



UNIVERSIDAD
POLITECNICA
DE VALENCIA



WUEP: Un Proceso de Evaluación de Usabilidad Web Integrado en el Desarrollo de Software Dirigido por Modelos

Tesina de Máster en Ingeniería del Software, Métodos Formales y Sistemas de Información (ISMFSI)

*Grupo de Ingeniería del Software y Sistemas de Información (ISSI)
Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)*

Diciembre 2009

Estudiante:

Adrián Fernández Martínez

Directores:

Dra. Silvia Abrahão Gonzales

Dr. Emilio Insfran Pelozo

AGRADECIMIENTOS

A mis directores de tesis Silvia Abrahão y Emilio Insfran por su esfuerzo, dedicación, motivación, apoyo y confianza que me han brindado.

A Rita Hurtado por su plena confianza en mí, por estar siempre a mi lado en los buenos y malos momentos, y por sus interminables ánimos.

A mis padres, por el apoyo que me ofrecieron, por la formación y por fomentar en mí el espíritu de querer estar aprendiendo siempre.

Por último, a todo el grupo ISSI, puesto que ir cada día a trabajar sin su compañía no sería lo mismo.

CONTENIDO

Resumen	7
Abstract.....	8
Resum	9
1. Introducción	10
1.1. Motivación.....	10
1.2. Problemática en la evaluación de la usabilidad.....	11
1.3. Objetivo de la tesina	13
1.4. Contexto de la tesina	14
1.5. Estructura de la tesina	14
2. Estado del arte.....	16
2.1. Revisión sistemática sobre métodos de evaluación de usabilidad Web	16
2.1.1. Fase de Planificación.....	17
2.1.2. Fase de Ejecución.....	25
2.1.3. Fase de Reporte	27
2.1.4. Discusión.....	37
2.2. Estándares relacionados con la evaluación de la usabilidad	40
2.2.1. Estándares orientados al proceso: ISO 9241 e ISO 13407	41
2.2.2. Estándares orientados al producto: ISO 9126 e ISO 14598	43
2.2.3. Estándar ISO 25000 (SQuaRE).....	47
2.3. Propuestas de evaluación de usabilidad Web basadas en estándares.....	50
3. WUEP: un proceso de evaluación de usabilidad Web para DSDM.....	56
3.1. Integración de Evaluaciones de Usabilidad en el Desarrollo Web Dirigido por Modelos....	56
3.2. Modelo de Usabilidad Web	60
3.2.1. Vista del modelo de usabilidad centrada en la calidad producto software.....	61
3.2.2. Vista del modelo de usabilidad centrada en la calidad en uso	68
3.2.3. Métricas Web genéricas	71
3.3. Definición del proceso de evaluación WUEP	81
3.3.1. Introducción a SPEM 2.....	81
3.3.2. WUEP definido en SPEM 2.....	84

4. Instanciación de WUEP en un método de desarrollo Web concreto	98
4.1. Introducción al método OO-H	98
4.2. Caso de estudio: Gestor de tareas	103
4.2.1. Modelo de casos de uso	103
4.2.2. Modelo de clases	104
4.2.3. Modelo de navegación y presentación	105
4.2.4. Aplicación Web final (Modelo de código).....	115
4.3. Instanciación de WUEP	118
4.3.1. Requisitos evaluación	118
4.3.2. Especificación de la evaluación.....	120
4.3.3. Diseño de la evaluación	128
4.3.4. Ejecución de la evaluación	128
4.4. Lecciones aprendidas.....	143
5. Conclusiones y Trabajos Futuros	146
5.1. Conclusiones	146
5.2. Trabajos futuros.....	150
5.3. Publicaciones relacionadas	152
Referencias.....	154
Anexo A: Lista de artículos incluidos en la revisión sistemática.	158
Anexo B: Modelo de Usabilidad Web	165
Anexo C: WUEP definido en SPEM 2 usando EPF-Composer	170

LISTA DE FIGURAS

Fig 2.1. Fases de una revisión sistemática.	17
Fig 2.2. Resultados de la fase de ejecución.	26
Fig 2.3. Porcentajes resultantes en cada criterio de extracción.	30
Fig 2.4. Número de publicaciones por fuente en los últimos años.....	36
Fig 2.5. Actividades propuestas por la ISO 13405 para el Diseño Centrado en el Usuario.....	42
Fig 2.6. Ciclo de vida de la calidad en la ISO 9126.	44
Fig 2.7. Proceso de evaluación de la ISO 14598 en conjunto con la ISO 9126.....	46
Fig 2.8. Modelo de medición de la calidad del producto software según SQuaRE.....	48
Fig 2.9. Vistas de los modelos de calidad según SQuaRE.....	49
Fig 3.1. Integración de evaluaciones de usabilidad en el contexto DSDM	59
Fig 3.2. Idea central de SPEM para representar procesos	81
Fig 3.3. Primitivas de modelado en SPEM 2.....	82
Fig 3.4. Proceso de desarrollo Web DSDM genérico en SPEM 2	84
Fig 3.5. Proceso de evaluación de usabilidad WUEP en SPEM 2	87
Fig 3.6. Fase “Especificación de Requisitos de Evaluación” de WUEP en SPEM 2	88
Fig 3.7. Fase “Especificación de la Evaluación” de WUEP en SPEM 2.....	90
Fig 3.8. Fase “Diseño de la Evaluación” de WUEP en SPEM 2	94
Fig 3.9. Fase “Ejecución de la Evaluación” de WUEP en SPEM 2	97
Fig 4.1. Proceso de desarrollo Web de OO-H	100
Fig 4.2. Elementos constructores de los NAD en OO-H	102
Fig 4.3. Modelo de casos de uso para TaskManager	104
Fig 4.4. Modelo de clases para TaskManager	105
Fig 4.5. NAD-0: NAD de primer nivel para TaskManager.....	106
Fig 4.6. APD-0: APD refinado y asociado al NAD-0	107
Fig 4.7. NAD-1: NAD de gestión de tareas	108
Fig 4.8. APD-1: APD refinado y asociado al NAD-1	110
Fig 4.9. NAD-2: NAD de gestión de contactos.....	111
Fig 4.10. APD-2: APD refinado y asociado al NAD-2	112
Fig 4.11. NAD-3: NAD de gestión de informes	113
Fig 4.12. APD-3: APD refinado y asociado al NAD-3	114
Fig 4.13. Representación de la plantilla Tlayout	115
Fig 4.14. IUF-1: Interfaz de usuario final para la gestión de tareas	116
Fig 4.15. IUF-2: Interfaz de usuario final para la gestión de contactos.....	117
Fig 4.16. IUF-3: Interfaz de usuario final para la gestión de informes.....	118
Fig 4.17. IUF: Interfaz de usuario final en distintos navegadores.....	139
Fig 4.18. Cambios realizados en el modelo de clases	141
Fig 4.19. Cambios realizados en el modelo de navegación	142
Fig 4.20. Cambios realizados en el modelo de presentación.....	143

LISTA DE TABLAS

Tabla 2.1. Resultados por criterios de extracción y fuentes.....	27
Tabla 2.2. Datos empíricos extraídos de los experimentos	34
Tabla 2.3. Vistas de la Calidad en Uso según las partes beneficiarias [8].....	49
Tabla 3.1. Descomposición de la sub-característica “Facilidad de entendimiento”.....	61
Tabla 3.2. Descomposición de la sub-característica “Facilidad de aprendizaje”.....	63
Tabla 3.3. Descomposición de la sub-característica “Facilidad de uso”.....	64
Tabla 3.4. Descomposición de la sub-característica “Facilidad de ayuda”.....	66
Tabla 3.5. Descomposición de la sub-característica “Accesibilidad técnica”.....	66
Tabla 3.6. Descomposición de la sub-característica “Grado de atracción”.....	67
Tabla 3.7. Descomposición de la sub-característica “Adherencia a normas o convenciones”.....	68
Tabla 3.8. Descomposición de la sub-característica “Efectividad en uso”.....	69
Tabla 3.9. Descomposición de la sub-característica “Eficiencia en uso”.....	70
Tabla 3.10. Descomposición de la sub-característica “Satisfacción en uso”.....	70
Tabla 3.11. Descomposición de la sub-característica “Usabilidad en uso adherida a normas”.....	71
Tabla 3.12. Subconjunto de elementos para modelar procesos en SPEM 2.....	82
Tabla 4.1. Plantilla para Informe de usabilidad.....	127
Tabla 4.2. Problema de usabilidad detectado: UP001.....	129
Tabla 4.3. Matriz de distancias convertidas para Cp(NAD-0).....	131
Tabla 4.4. Problema de usabilidad detectado: UP002.....	132
Tabla 4.5. Problema de usabilidad detectado: UP003.....	134
Tabla 4.6. Problema de usabilidad detectado: UP004.....	135
Tabla 4.7. Problema de usabilidad detectado: UP005.....	136
Tabla 4.8. Problema de usabilidad detectado: UP006.....	137
Tabla 4.9. Problema de usabilidad detectado: UP007.....	138
Tabla 4.10. Problema de usabilidad detectado: UP008.....	139
Tabla 4.11. Problema de usabilidad detectado: UP009.....	140

RESUMEN

Las aplicaciones Web son consideradas actualmente un elemento esencial e indispensable en toda actividad empresarial, intercambio de información y motor de redes sociales. La usabilidad, en este tipo de aplicaciones, es reconocida como uno de los factores clave más importantes, puesto que la facilidad o dificultad que los usuarios experimentan con estas aplicaciones determinan en gran medida su éxito o fracaso. Sin embargo, existe cierta problemática en la evaluación de usabilidad Web, debido a que la mayoría de propuestas actuales sólo tienen en cuenta aspectos muy concretos, sin llegar a contemplar de forma integral el concepto usabilidad, y en la mayoría de los casos, plantean la evaluación de usabilidad en las fases finales del desarrollo, donde se requiere invertir más recursos en el mantenimiento de la aplicación Web final.

La presente tesina de máster pretende contribuir a la mejora del contexto anterior proponiendo WUEP (Web Usability Evaluation Process), un proceso de evaluación de usabilidad Web genérico integrado en el Desarrollo de Software Dirigido por Modelos (DSDM). WUEP tiene la finalidad de mejorar la usabilidad de aplicaciones Web basándose en un proceso de evaluación bien definido conforme a la reciente norma ISO 25000 (SQuaRE). El proceso de evaluación propuesto hace uso de un Modelo de Usabilidad Web, definido también conforme a SQuaRE, que descompone la usabilidad en sub-características y atributos medibles a los que se les asocian métricas definidas genéricamente. Estas métricas se operacionalizan para poder ser aplicadas a artefactos de diferentes métodos de desarrollo Web y en distintos niveles de abstracción, permitiendo de esta forma, evaluar la usabilidad de forma iterativa y en distintas fases del proceso de desarrollo Web, especialmente en fases tempranas.

Esta propuesta se apoya en un amplio estudio del estado del arte, realizado mediante la conducción de una revisión sistemática, acerca de los métodos de evaluación de usabilidad existentes para el ámbito Web, el estudio de los estándares directamente relacionados con la evaluación de la calidad de productos software y un análisis de las diferentes propuestas existentes. Los resultados obtenidos de los estudios anteriores fueron empleados como punto de partida para la definición del Modelo de Usabilidad Web y la definición del proceso de evaluación de usabilidad que proporciona guías de cómo usar el modelo de usabilidad propuesto. Finalmente, WUEP fue instanciado en un método de desarrollo Web concreto (método OO-H) con el objetivo de mostrar la factibilidad de realizar evaluaciones de usabilidad en distintas fases de un proceso de desarrollo Web Dirigido por Modelos y a distintos niveles de abstracción.

ABSTRACT

Web applications have become the backbone of business, information exchange, and social networks. In this kind of applications, usability is considered as one of the most important quality factor, since the ease or difficulty that users experience with this kind of systems will determine their success or failure. However, there are several difficulties related to Web usability evaluation, since the majority of current proposals only take into account specific usability aspects, without supporting the overall concept of usability. In addition, in most cases, usability evaluations are performed mainly at late stages of the Web development lifecycle, requiring more resources to invest in the maintenance of the final Web application.

This master's thesis aims at contributing to improve the quality of Web applications by proposing WUEP (Web Usability Evaluation Process), a generic usability evaluation process integrated into Model-Driven Web development processes. The aim of WUEP is to improve the usability of Web applications, and it is based on a well-defined evaluation process according to the latest ISO 25000 standard (SQuaRE). The proposed evaluation process uses a Web Usability Model that is also defined according to the SQuaRE. This Web Usability Model decomposes usability into sub-characteristics and measurable attributes, which are then associated to generic metrics. These metrics must be operationalized in order to be applied to artifacts of different Web development methods and at different abstraction levels, thus allowing evaluating usability at several stages of the Web development process, especially at early development stages.

This approach is based on an extensive survey of the state of the art, performed as a systematic review, about usability evaluation methods in the Web engineering field, the study of standards that are directly related to assessing the quality of software products, and the analysis of current proposals for usability evaluation. These results were used as a starting point for the definition of the Web Usability Model, and the definition of the usability evaluation process, which provides guidelines on how to use the usability model proposed. Finally, WUEP was instantiated into a concrete Web development method (OO-H method) in order to show the feasibility of performing usability evaluations in a Model-Driven Web development process and at different abstraction levels.

RESUM

Les aplicacions Web són considerades actualment un element essencial i indispensable en tota activitat empresarial, intercanvi d'informació i motor de xarxes socials. La usabilitat, en aquest tipus d'aplicacions, és reconeguda com un dels factors clau més importants, ja que la facilitat o dificultat que els usuaris experimenten amb estes aplicacions determinen en gran manera el seu èxit o fracàs. No obstant això, hi ha certa problemàtica en l'avaluació d'usabilitat Web, pel fet que la majoria de propostes existents només tenen en compte aspectes molt concrets, sense arribar a contemplar de forma integral el concepte usabilitat, i en la majoria dels casos, plantegen l'avaluació d'usabilitat en les fases finals del desenvolupament on es requereix invertir més recursos en el manteniment de l'aplicació Web final.

La present tesina de màster pretén contribuir a la millora del context anterior proposant WUEP (Web Usability Evaluation Process), un procés d'avaluació d'usabilitat Web genèric integrat en el desenvolupament de Software Dirigit per Models (DSDM). WUEP té la finalitat de millorar la usabilitat d'aplicacions Web basant-se en un procés d'avaluació ben definit d'acord amb la recent norma ISO 25000 (SQuaRE). El procés d'avaluació proposat fa ús d'un Model d'Usabilidad Web definit, també d'acord amb SQuaRE, el qual descompon la usabilitat en sub-característiques i atributs mesurables als quals se'ls associen mètriques definides genèricament, les quals seran operacionalitzades per a poder ser aplicades a distints nivells d'abstracció i en artefactes de diferents mètodes de desenvolupament Web, permetent d'esta manera, avaluar la usabilitat de forma iterativa en distintes fases del procés de desenvolupament Web, especialment en fases primerenques.

Esta proposta es recolza en un ampli estudi de l'estat de l'art, per mitjà de la conducció d'una revisió sistemàtica sobre els mètodes d'avaluació d'usabilitat existents per a l'àmbit Web, l'estudi dels estàndards directament relacionats amb l'avaluació de la qualitat de productes Software i una anàlisi de les diferents propostes existents. Els resultats obtinguts dels estudis anteriors van ser emprats com a punt de partida per a la definició del Model d'Usabilidad Web i la definició del procés d'avaluació d'usabilitat que proporciona guies de com usar el model d'usabilitat proposat. Finalment, WUEP va ser instat en un mètode de desenvolupament Web concret amb l'objectiu de mostrar la factibilitat de realitzar avaluacions d'usabilitat en distintes fases d'un procés de desenvolupament Web Dirigit per Models i a distints nivells d'abstracció.

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

En la ingeniería del software se denomina “aplicación Web” a aquellos productos software que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador. Inicialmente, el concepto de Web era básicamente un conjunto de documentos estáticos accesibles desde cualquier parte del mundo. Esa ubicuidad sumada al desarrollo de nuevas tecnologías ha sido un aspecto esencial en la evolución hacia el concepto de las aplicaciones Web actuales, cuya misión es proveer una serie de funcionalidades y servicios al usuario, más allá del mero hecho de consultar cierta información. Esta misión ha puesto de manifiesto cómo la interacción entre usuario y aplicación Web se ha convertido en una prioridad máxima a fin de que la aplicación Web logre sus objetivos.

Las aplicaciones Web presentan una serie de características que las convierten en productos software de gran valor, de entre ellas podemos destacar algunas como la capacidad de ser accesibles y operables desde cualquier plataforma y ubicación, puesto que al no ser necesario su descarga para una posterior instalación o configuración, son capaces de distribuirse con mayor facilidad a un elevado número de usuarios. Otra característica relevante podría ser el hecho de disponer siempre una versión actualizada de la aplicación, sin involucrar al usuario en tareas de actualización. Todas estas características han propiciado que las aplicaciones Web hayan sido adoptadas actualmente como un elemento esencial e indispensable en toda actividad empresarial, intercambio de información y redes sociales [63].

Para que una aplicación Web tenga éxito no es suficiente con satisfacer todos los requisitos funcionales que se esperan de ella. La facilidad o dificultad que los usuarios experimenten con este tipo de sistemas determinará en gran medida el éxito o fracaso de las mismas. Por este motivo, la usabilidad es considerada como uno de los factores de calidad más importantes en las aplicaciones Web junto a la fiabilidad y la seguridad [66], que no sólo favorece a la experiencia del usuario, sino que es capaz de ahorrar recursos asignados al proceso de desarrollo Web, beneficiando tanto al proveedor de la aplicación Web como al usuario final. Nielsen [65] afirmaba: *“Es tan fácil ir a cualquier otra parte, que la competencia de todo el mundo está a tan sólo un click”*, esto viene a resaltar la gran importancia de la interacción del usuario con las aplicaciones Web. Si el usuario se siente frustrado al no lograr sus objetivos mientras usa una determinada aplicación Web, la alternativa más cómoda que adoptará es probar con otra aplicación Web. Esto pone de manifiesto que las aplicaciones Web deben ser usables para garantizar el éxito de las mismas. Son numerosos los beneficios que se pueden obtener si se considera incorporar la usabilidad en las aplicaciones Web [52]. Entre ellos podrían

destacarse la reducción de costes de producción, mantenimiento y soporte; incremento de la productividad del usuario al llevar a cabo sus objetivos con la aplicación; y un impacto directo en las ventas y alcance, ya que un producto más usable permite un mejor marketing y es más competitivo frente a otros. Ante este reto surge la necesidad de emplear métodos de evaluación de usabilidad elaborados específicamente para la Web y tecnologías robustas que soporten dicha evaluación en el proceso de desarrollo.

1.2. PROBLEMÁTICA EN LA EVALUACIÓN DE LA USABILIDAD

Existe una problemática en el empleo de métodos de evaluación de usabilidad referente a qué se está evaluando realmente, ya que el concepto de usabilidad no posee una definición homogénea. La definición de usabilidad más aceptada en el campo de la Interacción Persona-Ordenador (IPO) es la propuesta por la ISO 9241-11 [37]: *“La medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado”*. Esta definición es más cercana al punto de vista de la interacción del usuario. Sin embargo, la definición más empleada en el campo de la Ingeniería del Software (SE) se corresponde con la propuesta en la ISO 9126-1 [35]: *“La capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas”*. Esta definición sería más cercana a los atributos de calidad para un producto software.

Pese a que distintos autores han propuesto diferentes definiciones para el concepto de usabilidad, las dos anteriores son las más empleadas al estar recogidas en los estándares. El hecho de existir diferentes definiciones afecta directamente en cómo la usabilidad puede ser evaluada, puesto que cada método o técnica empleada podría centrarse en distintos aspectos como por ejemplo: la efectividad del usuario en la realización de tareas, la facilidad con la que un usuario aprende a usar una interfaz, etc.

Un método de evaluación de usabilidad es un procedimiento sistemático compuesto por un conjunto de actividades bien definidas [78]. Su propósito es medir una serie de características consideradas a priori como predictivas y/o informativas de la usabilidad de la aplicación, para posteriormente interpretar los resultados, que permitan cuantificar el grado de usabilidad alcanzado, siendo capaces de detectar problemas que afectan a la usabilidad de dicha aplicación. Esta evaluación será esencial para corregir los problemas detectados y mejorar la usabilidad de la aplicación final.

En el pasado, los métodos de evaluación de usabilidad surgieron con el propósito de evaluar las interfaces empleadas en la mayoría de aplicaciones de escritorio, también conocidas como WIMP (Windows, Icon, Menu Pointer). Sin embargo, el avance tecnológico de la Web ha propiciado que las interfaces Web sean cada vez más

importantes. Ante este hecho, han surgido nuevos (y adaptaciones) de métodos de evaluación de usabilidad para enfrentarse al reto de evaluar dichas interfaces.

Los métodos de evaluación de usabilidad pueden ser clasificados básicamente en dos tipos: métodos empíricos (*empirical methods*) y métodos de inspección (*inspection methods*) [78]. La mayoría de métodos empíricos se basan principalmente en el testeo con usuarios pretendiendo capturar y analizar datos acerca del uso por parte de los usuarios finales de la aplicación. En este tipo de métodos, se seleccionan un conjunto representativo de usuarios finales para que prueben el producto software o prototipo con el objetivo de completar un conjunto de tareas predefinidas mientras el evaluador (o un software específico) registra una serie de datos acerca de las acciones de dichos usuarios. Un correcto análisis de estos datos es capaz de proveer una valiosa información que permitirá detectar problemas de usabilidad surgidos en las tareas llevadas a cabo por los usuarios. Sin embargo, los métodos de inspección se basan en la revisión de diferentes aspectos de usabilidad presentes principalmente en las interfaces de usuario. Estas revisiones se apoyan en la mayoría de los casos en una serie de guías y/o patrones que permiten verificar si estos aspectos relacionados con la usabilidad se cumplen o no. Estas guías pueden ser de distinta naturaleza, desde listas de verificación de propiedades o patrones, hasta la aplicación de métricas. Los métodos de inspección son llevados a cabo principalmente por evaluadores expertos o no expertos (los propios diseñadores), mientras que los usuarios finales no intervienen en la evaluación.

En el caso concreto de las aplicaciones Web, los métodos anteriores presentan algunas ventajas e inconvenientes. La mayoría de aplicaciones Web son desarrolladas para un conjunto bastante amplio de usuarios con distintos perfiles, y por este motivo, los métodos empíricos serían una buena elección si nos basamos en que tienen en cuenta usuarios reales con distintos perfiles para realizar evaluaciones, sin embargo, el inconveniente viene dado por el hecho que llevar a cabo estos métodos suelen requerir una gran cantidad de recursos, siendo solo aplicables cuando se dispone de la aplicación final o de un prototipo bastante avanzado que permita realizar las tareas que se han diseñado para ser llevadas a cabo por los usuarios. Es por ello, que la evaluación de usabilidad tiene lugar en las últimas fases del ciclo de desarrollo Web. Por otro lado, los métodos de inspección permiten evaluar artefactos intermedios creados durante el proceso de desarrollo, especialmente en fases más tempranas, como por ejemplo, bocetos, prototipos en papel, modelos abstractos de interfaces de usuario, etc. Un beneficio añadido podría ser que la cantidad de recursos empleados en estos métodos, suele ser menor que la empleada en métodos de test de usuarios. Sin embargo, la evaluación mediante inspección puede estar limitada tanto por los aspectos de usabilidad considerados en las guías, como por las expectativas del usuario. A lo anterior, hay que sumarle el de hecho que estos métodos no contemplan la interacción del usuario final. Es por ello que un proceso de evaluación de usabilidad debería ser

capaz de integrar evaluaciones en las diferentes fases de desarrollo de una aplicación Web a fin de tener en cuenta la usabilidad a lo largo del ciclo de vida del desarrollo de la aplicación, intentando de esta forma, combinar las ventajas que ofrecen otros métodos.

La complejidad de integrar evaluaciones de usabilidad en las diferentes fases de desarrollo de una aplicación Web viene determinada principalmente por el método de desarrollo elegido. El problema principal parece ser que la mayoría de los procesos de desarrollo Web no aprovechan los artefactos producidos en las fases tempranas. Estos artefactos intermedios sirven únicamente para orientar a los desarrolladores y para documentar la aplicación Web. Dado que la trazabilidad entre estos artefactos y la aplicación Web final no está bien definida, las evaluaciones de estos artefactos podrían resultar no ser tan eficaces. Este problema no ocurre en el desarrollo de software dirigido por modelos (DSDM) donde los modelos (artefactos intermedios) que especifican una aplicación Web completa se aplican en todas las etapas del proceso de desarrollo, y el código fuente final se genera automáticamente a partir de estos modelos. La evaluación de estos modelos proporcionaría informes de evaluación de la usabilidad que presentan cambios en dichos modelos, los cuales se podrían reflejar directamente en el código fuente, sin necesidad de mantener dicho código.

1.3. OBJETIVO DE LA TESINA

El objetivo principal de la presente tesina de máster es proponer un proceso de evaluación de usabilidad que pueda ser integrado en cualquier proceso de desarrollo Web dirigido por modelos, permitiendo realizar evaluaciones de usabilidad sobre modelos producidos en diferentes fases de desarrollo.

Las metas que pretende alcanzar la presente tesina, las cuales descomponen el objetivo principal, son:

1. Analizar en profundidad el empleo de métodos de evaluación de usabilidad: qué tipos de métodos son los más usados, en qué fases se aplican, cuáles han sido validados, qué información proveen al evaluador y cuáles resultaron más efectivos.
2. Estudiar los estándares de evaluación de calidad para productos software que tratan la usabilidad como una característica de calidad; y analizar otras propuestas de evaluación de usabilidad.
3. Definir un modelo de usabilidad Web que descompone la usabilidad en sub-características, atributos medibles y métricas. Siendo éste acorde a los estándares más recientes, teniendo en cuenta las distintas definiciones de usabilidad.

4. Definir un proceso de evaluación de usabilidad Web genérico con la capacidad de ser integrado en diferentes métodos de desarrollo Web dirigido por modelos, haciendo uso del modelo de usabilidad como artefacto principal.
5. Instanciar el proceso de evaluación de usabilidad Web a un método concreto de desarrollo Web dirigido por modelos.

1.4. CONTEXTO DE LA TESINA

La presente tesina de máster se ha desarrollado en el seno del grupo ISSI (Ingeniería del Software y Sistemas de Información) y contribuye al desarrollo de los siguientes proyectos de investigación:

- Proyecto META (Models, Environments, Transformations and Applications), Subproyecto perteneciente al proyecto MOMENT: Un marco tecnológico y formal para la gestión de modelos en la ingeniería de modelos (Octubre 2006-Septiembre 2009). Financiado por el Ministerio de Educación y Ciencia - TIN2006-15175-C05-01.
- Proyecto MAUSE: Towards the Maturation of Information Technology Usability Evaluation (2005-2009). Financiado por European Union COST action - Núm. 294.
- Red CALIPSO: Calidad del producto y Proceso Software (2005-2007). Red temática de investigación financiada por el Ministerio de Ciencia y Tecnología - TIN2005-24055-E.
- Proyecto CALIMO: Integración de Calidad en el Desarrollo de Software Dirigido por Modelos (Enero 2009-Enero 2010). Financiado por la Generalitat Valenciana, Conselleria de Educación - GV/2009/103.
- Proyecto Transformación de Modelos Dirigida por Atributos de Calidad - Quality-Driven Model Transformations (Diciembre 2007-Diciembre2009). Financiado por la Universidad Politécnica de Valencia - PAID-06-07-3286.
- Proyecto MULTIPLE: Multimodeling Approach for Quality-Aware Software Product Lines (Octubre 2009-Septiembre 2013). Financiado por el Ministerio de Ciencia e Innovación - TIN2009-13838.
- Proyecto TwinTIDE: Towards the Integration of Transectorial IT Design and Evaluation (Noviembre 2009-Noviembre 2013). Financiado por European Union COST action IC0904.

1.5. ESTRUCTURA DE LA TESINA

En este capítulo se han presentado la motivación del trabajo de investigación, la problemática existente en las evaluaciones de usabilidad, el contexto de la tesina de máster y las metas que se pretende alcanzar con la presente tesina de máster. En los siguientes capítulos, la tesina se estructura de la siguiente forma:

El Capítulo 2 presenta el estado del arte acerca de los temas relacionados con la tesina: una revisión sistemática acerca de los métodos de evaluación de usabilidad Web, un estudio de los estándares de calidad que contemplan la usabilidad, y otras propuestas de evaluación de usabilidad existentes en la literatura.

El Capítulo 3 describe la contribución principal de esta tesina. Se explica brevemente la estrategia a seguir para integrar evaluaciones de usabilidad en un entorno DSDM, se describe el modelo de usabilidad Web definido junto con todas sus sub-características, atributos y métricas; y por último se detallan todas las fases que componen el proceso de evaluación de usabilidad, que ha sido llamado WUEP (*Web Usability Evaluation Process*).

El Capítulo 4 está destinado a mostrar cómo WUEP ha sido instanciado en Object Oriented Hypermedia (OO-H) [28], un método concreto de desarrollo Web basado en DSDM. Se introducen los conceptos básicos relacionados con OO-H y los modelos que emplea para definir aplicaciones Web, para posteriormente ilustrar la aplicación de WUEP en un caso de estudio, en el cual, se evaluará la usabilidad de una aplicación Web desarrollada en OO-H que consiste en un gestor de tareas. El capítulo termina con una serie de lecciones aprendidas.

Por último, el Capítulo 5 describe las conclusiones generales, los trabajos futuros y los resultados obtenidos en cuanto a publicaciones.

2. ESTADO DEL ARTE

En este capítulo se presentan los estudios que han sido realizados con fin de obtener una perspectiva global sobre la evaluación de la usabilidad en el ámbito Web, y que han servido para definir los objetivos de la presente tesina de máster.

En la Sección 2.1 se presenta una revisión sistemática sobre métodos de evaluación de usabilidad para la Web, analizando aspectos cómo qué métodos se han creado expresamente para el dominio Web, qué tipos de métodos son los más comunes y qué ventajas aportan, en qué fases del proceso de desarrollo Web suelen aplicarse, qué métodos han sido validados, qué información aportan al evaluador. Tras la revisión se plantea una discusión sobre los principales hallazgos, las limitaciones de la propia revisión y las implicaciones de los resultados obtenidos de cara al ámbito académico y empresarial.

En la Sección 2.2 se analizan brevemente las principales características de los estándares vinculados a la evaluación de la calidad de productos software, centrándose principalmente en la evaluación de la usabilidad. Estos son las normas: ISO 9241-11 [37], ISO 9126-1 [35], ISO 14598-1 [41], e ISO 25000 (SQuaRE) [42].

En la Sección 2.3 se recopilan las aportaciones existentes en la literatura acerca de procesos de evaluación basados en modelos de calidad, donde la usabilidad se considera como una característica principal.

2.1. REVISIÓN SISTEMÁTICA SOBRE MÉTODOS DE EVALUACIÓN DE USABILIDAD WEB

Una revisión sistemática es un método científico para evaluar e interpretar la información disponible que es relevante acerca de una “pregunta de investigación” concreta, un área temática o cualquier fenómeno de interés [47]. El objetivo es presentar una evaluación lo más objetiva posible sobre un tema concreto empleando una metodología fiable, rigurosa y auditable. De esta forma, estas revisiones son de gran utilidad para resumir toda la información existente sobre un tema de investigación de una manera imparcial, identificando posibles lagunas en la investigación actual con el objetivo de proporcionar un marco de trabajo donde desarrollar adecuadamente actividades de investigación futuras.

Una revisión sistemática consta de tres fases, que a su vez, pueden descomponerse en actividades (Fig. 2.1):

1. Fase de planificación: donde se identifica la *necesidad de una revisión* y el *protocolo de revisión* es definido y validado. Un protocolo de revisión específica en profundidad las siguientes actividades: establecimiento de la *pregunta de*

investigación; la *estrategia de búsqueda* a seguir incluyendo los términos de búsqueda y los recursos donde se van a aplicar; la *selección de estudios primarios* definiendo los criterios de inclusión y exclusión; la *estrategia de extracción de datos* de los estudios seleccionados; los *métodos de síntesis* aplicados a los datos extraídos; y la *validación del protocolo* describiendo todas las acciones realizadas en cada uno de las anteriores actividades con el fin de asegurar que la revisión será lo más imparcial posible.

2. Fase de ejecución: donde la revisión sistemática se ejecuta siguiendo el protocolo de revisión validado. El proceso de búsqueda se lleva a cabo según la estrategia definida, los estudios primarios se obtienen mediante la aplicación de criterios de inclusión y exclusión, de ellos se extraen los datos y se aplican los métodos de síntesis.
3. Fase de reporte: donde los resultados de la revisión son interpretados y presentados.

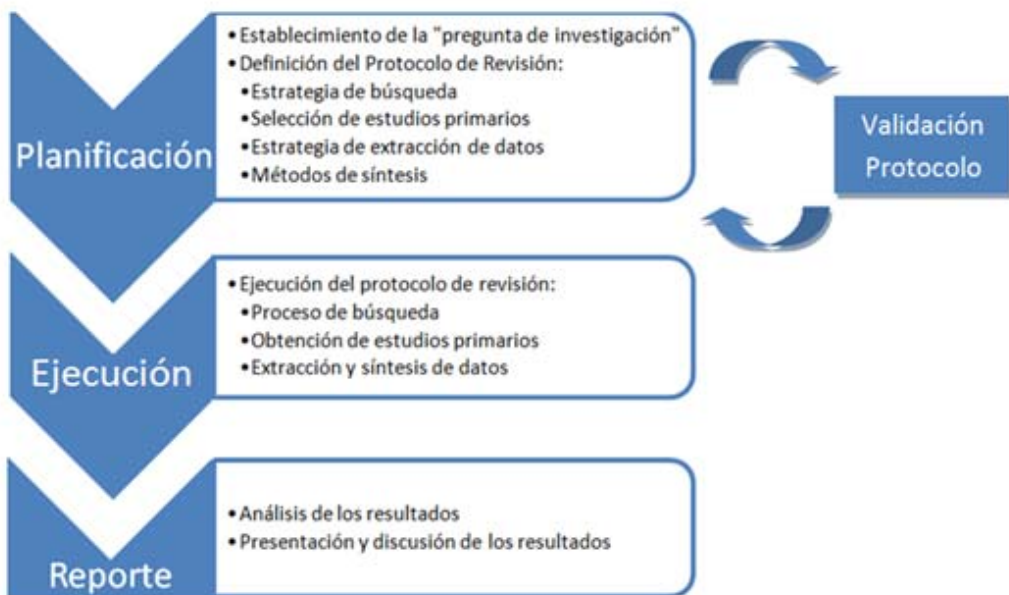


Fig 2.1. Fases de una revisión sistemática.

A continuación se describe la revisión sistemática sobre métodos de evaluación de usabilidad para la Web realizada describiendo cada una de las actividades anteriores.

2.1.1. FASE DE PLANIFICACIÓN

Aunque se han publicado diversos estudios sobre comparaciones de métodos de evaluación de usabilidad, no había constancia de ninguna revisión sistemática en el campo de la usabilidad Web hasta la fecha. La mayoría de artículos publicados son

estudios, más bien informales, sobre la literatura existente, o bien comparaciones entre métodos previamente seleccionados por el usuario que no definían ninguna pregunta de investigación y ninguna de las fases citadas anteriormente. Sin embargo, sí que se han encontrado revisiones sistemáticas realizadas en campos relacionados. Por ejemplo, Hornbæk [31] presenta una revisión acerca de medidas de usabilidad en la práctica, aunque no centrada en métodos de evaluación; Freire *et al.* [26] presenta una revisión sobre técnicas existentes para desarrollar aplicaciones Web más accesibles; y Mendes [58] presenta otra revisión sistemática investigando el rigor metodológico presente en la ingeniería Web.

Es por ello, que la necesidad de una revisión sistemática acerca de los métodos de evaluación de usabilidad Web queda justificada para reunir un mayor conocimiento del área e identificar las carencias de las investigaciones existentes.

Las siguientes sub-secciones describen las actividades realizadas en la fase de planificación. La primera establece la pregunta de investigación, mientras las restantes pertenecen a la definición del protocolo de revisión.

2.1.1.1. ESTABLECIMIENTO DE LA “PREGUNTA DE INVESTIGACIÓN”

El objetivo de la revisión es examinar el empleo de métodos de evaluación de usabilidad en el desarrollo Web desde el punto de vista de las siguientes preguntas:

- ¿Qué métodos de evaluación de usabilidad han sido empleados por investigadores para evaluar artefactos Web?
- ¿Cómo han sido empleados esos métodos?
- ¿Cuáles han resultado más efectivos al compararlos con otros?

Estas preguntas nos permitirán reunir el conocimiento actual sobre la evaluación de la usabilidad Web e identificar carencias con el objetivo de orientar futuras investigaciones. Estas preguntas han sido estructuradas siguiendo el criterio PICOC [70]: Población, Intervención, Comparación, Resultado (*Outcome*), Contexto.

- Población: Evaluación de usabilidad en la Web.
- Intervención: Métodos de evaluación de usabilidad.
- Comparación: Efectividad de los métodos de evaluación de usabilidad (cuando sea posible).
- Resultado: no centrado en el resultado en sí mismo.
- Contexto: artículos de investigación basados en evaluación de usabilidad Web.

Esta revisión sistemática es algo más limitada que la revisión sistemática propuesta que se propone en las guías de Kitchenham [47], dado que no se incluyen artículos que han sido encontrados a través de las referencias de otros artículos incluidos

previamente. Además, tampoco se han incluido otras referencias como trabajos en proceso, informes técnicos, tesis doctorales, etc.

2.1.1.2. ESTRATEGIA DE BÚSQUEDA

La estrategia de búsqueda se basa en una búsqueda automática en librerías digitales y una búsqueda manual en distintas fuentes que se consideran relevantes en el área, a fin de realizar una búsqueda lo más completa posible. Las principales librerías digitales empleadas han sido: IEEEExplore, ACM y ScienceDirect. Mientras que las fuentes manuales han sido las siguientes:

- Actas de World Wide Web conference (WWW) en temas de Usability and accessibility & Web engineering. Años: 2003, 2004, 2007 y 2008).
- Actas de International conference on Web Engineering (ICWE). Años: 2003-2008.
- Internet Research Journal: "Electronic Networking Applications and Policy". Años: 1996-2008) (ed. Emerald Group Publishing Limited)
- International Journal of Human-Computer Studies "Web Usability" Special Issue. (un volumen publicado en 1997).
- IEEE Internet Computing Special issue on "Usability and the Web" (un volumen publicado en 2002).
- Un libro sobre "Web Engineering" by Springer (LNCS) publicado en 2005.
- Actas de International Web Usability and Accessibility workshop (IWWUA). Años: 2007-2008.

Para obtener estudios candidatos, mediante una búsqueda automática, en las librerías digitales se ha definido la siguiente cadena de búsqueda:

"(usability) AND (web OR internet OR www) AND (evalu OR assess* OR measurement OR experiment* OR stud* OR testing)"*

Donde el asterisco '*' significa cualquier conjunto de caracteres. Aplicando este patrón se pretende incorporar las posibles variaciones de un mismo concepto, por ejemplo, "evalu*" haría referencia a las palabras: "evaluation", "evaluate", "evaluates", etc. Esta cadena de búsqueda fue aplicada en los mismos metadatos de los estudios: título, resumen y palabras clave. Las fechas de la búsqueda son desde 1996 hasta febrero de 2009. La selección de estas fechas viene justificada por el hecho que en 1996 nace el término "Web engineering" [58].

2.1.1.3. SELECCIÓN DE ESTUDIOS PRIMARIOS

Para cada estudio obtenido del proceso de búsqueda, tanto manual como automática, se aplicaron una serie de criterios de inclusión y exclusión. Las discrepancias se resolvieron por consenso. Los estudios acordes a los siguientes criterios fueron incluidos:

- Artículos que presentan/emplean algún método de evaluación de usabilidad que es aplicado en el desarrollo Web. Sólo estudios presentando un método formal (ej. Evaluación heurística, Paseos cognitivos, etc).
- Artículos de investigación completos (no se incluyen resúmenes o artículos cortos).

Los estudios acordes a los siguientes criterios fueron excluidos:

- Artículos que únicamente presentan recomendaciones y guías para el desarrollo Web.
- Artículos que sólo presentan métricas de usabilidad o técnicas de agregación por dichas métricas.
- Artículos que presentan procesos de testeo acerca del funcionamiento de la aplicación Web.
- Artículos introductorios (resúmenes, artículos cortos).
- Artículos que ya han aparecido en otras fuentes del proceso de búsqueda.
- Artículos no escritos en inglés.

2.1.1.4. ESTRATEGIA DE EXTRACCIÓN DE DATOS

Con el objetivo de extraer la misma información de todos los estudios seleccionados, se elaboró un formulario con una serie de criterios basados en preguntas (se adjuntan las posibles respuestas):

1. ¿Qué métodos de evaluación de usabilidad han sido empleados por investigadores para evaluar artefactos Web?
 - a) ¿Es un método Nuevo para la Web o es uno existente que proviene del campo IPO? (Nuevo, Existente).
 - b) ¿Cuál es el tipo de método de empleado? (Testeo usuarios, Inspección, Indagatorios, Modelado Analítico, Simulación).
2. ¿Cuál es la fase en la que el método se aplica? (Requisitos, Diseño, Implementación).
3. ¿Cómo es el tipo de evaluación llevada a cabo por el método? (Manual, Automática).
4. ¿El método ha sido validado? (Sí, No). En caso afirmativo:

- a) ¿Qué tipo de evaluación se llevó a cabo para evaluar el método?
(Cuestionario, Caso de estudio, Experimento)
 - b) Si la evaluación a través de un experimento controlado:
 - i. ¿Qué medidas fueron empleadas para evaluar la efectividad del método?
 - ii. ¿Qué tipos de artefactos fueron evaluados?
 - iii. ¿Se comparó el método con otros? (Sí, No). En caso afirmativo: ¿Qué método resultó más efectivo?
5. ¿La evaluación se llevó a cabo con el propósito de proveer retroalimentación a la fase de diseño? (Sí, No).

A continuación se explica cada criterio de extracción con más detalle:

Criterio 1a. El artículo se clasifica como “*nuevo*” si presenta, al menos, un método de evaluación de usabilidad que ha sido expresamente elaborado para ser aplicado en el contexto Web. Mientras que el artículo se clasifica como “*existente*” si emplea en el contexto Web un método proveniente del campo IPO.

Criterio 1b. El artículo se clasifica conforme a la taxonomía de métodos de evaluación propuesta por Ivory y Hearst [44]: *Testeo con usuarios*, *Inspección*, *Investigatorios*, *Modelado Analítico*, *Simulación*. El criterio no es exclusivo, eso quiere decir, que un artículo puede estar clasificado en más de una categoría si el/los método(s) que presenta poseen características de varias categorías. El artículo se clasifica en:

- *Testeo con usuarios*: si implica un evaluador observando a los participantes en el testeo interactuando con una interfaz de usuario para determinar problemas de usabilidad o analizando esas interacciones a posteriori, como por ejemplo, protocolos Thinking-Aloud (Pensar en voz alta), testeo remoto, análisis de logs, etc.
- *Inspección*: si implica a un evaluador experto empleando un conjunto de criterios para la identificación de problemas de usabilidad potenciales, como por ejemplo, evoluciones heurísticas, revisiones basadas en guías, Cognitive Walkthrough (Paseos cognitivos), etc.
- *Indagatorios*: si presenta un método cuyo objetivo es obtener impresiones subjetivas (preferencias, sentimientos) de los propios participantes, como por ejemplo, Focus group (discusiones en grupo), entrevistas, cuestionarios, etc.
- *Modelado analítico*: si presenta una propuesta ingenieril que permite al usuario predecir problemas de usabilidad mediante el análisis de modelos que representan las interfaces de usuario, el conocimiento y rendimiento del

usuario, el contexto de las tareas, etc. Algunos ejemplos serían el análisis GOMS, análisis cognitivo de tareas, etc.

- *Simulación*: si se presenta un método cuya finalidad es simular la interacción de usuario a través de algoritmos que interpretan datos obtenidos de usuarios, como por ejemplo, modelos de red Petri, information scent, etc.

Criterio 2. El artículo se puede clasificar en una o más fases de desarrollo propuestas por la norma ISO/IEC 12207 [39]: *Requisitos, Diseño e Implementación*. El artículo se clasifica en:

- *Requisitos*: si los artefactos que sirven de entrada al método de evaluación son especificaciones de alto nivel de la aplicación Web, como por ejemplo, casos de uso, escenarios, modelos de tareas, etc.
- *Diseño*: si los artefactos que sirven de entrada al método de evaluación son artefactos intermedios de la aplicación Web, como por ejemplo, modelos navegacionales, modelos de dialogo, modelos abstractos de interfaces de usuario, etc.
- *Implementación*: si la evaluación se lleva a cabo sobre la aplicación Web final ya desarrollada o sobre interfaces de usuario finales.

