentes (literatura sobre el dominio, software existente en el dominio, aplicaciones de software similares en otros dominios, estándares de desarrollo de software, entre otras). El modelo de requerimientos inicial, generado en la elicitación sirve de entrada a la *especificación* de requerimientos.

La especificación de requerimientos tiene como objetivo producir un modelo de especificación de requerimientos formal. Desarrollar una especificación consiste en la construcción de requerimientos en términos conceptuales (abstracciones, suposiciones, restricciones sobre el dominio de la aplicación); la especificación de requerimientos incluye los requerimientos funcionales, no funcionales y modelos de contexto, la cual será validada por el usuario en el proceso de *validación*.

La validación de requerimientos “es el proceso de certificar la corrección del modelo de requerimientos contra las intenciones de los usuarios” [35]. La actividad principal de este proceso es detectar y corregir errores del modelo de especificación de requerimientos, en la fase de requerimientos y no cuando el software este desarrollado. El modelo de requerimientos validado debe especificar una solución de software que satisface las necesidades del usuario.

A continuación se describen un grupo de técnicas utilizadas por la ingeniería de requerimientos, para la Elicitación, Especificación y Validación de requerimientos extraídas para su análisis a partir de la revisión de técnicas de ingeniería de requerimientos presentada por Escalona y Koch en [1].

Técnicas de Elicitación:

Técnica	Entrevistas
Descripción	Es una de las técnicas más utilizadas para tomar conocimiento del problema. Existen diversos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura para ofrecer guías para su correcta realización. Generalmente, la estructura de la entrevista consiste de tres pasos: la identificación de los entrevistados, la preparación de la entrevista, la realización de la entrevista y la documentación de los resultados.
Comentarios	La entrevista no es una técnica sencilla de aplicar, porque requiere que el entrevistador sea experimentado y tenga capacidad no solo para elegir bien a los entrevistados y obtener de ellos toda la información posible en un acotado período de tiempo, sino que además conozca técnicas o herramientas para la preparación de la entrevista y documentar los resultados.

Técnica	JAD (Joint Application Development/Desarrollo conjunto de aplicaciones)
Descripción	Es una alternativa a las entrevistas. Esta técnica consiste en una práctica de grupo que se desarrolla durante varios días y en la que participan varios roles del equipo de desarrollo (analistas, usuarios, administradores del sistema) y clientes. Se basa en cuatro principios: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y por último una filosofía de documentación WYSIWYG (<i>What You See Is What You Get, lo que ve es lo que obtiene</i>). En otras palabras durante la entrevista se trabajará sobre lo que se generará. Después una fase de preparación para poder aplicar la técnica, el equipo de trabajo se reúne en varias oportunidades; en cada reunión se establecen los requerimientos de alto nivel, el contexto del problema y la documentación. Durante cada reunión se debate en grupo sobre estos temas llegando a una serie de conclusiones que se documentan, de esta forma en cada reunión se van concretando más las necesidades que deberá soportar el sistema.
Comentarios	Esta técnica presenta ventajas frente a las entrevistas tradicionales ya que permite un ahorro de tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado. Sus desventajas son por un lado, que requiere de grupos de participantes bien integrados y organizados y por otro requiere disponibilidad de tiempo de los mismos.

Técnica	Brainstorming
Descripción	Es una técnica para reunión de grupos de personas cuyo objetivo es que los participantes expongan sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas a participar en estas reuniones no debe ser muy numeroso (máximo 10 personas) y una de ellas debe asumir el rol de moderador de la reunión, pero sin carácter de controlador.
Comentarios	La ventaja del Brainstorming como técnica de captura de requerimientos es que es sencilla de usar y de aplicar. La desventaja es que si bien ayuda a obtener una visión general de las necesidades del sistema, por lo general no sirve para identificar los detalles concretos del sistema, por lo que suele aplicarse solo en las primeras reuniones.

Técnica	Concept Mapping (Mapas de Concepto)
Descripción	Esta técnica se desarrolla con el equipo del proyecto y el usuario final, consiste en modelar los conceptos del sistema que está siendo construido mediante grafos; los vértices representan los conceptos y las aristas representan posibles relaciones entre dichos conceptos.
Comentarios	Es una técnica muy usada en la Ingeniería de Requerimientos porque que es de fácil comprensión para el usuario, más aún si es elaborada en el lenguaje de éste. Una desventaja, es que en algunos casos puede resultar ambigua si no se acompaña de una descripción textual adicional, sobre todo en sistemas complejos.

Técnica	Sketches y Storyboards
Descripción	Esta técnica consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard). Es comúnmente usada por los diseñadores gráficos de aplicaciones web.
Comentarios	Es una técnica de fácil aplicación.

Técnica	Casos de Uso
Descripción	Aunque inicialmente se desarrollaron como técnica para la definición de requerimientos, algunos autores proponen casos de uso como técnica para la captura de requerimientos. Los casos de uso permiten mostrar el alcance del sistema (requerimientos funcionales expresados como casos de uso). Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una función que produce un resultado de valor para el / los actores del mismo. Un actor es un rol, es un elemento externo (personas, otros sistemas, etc.) que interactúan con el sistema. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.
Comentarios	La ventaja de los casos de uso, es que el modelo de casos de uso resulta de fácil comprensión para el usuario o cliente y para el equipo de desarrollo, la desventaja es que muchas veces carecen de la precisión necesaria, por lo que es necesario complementarlos con otras técnicas como diagramas de actividades o deben ser acompañados por información textual adicional.

Técnica	Cuestionarios y Checklists
Descripción	Esta técnica consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas o limitadas por opciones en el propio cuestionario o checklist. Este cuestionario puede ser completado durante una entrevista o puede ser utilizado para recoger/ capturar información de un grupo de personas en forma independiente a una entrevista.
Comentarios	Su aplicación requiere que el ingeniero de software conozca el dominio del problema en el que está trabajando y la aplicación a ser desarrollada a fin de preparar el cuestionario y checklist.

Técnica	Comparación de terminología
Descripción	Uno de los problemas que surge durante la elicitación de requerimientos es que los usuarios, expertos del dominio, no llegan a entenderse con los ingenieros de software debido a problemas de comunicación generados por no usar el mismo lenguaje. Esta técnica consiste en identificar el uso de palabras diferentes para referirse a los mismos conceptos (correspondencia) y palabras similares para referirse a diferentes conceptos (conflictos) o cuando no hay concordancia exacta en el vocabulario y los conceptos (contraste).
Comentarios	Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo.

Especificación de requerimientos: Dentro de las técnicas utilizadas se destacan las siguientes

Técnica	Lenguaje natural
Descripción	Consiste en definir los requerimientos en lenguaje natural sin usar reglas para ello. A pesar de que es criticada por su ambigüedad, es una de las técnicas más usadas para la especificación de requerimientos.
Comentarios	Es de fácil aplicación, pero resulta una técnica ambigua para la definición de los requerimientos.

Técnica	Glosario y ontologías
Descripción	El equipo de desarrollo de software puede ser muy diverso, esta diversidad trae aparejada la necesidad de establecer un marco de terminología común a ser utilizado en el proyecto. En las aplicaciones web el equipo de desarrollo suele ser más interdisciplinario que en los equipos para desarrollo de aplicaciones no web, por esta razón son muchas las metodologías que proponen desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema. En esta misma línea se propone también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos.
Comentarios	Son de gran utilidad para establecer la terminología común y los conceptos relevantes a ser utilizado en el proyecto.

Técnica	Plantillas o patrones
Descripción	Esta técnica consiste en describir los objetivos y requerimientos usando el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida, las secciones de esta plantilla son completadas por el equipo de desarrollo usando lenguaje natural y la terminología del usuario.
Comentarios	Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece, sin embargo si el nivel de detalle requerido es demasiado detallado, el trabajo de completar las plantillas y mantenerlas se dificulta.

Técnica	Escenarios
Descripción	La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. La representación de los escenarios puede variar de un autor a otro, pueden ser textuales o adoptar representaciones gráficas en forma de diagramas de flujo o casos de uso. La notación de escenarios está integrada en muchas técnicas de análisis orientados a objetos.
Comentarios	El análisis de los escenarios, puede ofrecer información importante sobre las necesidades funcionales de sistema. Cuando son representados en forma textual, pueden sumar ambigüedad a la definición de los requerimientos.

Técnica	Casos de uso
Descripción	Esta técnica tiene gran aceptación tanto en la elicitación (ver técnicas de Elicitacion) como en la definición de requerimientos. Se ha propuesto como técnica básica del proceso RUP para describir las actividades que debe realizar el sistema para producir un resultado de valor para un actor. Dado que varios autores consideran que los casos de uso pueden resultar ambiguos a la hora de definir los requerimientos, las metodologías que lo utilizan proponen agregar a los mismos, descripciones textuales basadas en plantillas, diccionarios de datos o representaciones gráficas para eliminar su ambigüedad.
Comentarios	La desventaja de esta técnica es que resulta ambigua cuando definen requerimientos complejos; por lo que es necesario complementarlos con descripciones textuales basadas en plantillas, diccionarios de datos o representaciones gráficas.

Técnica	Lenguajes Formales
Descripción	En contraste a las descripciones en lenguaje natural, hay técnicas que utilizan lenguajes formales para describir los requerimientos del sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción formal, han sido aplicadas en la ingeniería de software desde hace años, sin embargo, utilizarlas resulta una actividad compleja y son de difícil comprensión para el usuario/cliente.
Comentarios	La ventaja es que es una de las formas menos ambigua para representar los requerimientos, lo cual permite la aplicación de técnicas de verificación automatizadas. La desventaja más notable de los Lenguajes Formales es que no favorecen la comunicación entre el usuario/cliente y el ingeniero de software.

Validación de requerimientos: El paso siguiente, una vez que los requerimientos fueron definidos es validarlos, dentro de las técnicas apropiadas para ello se encuentran las siguientes:

Técnica	Reviews o Walk-throughs
Descripción	Esta técnica consiste en la lectura y corrección de la documentación que contiene los requerimientos especificados y los modelos generados. Permite validar únicamente la correcta interpretación de la información transmitida. La verificación de consistencia de la documentación o información faltante demanda de métodos más sofisticados.
Comentarios	Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

Técnica	Auditorías
Descripción	La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra checklists predefinidos o definidos al comienzo del proceso.
Comentarios	La revisión de la información y resultados puede ser parcial o total, esto depende de la preparación de la auditoria y el nivel de detalle de los checklists utilizados para aplicar esta técnica.

Técnica	Matrices de trazabilidad
Descripción	Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requerimientos del mismo. Es necesario ir viendo qué objetivos cubre cada requerimiento, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.
Comentarios	Permite identificar la consistencia de los requerimientos especificados con los objetivos del sistema. Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

Técnica	Prototipos
Descripción	Es una técnica que consiste en construir herramientas basadas en la especificación de requerimientos. Los prototipos por lo general se construyen a partir de un conjunto parcial de los requerimientos funcionales del sistema, pero permiten al usuario tener una idea global de la estructura de la interfaz de usuario del sistema. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.
Comentarios	Permite validar un subconjunto de requerimientos de los requerimientos especificados. Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

Herramientas de Administración de requerimientos: Las herramientas propuestas para la administración de requerimientos se presentarán en el capítulo siguiente, junto a la metodología a la que dan soporte.

4. Metodologías de desarrollo de aplicaciones Web y los requerimientos

4.1 Aplicaciones Web

Una aplicación Web es un sistema de software basado en tecnologías y estándares del World Wide Web Consortium (W3C), que proporciona recursos web específicos, tales como los contenidos y servicios a través de una interfaz de usuario, el navegador Web [5]. Dicho en otras palabras una aplicación web en la actualidad es un complejo sistema de software que se codifica en un lenguaje soportado por navegadores web, donde la característica distintiva más relevante comparada con aplicaciones de software tradicionales, es la forma en la cual la aplicación web es usada.

Existen diferentes tipos de aplicaciones web, unas pueden ser de carácter puramente informativas otras pueden manejar aplicaciones de comercio electrónico, entre otras, Kappel, Pröll, Reich y Retschitzegger en [5] presentan una categorización de aplicaciones web organizada en función a la historia y evolución del grado de complejidad de las mismas:

1. Sitios Web centrados en documentación.
2. Aplicaciones Web interactivas.
3. Aplicaciones web transaccionales.
4. Aplicaciones web basadas en workflow
5. Aplicaciones web colaborativas
6. Web sociales
7. Aplicaciones web orientadas a portales
8. Aplicaciones web ubiquitous, basados en la localización
9. Web semánticas

Desde el punto de vista de la ingeniería del software el desarrollo de las aplicaciones web representan un nuevo dominio de aplicación [5] si bien tiene similitud con el desarrollo de las aplicaciones tradicionales, la característica especial de las aplicaciones web hace surgir la necesidad de una adaptación de los enfoques de desarrollo existentes o quizás el desarrollo de enfoques metodológicos nuevos. El desarrollo de aplicaciones web tiene que

contemplar varias características que difieren respecto del desarrollo de otros tipos de aplicaciones (no web). Por ejemplo:

- Los diferentes tipos de stakeholders involucrados en el proceso de desarrollo (cliente, usuarios, analistas, diseñadores gráficos, desarrolladores, expertos en multimedia y seguridad, marketing, etc.)
- La estructura de navegación, en la cual debe ser clara e intuitiva para el usuario.
- La interfaz gráfica de usuario, cuyo diseño debe tener en cuenta, aspectos de marketing y multimedia,

Los dos últimos si bien no solo deben ser tratados en forma diferente en la etapa de diseño, sino que deberían considerarse como parte de la aplicación de la ingeniería de requerimientos.

Si bien actualmente existen varias metodologías de desarrollo web en vigencia, en la sección siguiente, se analizarán un subconjunto de enfoques metodológicos considerados a partir del estudio comparativo realizado por Escalona y Koch en [1], a los efectos de identificar como estos enfoques, están aplicando la ingeniería de requerimientos para el tratamiento de los Requerimientos No Funcionales que son el objetivo de este trabajo.