Criterio 3. El artículo se puede clasificar en:

- *Automático*: si existe una herramienta que permite llevar a cabo de forma automatizada gran parte de las actividades (o todas) del método presentado, como por ejemplo, analizadores de logs, verificadores de código fuente, simuladores de usuario, etc.
- *Manual*: si el método se lleva a cabo manualmente, esto quiere decir que el método puede estar apoyado en herramientas software, pero las principales actividades concernientes a la evaluación necesitan ser llevadas a cabo por una persona.

Criterio 4a. El artículo se clasifica según qué tipo de técnica de evaluación se ha empleado para validar el método. Según Fenton y Pfleeger [24], existen tres tipos de estrategias dependiendo del propósito de la evaluación y las condiciones para la evaluación empírica:

- *Cuestionario (survey)*: es una investigación realizada en modo retrospectivo, cuando el método de evaluación de usabilidad ha sido empleado durante un cierto periodo de tiempo.
- *Caso de estudio*: es un estudio observacional donde los datos son recolectados con propósitos específicos durante todo el estudio.

- *Experimento*: es una investigación formal, rigurosa y controlada. Los experimentos son muy útiles para comparar métodos de evaluación de usabilidad entre ellos de una forma más rigurosa.

Criterio 4b. En el caso de encontrarnos ante un experimento controlado, obtendremos la información sobre:

- **4b-i)** Qué tipos de medidas son las más empleadas para evaluar la efectividad de los métodos, tales como por ejemplo, el número de problemas de usabilidad detectados, número de actividades del método que se han llevado a cabo sin incidentes, o utilidad de los resultados obtenidos del método.
- **4b-ii)** Qué tipos de artefactos de la aplicación Web son comúnmente evaluados, tales como las interfaces de usuario final, modelos o prototipos en papel.
- **4b-iii)** Qué experimentos han sido diseñados con el objetivo de comparar métodos de evaluación de usabilidad entre sí, en cuanto a la efectividad de los mismos.

Criterio 5. El artículo se clasifica en:

- *No*: si el método de evaluación de usabilidad solo provee una lista de errores detectados o una cuantificación del grado de usabilidad de la aplicación Web.
- *Sí*: si además de proveer un lista de errores, el método provee recomendaciones y guías al diseñador para corregir los problemas de usabilidad detectados.

2.1.1.5. MÉTODOS DE SÍNTESIS

Se han aplicado dos métodos de síntesis: un método cuantitativo basado en determinar porcentajes de artículos para cada criterio de extracción y para cada fuente bibliográfica; y un método cualitativo basado en extraer conclusiones acerca de los beneficios y limitaciones que presentan los métodos de evaluación de usabilidad clasificados en base a los criterios establecidos.

En el caso particular para la síntesis de resultados del criterio 4b-iii) “¿Se comparó el método con otros?, y ¿Qué método resultó más efectivo?”, la información extraída de las comparaciones fue codificada con el fin de facilitar la interpretación de evidencias empíricas. Estas codificaciones fueron tratadas mediante una estrategia de agregación similar a la presentada en Dieste *et al* [22]. Los métodos de evaluación de usabilidad presentados por los experimentos fueron codificados como Mx, y las medidas de efectividad fueron codificadas como Ey (siendo ‘x’ e ‘y’ números secuenciales). De esta forma se construyen expresiones como:

[Código evidencia, Métodos involucrados, medidas de efectividad] efecto.

Por ejemplo, si un método de evaluación de usabilidad M_1 ha detectado más problemas de usabilidad (E_1) que otro método M_2 en ese experimento. La expresión construida sería:

$$[0001, M_1/M_2, E_1] M_1 > M_2 \quad (1)$$

La expresión (1) puede ser leída como: “La evidencia 0001 muestra que M_1 es más efectivo que M_2 conforme al número de problemas de usabilidad detectados”. Cabe resaltar que el signo ‘>’ significa “más efectivo que”, mientras que también podemos usar el signo ‘≈’ para indicar “no existen diferencias significativas”.

Estas expresiones obtenidas de cada estudio, el cual presenta un experimento, pueden ser agregadas para sintetizar resultados. El proceso de agregación trata de combinar diferentes expresiones, siempre y cuando, las medidas que hayan empleado sean las mismas. Por ejemplo, la expresión (1) se podría combinar con otra expresión (2):

$$[0002, M_2/M_3, E_1] M_2 > M_3 \quad (2)$$

Para obtener como resultado la expresión (3):

$$[0003, M_1/M_2/M_3, E_1] M_1 > M_2 > M_3 \quad (3)$$

2.1.1.6. VALIDACIÓN DEL PROTOCOLO DE REVISIÓN

La validación del protocolo de revisión se ha llevado a cabo incorporando distintas acciones sobre las actividades descritas anteriormente para la fase de planificación. El objetivo es asegurar, en la medida de lo posible, que nuestra revisión es lo más correcta, completa y objetiva posible. Con respecto a la estrategia de búsqueda (ver Sección 2.1.1.2) se han llevado a cabo las siguientes acciones:

- Se han seleccionado tres librerías digitales (IEEEExplore, ACM, y ScienceDirect) las cuales contienen un elevado número de publicaciones en los campos de la Ingeniería del Software e Interacción Persona-Ordenador.
- Se han incluido otras fuentes bibliográficas para garantizar que la estrategia de búsqueda se aplica a revistas y actas de conferencias relevantes en el área.
- La cadena de búsqueda ha sido refinada aplicando diferentes combinaciones hasta encontrar aquella que obtenía el mayor número de artículos relevantes. Además, se tuvo en cuenta que la cadena de búsqueda se aplica sobre los mismos metadatos de cada artículo, adaptándola cuando fuera necesario según cada librería digital.
- Los términos de la cadena de búsqueda se basaron en términos que aparecían en artículos relevantes en el área, cuya existencia ya era conocida.

- Se ha verificado que los artículos considerados relevantes en el área, han sido obtenidos como resultado del proceso de búsqueda.

Con respecto a la selección de estudios primarios (ver Sección 2.1.1.3) se han llevado a cabo las siguientes acciones:

- Se ha verificado que los artículos considerados relevantes en el área, han sido incluidos en la revisión tras aplicar los criterios de inclusión y exclusión.
- La fiabilidad en la inclusión de un estudio candidato en la revisión se evaluó mediante la aplicación del estadístico Fleiss Kappa [25]. Fleiss Kappa es una medida estadística para evaluar la fiabilidad del acuerdo que existe entre un número fijo de evaluadores a la hora realizar una clasificación, en este caso “incluir artículo en la revisión” o “rechazar artículo”. Los valores que toma esta medida pertenece a un intervalo entre cero y uno. Un cero indica un acuerdo muy pobre, mientras que un uno indica un acuerdo total. Tres evaluadores independientes clasificaron una muestra aleatoria de 20 estudios, 10 de los cuales fueron incluidos en la revisión y otros 10 que no lo fueron. El valor del estadístico kappa Fleiss obtenido fue de 0.84, indicando un acuerdo aceptable entre evaluadores, lo cual viene a significar que la aplicación de los criterios de inclusión y exclusión por diferentes evaluadores, introduciría una mínima variación en el conjunto de artículos seleccionados.

Por último, con respecto a la estrategia de extracción de datos (ver Sección 2.1.1.4) se han llevado a cabo las siguientes acciones:

- Los criterios de extracción se definieron de forma que pudieran responder a la pregunta de investigación establecida.
- La fiabilidad en la clasificación de artículos, según los criterios de extracción definidos, se evaluó también mediante el estadístico Fleiss Kappa. Para ello, tres evaluadores independientes clasificaron una muestra aleatoria de 10 artículos que habían sido incluidos en la revisión. Los valores Fleiss Kappa obtenidos para cada criterio de extracción fueron “0.9” para el criterio 1a, “0.76” para el criterio 1b, “0.8” para el criterio 2, “0.82” para el criterio 3, “0.91” para el criterio 4a, y “0.84” para el criterio 5. Estos valores indican que la extracción de datos se puede calificar como un proceso bastante reproducible por diferentes evaluadores.

2.1.2. FASE DE EJECUCIÓN

Una vez definido y validado el protocolo de revisión, éste fue aplicado para realizar la revisión sistemática obteniendo los siguientes resultados preliminares (ver Fig. 2.2):

- Las bibliotecas digitales identificaron un total de 733 publicaciones potenciales (366 de IEEEExplore, 254 de ACM, y 113 de ScienceDirect). Tras aplicar los criterios de inclusión y exclusión, se seleccionaron finalmente 141 publicaciones (76 de IEEEExplore, 56 de ACM digital library, y 9 de ScienceDirect).
- La bibliografía manual identificó otras 114 publicaciones potenciales. Tras aplicar los criterios de inclusión y exclusión, se seleccionaron finalmente 22 artículos (5 del WWW; 5 del ICWE; 3 del Internet Research Journal; 1 del International Journal of Human-Computer Studies “Web Usability” Special Issue; 3 de IEEE Internet Computing special issue; 1 capítulo del libro “Web Engineering”; y 4 del IWWUA).

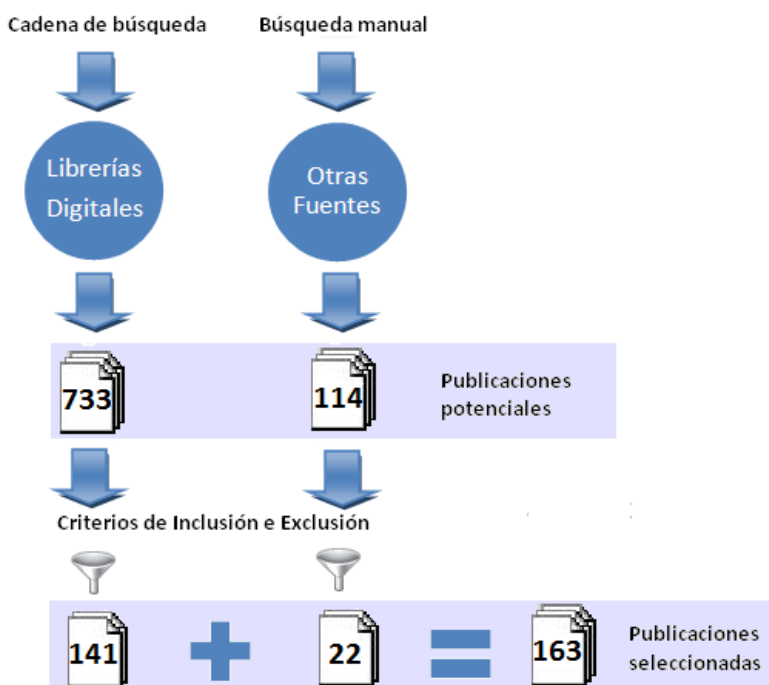


Fig 2.2. Resultados de la fase de ejecución.

De esta forma, fueron seleccionados un total de 163 artículos. Algunos aspectos a destacar en esta fase fueron:

- Algunos estudios similares del mismo autor han sido publicados en más de una fuente bibliográfica. En este caso se ha seleccionado la versión más completa del estudio.
- Algunos estudios se han encontrado replicados en distintas fuentes bibliográficas. En este caso, sólo se han tenido en cuenta una única vez siguiendo el orden de búsqueda establecido. Este orden de búsqueda fue definido arbitrariamente y se corresponde con el orden de aparición en la

sección 2.1.1.2. Por ejemplo, 1º IEEEExplore, 2º ACM, 3º ScienceDirect, 4º WWW, etc.

2.1.3. FASE DE REPORTE

En esta sección se muestran los resultados obtenidos de la revisión sistemática. La tabla 2.1 muestra los resultados organizados por criterios y fuente bibliográfica. El anexo A incluye el listado completo de artículos que han sido incluidos en la revisión.

Tabla 2.1. Resultados por criterios de extracción y fuentes.

Criterios de extracción de datos		IEEE	ACM	SD	WWW	ICWE	IR	IJH	IEEE-IC	Lib.	IWWUA	Tot
Método Evaluación Usabilidad	Nuevo	28	20	2	3	5	0	1	3	0	2	64
	Existente	48	36	7	2	0	3	0	0	1	2	99
Tipo de Método Evaluación Usabilidad	Testeo	39	38	6	4	2	1	0	2	1	1	94
	Inspección	29	18	5	0	4	0	0	0	1	2	59
	Indagatorio	29	19	4	0	0	2	0	0	1	1	56
	Modelado Analítico	13	13	2	2	2	0	1	2	1	2	38
	Simulación	5	8	2	0	0	0	0	0	1	0	16
Fase del desarrollo Web	Requisitos	3	1	0	0	1	0	0	0	0	1	6
	Diseño	17	7	5	0	3	0	1	3	1	3	40
	Implementación	67	53	8	5	5	3	0	1	1	1	144
Tipo de evaluación	Manual	57	37	6	2	3	3	0	1	1	4	114
	Automático	19	19	4	3	2	0	1	2	0	0	50
Validación	Cuestionario	6	6	2	0	0	0	0	0	0	0	14
	Caso estudio	8	3	1	2	1	0	0	0	1	3	19
	Experimento	11	13	2	0	0	0	0	0	0	1	27
	No	51	34	4	3	4	3	1	3	0	0	103
Resultados empíricos sobre efectividad		5	4	1	0	0	0	0	0	0	0	10
Retro-alimentación al diseño	Sí	35	16	2	0	1	0	0	2	0	3	59
	No	41	40	7	5	4	3	1	1	1	1	104

IEEE - IEEEExplore electronic database

ACM - ACM digital library

SD - ScienceDirect digital library

WWW - World Wide Web conference (2003-2008)

ICWE - International Conference on Web Engineering (2003-2008)

IR - Internet Research Journal. Emerald Group

IJH - International Journal of Human-Computer Studies "Web Usability" Special Issue (1997)

IE3IC - IEEE Internet Computing Special Issue on Usability and the Web (2002)

Lib - Un libro de Web Engineering by Springer (2006)

IWWUA - International Workshop on Web Usability and Accessibility (2007-2008)

Los resultados obtenidos por cada criterio de extracción serán comentados en las siguientes secciones (de la Sección 2.1.3.1 a la Sección 2.1.3.5), con respecto a estos resultados, los porcentajes obtenidos en cada criterio se muestran en la Figura 2.3. Por último, la Sección 2.1.3.6 presenta el interés del tema de investigación en los años cubiertos por la revisión.

2.1.3.1 CRITERIO 1: MÉTODOS EMPLEADOS PARA EVALUAR LOS ARTEFACTOS WEB

Los resultados para el criterio 1a (*nuevo método de evaluación o un método existente en el campo IPO*) indican que el 39% de las publicaciones analizadas han presentado métodos de evaluación de la usabilidad diseñados específicamente para la Web. Algunos de estos ejemplos se han encontrado en los trabajos de: Blackmon *et al.* [9], donde se presenta el método “Cognitive Walkthrough for the Web” (CWW) cuyo objetivo es simular la navegación de los usuarios mientras realizan tareas; Mariage *et al.* [54], donde se presenta el método WebMetro, consistente en una recopilación unificada y organizada de guías para la evaluación de usabilidad; o Triacca *et al.* [77] donde se presenta el método MiLE+, un método de inspección para evaluar los diseños Web teniendo en cuenta aspectos tanto independientes como dependientes del análisis.

El 61% restante de los estudios presentan métodos de evaluación existentes en el campo de IPO como “paseos cognitivos”, “evaluaciones heurísticas”, “cuestionarios” o “pruebas a distancia con usuarios”. En principio, estos métodos se definieron para ser aplicado en cualquier tipo de interfaces sin tener en cuenta el tipo de aplicación. Estos resultados podrían indicar que se han aplicado más métodos ya existentes pero adaptados al contexto Web, que métodos de evaluación que han sido elaborados teniendo en cuenta las características específicas de las aplicaciones Web. Tras el análisis de estos estudios, se puede observar que los métodos de evaluación de usabilidad creados para la Web resaltan la importancia del contenido y navegación, no únicamente el comportamiento de los usuarios. Este hecho es de suma importancia, ya que las principales dimensiones que caracterizan a las aplicaciones Web son: contenido, navegación y presentación. Estos argumentos refuerzan la idea de que un método de evaluación de usabilidad deberá abordar cómo los atributos, derivados de las dimensiones anteriores, impactan en la usabilidad de la aplicación Web, es decir, no sólo teniendo en cuenta los atributos de usabilidad relacionados con la eficacia y la eficiencia de las tareas realizadas por el usuario, o la satisfacción del mismo al usar la aplicación.

Los resultados obtenidos en el criterio 1b (*el tipo de métodos empleado*), revelan que el tipo más frecuente de método es el “Testeo con Usuarios”, es decir, el 36% de los artículos revisados reportan algún tipo de método que para ser llevado a cabo necesita la participación de los usuarios finales. Este resultado es un claro indicador de qué la mayoría de las evaluaciones de usabilidad se realizan, principalmente, durante las últimas etapas del ciclo de vida de desarrollo Web. Algunos de estos ejemplos los

podemos encontrar en trabajos como el de Van Waes [80], en el cual se analizaron un par de variantes del método "Thinking Aloud Protocol" (Pensar en voz alta). Como el nombre lo sugiere, este método requiere que los participantes vayan comentando todo lo que piensan mientras realizan una serie de tareas específicas. El estudio muestra cómo la elección de estas tareas, limita el alcance de la evaluación, ya que dependiendo del tipo de tareas, los usuarios se centraron en diferentes tipos de problemas de usabilidad.

Los métodos de Inspección abarcan el 22% de las publicaciones analizadas. Se trata principalmente en métodos basados en evaluaciones heurísticas, revisiones con guías o inspecciones basadas en estándares. El hecho que estos métodos puedan ser llevados a cabo sin necesidad de los usuarios finales permite su empleo en fases más tempranas, como por ejemplo, la fase de diseño donde se define la interfaz de usuario. Encontramos algunos. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Costabile y Matera [19], donde se propone un método de inspección basado en Tareas Abstractas (*Abstract Tasks* - AT). Estas tareas describen las actividades operacionales que los evaluadores deberán realizar durante la inspección. El objetivo de este método consiste en maximizar la reutilización de la experiencia de un evaluador, permitiendo que los evaluadores menos experimentados puedan obtener mejores resultados.

Los métodos de Indagación abarcan el 21% de los trabajos revisados. Dado que estos métodos se centran principalmente en la recopilación de datos subjetivos de los usuarios, la mayoría de estos métodos se utilizan en combinación con otros tipos de métodos como "testeo de usuarios" o "inspección" para realizar una evaluación más completa. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Jung *et al.* [46], donde se utiliza el enfoque de entrevistas en grupo (Focus Group Interview - FGI) para evaluar la experiencia subjetiva en el uso de los servicios de información académica. Este método permite detectar no sólo los problemas de usabilidad, sino además, otros aspectos relacionados con la funcionalidad.

Los métodos de Modelado Analítico abarcan 15% de las publicaciones analizadas. La gran mayoría de ellos están destinados a modelar algunos aspectos tales como interfaces de usuario, entornos de trabajo, o el rendimiento del usuario a fin de predecir la usabilidad. Uno de los más representativos de este tipo de métodos es análisis es el análisis de GOMS (Goals, Operators, Methods, Selection Rules). GOMS es un método para el modelado y la descripción de realización de las tareas de los usuarios. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Tonn-Eichstadt [76] donde se aplica una versión modificada de GOMS para medir los aspectos de la usabilidad Web para usuarios ciegos. El objetivo era encontrar la mejor alternativa de un conjunto de diseños desarrollados en la fase de diseño del proceso de desarrollo Web.

Los métodos de simulación sólo abarcan el 6% de las publicaciones analizadas. Apenas existen métodos que pueden considerarse exclusivamente de simulación, puesto que la mayoría presenta características de otros tipos (sobre todo de modelado analítico). Están basados principalmente en agentes o algoritmos que pretenden simular el comportamiento del usuario. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Chi *et al.* [21] donde se explica un procedimiento que mide la navegabilidad de un sitio mediante el cálculo de la probabilidad con la que los usuarios llegan a los destinos deseados. El simulador “Bloodhound InfoScen” fue desarrollado para apoyar a este método con el objetivo de generar automáticamente informes de usabilidad.

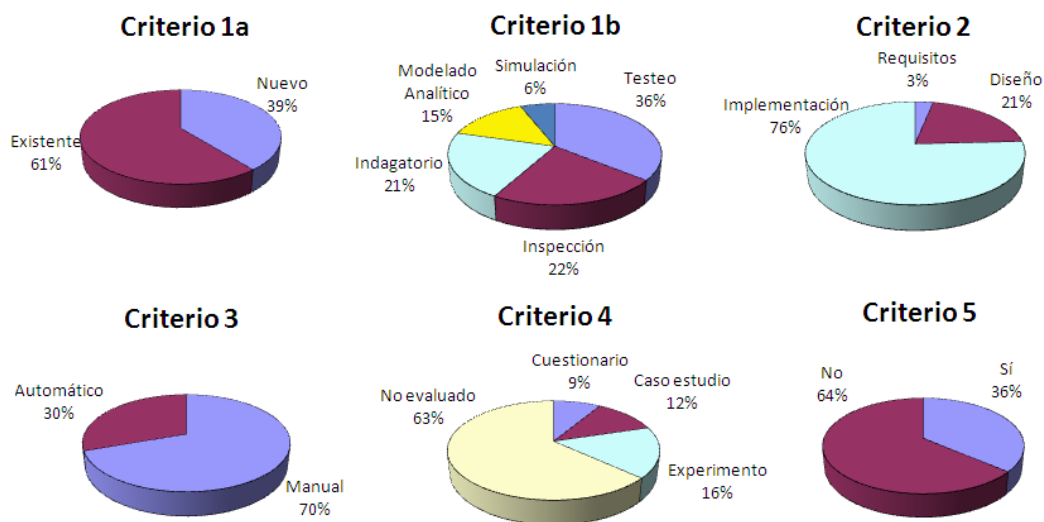


Fig 2.3. Porcentajes resultantes en cada criterio de extracción.

2.1.3.2. CRITERIO 2: FASES EN LA QUE SE APLICA EL MÉTODO DE EVALUACIÓN

El análisis de los resultados confirma que las evaluaciones se llevan a cabo principalmente (en torno al 76%) en la fase de implementación de la aplicación Web. Este tipo de evaluaciones de la usabilidad, también conocidas como evaluaciones sumativas, tienen lugar después de que el producto haya sido desarrollado, o cuando existe un prototipo ejecutable. De esta forma, los artefactos más comúnmente analizados resultan ser las interfaces de usuario final y los ficheros logs que registran las acciones del usuario. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Nakamichi et al. [62], donde se presenta la herramienta WebTracer destinada a la grabación y análisis de las operaciones del usuario al mismo tiempo que interactúa con el sitio Web, con el objetivo de recopilar datos cuantitativos que permitan detectar posibles problemas de usabilidad sin interrumpir la interacción del usuario.

En el 21% de los artículos analizados se describen evaluaciones realizadas a nivel de diseño, empleando para ello artefactos intermedios de la aplicación Web, como por ejemplo, bocetos de interfaces de usuario y modelos de navegación. Este tipo de evaluaciones de la usabilidad son también conocidas como evaluación formativa, ya que tienen lugar durante el proceso de desarrollo. Algunos de estos ejemplos los podemos encontrar en trabajos como el de Atterer y Schmidt [5], donde se propuso un prototipo de validador de usabilidad basado en modelos. El objetivo es realizar un análisis de las interfaces de usuario que se representan mediante modelos. Este enfoque pretende aprovechar los modelos de navegación y modelos de presentación que están disponibles en los actuales métodos de desarrollo dirigido por modelos pertenecientes a la Ingeniería de la Web (OO-H , UWE), ya que contienen información precisa acerca de cómo se navega por la información y cómo ésta se estructura para ser presentada al usuario.

Sólo en el 3% de los estudios se describen evaluaciones realizadas a nivel de especificación de requisitos. Estos trabajos están más orientados a validar los requisitos no-funcionales relacionados con la usabilidad, para facilitar después el cumplimiento de esos requisitos mediante técnicas de inspección. Algunos de estos pocos ejemplos los podemos encontrar en trabajos como el de Bolchini et al. [12] donde se presenta el marco AWARE (Analysis of Web Applications REquirements). Esta metodología se presenta una propuesta novedosa para analizar, organizar y documentar los requisitos haciendo uso de técnicas orientadas a metas y técnicas basadas en escenarios, presentado un entorno adecuado para la aplicación de métodos de inspección que permitan evaluar los objetivos de los usuarios.

Los resultados obtenidos indican que hay carencias en cuanto a métodos de evaluación de usabilidad que puedan ser utilizados en las fases tempranas del ciclo de vida de desarrollo Web. Aunque las evaluaciones en la fase de implementación son muy necesarias para analizar el comportamiento de los usuarios, puesto que hay aspectos de usabilidad que sólo se puede evaluar a través de la interacción del usuario, empleando métodos de evaluación únicamente en esta etapa puede conllevar varias dificultades al detectar problemas de usabilidad más tarde. La corrección de estos problemas puede contribuir a un mayor coste en el mantenimiento del código fuente. De esta forma, las evaluaciones de usabilidad se deberían realizar no sólo en la fase de implementación, sino también en cada fase del desarrollo de aplicaciones Web. Si los problemas de usabilidad pueden detectarse antes, se puede predecir y corregir la usabilidad de las aplicaciones Web finales, con el consiguiente ahorro de recursos, contribuyendo a reducir el costo del proceso de desarrollo Web.

2.1.3.3. CRITERIO 3: EL TIPO DE EVALUACIÓN REALIZADA POR LA UEM

En el 70% de los estudios revisados se contemplan evaluaciones de forma manual mientras que sólo el 30% de los estudios informó acerca de la existencia de algún tipo de herramienta automatizada para dar soporte al método propuesto. La mayoría de estas herramientas se basan principalmente en la comprobación de código fuente, el análisis de ficheros log, y la simulación del comportamiento del usuario. Algunos de estos ejemplos los podemos encontrar en trabajos como los de: Becker y Berkemeyer [7], donde se propone la técnica RAD-T, para dar soporte al desarrollo de aplicaciones Web más usables integrando requisitos de usabilidad haciendo uso de una extensión de HTML; Vanderdonckt et al. [81], donde se propone un método automático de revisión de guías definidas formalmente mediante XML; e Ivory y Megraw [45], donde se propone la metodología WebTango, consistente en un conjunto de medidas cuantitativas calculadas automáticamente en una amplia muestra de interfaces de usuario Web, con el objetivo de comparar las medidas obtenidas de nuevas interfaces de usuario con las medidas existentes de otras interfaces consideradas como referentes de usabilidad.

Estos resultados indican que la mayoría de los esfuerzos en los métodos automatizados para la evaluación de usabilidad se centran en el código fuente de la aplicación Web, ya que es el único artefacto empleado en la mayoría de los casos. Existe una carencia de métodos automatizados capaces de evaluar artefactos intermedios, tales como modelos abstractos de interfaces de usuario o modelos de navegación. La mayoría de las herramientas encontradas, se basan en operacionalizar guías de usabilidad existentes y en el cálculo e interpretación de medidas de usabilidad en las interfaces de usuario finales. Sin embargo, es importante señalar que en este tipo de evaluación presenta varias limitaciones, ya que estos métodos están orientados a recopilar datos de carácter objetivo, por lo que la percepción del usuario y el contexto de usuario no pueden ser consideradas. Pese a que un método automatizado es capaz de reducir esfuerzos y recursos, es aconsejable combinar estos métodos con otros que permitan considerar otras dimensiones de la usabilidad, como la satisfacción y percepción del usuario.

2.1.3.4. CRITERIO 4: VALIDACIÓN DEL MÉTODO

Los resultados obtenidos para el criterio 4a (*tipo de validación llevada a cabo*), reveló que el 63% de los estudios no aplicó ninguna técnica para validar el método de evaluación. El 9% de los artículos revisados presentan una validación del método realizada a través de técnicas basadas en cuestionarios. Algunos de estos ejemplos los podemos encontrar en trabajos como los de Zaharias [83], donde se propone un cuestionario para la evaluación de aplicaciones Web basadas en e-learning, cuya

cobertura fue validada realizando dos pruebas piloto donde el cuestionario se aplicó en un caso real.

En el 12% de los artículos revisados el método de evaluación se validó aplicando un caso de estudio. Algunos de estos ejemplos los podemos encontrar en trabajos como los de Matera et al. [55] donde se presenta un caso de estudio en el que tres métodos fueron aplicados a la evaluación de una aplicación Web: inspecciones para examinar la especificación de hipertexto, análisis del uso de la Web para analizar el comportamiento de los usuarios, y evaluación heurística para analizar los prototipos y la aplicación Web final. El estudio de caso se llevó a cabo en un proceso de desarrollo iterativo, en el que versiones de la aplicación Web han ido produciéndose, evaluándose y mejorándose teniendo en cuenta los problemas detectados por las evaluaciones anteriores.

Los experimentos controlados están presentes en el 16% de los artículos revisados. Algunos de estos ejemplos los podemos encontrar en trabajos como los de Bolchini y Garzotto [11], donde se realizó un estudio empírico para evaluar la calidad del método MiLE+. Los atributos evaluados del método fueron el grado de detección de problemas de usabilidad (rendimiento), la rapidez de esta detección (eficiencia), recursos necesarios para llevar a cabo una evaluación (coste-efectividad) y la facilidad de aprendizaje del método. Los resultados muestran cómo MiLE+ alcanza niveles aceptables en los atributos anteriores. Sin embargo, este experimento sólo se llevó a cabo con los expertos y los usuarios novatos, las medidas no fueron comparadas con las de otros métodos.

Estos resultados evidencian que el método más común de validación es el experimento controlado. Esto se explica debido a que la mayoría de estos estudios han sido publicados en revistas y conferencias del campo IPO, donde la experimentación es un método de investigación muy común en este ámbito. Sin embargo, los porcentajes también revelan que existe una carencia de validaciones de métodos de evaluación.

Los resultados obtenidos del criterio de 4b (*efectividad de los métodos*) se han extraído del análisis de los artículos que han sido clasificados como experimentos. Con respecto al criterio 4b-i (*medidas que se han empleado para evaluar la efectividad del método*), los resultados muestra que la medida más utilizada es el número de errores de usabilidad detectados, en ocasiones definido como "rigor", siendo éste el cociente entre el número de errores de usabilidad detectados y el total de errores existentes (número conocido previamente). Sin embargo, se han encontrado otras medidas no tan frecuentes tales como: calidad y tipo de problemas detectados, el número y la calidad de los comentarios elicitados acerca de los errores que se producen, o la dificultad de las tareas que se llevan a cabo en el método. Esto pone de manifiesto que no existe un consenso sobre cómo se mide la efectividad de un método de evaluación de usabilidad.

Esta evidencia se puede considerar como un inconveniente considerable a la hora de realizar comparaciones entre distintos métodos.

Con respecto al criterio 4b-ii (*tipo de artefactos que fueron evaluados en el experimento*), es importante observar que todos los experimentos se realizan empleando interfaces de usuario ya implementadas o partes de la aplicación Web final. No existen experimentos donde los métodos involucrados utilicen los artefactos intermedios. Esta carencia detectada podría ser una buena motivación para futuras investigaciones.

Por último, y en relación con el criterio 4b-iii (comparaciones entre los métodos), sólo 10 artículos realizaron comparaciones con otros métodos. Esto viene a ser, el 37% de todos los experimentos, y sólo el 6% de todos los artículos incluidos en la revisión. Algunos de estos ejemplos los podemos encontrar en trabajos como los de Hornbæk y Frøkjær [32], donde se compararon dos técnicas de inspección: Cognitive Walkthrough (CW), y Metaphor of Human Thinking (MOT). El objetivo del experimento fue comparar cómo 20 participantes podían evaluar y re-diseñar sitios Web de comercio electrónico. Los datos generados por el experimento incluyen listas de problemas de usabilidad, re-diseños, diarios, y preferencias de los participantes. Los resultados del experimento muestran que los participantes identificaron más problemas de usabilidad (un 30% más) cuando usaron MOT. La tabla 2.2 muestra los resultados obtenidos, en este conjunto de artículos, tras aplicar el método de agregación presentado en la Sección 2.1.1.5.

Tabla 2.2. Datos empíricos extraídos de los experimentos

Evidencia	Métodos de Evaluación de Usabilidad	Medida Efectividad	Efecto
0011	M1= Web Design Perspectives [18] M2= Abstract Tasks [19] M3= Gerhardt-Powals Principles [30] M4= Nielsen Heuristics HEV [30] M5= End-user surveys [73]	E1=Núm. de problemas de usabilidad detectados	$(M1, M2) > (M3, M4) > M5$
0012	M6= Think-Aloud Protocol [32] M7= Metaphor of Human-Thinking [33] M8= Cognitive Walkthrough for the Web [10] M9= Cognitive Walkthrough [10]	E1= Núm. de problemas de usabilidad detectados	$(M6, M7, M8) > M9$
0006	M6= Think-Aloud Protocol [32] M10= Entrevistas [79] M11= Cuestionarios [79]	E2=Comentarios relevantes elicitados	$M6 > (M10, M11)$
0007	M12 = Thinking-Aloud Protocol E&S [48] M13 = Thinking-Aloud Protocol B&R [48]	E3= Núm de intervenciones, etc.	$M12 \approx M13$
0009	M14= AWA algorithm [51] M15= User questionnaires [51]	E4= Tasa de error por tarea, etc.	$M14 \approx M15$

Los resultados de la tabla 2 nos muestran los métodos que resultaron más efectivos al ser comparados con otros fueron: Web Design Perspectives [18], Abstract Tasks [19],

Think-Aloud Protocol [79], Metaphor of Human-Thinking [32], y Cognitive Walkthrough for the Web [10]. Sin embargo, este resultado debería interpretarse como posibles métodos que resultan entre los mejores candidatos para su empleo en evaluaciones de usabilidad Web, puesto que debido a la escasez de experimentos, no es posible proporcionar una evidencia empírica más fuerte.

2.1.3.5. CRITERIO 5: RETROALIMENTACIÓN A LA FASE DE DISEÑO

Los resultados para el criterio 5 (*el método proporciona retroalimentación a la fase de diseño*), revelan que el 63% de los estudios únicamente proveen una lista de errores detectados o una cuantificación del grado de usabilidad de la aplicación Web, pero sin dar una clara retroalimentación sobre cómo debería ser el re-diseño de los artefactos correspondientes. Los artículos restantes (el 37%) presentan métodos que sugieren cambios de diseño basándose en los problemas de usabilidad detectados. Algunos de estos ejemplos los podemos encontrar en trabajos como los de Chi [20], donde se presenta un método de visualización basado en la minería de datos, con el objetivo de aplicar un conjunto de técnicas para identificar problemas específicos de usabilidad en sitios Web, descubrir los patrones de tráfico más importantes, y proponer cambios para mejorar el modo de usuario para acceder a la Web; Blackmon et al. [9], donde se aplica el método Cognitive Walkthrough for the Web (CWW), incorporando una variante del algoritmo LSA (Latent Semantic analysis), para identificar aquellos enlaces potencialmente confusos entre sí, y poder corregirlos; y Hornbæk y Frøkjær [33], donde se realiza un experimento que pone de manifiesto cómo las propuestas de re-diseño son de mayor utilidad que las propias descripciones de problemas de usabilidad obtenidas al aplicar los métodos de evaluación.

Estos resultados indican que la mayoría de los métodos de evaluación de usabilidad han sido diseñados principalmente para generar una lista de problemas de usabilidad, pero no para proporcionar una orientación más explícita acerca de cómo estos problemas podrían ser corregidos. Las evaluaciones de la usabilidad deben tener en cuenta tanto “descubrir” como “reparar” problemas de usabilidad. Empleando únicamente listas de problemas de usabilidad, los desarrolladores necesitan más recursos para explorar nuevas alternativas para sus diseños. Estos resultados evidencian la necesidad de integrar en los métodos de evaluación, más detalles acerca de los problemas de usabilidad y ayuda en cuestiones de rediseño, que permita localizar, entender y reparar el error de la forma más rápida posible. Para lograr este objetivo, los métodos de evaluación de usabilidad deben estar más integrados en el proceso de desarrollo Web a fin de comprender la trazabilidad entre los problemas de usabilidad detectados y los artefactos que se originan esos problemas de usabilidad.

2.1.3.6. INTERÉS DEL TEMA DE INVESTIGACIÓN

La evaluación de la usabilidad Web ha motivado la aparición de una gran cantidad de estudios en los últimos años. Estos estudios pueden encontrarse en artículos provenientes de las disciplinas: Interacción Persona-Ordenador (IPO) e Ingeniería Web. Todos los estudios coinciden en resaltar cómo las evaluaciones de usabilidad en el dominio Web son cruciales para el éxito de la aplicación. Sin embargo, la mayoría de los estudios encontrados se centran en la presentación de los resultados obtenidos tras la evaluación de una aplicación Web concreta. Son relativamente pocos los estudios cuyo objetivo principal es la definición y validación de un método de evaluación de usabilidad.

La Figura 2.4 muestra el número de publicaciones seleccionadas sobre los métodos de evaluación de usabilidad Web clasificadas por año y por fuente bibliográfica. Estos datos nos muestran que existe un interés creciente sobre la evaluación de usabilidad Web, especialmente desde 2004, indicando que la usabilidad es una característica de calidad de las aplicaciones Web que ha de considerarse como un factor crucial en el desarrollo de aplicaciones Web.

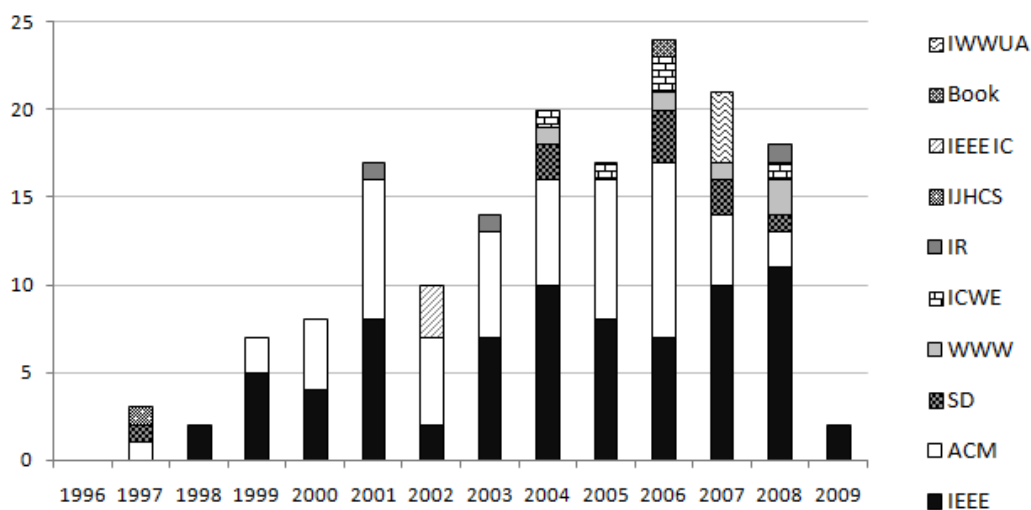


Fig 2.4. Número de publicaciones por fuente en los últimos años.

No se pueden extraer conclusiones acerca de las mejores fuentes bibliográficas, ya que estos datos no reflejan exactamente el número de documentos obtenidos en cada fuente. Esto es debido al hecho de considerar una única vez a aquellos artículos que han aparecido en varias fuentes. Sin embargo, la mayoría de los artículos relevantes en el estudio se han obtenido principalmente de las bibliotecas digitales: IEEEExplore y ACM. Es importante resaltar que en el año 2009 aparecen menos artículos debido a que la revisión sistemática se llevó a inicios de 2009.

2.1.4. DISCUSIÓN

En esta sección se recopilan los principales hallazgos de esta revisión sistemática, pone de manifiesto las limitaciones que podrían representar amenazas a la validez revisión sistemática, y por último se analizan las implicaciones de los resultados tanto para el ámbito académico como para profesionales de la industria.

2.1.4.1. PRINCIPALES HALLAZGOS

El objetivo de esta revisión fue examinar cómo se emplean actualmente los métodos de evaluación de usabilidad en el desarrollo Web. Los principales hallazgos de esta revisión son:

- Los métodos de evaluación de usabilidad han ido constantemente adaptándose para lograr mejores resultados en el dominio Web. Sin embargo, los métodos evalúan diferentes aspectos del concepto usabilidad, dependiendo de la definición de usabilidad de la que partan (ISO 9241-11, ISO 9126-1, etc).
- En la mayoría de los artículos, las evaluaciones tienen lugar en la fase de implementación (es decir, sobre interfaces de usuario final, análisis de archivos log, etc.). Además, es muy común aplicar evaluaciones solo en una única fase de desarrollo.
- Existe una escasez de métodos automatizados, especialmente aplicables en las primeras etapas (por ejemplo, especificaciones de requisitos, modelos de navegación, modelos de presentación).
- La gran mayoría de los artículos no presentan ningún tipo de validación acerca de los métodos que emplean. Sin embargo, la realización de experimentos es uno de los métodos de validación más empleados cuyos resultados proporcionan datos empíricos muy relevantes.
- Teniendo en cuenta las comparaciones métodos que se realizaron en los experimentos encontrados, los métodos que resultaron ser más eficaces fueron: Web Design Perspectives [18], Abstract Tasks [19], Think-Aloud Protocol [79], Metaphor of Human-Thinking [32], y Cognitive Walkthrough for the Web [10].
- La mayoría de los métodos sólo proporcionan una lista de problemas de usabilidad, no suelen proporcionar información explícita y sugerencias para ayudar a los diseñadores a mejorar los artefactos de la fase de diseño.
- En la actualidad, la evaluación de la usabilidad Web es un tema de gran relevancia e interés creciente donde varias disciplinas están implicadas.

2.1.4.2. LIMITACIONES DE LA REVISIÓN SISTEMÁTICA

Las principales limitaciones que presenta esta revisión sistemática son: el sesgo de publicación, el sesgo de selección, la inexactitud en la extracción de datos, y los errores

de clasificación [47]. El sesgo de publicación se refiere a que los resultados positivos tienen más probabilidades de ser publicados que los resultados negativos, ya que estos resultados negativos tardan más en ser publicados o son menos citados en otras publicaciones. Para mitigar esta amenaza, en medida de lo posible, se revisaron las actas de conferencias y revistas relevantes en el área. Sin embargo, no se ha considerado la literatura gris (tesis e informes técnicos), artículos de carácter industrial, y los trabajos en proceso. Este hecho podría afectar a la validez de los resultados obtenidos.

El sesgo de selección se refiere a la distorsión del análisis estadístico realizado basándonos en los criterios de selección de los artículos. Para mitigar esta amenaza, en medida de lo posible, se han elegido fuentes bibliográficas donde los artículos que tratan la usabilidad Web suelen ser publicados normalmente. Sin embargo, algunos artículos procedentes del área de la ingeniería Web quedaron excluidos al no tener acceso a sus fuentes bibliográficas. Este hecho podría afectar a la validez de nuestros resultados.

La inexactitud en la extracción de datos y los errores de clasificación se refieren a la variación introducida por el revisor a la hora de llevar a cabo dichas actividades. Para mitigar esta amenaza, al menos en cierta medida, se realizó la extracción y clasificación de los documentos con tres revisores. Todos los estudios fueron clasificados por cada revisor, resolviéndose las discrepancias por consenso.

2.1.4.3. IMPLICACIONES PARA INVESTIGADORES Y PROFESIONALES

Los resultados obtenidos de la revisión sistemática tienen implicaciones tanto para los investigadores que están planeando nuevos estudios de evaluación de la usabilidad de aplicaciones Web, como para los profesionales que trabajan en el desarrollo Web y necesitan integrar evaluaciones de usabilidad de la manera más eficaz posible en su proceso de desarrollo.

Con respecto al ámbito de la investigación, el principal problema viene determinado por el hecho que el concepto “usabilidad” no presenta una definición homogénea. Esto acarrea problemas, ya que las evaluaciones no abarcan realmente todos los aspectos de usabilidad, a pesar de que un método sea eficaz detectando problemas. Por este motivo, los métodos de evaluación de usabilidad deberían tener en cuenta todas las definiciones de usabilidad que se recogen en los estándares existentes y centrarse en las características específicas de aplicaciones Web con el fin de ser más completo.

Realizar evaluaciones de usabilidad en cada fase del desarrollo es crítico para asegurar que el producto será realmente aceptado por los usuarios. Dado que la revisión sistemática detectó que la mayoría de los artículos presentan evaluaciones en la fase de implementación o en una única fase de desarrollo. Se hace evidente la escasez de métodos para hacer frente a la usabilidad en las primeras etapas de desarrollo de

aplicaciones Web, no sólo cuando la aplicación está parcialmente desarrollada. El principal problema parece ser que la mayoría de los procesos de desarrollo Web no aprovechan los artefactos producidos en las fases tempranas del desarrollo. Estos artefactos intermedios son empleados, principalmente, para orientar a los programadores y para documentar la aplicación Web. De esta forma, la trazabilidad entre los artefactos y la aplicación Web finales no está claramente definida, por lo que las evaluaciones de estos artefactos podrían resultar no ser tan eficaces. Las investigaciones futuras deberían orientarse hacia la integración de evaluaciones de usabilidad en los procesos de desarrollo Web cuyos artefactos intermedios puedan ser evaluados de manera efectiva. Este problema no ocurre en el desarrollo de software dirigido por modelos (DSDM) donde los modelos (artefactos intermedios) que especifican una aplicación Web completa se aplican en todas las etapas del proceso de desarrollo, y el código fuente final se genera automáticamente a partir de estos modelos. La evaluación de estos modelos proporcionaría informes de evaluación de la usabilidad que presentan cambios en dichos modelos, los cuales se podrían reflejar directamente en el código fuente, sin necesidad de mantener dicho código. Por otra parte, este hecho se refleja también en la mayoría de los métodos automatizados, donde también se realizan las evaluaciones de los artefactos pertenecientes a la fase de implementación (ej. código HTML). La investigación acerca de los métodos automatizados debe ir más allá. Deberían centrarse en la evaluación de los artefactos intermedios que se emplean en las primeras etapas, como por ejemplo, los requisitos de las especificaciones, los modelos de navegación, los modelos de presentación.

Uno de los hallazgos detectados era que la mayoría de los métodos proporcionaban una lista de problemas de usabilidad, pero no ofrecían guías y sugerencias para ayudar a los diseñadores acerca de cómo estos problemas pueden ser corregidos. Este hecho sugiere que las nuevas propuestas necesitan abordar los problemas de usabilidad como una parte integral del proceso de desarrollo Web, mejorando la descripción de los problemas de usabilidad con el fin de adoptar fácilmente los cambios pertinentes.

Por último, se detectaron muy pocas validaciones de métodos en la literatura existente. Cuando se presenta un método, es interesante llevar a cabo experimentos que proporcionen información empírica sobre cómo se emplea el método (por ejemplo, la facilidad de uso, eficacia, eficiencia, aplicación de coste). En esta área, se requiere más experimentos para la comparación de métodos entre sí, donde se empleen las mismas variables a medir, a fin de mostrar cuáles son los métodos más apropiados que otros.

Con respecto al ámbito profesional, esta revisión presenta algunas lecciones aprendidas que pueden ser de utilidad al sector de desarrollo Web. Estas lecciones son acerca del tipo de métodos de evaluación de usabilidad que se pueden aplicar en las

diferentes fase del proceso de desarrollo Web, cómo se pueden combinar, y de qué métodos hay constancia que han sido más efectivos respecto de otros.

Debido a la carencia de métodos aplicados en la fase de requisitos, sólo es posible extraer conclusiones sobre las fases de diseño e implementación. Los tipos de métodos más aplicados en la fase de diseño fueron los de “Inspección” y “Modelado analítico”. Estos métodos se centran, principalmente, en la evaluación de interfaces de usuario abstractas. Ambos tipos tienen como base la evaluación heurística y guías de revisión que no requieren la participación del usuario final. Este hecho hace que los métodos sean útiles para que los mismos desarrolladores Web realicen evaluaciones. Sin embargo, en la mayoría de los casos, estos métodos requieren evaluadores expertos dependiendo de la complejidad del método y de cómo esté documentado. Por otra parte, los tipos de métodos más aplicados en la fase de implementación son el “testeo con usuarios” y los “métodos indagatorios”. Estos métodos se centran principalmente en la evaluación de la aplicación Web final o en ficheros log que muestran datos acerca del uso de la aplicación por parte de los usuarios. Ambos tipos requieren la participación de los usuarios reales y su planificación requiere más recursos que los “métodos de inspección” o “modelado analítico”.

Los profesionales deben tener en mente que no existe un método de evaluación de usabilidad infalible para abordar todos los problemas de usabilidad existentes. Por esta razón, muchos de los artículos emplearon más de un método aprovechando los recursos destinados a llevar a cabo el plan de evaluación. Por ejemplo, en la mayoría de los casos en los que se aplicó un método de “testeo con usuario” (Thinking Aloud Protocol, testeo remoto, etc), a menudo se combinaron con otros métodos indagatorios (Cuestionarios, Focus Group, entrevistas, etc) reutilizando recurso como la participación de los usuarios.

Por último, las tareas importantes dirigidas a los profesionales no se basan sólo en comparar los resultados de métodos diferentes, sino también recoger datos sobre cómo se aplica el método. Esta información puede ser muy útil para detectar deficiencias que permitan re-definir los métodos de evaluación de usabilidad para que estos sean más eficaces y más fáciles de aplicar. Como dato de interés, los profesionales podrían tomar de referencia los métodos que obtuvieron mejores resultados en base a su efectividad (ver Sección 2.1.3.4.)

2.2. ESTÁNDARES RELACIONADOS CON LA EVALUACIÓN DE LA USABILIDAD

La organización internacional para la estandarización (ISO) ha desarrollado una gran variedad de modelos para especificar y medir la usabilidad del software, entre muchas otras características de calidad. El empleo de estándares ofrece algunas ventajas, entre ellas se destacaría el hecho que los métodos de evaluación de usabilidad basados en ellos presentan homogeneidad en cuanto a definiciones de conceptos, ya que estos

conceptos han sido consensuados entre diferentes grupos que participan en la elaboración del estándar. Además, proveen una base de gran utilidad para realizar inspecciones. Debido a esto, a continuación se presentarán y analizarán los estándares que guardan relación con la evaluación de usabilidad. Para ello se ha categorizado los estándares en dos grupos: orientados al proceso (ISO 9241 e ISO 13407) y orientados al producto (ISO 9126 e ISO 14598). Por último se presentará la nueva serie de estándares que pretenden mejorar y unificar los anteriores (ISO 25000 - SQuaRE).

2.2.1. ESTÁNDARES ORIENTADOS AL PROCESO: ISO 9241 E ISO 13407

El estándar ISO 9241 es un conjunto de normas internacionales acerca los requisitos ergonómicos y recomendaciones, para oficinas con terminales visuales, relativas al hardware, al software, y los atributos del entorno, los cuales contribuyen a un nivel de usabilidad adecuado con respecto a unos principios ergonómicos subyacentes. El estándar se divide en 17 partes. Las partes 1 y 2 muestran una introducción y guías para el empleo del estándar. En las partes 3-9 se tratan los requisitos y guías relacionadas con el hardware que tienen su impacto en el funcionamiento del software. Por último, las partes 10-17 se centran en los aspectos del software.

En este estándar, la parte 11 “Medidas de especificación de la usabilidad” define la usabilidad como: “La medida en la que un producto se puede usar por determinados usuarios para sus conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado”. Por lo tanto, según este estándar, la medición de la usabilidad del sistema consta de tres atributos de la usabilidad evaluados de acuerdo con el contexto de la utilización del software:

- Efectividad: ¿Los usuarios alcanzan sus objetivos correctamente al usar el sistema?
- Eficiencia: ¿Qué recursos se han requerido a fin de alcanzar los objetivos del usuario?
- Satisfacción: ¿Qué impresiones tienen los usuarios sobre el uso del sistema?

Las características del contexto de uso (usuarios, tareas y entorno) pueden ser suficientes para determinar la usabilidad como una característica del producto: un cambio en cualquier aspecto relevante del contexto de uso puede afectar a la usabilidad del producto. Por ejemplo, un producto que podría resultar usable por usuarios expertos, podría no serlo por usuarios principiantes. Es más, aspectos del entorno de trabajo, tales como el ruido o la distribución espacial del lugar de trabajo, también podrían afectar a la usabilidad.

La ISO 9241-11 [37] recomienda un enfoque basado en procesos para evaluar la usabilidad, mediante el cual, un sistema se obtiene a través de un Diseño Centrado en el

Usuario. Por esto motivo, esta norma debe aplicarse conjuntamente con el estándar ISO 13407.

La ISO 13407 [40] proporciona guías acerca sobre las actividades involucradas en el ciclo de vida perteneciente al Diseño Centrado en el Usuario. En ella se describe el Diseño Centrado en el Usuario como una actividad multidisciplinar, que incorpora factores humanos y conocimientos ergonómicos con el objetivo de mejorar la efectividad y eficiencia, las condiciones de trabajo, y contrarrestar los posibles efectos adversos del uso que guardan relación con la salud, la seguridad y el rendimiento. La Figura 2.5 muestra las actividades que se llevan a cabo en un Diseño Centrado en el Usuario.

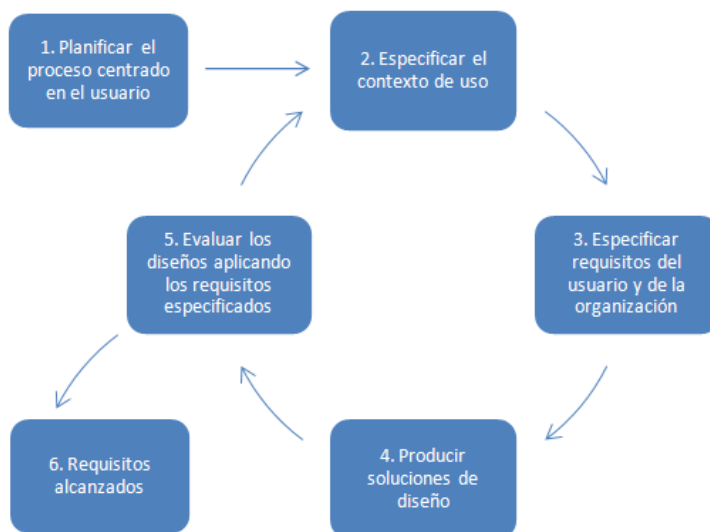


Fig 2.5. Actividades propuestas por la ISO 13405 para el Diseño Centrado en el Usuario.

Las ventajas [4] que conlleva emplear la definición de usabilidad de este conjunto de normas son:

- Se identifican aspectos de usabilidad y componentes del contexto de uso que deben tenerse en cuenta en las fases de especificación de requisitos, diseño y evaluación de usabilidad.
- El rendimiento y la satisfacción del usuario proporcionan medidas directas de usabilidad en un contexto específico.
- Las medidas obtenidas proporcionan una base para comparar la usabilidad de varios diseños en un mismo contexto.

Sin embargo, algunos de los puntos débiles de este conjunto de normas serían:

- La usabilidad está estrictamente tratada desde una perspectiva de proceso, no centrada en el producto.

- No tiene en cuenta la característica de “facilidad de aprendizaje”, siendo esta recomendada por otros estándares y expertos en usabilidad.
- No tiene en cuenta aspectos de seguridad, considerados importantes por expertos del dominio.

2.2.2. ESTÁNDARES ORIENTADOS AL PRODUCTO: ISO 9126 E ISO 14598

El estándar ISO 9126 es un conjunto de normas internacionales acerca de la calidad del software partiendo de la perspectiva del producto. Este conjunto presenta uno de los modelos de calidad más extensos desarrollados hasta la fecha, basado principalmente en el modelo de McCall [56], uno de los primeros modelos de calidad existentes. Una primera versión del modelo de calidad se publicó inicialmente en 1991, y posteriormente fue mejorado en los siguientes diez años (versión de 2001). Este modelo plantea seis categorías generales de características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

El objetivo de la ISO 9126 es proporcionar un marco para la evaluación de la calidad del software. Este estándar no establece requisitos específicos de calidad del software, su base es el modelo de calidad definido, el cual puede ser aplicado a todo tipo de software. La versión de 2001 ya incluye la perspectiva del usuario e introduce el concepto de “calidad en uso” como la capacidad del producto software para permitir a los usuarios alcanzar sus objetivos específicos con efectividad, productividad, satisfacción y seguridad. Estas sub-características ofrecen un acercamiento a la definición de usabilidad planteada en la ISO 9241-11.

De esta forma, la ISO 9126 se divide en cuatro partes.

- ISO/IEC 9126-1: Esta parte presenta el Modelo de Calidad en base a dos vistas que marcan el ciclo de vida de la calidad (ver Fig. 2.6):
 - Calidad interna y externa, las cuales se modelan con el mismo conjunto de características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.
 - Calidad en uso modelada con otras cuatro características: eficacia, productividad, seguridad y satisfacción.
- ISO/IEC 9126-2: Esta parte describe las métricas externas que se pueden utilizar para especificar o evaluar el comportamiento del software cuando es operado por el usuario.
- ISO/IEC 9126-3: Esta parte describe las métricas internas que se pueden utilizar para crear describir propiedades estáticas de la interfaz, que puede ser evaluadas por la inspección sin poner en funcionamiento el software.

- ISO/IEC 9126-4: Esta parte describe las métricas de calidad en uso que se pueden utilizar para especificar o evaluar el efecto del producto software cuando son operados por el usuario en determinados contextos de uso.

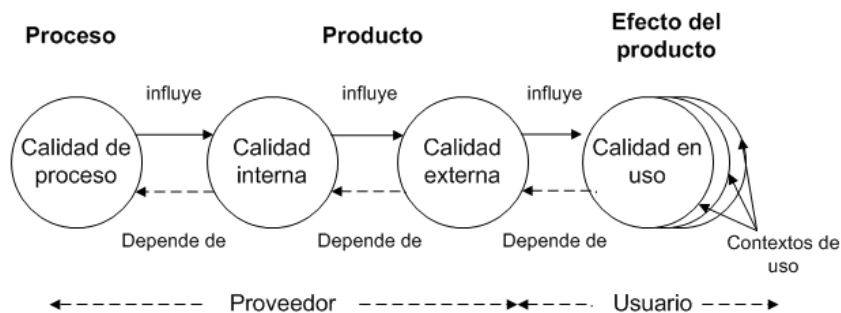


Fig 2.6. Ciclo de vida de la calidad en la ISO 9126.

En la versión de 1991 de la ISO 9126, el concepto de usabilidad se definió como *"el conjunto de atributos que influyen en el esfuerzo necesario para su utilización y en la evaluación individual de tal uso, por un conjunto de usuarios"*. De esta forma, el concepto de usabilidad queda más orientado al producto. Esto provocó que la usabilidad comenzara a verse como un factor independiente de la calidad del software, centrándose únicamente en los atributos de software, tales como su interfaz, que hacen que sea fácil de usar. Sin embargo, los atributos requeridos por un producto para alcanzar cierto grado de usabilidad dependerán también de la naturaleza del usuario, las tareas a realizar y el entorno. Por ese motivo, en la versión de 2001 encontramos una definición de usabilidad más precisa: *"La capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas"*. Aplicando esta definición, la usabilidad se descompone en las siguientes sub-características:

- *Facilidad de aprendizaje:* capacidad del producto de ser aprendido en su manejo.
- *Facilidad de entendimiento:* capacidad del producto de ser entendida por nuevos usuarios en términos de su propósito y cómo puede ser usado en tareas específicas.
- *Operabilidad:* capacidad del producto de ser operado y controlado en cualquier momento.
- *Grado de atracción:* capacidad del producto de ser atractivo para sus usuarios.
- *Conformidad:* capacidad del producto para adherirse a estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

De esta forma, un producto usable puede ser diseñado mediante la incorporación de características del producto y atributos, que son beneficiosos para los usuarios en determinados contextos de uso. Estos usuarios suelen ser los usuarios finales que

interactúan directamente con el sistema, pudiendo incluirse además, usuarios indirectos que son influenciados por o dependen de la utilización del software.

Puesto que la ISO 9126 se limita a la especificación de un modelo de calidad general, ésta se debe aplicar en conjunto con la ISO 14598. Éste estándar proporciona un marco de trabajo para evaluar la calidad de todo tipo de producto Software e indica los requisitos para los métodos de medición y el proceso de evaluación. De esta forma, la ISO/IEC 14598 consta de seis partes:

- ISO/IEC 14598-1: Esta parte provee una visión general de las otras cinco partes y explica la relación entre la evaluación del producto software y el modelo de calidad definido en la ISO/IEC 9126 (ver Fig. 2.7).
- ISO/IEC 14598-2: Esta parte contiene requisitos y guías para las funciones de soporte tales como la planificación y gestión de la evaluación del producto del software.
- ISO/IEC 14598-3: Esta parte provee los requisitos y guías para la evaluación del producto software cuando la evaluación es llevada a cabo en paralelo con el desarrollo por parte del desarrollador.
- ISO/IEC 14598-4: Esta parte provee los requisitos y guías para que la evaluación del producto software sea llevada a cabo en función a los compradores que planean adquirir o reutilizar un producto de software existente o pre-desarrollado.
- ISO/IEC 14598-5: Esta parte provee los requisitos y guías para la evaluación del producto software cuando la evaluación es llevada a cabo por evaluadores independientes.
- ISO/IEC 14598-6: Esta parte provee las guías para la documentación del módulo de evaluación.

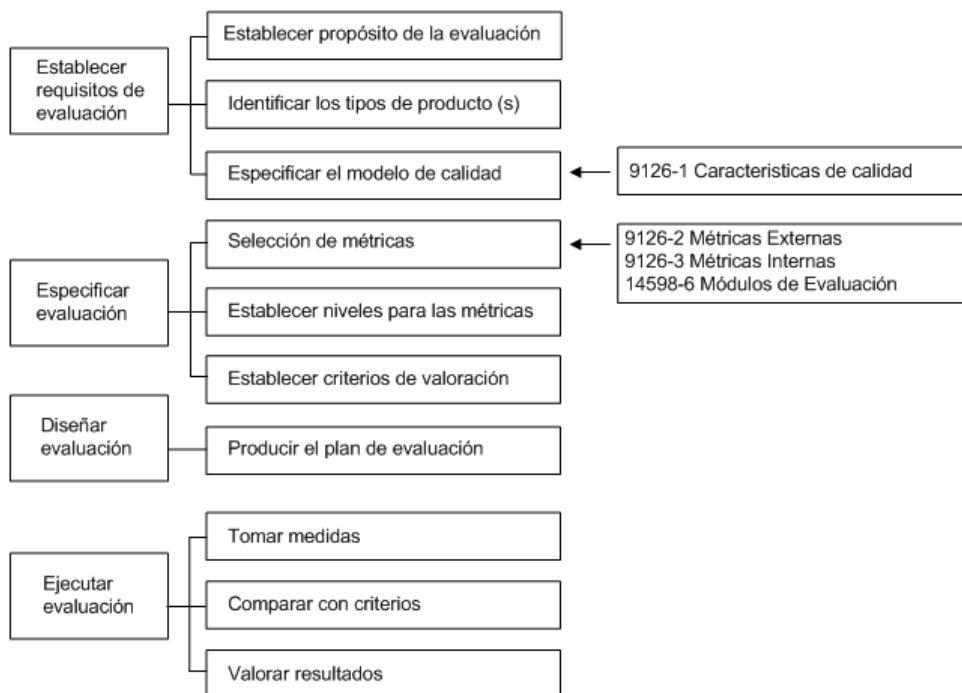


Fig 2.7. Proceso de evaluación de la ISO 14598 en conjunto con la ISO 9126.

Las ventajas que conlleva emplear la definición de usabilidad de este conjunto de normas son:

- Existe un marco claramente definido y acordado en el modelo, que aclara la definición de usabilidad, y propone métricas para proporcionar evidencia objetiva de los logros
- Se puede utilizar como una referencia para los acuerdos contractuales entre un comprador y un proveedor de software, pudiendo ser utilizados además, para eliminar una serie de malentendidos entre el comprador y el proveedor.
- Se propone un proceso de evaluación que puede ser adaptado a compradores, desarrolladores y evaluadores externos.

Sin embargo, existen algunas deficiencias en este conjunto de normas que aún no han sido plenamente abordados, tales como:

- El conjunto de métricas se proporciona para sub-características que a su vez engloban conceptos difícilmente medibles si no se descomponen en atributos
- Existe superposición de conceptos, al definir usabilidad como una característica de calidad interna-externa, y llamar calidad en uso a otras características también vinculadas a la usabilidad.
- El tener por separado dos normas que se aplican conjuntamente produce inconsistencias en los ciclos de vida de ambas dificultando su uso.

2.2.3. ESTÁNDAR ISO 25000 (SQUARE)

Aspectos como el hecho de tener dos normas complementarias: ISO/IEC 9126 (Calidad del producto software) e ISO/IEC 14598 (Evaluación del producto software), han propiciado el impulso necesario para la elaboración de la norma ISO/IEC 25000 (2005) conocida como SQuaRE (Software Quality Requirement Evaluation). La meta perseguida en la creación de esta norma es dar un paso hacia un conjunto de estándares organizados de manera más lógica, enriquecidos con nuevas aportaciones y unificados con respecto a las normas anteriores para ser capaces de cubrir los dos principales procesos: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportados por un proceso de medición. SQuaRE se centra exclusivamente en el producto software estableciendo criterios para su especificación, su medición y su evaluación. Es decir, básicamente se trata de una unificación y revisión de los estándares ISO/IEC 9126 e ISO/IEC 14598.

La norma está compuesta de cinco divisiones bien diferenciadas:

- División de gestión de calidad (ISO/IEC 2500n): Los estándares que forman esta división definen todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.
- División del modelo de calidad ¹ (ISO/IEC 2501n): El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.
- División de mediciones de calidad (ISO/IEC 2502n): Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software (ver Fig.2.8), definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.
- División de requisitos de calidad (ISO/IEC 2503n): Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado ó como entrada para un proceso de evaluación.
- División de evaluación de la calidad (ISO/IEC 2504n): Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores. Basándose en la ISO/IEC 14598.

¹ La versión de la ISO 25010 consultada en esta tesina es la de Julio 2008, por lo que en un futuro pueden existir variaciones en cuanto al modelo de calidad se refiere, al estar esta parte en fase de revisión a fecha de Octubre 2009.

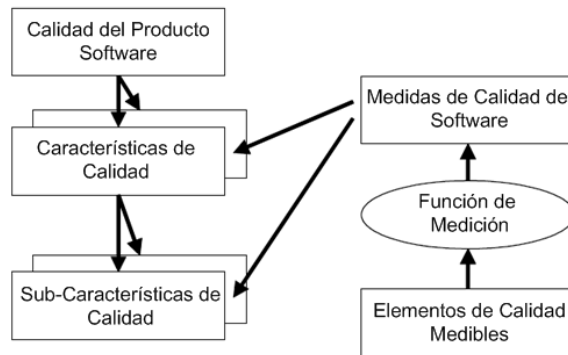


Fig 2.8. Modelo de medición de la calidad del producto software según SQuaRE.

Las principales ventajas con respecto a sus predecesores, ISO/IEC 9126 y ISO/IEC 14598 son:

- Mejor coordinación de la guía en la medición y evaluación de la calidad de los productos de software.
- Mejor orientación para la especificación de requisitos de calidad de los productos de software.
- Mejor distinción entre las partes beneficiarias del producto software (usuario final, organización y equipo de mantenimiento) del sistema y sus necesidades.
- Mejor integración de las definiciones de usabilidad gracias a las vistas de los modelos.

Las principales diferencias con respecto a los mismos son:

- La introducción de un modelo de referencia general.
- La introducción de guías detalladas y dedicadas a cada división.
- La introducción de elementos de medidas de calidad dentro de la división de medición de la calidad.
- Incorporación y revisión de un modelo de calidad para los datos.
- Incorporación y revisión del proceso de evaluación.
- Coordinación y armonización del contenido de ISO/IEC 15939 (2000).
- Introducción de guías para uso práctico en forma de ejemplos.
- Renombramientos de algunas sub-características para evitar ambigüedades.
- Incorporación de nuevas sub-características.

La principal desventaja de emplear este estándar reside en ser un estándar de reciente creación, ya que muchas de las partes que lo componen siguen en proceso de desarrollo y solo son borradores.

Con respecto al modelo de calidad planteado, existen tres vistas del modelo según el contexto donde se aplique (ver Fig.2.9): Modelo de Calidad de Software que será

utilizado para evaluar un producto software concreto; Modelo de Calidad de Datos que será utilizado para evaluar la calidad de la información que maneja el software; y el Modelo de Calidad en Uso que pretende evaluar cómo las partes beneficiarias del producto alcanzan sus objetivos usando el producto en un contexto específico de uso.

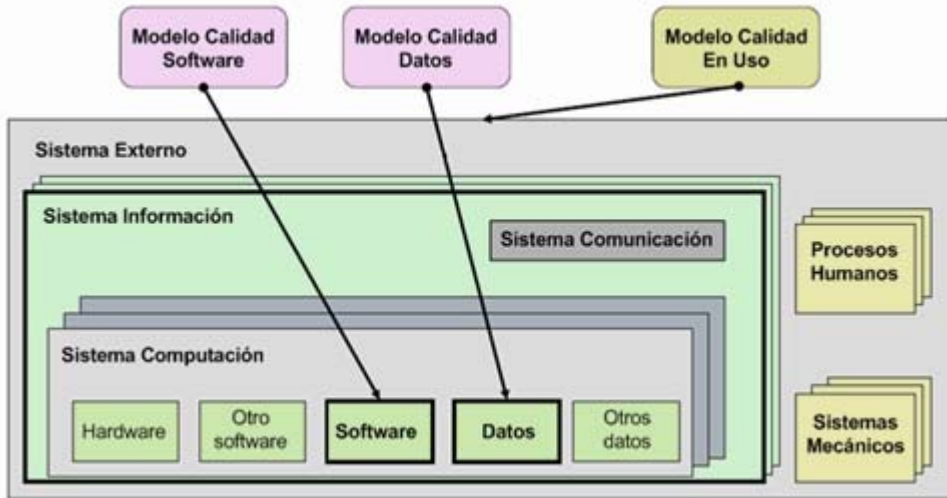


Fig 2.9. Vistas de los modelos de calidad según SQuaRE.

En cuanto al concepto de usabilidad, en esta norma se comienza a entrever una mayor convivencia entre las diferentes definiciones de usabilidad existentes. Esto es debido a una mejor distinción entre las partes beneficiarias del sistema y sus necesidades hacia el producto software: usuario final, organización y soporte técnico. De esta forma, tal y como presenta la tabla 2.3, la Calidad en Uso puede tener distintas vistas según la parte beneficiaria que consideremos [8]. Considerando la vista del usuario final tendríamos una definición de usabilidad más cercana a la planteada en la ISO 9241-11.

Tabla 2.3. Vistas de la Calidad en Uso según las partes beneficiarias [8].

Parte beneficiaria	Usuario Final	Organización	Soporte técnico
Características			
Efectividad	Efectividad del usuario	Efectividad de las tareas	Efectividad del mantenimiento
Recursos	Productividad del usuario (tiempo)	Coste-Eficiencia (dinero)	Coste del mantenimiento
Consecuencias adversas	Riesgos para el usuario (salud y seguridad)	Riesgo comercial	Corrupción o fallos del software
Satisfacción	Satisfacción del usuario	Satisfacción en la gestión	Satisfacción del mantenimiento

Además, en el modelo de calidad de software, la usabilidad sigue presente como una característica más del producto software, definida de forma muy parecida a la ISO 9126. Sin embargo, existe una propuesta de renombramiento del término *usabilidad* del Modelo de Calidad de Software por el término de *operabilidad*, haciendo que su sub-característica *operabilidad* pase a llamarse, a su vez, *facilidad de uso*. El objetivo de este renombramiento sería no provocar conflictos al llamar también *usabilidad* a la vista del usuario final con respecto de la Calidad en Uso. Además, se incorporan dos nuevas sub-características: “facilidad de ayuda”, que hace referencia al grado con el que el producto ofrece ayuda a los usuarios cuando la necesitan (en la ISO 9126-1 esta definición se consideraba implícita en la facilidad de aprendizaje) y “accesibilidad técnica”, que hace referencia a cómo el producto software puede ser usado por personas con discapacidades físicas. La propuesta de última sub-característica entraría en conflicto con los intereses de los ingenieros de accesibilidad y los de usabilidad, al tratar de incluir una en la otra.

Pese a que siguen existiendo objeciones de cara al renombramiento y descomposición de características, el aspecto positivo a destacar reside en que esta norma se puede interpretar como un primer paso hacia la armonía entre los diferentes conceptos de usabilidad, puesto que gracias a esto, la usabilidad puede tenerse en cuenta tanto en las primeras fases de desarrollo como en los contextos finales de uso.

2.3. PROPUESTAS DE EVALUACIÓN DE USABILIDAD WEB BASADAS EN ESTÁNDARES

Existe una gran variedad de propuestas de evaluación basadas en modelos de usabilidad (o de calidad, si además de la usabilidad tratan otras características de calidad) que siguen los estándares mencionados en la sección anterior o bien parten de modelos de usabilidad propuestos por el propio autor. Pese a que en esta sección se enfoca hacia las propuestas de evaluación de usabilidad basadas en estándares y orientadas a aplicaciones Web, cabe resaltar brevemente las aportaciones orientadas a aplicaciones genéricas realizadas antaño por McCall [56], Nielsen [64] y Dromey [23], ya que los estándares descritos han usado como base muchos de estos trabajos para afianzar el concepto de usabilidad.

McCall [56] presenta uno de los modelos de calidad más antiguos donde los atributos claves de un producto final desde el punto de vista del usuario son los denominados factores de calidad. Estos factores fueron clasificados en tres grupos principales: revisión del producto (*mantenibilidad, flexibilidad,...*), transición del producto (*portabilidad, interoperabilidad,...*) y operación del producto (*usabilidad, eficiencia,...*). El concepto de usabilidad comienza a estar vinculado a cómo los usuarios pueden operar con el producto de la forma más satisfactoria posible.

Nielsen [64], uno de los autores más referenciados en el tema de la usabilidad, propone un modelo de usabilidad bastante detallado centrado en los conceptos de aceptabilidad social y aceptabilidad práctica. En él se define la usabilidad como una sub-característica del concepto “utilidad”, que es a su vez, una sub-característica de la aceptabilidad práctica. Mantiene que la usabilidad puede ser descompuesta en los siguientes atributos: facilidad de aprendizaje, eficiencia de uso, facilidad para recordar, errores mínimos, y atracción subjetiva; haciendo especial hincapié en el atributo “facilidad de aprendizaje” ya que es una de las novedades que se introdujeron.

Dromey [23] plantea un modelo de calidad basado en comportamientos y usos. Donde un comportamiento es algo que exhibe el propio software en un determinado contexto (por ejemplo, la usabilidad), y los usos son las acciones de los usuarios con el software. Este modelo enumera atributos de calidad específicos y los clasifica asociándolos a determinadas características propias del software.

Aunque los modelos anteriores asentaron las bases de la usabilidad para productos Software genéricos y el desarrollo de los estándares actuales, en lo concerniente a los productos software orientados al ámbito Web hay que tener presente las características específicas que condicionan los aspectos a tener en cuenta en dichas sub-características pertenecientes a la usabilidad. Si bien las aplicaciones Web son un tipo concreto de producto software, su usabilidad puede evaluarse empleando los estándares descritos en la sección anterior. Sin embargo, el problema reside en que estos estándares, aunque resultan muy útiles como guías y recomendaciones, son muy genéricos y proponen una descomposición de la usabilidad en distintas sub-características cuyas definiciones son demasiado abstractas para ser cuantificables directamente. Por este motivo, los modelos de calidad propuestos en estos estándares deben ampliarse y/o descomponerse en atributos medibles que además tengan en cuenta las características específicas de la Web. Esto ha motivado la aparición de propuestas de evaluación de usabilidad Web (y calidad, en general). Algunos de estos ejemplos los podemos encontrar en trabajos como los de Ivory [43], Olsina y Rossi [67], Granollers [29], Calero *et al.* [15], Seffah *et al.* [72] y Moraga *et al.* [61].

Ivory [43] presenta una metodología para evaluar sitios Web la cual propone cinco etapas: a) identificación de un conjunto exhaustivo de medidas cuantitativas para interfaces de usuario, tales como la cantidad de texto en una página, el uso de los colores, consistencia, etc; b) Cálculo de dichas medidas para una muestra amplia de interfaces de usuario consideradas bien calificadas; c) Derivación de modelos estadísticos acerca de las puntuaciones obtenidas; d) Empleo de los modelos estadísticos para predecir la calificación de nuevas interfaces a evaluar, y e) Validación del modelo de predicción. Uno de los puntos fuertes de esta metodología es la automatización del proceso llevada a cabo por la herramienta WebTango, mientras que

uno de los puntos débiles es que únicamente se consideran aspectos del código final de las interfaces de usuario, basando la cuantificación del grado de usabilidad comparando similitudes entre resultados cuantitativos. Este aspecto es muy útil para establecer rankings de interfaces de usuario, pero el análisis cualitativo no queda muy bien definido, es decir, qué errores de usabilidad se encuentran y cómo solucionarlos.

Olsina y Rossi [67] presentan la metodología WebQEM cuyo objetivo es la definición de un proceso de evaluación cuantitativa de la calidad para aplicaciones Web. El proceso de evaluación consta de cuatro fases principales: a) definición y especificación de requisitos de calidad, donde se especifican características y atributos basados en la norma ISO/IEC 9126-1 (usabilidad, funcionalidad, etc.) teniendo en cuenta, además, las necesidades explícitas de los usuarios Web; b) evaluación elemental, donde se aplican métricas para cuantificar los atributos; c) evaluación global donde se produce la selección de criterios de agregación y un modelo de calificación; y d) conclusión, donde se ofrecen recomendaciones. Uno de los puntos fuertes de este trabajo es que el proceso de evaluación está claramente definido, sin embargo, la mayoría de las evaluaciones que se realizan tienen lugar sobre interfaces de usuario finales y sobre la aplicación Web totalmente operativa.

Granollers [29] propone la metodología MPLu+a que posibilita un trabajo multidisciplinar efectivo para el desarrollo de sistemas interactivos usables y accesibles, permitiendo la convergencia de personas que pertenezcan a diversas áreas del conocimiento. El modelo presenta una articulación entre los principios de la Ingeniería de la Usabilidad, la Accesibilidad y la Ingeniería de Software, y considera a los factores humanos como el aspecto relevante para el diseño de sistemas interactivos centrados en el usuario, empleando además, tanto los estándares ISO/IEC 9126 e ISO/IEC 9241, como las guías de accesibilidad propuestas por el World Wide Web Consortium (W3C). Uno de los puntos fuertes de MPLu+a, es considerar las fases de prototipado y evaluación como parte integrante e indivisible del ciclo de vida de desarrollo. Sin embargo, siempre se realizan evaluaciones con prototipos bastante evolucionados.

Calero *et al.* [15] presenta un modelo de calidad específico para la Web denominado Web Quality Model (WQM). Este modelo está definido en tres dimensiones: características de la Web (contenido, presentación y navegación), características de calidad basadas en la ISO/IEC 9126-1 (funcionalidad, fiabilidad, usabilidad, eficiencia, portabilidad y mantenibilidad) y los procesos del ciclo de vida de un sitio Web basándose en la ISO/IEC 12207 (desarrollo, operación, mantenimiento) incorporando además procesos que tienen que ver con la empresa que lleva a cabo el proyecto (gestión de proyecto y gestión de la reutilización). WQM incorpora un total de 326 métricas, específicas para productos Web, que han sido clasificadas en base a esas tres dimensiones. Uno de los aspectos más importantes es la información que aporta sobre

qué métricas Web han sido validadas teóricamente y/o empíricamente, y cuáles de ellas se prestan mejor a automatizar su cálculo. Un proceso de evaluación puede ser definido a partir de la selección del conjunto de métricas que resulte más útil para construir una expresión ponderada para presentar la "calidad total de la Web" que podría ser utilizada para cuantificar la calidad de una determinada aplicación Web. Sin embargo, no se presenta ninguna definición de este proceso de evaluación que ayude a su puesta en práctica.

Seffah *et al.* [72] propone un modelo de usabilidad consolidado denominado QUIM (Quality in Use Integrated Measurement) aplicable a diferentes tipos de interfaces, sobretodo, las pertenecientes a un sitio Web. QUIM define un primer nivel incluyendo 10 sub-características que definen la usabilidad (eficiencia, efectividad, productividad, satisfacción, aprendizaje, seguridad, confianza, accesibilidad, universalidad y utilidad). En el segundo nivel, las sub-características se descomponen en 26 atributos o criterios medibles (controlabilidad, privacidad, familiaridad, etc.), pero éstos no son el resultado de ir descomponiendo una única subcaracterística en varios atributos, si no que un mismo atributo forma parte de la definición de varias sub-características del nivel superior. En el tercer nivel encontramos 127 métricas de usabilidad asociadas a los atributos. Una de las principales ventajas de este modelo es contar con una herramienta de gestión llamada QUIM Editor que permite la edición del modelo para posibles mejoras, acceder a un repositorio con todas las métricas de usabilidad categorizadas y proveer información sobre cómo aplicar esas métricas para evaluar productos software. Sin embargo, el proceso de evaluación queda relegado exclusivamente a la aplicación de métricas, sin haber definido guías detalladas sobre como planificar la evaluación.

Moraga *et al.* [61] presenta un modelo exclusivo para la usabilidad pero orientado a la evaluación de los portlets que componen un portal Web. Los portlets son componentes modulares, pertenecientes a la interfaz de usuario, gestionadas y visualizadas en un portal Web. El modelo se basa en la ISO/IEC 9126-1 (2001) y en sus sub-características de usabilidad: facilidad de entendimiento, facilidad de aprendizaje, conformidad; sin embargo, la operabilidad se sustituye por personalización que se adapta más al contexto de los portlets. Cada una de estas sub-características se descompone en atributos a los que se les asocia una medida. Sin embargo, la finalidad de estas medidas es establecer un ranking de puntuaciones determinando umbrales de aceptación para cada atributo.

Las propuestas anteriores son representativas en cuanto a los trabajos relacionados en el área de la usabilidad Web, sin embargo, tal y como apuntaba la revisión sistemática en cuanto a métodos de evaluación de usabilidad para la Web, la mayoría se centra en las fases de implementación de las aplicaciones Web. En el caso de las propuestas de evaluación basadas en modelos de usabilidad, la situación es la misma,

los atributos en los que se descompone la usabilidad se asocian a métricas que únicamente pueden calcularse teniendo un prototipo o una aplicación Web final. El hecho de tener la usabilidad descompuesta en atributos medibles, es una ventaja de la cual se puede sacar beneficio si esos atributos pueden ser evaluados en fases tempranas del desarrollo. La mejor forma de llevar esto a cabo, es integrando evaluaciones de usabilidad en metodologías de desarrollo Web basadas en el desarrollo de software dirigido por modelos (DSDM). Esto es debido a la existencia de modelos que consisten en artefactos intermedios que permiten especificar una aplicación Web completa, los cuales además, se aplican en todas las etapas del proceso de desarrollo, permitiendo realizar evaluaciones sobre ellos. Esta línea de investigación está prácticamente en sus inicios y existen muy pocos trabajos relevantes relacionados. Los ejemplos más representativos los encontramos en los trabajos de Abrahão e Insfran [3], Panach *et al.* [69], y Molina y Toval [60].

Abrahão e Insfran [3] presentan un modelo de calidad basado en la ISO/IEC 9126-1 (2001), centrado exclusivamente en la usabilidad, siendo uno de los primeros modelos de usabilidad integrado en las fases tempranas de un proceso de desarrollo dirigido por modelos (DSDM). Se adoptó la misma estructura jerárquica propuesta en el estándar pero descomponiendo con más detalle las sub-características que definen la usabilidad (*facilidad de entendimiento, facilidad de aprendizaje, operabilidad, grado de atracción y conformidad*) en otras sub-características, y éstas en atributos medibles. Estos atributos han sido definidos teniendo en cuenta factores ergonómicos de interfaces de usuario propuestas en trabajos como los de Bastien y Scapin [6]. El principal valor añadido en este modelo fue la capacidad de evaluar los modelos independientes de plataforma que se producen como resultado de las fases tempranas. Sin embargo, este modelo fue definido para productos Software genéricos, no para aplicaciones Web. Además, el modelo carece de métricas asociadas a dichos atributos.

Panach *et al.* [69] hace uso del modelo de usabilidad definido en Abrahão e Insfran [3] para la evaluación de aplicaciones Web generadas con un método específico de desarrollo Web dirigido por modelos llamado Object Oriented Web Solutions (OOWS). El objetivo fue dotar de métricas calculables automáticamente a un conjunto de atributos relacionados con la *facilidad de aprendizaje* y la *facilidad de entendimiento*. Estas métricas se aplican a modelos conceptuales (modelo independiente de plataforma que representa la estructura estática de la aplicación Web) para establecer indicadores de usabilidad. Estos indicadores están basados en rangos de valores obtenidos tras aplicar las métricas. Uno de los puntos fuertes de este trabajo es el estudio acerca de la correlación existente entre las métricas calculadas en las fases tempranas y las impresiones recogidas en usuarios reales a través de cuestionarios. Sin embargo, no se aplican métricas a modelos, tales como interfaces abstractas de usuario que representan otras características de las aplicaciones Web.

Molina y Toval [60] presentan una propuesta para integrar requisitos de usabilidad en el desarrollo de aplicaciones Web dirigido por modelos. El objetivo es definir un meta-modelo de requisitos que describe las metas a alcanzar por los modelos navegacionales que especifican la navegación de la aplicación Web. De esta forma, extendiendo los meta-modelos de navegación se puede controlar este tipo de requisitos, esto quiere decir que un modelador podrá etiquetar los nodos presentes en sus modelos con diferentes niveles de importancia y expresar requisitos como la definición (para cada nivel de importancia) de una distancia máxima y mínima que puede separar a los nodos etiquetados con ese nivel del punto de entrada al sistema. Uno de los puntos fuertes de esta propuesta es cubrir ciertas carencias que se detectaron en la revisión sistemática, como la falta de propuestas para evaluación de usabilidad en fases tempranas. Sin embargo, los requisitos establecidos son muy dependientes a los valores de las métricas, basándose en restricciones sobre rangos de valores, lo cual puede dificultar el proceso de elicitación si las métricas poseen una definición muy técnica.

Al considerar el desarrollo dirigido por modelos como la estrategia clave en el desarrollo de aplicaciones Web, los trabajos mencionados anteriormente presentan las primeras aproximaciones hacia una evaluación temprana de la usabilidad. Sin embargo, no existe ningún proceso genérico de evaluación que se pueda emplear indistintamente en cualquier metodología de desarrollo Web dirigida por modelos, y que además, recopile de la forma más completa posible, todos los atributos de usabilidad de una aplicación Web tanto desde la perspectiva de la calidad de producto software, como desde la perspectiva de la calidad en uso. La contribución de esta tesina de máster, que se describe en el siguiente capítulo, está en sintonía con la línea de investigación marcada por los trabajos anteriores basados en DSDM. El objetivo es definir un proceso genérico de evaluación de usabilidad (WUEP - Web Usability Evaluation Process) para metodologías de desarrollo Web basadas en modelos, que presente guía detallada para realizar evaluaciones de usabilidad en cualquier metodología DSDM e integre todos los atributos de usabilidad posibles empleando para ello la perspectiva del concepto usabilidad que ofrece el nuevo estándar SQuaRE.

3. WUEP: UN PROCESO DE EVALUACIÓN DE USABILIDAD WEB PARA DSDM

En este capítulo se presenta WUEP (Web Usability Evaluation Process), un proceso genérico para la evaluación de usabilidad de aplicaciones Web desarrolladas en el contexto del desarrollo de software dirigido por modelos (DSDM).

En la sección 3.1 se ofrece una perspectiva global sobre cómo un modelo de usabilidad Web puede ser integrado y empleado para la mejora y evaluación de la usabilidad en procesos de desarrollo Web dirigido por modelos.

En la sección 3.2 se describe todas las características, sub-características y atributos que componen el modelo de usabilidad Web. Esta descripción está separada en las vistas de la calidad ofrecidas en el estándar SQuaRE: Calidad del Producto Software y Calidad en Uso.

Por último, en la sección 3.3 se detalla el proceso de evaluación haciendo énfasis en todas las fases que lo componen y en los entregables que se producen durante el proceso. Para ello se hace uso del meta-modelo SPEM 2 (Software & Systems Process Engineering Metamodel Specification, v2.0 [68]).