4.2 Metodologías de desarrollo web y los Requerimientos No Funcionales

El trabajo de Escalona y Koch [1], aporta una comparación de diez metodologías de desarrollo web, destacando entre otros aspectos, técnicas de ingeniería de requerimientos y tipos de requerimientos que manejan. Respecto a los Requerimientos No Funcionales se limita a mencionar cuales metodologías detectan los mismos.

En este capítulo se analizan puntualmente los enfoques que mencionan Requerimientos No Funcionales en [1] a efectos de determinar cuál es el concepto que estas

metodologías tienen de los distintos tipos de Requerimientos No Funcionales y las técnicas de detección, especificación, validación y gestión de requerimientos que aplican las mismas.

La mayoría de las metodologías comparadas en el trabajo de Escalona y Koch [1] tienen una clasificación de requerimientos que difiere entre una y otra, los autores propusieron una clasificación general de requerimientos, a fin de poder realizar una comparación sobre cuales tipos de requerimientos utiliza cada enfoque metodológico.

A continuación se describe la clasificación realizada en [1] para los requerimientos de las aplicaciones Web:

- Se identifica por un lado, los Requerimientos Funcionales: *como capacidades que el sistema debe exhibir para resolver un problema*. Estos a su vez fueron subclasificados en:
 - Requerimientos de datos, *los cuales establecen “como” la información es almacenada y administrada por la aplicación. También son denominados requerimientos de contenido, requerimientos conceptuales o requerimientos de almacenamiento de información.*
 - Requerimientos de Interface de usuario, *que describen “como” el usuario va a interactuar con la aplicación Web. También conocidos en algunas propuestas como requerimientos de interacción o requerimientos de usuario.*
 - Requerimientos de Navegación, *que representan la navegación que los usuarios requieren a través de la aplicación Web.*
 - Requerimientos de Personalización, *describen “como” la aplicación Web debe adaptarse dinámicamente, dependiendo del usuario que interactúe con ella.*
 - Requerimientos Transaccionales, *describen “que” es lo que la aplicación Web tiene que realizar internamente sin considerar interfaces y aspectos de interacción. Son conocidos además como requerimientos de servicios.*

- Y los Requerimientos No Funcionales, *los cuales son considerados como restricciones a la solución, los autores mencionan como ejemplos: los requerimientos de portabilidad, los requerimientos de reutilización, requerimientos de usabilidad, requerimientos de disponibilidad, requerimientos de performance etc.*

La tabla 4.1, fue extraída del análisis comparativo realizado en [1] por Escalona y Koch. Presenta un resumen de los diferentes tipos de requerimientos que son tratados por cada una de las metodologías estudiadas. Los enfoques metodológicos están ordenados cronológicamente y de esta forma puede verse la evolución e importancia que fueron adquiriendo los requerimientos en el entorno de la Web.

	Req. Datos	Req. Interfaz usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
WebML	✓	✓		✓		✓
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

TABLA 4.1: Requerimientos tratados por cada metodología

La columna titulada “Req. No funcionales” de la tabla 4.1, muestra cuales de las metodologías para desarrollo de aplicaciones web estudiadas, identifican Requerimientos No Funcionales, de ahí que: WSDM, HFPM, UWE, WebML, NDT, DDDP serán a partir de ahora el foco de análisis.

La figura 4.1, es una representación gráfica que surge a partir de la tabla 4.1; ilustra la relación entre las metodologías Web y los requerimientos. El eje vertical representa el tiempo y permite identificar el año en el que surgieron los diferentes enfoques metodológicos; el eje horizontal representa a los requerimientos clasificados como funcionales y no funcionales. En el espacio se destaca la relación de las metodologías y los requerimientos manejados por cada una, se han señalado con azul aquellas que son el foco de nuestro estudio.

Durante el desarrollo de este trabajo se pretende evolucionar la tabla 4.1 detallando los tipos de Requerimientos No Funcionales mencionados o tratados en cada enfoque.

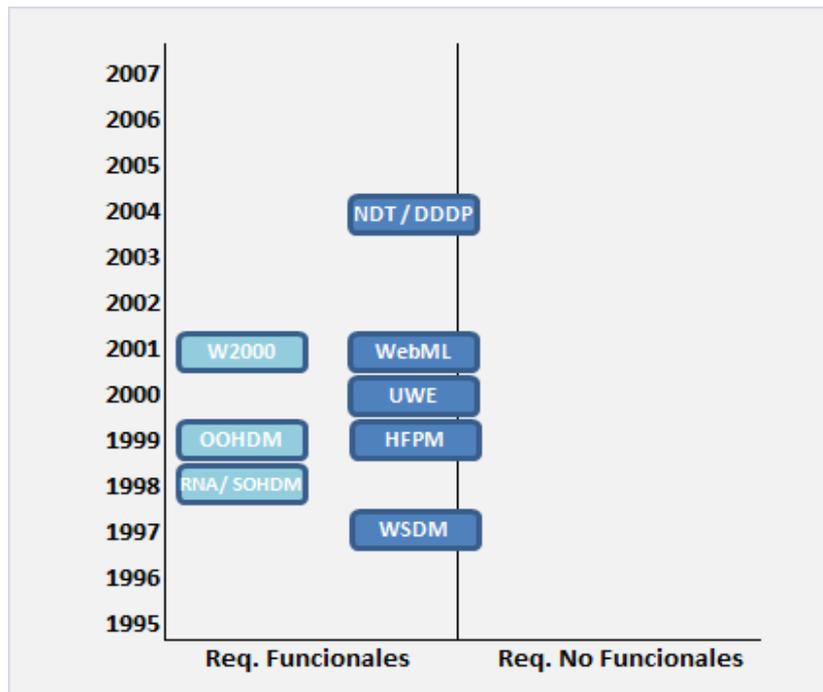


FIGURA 4.1: Metodologías para aplicaciones Web y los Requerimientos

Análisis de los enfoques metodológicos para desarrollo de aplicaciones Web

El análisis de las metodologías WSDM, HFPM, UWE, WebML, NDT, DDDP se realizará en base a la siguiente estructura de presentación:

- ¿Cuáles son las características relevantes?: *En esta sección se enumeran los aspectos y o cualidades significativas y representativas del enfoque metodológico.*
- Descripción de las fases del proceso: *En esta sección se presenta un grafico que ilustra las fases del ciclo de vida del enfoque metodológico, se describen las actividades relevantes y/o productos generados en cada una de ellas.*

Dado que los diferentes enfoques metodológicos presentados en este trabajo utilizan los términos fases o etapas en la descripción de su ciclo de desarrollo, proponemos el término “Fase” a fin de unificar términos comparables.

- *¿Qué tipos de Requerimientos No Funcionales identifica?: En esta sección se detallan los diferentes tipos de Requerimientos No Funcionales que se identifican en las diferentes fases del ciclo de vida del enfoque metodológico.*
- *¿Qué técnicas de análisis y administración de requerimientos emplea?: En esta sección se detallan las técnicas de elicitación, especificación, validación y administración de requerimientos que menciona y emplea cada enfoque en la fase de requerimientos. Se pone foco en evaluar si el enfoque utiliza técnicas específicas para elicitación, especificación y validación de Requerimientos No Funcionales.*

La presentación de cada enfoque sigue el orden cronológico de publicación, lo cual permite visualizar y analizar cómo fue evolucionando la ingeniería de requerimientos aplicada a metodologías web.

4.2.1 WSDM: Web Site Design Method

Este enfoque fue propuesto en el año 1997 por De Troyer y Leune [9], presenta un tratamiento interesante de la visión de los usuarios; plantea el diseño de una aplicación Web de tipo Kiosco” a partir de los potenciales grupos de usuarios.

¿Cuáles son las características relevantes?

- El proceso consiste en cuatro fases.
- Sirve para desarrollar kioscos web, es decir, aplicaciones que muestran una determinada información al usuario y les permiten navegar hacia ella.

- El proceso de definición de requerimientos se centra en detectar los perfiles de usuarios potenciales mediante dos tareas:
 - Identificar y clasificar usuarios mediante el estudio del entorno
 - Describir las características de los grupos de usuarios detectados y sus necesidades de información, lo cual se denomina “perspectivas”.
- El diseño conceptual comienza a partir de las perspectivas detectadas.
- La implementación puede ser generada una vez que el diseño conceptual está completo.
- “Es un enfoque centrado en el usuario esto no es lo mismo que a un enfoque conducido por el usuario”... [9].

La figura 4.2 muestra un diagrama de las fases y subfases que propone WSDM.

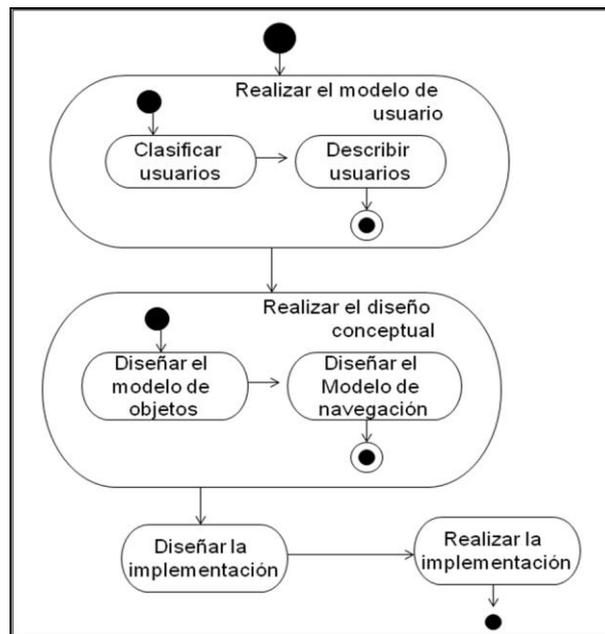


FIGURA 4.2: Diagrama de las fases de WSDM

A continuación se presentan y describen en forma general las actividades desarrolladas en cada una de las fases del proceso WSDM.

Realizar el modelo de usuario: Esta Fase se concentra en los potenciales usuarios de la aplicación web, a fin identificarlos, clasificarlos y luego describir sus características y los requerimientos de información, para lo cual cuenta con dos subfases:

- Clasificar usuarios: WSDM propone estudiar del entorno de la organización o los procesos de negocio para los cuales se implantará la aplicación web a fin de identificar las actividades y las personas (potenciales usuarios) involucradas en cada una. Luego describe las relaciones entre usuarios y las actividades que realizan los mismos; para la representación gráfica de estas relaciones WSDM propone mapas de conceptos de roles y actividades.
- Describir usuarios: Los grupos de usuarios detectados en la subfase anterior son descritos con más detalle. Mediante un diccionario de datos, se describen los requerimientos de información, requerimientos funcionales y requerimientos de seguridad, requerimientos de usabilidad para cada grupo de usuarios.

En el trabajo de [9] algunos ejemplos de las características que son analizadas para cada grupo de usuarios son: nivel de experiencia con sitios web en general, frecuencia de uso de esta la aplicación web, mandatorio/ discreto uso del sitio web, Motivación para el uso de la aplicación web, lenguajes, nivel de educación, habilidades, edad.

Se hace hincapié en este punto el cual es relevante para este trabajo.

Realizar el diseño conceptual: Durante esta fase se realiza el modelo conceptual de los requerimientos detectados en la fase anterior y se realiza el diseño de la navegación que describe como los diferentes grupos de usuarios podrán navegar a través de la aplicación web. Para ello plantea las siguientes subfases:

- Diseñar modelo de objetos: El propósito de esta fase es desarrollar un modelo de objetos conceptual para los diferentes grupos de usuarios, los mismos son llamados MOU (Modelos de objetos usuario). El modelo conceptual describe diferentes tipos de objetos, las relaciones, restricciones entre ellos y describe el comportamiento de los objetos.

Un punto a destacar es que un grupo de usuarios puede tener diferentes perspectivas generando así diferentes requerimientos de usabilidad. Para este caso se pueden diseñar MPO (Modelos de Perspectiva de Objetos).

- Diseñar modelo de navegación: El modelo de navegación que se genera en esta fase muestra un número de pistas de navegación por cada MPO. Dicho en otras palabras, expresa como un usuario de un grupo en particular puede navegar hacia la información que necesita. Se modela a través de componentes (de información, de navegación y externos) y links o enlaces que conectan los mismos. Un modelo de navegación cuenta de tres capas: Capa de contexto es el nivel más alto de las pistas de navegación, la capa de navegación que proporciona las diferentes formas para acceder a la información (conecta la capa de contexto con la información) y la capa de Información contiene los componentes de información.

Puede existir más de un modelo de navegación, dependiendo de los grupos de usuarios detectados durante la primera fase.

Diseñar la implementación: Durante esta fase se realiza el diseño de interface (look and feel) de la aplicación web para cada uno de los grupos de usuarios definidos. El resultado es el Modelo de implementación; el cual ayuda a que los usuarios tengan un modelo mental de la aplicación.

Una aplicación Web construida con WSDM consiste de un número de pistas de navegación para cada perspectiva de usuario identificada.

Es importante destacar que el diseño de la implementación puede depender de las limitaciones del lenguaje de programación seleccionado.

Realizar la implementación: Durante esta fase se realiza la codificación del modelo obtenido en la fase anterior.

Se deben tener en cuenta las consideraciones y buenas prácticas para mejorar la mantenibilidad de la información que la aplicación web brindará a los diferentes usuarios, a fin de que conservarla actualizada.

¿Qué tipos de Requerimientos No Funcionales identifica?

- Usabilidad - (Relevamiento)
- Seguridad - (Relevamiento)
- Mantenibilidad - (Diseño - Implementación)
- Restricciones: De proceso, Lenguaje de programación.

Las restricciones y/o limitaciones del lenguaje de programación y necesidades de Mantenibilidad surgen en el momento de diseñar la implementación e implementación propiamente dicha.

El Modelo de Implementación puede depender de las limitaciones del lenguaje de programación.

¿Qué técnicas de Análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/ Captura de requerimientos: Entrevistas.
- Especificación/Definición de requerimientos: Utiliza lenguaje natural.
- Validación: No menciona el empleo de técnicas para la validación de requerimientos.

Requerimientos No Funcionales

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación: No especifica.

Administración de requerimientos: No menciona técnicas o herramientas.

4.2.2 HFPM: Hypermedia Flexible Process Modeling

Propuesto por Olsina [10] en año 1998, este enfoque metodológico presenta fases para todo el proceso de desarrollo; define un conjunto de actividades que van desde el análisis hasta el desarrollo de documentación y el mantenimiento.

¿Cuáles son las características relevantes?

- El proceso detallado cubre todo el ciclo de vida de desarrollo de la aplicación.
- Está compuesto por 13 fases.
- Describe las tareas y subtareas a realizar en cada fase pero, no ofrece modelos o técnicas de modelado.
- Incluye al usuario desde el principio del proceso de desarrollo.
- Si bien no define lineamientos de modelado, presenta la necesidad de definir modelos específicos para el usuario.
- Presenta la necesidad de elaborar la documentación específica a modelos y actividades desarrolladas en el ciclo de vida.
- Aunque la presentación de las fases es secuencial, HFPM permite la realización de iteraciones en el proceso de desarrollo.

La figura 4.3 muestra un diagrama de las fases que propone HFPM,

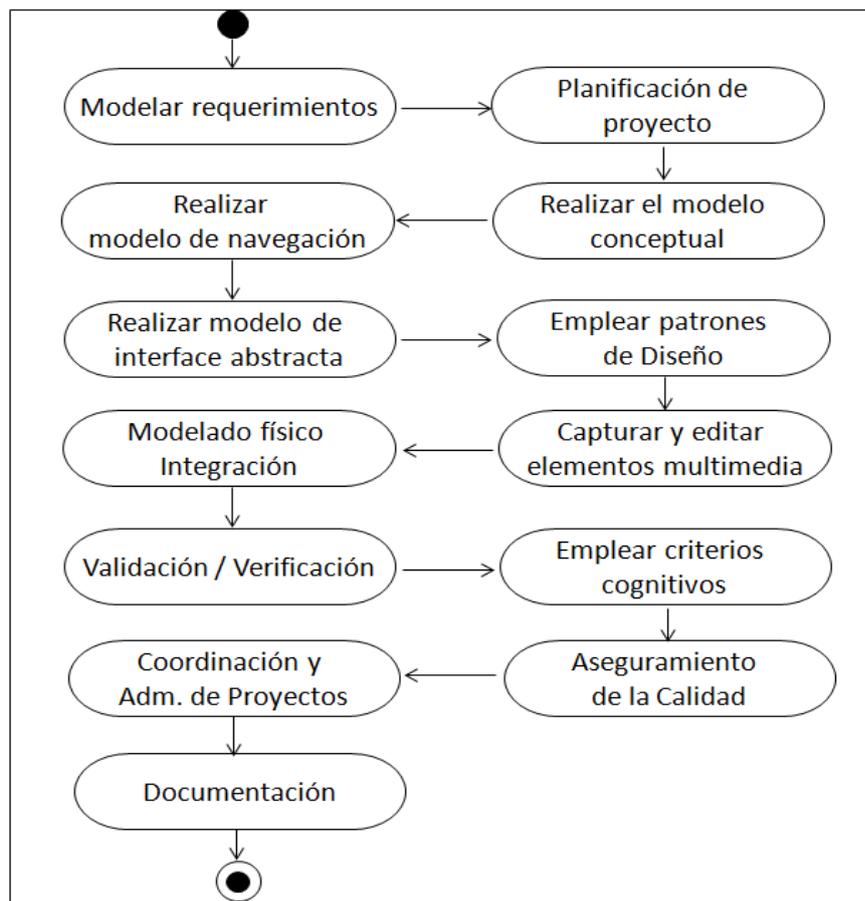


FIGURA 4.3: Diagrama de las fases de HFPM

A continuación se presentan y describen en forma general las actividades propuestas para cada una.

Modelar los requerimientos: En esta fase se proponen las siguientes tareas:

- Realizar encuestas iniciales.
- Describir los requerimientos funcionales mediante casos de usos.
- Modelar un glosario.
- Modelar la interface de usuario.
- Modelar los Requerimientos No Funcionales. En éstos incluyen la performance, la navegación, la reutilización, etc.

En principio no da ninguna norma a seguir para realizar estas tareas, dejando flexible las representaciones.

Planificación de proyecto: En esta fase se analizan y se especifican las actividades del plan de proyecto; no presenta ningún lineamiento o proceso a seguir para realizar el mismo.

Realizar el modelado conceptual: En esta fase se genera el modelo de clases que representa la aplicación. No se establecen los lineamientos a seguir para el realizar el modelo, se menciona que el objetivo de esta fase es similar a la fase de modelado conceptual de OOHDM (Object Oriented Hypermedia Design Model) en el cual, el modelo conceptual es representado como un modelo de clases para mostrar el aspecto estático del sistema. El enfoque OOHDM [11] no será descripto en este trabajo, dado que en el estudio realizado por Escalona y Koch no se presenta que OOHDM identifique los Requerimientos No Funcionales.

Las tareas a que propone HFPM en esta fase son:

- Analizar el dominio del problema.
- Realizar el modelo de dominio, mediante un modelo de clases.

Realizar el modelado de navegación: Esta fase propone realizar el modelo de navegación, el cual representará las posibilidades de navegación de la aplicación. Para realizar el modelo se menciona que los enfoques OOHDM, EORM (Enhanced Object Relationship Methodology) [12] y RMM (Relationship Management Methodology) [13] ofrecen procesos que pueden dar

soporte a las tareas de esta fase. Los procesos EORM [12] y RMM [13] no serán descriptos en este trabajo.

Las actividades de esta fase son:

- Analizar las necesidades de los usuarios.
- Identificar las clases de navegación.
- Especificar las clases de navegación y de esquema.
- Analizar las transformaciones de navegación.
- Especificar las transformaciones de navegación.

Realizar el modelo de interfaz abstracta: El objetivo de esta fase es realizar un diseño de la interface. Las tareas que se proponen son:

- Analizar las necesidades de interfaz de usuario.
- Analizar y especificar los objetos de interfaz, los eventos y las transformaciones mediante el diseño de prototipos.
- Realizar el modelo de interfaz

Emplear patrones de diseño: El objetivo de esta fase es emplear patrones de diseño a fin de enriquecer los modelos obtenidos en las fases anteriores. En esta fase se decide y diseña la arquitectura que tendrá la aplicación. Las tareas son las siguientes:

- Aplicar patrones de diseño para mejorar el modelo de interfaz de usuario.
- Emplear patrones de navegación y patrones de arquitectura
- Diseñar la arquitectura.

Capturar y editar elementos multimedia: En esta fase se plantean los múltiples medios con los que se va a trabajar la aplicación, tales como medios de almacenamiento que se usarán en la misma. No describe tareas.

Modelado físico / Integración: Esta fase si bien no especifica tareas, implica la creación de prototipos funcionales, prototipos evolutivos, bocetos o storyboarding e Integración de componentes.

Verificación y validación: En esta fase se va a analizar si el resultado es adecuado en base a los requerimientos detectados en la primera fase del proceso. No se establecen lineamientos a seguir dentro de la propuesta HFPM.

Emplear de criterios cognitivos: Esta fase propone emplear criterios de coherencia, cohesión y navegación. No especifica lineamientos para ello.

Aseguramiento de la calidad: Esta fase si bien HFPM no especifica qué entiende por calidad, ni como evaluar la calidad del producto propone estas actividades:

- Analizar y determinar las estrategias de calidad y mejora
- Especificar el plan de calidad

Coordinación y gestión de proyectos: En general, esta fase involucra las actividades de administrar y controlar los procesos, artefactos y Recursos. No se especifican pautas para ello.

Documentación: Esta fase involucra las tareas para generar la documentación de la aplicación. Las cuales implican:

- La documentación de los modelos generados en cada fase.
- La documentación del resultado de las pruebas y las valoraciones realizadas en
- las fases: Verificación/Validación y Aseguramiento de la calidad.

¿Qué tipos de Requerimientos No Funcionales identifica?

- Performance – (Relevamiento)
- Usabilidad – (Relevamiento - Diseño) si bien no es mencionado expresamente, se infieren cuando se plantea la actividad de Modelar la interfaz de usuario y considerar las necesidades del usuario para realizar el modelo de navegación.
- Reutilización – (Requerimientos)

Para realizar el modelo de navegación propone utilizar las bases de otros enfoques OOHDM, EORM (Enhanced Object Relationship Methodology) [12] y RMM (Relationship Management Methodology).

¿Qué técnicas de análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/ Captura de requerimientos: Encuesta,
- Especificación/ Definición de requerimientos: Utiliza, Glosario, Casos de uso de análisis, Interfaces sketches.
- Validación: Utiliza prototipos para la validación de requerimientos.

Requerimientos No Funcionales

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación: Utiliza Prototipos.

Administración de requerimientos: No menciona técnicas o herramientas.

4.2.3 UWE: UML-based Web Engineering

Propuesto por Koch [14], Koch y Wirsing [15] años 2000-2001, este enfoque metodológico está basado en el proceso de desarrollo unificado [17] y el lenguaje de modelado UML (Unified Modeling Language) [18] ambos, adaptados para aplicaciones Web.

Este proceso, además ha evolucionado hacia el desarrollo MDD (Model-driven software development) [16]¹, donde define los conceptos en base a un conjunto modelos.

¿Cuáles son las características relevantes?

- El proceso de UWE alcanza todo el ciclo de vida de desarrollo de aplicaciones web.
- Está compuesto por 3 fases o etapas.
- Se centra en el usuario en la etapa de la captura de requerimientos [20].
- Centra su atención en aplicaciones Web personalizadas o adaptativas.

¹ En [16] This article is an extended version of “Transformation Techniques in the Model-Driven Development Process of UWE” presented at the 2nd Workshop on Model-Driven Web Engineering (MDWE’06) at the 6th International Conference on Web Engineering (ICWE 2006).

- En la fase de captura de requerimientos, hace la separación entre actividades de elicitación, especificación y validación de requerimientos.
- Utiliza UML extendido denominado UML Profile, para generar los modelos propuestos en el proceso.
- El proceso está más enfocado a actividades de análisis y de diseño utiliza el paradigma de orientación a objetos.
- Los modelos son refinados en las iteraciones que plantea el proceso de desarrollo.
- Existe una herramienta Case, denominada ArgoUML [19] que soporta UWE.

La figura 4.4 muestra un diagrama de las etapas de UWE.

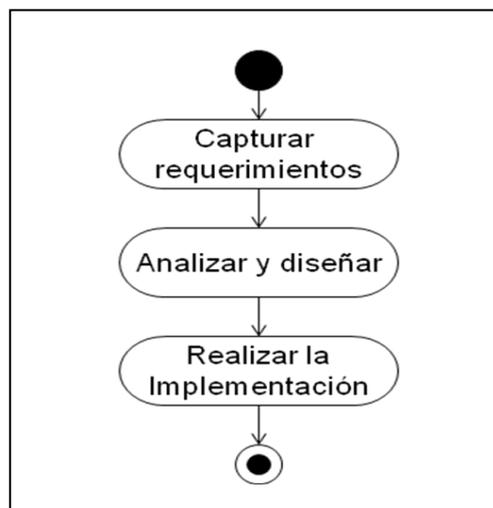


FIGURA 4.4: Diagrama de Proceso de UWE

A continuación se describen en forma general las fases/ etapas del proceso.

Capturar requerimientos: Esta etapa tiene por objetivo encontrar los requerimientos que la aplicación Web debe satisfacer y representarlos a través de un modelo de casos de uso, dado que es una técnica centrada en el usuario que permite definir quienes serán los *actores* que van a interactuar con la aplicación y representa la funcionalidad que la aplicación Web deberá cumplir para cada actor.

Como técnicas de captura, especificación y validación de requerimientos UWE propone [1]: para captura *entrevistas, cuestionarios y checklists*; para la especificación *glosarios, escenarios y casos de usos*, para validación *Walk-Through, auditorias y prototipos*.

UWE identifica requerimientos funcionales (contenido, presentación, navegación, adaptación) y no funcionales, si bien no establece lineamientos para representar los Requerimientos No Funcionales, los mismos serán considerados en las etapas siguientes.

Analizar y diseñar: Esta etapa toma como entrada el modelo de casos de uso a fin de generar: El modelo conceptual del dominio: Debe tratar de no tomar en cuenta los aspectos de navegación, presentación y adaptación. El resultado es un modelo de clases UML que contiene los atributos, operaciones de cada clase, además muestra gráficamente las relaciones de asociación, dependencia, jerarquía entre las clases del modelo; identifica interfaces y restricciones.

El modelo de navegación: Este se diseña a partir del modelo conceptual, las decisiones de diseño están basadas en modelo conceptual, modelo de casos y los requerimientos de navegación. El resultado es un modelo que muestra un diagrama de clases UML que representa las clases de navegación y las asociaciones entre estas.

El modelo de estructura de navegación: Describe como es la navegación a través del espacio de navegación. Dicho en otras palabras muestra como el usuario alcanzará las clases definidas en el modelo de navegación. Este modelo sirve de guía para construir el modelo de Presentación.

El modelo de presentación: Se enfoca en visualizar como es la organización estructural no en la apariencia física (formato, color etc.). Hay muchas posibilidades de construir un Modelo de presentación en base a Modelo de estructura de navegación dado. Los autores [20] proponen dividir el modelo en dos. Por un lado presentar el usuario el árbol de navegación y por otro el contenido. La profundidad del árbol de navegación en esta etapa de modelado debe ser limitada. El resultado es un Diagrama UML de clases construidas con estereotipos UML para elementos de presentación.