3.1. INTEGRACIÓN DE EVALUACIONES DE USABILIDAD EN EL DESARROLLO WEB DIRIGIDO POR MODELOS

La definición de aplicaciones Web presenta una clara separación entre dimensiones (contenido, navegación y presentación) permitiendo establecer niveles adecuados de abstracción. Un ingeniero Web puede capturar las características asociadas a cada dimensión en diferentes modelos de acuerdo a su propósito: modelos de requisitos, modelos de datos, modelos de navegación, modelos de presentación, etc. De esta forma, el ingeniero Web puede centrarse en una dimensión concreta de la aplicación Web, haciendo que la complejidad del desarrollo se reduzca efectivamente [16].

En un contexto más general, esta forma de ingeniería donde se especifican y se refinan modelos que definen un producto software se le denomina Desarrollo de Software Dirigido por Modelos (DSDM) o bien por su término internacional: *Model-Driven Engineering* (MDE). Un proceso de desarrollo Web basado en esta aproximación, básicamente transforma, mediante reglas de transformación, modelos independientes de detalles de implementación (modelos PIM - *Platform Independent Model*) en otros que aportan los aspectos específicos de una plataforma concreta (modelos PSM - *Platform Specific Model*). Estos modelos PSM contienen la información necesaria para, a partir de ellos, generar automáticamente el código fuente final (modelos CM - *Code Model*). Las ventajas de esta aproximación residen en una separación rigurosa de las

vistas que definen el producto, empleando primitivas de modelado más cercanas a conceptos del dominio que a los conceptos de la plataforma tecnológica, permitiendo además, obtener el código ejecutable de la aplicación final, de forma automática, sin invertir esfuerzos en la fase de codificación.

Estudios recientes indican que la adopción de DSDM ha crecido, y actualmente, existen varias metodologías de desarrollo Web que siguen este enfoque, como por ejemplo, OO-H [28], WebML [17], o UWE [49]. Estos métodos soportan la construcción de una aplicación Web mediante la definición de distintas vistas (modelos), incluyendo al menos un modelo estructural que representa el contenido y su comportamiento, un modelo de navegación que representa como el usuario accede al contenido, y un modelo de presentación abstracta que modela como ese contenido es mostrado al usuario. Algunos métodos también proporcionan reglas de transformación entre modelos y generación automática de código.

La Figura 3.1 muestra como la usabilidad de una aplicación Web, obtenida como resultado de un proceso DSDM en el que se definen modelos y se aplican transformaciones, puede evaluarse en distintas fases de dicho proceso [2]. Un modelo de usabilidad puede ser aplicado en los siguientes niveles de abstracción: a nivel de modelos independientes de plataforma (PIM), para evaluar modelos estructurales, modelos de navegación o modelos que representan el interfaz abstracto del usuario; a nivel de modelos específicos de plataforma (PSM), para evaluar los modelos de interfaz concreto (si los hay); a nivel de la aplicación Web final (CM), para evaluar el interfaz de usuario final; y a nivel de interacción, cuando la aplicación Web está siendo utilizada en un contexto específico.

Aunque el modelo de usabilidad Web presente un conjunto muy grande de atributos medibles, es importante resaltar que previamente debe haber un proceso de selección de atributos de usabilidad que se consideren relevantes según varios factores como el objetivo de la aplicación Web, el tipo de aplicación Web, el perfil de los usuarios, etc. Este proceso, se consideraría como parte de la elicitación de requisitos no-funcionales. Otro aspecto a tener muy en cuenta es que todos los atributos pertenecientes al modelo de usabilidad pueden ser evaluados en todos los niveles de abstracción. Cuanto mayor es el nivel de abstracción, menos atributos pueden ser considerados. También se ha de tener en cuenta que la retroalimentación que se obtiene en cada tipo de evaluación tiene distintos objetivos en función del nivel de abstracción:

- A nivel de PIM es posible evaluar los modelos que especifican la aplicación Web independientemente de la plataforma tecnológica donde se vaya a ejecutar. Estos modelos serían: modelos de clases, modelos de navegación, modelos de presentación, modelos de diálogo, etc. El conjunto de atributos medibles que pueden ser evaluados en este nivel son los principalmente relacionados con la

forma en que los usuarios navegan a través de la información, y cómo ésta será presentada por patrones que definen una interfaz de usuario (por ejemplo, la navegabilidad, la densidad de información, etc.). Sin embargo, este conjunto de atributos puede variar en función de la expresividad que posean los modelos PIM de cada método de desarrollo Web concreto. Esta evaluación generará un *informe de usabilidad independiente de plataforma* a fin de proporcionar información acerca de cómo corregir estos PIM modelos en las fases de análisis y diseño ((1) en Fig. 3.1). Por medio de las transformaciones entre modelos y la trazabilidad explícita entre éstos, los cambios realizados en el PIM son reflejados en el CM, evitando de esta forma problemas de usabilidad en la aplicación generada (CM).

- En el nivel PSM es posible evaluar los modelos concretos de interfaz de usuario asociados a una plataforma tecnológica específica. El conjunto de atributos medibles que pueden ser evaluados en este nivel es más amplio, ya que incluye atributos relacionados con componentes de software específicos (widgets) que no pueden ser considerados a nivel de PIM (por ejemplo, comportamiento de las barras de exploración, información visual desde los botones de radio, etc.) Esta evaluación generará un *informe de usabilidad específico de plataforma* a fin de proporcionar información a las fases de análisis y diseño ((1) en Fig. 3.1) si el problema de usabilidad detectado tiene su origen en el nivel PIM o a la fase de transformación de modelos ((2) en Fig. 3.1) si el problema de usabilidad detectado tiene su origen en los modelos propios PSM, así como la reglas de transformación entre PIM y PSM.
- En el nivel CM es posible evaluar la interfaz de usuario final. El conjunto de atributos que pueden ser evaluados en este nivel es el más amplio, ya que es posible considerar más aspectos relacionados con la perspectiva del usuario final (por ejemplo, compatibilidad con el navegador, metáforas cognitivas, atracción subjetiva, etc.) Esta evaluación generará un *informe de usabilidad de la aplicación final* a fin de proporcionar retroalimentación a las fases de análisis y diseño ((1) en Fig. 3.1) si el problema de usabilidad detectado tiene su origen en el nivel PIM; a la fase de transformación de modelos ((2) en Fig. 3.1) si el problema de usabilidad detectado tiene su origen en los modelos propios PSM, así como la reglas de transformación entre PIM y PSM, o a la fase de generación de código automática ((3) en Fig. 3.1) si el problema de usabilidad detectado tiene su origen en las reglas de transformación que se aplican a nivel PSM para obtener el código de la aplicación final.

Las anteriores evaluaciones tienen lugar durante el desarrollo de la aplicación Web (evaluaciones de usabilidad formativas) y pueden ser llevadas a cabo por el mismo desarrollador mediante la inspección de los modelos en diferentes niveles de

abstracción. Estas evaluaciones requerirán como apoyo la vista del modelo de usabilidad centrada en la calidad del producto software, y podrán ser aplicadas de forma iterativa hasta que los modelos (PIM, PSM, y CM) obtengan el grado de usabilidad deseado.

Sin embargo, para la evaluación de la aplicación Web en uso se requiere la participación de usuarios finales que permitan recolectar información acerca de cómo los usuarios utilizan la aplicación Web. En el nivel “en Uso” es posible evaluar la interacción de un usuario final con la aplicación Web en un determinado contexto de uso. El conjunto de atributos que pueden ser evaluados en este nivel son los referentes a cómo los usuarios alcanzan sus objetivos con efectividad, productividad, seguridad y satisfacción. Estas evaluaciones requerirán como apoyo la vista del modelo de usabilidad centrada en la calidad en Uso y se llevaran a cabo mediante “testeo con usuarios”. Esta evaluación generará un informe de usabilidad en uso a fin de proporcionar retroalimentación a todas las fases del proceso de desarrollo según el origen del problema de usabilidad detectado ((1), (2) y (3) en Fig. 3.1).

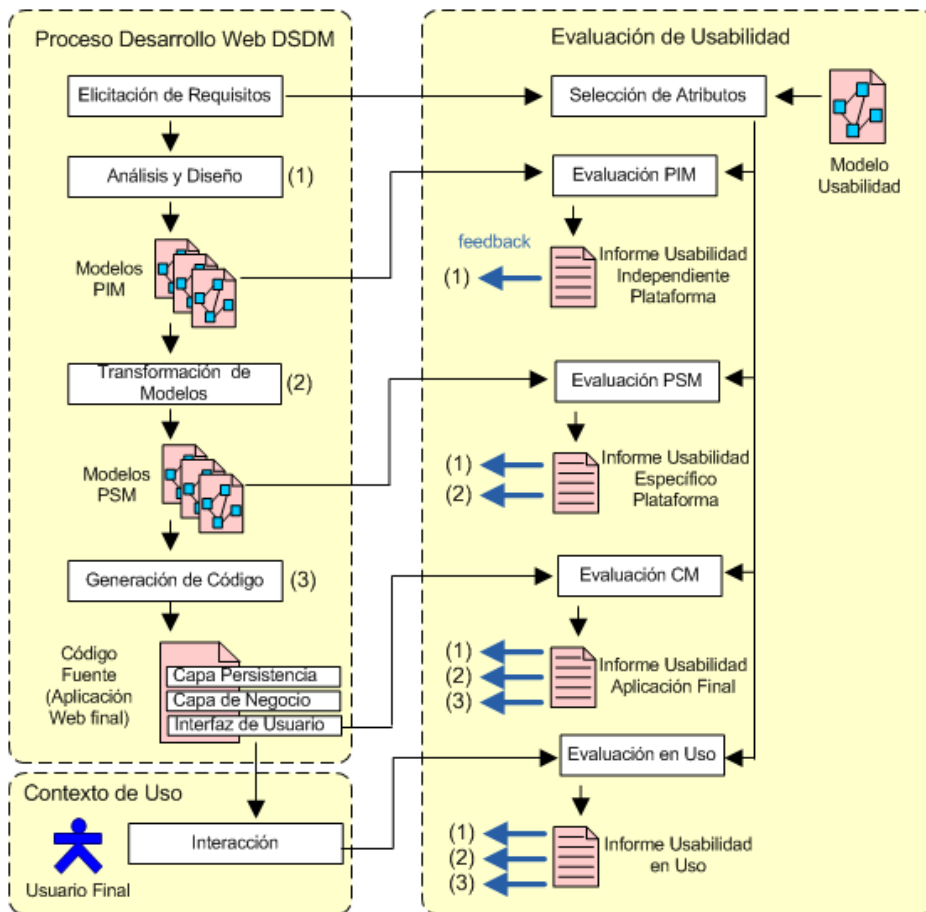


Fig 3.1. Integración de evaluaciones de usabilidad en el contexto DSDM

3.2. MODELO DE USABILIDAD WEB

El modelo de usabilidad Web propuesto toma como base el modelo de usabilidad propuesto en Abrahão e Insfran [3] para productos software genéricos. Este modelo se ha extendido y adaptado a productos orientados a la Web y a la última norma ISO 25000 SQuaRE. Para ello, el Modelo de Usabilidad Web tuvo en cuenta la división de sub-características que se realiza en la ISO 25010 que hace referencia al modelo de calidad propuesto por SQuaRE. Sin embargo, como se ha comentado anteriormente, estas sub-características son muy genéricas y están a un alto nivel de abstracción, por este motivo, el Modelo de Usabilidad Web propuesto en esta sección, descompone dichas sub-características en atributos cercanos al dominio Web, intentando que las definiciones de los atributos se solapen lo mínimo posible entre ellas, con el objetivo de obtener un modelo de usabilidad lo más amplio posible centrado en el ámbito de las aplicaciones Web.

En la extensión del modelo hacia productos Web se han tenido en cuenta los criterios ergonómicos propuestos en Bastien y Scapin [6], guías de usabilidad basadas para el desarrollo Web para descubrir que atributos relacionados con la usabilidad son relevantes en este ámbito [50][53], y métricas Web que han sido aplicadas en la literatura existente para descubrir qué atributos de usabilidad pretenden cuantificar. Por otro lado, la adaptación al modelo de calidad propuesto por la norma ISO 25000 SQuaRE [42], ha desencadenado la necesidad de contemplar dos vistas del modelo de usabilidad: una que contemple la usabilidad de la aplicación Web vista desde la perspectiva de un producto software, y otra que contemple la usabilidad de la aplicación Web desde la perspectiva de los usuarios en un determinado contexto (“usabilidad en uso”).

Por último, se ha estudiado métricas Web que han sido empleadas en la literatura existente, con el objetivo de proveer un conjunto de métricas genéricas que puedan ser operacionalizadas en distintos niveles de abstracción y en distintos métodos de desarrollo Web bajo enfoque DSDM. Cada métrica fue asociada a un único atributo, pudiendo pertenecer éste a cualquier vista del modelo, con el objetivo de descubrir problemas de usabilidad en base a los valores obtenidos de la métrica, pudiendo además cuantificar cómo dicho atributo afecta al nivel de usabilidad de la aplicación Web.

En las dos siguientes sub-secciones (3.2.1 y 3.2.2) se presentan una descripción detallada de cada vista del Modelo de Usabilidad Web, describiendo cada sub-característica y atributo en los que estas vistas se descomponen. La última sub-sección (3.2.3) se describe una muestra del conjunto de métricas genéricas que se asocian a cada atributo.

3.2.1. VISTA DEL MODELO DE USABILIDAD CENTRADA EN LA CALIDAD PRODUCTO SOFTWARE

SQuaRE propone que la usabilidad de un producto software puede ser descompuesta en las siguientes características: Facilidad de entendimiento (*Appropriateness recognisability*), Facilidad de aprendizaje (*Learnability*), Facilidad de uso (*Ease of use*), Facilidad de ayuda (*Helpfulness*), Accesibilidad técnica (*Technical accesibility*), Grado de atracción (*Attractiveness*), y Adherencia a normas o convenciones (*Compliance*). Sin embargo, estas características son demasiado abstractas y necesitan ser descompuestas en atributos más fácilmente medibles.

Las cinco primeras sub-características (facilidad de entendimiento, facilidad de aprendizaje, facilidad de ayuda, facilidad de uso y accesibilidad técnica) están relacionadas con el rendimiento del usuario hacia la aplicación y pueden ser cuantificadas de una forma más objetiva.

La **facilidad de entendimiento** (*Appropriateness recognisability*), hace referencia a todas aquellas características de la aplicación Web que facilitan su entendimiento. Se corresponde con el atributo *Understandability* definido en la ISO 9126-1 [35], el cual ha sido renombrado para ofrecer una definición más precisa. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto diferenciando entre aquellos atributos que permiten la *legibilidad* visual de los textos e imágenes y aquellos atributos que permiten *facilidad de lectura de la información* en lo que se refiere a la densidad y agrupación cohesionada de la información. Además, se ha descompuesto en otras sub-características como la *familiaridad* con la que un usuario reconoce aspectos de la interfaz de usuario; el *ahorro de esfuerzo cognitivo* reduciendo los pasos necesarios para completar tareas, y *orientación al usuario* a través de mensajes de aviso y actualización junto con una retroalimentación inmediata de los elementos de la interfaz que indique su posición actual o acciones que está realizando en ese mismo instante.

Las sub-características anteriores provienen de criterios ergonómicos que bien podrían ser aplicables a cualquier tipo de interfaz de usuario, sin embargo, teniendo en cuenta las guías de diseño Web y usabilidad, se ha incorporado la sub-característica *navegabilidad* por ser de gran relevancia en cualquier producto software orientado a la Web, la cual hace referencia a la facilidad que aporta la aplicación Web para navegar sobre su contenido. La Tabla 3.1 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.1. Descomposición de la sub-característica “Facilidad de entendimiento”.

Subcaracterística	Atributo	Significado
1.1 Legibilidad visual	1.1.1 Adecuación de fuente	Adecuación de la fuente (color, tipo, tamaño) al contexto.
	1.1.2 Adecuación de la visualización textual	La combinación de colores de texto y su fondo no debe impedir su lectura.

Subcaracterística	Atributo	Significado
	1.1.3 Disposición	Posición del texto visible en cualquier situación (evitando scroll horizontal)
1.2 Facilidad de lectura	1.2.1 Agrupación Cohesiva de la Información	La información se presenta en grupos con un mismo núcleo temático.
	1.2.2 Densidad de información	Cantidad de información necesaria para evitar sobrecarga.
1.3 Familiaridad	1.3.1 Consistencia de formato	Conceptos representados siempre con los mismos formatos (Fecha dd/mm/aaaa)
	1.3.3 Metáfora	Uso de metáforas que ayuden a una interacción más natural imitando objetos del mundo real
	1.3.2 Internacionalización	Uso de elementos y formas que siguen estándares.
1.4 Ahorro de esfuerzo	1.4.1 Acciones mínimas	Reducción del esfuerzo cognitivo (ej. realizar acciones en pocos pasos).
	1.4.2 Auto-descripción	Los elementos transmiten un concepto de la forma más clara y concisa posible.
	1.4.3 Complejidad de la información	Dificultad de entendimiento de la información proporcionado por la aplicación Web.
1.5 Orientación al usuario	1.5.1 Calidad de los mensajes de actualización	Los mensajes son útiles para que el usuario identifique nuevas interacciones que provee la aplicación Web.
	1.5.2 Calidad de los mensajes de aviso	Los mensajes son avisan adecuadamente al usuario sobre la acción que va a llevar a cabo.
	1.5.3 Retroalimentación inmediata de los controles	Los controles de la interfaz muestran al usuario la posición actual en la aplicación Web.
1.6 Navegabilidad	1.6.1 Soporte a búsqueda interna	La aplicación Web permite alcanzar contenidos si navegar explícitamente por los enlaces determinados a alcanzar ese contenido.
	1.6.2 Clickabilidad	Capacidad de un enlace para ser reconocido como tal.
	1.6.3 Interconectividad	Grado de interconexión entre los contenidos y/o acciones de la aplicación Web.
	1.6.4 Alcanzabilidad	Facilidad de acceder a los contenidos y/o acciones de la aplicación Web.

La **facilidad de aprendizaje** (*Learnability*) hace referencia a todos aquellos atributos presentes en una aplicación Web que hacen posible que el usuario aprenda su uso. Su definición se viene del concepto “adecuación para el aprendizaje” recogido en la ISO 9241-10 [37], y que debido a su relevancia, ya fue incorporada en modelos anteriores como una sub-característica más. En el Modelo de Usabilidad Web propuesto, se ha

descompuesto en las sub-características: *predictibilidad*, para referirse a la facilidad con la que el usuario puede predecir el resultado de las acciones que va a realizar basándose en los elementos de la interfaz; *potencialidad*, para referirse a cómo los usuarios pueden descubrir qué acciones pueden realizar en su siguiente paso de interacción; y *retroalimentación informativa*, para referirse a la información que se proporciona acerca del resultado de las acciones que se han llevado a cabo anteriormente. Muchos de los conceptos anteriores, están relacionados con el término *Affordance*, empleado en la comunidad IPO para determinar como de intuitiva es la interacción, pero proyectado en la propia aplicación Web. Estas sub-características cobran especial interés en las aplicaciones Web, ya que un usuario no suele dedicar mucho tiempo a aprender el funcionamiento, ya que si siente frustración es probable que recurra a otras alternativas. La Tabla 3.2 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.2. Descomposición de la sub-característica “Facilidad de aprendizaje”.

Subcaracterística	Atributo	Significado
2.1 Predictibilidad	2.1.1 Nombres de enlaces significativos	Capacidad de predecir la acción que se va a realizar atendiendo al nombre de los enlaces.
	2.1.2 Etiquetas significativas	Capacidad de predecir qué concepto representa una etiqueta asociada a cualquier elemento de la interfaz.
	2.1.3 Controles significativos	Capacidad de predecir que acción realizará un control determinado.
	2.1.4 Contenido multimedia significativo	Capacidad de predecir el objetivo de la aplicación Web atendiendo a los elementos multimedia que emplea (imágenes, videos, sonidos)
2.2 Potencialidad	2.2.1 Determinación de acciones posibles	Facilidad con la que el usuario puede reconocer de forma rápida y clara qué acciones puede realizar.
	2.2.2 Determinación de acciones prometedoras	Facilidad con la que el usuario puede reconocer de forma rápida y clara qué acciones son más relevantes.
2.3 Retroalimentación informativa	2.3.1 Progreso explícito de las transacciones	Capacidad de la aplicación Web de proporcionar a los usuarios el estado de las transacciones que se realizan (tareas completadas con éxito, estado en una transacción, etc.).
	2.3.2 Contexto explícito del usuario	Capacidad de proporcionar a los usuarios el contexto en el que se encuentran dentro de la aplicación Web. (Sesión iniciada, nivel de privacidad de la información, etc.).

La **facilidad de uso** (*Ease of use*) hace referencia a todos aquellos atributos de la aplicación Web que permiten controlarla y operarla adecuadamente. Se corresponde con el atributo *Operability* definido en la ISO 9126-1 [35], estando su definición basada en los términos controlabilidad, tolerancia a fallos y conformidad con las expectativas del usuario definidos en la ISO 9241-10 [37]. En el Modelo de Usabilidad Web propuesto, la facilidad de uso se ha descompuesto en otras sub-características más relacionadas con aspectos técnicos de la aplicación Web: *Compatibilidad* con los otros productos software que influyen en su funcionamiento y otros agentes externos; *Gestión de datos* en cuanto a la validez en su introducción y representación de los mismos en la interfaz; *Controlabilidad* en la ejecución de los servicios que se proveen al usuario; *Capacidad de adaptación* diferenciando entre “adaptabilidad”, que se refiere a la capacidad de la aplicación Web de ser adaptada por el usuario, y “adaptativo” que se corresponde con la capacidad de la aplicación Web de adaptarse a las necesidades de los usuarios; Consistencia en el comportamiento de los enlaces y controles de la aplicación Web; y gestión de errores, que hace referencia a la capacidad de prevenir y recuperarse de los errores que surgen en la ejecución, refiriéndose también a la calidad de los mensajes de error. La Tabla 3.3 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.3. Descomposición de la sub-característica “Facilidad de uso”.

Subcaracterística	Atributo	Significado
3.1 Compatibilidad	3.1.1 Compatibilidad con navegadores y plugins	Capacidad de la aplicación Web de ser ejecutada en los navegadores más comunes sin alterar su comportamiento y apariencia.
	3.1.2 Compatibilidad con sistemas operativos	Capacidad de la aplicación Web de ser visualizada los sistemas operativos más comunes sin alterar su comportamiento y apariencia.
	3.1.3 Compatibilidad con velocidades de conexión	Capacidad de la aplicación Web de ser usada bajo las velocidades de conexión más comunes.
	3.1.4 Compatibilidad con la resolución de pantalla	Capacidad de la aplicación Web de ser adaptable a las resoluciones de pantalla más comunes.
3.2 Gestión de los datos	3.2.1 Validez de los datos de entrada	Se proveen mecanismos que verifiquen la validez datos que introduce el usuario.
	3.2.2 Visibilidad de los datos	Se proveen mecanismos que muestren la información de acuerdo a su privacidad.
3.3 Controlabilidad	3.3.1 Edición posterior	El contenido introducido por el usuario se puede editar en cualquier momento.
	3.3.2 Soporte a operaciones de cancelación	Las acciones se pueden cancelar sin efectos perjudiciales al funcionamiento

Subcaracterística	Atributo	Significado
		normal.
	3.3.3 Soporte a la interrupción	Las acciones se pueden interrumpir sin perjudiciales al funcionamiento normal.
	3.3.4 Soporte a operaciones de deshacer	Las acciones se pueden deshacer sin perjudiciales al funcionamiento normal.
	3.3.5 Soporte a operaciones de rehacer	Las acciones se pueden rehacer para ahorrar trabajo al usuario.
	3.3.6 Soporte a la impresión	Capacidad de la aplicación Web de imprimir su contenido de forma correcta
3.4 Capacidad de adaptación	3.4.1 Adaptabilidad	Capacidad de la aplicación Web de ser adaptada por los usuarios
	3.4.2 Adaptativo	Capacidad de la aplicación Web para adaptarse a las necesidades de los distintos usuarios.
3.5 Consistencia	3.5.1 Comportamiento constante de los enlaces	Enlaces con el mismo nombre siempre apuntan al mismo destino.
	3.5.2 Comportamiento constante de los controles	Los controles siempre se comportan de la misma forma.
	3.5.3 Permanencia de los enlaces	Los enlaces se mantienen siempre en la misma posición de la interfaz
	3.5.4 Permanencia de los controles	Los controles aparecen siempre que las acciones asociadas a ellos se puedan realizar.
	3.5.5 Consistencia en el orden de los enlaces	Los enlaces pertenecientes a un mismo grupo siempre aparecen en el mismo orden.
	3.5.6 Consistencia en el orden de los controles	Los controles respetan su posición respecto a otros para no confundir al usuario.
	3.5.7 Consistencia en las etiquetas	Las etiquetas se corresponden con las acciones que representan.
3.6 Gestión de errores	3.6.1 Prevención de errores	Capacidad de la aplicación Web de proporcionar mecanismos para prevenir errores comunes.
	3.6.2 Recuperación ante errores	Capacidad de la aplicación Web de volver a un estado estable tras un error.
	3.6.3 Calidad de los mensajes de error	El mensaje representa de forma clara y concisa el error ocurrido

La **facilidad de ayuda** (*Helpfulness*) hace referencia a todos aquellos atributos de la aplicación Web que proveen ayuda a los usuarios cuando lo necesitan. Esta subcaracterística es una de las novedades presentes en SQuaRE, ya que en modelos de calidad anteriores como el de la ISO 9126-1 [35], esta definición se incluía dentro de la facilidad de aprendizaje. La *facilidad de ayuda* cobra un especial interés en las aplicaciones Web puesto que en estos entornos, los típicos manuales de usuario de antaño desaparecen para dar paso a la ayuda online, en la cual se valora más la facilidad

para encontrar respuestas rápidas a problemas comunes (el origen de las FAQs: Frequent Answered Questions). En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en atributos centrados en ofrecer al usuario un apoyo adicional en sus tareas mientras usa la aplicación Web, tal y como se muestra en la tabla 3.4.

Tabla 3.4. Descomposición de la sub-característica “Facilidad de ayuda”.

Subcaracterística	Atributo	Significado
4. Facilidad de ayuda	4.1 Completitud de la ayuda online	Los ayuda online contempla todas las funcionalidades especificadas para la aplicación Web.
	4.2 Documentación Multi-usuario	La ayuda está personalizada según el público objetivo de la aplicación Web (rol del usuario, idioma del usuario, etc)
	4.3 Completitud del mapa de sitio	El mapa del sitio Web es de gran utilidad si contempla, de forma jerárquica, la estructura de la Web para que el usuario alcance sus objetivos
	4.4 Calidad de los mensajes de asesoramiento	Los mensajes de asesoramiento han de ser claros y concisos para ayudar de la mejor forma posible al usuario (uso de tooltips, consejos antes de empezar, sugerencias, etc)

La **accesibilidad técnica** (*Technical accesibility*) hace referencia a todos aquellos atributos de la aplicación Web que permiten a los usuarios con discapacidades específicas operar con la aplicación Web. Esta sub-característica es otra de las novedades de la norma SQuARE, en un intento de unir los conceptos accesibilidad y usabilidad. Pese a que el concepto de accesibilidad es tan amplio, que probablemente requeriría un modelo específico, en el Modelo de Usabilidad Web propuesto, la presente sub-característica se ha considerado contemplando no solo discapacidades del usuario, sino también las de su entorno. De esta forma, en la tabla 3.5 se muestra en qué atributos se ha descompuesto la accesibilidad técnica, tomando como base guías de accesibilidad propuestas por la iniciativa de accesibilidad Web (WAI) del World Wide Web Consortium (W3C) [82], los cuales hemos adaptado para ofrecer una definición independiente de la plataforma tecnológica (ej. Validación código HTML) y más centrada en cualidades deseables de la aplicación Web.

Tabla 3.5. Descomposición de la sub-característica “Accesibilidad técnica”.

Subcaracterística	Atributo	Significado
5. Accesibilidad técnica	5.1 Soporte a la ampliación/reducción del texto	El texto de la aplicación Web debe redimensionarse independientemente de las opciones que ofrezca el navegador para ello.
	5.2 Independencia de dispositivos de control	El contenido debe ser accesible sin importar el tipo de dispositivo de entrada empleado

Subcaracterística	Atributo	Significado
		(ratón, teclado, entrada por voz).
	5.3 Soporte a texto alternativo	El contenido multimedia (imágenes, sonidos, animaciones) deben tener una descripción alternativa.
	5.4 Colores seguros	Los colores empleados no dañan la integridad de los usuarios con problemas específicos.
	5.5 Grado de conformidad con WCAG	Capacidad de la aplicación Web de seguir las directrices del Web Content Accessibility Guidelines.

Las dos últimas sub-características están relacionadas con la percepción subjetiva del usuario final en el caso del *Grado de atracción*; o con la del evaluador en el caso de *Adherencia a normas*.

El **grado de atracción** (*Attractiveness*) hace referencia a aquellos aspectos estéticos de la aplicación Web que influyen en las impresiones que el usuario percibe. Esta sub-característica ya aparecía en normas anteriores a la SQuaRE. Si bien es cierto que la atracción de un usuario es un aspecto claramente subjetivo y condicionado por muchos factores en un contexto determinado de uso, es posible definir atributos presentes en la misma aplicación Web que tienen un gran impacto en la percepción del usuario. De esta forma, en el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en atributos como la *uniformidad* de los elementos presentes en las interfaces de usuario (fuente, color, posición, etc), la *personalización de la estética* de la interfaz de usuario, la cual no se debe confundir con la sub-característica 3.4 “*Capacidad de adaptación*”, ya que en esta última se hacía referencia a las necesidades del usuario, no a las preferencias estéticas; y al *grado de interactividad*, tomando la definición de interactividad propuesta por Steuer [74]: “grado en el que un usuario puede participar en la modificación de la forma y contenido de un entorno mediático en tiempo real”, este es un concepto que últimamente ha ido ganando relevancia gracias a los entornos colaborativos y las redes sociales a través de aplicaciones Web. La tabla 3.5 presenta la descomposición del grado de atracción en atributos que representarían aquellos aspectos medibles en los que no sería estrictamente necesaria la intervención del usuario.

Tabla 3.6. Descomposición de la sub-característica “Grado de atracción”.

Subcaracterística	Atributo	Significado
6. Grado de atracción	6.1 Uniformidad del color de fondo	Los colores de fondo empleados en los mismos elementos de las interfaces el usuario son siempre los mismos.
	6.2 Uniformidad de la fuente	El color, estilo y tipo de fuente empleados en los mismos elementos de las interfaces de usuario no varía.

Subcaracterística	Atributo	Significado
	6.3 Uniformidad en la posición de las secciones de la interfaz	Las secciones en las que se divide la interfaz de usuario se mantienen a lo largo de toda la aplicación Web.
	6.4 Personalización de la estética de la interfaz de usuario	Las características estéticas (color, estilos temáticos) de una interfaz pueden ser seleccionadas por el usuario a su gusto.
	6.5 Grado de interactividad	El usuario es capaz puede participar en la modificación de la forma y contenido de la aplicación Web interactuando con ella.

Para finalizar la vista del Modelo de Usabilidad Web centrada en la perspectiva del producto software, la última sub-característica **adherencia a normas o convenciones** (*Compliance*) hace referencia a cómo la aplicación Web es conforme respecto a normas, estándares, convenciones o guías de diseño en el dominio Web. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en atributos como la conformidad a la norma ISO/IEC 25000 SQuaRE [42] por ser ésta la norma en la que se basa el modelo, y la conformidad con respecto algunas de las guías de diseño Web más relevantes, tal y como muestra la Tabla 3.7. Estos atributos se cuantificarían comprobando que porcentaje de patrones o guías propuestas en dichas normas se han tenido en cuenta en el desarrollo de la aplicación web

Tabla 3.7. Descomposición de la sub-característica “Adherencia a normas o convenciones”.

Subcaracterística	Atributo
7. Adherencia a normas o convenciones	7.1 Grado de conformidad a la ISO/IEC 25000 SQuaRE [42]
	7.2 Grado de conformidad a “Research-Based Web Desing & Usability Guidelines” [50].
	7.3 Grado de conformidad a “Web Style Guide” [53].
	7.4 Grado de conformidad a “Microsoft Web Design Guidelines” [59]
	7.5 Grado de conformidad a “Sun Guide to Web Style” [75]
	7.6 Grado de conformidad a “IBM Web Design Guidelines” [34]

3.2.2. VISTA DEL MODELO DE USABILIDAD CENTRADA EN LA CALIDAD EN USO

Una de las novedades más importante de SQuaRE es la descomposición de la Calidad en Uso de un producto software en tres sub-características de alto nivel: Usabilidad en Uso (*Usability in Use*), Flexibilidad en Uso (*Flexibility in Use*) y la Seguridad (*Safety*). Por lo tanto, el Modelo de Usabilidad Web propuesto contempla esta nueva vista de la usabilidad, permitiendo una convivencia entre las distintas definiciones de usabilidad proporcionadas en anteriores normas.

SQuaRE propone que la *usabilidad en uso* de un producto software puede ser descompuesta en las siguientes características: Efectividad en uso (*Effectiveness in use*), Eficiencia en uso (*Efficiency in use*), Satisfacción en uso (*Satisfaction in use*) y Usabilidad

en uso adherida a normas (*Usability in use compliance*). Sin embargo, estas características son demasiado abstractas y necesitan ser descompuestas en atributos medibles que requieren de la participación del usuario final.

La **Efectividad en uso** se define como “el grado con el cual usuarios específicos pueden alcanzar específicas metas con completitud y precisión en un contexto de uso específico”, siendo esta definición muy similar a la que se propone en la norma ISO 9241-11. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en otras sub-características como la *facilidad de la ayuda*, la cual aparece también en la vista del Modelo de Usabilidad Web asociada al producto, pero en este caso, los atributos en los que se descompone están orientados a los resultados de la interacción del usuario final; y el *rendimiento de las tareas del usuario*, donde se contempla si los usuarios son capaces de llevar a cabo todas sus tareas en la aplicación Web de la forma más exacta posible. La Tabla 3.8 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.8. Descomposición de la sub-característica “Efectividad en uso”.

Subcaracterística	Atributo	Significado
8.1. Facilidad de ayuda	8.1.1 Efectividad de la ayuda online	La ayuda online que se proporciona permite al usuario entender qué procedimientos debe seguir para realizar sus tareas
	8.1.2 Completitud de la ayuda online	La ayuda online cubre todos los problemas que los usuarios han encontrado en la realización de sus tareas.
	8.1.3 Frecuencia de consulta de ayuda	La frecuencia con la que un usuario se desorienta y necesita recurrir a recursos extra.
8.2 Rendimiento de las tareas del usuario	8.2.1 Completado de las tareas	El usuario es capaz de realizar todas sus tareas sin importar el procedimiento empleado.
	8.2.2 Exactitud de las tareas	El usuario es capaz de realizar todas sus tareas de la forma correcta, siguiendo los procedimientos que en la especificación de la aplicación Web se contemplaron.

La **Eficiencia en uso** se define como “el grado con el cual usuarios específicos utilizan la cantidad adecuada de recursos en la relación a la efectividad obtenida en un contexto de uso específico”, siendo esta definición muy parecida a la del concepto *eficiencia* que se propone en la norma ISO 9241-11, y al concepto *productividad* que se propone en la norma ISO 9126-1. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en otras sub-características relacionadas con la *eficiencia de las tareas* llevadas a cabo por el usuario, el *esfuerzo cognitivo* que representa para el usuario enfrentarse al uso de la aplicación Web; y las *limitaciones del contexto*, que pese a no depender estrictamente del usuario, condicionan en buena medida la eficiencia en uso.

La Tabla 3.9 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.9. Descomposición de la sub-característica “Eficiencia en uso”.

Subcaracterística	Atributo	Significado
9.1 Eficiencia de las tareas del usuario	9.1.1 Tiempo para completar las tareas	Los usuarios realizan sus tareas correctamente en el menor tiempo posible.
	9.1.2 Carga de la tarea	La tarea está diseñada para realizarse de la forma más rápida e intuitiva posible.
9.2 Esfuerzo cognitivo	9.2.1 Esfuerzo mental subjetivo	Grado de esfuerzo mental que tiene que realizar el usuario para obtener un nivel de rendimiento adecuado.
	9.2.2 Facilidad de recordar (memorabilidad)	Tiempo necesario para que el usuario recuerde de forma precisa una funcionalidad de la aplicación Web usada anteriormente.
9.3 Limitaciones del contexto	9.3.1 Carga del sistema	Grado en el que los procesos ajenos a la aplicación Web afectan al correcto funcionamiento de ésta.
	9.3.2 Adaptación a las habilidades del usuario	Grado en el que se han considerado algunas limitaciones del usuario como la edad o contextos culturales.

La **Satisfacción en uso** se define como “el grado con el cual los usuarios se sienten satisfechos en un contexto de uso específico”, siendo esta definición muy parecida a la del concepto *satisfacción* que se propone en la norma ISO 9241-11. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en otras sub-características basadas en distintas dimensiones de la satisfacción. Estas dimensiones son: cognitiva, cuando el usuario percibe que la aplicación cumple con la funcionalidad que él esperaba encontrar; emocional, cuando el usuario se siente atraído mientras usa la aplicación Web; física, cuando el usuario no percibe que su integridad física está amenazada; y basada en la confianza, cuando el usuario confía en que el funcionamiento de la aplicación Web no perjudicará sus intereses. La Tabla 3.10 muestra, de forma más detallada, la descomposición de estas sub-características anteriores en atributos medibles.

Tabla 3.10. Descomposición de la sub-característica “Satisfacción en uso”.

Subcaracterística	Atributo	Significado
10. 1 Satisfacción cognitiva	10.1.1 Utilidad percibida	El usuario percibe que la aplicación Web cubre las necesidades que le condujo a usarla.
	10.1.2 Calidad de los resultados	Los resultados que obtiene el usuario tras la interacción son los deseados.
10.2 Satisfacción emocional	10.2.1 Atracción subjetiva percibida	El usuario encuentra atractivo el diseño y apariencia de la interfaz de usuario.
	10.2.2 Frustración	El usuario percibe que no es capaz de lograr

Subcaracterística	Atributo	Significado
	percibida	su objetivo tras varios intentos.
	10.2.3 Riesgos del contenido	El usuario percibe una discriminación hacia él basada en aspectos socio-culturales.
10.3 Satisfacción física		El usuario puede realizar todas sus tareas sin correr ningún riesgo que afecte a su salud. (Epilepsia).
10.4 Confianza	10.4.1 Aparición de errores	El usuario tiende a desconfiar de una aplicación Web cuando ésta muestra una cantidad considerable de errores.
	10.4.2 Credibilidad del sitio	El usuario percibe que la información que recibe acerca de sus tareas es verdadera y contrastada.
	10.4.3 Riesgos económicos	El usuario puede realizar todas sus tareas sin correr ningún riesgo que afecte a la pérdida de dinero por su parte.

La **Usabilidad en uso adherida a normas o convenciones** (*Usability in use compliance*) hace referencia a cómo los usuarios interactúan conforme a normas, estándares, convenciones o guías de diseño en el dominio Web. En el Modelo de Usabilidad Web propuesto, ésta se ha descompuesto en atributos como la conformidad a la norma ISO/IEC 25000 SQuaRE [42] por ser ésta la norma en la que se basa el modelo, la conformidad con respecto a criterios ergonómicos relevantes en el área IPO, y la conformidad con respecto a los cuestionarios más empleados en la calidad en uso: SUMI, SUS y QUIS (ver tabla 3.11).

Tabla 3.11. Descomposición de la sub-característica “Usabilidad en uso adherida a normas”.

Subcaracterística	Atributo
11. Usabilidad en uso adherida a normas	11.1 Grado de conformidad a la ISO/IEC 25000 SQuaRE [42]
	11.2 Grado de conformidad a criterios ergonómicos [6].
	11.3 Grado de conformidad con cuestionarios SUMI, SUS y QUIS [78]

3.2.3. MÉTRICAS WEB GENÉRICAS

Tras la definición del Modelo de Usabilidad Web, el siguiente paso consiste en asociar métricas genéricas que permitan detectar problemas de usabilidad y cuantificar los atributos en los cuales se ha descompuesto las distintas sub-características. El objetivo de incorporar métricas genéricas reside en dotar al Modelo de Usabilidad Web de la capacidad de ser operacionalizado en diferentes métodos de desarrollo Web, especialmente los dirigidos por modelos.

Para esta tarea se ha tomado, como punto de partida, aquellas métricas para la Web que fueron propuestas anteriormente en diversos trabajos de la literatura existente. Los principales trabajos a destacar, debido al gran número de métricas que recopilan, fueron

las métricas clasificadas con la metodología WQM como métricas relacionadas con la usabilidad [15], donde se prestó especial interés a las métricas validadas teórica o empíricamente; las métricas propuestas en la norma SQuaRE, las cuales hacen referencia directa a las métricas descritas en las partes 2 y 3 de la norma ISO 9126 [36], y métricas propuestas por el World Wide Web Consortium W3C [82]. Cabe resaltar que para aquellos atributos a los que no se han encontrado métricas a asociar de entre la literatura existente, se han propuesto nuevas métricas para ser asociadas a dichos atributos, con el objetivo de ser aplicables en más de un nivel de abstracción.

La gran mayoría de estudios que presentan métricas para Web no suelen asociarlas a atributos de calidad específicos, principalmente se basan en definir métricas que suelen aplicarse cuando la aplicación Web ya está prácticamente desarrollada, es decir, métricas sobre la interfaz de usuario o sobre el código fuente. Sin embargo, este hecho es distinto respecto a métricas relacionadas con la navegación, donde existe una gran variedad de métricas provenientes de teorías de grafos, pudiéndose aplicar en modelos de navegación que son desarrollados en fases tempranas del desarrollo Web.

El estudio de cada métrica se realiza teniendo en cuenta algunos de sus parámetros como: objetivo, interpretación, método de cálculo, artefactos/fase del proceso en los que se aplica y evidencias de su validez. Este análisis permite entender qué atributo del modelo de usabilidad podría estar más relacionado con el concepto que se pretende medir con dicha métrica. Por ejemplo, la métrica *Número de enlaces de navegación* [1] tiene como objetivo cuantificar el total de enlaces que existen entre “contextos navegacionales” aplicándose a modelos que definen los caminos de navegación del usuario, por lo tanto, sería asociada al atributo *Alcanzabilidad* (ver Tabla 3.1, atributo 1.6.4), perteneciente a la característica Navegabilidad; mientras que la métrica *estilo de fuente* [43] que pretende contabilizar el número de combinaciones diferentes de estilos de fuente aplicándola sobre la interfaz de usuario final, por lo tanto, ésta métrica sería asociada al atributo Uniformidad de la fuente (ver Tabla 3.6, atributo 2.6).

Estudiando métricas como las anteriores, es posible dar una definición de la métrica que pueda ser aplicable, en la medida de lo posible, en cada:

- **Nivel de abstracción**, puesto que una métrica hace referencia a una entidad medible, esa entidad puede estar presente en los modelos que especifican la aplicación Web a diferentes niveles de abstracción, pudiéndose evaluar en distintos artefactos. Por ejemplo, la métrica *Número de enlaces de navegación* [1] podría ser medida a nivel del modelos independientes de plataforma (PIM) si la navegación es modelada como un grafo donde los nodos representan la información a la que se accede y las aristas representan dichos enlaces

navegacionales, a nivel de modelos específicos de plataforma (PSM) si la navegación se modela en una determinada plataforma tecnológica representando los elementos que permiten la navegación entre ellos; y a nivel del CM (aplicación final) si se contabilizaran los elementos del código final (etiquetas HTML, elementos javascript, etc.) que representan dichos enlaces de navegación.

- **Método de desarrollo Web dirigido por modelos**, puesto que cada método tiene sus diferentes modelos independientes de plataforma con sus respectivas propias primitivas de modelado, las cuales pueden variar en cuanto a expresividad entre distintos métodos. Siguiendo con el ejemplo de la métrica *Número de enlaces de navegación* [1] está definida en base a los “contextos navegacionales”, el cual es un concepto propio vinculado a un método concreto de desarrollo OOWS, mientras que en otro método de desarrollo como por ejemplo OO-H, el mismo concepto se denomina destino navegacional.

La definición genérica de la métrica junto con una descripción de la misma, será una información relevante a la hora de operacionalizar dicha métrica en los distintos artefactos en diferentes métodos de desarrollo Web, haciendo del Modelo de Usabilidad Web un artefacto versátil.