Realizar la implementación: Esta consiste en implementar los modelos validados e integrarlos. Se implementan desde los prototipos de interfaz de usuario hasta el código.

¿Qué tipos de Requerimientos No Funcionales identifica?

- Seguridad - (Relevamiento).
- Usabilidad – (Relevamiento - Diseño).

UWE identifica requerimientos de presentación, navegación, como requerimientos funcionales.

¿Qué técnicas de análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/ Captura de requerimientos: entrevistas, cuestionarios y checklists;
- Especificación/ Definición: Glosarios, escenarios y casos de usos,
- Validación: Walk-Through, auditorias y prototipos.

Requerimientos No Funcionales

- Elicitación/ Captura de requerimientos: No especifica.
- Especificación/ Definición: No especifica.
- Validación: Utiliza Prototipos.

Administración de requerimientos: UWE utiliza ArgoUML [19] herramienta case (No comercial) la cual:

- Permite la administración de los requerimientos y modelos generados a partir de los mismos.
- Soporta la notación de UWE para diseñar aplicaciones Web.
- Implementa el proceso de desarrollo de UWE
- Permite modelar los conceptos de (Contexto, navegación y presentación)
- La desventaja es que esta herramienta no ha evolucionado a UML 2.0

4.2.4 WebML: Web Modeling Language

Propuesto en el año 2000, por Ceri, Fraternali y Bongio [21], quienes lo presentan como una notación para especificar sitios web complejos a un nivel conceptual.

Los elementos de WebML están asociados a una notación gráfica y utiliza sintáxis de XML (eXtensible Markup Language).

WebML presenta un proceso de desarrollo de aplicaciones Web dirigido por modelos.

¿Cuáles son las características relevantes?

- El proceso de WebML alcanza todo el ciclo de vida de desarrollo de aplicaciones Web.
- Permite la descripción en alto nivel de una aplicación web, bajo perspectivas ortogonales, tales como: el contexto de datos, las páginas que la componen, los link o enlaces entre las páginas, la presentación gráfica por página y la personalización por grupo de usuarios.
- Cubre aspectos avanzados del modelado de aplicaciones Web incluyendo presentación, modelado de usuarios y personalización.
- Es compatible con la notación clásica del modelo de E/R y UML [18].
- El proceso de diseño consta de 4 fases.
- Considera por un lado la captura de requerimientos y por otro la especificación a la cual denomina *Análisis de requerimientos*, donde utiliza casos de usos y diagrama de actividades para expresar el workflow de casos de uso complejos.
- Propone test de aceptación para validar los requerimientos funcionales y no funcionales.
- Esta soportado por dos herramientas CASE TotiiSoft [23] y WebRatio [24].

La figura 4.5 muestra un diagrama de las fases de WebML.

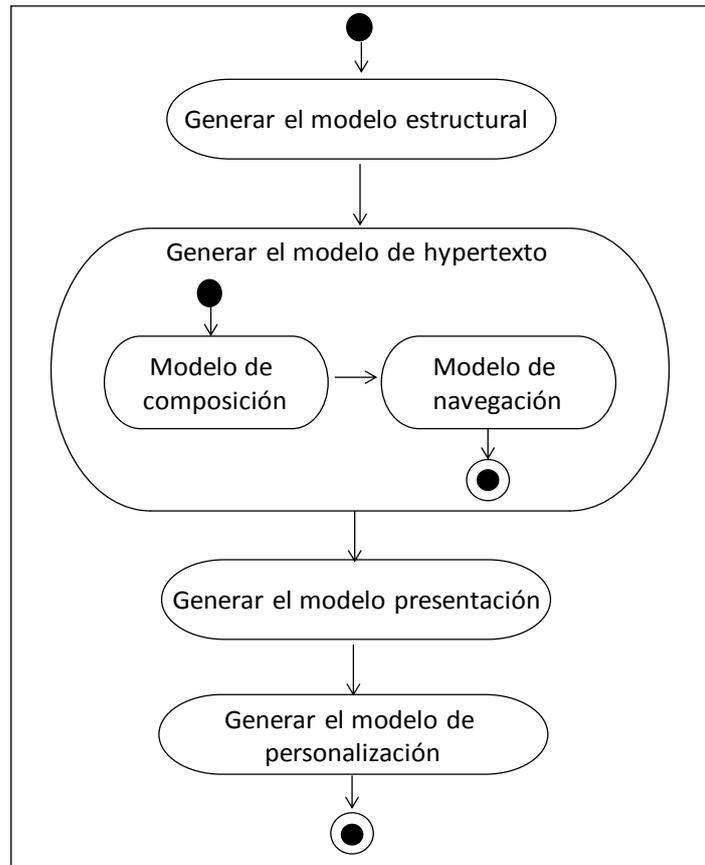


FIGURA 4.5: Diagrama de Proceso de WebML

A continuación se describen las cuatro fases o perspectivas de diseño de que propone WebML, las mismas se complementan con las actividades de captura de Requerimientos y validación propuestas en [22].

Captura y Análisis de requerimientos: Comienza con la identificación de usuarios y grupos y personalización de sus necesidades, para realizar la recolección de requerimientos propone técnicas como entrevistas, análisis de documentación y checklists de captura de requerimientos. Identifica requerimientos funcionales, de estructura o navegación y Requerimientos No Funcionales tales como: Performance, Disponibilidad, Escalabilidad, Mantenibilidad y Seguridad.

La especificación de los requerimientos (es denominada análisis) en esta etapa se obtiene un conjunto de especificaciones semi-formales, que incluyen, la lista de grupos de usuarios con sus derechos de acceso, un diccionario de datos de los objetos más relevantes del dominio de

la aplicación, una especificación informal de las vistas del sitio, una lista de Requerimientos No Funcionales, un conjunto de guías de presentación que dan indicaciones para el look and feel de la interface a ser desarrollada. Utiliza casos de usos para expresar las interacciones entre los grupos de usuarios y la aplicación, dependiendo de la complejidad del caso de uso, se modelan diagramas de actividad del workflow de proceso. Se utiliza notación UML.

Validación de requerimientos: propone elaborar test de aceptación para validar los Requerimientos No Funcionales.

Generar el modelo estructural: Pone foco en las entidades relevantes del contexto de datos de la aplicación Web. Representa mediante notación del modelo de E/R y clases de UML las entidades y relaciones entre estas.

Generar el modelo de hipertexto: Describe los hipertextos que pueden ser publicados en la aplicación. Está compuesto de dos submodelos.

Modelo de composición: Muestra cuales son las páginas que pueden ser accedidas y las unidades de contenido que componen cada página.

Modelo de Navegación: Muestra como las páginas y las unidades de contenido de cada páginas son accedidas o están linkeadas.

Generar el modelo de presentación: Expresa mediante sintaxis XML, la apariencia de las páginas y los objetos incluidos en cada página, independientemente del dispositivo de salida. Se pueden generar páginas específicas o genéricas.

Generar el modelo de personalización: En el modelo de estructura los usuarios y grupos han sido modelados como entidades llamadas Usuarios y Grupos, las cuales tiene características almacenadas que pueden ser usadas para guardar grupos específicos o contenidos individuales. Mediante sentencias declarativas con OQL se pueden agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.

Que tipos de Requerimientos No Funcionales identifica?

- Usabilidad - Relevamiento - Diseño
- Seguridad - Relevamiento - Diseño
- Performance - Relevamiento
- Disponibilidad - Relevamiento
- Escalabilidad - Relevamiento
- Mantenibilidad - Relevamiento

WebML propone utilizar sentencias declarativas con OQL para agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.

¿Qué técnicas de análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/captura de requerimientos: Entrevistas, análisis de documentación y checklists de captura de requerimientos.
- Especificación/ Definición de requerimientos: Utiliza Casos de uso de análisis, Diagramas de Actividad para expresar workflow de casos de uso complejos. Diccionario de datos.

Además menciona:

- Lista de grupos de usuarios con sus derechos de acceso.
- Diccionario de datos de los objetos más relevantes del dominio de la aplicación.
- Una especificación informal de las vistas del sitio.
- Un conjunto de guías de presentación que dan indicaciones para el look and feel de la interface a ser desarrollada.

Requerimientos No Funcionales

- Elicitación/captura de requerimientos: No especifica
- Especificación/ Definición de requerimientos: Lista de Requerimientos No Funcionales.

- Validación: Test de aceptación para validar Requerimientos No Funcionales tales como Performance, Disponibilidad, Escalabilidad, Mantenibilidad y Seguridad.

Administración de requerimientos: WebML esta soportado por dos herramientas CASE TotiiSoft [23] y WebRatio [24].

TotiiSoft

- Cubre las actividades del ciclo de vida de desarrollo de aplicaciones web.
- Esta centrado en el uso de WebML.
- Tiene un módulo de diseño para edición de especificaciones de estructura, hipertextos y de modelos de personalización.
- Módulo de Manager para la administración y evolución de la aplicación en el tiempo.

WebRatio

- Entorno de desarrollo dirigido por modelos
- Para el desarrollo de aplicaciones complejas Web Java basadas en lenguajes BPMN y WebML.
- Desarrollo DDM de aplicaciones personalizadas BPM/BAM basado en BPMN y WebML
- Desarrollo DDM de aplicaciones personalizadas Web/SOA basado en WebML

4.2.5 NTD: Navigational Development Techniques

Propuesto en el año 2004, por Escalona [27] y Escalona, Torres, Mejias, [25][26], como sus autores lo indican, es una metodología centrada en la primera fase del ciclo de vida, su proceso de desarrollo se enfoca en la ingeniería de requerimientos, para lo cual presenta las actividades y técnicas para elicitar, analizar, especificar y validar los requerimientos de aplicaciones Web.

¿Cuáles son las características relevantes?

- El proceso no contempla las fases de implementación o mantenimiento del ciclo de vida de desarrollo de aplicaciones Web, está centrado en la fase de definición de requerimientos.
- El proceso de definición de requerimientos se basa los diferentes roles de usuarios que pueden aparecen en el sistema.
- Es un enfoque orientado al *proceso* debido a que para cada actividad describe concretamente: los pasos a seguir para obtener los requerimientos y en base a ellos generar los modelos.
- Es un enfoque orientado a la *técnica*, debido a que describe las técnicas que se deben aplicar, la forma de aplicarlas y los resultados que debe ser obtenidos.
- Es un enfoque orientado al *resultado*, puesto que NDT describe los modelos de estructura y nomenclatura para los resultados obtenidos tras la aplicación de cada técnica y los resultados finales del proceso que son: la especificación de requerimientos, la especificación de modelos de análisis y los prototipos de navegación.
- Propone la validación de las especificaciones y modelos conceptuales obtenidos.
- Ofrece una guía sistemática para el tratamiento de la navegación.
- NDT no propone lenguajes de modelado, sino que utiliza lenguajes de modelado estándares como UML y UWE definido en otro enfoque metodológico.
- Define un proceso iterativo para generar la especificación de requerimientos y especificación de modelos de análisis, dado que estos modelos son estándares, pueden ser tomados por otra metodología basada en esos estándares y continuar el ciclo de vida de desarrollo.
- Propone 2 fases: Captura de Requerimientos la cual se divide en 7 subfases y por otro lado la fase de Análisis que comprende 3 subfases.
- Es soportado por una herramienta CASE NT-Tool [25][27].

La figura 4.6 muestra un diagrama de las fases y subfases de NDT.

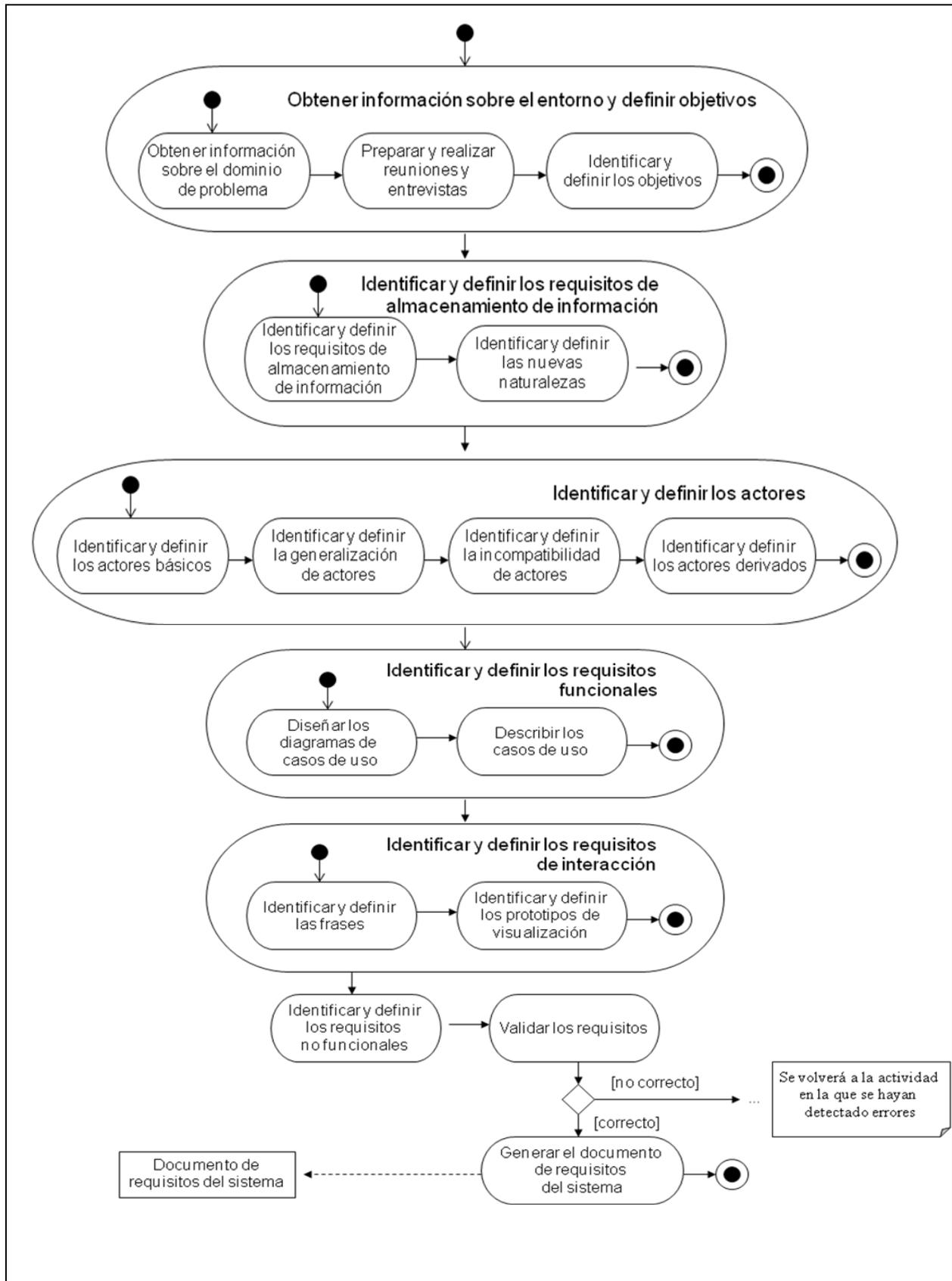


FIGURA 4.6: Diagrama de Proceso de NDT

A continuación se describen brevemente las fases y subfases que propone NDT.

Fase de Captura de requerimientos: El objetivo de esta fase es obtener como resultado el *Documento de especificación de requerimientos*. NDT clasifica a los requerimientos en: de Almacenamiento de información, de actores, de interacción, funcionales y no funcionales.

El proceso comienza definiendo los objetivos del sistema para lo cual plantea las siguientes actividades:

- Obtener información sobre el entorno y definir objetivos: Las actividades principales consisten en:
 - Obtener información sobre el dominio de problema.
 - Preparar y realizar reuniones y entrevistas.
 - Identificar y definir los objetivos.

Las técnicas propuestas son entrevistas, brainstorming, cuestionarios, concept mapping, patrón para la definición de objetivos.

- Identificar y definir los requerimientos de almacenamiento de información: se propone realizar las actividades de
 - Identificar y definir los requerimientos de almacenamiento de información
 - Identificar y definir las nuevas naturalezas

Las técnicas propuestas son técnicas de estudios de sistemas anteriores, patrón para la definición de requerimientos de información, patrón para la definición de las nuevas naturalezas.

- Identificar y definir los actores: Consiste en las siguientes actividades
 - Identificar y definir los actores básicos
 - Identificar y definir la generalización de actores
 - Identificar y definir la incompatibilidad de actores
 - Identificar y definir los actores derivados

Las técnicas propuestas para cada actividad son: Patrón para la definición de actores básicos, diagrama de representación de actores generalizados, matriz

para la definición de incompatibilidad de actores, matriz para la definición de actores derivados.

- Identificar y definir los requerimientos funcionales: en base a las siguientes actividades,
 - Diseñar los diagramas de casos de uso
 - Describir los casos de uso

Las técnicas propuestas son: Diagramas de casos de uso y un patrón para la definición de requerimientos funcionales.

- Identificar y definir los requerimientos de interacción: las actividades que se realizan son
 - Identificar y definir las frases
 - Identificar y definir los prototipos de visualización

Las técnicas propuestas son: BNL (Bounded natural language), un patrón para definición de frases y un patrón para la definición de prototipos de visualización.

- Identificar y definir los Requerimientos No Funcionales: Se enumeran todas las necesidades que no están involucradas en ninguna de las categorías anteriores. NDT menciona que no es posible detectar todos los Requerimientos No Funcionales del sistema, debido a la diversidad que existe, solamente propone un patrón para describirlos. Dentro de los Requerimientos No Funcionales NDT identifica a los requerimientos de comunicación, de fiabilidad, de entorno de desarrollo, de portabilidad.
Como técnica propone un patrón para describir Requerimientos No Funcionales.

- Validar los requerimientos: una vez que se identificado y definido los requerimientos, se hace necesaria su validación.
Las técnicas propuestas para esta fase son: revisiones, auditorias, glosarios, tesauros y ontologías y matriz de trazabilidad.

Si en la validación se detectaron errores, se volverá a la actividad en la que se hayan detectado los mismos para corregirlos.

- Generar el documento de requerimientos del sistema: NDT propone la estructura que debe tener este documento; el cual es la base para la fase de análisis.

Fase de Captura de Análisis: El objetivo de esta fase es obtener como resultado el documento de Análisis del Sistema y los Prototipos evaluables, para ello plantea las siguientes actividades:

- Realizar el Modelo Conceptual: la cual consiste en
 - Realizar el Modelo Conceptual Básico
 - Realizar el Modelo Conceptual Final

Las técnicas propuestas son diagrama de clases, patrón para la definición de clases, patrón para la definición de asociaciones, patrón para la definición de paquetes, proceso de generación de modelo básico, proceso de revisión de modelo básico.

- Realizar el Modelo de Navegación: esta actividad tiene dos tareas
 - Realizar el Modelo de Navegación Básico
 - Realizar el Modelo de Navegación Final

Las técnicas propuestas son diagrama de navegación de UWE, patrón para la definición de nodos, enlaces, menús, índices, queries, proceso de generación de modelo básico, proceso de revisión de modelo básico algoritmos de Kruskal, y algoritmo de Warshall.

- Realizar y validar el conjunto de prototipos: que consiste en
 - Realizar el Modelo de Prototipos básicos:
 - Realizar el Modelo de Prototipos Final

Las técnicas son procesos de generación de prototipos, revisión y auditorías.

- Generar el documento de análisis del sistema: NDT propone la estructura que debe tener este documento; la cual es similar a la del documento de requerimientos. En este documento se incluyen y describen los modelos generados durante la fase de análisis mencionados anteriormente.

En caso de detectarse errores en los modelos, se volverá a la actividad en la que se hayan detectado los mismos para corregirlos.

Es importante destacar que los documentos de especificación de requerimientos, de modelo de análisis y los prototipos obtenidos por NDT, sirven de base para el diseño del sistema.

¿Qué tipos de Requerimientos No Funcionales identifica?

- Usabilidad - Relevamiento
- Fiabilidad - Relevamiento
- Portabilidad - Relevamiento
- Entorno de desarrollo - Relevamiento
- Comunicación - Relevamiento

NDT menciona que no es posible detectar todos los Requerimientos No Funcionales del sistema, debido a la diversidad que existe.

¿Qué técnicas de análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/Captura: Las técnicas propuestas son entrevistas, brainstorming, cuestionarios, concept mapping, patrón para la definición de objetivos.
- Especificación/Definición: utiliza técnicas estándares conocidas para especificar los diferentes tipos de requerimientos tales como:

- Estudios de sistemas anteriores, patrón para la definición de requerimientos de información,
 - Patrón para la definición de actores básicos, Diagramas de casos de uso y un patrón para la definición de requerimientos funcionales, BNL (Bounded natural language), diagrama de clases, diagrama de navegación de UWE, algoritmos de Kruskal y algoritmo de Warshall.
 - Y técnicas propuestas por NDT tales como: Patrón para la definición de las nuevas naturalezas, diagrama de representación de actores generalizados, matriz para la definición de incompatibilidad de actores, matriz para la definición de actores derivados, patrón para definición de frases, patrón para la definición de prototipos de visualización, patrón para la definición de clases, asociaciones y paquetes, patrón para la definición de nodos, enlaces, menús, índices y queries, proceso de generación de modelo básico conceptual, proceso de revisión de modelo básico conceptual, proceso de revisión de modelo básico navegacional, proceso de generación de modelo básico navegacional, procesos de generación de prototipos.
- Validación: Las técnicas propuestas para esta fase son: revisiones, auditorias, glosarios, tesauros y ontologías y matriz de trazabilidad.

Requerimientos No Funcionales

- Elicitación/captura de requerimientos: No especifica.
- Especificación/ Definición de requerimientos: Patrón para describir los Requerimientos No Funcionales.
- Validación: Revisiones, Auditorias, Matriz de trazabilidad, Prototipos.

Administración de requerimientos: NDT está soportado por la herramienta NDT-Tool [25] [27] la cual:

- Sirve de soporte en la aplicación de las técnicas de captura o elicitación y especificación de requerimientos propuestas por NDT.

- Automatiza los procedimientos para generar modelos básicos conceptual y de navegación propuestos por NDT.
- Permite generar automáticamente el documento de especificación de requerimientos, de análisis y los prototipos.
- Controla las inconsistencias y errores producidos en la implementación de los patrones.
- Utiliza rational rose para visualizar los modelos (conceptual y de navegación) generados.
- Permite administrar varios proyectos en forma paralela.
- Permite a varios desarrolladores trabajar en forma concurrente.

A diferencia de otros enfoques metodológicos como UWE y WebML que tienen herramientas que soportan el proceso. NDT tool es una herramienta que soporta la primera fase del ciclo de vida y ayuda a elaborar los modelos y resultados de la aplicación de ingeniería de requerimientos propuestos por NDT.

4.2.6 DDDP: Design-Driven Requirements Elicitation

Propuesto por Lowe [28], Lowe, D., Eklund, J. [29], la denominación DDDP fue realizada por Escalona en [1].

Este proceso, propone un modelo iterativo para el desarrollo de aplicaciones Web que incorpora el uso de prototipos de diseño parcial como un elemento crucial para identificar, entender las necesidades del cliente y conducir a la formulación de los requerimientos.

¿Cuáles son las características relevantes?

- El proceso de DDDP es derivado de un resultado de un análisis de best practices en el desarrollo de aplicaciones comerciales para la web.

- La propuesta para el tratamiento de requerimientos es parte del proceso design-Driven que consiste en realizar la captura, la definición y la validación de requerimientos durante el proceso de diseño.
- Se enfoca en establecer el cómo y cuando los requerimientos son identificados.
- El proceso iterativo incorpora al cliente-desarrollador explorando en forma conjunta los diseños parciales, como parte del desarrollo de la especificación del sistema.
- Está compuesto por 2 fases o ciclos.

La figura 4.7 muestra un diagrama de las fases de DDDP.

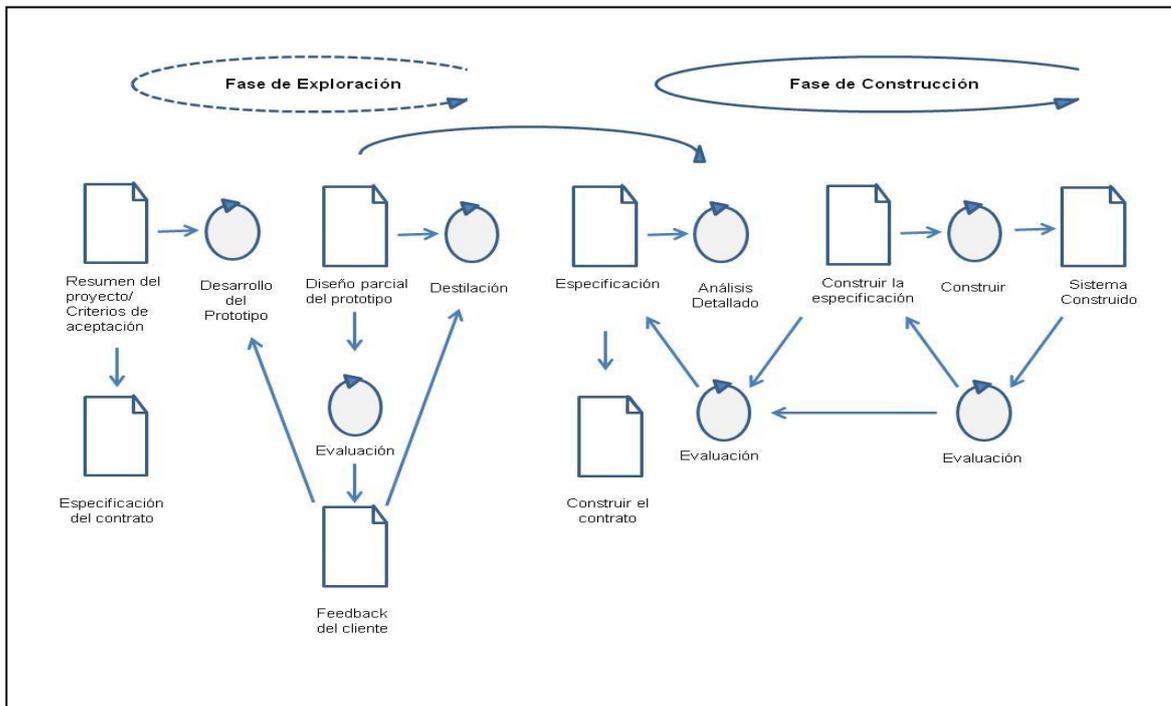


FIGURA 4.7: Diagrama de Proceso de DDDP

A continuación se describen brevemente las fases que propone DDRE o DDDP.

Fase de Exploración: Esta fase parte de un resumen de las necesidades del proyecto y una especificación de contrato, en base a los cuales, los desarrolladores diseñan prototipos

parciales, luego exploran y evalúan los mismos con el cliente a fin de obtener feedback e ir generando la especificación de requerimientos en forma iterativa.

Este proceso identifica requerimientos de contenido, protocolos de interface, estructura de navegación, look and feel, representación interna de datos, versiones, control de cambio, seguridad, eficiencia, adaptación del sistema, performance.

Este enfoque no define el template para los artefactos resultantes de cada etapa. Una vez que se obtuvo la especificación de requerimientos se realiza un contrato.