A continuación se muestran algunas de las asociaciones establecidas entre atributos del modelo de usabilidad y métricas incorporadas describiendo éstas e indicando la referencia de la métrica original estudiada o si es una propuesta, la descripción genérica de la métrica, el tipo de valor obtenido, su interpretación y sobre qué niveles de abstracción (modelos en un proceso DSDM) podrían aplicarse:

- Disponibilidad de valores por defecto [36]:
 - *Atributo asociado*: Acciones mínimas (ver descripción del atributo 1.4.1).
 - *Descripción*: Ratio entre los datos que tienen un valor defecto y el total de datos, que por sus características, son dados a ofrecer un valor por defecto.
 - *Escala*: Valor real entre 0 y 1
 - *Interpretación*: Valores cercanos a 0 indican que el usuario tiene que esforzarse en introducir manualmente datos que podría proporcionarle la aplicación Web, mientras que valores cercanos a 1 indican que el usuario ahorra tiempo en introducir datos.
 - *Aplicación*: A nivel PIM si existe un modelo estructural que defina los datos que se van a emplear posibilitando la asignación de un valor por defecto. A nivel PSM si existe un modelo de datos asociado a una plataforma que cuenta con la opción e valores por defecto. A nivel CM

analizando los campos de formulario de las interfaces que por defecto aparecen con un valor asignado.

- Amplitud de la navegación [1]:
 - o *Atributo asociado*: Alcanzabilidad (ver descripción del atributo 1.6.4).
 - o *Descripción*: Nivel de amplitud de la navegación de primer nivel, es decir número de opciones de navegación al mismo tiempo.
 - o *Escala*: Entero mayor que 0.
 - o *Interpretación*: Cuanto más alto es el valor, más difícil le resulta al usuario entender la funcionalidad y contenido de la aplicación Web
 - o *Aplicación*: A nivel PIM si la navegación de primer nivel es modelada como un grafo donde los nodos representan la información a la que se accede y las aristas representan los enlaces navegacionales entre dicha información. A nivel CM si se analizan los destinos de los hipervínculos incluidos en el código fuente de la página principal de la aplicación Web.

- Profundidad de la navegación [1]:
 - o *Atributo asociado*: Alcanzabilidad (ver descripción del atributo 1.6.4).
 - o *Descripción*: Nivel de profundidad de la navegación, es decir, el número de pasos de navegación que el usuario necesita para acceder a cualquier contenido o funcionalidad de la aplicación Web.
 - o *Escala*: Entero mayor que 0.
 - o *Interpretación*: Cuanto más alto es el valor, más difícil le resulta al usuario acceder a la funcionalidad y contenido de la aplicación Web.
 - o *Aplicación*: A nivel PIM si la navegación es modelada como un grafo donde los nodos representan la información a la que se accede y las aristas representan los enlaces navegacionales entre dicha información. A nivel CM si se analizan los destinos de los hipervínculos incluidos en el código fuente de todas las páginas de la aplicación Web.

- Compactibilidad (*Compactness - Cp*) [13]:
 - o *Atributo asociado*: Interconectividad (ver descripción del atributo 1.6.3).
 - o *Descripción*: Grado de interconexión entre los nodos pertenecientes a un grafo hipermedia determinado por la fórmula:

$$Cp = \frac{Max - \sum_i \sum_j Cij}{Max - Min}$$

Donde $Max = (n^2 - n) * k$; $Min = (n^2 - n)$, n = cantidad de nodos en el grafo; k = constante superior a la cantidad de nodos; $\sum_i \sum_j Cij$ = la suma de las distancias pertenecientes a una matriz de distancias normalizadas con el factor k ; y Cij = la distancia mínima entre los nodos i y j .

- *Escala*: Valor real entre 0 y 1
 - *Interpretación*: Valores cercanos a 1 indican que desde cada nodo se puede llegar fácilmente a cualquier otro nodo en el gráfico. Sin embargo, un gráfico completamente conectado podría conducir a la desorientación. Valores cercanos a 0 indican una interconexión de la información muy pobre.
 - *Aplicación*: A nivel PIM si la navegación es modelada como un grafo donde los nodos representan la información a la que se accede y las aristas representan los enlaces navegacionales entre dicha información. A nivel CM si se analizan los destinos de los hipervínculos (enlaces) incluidos en el código fuente de todas las páginas (nodos) de la aplicación Web.
- Densidad de la navegación [1]:
- *Atributo asociado*: Alcanzabilidad (ver descripción del atributo 1.6.4).
 - *Descripción*: Indicador del volumen de información y funcionalidad que representa la navegación determinado por el cociente entre el número de enlaces navegacionales y nodos navegacionales.
 - *Escala*: Valor real mayor que 0.
 - *Interpretación*: Cuanto mayor es la densidad de la navegación, más tiempo ha de invertir el usuario en realizar una acción.
 - *Aplicación*: A nivel PIM si la navegación es modelada como un grafo donde los nodos representan la información a la que se accede y las aristas representan los enlaces navegacionales entre dicha información. A nivel CM si se analizan las páginas y los hipervínculos entre ellas incluidos en el código fuente de la aplicación Web.
- Proporción de nombres adecuados para enlaces (métrica propuesta):
- *Atributo asociado*: Nombres de enlaces significativos (ver descripción del atributo 2.2.1)
 - *Descripción*: Ratio entre el número de enlaces con nombre adecuado y el total de enlaces en una misma interfaz.
 - *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 1 indican que el usuario puede predecir correctamente el destino de los enlaces navegacionales. Valores cercanos a 0 indicarían una falta de predictibilidad de los enlaces.
 - *Aplicación*: A nivel PIM si existe una interfaz abstracta de usuario que refleja los nombres de los enlaces de la interfaz de usuario final. A nivel PSM si existe una interfaz concreta de usuario que refleja los nombres

de los enlaces. A nivel CM si se analizan los nombres de los hipervínculos reflejados en las páginas Web de la aplicación final.

- Proporción de mensajes de aviso significativos (métrica propuesta):
 - *Atributo asociado*: Calidad de los mensajes de aviso (ver atributo 1.5.2)
 - *Descripción*: Ratio entre el número de mensajes que muestran de forma concisa y clara el aviso y el número total de mensajes.
 - *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 1 indican que los mensajes de aviso poseen la suficiente calidad como para orientar al usuario en el transcurso de su interacción, mientras que valores cercanos a 0 indican lo contrario.
 - *Aplicación*: A nivel PIM y PSM si los mensajes de aviso se pueden representar como un contenido que se accede a través de un camino de navegación. A nivel CM si los mensajes de aviso se corresponden con páginas de aviso.

- Contraste de color [82]:
 - *Atributo asociado*: Visualización textual (ver descripción del atributo 1.1.2)
 - *Descripción*: Grado con el cual se diferencian color del texto (C1) con el color de fondo (C2). Viene determinado por la siguiente fórmula:
$$\sum |C1(i) - C2(i)|$$

Donde i se refiere a cada componente de color de la notación RGB en forma decimal, es decir, $i \in \{\text{Rojo, Verde, Azul}\}$.
 - *Escala*: Valor entero mayor o igual a 0.
 - *Interpretación*: Cuanto mayor es el valor obtenido, mejor contraste existe entre ambos colores.
 - *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstracta que contemplen los colores como un atributo más. A nivel CM si los colores están definidos en las mismas páginas que representan la interfaz de usuario final o en archivos de hojas de estilos en cascada.

- Variación en la composición de los marcos (métrica propuesta):
 - *Atributo asociado*: Uniformidad en la posición de las secciones de la interfaz (ver descripción del atributo 6.3)
 - *Descripción*: Número de veces en las que la composición de los marcos que definen una página varía.
 - *Escala*: Entero mayor que 0.

- *Interpretación*: Cuanto mayor es el resultado obtenido, peor sensación recibe el usuario de cara a la uniformidad en el diseño de la aplicación Web.
- *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstracta que permitan definir el conjunto de marcos que definen la vista final de todas las interfaces de usuario. A nivel CM evaluando la estructura de las interfaces de usuario finales.
- Número de elementos no alineados o desencuadrados (métrica propuesta):
 - *Atributo asociado*: Uniformidad en la posición de las secciones de la interfaz (ver descripción del atributo 6.3).
 - *Descripción*: Número de elementos que sobrepasan las dimensiones predefinidas en los marcos que los contienen.
 - *Escala*: Entero mayor o igual que 0.
 - *Interpretación*: Cuanto mayor es el resultado obtenido más se rompe la percepción que el usuario tiene acerca de la uniformidad y la estética de la aplicación Web.
 - *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstracta que permitan definir las propiedades de tamaño en los elementos de la interfaz. A nivel PSM si existen modelos de interfaz concretos con las mismas propiedades de tamaño especificadas. A nivel CM evaluando el aspecto visual de los elementos de las interfaces de usuario finales o bien las etiquetas propias del lenguaje de presentación que definen los tamaños máximos de los elementos.
- Proporción de imágenes con texto alternativo asociado [82]:
 - *Atributo asociado*: Soporte a texto alternativo (ver descripción del atributo 5.3)
 - *Descripción*: Ratio entre el número de imágenes con texto alternativo asociado y el número total de imágenes pertenecientes a una misma interfaz.
 - *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 1 contribuyen a una mejora en la accesibilidad técnica de la aplicación Web, ya no exclusivamente para proporcionar información textual sobre las imágenes a usuarios discapacitados, sino además, esa información textual resulta de gran utilidad para ubicar e interpretar el objetivo de las imágenes cuando existen problemas con la carga de las mismas.
 - *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstracta que permitan definir las propiedades de texto alternativo en

las imágenes asociadas a una interfaz. A nivel PSM si existen modelos de interfaz concretos con las mismas propiedades de texto especificadas. A nivel CM evaluando las etiquetas propias del lenguaje de presentación que definen los textos asociados a dichas imágenes.

– Enlaces distinguibles (métrica propuesta):

- *Atributo asociado*: Clickabilidad (ver descripción del atributo 1.6.2)
- *Descripción*: Grado con el que los elementos que sirven de enlaces con otras partes de la aplicación Web (enlaces) se diferencian de aquellos que son simple texto (etiquetas u otros elementos textuales). Viene definido por la siguiente fórmula:

$$1 - 0.5 \left(\frac{\text{Enlaces confundibles}}{\text{Número total de enlaces}} + \frac{\text{Etiquetas confundibles}}{\text{Número total de etiquetas}} \right)$$

Siendo *Enlaces confundibles* el número de enlaces que al no poseer alguno de los atributos de formato que caracterizan a un enlace (color azul y subrayado) pueden confundirse con etiquetas; y siendo *Etiquetas confundibles* el número de etiquetas que posee alguna de las características que caracterizan a un enlace (color azul o subrayado).

- *Escala*: Valor real entre 0 y 1.
- *Interpretación*: Valores cercanos a 1 indican que el usuario puede diferenciar claramente que elementos enlazan con otras partes de la aplicación Web contribuyendo a una mejora de la navegabilidad. Mientras que valores cercano a 0 indican lo contrario.
- *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstracta que permitan definir las propiedades de color y estilo de los enlaces y etiquetas. A nivel PSM si existen modelos de interfaz concretos con las mismas propiedades especificadas. A nivel CM evaluando el aspecto visual de los enlaces y otros elementos textuales de las interfaces de usuario finales o bien los elementos propios del lenguaje de presentación que definen los estilos (ej. Hojas de estilos en cascada, etiquetas HTML para color y fuente, etc).

– Proporción de elementos que muestran estado actual (métrica propuesta):

- *Atributo asociado*: Retro-alimentación inmediata (ver descripción del atributo 1.5.3)
- *Descripción*: Ratio entre el número de elementos que muestran estado actual del usuario sobre la interfaz y el número total de elementos que propensos a tener que dar retro-alimentación de las acciones ejecutadas sobre ellos. Estos elementos en una aplicación Web son básicamente el menú de navegación (si indica en que opción te encuentras

- actualmente), los enlaces (si se muestra retro-alimentación visual cuando el cursor está sobre ellos), los controles de entrada de datos como caja de texto o listas desplegables (si se resaltan cuando se está insertando contenido en ellas)
- *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 1 indican que la interfaz provee una retroalimentación inmediata adecuada al usuario, guiándolo en el uso de la aplicación Web.
 - *Aplicación*: A nivel PSM si existen modelos de interfaces de usuario concretos que capturan los efectos de resaltado o retro-alimentación cuando el foco están sobre ellos. A nivel CM analizando la interfaz de usuario final.
- Operaciones de usuario cancelables [36]:
- *Atributo asociado*: Soporte a opciones de cancelación (ver descripción del atributo 3.3.2)
 - *Descripción*: Ratio entre el número de funciones cancelables por el usuario y el número total de funciones que requieren la capacidad de cancelación. Estas funciones suelen ser aquellas que están vinculadas a la manipulación de datos por parte del usuario (crear, modificar y borrar).
 - *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 0 indican que el usuario no es capaz de cancelar una operación que ha comenzado, teniendo que hacer uso de opciones como el navegador como “atrás”, acción que podría no tener los efectos deseados cuando se trabaja con páginas generadas dinámicamente. Esto repercute negativamente en la controlabilidad de la aplicación Web.
 - *Aplicación*: A nivel CM evaluando si la interfaz ofrecida a cada operación provee la opción de cancelar y si esta no altera el funcionamiento normal de la aplicación Web.
- Proporción de mecanismos de validación de datos de entrada (métrica propuesta):
- *Atributo asociado*: Validez de los datos de entrada (ver descripción del atributo 3.2.1)
 - *Descripción*: Ratio entre el número de elementos de un formulario que proporcionan ayuda en la inserción de un dato correcto y el número total de elementos que son propensos a realizar una validación del formato del dato de entrada. Estos elementos suelen ser cajas de texto

- donde los datos a introducir requieren de un formato específico (ej. fechas, precios, etc.) o de un conjunto de valores concreto. (ej. cargos, recursos existentes, etc.)
- *Escala*: Valor real entre 0 y 1.
 - *Interpretación*: Valores cercanos a 1 indican que se provee una adecuada gestión de los datos ayudando a que el usuario evite encontrarse con errores que le hagan perder tiempo.
 - *Aplicación*: A nivel PIM si existen modelos de interfaz de usuario abstractos que permiten definir propiedades de los elementos que aparecen en un formulario (ej. tipo de datos, tipo de caja de texto, etc). A nivel PSM si existen modelos de interfaz de usuario concretos que permiten definir las mismas características. A nivel CM si se analiza la los campos de los formularios que presenta la interfaz de usuario final.
- Diferencias de comportamiento y aspecto de elementos de interfaz entre navegadores (métrica propuesta):
- *Atributo asociado*: Compatibilidad con navegadores y plugins (ver descripción del atributo 3.1.1)
 - *Descripción*: número de tipos de elementos que no se visualizan y/o comportan de la misma manera dependiendo del navegador Web que se utilice. Entre estos elementos se puede encontrar la tipografía y estilo, los controles de la interfaz, etc.
 - *Escala*: Entero mayor que 0
 - *Interpretación*: Cuanto mayor es el valor, peor es la compatibilidad de aplicación Web con respecto a diferentes navegadores, perjudicando directamente a la facilidad de uso de la aplicación pudiendo limitar la interacción del usuario o el alcance de sus metas solo por el hecho de utilizar un navegador distinto.
 - *Aplicación*: A nivel CM analizando las diferentes interfaces de usuario final en distintos navegadores Web.

3.3. DEFINICIÓN DEL PROCESO DE EVALUACIÓN WUEP

En esta sección se detalla WUEP (Web Usability Evaluation Process), el cual hace uso del Modelo de Usabilidad Web descrito anteriormente. Para la definición se ha empleado la notación SPEM 2 (Software & Systems Process Engineering Metamodel Specification, v2.0) [68] propuesta por el Object Management Group como uno de los estándares para especificar procesos Software. El objetivo de proveer una definición detallada del proceso de evaluación de usabilidad es guiar al evaluador de la forma más clara posible para llevar a cabo el proceso.

En la sub-sección 3.3.1 se describe brevemente los conceptos básicos de SPEM 2, sus ventajas, y la justificación de su elección para la definición de WUEP, mientras que en la sub-sección 3.3.2 se emplean los conceptos definidos anteriormente para dar una definición detallada de WUEP.

3.3.1. INTRODUCCIÓN A SPEM 2

SPEM 2 (Software & Systems Process Engineering Metamodel Specification, v2.0) es un meta-modelo para definir modelos de procesos de Ingeniería del Software y de Ingeniería de sistemas. Su alcance se limita a los elementos mínimos necesarios para definir dichos procesos sin añadir características específicas de un dominio o disciplina particular; pero sirve para procesos de diferentes estilos, culturas, niveles de formalismo, o modelos de ciclos de vida. Esta característica permite que sea un buen candidato no solo para modelar procesos de desarrollo de software, sino también procesos de evaluación de calidad, donde es necesario definir las guías y los artefactos a emplear.

La idea central de SPEM 2 para representar procesos está basada en tres elementos básicos: rol, producto de trabajo y tarea (ver Fig.3.2). Las tareas representan el esfuerzo a hacer, los roles representan quien lo hace y los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen. De esta forma, se especifica: “quien (rol) realiza qué (tarea) para, a partir de unas entradas (productos de trabajo) obtener unas salidas (productos de trabajo)”.

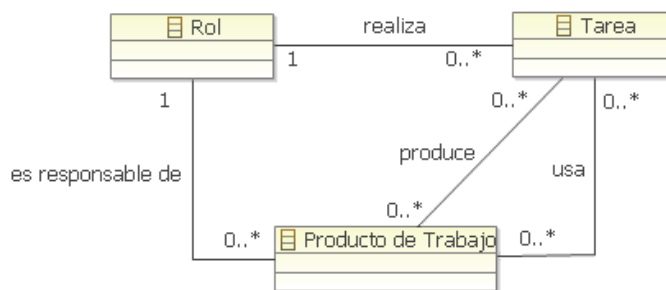


Fig 3.2. Idea central de SPEM para representar procesos

A la hora de definir un proceso, SPEM 2 distingue dos etapas (ver Fig.3.3): primero se definen los elementos esenciales que proporcionarán contenido al proceso. Estos elementos (roles, tareas, productos de trabajo, etc.) conforman el repositorio llamado Contenido (*Method Content*); y en segundo lugar, se combinan y reutilizan dichos elementos para ensamblar actividades y procesos, definiendo el flujo de trabajo entre las mismas.

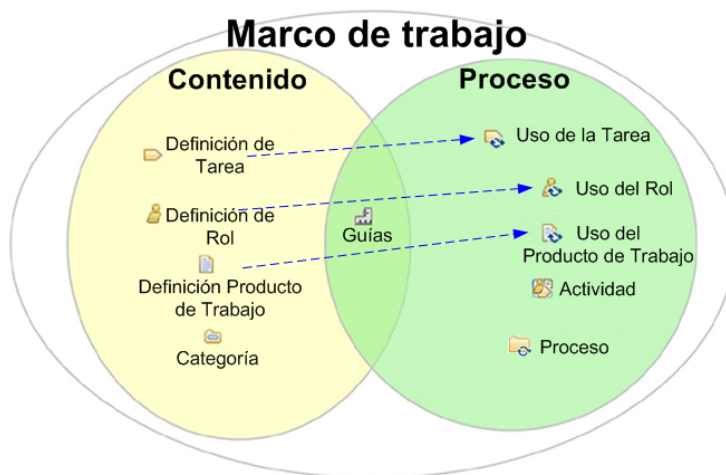














Fig 3.3. Primitivas de modelado en SPEM 2

La tabla 3.12 describe brevemente el subconjunto de primitivas de modelado que aparecen en la Figura 3.3. Éstas son las más comúnmente utilizadas a la hora de definir un proceso.

Tabla 3.12. Subconjunto de elementos para modelar procesos en SPEM 2.

Icono	Nombre	Descripción
	Definición de Rol	Es un conjunto de habilidades, competencias y responsabilidades relacionadas, de un individuo o de un grupo.
	Definición de Tarea	Describe una unidad de trabajo asignable y gestionable, identificando el trabajo que se ejecuta por los roles. Puede dividirse en varios pasos.
	Definición de Producto de trabajo	Es el producto usado o producido por las <i>Tareas</i> . Existen dos tipos de productos: <i>Artefacto</i> de naturaleza tangible (modelo, documento, código, archivos, etc.) y <i>Entregable</i> para empaquetar productos con fines de entrega a un cliente interno o externo. Se pueden asociar entre ellos mediante relaciones de agregación, composición e impacto.
	Categoría	Clasifica elementos como <i>Tareas</i> , <i>Roles</i> y <i>Productos</i> en base a los criterios que desee el ingeniero de procesos. Existen diversos tipos de categorías: <i>Conjunto de Roles</i> (para <i>Roles</i>), <i>Disciplina</i> (para <i>Tareas</i>), <i>Dominio</i> (para <i>Productos</i>),

Icono	Nombre	Descripción
	Guías	Provee información adicional relacionada con otros elementos. Los sub-tipos de guías pueden ser (entre otros): <i>Activo Reutilizable, Directriz, Documentación, plantillas</i> . El icono presentado es genérico, pero se pueden emplear
	Uso de Rol	Representa al <i>Rol</i> que lleva a cabo una <i>Tarea</i> o <i>Actividad</i> dentro de un proceso determinado. Hace referencia a una Definición de Rol (elemento de Contenido).
	Uso de Tarea	Representa una <i>Tarea</i> atómica dentro de un proceso determinado. Hace referencia a una Definición de Tarea (elemento de Contenido).
	Uso de Producto de Trabajo	Representa un <i>Producto de Trabajo</i> de entrada o salida, relacionado con una <i>Actividad</i> o <i>Tarea</i> . Hace referencia a una Definición de un Producto de Trabajo (elemento de Contenido)
  	Actividad Fase Iteración	Representan a un conjunto de <i>Tareas</i> que se ejecutan dentro del proceso, junto con sus <i>Roles</i> y <i>Productos</i> asociados. Si únicamente se quiere representar una agrupación de tareas, se puede usar los elemento Actividad o Fase (incluido por retro-compatibilidad y más empleado en tareas de desarrollo), o bien si es un conjunto de tareas que se repite un determinado número de veces, se puede usar el elemento Iteración.
	Paquete de Proceso	Representa un paquete agrupando todos los elementos del proceso

El emplear un marco de trabajo como el ofrecido por SPEM 2 aporta numerosas ventajas, ya que se puede disponer de modelos de Procesos Software en formato procesable por computador, lo que a su vez, proporciona capacidades para:

- Facilitar la comprensión y comunicación entre las personas, debido a que se provee un marco de trabajo común donde los conceptos poseen una definición formal, propiciando de esta forma un conocimiento homogéneo.
- Facilitar la reutilización, debido a que la definición de un proceso se puede integrar como partes o patrones de otros modelos de proceso.
- Dar soporte a la mejora de procesos, ya que al definir formalmente las actividades junto con sus parámetros se facilita la evaluación de las mismas a través de procesos de medición, los cuales proveen retro-alimentación ayudando a mejorar dichos procesos.
- Dar soporte a la gestión de procesos, ya que al poseer un repositorio con todo el contenido del proceso, se facilita el acceso por las diferentes personas responsables del mismo atendiendo a sus roles.
- Guiar la automatización de procesos y dar soporte para la ejecución automática a través de la creación de flujos de trabajo (*workflows*) que pueden ser implementados en herramientas informáticas.

3.3.2. WUEP DEFINIDO EN SPEM 2

Puesto que una de las principales características de WUEP es su integración en el desarrollo Web dirigido por modelos, la presente sub-sección comienza especificando cómo sería un proceso de desarrollo en este ámbito. La Figura 3.4 define un proceso de desarrollo Web que refleja, de forma genérica, las fases más comunes que suelen incorporar la mayoría de los métodos de desarrollo Web que siguen el enfoque DSDM. En la figura se puede observar que las fases han sido representadas mediante “Tareas”, esto es debido a que no se entra en detalle en ellas, puesto que el objetivo es simplemente tener una noción de los roles que realizan tales fases junto a los productos que son consumidos y generados. El anexo C muestra la vista consolidada de este proceso, la cual ha sido obtenida como resultado de su definición en SPEM 2 utilizando la herramienta Eclipse Process Framework Composer.

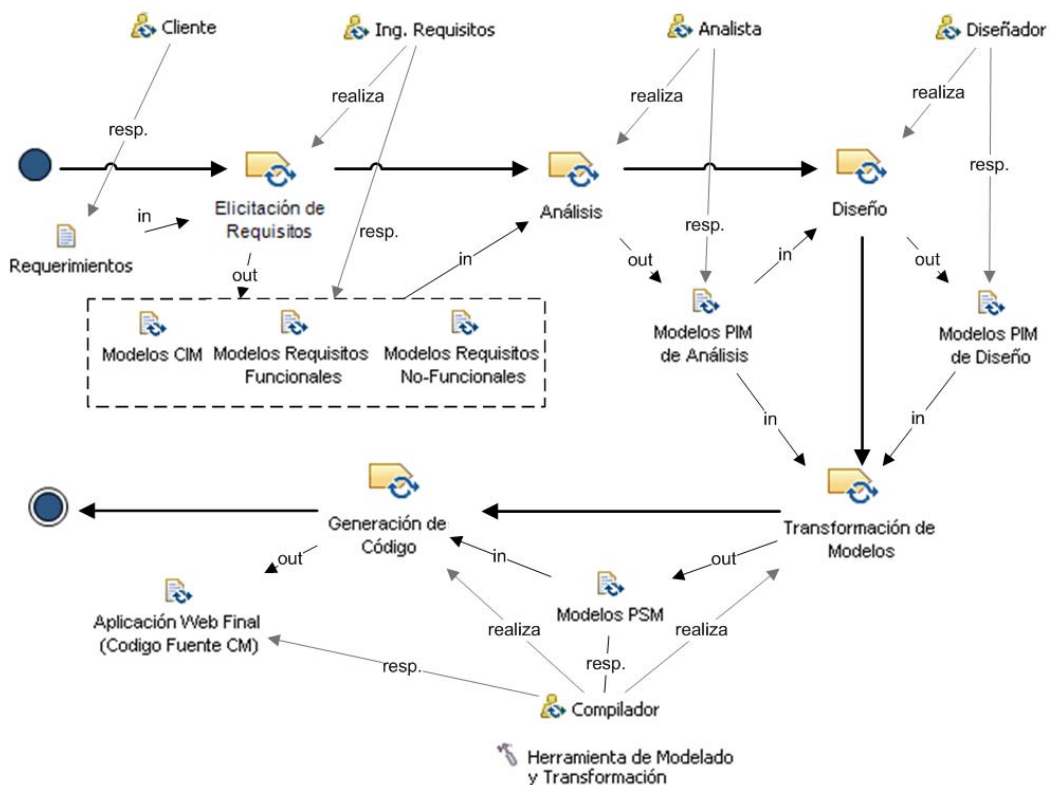


Fig 3.4. Proceso de desarrollo Web DSDM genérico en SPEM 2

Las fases más comunes de un proceso de desarrollo Web genérico basado en el enfoque DSDM son:

- **Elicitación de Requisitos:** donde el *Cliente* proporciona unos *Requerimientos* que sirven como entrada necesaria para que el *Ingeniero de Requisitos* elicite los *Requisitos funcionales* y *no-funcionales*, que contemplan las características que

ha de poseer de la aplicación Web; y se desarrollen los *Modelos Independientes de la Computación* (modelos CIM) que contemplan los procesos de negocio.

- **Análisis:** donde el *Analista* recibe como entrada los productos generados en la fase anterior (*Modelos de requisitos funcionales y no-funcionales*, y *Modelos CIM*) para comenzar con el análisis de la aplicación Web. En esta fase se define la estructura estática de la información y de qué forma el usuario puede navegar a través de dicha información. Estas dimensiones de la aplicación Web son reflejadas en *Modelos Independientes de Plataforma* (Modelos PIM), como por ejemplo, modelos de clases para la estructura estática, y modelos navegacionales para el acceso del usuario a la información.
- **Diseño:** donde el *Diseñador* recibe como entrada los *Modelos independientes de Plataforma* provenientes de la fase anterior para desarrollar las interfaces de usuario abstractas que especifican cómo esa información, y su acceso a ella, va a ser presentada al usuario de la aplicación Web. Estas dimensiones de la aplicación Web también son reflejadas en *Modelos Independientes de Plataforma* (Modelos PIM), como por ejemplo, modelos de diálogo, y modelos de presentación.
- **Transformación de modelos:** Donde el *Compilador de Modelos* (Rol llevado por un Herramienta de Modelado y Transformación) convierte, de forma automática, los modelos PIM de las fases de Análisis y Diseño en Modelos Específicos de Plataforma (modelos PSM) mediante reglas de transformación M2M (Modelo a Modelo). Los modelos PSM contienen los detalles de implementación asociados a una tecnología de desarrollo concreta, como por ejemplo, modelos de interfaz concretos, scripts de generación de bases de datos, etc.
- **Generación de código:** Donde el *Generador de Código* (Rol llevado también por la Herramienta de Modelado y Transformación) convierte, de forma automática, los modelos PSM de la fase anterior en el Código fuente final de la aplicación Web (Modelo CM) mediante reglas de transformación M2T (Modelo a Texto). El modelo CM hace referencia a la aplicación Web final, la cual ya puede ser ejecutada en la plataforma destino.

El proceso de evaluación de usabilidad WUEP ha sido definido tomando como punto de partida el proceso de evaluación de calidad propuesto en la división 2504n de la norma SQuaRE, el cual es prácticamente una revisión menor de la norma ISO 14598-1, teniendo en cuenta las fases y productos pertenecientes al proceso de desarrollo Web descrito anteriormente.

La división 2504n de la norma SQuaRE propone un método de evaluación de calidad basado en cuatro principales fases:

1. Establecimiento de los requisitos de evaluación: Donde se establece el propósito de la evaluación, se identifica el tipo de aplicación a evaluar y se propone un modelo de calidad acorde con el ámbito de la aplicación.
2. Especificación de la evaluación: Donde se seleccionan las métricas que se van a aplicar, se establecen umbrales para dichas métricas y se definen criterios de agregación para combinar los valores de las métricas.
3. Diseño de la evaluación: Donde se realiza un plan de evaluación que identifica todos los recursos asignados, las restricciones técnicas, y una agenda de tareas a realizar por los evaluadores destinados a llevar a cabo para la ejecución de la evaluación.
4. Ejecución de la evaluación: Donde se lleva a cabo el plan de evaluación diseñado, aplicando las métricas seleccionadas, ésta se comparan con los criterios definidos en la fase de especificación de la evaluación y se analizan los resultados obtenidos.

WUEP extiende y adapta las tareas que se realizan en las fases anteriores para su posible integración de evaluaciones de usabilidad Web en un entorno de desarrollo DSDM. La Figura 3.5 muestra una vista general de las principales fases que tienen lugar en el proceso y cuáles son los productos que comunican información al método de desarrollo Web, estableciendo de esta forma una integración entre ambos procesos. El anexo C muestra la vista consolidada de WUEP, la cual ha sido obtenida como resultado de su definición en SPEM 2 utilizando la herramienta Eclipse Process Framework Composer.

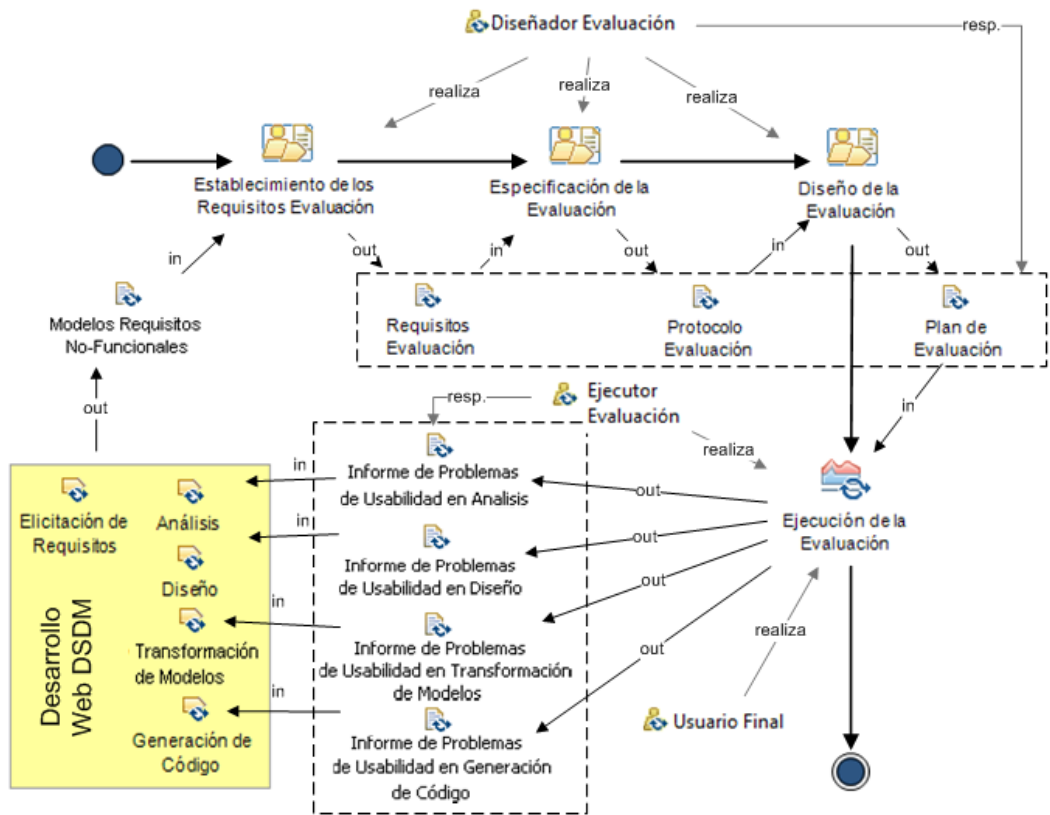


Fig 3.5. Proceso de evaluación de usabilidad WUEP en SPEM 2

Las siguientes sub-secciones presentan cada una de las cuatro fases anteriores, descomponiéndola en tareas más específicas y detalladas.

3.3.2.1. ESPECIFICACIÓN DE LOS REQUISITOS DE EVALUACIÓN

En esta fase se especifican los requisitos que ha de tener en cuenta el Diseñador de la evaluación para llevar a cabo la evaluación de usabilidad. El objetivo es centrar el enfoque de la evaluación de usabilidad atendiendo a todos los factores que condicionan la aplicación Web. La Figura 3.6 muestra cómo esta fase consta de cuatro tareas bien diferenciadas: establecimiento del propósito de evaluación; especificación de perfiles; selección de atributos de usabilidad; y elaboración del documento “Requisitos de evaluación”.

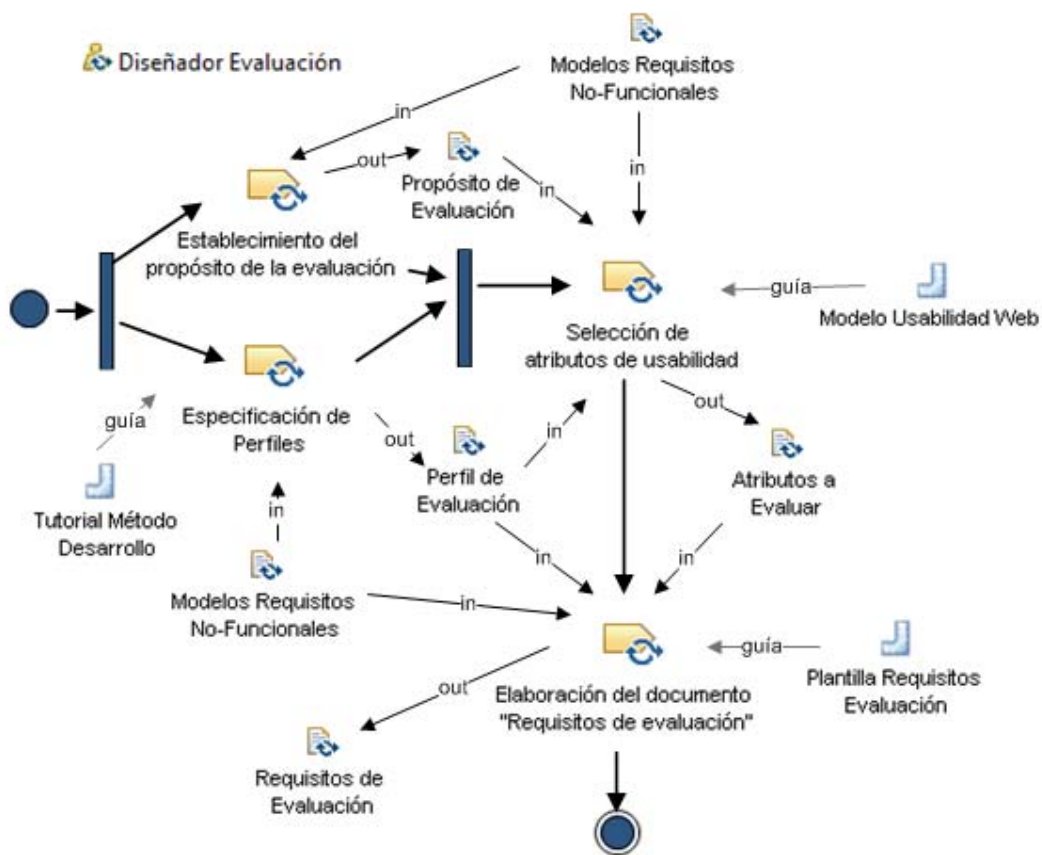


Fig 3.6. Fase “Especificación de Requisitos de Evaluación” de WUEP en SPEM 2

A continuación se detalla cada una de las cuatro tareas anteriores:

- **Establecimiento del propósito de la evaluación.** En esta tarea se debe determinar cuántas aplicaciones Web se van a evaluar y en qué momento se realizará la evaluación:
 - Mientras la aplicación Web está en proceso de desarrollo (evaluación de usabilidad formativa) si se pretende realizar una predicción de la usabilidad de la aplicación Web desde el punto de vista del producto software.
 - Cuando la aplicación Web ya está siendo usada por los usuarios finales en un determinado contexto de uso (evaluación de usabilidad sumativa) si se pretende evaluar la usabilidad en un contexto de uso determinado.
 - En ambos casos anteriores si se desea realizar una evaluación completa pudiendo comparar las predicciones de usabilidad en las fases tempranas con los resultados obtenidos a través del testeo con usuarios.

- **Especificación de perfiles.** En esta tarea se debe determinar los distintos factores que van a condicionar la evaluación:
 - *Tipo de aplicación Web:* Cada familia de aplicaciones Web posee ciertas características que las diferencian entre sí. Existe una gran variedad de tipos de aplicaciones web: intranets, redes sociales, gestores de contenido, portales de servicios, ofimáticas, etc. El tipo de aplicación Web condicionará que sub-características de usabilidad son más relevantes, por ejemplo, una intranet o portal que ponga a disposición del usuario una serie de servicios tendría más en cuenta la navegabilidad, mientras que una aplicación de acceso a una red social podría tener más en cuenta el grado de atracción para fidelizar usuarios.
 - *Método de desarrollo Web empleado:* Se debe tener un conocimiento sobre el método de desarrollo, principalmente, para poder establecer las relaciones entre los atributos de usabilidad que se pretenden evaluar y los artefactos intermedios propios del método que podrían ser evaluados. Este factor es de suma importancia cuando la evaluación de usabilidad es formativa.
 - *Contexto de uso:* Es necesario tener una descripción del contexto de uso donde la aplicación Web será empleada. Para ello se debe definir el perfil del usuario a la que va destinada (edad, implícitos culturales, idioma, etc.), los requisitos tecnológicos (sistemas operativos, dispositivos de acceso, terminales, etc.) y el entorno de trabajo.

- **Selección de atributos de usabilidad:** Empleando el Modelo de Usabilidad Web, descrito en secciones anteriores, como un artefacto de entrada a modo de catálogo, se seleccionarán que sub-características y atributos de usabilidad se desean evaluar. Esta decisión vendrá condicionada por los perfiles establecidos anteriormente.

- **Elaboración del documento “Requisitos de evaluación”:** reuniendo toda la información anterior se elaborará un documento que refleje todos los requisitos de la evaluación. Para ello se podrán utilizar plantillas, que básicamente contendrán como apartados los aspectos descritos en las tareas anteriores.

3.3.2.2. ESPECIFICACIÓN DE LA EVALUACIÓN

En esta fase se especifica en detalle la evaluación que se va a realizar, es decir, qué artefactos se tiene previsto evaluar, qué y cómo se van a aplicar las métricas, qué umbrales o indicadores se considerarán adecuados para los resultados obtenidos, cómo se combinan los resultados anteriores, y cómo se reportan los problemas de usabilidad detectados. La Figura 3.7 muestra cómo esta fase consta de siete tareas bien

diferenciadas: selección de artefactos a evaluar; selección de métricas a emplear; operacionalización de las métricas; establecimiento de umbrales e indicadores; establecimiento de criterios de normalización y agregación; definición de la plantilla de informe de usabilidad; y elaboración del documento “Especificación de la evaluación”.

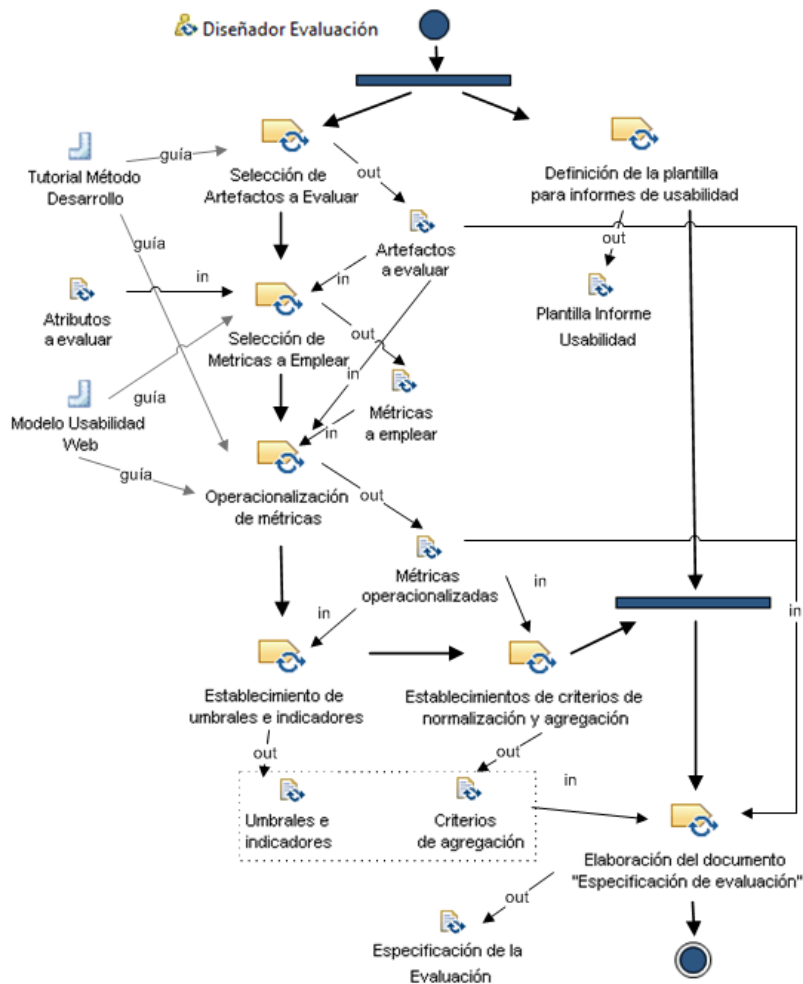


Fig 3.7. Fase “Especificación de la Evaluación” de WUEP en SPEM 2

A continuación se detalla cada una de las siete tareas anteriores:

- **Selección de artefactos a evaluar:** Los artefactos a evaluar pueden ser dependientes tanto del método de desarrollo Web empleado, como de la plataforma tecnológica destino. Los artefactos que se pueden considerar a evaluar podría ser:
 - o Los modelos independientes de plataforma (PIM) tales como: modelos estructurales, modelos navegacionales, modelos de interfaz abstractos, modelos de diálogo, etc. (Cada método tiene sus propios modelos, para

especificar una aplicación Web, que son conformes a distintos meta-modelos).

- Los modelos específicos de plataforma (PSM) tales como: modelos de arquitecturas, modelos concretos de interfaces de usuario, etc. (Cada plataforma tecnológica de desarrollo tiene sus propios modelos).
- Los modelos de código (CM) tales como el código fuente o las interfaces de usuario finales de la aplicación Web.
- La interacción del usuario final con respecto a la aplicación Web en uso.

Cabe resaltar que no se evalúan, en sí mismo, los modelos resultantes de la fase de elicitación de Requisitos, tales como modelos independientes de computación (CIM) y modelos de requisitos. El motivo reside en que estos modelos están más orientados a una perspectiva de negocio, que a una perspectiva del producto software capaz de evaluarse con el Modelo de Usabilidad Web. No obstante, estos modelos contribuyen a la evaluación de usabilidad guiándolo a través de los requisitos no funcionales.

- **Selección de métricas a emplear:** A partir de los atributos de usabilidad y los artefactos seleccionados, podemos emplear el Modelo de Usabilidad Web para obtener qué métricas están asociadas a dichos atributos. Una vez obtenidas las métricas, se consulta su definición genérica para conocer qué aspectos pretende medir y a qué niveles de abstracción podrían evaluarse. De esta forma, podemos verificar si existe una relación entre la definición genérica de la métrica y los conceptos que representa el artefacto.
- **Operacionalización de las métricas:** Para aquellas métricas que han sido vinculadas a los artefactos a evaluar, se ha de establecer un mapeo entre la descripción que plantea la métrica y los elementos propios del artefacto a evaluar. El objetivo de este mapeo es ofrecer una operacionalización de la fórmula de cálculo de la descripción genérica que pueda ser aplicable a los elementos propios pertenecientes al artefacto a evaluar. En el caso de la evaluación de modelos PIM, PSM o CM, estas fórmulas podrían requerir la ayuda de un evaluador para determinar los valores de la variables que participan en ella, o bien de una herramienta de verificación de modelos si las fórmulas se expresan en variables que pueden ser calculadas automáticamente a partir de los modelos de entrada y un lenguaje que permita acceder a los elementos del modelo. Además, podría ocurrir que no pudiera establecerse ningún mapeo debido a la expresividad de los modelos del método en concreto, en este supuesto, esta tarea ayudaría a establecer las limitaciones de los modelos a fin de mejorar su expresividad de cara a tener en cuenta atributos de usabilidad. En

el caso de la evaluación en uso, las métricas necesitan ser calculadas a partir de los resultados obtenidos durante, o tras, la interacción del usuario final, por lo que la operacionalización de estas métricas requiere definir claramente las tareas que va a llevar a cabo el usuario final y que datos se van a recoger de estas tareas. Estos datos pueden requerir de un proceso de transformación y/o interpretación para obtener el valor de las variables que participan en la fórmula de la métrica.

- **Establecimiento de umbrales e indicadores:** Esta tarea es relevante para realizar una evaluación cualitativa. Para cada métrica se establecen los intervalos de valores que representan si los resultados de la métrica permiten conocer si su atributo asociado contribuye a mejorar el grado de usabilidad (o no) de la aplicación Web. Gracias a estos intervalos de valores se podrán detectar qué problemas de usabilidad existen y clasificarlos según su nivel de criticidad. La forma de proceder es dividir en intervalos todos los valores posibles que la métrica ofrece y asignar a cada intervalo un indicador del nivel de criticidad. Por ejemplo, si el rango de valores que ofrece una métrica es un número real entre 0 y 1 (siendo '1' el valor que contribuye a que su atributo asociado mejore el grado de usabilidad), siendo X el valor obtenido de una métrica, un posible establecimiento de umbrales e indicadores podría ser el siguiente: Si $[0 \leq x \leq 0.5]$ existe problema de usabilidad grave; si $[0.5 \leq x \leq 0.7]$ existe un problema de usabilidad medio; si $[0.7 \leq x \leq 0.9]$ existe un problema de usabilidad leve, y por último, si $[0.9 \leq x \leq 1]$ no se considera problema de usabilidad alguno.
- **Establecimientos de criterios de normalización y agregación:** Esta tarea consiste en establecer una evaluación cuantitativa, principalmente para comparar distintas aplicaciones Web en base a atributos de usabilidad. Para cada sub-característica y atributo seleccionados del Modelo de Usabilidad Web se puede definir una fórmula que combina los valores obtenidos (numéricos o no numéricos) de distintas métricas para obtener un indicador que cuantifique el grado de contribución a la usabilidad de la aplicación Web. En el caso de atributos con más de una métrica asociada, se pueden emplear expresiones que asignen pesos para ponderar los valores numéricos, o bien reglas de combinación para valores no numéricos. Para las combinaciones de métricas suele ser necesaria una normalización previa de los valores si sus tipos de escalas son distintas. Una vez los atributos tienen asignado su valor, el mismo proceso se puede repetir con las sub-características que tienen asociados distintos atributos.

- **Definición de la plantilla para informes de usabilidad:** Para presentar los problemas de usabilidad detectados se deben detallar para facilitar el hecho de emprender acciones que permitan solucionarlos. En esta tarea se debe decidir cuál será la estructura de los informes de usabilidad. En este caso, un informe de usabilidad es una lista de problemas de usabilidad detectados. Para cada problema de usabilidad se deberían incluir al menos los siguientes campos descriptivos:
 - *Identificador:* Código que identifica unívocamente al problema.
 - *Descripción:* Descripción textual del problema detectado.
 - *Atributo afectado:* Atributo de usabilidad que se ve afectado por este problema, indicando las sub-características de las que proviene.
 - *Nivel de criticidad:* Si se trata de un problema grave, medio o leve.
 - *Artefacto evaluado:* Artefacto donde se ha aplicado la métrica. (Modelos en el caso de la evaluación formativa, y Tareas en el caso de la evaluación sumativa)
 - *Origen del problema:* Hace referencia a los artefactos que han originado dicho problema, el cual no tiene por qué ser necesariamente el artefacto evaluado. En el caso de la evaluación de modelos PIM, los problemas sí que suelen tener su origen en los mismos modelos, sin embargo, en el resto de casos (evaluación de modelos PSM, CM, o en uso) los problemas pueden provenir de artefactos intermedios de fases anteriores. Estos artefactos pueden ser otros modelos, o bien, las reglas de transformación que se han aplicado anteriormente para obtener el artefacto que se ha evaluado.
 - *Ocurrencias:* Número de veces que el problema de usabilidad se repite.
 - *Recomendaciones:* Sugerencias dadas por el evaluador que permiten prevenir y/o corregir dichos problemas (algunas sugerencias podrían generarse automáticamente a partir de la interpretación de los umbrales). El desarrollador también puede completar este campo aportando nuevas recomendaciones si la evaluación no es capaz de ofrecer alternativas.

Además puede ser interesante incorporar otros campos tales como:

- *Impacto:* Grado en el que la mejora del atributo podría afectar (negativa o positivamente) a otros atributos evaluados.
- *Prioridad:* Importancia que se le otorga al problema a la hora de corregir el problema de usabilidad (Suele decidirlo el desarrollador).
- *Esfuerzo:* recursos que se deberían invertir para poder corregir el problema de usabilidad (Suele decidirlo el desarrollador).

- *Cambios a realizar*: Basándose en la prioridad, esfuerzo e impacto asociados a las recomendaciones propuestas, se decide qué cambios son factibles por los desarrolladores (Suele decidirlo el desarrollador).
- **Elaboración del documento “Especificación de evaluación”**: reuniendo toda la información anterior se elaborará un documento que refleje todos los detalles de la evaluación, anexando además los requisitos de la evaluación obtenidos como resultado en la primera fase. Para ello se podrán utilizar plantillas, que básicamente contendrán como apartados los aspectos descritos en las tareas anteriores.

3.3.2.3. DISEÑO DE LA EVALUACIÓN

En esta fase ya se posee la información necesaria para establecer un plan de evaluación, en el cual se detallan los aspectos más técnicos sobre la evaluación, cómo la agenda y los tiempos programados para las ejecuciones; todo tipo de restricciones que condicionen la evaluación, ya sean tecnológicas, económicas o de cualquier naturaleza, o la organización referente a las tareas de usuario (en el caso de una evaluación de la aplicación Web en uso). La Figura 3.8 muestra cómo esta fase consta de dos tareas bien diferenciadas: definición de restricciones; y elaboración del plan de evaluación.

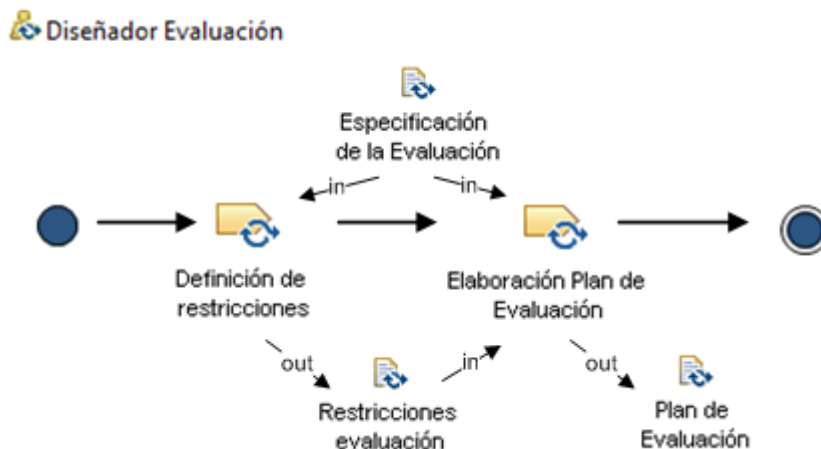


Fig 3.8. Fase “Diseño de la Evaluación” de WUEP en SPEM 2

A continuación se detalla cada una de las cuatro tareas anteriores:

- **Definición de restricciones**: donde se deja patente de las restricciones anteriores.
- **Elaboración del plan de evaluación**: Donde en base a todos los artefactos que se pretende evaluar, se establece un orden de evaluación y se asignan tareas a los

encargados de ejecutar la evaluación. El orden recomendado es comenzar con los modelos pertenecientes a un nivel mayor de abstracción (modelos PIM) puesto que estos modelos dirigen en su totalidad la construcción de la aplicación Web final. De esta forma, se pueden obtener los primeros resultados de usabilidad en fases tempranas del proceso de desarrollo, permitiendo además, evaluar de forma iterativa esos modelos hasta que los atributos obtengan valores que contribuyan a mejorar el grado de usabilidad. Continuando después con los modelos de menor nivel de abstracción (PSM y CM) hasta llegar a la evaluación en uso. Se debe tener en cuenta además, que si estas últimas evaluaciones proporcionan cambios en modelos PIM, éstos podrían evaluarse de nuevo.

3.3.2.4. EJECUCIÓN DE LA EVALUACIÓN

En esta fase se recibe la especificación y el plan de evaluación, pudiendo proceder a la aplicación de las métricas ya operacionalizadas para evaluar los artefactos propuestos y obtener los resultados que permitan rellenar los informes de usabilidad correspondientes. Se incluye además, un análisis de los cambios propuestos por estos informes. Es importante remarcar, tal y cómo se apuntaba en el plan de evaluación, que esta fase es una iteración, ya que tras efectuar los cambios propuestos por los informes de usabilidad, podría volverse a repetir el proceso de evaluación de artefactos. La Figura 3.9 muestra cómo esta fase consta de dos tareas bien diferenciadas: evaluación de los artefactos (a distintos niveles) y análisis de cambios.

A continuación se detalla cada una de las dos tareas anteriores:

- **Evaluación de los artefactos:** En esta tarea se aplican las métricas operacionalizadas sobre los artefactos que se seleccionaron. Tras obtener los valores de dichas métricas, se determina el nivel de criticidad del problema de usabilidad en base a los umbrales establecidos. Observando los elementos del artefacto, implicados en el cálculo de la métrica y que han provocado ese problema, se puede determinar el origen de esos elementos gracias a la trazabilidad entre modelos existente en entornos de desarrollo dirigido por modelos. Con la información obtenida, el ejecutor de la evaluación puede rellenar el informe de usabilidad correspondiente, que es básicamente la colección de problemas de usabilidad detectados y documentados conforme a la plantilla elaborada en la fase de Especificación. En estos informes se deben, rellenar todos los campos excepto aquellos que van dirigidos a los desarrolladores, como por ejemplo, la prioridad, el esfuerzo y recursos necesarios para realizar corregir los problemas. Las evaluaciones puede ser a cuatro niveles de abstracción:

- *Modelos PIM*: Donde el informe que se obtiene recibe el nombre de *informe de problemas de usabilidad independiente de plataforma*. Estos problemas tendrán su origen en modelos PIM creados, bien en la fase de análisis, o bien en la fase de Diseño. De esta forma, la retro-alimentación producida en esta evaluación irá orientada hacia el Analista y el Diseñador.
- *Modelos PSM*: Donde el informe que se obtiene recibe el nombre de *informe de problemas de usabilidad específico de plataforma*. Estos problemas podrán tener su origen en modelos PIM, en los propios modelos PSM que se evalúan, o bien en las reglas de transformación entre ambos. De esta forma, la retro-alimentación producida en esta evaluación podrá ir orientada hacia el Analista y el Diseñador (Modelos PIM) o hacia el responsable del compilador de modelos (Modelos PSM o reglas de transformación).
- *Aplicación Web final (CM)*: Donde el informe que se obtiene recibe el nombre de *informe de problemas de usabilidad aplicación Web final*. Estos problemas podrán tener su origen en modelos PIM, en los propios modelos PSM que se evalúan, en las reglas de transformación entre ambos, o bien en las reglas de transformación que se aplican a los PSM para obtener el código fuente. De esta forma, la retro-alimentación producida en esta evaluación podrá ir orientada hacia el Analista y el Diseñador (Modelos PIM) o hacia el responsable del compilador de modelos (Modelos PSM o reglas de transformación).
- *En Uso*: Donde el informe que se obtiene recibe el nombre de *informe de problemas de usabilidad en uso*. El origen de estos problemas es análogo a los de la evaluación de la aplicación Web final.

En las evaluaciones anteriores, para cada problema se rellenan todos los campos que propone la plantilla del informe, exceptuando aquellos que necesitan ser rellenos por el desarrollador (analista o diseñador), como el esfuerzo necesario para corregirlos, los recursos a emplear.

- **Análisis de cambios**: A partir de los informes de usabilidad anteriores, se clasifican los problemas según su origen y se ordenan según su criticidad. Esta clasificación pasará a componer los *informes de problemas de usabilidad* para cada una de las fases del desarrollo Web: Análisis, Diseño, Transformación de modelos y Generación de código. Los problemas de usabilidad contenidos en estos informes serán valorados por los desarrolladores.

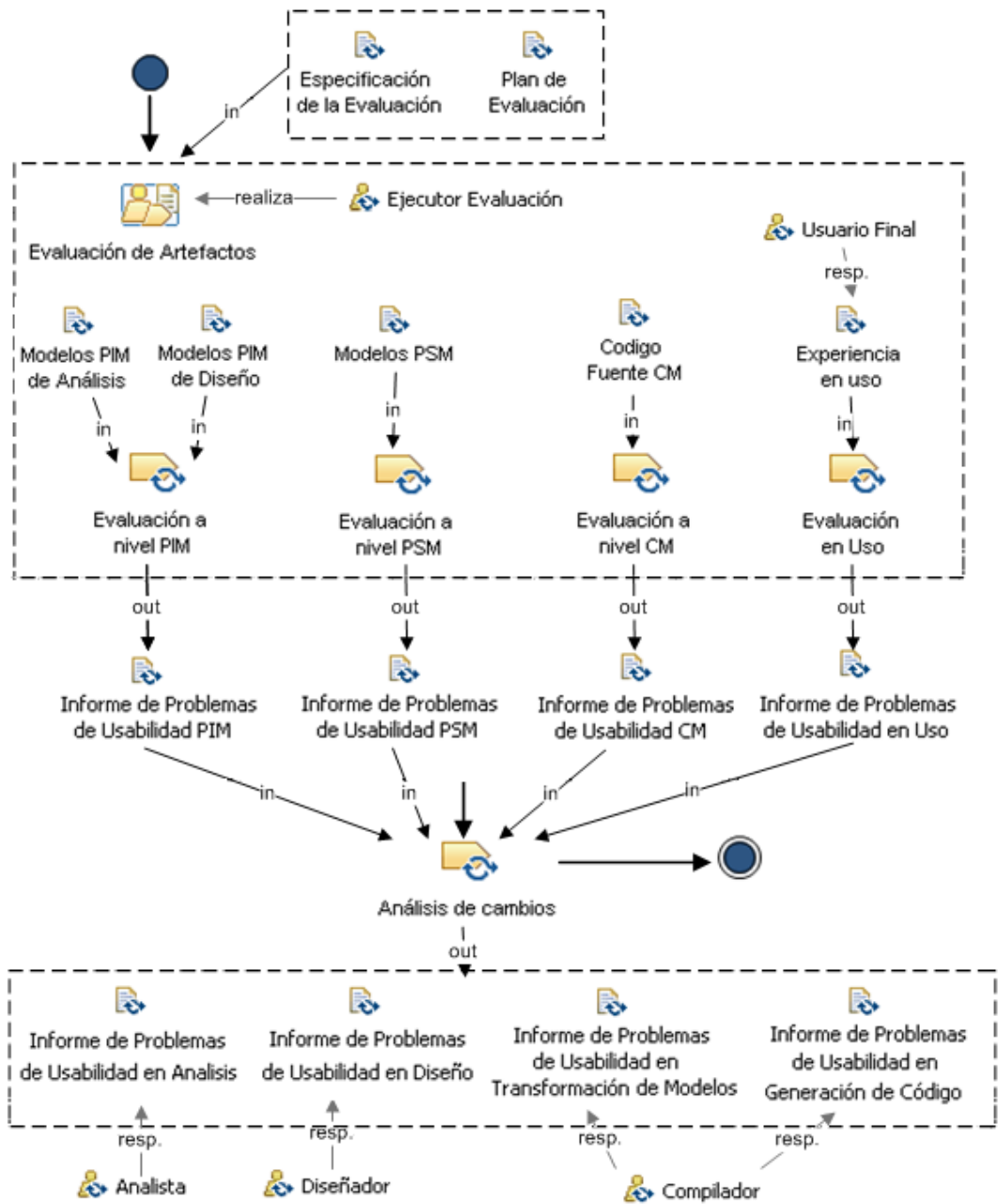


Fig 3.9. Fase "Ejecución de la Evaluación" de WUEP en SPeM 2

4. INSTANCIACIÓN DE WUEP EN UN MÉTODO DE DESARROLLO WEB CONCRETO

En este capítulo se presenta como WUEP podría ser instanciado, a modo de ejemplo, para la evaluación de una aplicación Web desarrollada empleando un método de desarrollo Web dirigido por modelos concreto. El objetivo es mostrar la factibilidad del proceso de evaluación WUEP, proveer un ejemplo que ayude a clarificar las tareas definidas, y obtener una serie de lecciones aprendidas que ayuden en la mejora, no solo del proceso de evaluación, sino que también proporcionen información acerca de cómo mejorar la expresividad de los modelos de un método de desarrollo Web.

Para este capítulo se ha empleado como aplicación Web a evaluar, un gestor de tareas (Task Manager) que ha sido desarrollado siguiendo el método de desarrollo Web Object-Oriented Hypermedia [28] (OO-H) y empleando la herramienta VisualWade², la cual ofrece soporte completo para la edición y compilación de modelos propuestos por el método.

La Sección 4.1 introduce brevemente el método de desarrollo Web, ofreciendo una perspectiva general y resaltando qué tipos de modelos concretos para el ámbito Web se emplean y sus principales primitivas de modelado.

La Sección 4.2 presenta la aplicación Web a evaluar, incluyendo una breve explicación sobre su funcionalidad y modelos OO-H que la especifican. Esta aplicación fue desarrollada empleando VisualWade y fue proporcionada por una empresa de desarrollo Web ubicada en Alicante.

La sección 4.3 hará uso del contenido de las secciones anteriores y de la definición del proceso de evaluación de usabilidad WUEP (ver Sección 3.3) para instanciar el proceso sobre el ejemplo del caso de estudio, siguiendo para ello las fases establecidas.

Por último, la sección 4.4 recoge las lecciones aprendidas tras la aplicación del WUEP.

4.1. INTRODUCCIÓN AL MÉTODO OO-H

OO-H (Object-Oriented Hypermedia) [28] es un método de desarrollo Web dirigido por modelos, basado en el paradigma orientado a objetos, que proporciona la semántica y la notación necesaria para especificar aplicaciones basadas en entornos Web. La Figura 4.1 muestra un esquema del proceso planteado por el método donde participan los siguientes modelos propuestos:

² Disponible en <http://www.visualwade.com>

- **Modelo de casos de uso:** Conjunto de diagramas que representan las funcionalidades de la aplicación Web para los distintos usuarios posibles. Nota: No se proveen mecanismos para especificar requisitos no funcionales.
- **Modelo de clases:** Diagrama que representa la estructura conceptual estática del dominio de la aplicación Web. (Análogo a un modelo de clases UML). Está compuesto por clases, atributos, métodos y relaciones entre clases. Además, permite la integración de fórmulas OCL para definir restricciones adicionales o derivaciones de atributos.
- **Modelo navegacional:** Conjunto de Diagramas de Acceso Navegacional (*Navigational Access Diagram – NAD*) que capturan las propiedades de navegación e interacción de la aplicación Web. Cada NAD es una vista parcial del modelo de clases para un tipo específico de usuario. Además, permite la integración de fórmulas OCL para definir restricciones y filtros de navegación.
- **Modelo de presentación:** Conjunto de Diagramas de Presentación Abstracta (*Abstract Presentation Diagram - APD*) que representa la interfaz de usuario abstracta. Cada APD es una colección de páginas abstractas trazables a partir cada NAD. Posteriormente, el APD se refina para enriquecer la interacción del usuario incluyendo otros aspectos más relacionados con la interfaz de usuario.

Cabe resaltar que en éste método sigue una visión *traduccionista* del desarrollo dirigido por modelos [57], es decir, sólo participan explícitamente los modelos CIM (Modelo casos de uso) y los modelos PIM (Modelo de clases, Modelo navegacional y Modelo de presentación). Los modelos PIM sirven de entrada a un compilador de modelos que obtiene el código fuente de la aplicación Web final. De esta forma, los modelos específicos de plataforma (PSM) quedan embebidos en la definición de dicho compilador, en contraste con las visiones *elaboracionistas* del desarrollo dirigido por modelos donde los PSM participan explícitamente.

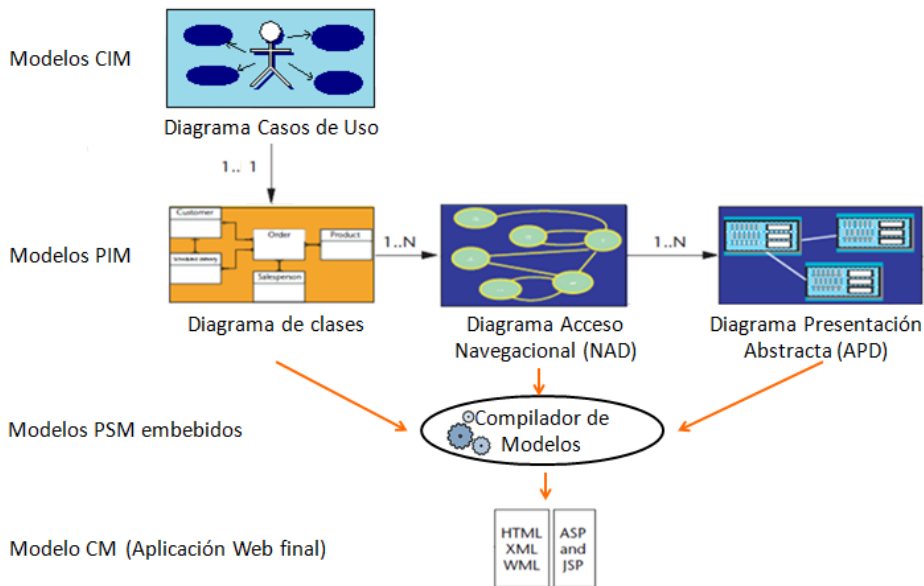


Fig 4.1. Proceso de desarrollo Web de OO-H

A continuación, se detallan algunas de las primitivas de modelado pertenecientes a los Modelos PIM más importantes de OO-H: Modelo Navegacional y Modelo de Presentación Abstracta.

El Modelo Navegacional está definido por un conjunto de Diagramas de Acceso Navegacional (NAD). Un NAD está basado en cinco tipos de constructores (ver Fig.4.2):

- **Clases Navegacionales:** Son vistas parciales de las clases pertenecientes al Modelo de clases, donde la visibilidad de los atributos y los métodos se restringe de acuerdo con los permisos de cada tipo de usuario y los requisitos de navegación. En estas Clases Navegacionales se distinguen tres tipos de atributos:
 - o *Atributos visibles:* Siempre visibles por el usuario.
 - o *Atributos referenciados:* Disponibles bajo demanda del usuario.
 - o *Atributos ocultos:* Empleados con fines de depuración.
- **Destinos Navegacionales:** agrupa todo tipo de elementos (Clases Navegacionales, Enlaces Navegacionales, Colecciones) que colaboran cubriendo los requisitos funcionales de un usuario.
- **Enlaces Navegacionales:** Definen las rutas de navegación que el usuario es capaz de seguir. Estos pueden tener asociados patrones de navegación y/o un conjunto de filtros. Existen seis tipos de Enlaces Navegacionales:
 - o *Enlaces internos (LI):* Muestran una ruta de navegación entre Clases Navegacionales pertenecientes al mismo Destino Navegacional.

- *Enlaces transversales (LT)*: Muestran una ruta de navegación entre Clases Navegacionales pertenecientes a distintos Destinos Navegacionales
- *Enlaces de requisitos (LR)*: Muestran el punto de entrada del usuario para comenzar la navegación.
- *Enlaces de salida (LX)*: Muestran los puntos de salida de la aplicación hacia otros sistemas.
- *Enlaces de servicio (LS)*: Muestran los servicios disponibles para el tipo de usuario asociado al NAD. Hacen referencia a los métodos de las clases pertenecientes al Modelo de Clases, los cuales se han de invocar con los parámetros correspondientes.
- *Enlaces de respuesta (LResp)*: Muestran la vista de retorno tras un Enlace de servicio.

Los Enlaces Navegacionales poseen atributos que condicionan su comportamiento. Entre ellos se destacan los siguientes:

- *Visualización*: Puede tomar dos valores posibles: *Mostrar en origen* o *Mostrar en destino*. Asumiendo que cada enlace navegacional conecta un origen con un conjunto de información destino, cuando el atributo toma el valor *Mostrar en origen* el conjunto de información destino se visualiza conjuntamente con el de origen, es decir, en la misma página abstracta. Mientras que si el atributo toma el valor *Mostrar en destino*, el conjunto de información destino se visualiza en otra página abstracta (ver Fig. 4.2 para ambas representaciones gráficas).
 - *Interacción*: Puede tomar dos valores posibles: *Manual* o *Automático*. Un enlace navegacional *automático* indica que el destino se alcanza de forma automática sin la intervención del usuario. Por el contrario, *manual* indica que el usuario debe explícitamente intervenir para alcanzar dicho destino. (ver Fig. 4.2 para ambas representaciones gráficas).
- **Colecciones**: Se corresponden con estructuras jerárquicas que agrupan un conjunto de Enlaces Navegacionales proporcionando nuevas rutas al usuario para acceder a la información. Básicamente es un mecanismo de abstracción del concepto menú.
 - **Etiquetas**: Son nodos especiales que muestran información derivada de otros constructores a través de fórmulas OCL.

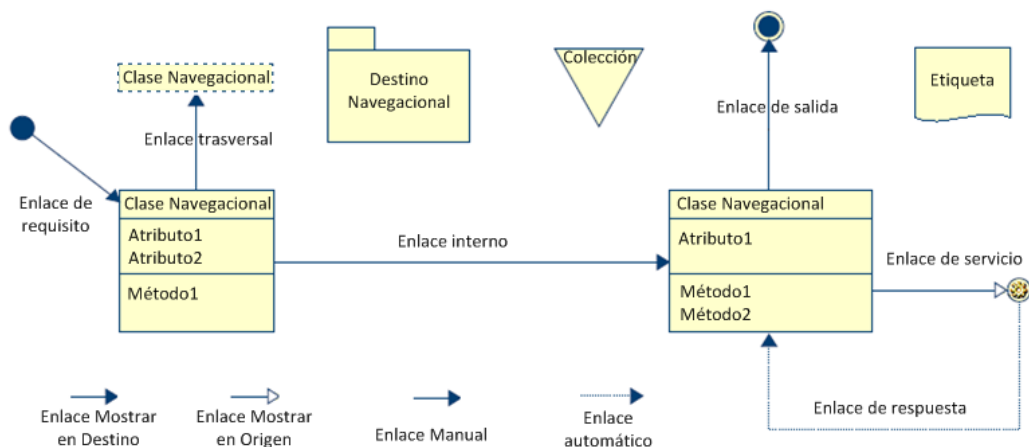


Fig 4.2. Elementos constructores de los NAD en OO-H

Además, para los NADs se incluyen filtros y patrones de navegación. Los filtros de navegación se expresan mediante restricciones OCL y contribuyen a restringir la vista del usuario según sus requerimientos de navegación. Los patrones de navegación proveen las características de Indexación y Navegación. Si se activa la indexación, la información puede ser mostrada en distintas paginas para mejorar su visualización. Si se activa la Navegación, se muestra el número de elementos por página para guiar al usuario.

El Modelo de Presentación está definido por un conjunto de Diagramas de Presentación Abstracta (APD). Cada APD provee mecanismos para refinar las interfaces de usuario a un nivel de abstracción más cercano al diseño gráfico. A partir de un NAD se puede obtener una primera versión muy simple de un APD, reflejando la estructura de páginas abstractas que se derivan del NAD. El APD separa las diferentes características que contribuyen a la apariencia y comportamiento de la interfaz de usuario final empleando una taxonomía basada en el concepto de plantillas y expresada en formato XML (Extensible Markup Language). Estos documentos XML son los siguientes:

- **Tstruct:** Se usa para capturar la información que se ha de mostrar.
- **Tform:** Se usa cuando la página abstracta, a parte de la información, incluye llamadas a la lógica subyacente.
- **Tlink:** Se usa para capturar la interconexión y dependencia entre páginas.
- **Tfunction:** Se usa para capturar la funcionalidad del cliente usada en las diferentes páginas.
- **Texternal:** Se usa para capturar el tipo, posición y comportamiento de elementos externos, como por ejemplo, imágenes o applets.
- **Tlayout:** Se usa para capturar las posiciones de los elementos y la definición de vistas simultaneas junto con su sincronización.

- **Tstyle:** Se usa para capturar aspectos tales como la tipografía o paletas de color para cada elemento de la interfaz de usuario.

Puesto que es muy difícil y poco intuitivo trabajar con ficheros XML, la herramienta VisualWade ofrece mecanismos e interfaces más intuitivas para el diseñador, traduciéndose esas acciones en las modificaciones pertinentes de los documentos XML.

4.2. CASO DE ESTUDIO: GESTOR DE TAREAS

La aplicación Web a utilizar como caso de estudio es un gestor de tareas (*Task Manager*) desarrollada para el control y seguimiento de proyectos de desarrollo Web. Un proyecto de desarrollo Web implica una serie de tareas (*Tasks*) que se asignan a un usuario (*User*), que es un programador de la empresa. En cada tarea se registra la fecha de inicio, la estimación de la fecha de finalización, prioridad, etc. El director del proyecto (*Admin*) organiza las tareas en carpetas (*Folders*) de acuerdo a ciertos criterios: las tareas pendientes, tareas críticas, etc. Además, se pueden asociar archivos externos (*files*) a una tarea (por ejemplo, documentos de requisitos, código, etc.). Los programadores pueden también incorporar comentarios (*Comments*) a las tareas y enviar mensajes (*Messages*) a otros programadores. Todos los días laborables, los programadores pueden generar un informe (*Daily Report*), incluyendo la información relacionada con las tareas en las que están trabajando. Por último, los clientes del proyecto se registran como Contactos (*Contacts*) en la aplicación.

Cada una de las siguientes sub-secciones describe los modelos que representan esta aplicación Web.

4.2.1. MODELO DE CASOS DE USO

La Figura 4.3 presenta el Modelo de casos de uso del Gestor de Tareas. Fue el único Modelo Independiente de la computación (CIM) que fue suministrado, el cual sirvió como base para elaborar los Modelos Independientes de Plataforma de análisis (PIM).

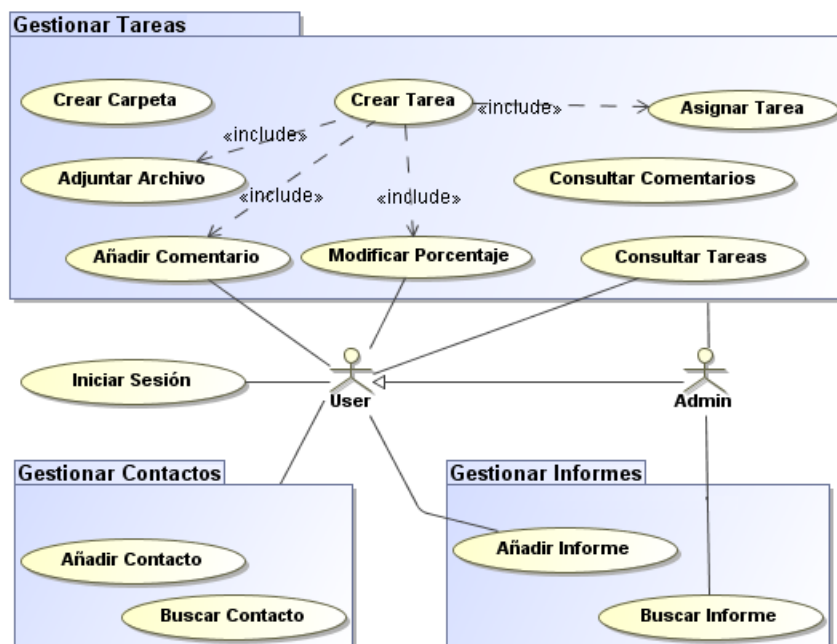


Fig 4.3. Modelo de casos de uso para TaskManager

En la Figura 4.3 se puede apreciar que existen dos tipos de usuario: Programador (*User*) y Director (*Admin*). Básicamente, el actor *Admin* puede realizar todas las funcionalidades de la aplicación Web exceptuando “Añadir un Informe”. Mientras que la funcionalidad del actor *User* se limita a la gestión de contactos, crear un informe diario, y a consultar sus propias tareas, pudiendo únicamente añadir comentarios a la tarea o modificar el porcentaje que lleva realizado para cada tarea que le han asignado. La herramienta VisualWade no da soporte a este modelo.

4.2.2. MODELO DE CLASES

La Figura 4.4 presenta el modelo de clases. En él se representan los conceptos del dominio de la aplicación Web y las asociaciones que se establecen entre ellos. Este primer modelo PIM toma como base los requerimientos del cliente y el modelo de casos de uso.

Como se puede observar en la figura, cada concepto se corresponde con una clase, y ésta a su vez contiene un conjunto de atributos y métodos. Cada clase tiene un atributo que identifica las instancias creadas (atributos con el icono de la llave), atributos que mantienen la información deseada y atributos derivados cuyo valor se obtiene a partir de otros (atributos con la barra '/'). Los métodos de las clases van destinados a crear/eliminar instancias o modificar atributos de la misma. A su vez, los atributos poseen propiedades que ayudan a definirlo: descripción, tipo de datos, valor por defecto, etc.

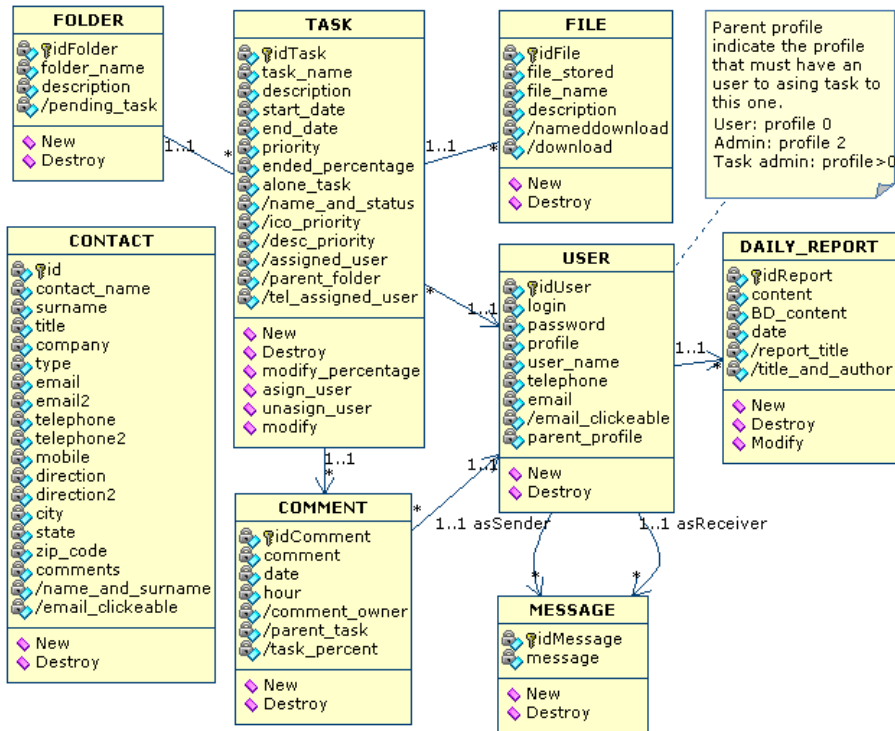


Fig 4.4. Modelo de clases para TaskManager

Es importante resaltar que ambos tipos de usuarios han sido representados por única clase diferenciados por el atributo *profile*, de tal forma que de cara a los modelos navegacionales solo se considerará un tipo de usuario (añadiendo restricciones OCL para diferenciar la información visible y caminos posibles del programador o usuario y el director o administrador). La herramienta VisualWade soporta estos modelos, permitiendo asignar propiedades de visibilidad o valores por defecto a los atributos. En el caso de los métodos permite especificar los parámetros necesarios.

4.2.3. MODELO DE NAVEGACIÓN Y PRESENTACIÓN

Como se ha podido observar en la introducción al método OO-H, los modelos de navegación y presentación están estrechamente relacionados. Por este motivo, en esta sub-sección se presentan los Diagramas de Acceso Navegacional (NADs) que componen el modelo de navegación junto con los Diagramas de Presentación Abstracta (APDs) que han sido refinados a partir de una primera versión obtenida directamente de cada NAD. El modelo de navegación consta de cuatro NADs: Uno para representar el acceso al Home principal, y tres para representar las funcionalidades de la aplicación Web, mientras que el modelo de presentación consta también de cuatro APDs, uno por cada NAD.

La Figura 4.5 representa el NAD de primer nivel (NAD-0) donde se representa el acceso de un programador o director (*User/Admin*) mediante el enlace de requisito *Entry point User*. La colección *Home* incluye al enlace interno *autenticar* con un filtro (que se apoya en la información de la clase navegacional *user*) el cual permite iniciar sesión al usuario si sus credenciales *user_name* y *password* son correctas. Si son correctas (restricción que se refleja mediante una precondición asociada al enlace navegacional automático *LI4*) se alcanza la colección *home restringido*, la cual incluye los enlaces a cada uno de los cuatro destinos navegacionales posibles: *Projects*, *Contacts*, *Reports* y *Notes*. (El enlace automático *LI63* ofrece que el destino navegacional *Projects* sea el que se muestre por defecto al iniciar sesión). Si por el contrario, las credenciales no son válidas, un enlace interno automático (*LI2*) accede a la colección *Error*, donde se muestra el mensaje de error pertinente, ofreciendo además, la opción de volver al formulario inicial mediante otro enlace (*LI6*).

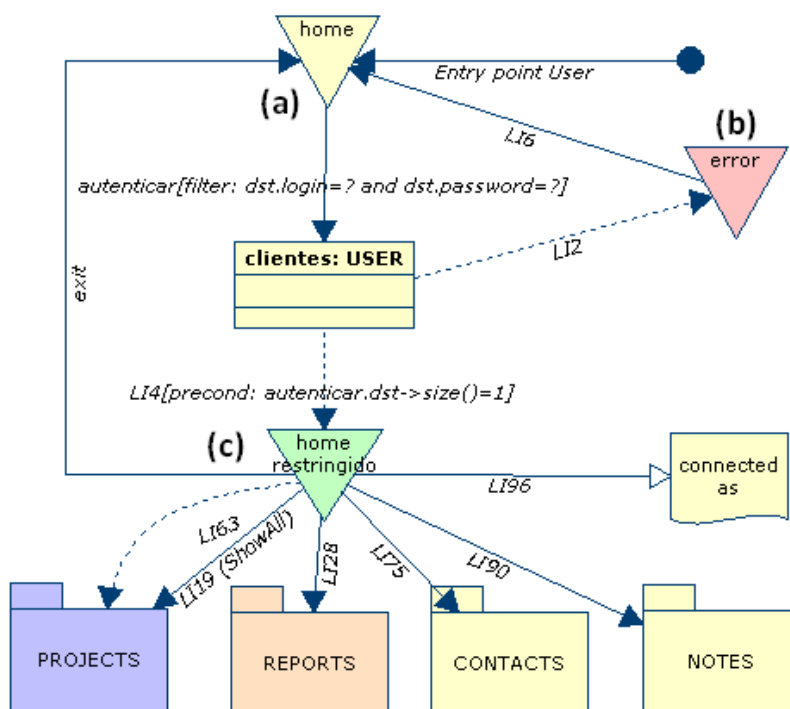


Fig 4.5. NAD-0: NAD de primer nivel para TaskManager

La Figura 4.6 presenta el APD asociado al NAD anterior (APD-0), el cual incluye tres páginas abstractas:

- La primera página abstracta (Fig. 4.6 (a)) se corresponde la colección Home y el enlace autenticar cuyo destino es la clase navegacional Clientes (Fig. 4.5 (a)).

- La segunda página abstracta (Fig. 4.6 (b)) se corresponde con la colección Error y en el enlace navegacional *LI6* (Fig. 4.5 (b)). Esta página es consecuencia de un enlace navegacional con visualización en destino que accede a la colección *Error*. Aquí se muestra el error asociado a las credenciales.
- La tercera página abstracta (Fig. 4.6 (c)) se corresponde con la colección home restringido y sus enlaces navegacionales hacia los demás destinos navegacionales (Fig. 4.5 (c)). Las correspondencias entre el NAD y el APD son respectivamente: *Projects->Tasks*, *Reports->Reports*, *Contacts->Contacts*, *Notes->Whats new*. Sin embargo, la etiqueta *connected as*, al ser accesible mediante un enlace navegacional con visualización en origen (*LI96*), se muestra en la misma página abstracta.

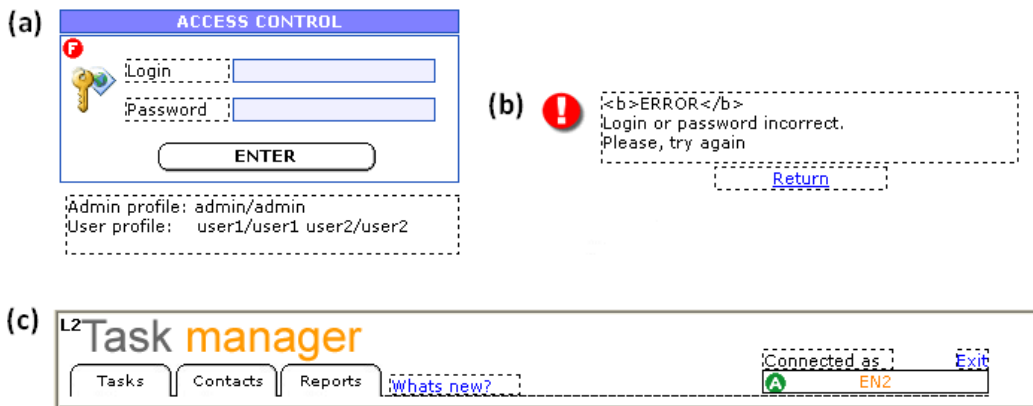


Fig 4.6. APD-0: APD refinado y asociado al NAD-0

La Figura 4.7 presenta el siguiente NAD correspondiente al destino navegacional *Projects* que representa la gestión de las tareas (NAD-1). El usuario puede iniciar la navegación (*LR3*) seleccionando una de las carpetas que contiene Tareas. Estas carpetas pueden ser las predefinidas (enlaces: *todas*, *fuera de plazo*, *pendientes o terminadas*), las que filtran por nombre de usuario cuyo acceso está habilitado mediante una precondición para el administrador (*Usuario2*) o pueden haber sido creadas por el administrador (método *new* de la clase *Clasificador*). Cuando se accede a la información de una carpeta (*Clasificador_detail*), el enlace *LI12* de visualización en origen muestra todas las tareas de dicha carpeta, permitiendo crear nuevas tareas (método *new* de la clase *Tarea*), para posteriormente poder acceder a sus detalles (*LI49*) con el fin de modificar sus propiedades, además de poder asociarles comentarios (clase *Comentario*) y archivos (clase *Fichero*).