Fase de Construcción: Esta fase comienza después que se obtuvo el contrato. Toma como entrada, el/ los diseños de prototipos parciales y la especificación de requerimientos generados en la fase anterior y plantea actividades para construir la aplicación en forma iterativa, refinando los diseños de prototipos, codificando, evaluando los resultados con el cliente a fin de actualizar la especificación de requerimientos y construir la aplicación en forma incremental.

Este enfoque plantea que muchos requerimientos son expresados como parte del diseño. Lo cual permite que ciertas características o necesidades del cliente que surgen al inicio del proceso, tales como, look and feel, estructura de contenido sean realizadas y exploradas a nivel de arquitectura, antes de que las funcionalidades del sistema sean finalizadas.

Las restricciones de performance, protocolos de interface, decisiones sobre la plataforma técnica, que con frecuencia aparecen al momento de construir la aplicación, pueden ser analizadas utilizando diseños de prototipos que ayudan al cliente a explorar posibles soluciones y de esa forma, formular los requerimientos.

¿Qué tipos de Requerimientos No Funcionales identifica?

- Estructura de navegación – Relevamiento.
- look and feel - Relevamiento.
- Representación interna de datos: Relevamiento.
- Seguridad - Relevamiento.
- Versiones, control de cambio: Implementación

- Eficiencia: Implementación
- Adaptación del sistema: Implementación
- Protocolos de interface: Implementación
- Performance: Implementación

¿Qué técnicas de análisis y administración de requerimientos emplea?

Requerimientos Funcionales

- Elicitación/Captura: Entrevistas.
- Especificación/ Definición: No especifica.
- Validación: Prototipos.

Requerimientos No Funcionales

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación, propone prototipos para validar restricciones durante la etapa de diseño.

Administración de requerimientos: No menciona técnicas o herramientas.

5. Estudio Comparativo tratamiento de Requerimientos No Funcionales en las Metodologías de desarrollo de aplicaciones web.

En este capítulo se presentan los resultados del estudio comparativo de las metodologías de desarrollo de aplicaciones web con tres puntos de vista:

- Qué tipos de Requerimientos No Funcionales identifica cada enfoque.
- Qué técnicas para tratamiento de los Requerimientos No Funcionales utiliza cada enfoque.
- En qué fase del ciclo de vida se identifican los Requerimientos No Funcionales en cada enfoque.

En las secciones, 5.1, 5.2 y 5.3 se desarrollan cada uno de estos puntos.

5.1 Tipos de Requerimientos No Funcionales en cada enfoque

En el capítulo 2 se presentaron las bases conceptuales a ser aplicadas en este trabajo. A partir de lo expuesto en Cysneiros [7] se considera que los Requerimientos No Funcionales son *requerimientos de calidad* y son *restricciones*. Esta sección presenta los resultados de la aplicación de estos conceptos en la revisión y análisis de los enfoques metodológicos de desarrollo de aplicaciones web descritos en el capítulo 4, a fin de determinar los tipos de Requerimientos No Funcionales que son identificados y tratados en cada enfoque. En las tablas resultantes los enfoques metodológicos siguen un orden cronológico, que permite ver la evolución e importancia que fueron adquiriendo los diferentes tipos de Requerimientos No Funcionales en el entorno de la Web.

Finalmente se presenta la evolución de la tabla 4.1 presentada por Escalona y Koch en [1], citada en el capítulo 4, a fin de realizar una extensión de la columna “Req. No funcionales” con los conceptos expuestos en este trabajo.

Los Requerimientos No Funcionales como requerimientos de calidad:

En el análisis de las metodologías surgieron atributos de calidad adicionales a las características del estándar ISO/IEC 9126 [34], se decidió incluirlos a efectos de establecer todos los atributos que son considerados por cada enfoque. La tabla 5.1 ilustra el resultado de este análisis, presenta las seis características de calidad del estándar ISO/IEC 9126 y otras características, mostrando cuales de ellas son consideradas por cada proceso metodológico.

	Atributos de calidad ISO/IEC 9126-1						Otros				
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilizacion	Fiabilidad	Disponibilidad	Escalabilidad
WSDM	✓		✓		✓						
HFBM	✓	✓						✓			
UWE	✓		✓								
WebML	✓	✓	✓		✓					✓	✓
NDT	✓					✓			✓		
DDDP	✓	✓	✓				✓				

(1) Subcaracterística de la Eficiencia

(2) Subcaracterística de la Funcionalidad

(3) Subcaracterística de la Portabilidad

TABLA 5.1: Requerimientos No Funcionales de calidad y las metodologías Web.

A partir de la tabla 5.1 se pueden extraer las siguientes conclusiones:

- WebML. Es en cierta forma el enfoque más preocupado por *requerimientos de calidad*, considera cuatro características del estándar ISO/IEC 9126 y dos características adicionales. En contraposición UWE es el que menos atributos de calidad considera en su proceso.

Después de WebML, es DDDP el segundo interesado en los requerimientos de calidad, pero solo se limita a las características de modelo ISO/IEX 9126.

- La *usabilidad* es la característica del modelo ISO/IEX 9126 considerada por todos procesos.
- La *confiabilidad* no es tratada por ninguna de las metodologías estudiadas.
- La *usabilidad*, *performance* y *seguridad*, son las características más relevantes.
- La *mantenibilidad* y la *portabilidad* son consideradas por dos enfoques cada una.
- La *portabilidad* es relevante para los enfoques más recientes, de acuerdo al orden cronológico de presentación.
- La *reutilización*, la *fiabilidad*, la *disponibilidad*, la *escalabilidad* son las características no mencionadas explícitamente en el estándar ISO/IEX 9126; es importante destacarlas para su futuro análisis.

Los Requerimientos No Funcionales como restricciones:

Esta sección está enfocada principalmente en las *restricciones del proceso de desarrollo* que fueron propuestas y definidas en el capítulo 2 tales como: *Alcance del proceso* (Parcial, Completo) y *Nivel de detalle del proceso* (Parcial o completo). La importancia de considerarlas radica en que un proceso de desarrollo incompleto, tendrá impacto directo en el tratamiento de los Requerimientos No Funcionales, debido a que carece de actividades, técnicas o productos para capturarlos, documentarlos y validarlos. Por consiguiente impactará en la calidad de las aplicaciones web que se desarrollen usando alguno de ellos.

También se identificaron y se agruparon otras restricciones a fin de obtener la visión global de las todas restricciones tratadas por cada enfoque. La tabla 5.2, presenta el resumen de este análisis.

	Restricciones										
	Restricciones de proceso de desarrollo				Otras						
	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas				
	Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo
WSDM		✓	✓			✓					
HFPM		✓	✓		✓						
UWE		✓	✓		✓						
WebML		✓		✓	✓	✓					
NDT	✓			✓	✓					✓	✓
DDDP		✓	✓		✓	✓	✓	✓	✓		

TABLA 5.2: Requerimientos No Funcionales – Restricciones y las metodologías Web.

Del análisis de la tabla 5.2 se pueden extraer las siguientes conclusiones:

- Las *restricciones de proceso* aplicadas en las metodologías estudiadas permiten observar que:
 - WebML ofrece un enfoque que cubre todas las fases del ciclo de vida de desarrollo y además el brinda un nivel de detalle completo para cada fase, propone las actividades, técnicas y ofrece detalles de los productos a producir en cada fase.
 - WSDM, HFPM, UWE, DDDP si bien presentan un proceso de desarrollo completo, el nivel de definición de cada proceso es parcial, esto implica que carecen de lineamientos específicos de las actividades o la aplicación de técnicas, o el detalle de los productos esperados como resultado de cada fase. A primera vista estos enfoques demandan la formación de un equipo de trabajo con mucha experiencia en el desarrollo de aplicaciones web para su aplicación.
 - NDT tiene un proceso de desarrollo parcial. Aunque el nivel de detalle de la definición del proceso es completa porque detalla las actividades, las técnicas y ofrece detalles de los productos a producir en las fases que propone. El proceso

está acotado solamente a las fases de relevamiento y análisis de requerimientos.

- DDDP el proceso plantea la captura, definición y validación de requerimientos durante el proceso de diseño. Propone analizar las restricciones mediante prototipos; el nivel de detalle del proceso es parcial, menciona las actividades para cada fase, no especifica las técnicas, ni ofrece detalles de los productos a generar en cada fase.

Las otras restricciones encontradas en el análisis de las metodologías fueron agrupadas como: restricciones de *lenguaje* (son las *restricciones de lenguaje de Modelado* y las *restricciones de lenguajes de implementación*), y restricciones *técnicas* (estas comprenden a *Restricciones de representación interna de datos, control de versiones, protocolos de interface, comunicación y las restricciones de ambiente de desarrollo*).

- *Restricciones de Lenguaje de modelado:*

- WSDM, es el único enfoque que no menciona restricciones de lenguaje de modelado.
- HFPM, demanda el conocimiento adicional de otros enfoques OOHDM [11], EORM [12] y RMM [13] para generar modelos de navegación.
- UWE, requiere de conocimiento de enfoques UML profile y paradigma de orientación a objetos.
- WebML, demanda conocimiento de lenguajes de modelado UML para generar el modelo estructural y XML para mostrar la apariencia de las páginas y los objetos incluidos en cada página.
- NDT, requiere conocimiento de lenguajes de modelado UML y los diagramas de navegación de UWE.

- *Restricciones de Lenguaje de implementación:* WSDM, WebML y DDDP mencionan restricciones de este tipo:

- En WSDM el modelo de implementación, puede depender de las limitaciones del lenguaje de programación.

- WebML propone utilizar sentencias declarativas con OQL para agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.
- DDDP, demanda conocimiento de lenguajes de programación que permitan la generación de prototipos desde la fase de exploración.
- *Las restricciones de representación interna de datos, control de versiones, protocolos de interface, son detectadas solo por DDDP.*
- *Comunicación y las restricciones de ambiente de desarrollo son detectadas únicamente por NDT.*

Hay dos puntos de vistas significativos que se desprenden de las restricciones de proceso de desarrollo analizadas en la tabla 5.2:

- El primero está relacionado al nivel de detalle de especificación parcial, de un proceso de desarrollo completo: *Un proceso de desarrollo que contempla todas las fases del ciclo de vida, pero no brinda un nivel de detalle completo en sus fases, dará lugar a una forma de trabajo no sistemática. La calidad de los productos intermedios y la aplicación web resultante estarán ligados a la idoneidad y experiencia del ingeniero de software.*
- El segundo, está vinculado al nivel de detalle de especificación completa de un proceso de desarrollo de alcance parcial: *Un proceso de desarrollo cuyo nivel de especificación de proceso es completa solo para las primeras fases del ciclo de vida, permitirá, en cierta forma, dar garantía de la calidad de los productos intermedios, pero no dará visibilidad de la calidad la aplicación web resultante, ya que las fases del ciclo de desarrollo ausentes estarán ligadas a la experiencia del equipo de trabajo.*

En ambos casos, se puede ver que contar con un proceso de desarrollo incompleto o con un nivel de detalle de especificación parcial, impactará directamente en la calidad

de los productos intermedios y en la aplicación web resultante. Por consiguiente un proceso que no contempla la identificación de Requerimientos No Funcionales es incompleto y un proceso que si identifica Requerimientos No Funcionales a lo largo de su ciclo de vida, pero no ofrece actividades, técnicas, herramientas para su tratamiento, también lo es.

Extensión de la tabla de Escalona y Koch con los diferentes tipos de Requerimientos No Funcionales

La tabla 5.3 “Requerimientos identificados por cada metodología”, extraída del estudio realizado por Escalona y Koch, resume los tipos de requerimientos tratados por cada enfoque metodológico estudiado por los autores, en esta tabla la columna “Req. No Funcionales”, señala simplemente cual de los enfoques los identifica, sin discriminarlos.

	Req. Datos	Req. Interfaz usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFBM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
WebML	✓	✓		✓		✓
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

TABLA 5.3: Requerimientos identificados por cada metodología

A continuación, partiendo del análisis realizado en este trabajo, se presentan tablas y gráficos que evolucionan la tabla 5.3 de Escalona y Koch [1], agregando los tipos de Requerimientos No Funcionales identificados y tratados en cada enfoque.

La apertura de la columna “Req. No Funcionales” en las tablas 5.4 y 5.5 se realizó considerando a los enfoques WSDM, HFPM, UWE, WebML, NDT, DDDP analizados en el capítulo 4.

En primer lugar se realizó la tabla 5.4, en la cual se agregaron los requerimientos de calidad y las restricciones de proceso, dejando a un lado requerimientos calidad y restricciones adicionales a los conceptos introducidos en el capítulo 2. El gráfico 5.1 surgió a partir de esta tabla.

La tabla 5.5 suma a la tabla anterior los *requerimientos de calidad adicionales* y *otras restricciones* detectadas a partir del análisis de los enfoques, a fin de tener una visión global de todos los requerimientos identificados y tratados por cada metodología de desarrollo estudiada. La visión gráfica de estos conceptos se muestra en el gráfico 5.2.

	Atributos de calidad							Restricciones			
	Atributos de calidad ISO/IEC 9126-1							Restricciones de proceso de desarrollo			
								Alcance del proceso		Detalle de proceso	
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Parcial	Completo	Parcial	Completo
WSDM	✓		✓		✓				✓	✓	
HFPM	✓	✓							✓	✓	
UWE	✓		✓						✓	✓	
WebML	✓	✓	✓		✓				✓		✓
NDT	✓					✓		✓			✓
DDDP	✓	✓	✓				✓		✓	✓	

(1) Subcaracterística de la Eficiencia
 (2) Subcaracterística de la Funcionalidad
 (3) Subcaracterística de la Portabilidad

TABLA 5.4: Requerimientos No Funcionales identificados y tratados por cada metodología de desarrollo Web - Parcial

La tabla 5.4 muestra que:

- WebML y DDDP son los enfoques que más Requerimientos No Funcionales de calidad identifican.
- WebML a diferencia de DDDP ofrece un proceso de desarrollo completo en cuanto a alcance y el nivel de especificación del proceso.