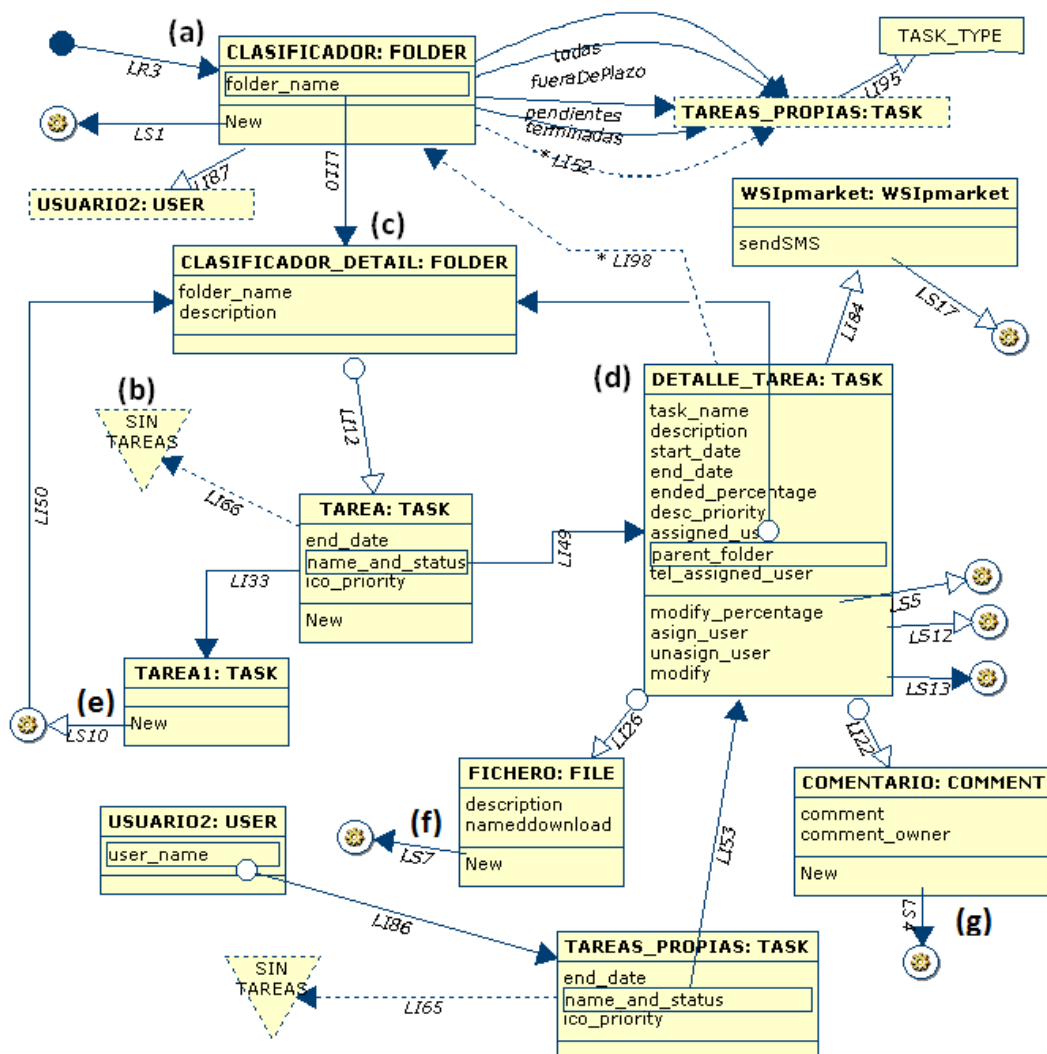


Fig 4.7. NAD-1: NAD de gestión de tareas

Debido al extenso número de enlaces internos con visualización en destino existen diversas páginas abstractas que se derivan del anterior NAD, la Figura 4.8 sólo presenta el APD asociado. En él se muestran las siguientes páginas abstractas refinadas:

- La primera página abstracta (Fig. 4.8 (a)) se corresponde con las clases navegacionales *Clasificador* y *Usuario2* (Fig 4.7 (a)). En ella se muestran las carpetas disponibles y el enlace para permitir la creación de una nueva carpeta (permitido al administrador).
- La segunda página abstracta (Fig. 4.8 (b)) se corresponde con las clases navegacionales *Clasificador_detail* y *Tarea* (Fig 4.7 (b)) ya que existe un enlace

con visualización en origen entre ambas clases. En ella se muestra la lista de tareas filtrada por carpeta.

- La tercera página abstracta (Fig. 4.8 (c)) se corresponde con la colección *Sin tareas* (Fig 4.7 (c)). En ella se muestra el mensaje de error asociado.
- La cuarta página abstracta (Fig. 4.8 (d)) se corresponde con las clases Navegacionales: *Detalle_tarea*, *Fichero* y *Comentario* (Fig 4.7 (d)) ya que entre ellas existe un enlace con visualización en origen. En ella se muestra la información detallada de la Tarea junto con las acciones que se pueden realizar.
- La quinta página abstracta (Fig. 4.8 (e)) se corresponde con el método *new* de la clase navegacional *Tarea1* (Fig. 4.7 (e)). En ella se provee la interfaz para introducir una nueva tarea en la aplicación Web. Los campos del formulario se corresponden a los parámetros del método que debe introducir el usuario (definidos en el modelo de clases).
- La sexta página abstracta (Fig. 4.8 (f)) se corresponde con el método *new* de la clase navegacional *Fichero* (Fig. 4.7 (f)). En ella se provee la interfaz para asociar un nuevo archivo a una tarea ya existente. Los campos del formulario se corresponden a los parámetros del método que debe introducir el usuario (definidos en el modelo de clases).
- La séptima página abstracta (Fig. 4.8 (g)) se corresponde con el método *new* de la clase comentario (Fig. 4.7 (g)). En ella se provee la interfaz para asociar un nuevo comentario a una tarea ya existente. En este caso el único campo de formulario es el texto del comentario.

Nota: Se han omitido las páginas abstractas relacionadas con los métodos *modify*, *assign_user*, *unassign-user* de la clase por ser muy parecidas a la generada por el método *new*; y *send_SMS* de la clase *WSIPmarket* por ser una funcionalidad aportada por la lógica de un servicio Web ajeno a *TaskManager*.

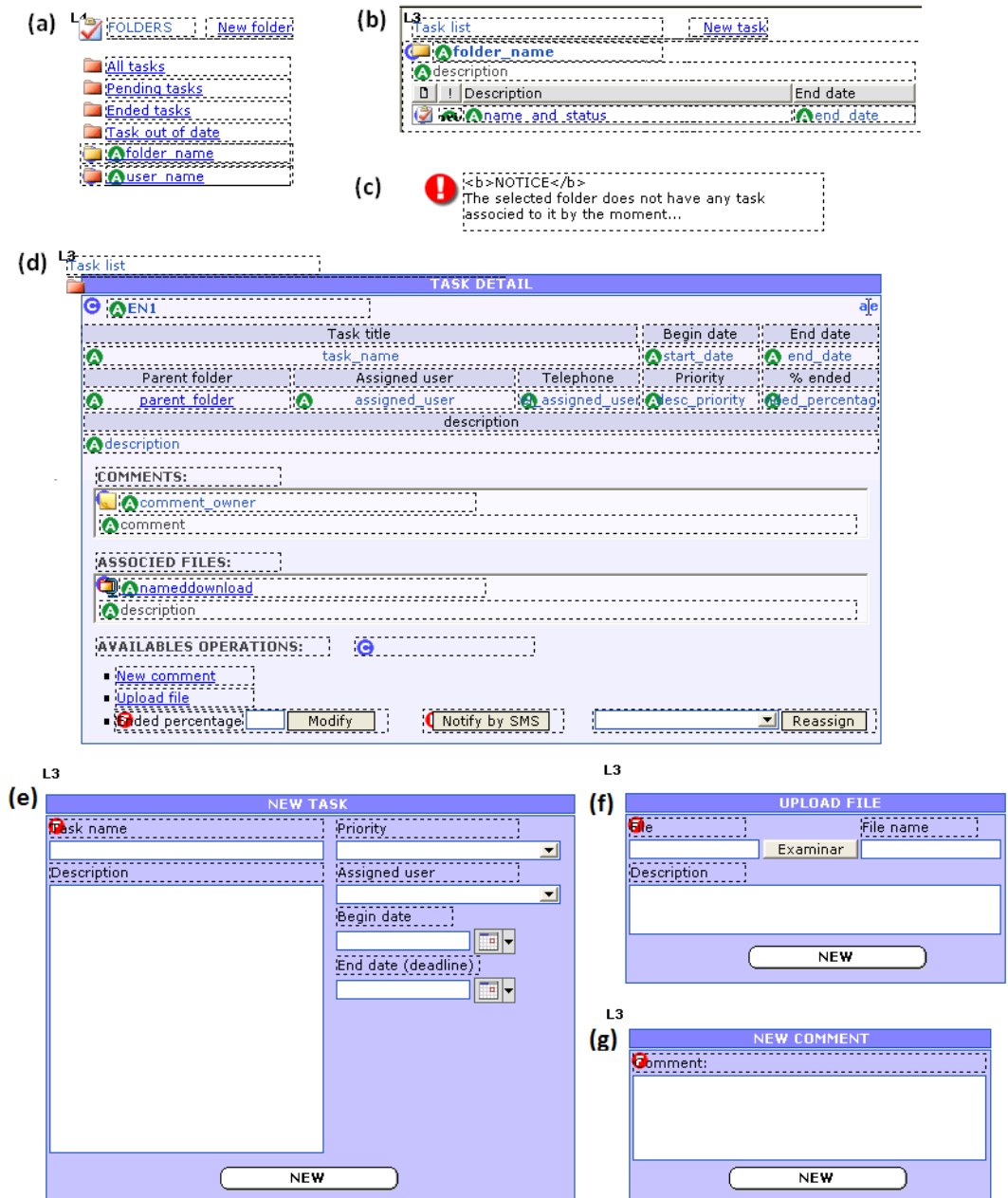


Fig 4.8. APD-1: APD refinado y asociado al NAD-1

La Figura 4.9 presenta el NAD correspondiente a la explosión del destino navegacional *Contacts* (NAD-2) que representa la gestión de contactos. El usuario (*user*) inicia la navegación (*LR5*) en la colección *contact menu* donde podrá consultar la información (*name, surname, etc.*) de todos los contactos registrados, o bien, hacer una búsqueda específica indicando las iniciales del contacto o una cadena de texto (enlaces: *byString* y *ByInitial*). Si la búsqueda no produce ningún resultado (*LI83*) se llega a un estado de advertencia representado mediante la colección *No Coincidences*. Además, el programador podrá añadir nuevos contactos ejecutando el método *new* de la clase navegacional *Contact1* accediendo a ella a través del enlace *newContact*.

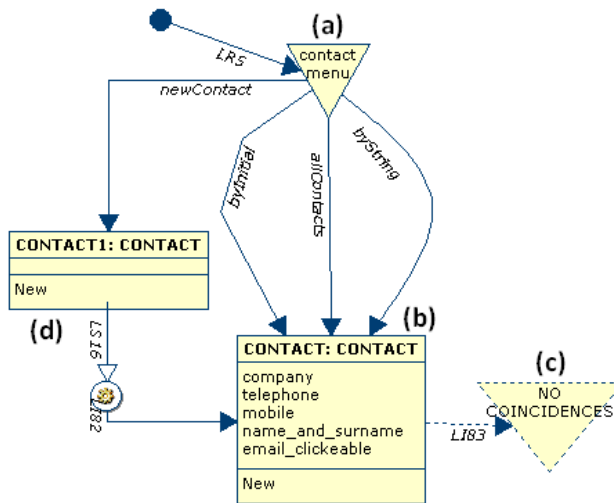


Fig 4.9. NAD-2: NAD de gestión de contactos

La Figura 4.10 presenta el APD asociado al NAD anterior (APD-2), el cual incluye cuatro páginas abstractas:

- La primera página abstracta (Fig 4.10 (a)) se corresponde con la colección *Contact menú* (Fig 4.9 (a)) En ella se representa la interfaz con los distintos modos de acceso a los contactos y la posibilidad de añadir uno nuevo.
- La segunda página abstracta (Fig 4.10 (b)) se corresponde con la clase navegacional *Contact* (Fig 4.9 (b)). En ella se muestra el listado de contactos, obtenidos como resultado tras uno de los modos de acceso anteriores, junto a los atributos visibles que contiene la clase navegacional.
- La tercera página abstracta (Fig 4.10 (c)) se corresponde con la colección *No Coincidences* (Fig 4.9 (c)). En ella se presenta el mensaje de advertencia cuando no se ha encontrado ningún contacto.

- La cuarta página abstracta (Fig 4.10 (d)) se corresponde con el método *new* de la clase navegacional *Contact1* (Fig 4.9 (d)). En ella se provee la interfaz para introducir un nuevo contacto en la aplicación Web. Los campos del formulario se corresponden a los parámetros del método que se le piden al usuario (definidos en el modelo de clases).

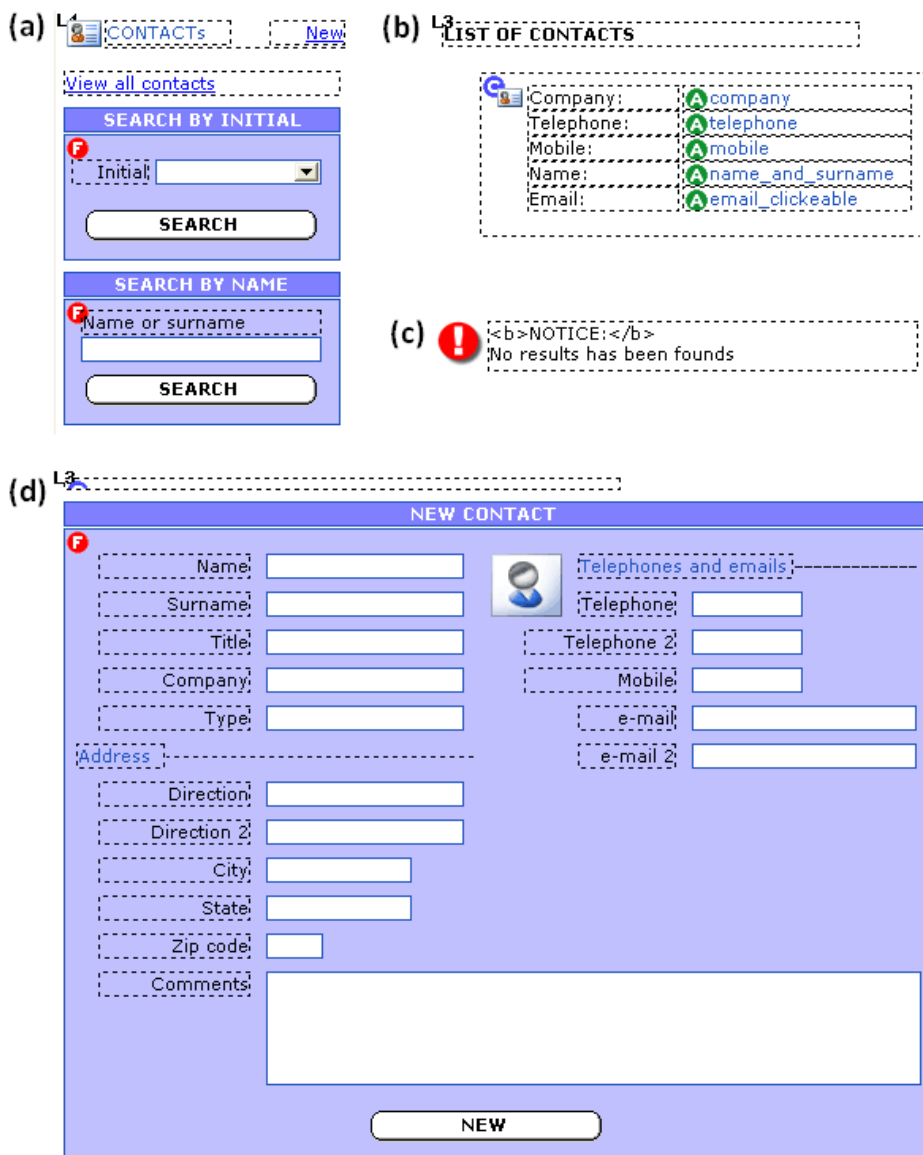


Fig 4.10. APD-2: APD refinado y asociado al NAD-2

La Figura 4.11 presenta el siguiente NAD correspondiente al destino navegacional Reports (NAD-3). El usuario inicia la navegación (LR4) en la colección *Reports* que accede en modo origen a los títulos de los informes (*Daily_Report4*), a los informes de cada usuario (*User1*) y al informe diario (*Daily_Report3*). Además, desde la colección *Reports*, el administrador puede buscar los informes por contenido (enlace *byContent*) o fechas (enlace *byDates*), y en el caso del usuario, puede crear un informe diario (*Daily_Report*). En caso que no se encuentren informes (LI45) se accede a la colección *No coincidencias*, mientras que si existen informes, se muestra su contenido (*Daily_Report2*).

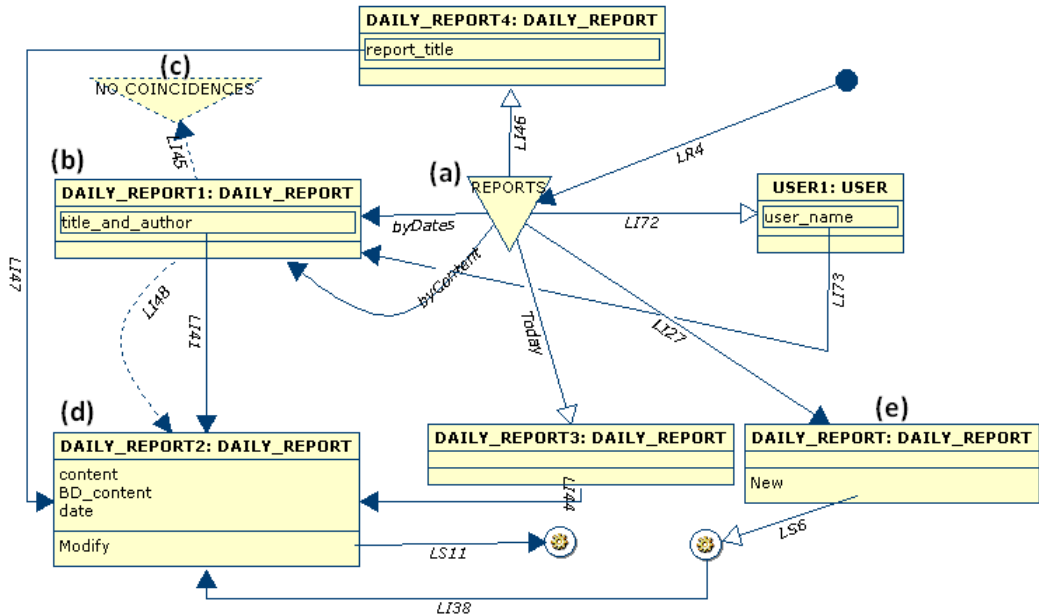


Fig 4.11. NAD-3: NAD de gestión de informes

La Figura 4.12 presenta el APD asociado al NAD anterior (APD-3), el cual incluye cinco páginas abstractas:

- La primera página abstracta (Fig. 4.12 (a)) se corresponde con la colección *Reports* y las clases Navegacionales: *Daily_Report3*, *Daily_Report4* y *User1* (Fig.4.11 (a)), al estar conectadas a la colección mediante enlaces de visualización en origen. En ella se representa el acceso a los informes según los diferentes modos de acceso.
- La segunda página abstracta (Fig. 4.12 (b)) se corresponde con la clase navegacional *Daily_Report1* (Fig. 4.11 (a)). En ella se representa la lista de informes obtenidos tras los diferentes modos de acceso, junto con los nombres de sus autores.

- La tercera página abstracta (Fig. 4.12 (c)) se corresponde con la colección *No Coincidencias* (Fig. 4.11 (c)). En ella se muestra el mensaje de advertencia cuando no se han encontrado informes.
- La cuarta página abstracta (Fig. 4.12 (d)) se corresponde con la clase navegacional *Daily_Report2* (Fig. 4.11 (d)). En ella se representa la información asociada al informe que se ha accedido.
- La quinta página abstracta (Fig. 4.12 (e)) se corresponde con el método *new* de la clase *Daily_Report* (Fig. 4.11 (e)). En ella se provee la interfaz para que el usuario introduzca un nuevo informe en la aplicación Web. Los campos del formulario se corresponden a los parámetros del método que se le piden al usuario (definidos en el modelo de clases).

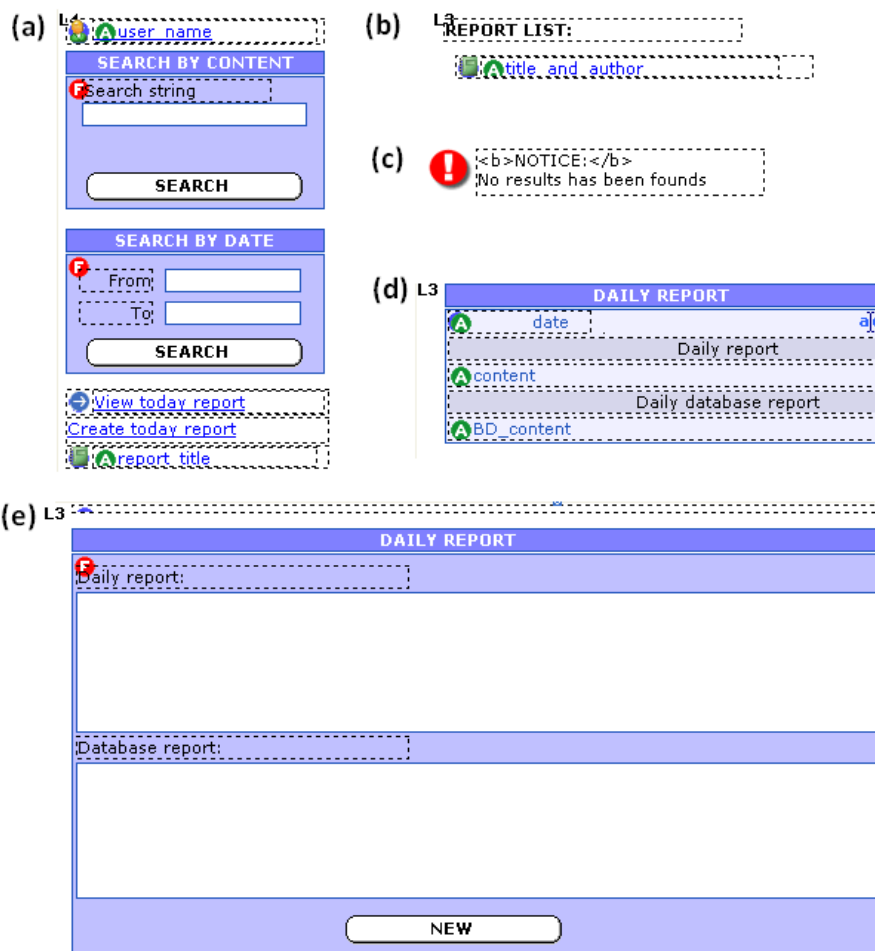


Fig 4.12. APD-3: APD refinado y asociado al NAD-3

Cabe resaltar que en el modelo de presentación, todas las páginas abstractas pertenecientes a los APD anteriores son ubicadas mediante marcos para componer la página Web de la interfaz de usuario final. Estos marcos se definen en la plantilla Tlayout, pudiendo asignarle a cada página abstracta un marco donde será visualizada. La Figura 4.13 muestra la representación esquemática de la plantilla Tlayout para la aplicación Web TaskManager. Los identificadores de los marcos (L1, L2, L3 y L4) se corresponden con los números que aparecen al lado de cada página abstracta.

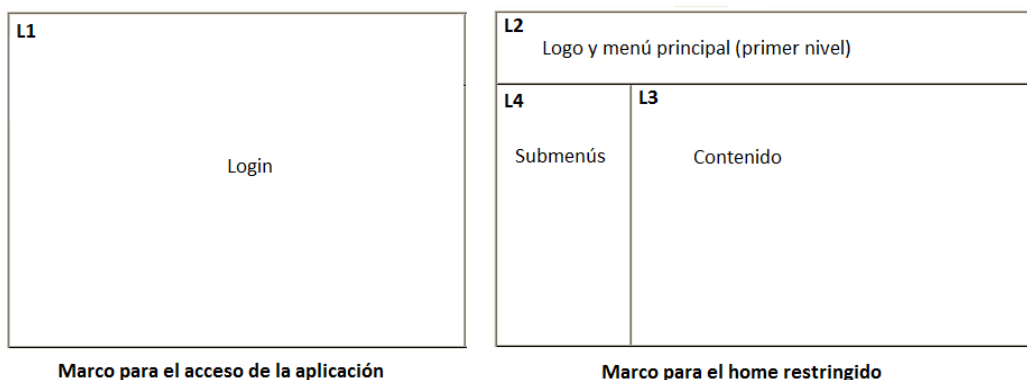


Fig 4.13. Representación de la plantilla Tlayout

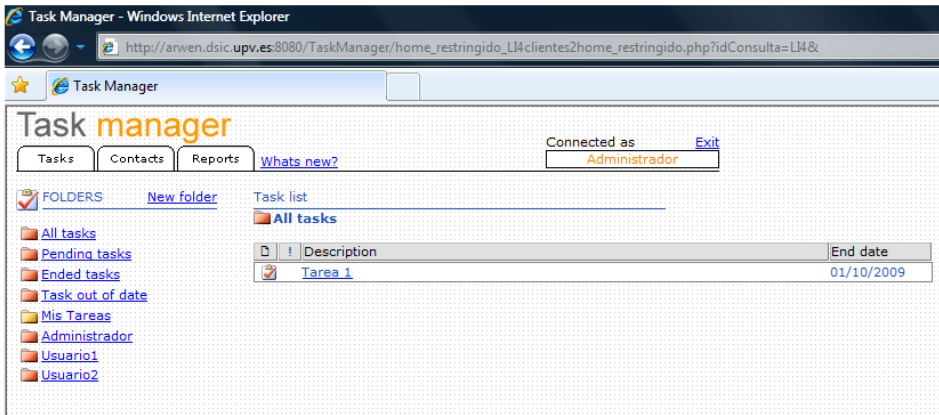
Todos los modelos PIM presentados anteriormente sirven de entrada al compilador de modelos para producir el código de la aplicación Web final.

4.2.4. APLICACIÓN WEB FINAL (MODELO DE CÓDIGO)

La aplicación Web final es obtenida directamente por el compilador de modelos. A partir de modelos PIM se obtiene la capa de negocio e interfaz de usuario en PHP (se puede elegir la versión PHP), mientras que para la capa de persistencia se provee un esquema que conforma la definición de la base de datos, pudiendo elegir el formato del esquema (MySQL, Excel, Oracle, etc.). A continuación se muestran ejemplos de la interfaz de usuario final, tras ejecutar el código fuente proporcionado, en las tres funcionalidades principales de la aplicación Web. Para ello se ha contado se han creado dos instancias de usuario normales (*user1* y *user2*) y una de administrador (*admin*).

La Figura 4.14 muestra la interfaz de usuario final asociada a la gestión de tareas (IUF-1), la Figura 4.15 la asociada a la gestión de contactos (IUF-2), y por último, la Figura 4.16 la asociada a la gestión de informes (IUF-3).

(1)



(2)

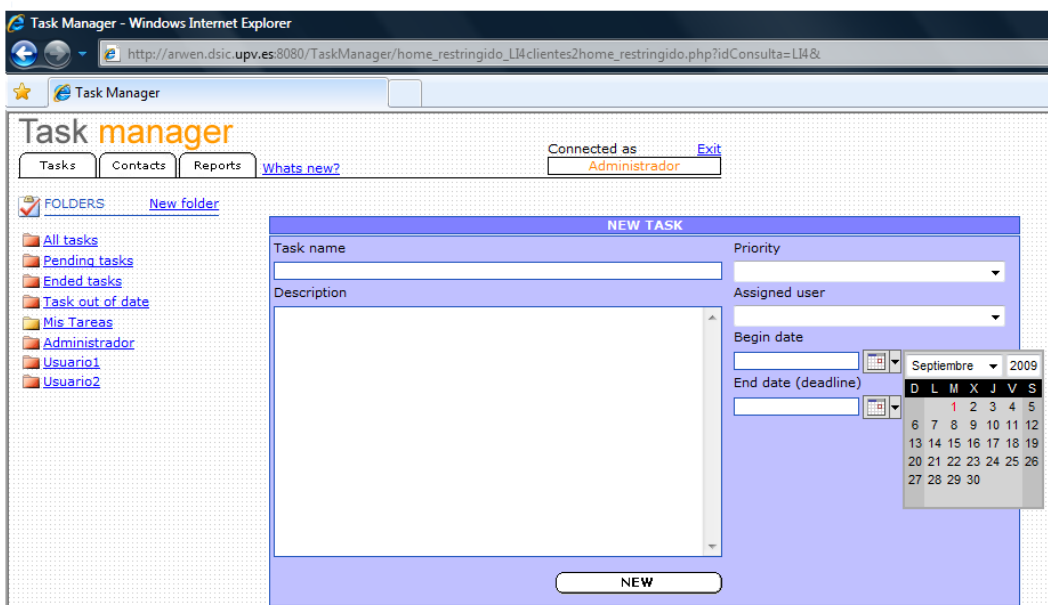
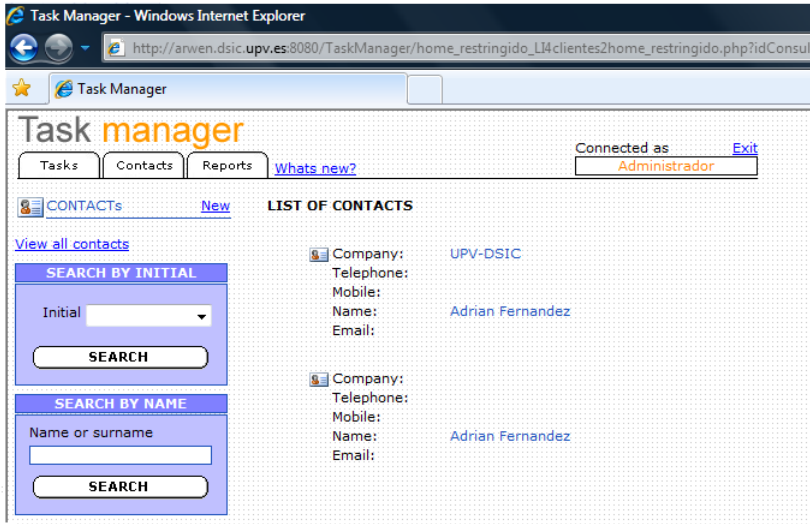


Fig 4.14. IUF-1: Interfaz de usuario final para la gestión de tareas

(1)



(2)

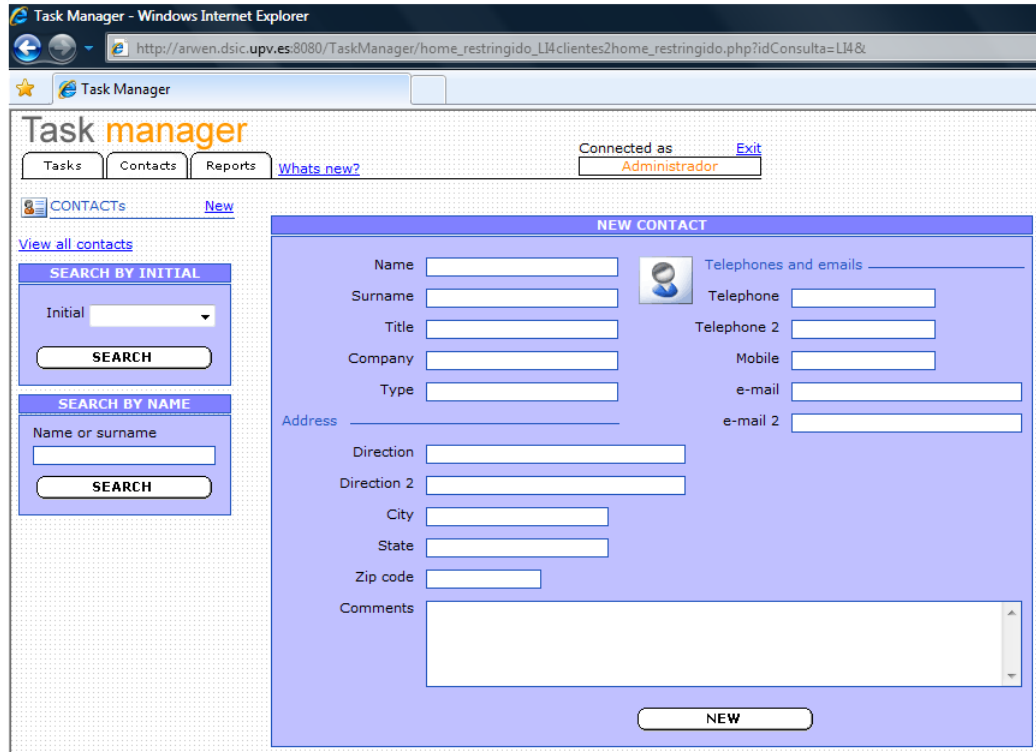


Fig 4.15. IUF-2: Interfaz de usuario final para la gestión de contactos

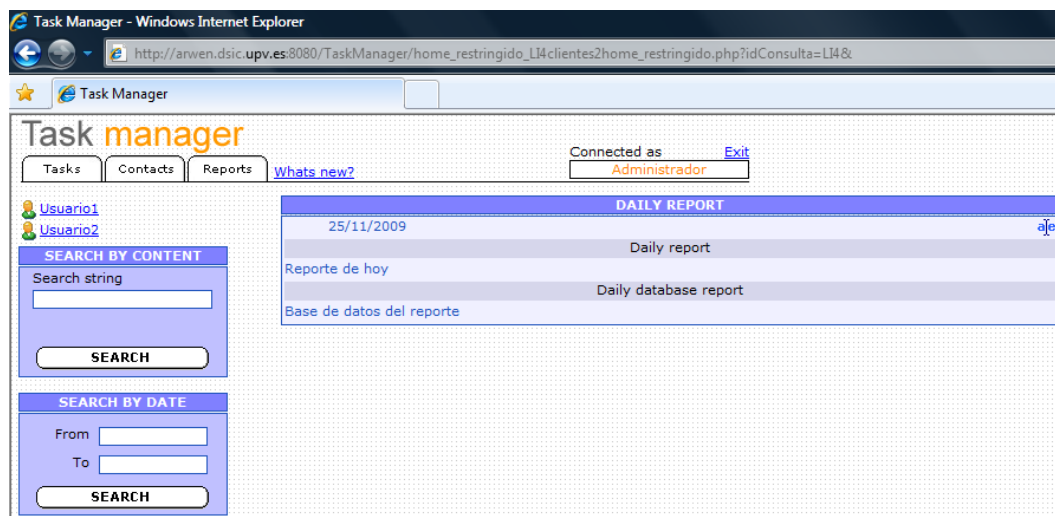


Fig 4.16. IUF-3: Interfaz de usuario final para la gestión de informes

4.3. INSTANCIACIÓN DE WUEP

En esta sección se describe una evaluación de usabilidad de la aplicación Task Manager. Para ello se ha instanciado el proceso WUEP siguiendo las tareas definidas en el Capítulo 3. Cada una de las sub-secciones siguientes se corresponde con cada una de las fases de WUEP.

4.3.1. REQUISITOS EVALUACIÓN

En esta fase se establecía el propósito de la evaluación; los perfiles relacionados con el tipo de aplicación Web, el método de desarrollo y el contexto de uso; y la selección de atributos de usabilidad. Todos estos aspectos vienen reflejados en el documento *Requisitos de evaluación*:

Propósito de la evaluación: El objetivo es realizar una evaluación formativa de la usabilidad de la aplicación Web TaskManager. Es decir, la evaluación de usabilidad tendrá lugar en paralelo con el proceso de desarrollo. Esta decisión se justifica debido al hecho que la empresa que ha proporcionado la aplicación Web, aun tiene pensado continuar el desarrollo para implantarla en un futuro cercano, por lo que una evaluación con usuarios finales sería prematuro.

Perfiles: A continuación se definen los perfiles relacionados con el tipo de aplicación Web, el método de desarrollo y el contexto de uso:

- *Tipo de Aplicación Web:* La aplicación TaskManager pertenece a la familia de aplicaciones Web determinadas Intranet. Algunas de las características básicas más comunes en este tipo de aplicaciones son:

- Totalmente basada en Web proporcionando las ventajas inherentes a este ámbito: acceso a la aplicación desde cualquier plataforma, centralización y sincronización de contenidos, aprovechamiento de recursos, etc.
 - Foros o métodos de contacto internos para discusión según temáticas.
 - Carpetas para todos los tópicos relevantes.
 - Determina códigos de acceso según niveles de seguridad.
 - Crea lugares para publicar notas, artículos, opinión, etc.
 - Define explícitamente quién origina cada tipo de información y qué las personas que tienen acceso a dicha información.
 - Facilita la realización del trabajo colaborativo de manera ágil y simple.
 - Permite o niega el acceso a los resultados publicados según el perfil de seguridad.
 - Calendario personal.
 - Agenda de contactos (base de datos sobre los contactos de la empresa).
 - Publica eventos destacados, novedades, etc.
- *Método de desarrollo:* El método de desarrollo Web dirigido por modelos empleado fue OO-H [28] (La información referente al método se encuentra en la Sección 4.1). Mientras, que la herramienta CASE empleada en el desarrollo de la aplicación Web fue VisualWade, debido al completo soporte que ofrece para el desarrollo basado en el método OO-H.
- *Contexto de uso:* A continuación se detallan algunos de sus aspectos más relevantes:
- La aplicación Web va destinada a empresas de desarrollo Web que necesitan cubrir el control y seguimiento de los proyectos que realizan.
 - El perfil del usuario es el de un desarrollador o programador perteneciente intervalo de edad entre 20 y 40 años con conocimientos de nivel alto en informática. Mientras que el perfil del administrador es el del director de proyectos, sin un intervalo de edad específico y con conocimientos de nivel medio-alto en informática.
 - Los requisitos tecnológicos empleados son: ordenadores personales con sistema operativo Windows XP/Vista y navegadores Web internet Explorer y Mozilla Firefox.

Atributos de usabilidad: A falta de una especificación de requisitos no-funcionales, los atributos de usabilidad pertenecientes al Modelo de Usabilidad Web pueden ser empleados a modo de catálogo. De esta forma, para una intranet y a modo de ejemplo se ha seleccionado los siguientes atributos:

- Atributo 1.1.2: *Visualización textual* (sub-característica: *Legibilidad visual*).
- Atributo 1.4.1: *Acciones mínimas* (sub-característica: *Ahorro de esfuerzo*).
- Atributo 1.5.3: *Retro-alimentación inmediata* (sub-característica: *Orientación al usuario*).
- Atributo 1.5.2: *Calidad de los mensajes de aviso* (sub-característica: *Orientación al usuario*).
- Atributo 1.6.2: *Clickabilidad* (sub-característica: *Navegabilidad*).
- Atributo 1.6.3: *Interconectividad* (sub-característica: *Navegabilidad*).
- Atributo 1.6.4: *Alcanzabilidad* (sub-característica: *Navegabilidad*).
- Atributo 2.2.1: *Nombre de enlaces significativos* (sub-característica: *Predictibilidad*).
- Atributo 3.1.1: *Compatibilidad con navegadores y plugins* (sub-característica: *Compatibilidad*).
- Atributo 3.2.1: *Validez de datos de entrada* (sub-característica: *Gestión de los datos*).
- Atributo 3.3.2: *Soporte a operaciones cancelables* (sub-característica: *Controlabilidad*).
- Atributo 5.3: *Soporte a texto alternativo* (sub-característica: *Accesibilidad técnica*).
- Atributo 6.3: *Uniformidad en la posición de las secciones de la interfaz* (sub-característica: *Grado de atracción*).

4.3.2. ESPECIFICACIÓN DE LA EVALUACIÓN

En esta fase se seleccionaban aquellos artefactos de la aplicación Web que se desea evaluar y las métricas que se van a emplear. Las métricas seleccionadas se operacionalizaban y se establecían umbrales para poder detectar problemas de usabilidad y criterios de agregación para combinar las cuantificaciones de los atributos. Finalmente se definía la plantilla del informe de usabilidad para presentar dichos problemas de usabilidad. Todos estos aspectos vienen reflejados en el documento *Especificación de la evaluación*:

Artefactos a evaluar: Atendiendo a los requisitos de la evaluación (evaluación formativa) y a los artefactos que proporciona el método OO-H: modelos PIM y aplicación Web final (CM). Los artefactos que se evaluarán serán:

- Modelo de clases: Sólo existe uno. (ver sub-sección 4.2.2).
- Modelo navegacional: NAD-0, NAD-1, NAD-2, NAD-3 (ver sub-sección 4.2.3).
- Modelo de presentación: APD-0, APD-1, APD-2, APD-3, (ver sub-sección 4.2.3).
- Modelo código: IUF-1, IUF-2, IUF-3 (ver sub-sección 4.2.4).

Métricas a emplear: Atendiendo a la selección de atributos de usabilidad y artefactos del proceso de desarrollo. Se han seleccionado métricas asociadas a dichos atributos (ver descripción genérica en Sección 3.2.3) y se han clasificado según en qué artefactos se pueden aplicar:

- Modelo de clases:
 - Disponibilidad de valores por defecto.
- Modelo navegacional:
 - Amplitud de la navegación.
 - Profundidad de la navegación.
 - Densidad de la navegación.
 - Compactibilidad
- Modelo de presentación:
 - Proporción de nombres adecuados para enlaces.
 - Proporción mensajes de aviso significativos.
 - Contraste de color.
 - Enlaces distinguibles.
 - Proporción de imágenes con texto alternativo asociado.
 - Variación en la composición de los marcos.
- Modelo código:
 - Proporción de elementos que muestran estado actual.
 - Operaciones de usuario cancelables.
 - Número de elementos no alineados o desencuadrados.
 - Proporción de mecanismos de validación de datos a la entrada.
 - Diferencias de comportamiento y aspecto de elementos de interfaz entre navegadores.

Operacionalización de las métricas y establecimiento de umbrales e indicadores: A continuación se propone la operacionalización de cada métrica anterior para ser aplicada en el artefacto correspondiente al método OO-H, basándonos en su descripción genérica (ver Sección 3.2.3):

- *Disponibilidad de valores por defecto (DVP):* En el modelo de clases, los valores por defecto hacen referencia a aquellos atributos de las clases que requieren de un valor por defecto. Esta propiedad puede ser consultada en las propiedades del atributo. Por lo tanto, la fórmula instanciada a aplicar sería:

$$DVP(\text{Modelo Clases}) = \frac{\text{Atributos con valor por defecto}}{\text{Núm.total de atributos potenciales}}$$

Siendo los atributos potenciales, aquellos atributos que son dados a tener valor por defecto: atributos identificadores, atributos derivados, atributos con un único conjunto de valores posible, etc.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0 \leq DVP \leq 0.5]$
- Problema usabilidad Medio: $[0.5 < DVP \leq 0.7]$
- Problema usabilidad Leve: $[0.7 < DVP \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < DVP \leq 1]$

- *Amplitud de la navegación (AN)*: En el modelo navegacional, la navegación de primer nivel se representa en el NAD que incluye los principales *Destinos navegacionales*. Por lo tanto, la fórmula instanciada a aplicar sería:

$$AN(NAD) = \text{Número total de Destinos Navegacionales}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Entero mayor que 0) serían:

- Problema usabilidad Grave: $[AN > 5]$
- No hay problema de usabilidad: $[1 \leq AN \leq 5]$

- *Profundidad de la navegación (PN)*: En el modelo navegacional, la profundidad de la navegación puede calcularse en cada NAD, siendo ésta el camino más largo de navegación, siguiendo enlaces de navegación (internos o de servicio) manuales con visualización en destino (por representar un paso de navegación), hasta alcanzar una clase navegacional o servicio, a partir de la cual no se pueda llegar a una clase navegacional o servicio ya visitado por el camino de navegación. Por lo tanto, la fórmula instanciada a aplicar sería:

$$PN(NAD) = \text{Max (camino navegación entre } x \text{ e } y)$$

Siendo 'x' la primera clase navegacional o colección donde se inicia la navegación en un mismo NAD, e 'y' la clase navegacional o servicio tras la cual solo es posible producirse un ciclo de navegación o el fin de ella.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Entero mayor que 0) serían:

- Problema usabilidad Grave: $[PN > 9]$
- No hay problema de usabilidad: $[1 \leq PN \leq 9]$

- *Densidad de la navegación (DN)*: En el modelo navegacional, la densidad de la navegación puede calcularse en cada NAD, siendo está el cociente entre el número total de enlaces manuales (internos y de servicio) con visualización en destino (por representar un paso de navegación) y el número total de nodos navegacionales (Clases navegacionales, Colecciones, Servicios) a los cuales se accede mediante un enlace de visualización en destino. Por lo tanto, la fórmula instanciada a aplicar sería:

$$DN(NAD) = \frac{\text{Núm.Enlaces (visualización destino)}}{\text{Núm.total de nodos (accedidos por visualización destino)}}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Real mayor que 0) serían:

- Problema usabilidad Medio: $[DN > 3]$
 - No hay problema de usabilidad: $[0 < DN \leq 3]$
- *Compactibilidad (Compactness - Cp)*: En el modelo navegacional, la compactibilidad puede calcularse en cada NAD, si tomamos como referencia los elementos: clase navegacional, colección y destino navegacional del NAD como nodos, y todos los tipos de enlaces internos que los conectan. Por lo tanto, la fórmula instanciada a aplicar sería:

$$Cp(NAD) = \frac{Max - \sum_i \sum_j Cij}{Max - Min}$$

Donde $Max = (n^2 - n) * k$; $Min = (n^2 - n)$, n = cantidad de nodos (clase navegacional, colección y destino navegacional) en el NAD; k = constante superior a la cantidad de nodos (normalmente suele ser igual a n); $\sum_i \sum_j Cij$ = la suma de las distancias pertenecientes a una matriz de distancias normalizadas con el factor k ; y Cij = la distancia mínima entre los nodos i y j .