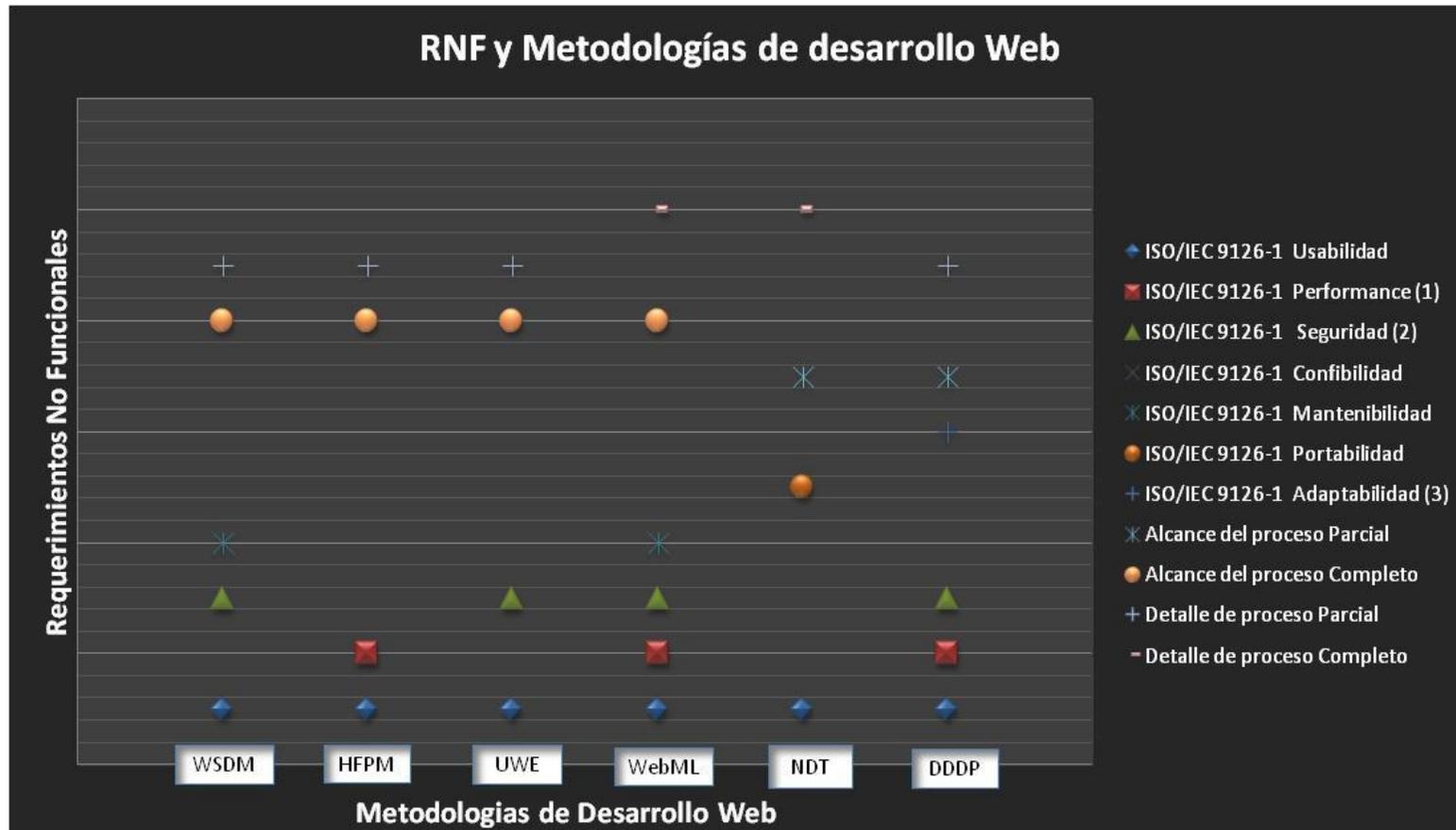


FIGURA 5.1: Requerimientos No Funcionales y Metodologías de desarrollo Web - Parcial.

	Atributos de calidad											Restricciones										
	Atributos de calidad ISO/IEC 9126-1							Otros				Restricciones de proceso de desarrollo				Otras						
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilizacion	Fiabilidad	Disponibilidad	Escalabilidad	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas				
												Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo
WSDM	✓		✓		✓								✓			✓						
HFPM	✓	✓					✓						✓		✓							
UWE	✓		✓										✓		✓							
WebML	✓	✓	✓		✓				✓	✓			✓		✓	✓						
NDT	✓							✓				✓			✓						✓	✓
DDDP	✓	✓	✓			✓							✓		✓	✓	✓	✓	✓			

(1) Subcaracterística de la Eficiencia

(2) Subcaracterística de la Funcionalidad

(3) Subcaracterística de la Portabilidad

TABLA 5.5: Requerimientos No Funcionales identificados y tratados por cada metodología de desarrollo Web.

Una mirada integradora posiciona a DDDP como el enfoque que más Requerimientos No Funcionales identifica, a pesar que ofrece un proceso de desarrollo parcial en cuanto a alcance y el nivel de especificación del proceso.

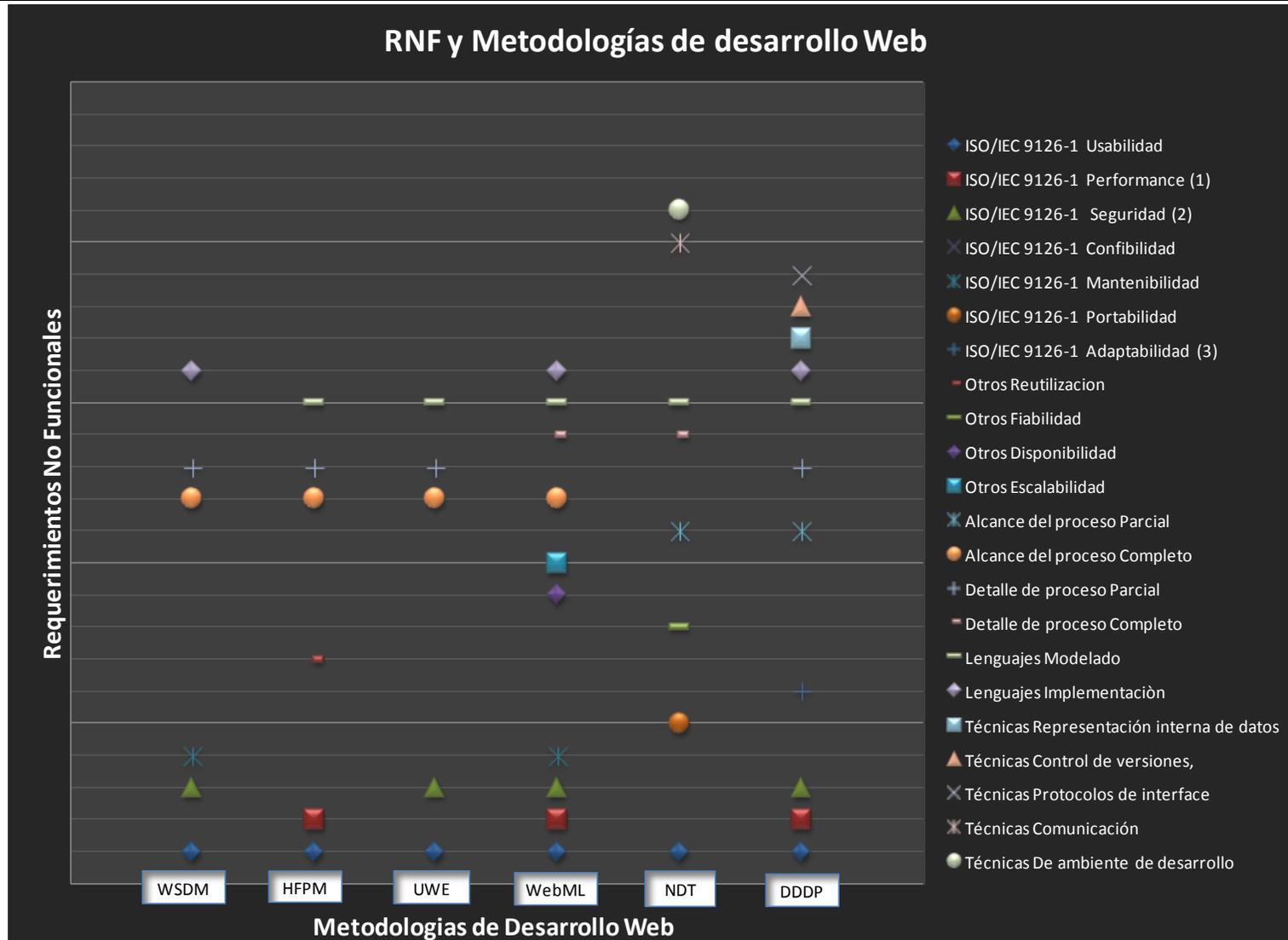


FIGURA 5.2: Requerimientos No Funcionales y Metodologías de desarrollo Web.

5.2 Técnicas de elicitación, especificación, validación y administración para tratamiento de Requerimientos No Funcionales.

Esta sección presenta el resultado del análisis enfocado a establecer que técnicas de elicitación, especificación y validación, mencionadas en el capítulo 3, se aplican al tratamiento de los Requerimientos No Funcionales y que herramientas se proponen para su administración, de acuerdo a las herramientas presentadas en el capítulo 4.

Elicitación: Ninguno de los procesos de desarrollo analizados, mencionan con exactitud cual/es técnicas de elicitación es/ son utilizadas para identificar Requerimientos No Funcionales. Si bien algunos de ellos como UWE, WebML, mencionan el uso de Checklist durante la captura de requerimientos, no señalan que estos se apliquen puntualmente para capturar los Requerimientos No Funcionales.

Especificación: WebML y NDT hacen referencia a técnicas para la especificación de Requerimientos No Funcionales.

WebML, propone generar una “Lista de Requerimientos No Funcionales”, pero no ofrece lineamientos, ni una estructura de organización para los mismos.

NDT, ofrece “Patrón para describir los Requerimientos No Funcionales”, el cual es un template que representa un producto dentro de su proceso.

Validación: Cinco de los enfoques analizados consideran la validación de requerimientos como una actividad relevante dentro del marco de la ingeniería de requerimientos. Sobre los Requerimientos No Funcionales puntualmente se puede observar lo siguiente:

HFBM, propone en la primera fase de su proceso modelar los Requerimientos No Funcionales, si bien menciona el uso de prototipo, no restringe su aplicación a la fase de Requerimientos, sino que ofrece libertad para aplicarlo durante las fases de diseño e implementación.

UWE, si bien no propone lineamientos para documentar Requerimientos No Funcionales, hace alusión al modelado de Interfaz de usuario a través de prototipos, durante la fase de análisis y diseño.

WebML, propone realizar test de aceptación para validar los Requerimientos No Funcionales, detallados en una “Lista de Requerimientos No Funcionales”.

NDT, menciona el uso de Revisiones, Auditorias, Matriz de trazabilidad, Prototipos, para validar requerimientos, pero no dice puntualmente cuál de ellas es aplicada para validar los Requerimientos No Funcionales, detallados en el “Patrón para describir los Requerimientos No Funcionales” que ofrece su proceso.

DDDP, este enfoque propone analizar las restricciones mediante diseño de prototipos durante la fase de exploración de su proceso, la cual contempla actividades de identificación, análisis y diseño de necesidades.

El enfoque que no incluye a la validación como parte del proceso de Ingeniería de requerimiento es: WSDM.

Administración de requerimientos: Tres de los seis enfoques analizados, son soportados por herramientas CASE, que permiten entre otras actividades, la administración y trazabilidad de requerimientos.

ArgoUML, da soporte al proceso de UWE y la administración de Requerimientos. Esta herramienta presenta una limitación, no ha evolucionado a UML 2.0.

TotiiSoft y WebRatio dan soporte al proceso de WebML. TotiiSoft permite la administración de requerimientos mientras que WebRatio no.

NDT-Tool, brinda soporte y permite la administración de los requerimientos identificados con el proceso de NDT. Con lo cual, solo se limita a la fase de relevamiento y análisis.

Puesto que WebML y NDT ofrecen elementos para especificar los Requerimientos No Funcionales, las herramientas que dan soporte a estos procesos TotiiSoft y NDT-Tool, permiten la administración de Requerimientos No Funcionales.

La tabla 5.6 resume lo expuesto en esta sección.

	Técnicas			Herramienta CASE	
	Elicitación/ Captura	Especificación/ Definición	Validación	Soporta el proceso de desarrollo	Soporta la Administración de Requerimientos
WSDM					
HFPM			Prototipos		
UWE			Prototipos	ArgoUML	ArgoUML
WebML		Lista de requerimientos no funcionales	Test de aceptación	TotiiSoft WebRatio	TotiiSoft
NDT		Patrón para describir los requerimientos no funcionales	- Revisiones - Auditorias - Matriz de trazabilidad - Prototipos	NDT-Tool	NDT-Tool
DDDP			Prototipos		

TABLA 5.6: Técnicas de Ingeniería de requerimientos y herramientas CASE aplicadas a RNF

5.3 Ciclo de vida y los Requerimientos No Funcionales

En la sección 5.1 se presentaron los tipos de Requerimientos No Funcionales que son identificados por las metodologías WSDM, HFPM, UWE, WebML, NDT, DDDP. En esta sección el análisis de cada enfoque está orientado a establecer: (a) en qué fase del Ciclo de Vida se identifica y es tratado cada Requerimiento No Funcional (ver Tabla 5.7), y (b) para cada enfoque metodológico se analiza la influencia en el Ciclo de Vida de las restricciones de proceso (ver tabla 5.8.)

Debido a que la estructura de fases que proponen los enfoques metodológicos estudiados difieren notablemente, para poder hacer la comparación en este trabajo se contemplan las fases del ciclo de vida en forma general: Relevamiento, Análisis, Diseño, Implementación y Prueba.

- La *mantenibilidad* es identificada por dos enfoques, uno lo hace en el relevamiento y el otro durante el diseño e implementación.
- La *portabilidad* es identificada por uno de los enfoques más recientes, durante el relevamiento.
- La *adaptabilidad* (subcaracterística de la portabilidad), es identificada por un solo enfoque recién en la implementación.
- La *reutilización*, la *fiabilidad*, la *disponibilidad*, la *escalabilidad* son identificadas durante el relevamiento.
- Todas las *restricciones del proceso de desarrollo* son identificadas en la fase de relevamiento.
- Las *restricciones de lenguaje* surgen en las fases de Diseño e implementación.
- Las *restricciones técnicas de Comunicación y Ambiente de desarrollo* son identificadas por NDT en la fase de relevamiento.
- Las *restricciones técnicas de Representación interna de datos, Control de Versiones y Protocolos de Interface* son identificadas por DDDP en la fase de implementación.
- Como conclusión general de la tabla 5.7 se desprende que existen Requerimientos No Funcionales que no son identificados en la fase de relevamiento; y el hecho de que surjan tarde en el ciclo de vida, puede afectar la calidad de la aplicación web resultante.

La otra perspectiva analizada en esta sección, es cómo influyen las restricciones de proceso, presentadas en el capítulo 2 de este trabajo, en el ciclo de vida de desarrollo. La tabla 5.8 refleja la forma en la que cada metodología aborda el tratamiento de los Requerimientos No Funcionales a lo largo del ciclo de vida.

El símbolo ‘S’ se utiliza para indicar que el enfoque metodológico identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada y que el proceso propone actividades, técnicas y ofrece detalles de los productos a generar para el tratamiento de los mismos.

El símbolo ‘I’ indica que a pesar de que el enfoque metodológico identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada, el proceso no propone una descripción suficiente de las técnicas y productos involucrados en su tratamiento.

El símbolo ‘N’ indica que el enfoque metodológico no identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada.

	Relevamiento	Análisis	Diseño	Implementación	Prueba
WSDM	I	I	I	I	N
HFPM	I	I	I	I	N
UWE	I	I	I	I	N
WebML	S	I	I	I	I
NDT	S	S	N	N	N
DDDP	I	I	I	I	N

TABLA 5.8: Influencia de las Restricciones de Proceso en el Ciclo de Vida.

Del análisis de la tabla 5.8 se pueden extraer las siguientes conclusiones:

- WebML es el único proceso que identifica y trata los Requerimientos No Funcionales a lo largo del todo el ciclo de vida de desarrollo. Aunque solamente ofrece técnicas y detalles de los productos que se deben generar para la fase de relevamiento.
- WSDM, HFPM, UWE, DDDP identifican y tratan requerimientos no funcionales durante las fases de relevamiento, análisis, diseño e implementación, pero en la fase de prueba no hacen mención al tratamiento de los mismos.
- NDT, el alcance de su proceso no cubre todas las fases del ciclo de vida de desarrollo; pero ofrece un nivel de detalle de la definición del proceso completo para las fases que cubre. NDT especifica las actividades, las técnicas y ofrece detalles de los productos a generar en las fases de relevamiento y análisis.
- Solo dos de los seis enfoques estudiados, ofrecen técnicas y brindan el detalle de los productos a generar en la fase de relevamiento para el tratamiento de los Requerimientos No Funcionales.
- Cinco de los seis enfoques estudiados no ofrecen lineamientos para las actividades de la fase de prueba, lo cual no permite establecer como realizan estos la validación de los Requerimientos No Funcionales identificados en las etapas iniciales del ciclo de vida.

6. Conclusiones

El objetivo de este trabajo fue presentar un estudio comparativo de metodologías de desarrollo web, destacando principalmente que tipos de Requerimientos No Funcionales identifica cada enfoque en sus diferentes fases del proceso y cuáles de las técnicas propuestas por de ingeniería de requerimientos son aplicadas para su tratamiento y gestión. Para ello se comenzó presentando un estado de arte de los conceptos que definen a los Requerimientos No Funcionales dentro de la literatura existente en la Ingeniería de Requerimientos, quedo expuesto que la carencia de consenso surge a partir de la diversidad de términos utilizados en las definiciones, con significados poco claros, que dan lugar a la ambigüedad sobre su alcance o lo que representan los mismos; las discrepancias conceptuales también se encontraron en las clasificaciones propuestas para Requerimientos No Funcionales, que incluyen árboles o categorías de conceptos que se muestran en forma separada en otras definiciones. Luego se establecieron las bases conceptuales a partir de lo expuesto por Cysneiros [7], se considera que los Requerimientos No Funcionales son *requerimientos de calidad y son restricciones*.

Se analizaron los procesos de desarrollo de WSDM, HFPM, UWE, WebML, NDT, DDDP en forma completa, se identificaron los tipos de Requerimientos No Funcionales, contemplados por cada uno y las técnicas de ingeniería de requerimientos propuestas por estos para su tratamiento y gestión. Además se estableció en cual/es fase/s del ciclo de vida de desarrollo de software son tratados los Requerimientos No Funcionales.

Los resultados del estudio comparativo se detallan específicamente para cada punto de vista analizado en el capítulo 5.

Los interrogantes que dieron origen al trabajo y sus respuestas son:

- 1- ¿Se contemplan los Requerimientos No Funcionales en las metodologías de aplicaciones Web?
Si, a pesar de la carencia de consenso en su significado.
- 2- ¿Utilizan técnicas de la ingeniería de requerimientos para el tratamiento de los Requerimientos No Funcionales?,

En los enfoques estudiados se puede ver que no hay técnicas específicas para la elicitación de Requerimientos No Funcionales, no hay consenso sobre como especificarlos y tampoco lineamientos de como validarlos.

3- ¿Hay herramientas que soporten a las metodologías de desarrollo web?

Existen herramientas, pero las mismas son particulares del proceso metodológico al que soportan.

4- ¿Cómo es el tratamiento de los Requerimientos No Funcionales en el ciclo de vida de desarrollo?

Los procesos de desarrollo de los enfoques estudiados no brindan técnicas que soporten el tratamiento de los Requerimientos No Funcionales a lo largo del ciclo de vida.

Hay carencia de consenso para establecer en que fase del ciclo de vida de desarrollo se identifican los Requerimientos No Funcionales. De hecho hay Requerimientos No Funcionales que no son identificados durante la fase de relevamiento y análisis.

Si bien actualmente la ingeniería de requerimientos proporciona numerosas técnicas y herramientas para identificar, describir, validar y gestionar requerimientos, estos no son aplicados muy a menudo por los profesionales de sistemas, particularmente en el desarrollo de aplicaciones web. La madurez del proceso de ingeniería de requerimientos parece ser insuficiente y demanda que surjan nuevos enfoques o se evolucionen de los enfoques existentes para el tratamiento de los Requerimientos No Funcionales en particular.

Por su importancia, las actividades para captura, especificación y validación de los Requerimientos No Funcionales, merece contar con técnicas y lineamientos específicos.

Un trabajo a futuro, a partir de este podría enfocarse en proponer un conjunto de técnicas y lineamientos específicos para la aplicación de Ingeniería de Requerimientos Web al tratamiento de los Requerimientos No Funcionales.

7. Bibliografía

1. Escalona, M. J., Koch, N.: *Requirements Engineering for Web applications – A Comparative Study*. Journal of Web Engineering, Vol. 2 No. 3, pp. 193-212 (2004).
2. Escalona, M. J., Koch, N.: *Metamodeling the Requirements of Web Systems*. J. Filipe, J. Cordeiro, and V. Pedrosa (Eds): WEBIST 2005/2006, LNBIP 1, pp. 267-280. Springer, Heidelberg (2007).
3. IEEE Standard 610. *IEEE Standard Glossary of Software Engineering Terminology*. Std 610.12-1990, IEEE, New York. (1990).
4. Cysneiros, L.M.: *Requerimientos No Funcionales desde la Elicitación al Modelo Conceptual*. Tesis Doctoral. Departamento de Informática. Politécnica Universidad Católica de Río de Janeiro (2001).
5. Kappel, G., Pröll, B., Reich, S., Retschitzegger, W.: *Web Engineering. The Discipline of Systematic Development of Web Applications*. G. Kappel, B. Pröll, S. Reich, & W. Retschitzegger (eds). John Wiley & Sons Inc. (2006).
6. Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. *Web Engineering: Modelling and Implementing Web Applications*. London: Springer (2008).
7. Cysneiros L.M. and Yu E.: *Non-Functional requirements Elicitation*. “Book” Chapter #, pp. 1-24 (2004).
8. Chung, L., Nixon, B.: “*Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach*” Proc. 17th Int. Con. on Software Eng. Seattle, Washington, April pp: 24-28, 1995.
9. De Troyer, O., Leune, C.: *WSDM: A User Centered Design Method for Web Sites*. Technical Report of Tilburg University, Infolab. Belgium (1997).
10. Olsina, L.: *Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy*. 1st International Workshop on Hypermedia Development, Hypertext’98, Pittsburg, USA. (1998).
11. Schwabe D., Rossi G: *Developing Hypermedia Applications using OOHDM*. Workshop on Hypermedia Development Process, Methods and Models, Hypertext’98, Pittsburg, USA. (1998).
12. Lange D.B: *An Object-Oriented Design Approach for Developing Hipermedia Information Systems*. Research Report RT00112, IBM Research, Tokyo Research Laboratory, Japón, (1995).
13. Isakowitz, T.; Stohr, E.; Balasubramanian, P.: *RMM: a methodology for structured hypermedia design*. Comm (D.B. 1995). ACM 38, 8; pp. 34-48. (1995).

14. Hennicker, R., Koch, N.: *A UML-based Methodology for Hypermedia Design*. Lecture Notes in Computer Science. Proc. UML'2000. York, England. (2000).
15. Koch, N., Wirsing, M.: *Software Engineering for Adaptive Hypermedia Applications ?* Poster Presentation, Third Workshop on Adaptive Hypertext and Hypermedia at the 8th International Conference on User Modeling. Sonthofen, Germany; pp 13-17. July (2001).
16. Koch N.: *Classification of Model Transformation Techniques used in UML-based Web Engineering*. Ludwig-Maximilians-Universität. Oettingenstr. 67, 80538. And FAST GmbH Arabellastr. 17, 81925 München, Germany.
17. Jacobson I., Booch G., Rumbaugh J.: *The Unified Software Development Process*. Addison Wesley. (1999).
18. Booch G., Rumbaugh J., Jacobson I.: *The Unified Modeling Language: A User Guide*. Addison Wesley (1999).
19. <http://argouml.tigris.org/>
20. Koch, N., Kraus, A., Hennicker, R.: *The Authoring Process of the UML-based Web Engineering Approach*. Institute of Computer Science. Ludwig-Maximilians University of Munich - Oettingenstr. 67, D-80538. And F.A.S.T. Applied Software Technology GmbH Arabellastr. 17, D-81925 München, Germany.
21. Ceri, S., Fraternali, P., Bongio, A.: *Web Modeling Language (WebML): a modeling language for designing Web sites*. Dipartimento di Elettronica e Informazione, Politecnico di Milano. Piazza L. da Vinci, 32-20133, Milano, Italy. (2000).
22. Ceri, S. Fraternali, P., Bongio, A., Brambilla M., Comai S., Matera, M.: *Designing Data-Intensive Web Applications*. Morgan Kaufman. (2003).
23. http://www.txtgroup.com/newsletter/attachment/w3i3_paper.pdf
24. <http://www.webratio.com/sv1.do>
25. Escalona, M.J., Mejías, M., Torres, J., Jurado, M.C.: *NDT: Navigational Development Techniques*. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Avenida Reina Mercedes, S/N 41012. Sevilla.
26. Escalona, M.J., Mejías, M., Torres, J.: *Developing Systems with NDT and NDT-Tool*. Department of Computer Languages and Systems University of Seville. ETS Ingeniería Informática. AV. Reina Mercedes S/N 41012 Sevilla, Spain.
27. Escalona, M., J.: *Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software – Tesis Doctoral*. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Avenida Reina Mercedes, S/N 41012. Sevilla. Anexo A Manual de referencia NDT p. 188-236. (2004).

28. Lowe, D.: *Web System Requirements: an overview*. Requirements Eng. Vol. 8. 102–113. - 10.1007/s00766-002-0153-x, DOI (2003).
29. Lowe, D., Eklund, J.: Client Needs and the design process in web projects. *Journal of Web Engineering*, Vol. 1, No.1 - 023-036. © Rinton Press. (2002).
30. Cysneiros, L.M., Leite, J.C. S.: *NonFunctional Requirements: From Elicitation to Conceptual Models*. Published by the IEEE Computer Society, (2004).
31. Glinz, M.: *On-Non-Functional Requirements*. 15th IEEE International Requirements Engineering Conference. 21-26. DOI 10.1109/RE.2007.45. IEEE Computer Society, (2007).
32. Chung, L., Leite, J.C. S.: *On-Non-Functional Requirements in Software Engineering*. A.T.Borgida et al. (Eds.): Mylopoulos Festschrift, LNCS 5600, 363–379. © Springer – Verlag Berlin Heidelberg. (2009).
33. Sutcliffe, A. “On the Inevitable Interwining of Requirements and Architecture”. *Design Requirements Workshop* (págs. 168-185). Cleveland: Springer. (2007).
34. ISO/IEC 9126-1:2001: *Software Engineering - Product Quality - Part 1: Quality Model* (2001).
35. Loucopoulos, P.; Karakostas, V.: *System Requirements Engineering*. Copyright© McGraw-Hill Company (UK) Limited –London (1995).