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Leve: $[0 \leq Cp < 0.2]$ y $[0.8 < Cp \leq 1]$
 - No hay problema de usabilidad: $[0.2 \leq Cp \leq 0.8]$
- *Proporción de nombres adecuados para enlaces (PNAE)*: En el modelo de presentación, la proporción de nombres adecuados para los enlaces puede calcularse en cada APD, mediante la siguiente fórmula:

$$PNAE(APD) = \frac{\text{Núm.Enlaces adecuados}}{\text{Núm.total de enlaces en el APD}}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq PNAE \leq 0.5]$
 - Problema usabilidad Medio: $[0.5 < PNAE \leq 0.7]$
 - Problema usabilidad Leve: $[0.7 < PNAE \leq 0.99]$
 - No hay problema de usabilidad: $[0.99 \leq PNAE \leq 1]$
- *Proporción mensajes de aviso significativos (PMAS)*: En el modelo de presentación, los mensajes de aviso se representan cada uno en una página abstracta distinta perteneciente a un APD. De esta forma, es posible calcular la

proporción de mensajes significativos mediante la siguiente fórmula instanciada en cada APD:

$$PMAS(APD) = \frac{\text{Núm.mensajes de aviso significativos}}{\text{Núm.total de mensajes de aviso en el APD}}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq PMAS \leq 0.5]$
 - Problema usabilidad Medio: $[0.5 < PMAS \leq 0.7]$
 - Problema usabilidad Leve: $[0.7 < PMAS \leq 0.99]$
 - No hay problema de usabilidad: $[0.99 < PMAS \leq 1]$
- *Contraste de color (CC)*: En el modelo de presentación, cada elemento textual (etiquetas predefinidas, enlaces, contenido de los atributos del modelo de clases, etc.) perteneciente a cada página abstracta de un APD, tiene asociado los atributos *Forecolor* (color de texto) y *Backcolor* (color de fondo), ambos en formato hexadecimal, que definen los colores del texto y fondo del elemento. El contraste de color de cada uno de estos elementos se puede calcular mediante la siguiente fórmula:

$$\begin{aligned} &\forall \text{elemento } e \in APD, \\ CC = &|Forecolor(e,rojo) - Backcolor(e,rojo)| + \\ &|Forecolor(e,verde) - Backcolor(e,verde)| + \\ &|Forecolor(e,azul) - Backcolor(e,azul)| \end{aligned}$$

Siendo rojo, verde y azul cada una de las componentes RGB base decimal.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Valor entero mayor o igual que 0) y la propuesta del W3C serían:

- Problema usabilidad Grave: $[0 \leq CC < 500]$
 - No hay problema de usabilidad: $[CC \geq 500]$
- *Enlaces distinguibles (ED)*: En el modelo de presentación, se encuentran representadas las propiedades gráficas de los elementos textuales: enlaces (*E*), etiquetas predefinidas (*EP*) y el contenido de los atributos del modelo de clases (*CA*) en cada APD. De esta forma, la fórmula a aplicar sería:

$$ED(APD) = 1 - 0.5 \left(\frac{E \text{ confundibles}}{\text{Núm. total de } E} + \frac{EP \text{ y } CA \text{ confundibles}}{\text{Núm. total de } EP \text{ y } CA} \right)$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq ED \leq 0.5]$
- Problema usabilidad Medio: $[0.5 < ED \leq 0.7]$

- Problema usabilidad Leve: $[0.7 < ED \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < ED \leq 1]$

- *Proporción de imágenes con texto alternativo asociado (PIT)*: En el modelo de presentación se pueden integrar imágenes externas en cada APD, las cuales tienen un atributo asociado llamado *text* para proporcionar el texto alternativo. Por lo tanto, la fórmula a aplicar sería:

$$PIT(APD) = \frac{\text{Núm.de imágenes con atributo "text "no vacío}}{\text{Núm.total de imágenes en el APD}}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq PIT \leq 0.5]$
- Problema usabilidad Medio: $[0.5 < PIT \leq 0.7]$
- Problema usabilidad Leve: $[0.7 < PIT \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < PIT \leq 1]$

- *Variación en la composición de los marcos (VCM)*: En el modelo de presentación, los marcos que visualizan las páginas abstractas se definen mediante la plantilla *tlayout*. De esta forma, se pueden contabilizar cuantas combinaciones de marcos se emplean para representar el diseño estructural de la aplicación Web:

$$VCM(\text{Modelo Presentación}) = \text{Núm. de combinaciones de marcos}$$

Cada combinación se corresponde con un atributo *FrameSet* perteneciente a la plantilla *tlayout*.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Entero mayor que 0) serían:

- Problema usabilidad Grave: $[VCM > 2]$
- No hay problema de usabilidad: $[0 < VCM \leq 2]$

- *Proporción de elementos que muestran estado actual (PEEA)*: En la interfaz de usuario final se puede observar si el menú de navegación, los enlaces y los controles de entrada de datos ofrecen la posibilidad de mostrar al usuario el estado inmediato de la interfaz.

$$PEEA(IUF) = \frac{\text{Núm.de tipos de elementos que muestran estado actual}}{\text{Núm.total de tipos de elementos}}$$

Siendo ejemplos de tipo de elementos, los citados anteriormente: menú de navegación, enlaces y controles de entrada de datos.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq PEEA \leq 0.5]$

- Problema usabilidad Medio: $[0.5 < PEEA \leq 0.7]$
- Problema usabilidad Leve: $[0.7 < PEEA \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < PEEA \leq 1]$

- *Número de elementos no alineados o desencuadrados (NED)*: En la interfaz de usuario final se puede observar si los elementos contenidos dentro de un mismo marco exceden de sus dimensiones, provocando que no se ajusten adecuadamente para su correcta visualización:

$$NED(IUF) = \text{Núm. de elementos no ajustados en cada interfaz}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Entero mayor que 0) serían:

- Problema usabilidad Grave: $[NED > 0]$
- No hay problema de usabilidad: $[NED = 0]$

- *Operaciones de usuario cancelables (OUC)*: En la interfaz de usuario final asociada a cada una de las operaciones que el usuario puede realizar, se puede observar si existe un control o enlace que permita cancelar la operación en curso pudiendo regresar a la interfaz anterior. De esta forma, la fórmula instancia a cada interfaz de usuario final sería:

$$OUC(IUF) = \frac{\text{Núm.de operaciones cancelables}}{\text{Núm.total operaciones con requisito de cancelación}}$$

Siendo las operaciones con requisito de cancelación, aquellas que pretenden realizar cambios permanentes en la persistencia de los datos que maneja la aplicación Web.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq OUC \leq 0.5]$
- Problema usabilidad Medio: $[0.5 < OUC \leq 0.7]$
- Problema usabilidad Leve: $[0.7 < OUC \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < OUC \leq 1]$

- *Proporción de mecanismos de validación de datos a la entrada (PMV)*: En la interfaz de usuario final se puede observar los controles de entrada de datos que permiten realizar una validación del formato de entrada. La fórmula instanciada sería:

$$PMV(IUF) = \frac{\text{Núm.de elementos con mecanismo de validación}}{\text{Núm.total elementos con requisito de validación}}$$

Siendo los elementos con requisito de validación, aquellos controles donde los datos que se introducen debe estar en un formato correcto.

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[0.0 \leq PMV \leq 0.5]$
- Problema usabilidad Medio: $[0.5 < PMV \leq 0.7]$
- Problema usabilidad Leve: $[0.7 < PMV \leq 0.99]$
- No hay problema de usabilidad: $[0.99 < PMV \leq 1]$

- *Diferencias de comportamiento y aspecto de elementos de interfaz entre navegadores (DCN)*: En la interfaz de usuario final se puede observar si los controles de cada interfaz se muestran y comportan igual en los distintos navegadores que se vayan a emplear. La fórmula instanciada sería:

$$DCN(IUF) = \text{Núm. de elementos con comportamiento o aspecto diferente}$$

Los umbrales establecidos teniendo en cuenta la interpretación del rango de valores posibles que toma la métrica (Reales entre 0 y 1) serían:

- Problema usabilidad Grave: $[DCN > 0]$
- No hay problema de usabilidad: $[DCN = 0]$

Criterios de agregación: En esta evaluación no se definen criterios de agregación puesto que el objetivo está más enfocado en descubrir problemas de usabilidad y no en comparar resultados de atributos entre distintas aplicaciones web.

Plantilla para Informe de usabilidad: Para la presentación de problemas de usabilidad se usa una plantilla con los siguientes campos a rellenar por la fase de ejecución:

Tabla 4.1. Plantilla para Informe de usabilidad.

ID	<i>UPXXX.</i> Código que para hacer referencia al problema de usabilidad detectado, donde XXX es un número secuencial (001, 002, etc.)
Descripción	Descripción textual del problema detectado en base al resultado obtenido de la métrica y los elementos que participan en el cálculo de la misma
Atributo Afectado	<i>ID. Sub-característica / ... / ID. Atributo.</i> Atributo de usabilidad perteneciente al Modelo de Usabilidad Web que se ve afectado por este problema, indicando las <i>sub-características</i> de las que proviene.
Nivel de Criticidad	<i>[Grave / Medio / Leve] (resultado de la métrica)</i> Nivel de criticidad según los intervalos definidos anteriormente para las métricas. Si el problema existe en varios de los artefactos evaluados (del mismo tipo y nivel de abstracción), se considera como nivel de criticidad el más alto de ellos.
Artefacto(s) evaluado(s)	<i>Nombre de los diagramas evaluados (del mismo tipo y nivel de abstracción) donde ha surgido el problema:</i> (diagrama de clases, NADs, APDs, interfaz de usuario final).

Origen(es) del problema	<i>Diagramas causantes del problema:</i> (diagrama de clases, NADs, APDs, interfaz de usuario final) o (reglas de transformación).
Ocurrencias	<i>Número de veces</i> que el problema de usabilidad se repite en cada artefacto evaluado.
Recomendaciones	Sugerencias que permiten prevenir y/o corregir dichos problemas
Prioridad	[Alta, Media, Baja] Prioridad a la hora de realizar los cambios propuestos en el campo recomendaciones (a rellenar por el desarrollador responsable del artefacto a modificar).
Recursos	Recursos (temporales, económicos, etc.) o la falta de ellos para realizar los cambios propuestos (a rellenar por el desarrollador responsable del artefacto a modificar).

4.3.3. DISEÑO DE LA EVALUACIÓN

En esta fase se manifestaban aquellas restricciones técnicas que condicionan la evaluación, y se procede a la elaboración del plan de evaluación.

Restricciones: Las principales restricciones técnicas con las que cuenta la evaluación residen en no tener acceso a las reglas de transformación de modelos y generación de código que provee el compilador de modelos proporcionado por VisualWade para el método OO-H. De esta forma, se podría detectar que existen problemas de usabilidad asociados a dichas reglas, pero no se podría concretar exactamente qué regla es la que origina el problema.

Plan de evaluación: El plan de evaluación a seguir se basa en realizar la evaluación de los artefactos anteriores ordenados de mayor a menor nivel de abstracción, es decir, primero la evaluación a nivel PIM:

- 1º. Modelo de clases.
- 2º. Modelo de navegación: NAD-0, NAD-1, NAD-2 y NAD-3.
- 3º. Modelo de presentación: APD-0, APD-1, APD-2 y APD-3.
- 4º. Modelo código: IUF-1, IUF-2 y IUF-3.

4.3.4. EJECUCIÓN DE LA EVALUACIÓN

En esta última fase se aplican las formulas de las métricas operacionalizadas a los artefactos seleccionados en el orden previsto por el plan de evaluación. Los resultados de las métricas ayudan a detectar los problemas de usabilidad para generar los informes de problemas de usabilidad y proponer cambios que mejoren la usabilidad de la aplicación Web final.

A continuación se presenta la evaluación a nivel PIM que comprende los artefactos intermedios (modelo de clases, modelo de navegación y modelo de presentación) y a nivel CM (interfaz de usuario de la aplicación Web final).

4.3.4.1. EVALUACIÓN A NIVEL PIM

Modelo de clases (ver Fig. 4.4): En este artefacto se ha aplicado la métrica *Disponibilidad de valores por defecto (DVP)* teniendo en cuenta que los atributos que requieren valores por defecto serían en este caso:

- *start_date, priority, ended_percentage* de la clase *Task* (3 atributos)
- *date* y *hour* de la clase *Comment* (2 atributos)
- Todos los atributos identificadores (con el icono de la llave) y derivados (25 atributos)

Es decir, un total de 30 atributos, de los cuales los 5 primeros no tienen asociado ningún valor por defecto. Aplicando la fórmula:

$$DVP = 25/30 = 0.83$$

Por lo tanto, estamos ante el problema de usabilidad leve UP001 ya que el valor pertenece al intervalo $[0.7 < DVP \leq 0.99]$. La Tabla 4.2 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.2. Problema de usabilidad detectado: UP001.

ID	UP001
Descripción	Los atributos <i>start_date, priority, ended_percentage</i> de la clase <i>Task</i> ; y <i>date</i> y <i>hour</i> de la clase <i>Comment</i> no tienen valores por defecto asociados.
Atributo Afectado	1. Facilidad de entendimiento / 1.4. Ahorro de esfuerzo / 1.4.1 Acciones mínimas
Nivel de Criticidad	Leve
Artefacto evaluado	Modelo de clases
Origen del problema	Modelo de clases
Ocurrencias	5 (una por cada atributo)
Recomendaciones	Asociar un valor por defecto a cada uno de ellos mediante la propiedad "Valor por defecto"

Modelo de navegación: los artefactos correspondientes a este modelo son NAD-0 (ver Fig. 4.5), NAD-1 (ver Fig. 4.7), NAD-2 (ver Fig. 4.9) y NAD-3 (ver Fig. 4.11). En estos artefactos se han aplicado las métricas:

- **Amplitud de la navegación (AN).** Esta métrica únicamente se podía aplicar en el NAD-0 ya que hace referencia a la navegación de primer nivel:
 - $AN(NAD-0) = 5$, ya que existen 4 destinos Navegacionales (*Projects, Contacts, Reports, Notes*) más el acceso a esos destinos, que se considera como el destino navegacional *home*.
- **Profundidad de la navegación (PN).** Esta métrica se puede aplicar a todos los NADs:

- $PN(NAD-0) = 2$, ya que el camino de navegación más largo desde el inicio de la navegación hasta alcanzar uno de los cuatro Destinos Navegacionales (*Projects, Contacts, Reports, Notes*) pasa por dos enlaces de visualización en destino manuales.
- $PN(NAD-1) = 4$, ya que el camino de navegación más largo pasa por cuatro enlaces de visualización en destino manuales: *Clasificador* → *Tareas_propias* → *Detalle_tarea* → *Clasificador_detail* → *Tarea1*.
- $PN(NAD-2) = 2$, ya que el camino de navegación más largo pasa por dos enlaces de visualización en destino manuales: *Contact_menu* → *Contact1* → *Contact*.
- $PN(NAD-3) = 3$, ya que el camino de navegación más largo pasa por tres enlaces de visualización en destino manuales: *Reports* → *Daily_report1* → *Daily_report2* → *(LS11) a Servicio*.

Puesto que todos los valores pertenecen al intervalo $[1 \leq PN \leq 9]$ no existen problemas de usabilidad.

- **Densidad de la navegación (DN).** Esta métrica se puede aplicar a todos los NADs:

- $DN(NAD-0) = 7/6 = 1.16$, ya que existen siete enlaces con visualización en destino (*autenticar, exit, LI6, LI63, LI28, LI75, LI90*) y seis nodos que se acceden mediante enlaces con visualización en destino (*home, clientes, Projects, Contacts, Reports, Notes*).
- $DN(NAD-1) = 15/8 = 1.87$, ya que existen quince enlaces con visualización en destino (*LS1, todas, fueraDePlazo, pendientes, terminadas, LI10, LI50, ...*) y ocho nodos que se acceden mediante enlaces con visualización en destino (*Clasificador_detail, Tarea1, Detalle_tarea, Tareas_propias, Servicio(LS1), Servicio(LS4), Servicio(LS7) y servicio (LS13)*).
- $DN(NAD-2) = 5/2 = 2.5$, ya que existen cinco enlaces con visualización en destino (*newContact, byInitial, allContacts, ByString y LI82*) y dos nodos que se acceden mediante enlaces con visualización en destino (*Contact y Contact1*).
- $DN(NAD-3) = 9/4 = 1.75$, ya que existen nueve enlaces con visualización en destino (*byDates, byContent, LI73, LI27, LI41, LI47, LI44, LS11, LS38*) y cuatro nodos que se acceden mediante enlaces con visualización en destino (*Daily_report, Daily_report1, Daily_report2 y servicio(LS11)*).

Puesto que todos los valores pertenecen al intervalo $[0 < DN \leq 3]$ no existen problemas de usabilidad.

- **Compactibilidad (Cp).** Esta métrica se puede aplicar a todos los NADs, pero un prerrequisito para el cálculo es hallar la matriz de distancias convertidas. A continuación solo se detalla el cálculo para el NAD-0, mostrando para los demás el resultado final tras aplicar la fórmula.
 - o $Cp(NAD-0) = 0.42$, ya que asignando letras de la “A” a la “H” a los nodos del NAD ($A=home, B=clientes, C=error, D=home_restringido, E=Projects, F=Reports, G=Contacts, H=Notes$), se podría hallar la matriz de distancias convertidas contando la distancia mínima de un nodo X a otro nodo Y. En caso no ser alcanzables, el valor a rellenar (factor k) se corresponde el número total de nodos (en este caso $k=n=8$). Sumando cada fila se obtiene la distancia de salida convertida (*Converted Out Distance - COD*), y sumando cada columnas se obtiene la distancia de entrada convertida (*Converted In Distance - CID*), así pues, sumando todos los CIDs o CODs se obtiene el término $\sum_i \sum_j C_i = 285$ (ver Tabla 4.3). De esta forma, si $Max = (n^2 - n)*k = (8^2-8)*8 = 448$; y $Min = (n^2 - n) = (8^2-8)=56$; ya se puede aplicar la fórmula de la compactibilidad:

$$Cp(NAD-0) = \frac{448-285}{448-56} = 0.42$$

Tabla 4.3. Matriz de distancias convertidas para Cp(NAD-0).

	A	B	C	D	E	F	G	H	COD
A	0	1	2	2	3	3	3	3	17
B	2	0	1	1	2	2	2	2	12
C	1	2	0	3	4	4	4	4	22
D	1	2	3	0	1	1	1	1	10
E	8	8	8	8	0	8	8	8	56
F	8	8	8	8	8	0	8	8	56
G	8	8	8	8	8	8	0	8	56
H	8	8	8	8	8	8	8	0	56
CID	36	37	38	38	34	34	34	34	285

- o $Cp(NAD-1) = 0.79$
- o $Cp(NAD-2) = 0.58$
- o $Cp(NAD-3) = 0.26$

Puesto que todos los valores pertenecen al intervalo $[0.2 < Cp \leq 0.8]$ no existen problemas de usabilidad.

Modelo de presentación: los artefactos correspondientes a este modelo son APD-0 (ver Fig. 4.6), APD-1 (ver Fig. 4.8), APD-2 (ver Fig. 4.10) y APD-3 (ver Fig. 4.12). En estos artefactos se han aplicado las métricas:

- **Proporción de nombres adecuados para enlaces (PNAE).** Esta métrica se puede aplicar a todos los APDs:
 - $PNAE(APD-0) = 7/7 = 1$, ya que los siete enlaces existentes: *Enter, return, Tasks, Reports, contacts, whats new, y exit*; indican de forma clara y concisa sus respectivos destinos.
 - $PNAE(APD-1) = 11/14 = 0.78$, ya que de los catorce enlaces existentes: *New folder, All tasks, Pending tasks, ..., New comment, Upload file, New(Fig. 4.8(e)), New (Fig. 4.8(f)), New (Fig. 4.8(g))*; los últimos enlaces *new* no indican de forma clara que la operación a realizar es una confirmación de los datos introducidos en el formulario.
 - $PNAE(APD-2) = 4/5 = 0.8$, ya que de los cinco enlaces: *New (Fig. 4.10(a)), View all contacts, Search (x2), y New (Fig. 4.10(d))*; sólo el último *New* no indica de forma clara que la operación a realizar es una confirmación de los datos introducidos en el formulario.
 - $PNAE(APD-3) = 7/8 = 0.87$, ya que de los siete enlaces: *user_name, search (x2), View today report, report title, title and autor, y New*; sólo el último *New* no indica de forma clara que la operación a realizar es una confirmación de los datos introducidos en el formulario. Por lo tanto aquí se detecta el problema de usabilidad UP004.

Por lo tanto, estamos a ante el problema de usabilidad leve UP001 ya que los valores $PNAE(APD-1)$, $PNAE(APD-2)$ y $PNAE(APD-3)$ pertenecen al mismo intervalo $[0.7 < PNAE \leq 0.99]$. La Tabla 4.4 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.4. Problema de usabilidad detectado: UP002.

ID	UP002
Descripción	Los enlaces new asociados a las interfaces de usuario abstractas no indica de forma clara que la operación a realizar es una confirmación de los datos introducidos en el formulario. Pudiendo confundirse con un enlace a otro formulario vacío para insertar datos.
Atributo Afectado	2. Facilidad de aprendizaje / 2.1. Predictibilidad / 2.1.1. Nombres de enlaces significativos
Nivel de Criticidad	Leve
Artefactos evaluados	Modelo de Presentación: APD-1 (0.78); APD-2 (0.8); APD-3(0.87)
Orígenes del problema	NAD-1, NAD-2, NAD-3 respectivamente.
Ocurrencias	5 (3 en APD-1; 1 en APD-2; 1 en APD-3.)

Recomendaciones	Cambiar el nombre del enlace New, por otro más significativo como por ejemplo: "Accept" o "Confirm" (Aceptar o Confirmar) en los alias que se asignan a los enlaces en el modelo navegacional, manteniéndolo por trazabilidad en el APD.
------------------------	--

- **Proporción de mensajes de aviso significativos (PMAS).** Esta métrica se puede aplicar en los siguientes APDs, por contener una página abstracta de que muestra un mensaje de aviso:
 - PMAS (APD-1) = $1/1 = 1$, ya que único mensaje de aviso existente en la página abstracta c) de la Figura 4.8: *"The selected folder does not have any task associated to it by the moment"* es conciso y claro.
 - PMAS (APD-2) = $1/1 = 1$, ya que único mensaje de aviso existente en la página abstracta c) de la Figura 4.10: *"No results has been found"* es conciso y claro.
 - PMAS (APD-3) = $1/1 = 1$, ya que único mensaje de aviso existente en la página abstracta c) de la Figura 4.12: *"No results has been found"* es conciso y claro.

Puesto que todos los valores pertenecen al intervalo $[0.99 < PMAS \leq 1]$ no existen problemas de usabilidad.

- **Contraste de color (CC).** Esta métrica se puede aplicar a todos los elementos de cada APDs. Debido a que la mayoría de los resultados tras aplicar la métrica (en esta aplicación Web en concreto) han pertenecido al intervalo correspondiente a $[CC \geq 500]$, a continuación se muestra el único cálculo que ha generado un problema de usabilidad
 - CC (Etiquetas *"Telephones and mails"* en el APD-2) = 332, ya que los valores RGB para los atributos *Forecolor* y *Backcolor* de esta etiqueta son (33, 85, 189) y (192, 192, 255) respectivamente. Por lo tanto $|33-192| + |85-192| + |189-255| = 332$.
 - CC (Etiqueta *"Address"* en el APD-2) = 332, ya que los valores RGB para los atributos *Forecolor* y *Backcolor* de esta etiqueta son (33, 85, 189) y (192, 192, 255) respectivamente. Por lo tanto $|33-192| + |85-192| + |189-255| = 332$.

Por lo tanto, estamos a ante el problema de usabilidad Medio UP003 ya que ambos valores pertenecen al intervalo $[0 < CC < 500]$. La tabla 4.5 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.5. Problema de usabilidad detectado: UP003.

ID	UP003
Descripción	Las etiquetas “ <i>Telephones and mails</i> ” y “ <i>Address</i> ” no son legibles debido al contraste entre su color de texto y su color de fondo.
Atributo Afectado	1. Facilidad de entendimiento / 1.1. Legibilidad visual / 1.1.2. Adecuación de la visualización textual
Nivel de Criticidad	Medio
Artefactos evaluados	Modelo de Presentación: APD-2 (332)
Orígenes del problema	APD-2
Ocurrencias	2 en APD-2
Recomendaciones	Asignar otro valor al atributo Forecolor con menos componente de color azul en ambas etiquetas para mejorar el contraste.

– **Enlaces distinguibles (ED)**. Esta métrica se puede aplicar a todos los APDs que contienen enlaces y etiquetas:

- $ED(APD-0) = 1 - 0.5 \left(\frac{3}{6} + \frac{0}{7} \right) = 0.78$, ya que existen tres enlaces: *Task*, *Contacts* y *Report* (Fig 4.6 (c)) de los seis posibles (*Return*, *Tasks*, *Contacts*, *Reports*, *Whats new*, *Exit*) que son confundibles con etiquetas al no tener la misma tipografía, mientras que ninguna de las siete etiquetas existentes (*Access control*, *login*, *password*, “*Admin profile ..*”, “* Error ...*”, *Connected as*, *EN2*) se confunde con un enlace.
- $ED(APD-1) = 1 - 0.5 \left(\frac{1}{14} + \frac{11}{40} \right) = 0.82$, ya que existe un enlace confundible con una etiqueta (*name_and_status*) y once etiquetas confundibles con enlaces (*FOLDERS*, *Tasklist*, *EN1*, *assigned_user*, *unassigned_user*, *desc_priority*, *task name*, *start_date*, *end_date*, *description*, *comment_owner*).
- $ED(APD-2) = 1 - 0.5 \left(\frac{1}{3} + \frac{7}{35} \right) = 0.73$, ya que existe un enlace confundible con una etiqueta (*email_clickable*) y siete etiquetas confundibles con enlaces (*CONTACTS*, *company*, *telephone*, *mobile*, *name and surname*, *address*, *telephone and emails*).
- $ED(APD-3) = 1 - 0.5 \left(\frac{2}{5} + \frac{3}{16} \right) = 0.71$, ya que existen dos enlaces confundibles con etiquetas (*title and autor*, *report_title*) y tres etiquetas confundibles con enlaces (*date*, *content*, *BD_content*).

Por lo tanto, estamos a ante el problema de usabilidad leve UP004 ya que los valores $ED(APD-1)$, $ED(APD-2)$ y $ED(APD-3)$ pertenecen al mismo intervalo $[0.7 < ED \leq 0.99]$. La Tabla 4.6 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.6. Problema de usabilidad detectado: UP004.

ID	UP004
Descripción	Los enlaces: <i>Task</i> , <i>Contacts</i> y <i>Report</i> (APD-0); <i>name_and_status</i> (APD-1); <i>email_clickable</i> (APD-2); <i>title_and_autor</i> , <i>report_title</i> (APD-3) se confunden con etiquetas textuales. Las etiquetas: <i>FOLDERS</i> , <i>Tasklist</i> , <i>EN1</i> , <i>assigned_user</i> , <i>unassigned_user</i> , <i>desc_priority</i> , <i>task name</i> , <i>start_date</i> , <i>end_date</i> , <i>description</i> , <i>comment_owner</i> (APD-1); <i>CONTACTS</i> , <i>company</i> , <i>telephone</i> , <i>mobile</i> , <i>name and surname</i> , <i>address</i> , <i>telephone and emails</i> (APD-2); <i>date</i> , <i>content</i> , <i>BD_content</i> (APD-3) se confunden con enlaces
Atributo Afectado	1. Facilidad de entendimiento / 1.6. Navegabilidad / 1.6.2. Clickabilidad
Nivel de Criticidad	Leve
Artefactos evaluados	Modelo de Presentación: APD-0 (0.78); APD-1(0.82); APD-2 (0.73); APD(0.71)
Orígenes del problema	APD-0, APD-1, APD-2, APD-3
Ocurrencias	3 en APD-0, 12 en APD-1, 8 en APD-2, 5 en APD-3
Recomendaciones	Asignar otra tipografía o color a los enlaces y etiquetas problemáticas para diferenciarlos del resto. En los enlaces se recomienda emplear la misma que el resto (color azul y subrayado)

- **Proporción de imágenes con texto alternativo asociado (PIT)**. Esta métrica se puede aplicar a todos los APDs que contengan imágenes insertadas como elemento extra en sus páginas abstractas. En la figuras asociadas a los APD no se puede apreciar el valor de la propiedad “*text*” asociado a cada imagen, sin embargo, este valor recibe por defecto el nombre del archivo de imagen asociado, permitiendo que las imágenes contengan un texto alternativo por defecto. De esta forma:
 - $PIT(APD-0) = 2/2 = 1$, imágenes: llave y exclamación.
 - $PIT (APD-1) = 13/13 = 1$, imágenes: icono portafolio (x3), icono carpeta (x8), icono exclamación, icono zipfile.
 - $PIT (APD-2) = 4/4 = 1$, imágenes: icono tarjeta contacto, icono avatar contacto, icono exclamación.
 - $PIT (APD-3) = 4/4 = 1$, imágenes: icono usuario, icono informe, icono exclamación, icono flecha.

Puesto que todos los valores pertenecen al intervalo $[0.99 < PIT \leq 1]$ no existen problemas de usabilidad. Además, nos encontramos ante un atributo de usabilidad soportado automáticamente en tiempo de modelado.

- **Variación en la composición de los marcos (VCM)**. Está métrica se puede aplicar en el modelo de presentación analizando la plantilla *Tlayout* que define la composición de los marcos (ver Fig. 4.13). En ella se observa que existen dos

composiciones distintas de marcos: una para el acceso de la aplicación (un único marco L1), y otra para la funcionalidad de la aplicación Web (marcos L2, L3 y L4). Este valor pertenece al intervalo $[0 < VCM \leq 2]$, por lo tanto no se contempla ningún problema de usabilidad. Además, el hecho de usar una única plantilla asegura que no existan más variaciones en toda la aplicación Web, encontrándonos ante un atributo de usabilidad soportado automáticamente en tiempo de modelado.

Como consecuencia de la evaluación a nivel PIM, se obtiene el Informe de problemas de usabilidad independiente de plataforma que estará compuesto por los problemas: UP001, UP002, UP003 y UP004.

4.3.4.2. EVALUACIÓN A NIVEL CM

En esta evaluación se aplicarán las métricas seleccionadas sobre las interfaces de usuario finales: IUF-1, IUF-2, IUF-3.

- **Proporción de elementos que muestran estado actual.** Esta métrica puede aplicarse en todas las interfaces de usuario finales:
 - o $PEEA(IUF-1, IUF-2, IUF-3) = 1/3 = 0.33$, ya que el único tipo de elemento que muestra el estado de la interfaz son los campos del formulario, en los cuales el cursor parpadea ayudando a localizar el centro de atención del usuario (propiedad ya ofrecida por el propio navegador Web). Con respecto a los otros dos tipos de elementos: el menú de navegación no muestra en qué sección de la aplicación Web se encuentra y los enlaces no se resaltan cuando el cursor pasa por encima de ellos, hecho que puede desorientar al usuario al existir enlaces en el sub-menú con poca separación.

Por lo tanto, estamos ante el problema de usabilidad grave UP005 ya que el resultado de la métrica pertenece al intervalo $[0 < PEEA \leq 0.5]$. La Tabla 4.7 muestra la plantilla rellenada para este problema de usabilidad detectado.

Tabla 4.7. Problema de usabilidad detectado: UP005.

ID	UP005
Descripción	Ni el menú de navegación muestra en qué sección de la aplicación Web se encuentra, ni los enlaces se resaltan cuando el cursor pasa por encima de ellos.
Atributo Afectado	1. Facilidad de entendimiento / 1.5. Orientación al usuario / 1.5.3. Retro-alimentación inmediata
Nivel de Criticidad	Grave
Artefactos evaluados	IUF-1 (0.33); IUF-2 (0.33); IUF-3(0.33)
Orígenes del problema	Regla de transformación que establecen el mapeo entre el menú de navegación del modelo de presentación y el componente específico

	de plataforma que muestra el menú. Regla de generación de código que establece el mapeo entre los enlaces representados en el modelo de presentación con sus estilos en el código fuente final
Ocurrencias	1 en IUF-1, 1 en IUF-2, 1 en IUF-3
Recomendaciones	Elegir otro componente destino para el menú de navegación e incluir reglas de generación de código que proporcionen el resaltado de los enlaces.

- **Número de elementos no alineados o desencuadrados.** Esta métrica puede aplicarse en todas las interfaces de usuario finales:
 - NED (IUF-1, IUF-2, IUF-3)= 1, ya que el contenido del marco L3 según la plantilla *tLayout* que representa la sección de la aplicación Web donde se muestra el contenido de la misma, sobrepasa las dimensiones establecidas por el marco, haciendo que la visión global de la interfaz de usuario esté desencuadrada.

Por lo tanto, estamos a ante el problema de usabilidad grave UP006 ya que el resultado de la métrica pertenece al intervalo [$NED > 0$]. La Tabla 4.8 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.8. Problema de usabilidad detectado: UP006.

ID	UP006
Descripción	La sección que engloba el contenido de la aplicación Web excede de las dimensiones del marco.
Atributo Afectado	6. Grado de atracción / 6.3 Uniformidad en la posición de las secciones de la interfaz
Nivel de Criticidad	Grave
Artefactos evaluados	IUF-1 (1); IUF-2 (1); IUF-3(1)
Orígenes del problema	Modelo de presentación: plantilla Tlayout
Ocurrencias	1 en IUF-1, 1 en IUF-2, 1 en IUF-3
Recomendaciones	Fijar unos valores máximos y mínimos en todos los marcos para que no haya desajustes en las dimensiones.

- **Operaciones de usuario cancelables.** Esta métrica puede aplicarse en todas las interfaces de usuario finales:
 - OUC (IUF-1)= $0/3 = 0$, ya que existen tres operaciones que requieren de la capacidad de ser canceladas: *Nueva Tarea*, y otras dos que no se muestran explícitamente en la interfaz de usuario final: *Subir fichero* y *Añadir Comentario*. Ninguna de ellas tiene asociada la opción Cancelar, obligando al usuario a emplear la tecla “Atrás” de su navegador pudiendo afectar a la consistencia del estado de la aplicación Web.

- OUC (IUF-2)= 0/1 = 0, ya que existe la operación *Nuevo contacto* sin soporte a la cancelación.
- OUC (IUF-3)= 0/1 = 0, ya que existe la operación *Nuevo informe* sin soporte a la cancelación.

Por lo tanto, estamos a ante el problema de usabilidad grave UP007 ya que el resultado de la métrica pertenece al intervalo $[0.0 \leq OUC \leq 0.5]$. La Tabla 4.9 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.9. Problema de usabilidad detectado: UP007.

ID	UP007
Descripción	No existe soporte de cancelación para las operaciones: <i>Nueva tarea, Subir fichero, Añadir comentario, Nuevo contacto, nuevo informe</i>
Atributo Afectado	3. Facilidad de uso / 3.3 Controlabilidad / 3.3.2 Soporte a operaciones de cancelación
Nivel de Criticidad	Grave
Artefactos evaluados	IUF-1 (1); IUF-2 (1); IUF-3(1)
Orígenes del problema	Modelo de navegación: NAD-1, NAD-2 y NAD-3 respectivamente.
Ocurrencias	3 en IUF-1, 1 en IUF-2, 1 en IUF-3
Recomendaciones	Definir caminos de navegación hacia estados anteriores que permitan al usuario regresar si ha navegado hacia una operación por error.

- **Proporción de mecanismos de validación de datos de entrada.** Esta métrica puede aplicarse en todas las interfaces de usuario finales:
 - PMV (IUF-1)= 4/4 = 1, ya que los campos del formulario: *priority, assigned user, Begin date* y *End date* son los candidatos a tener un formato válido para ser aceptado por la aplicación Web (valores pertenecientes a un conjunto definido o fechas). Todos poseen mecanismos de validación de datos de entrada: para los valores de un conjunto se provee un listbox y para las fechas se provee un calendario.
 - PMV (IUF-2)= 1/1 = 1, ya que existe un único campo del formulario candidato: *Initial*. Para seleccionar la inicial del contacto a buscar se provee un listbox.
 - PMV (IUF-3)= 0/2 = 0, ya que los campos: *From* y *To* hacen referencia a fechas y no se provee ningún calendario o etiqueta que ayude al usuario a introducir el dato de forma correcta.

Por lo tanto, estamos a ante el problema de usabilidad grave UP008 ya que el resultado de la métrica pertenece al intervalo $[0.0 \leq PMV \leq 0.5]$. La Tabla 4.10 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.10. Problema de usabilidad detectado: UP008.

ID	UP008
Descripción	Los campos del formulario From y To no poseen ningún mecanismo de validación de datos de entrada.
Atributo Afectado	3. Facilidad de uso / 3.2 Gestión de los datos / 3.2.1 Validez de los datos de entrada
Nivel de Criticidad	Grave
Artefactos evaluados	IUF-3
Orígenes del problema	Modelo de presentación: APD-3
Ocurrencias	2 en IUF-3
Recomendaciones	Definir en el modelo de presentación que esos campos son del tipo Fecha para proveer automáticamente un mecanismo de validación de datos de entrada.

- **Diferencias de comportamiento y aspecto de elementos de interfaz entre navegadores.** Esta métrica puede aplicarse en todas las interfaces de usuario finales, para ello se han comparado las interfaces de usuario final en los navegadores que se especificaron en el perfil de la evaluación: Internet Explorer y Mozilla Firefox (ver Fig.4.17).

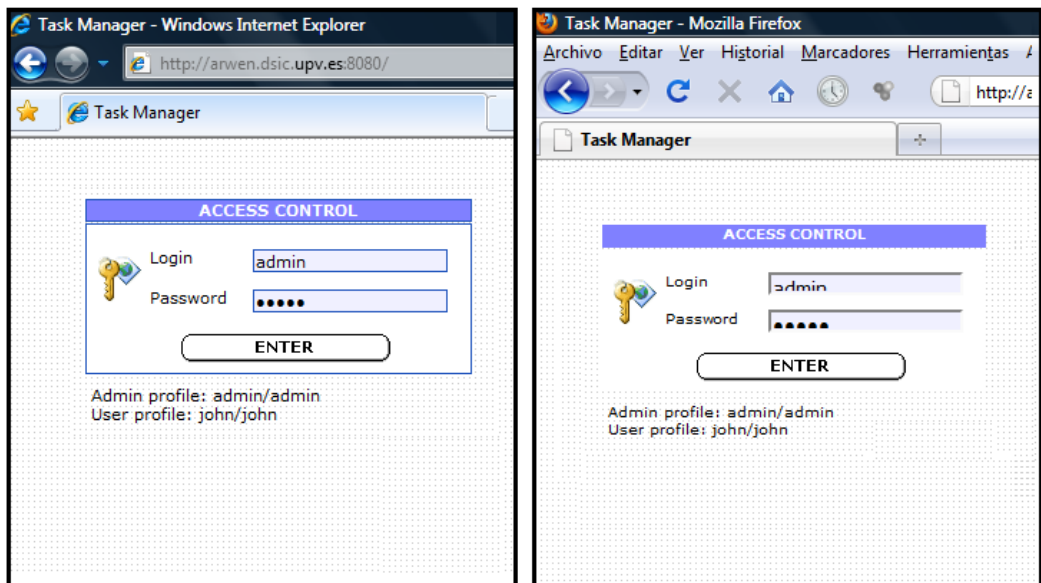


Fig 4.17. IUF: Interfaz de usuario final en distintos navegadores

- o DCN (IUF)= 2, ya que se detectaron dos diferencias principales: En Mozilla Firefox, los campos de todos los formularios no muestran de forma claramente legible su contenido, y los mecanismos de validación de entrada de datos para las fechas no muestran su funcionalidad. Sin embargo, en Internet Explorer no se presenta ningún problema.

Por lo tanto, estamos a ante el problema de usabilidad grave UP009 ya que el resultado de la métrica pertenece al intervalo $[DCN > 0]$. La Tabla 4.11 muestra la plantilla rellena para este problema de usabilidad detectado.

Tabla 4.11. Problema de usabilidad detectado: UP009.

ID	UP009
Descripción	Cuando se usa el navegador Mozilla Firefox, el contenido de los campos de los formularios es ilegible, además, la funcionalidad del calendario no funciona correctamente.
Atributo Afectado	3. Facilidad de uso / 3.1 Compatibilidad / 3.1.1 Compatibilidad con navegadores y plugins
Nivel de Criticidad	Grave
Artefactos evaluados	IUF
Orígenes del problema	Reglas de transformación que mapean las definiciones de los campos de formularios presentes en el modelo de presentación con los componentes específicos de plataforma que representan dichos campos. Reglas de generación de código que permiten el funcionamiento de los mecanismos basados en calendarios.
Ocurrencias	En toda la interfaz de usuario
Recomendaciones	Elegir otro componente destino para los campos del formulario que se adapten mejor a la plataforma específica. Mejorar la regla de generación de código para incluir la funcionalidad de los calendarios en distintos navegadores.

Como consecuencia de la evaluación a nivel CM, se obtiene el Informe de problemas de usabilidad de la aplicación Web final que estará compuesto por los problemas: UP005, UP006, UP007, UP008 y UP009.

4.3.4.3. ANÁLISIS DE CAMBIOS

Esta fase ya se disponen de los Informes de Problemas de Usabilidad a nivel PIM (UP001 a UP004) y a nivel CM (UP005 a UP009). Por lo tanto, podemos clasificar los problemas de usabilidad según los artefactos que lo han originado para componer los Informes de Problemas de Usabilidad que retro-alimentarán cada fase del proceso desarrollo Web. De esta forma, obtenemos los siguientes informes:

- Informe de Problemas de Usabilidad en Análisis: UP001 (Modelo de clases), UP002 y UP007 (Modelo navegacional).
- Informe de Problemas de Usabilidad en Diseño: UP003, UP004, UP006 y UP008 (Modelo de presentación).
- Informe de Problemas de Usabilidad en Transformación de Modelos: UP005 y UP009.

- Informe de Problemas de Usabilidad en Generación de Código: UP005 y UP009.

Puesto que una de las restricciones de la evaluación era el hecho de no tener acceso a las reglas de transformación de modelos y generación de código que provee el compilador de modelos proporcionado por VisualWade para el método OO-H, el objetivo de esta fase se centrará a realizar los cambios pertinentes en los modelos PIM, de análisis y diseño, atendiendo a las recomendaciones de cada problema de usabilidad detectado, con el objetivo de obtener una aplicación Web más usable.

Cambios en el Modelo de clases: El principal cambio en este modelo proviene del UP001 donde se recomienda asignar valores por defecto a aquellos atributos que requieren de esta propiedad. Para ello, es suficiente con asignar dichos valores mediante el cuadro de propiedades de atributo que ofrece VisualWade (ver Fig. 4.18).

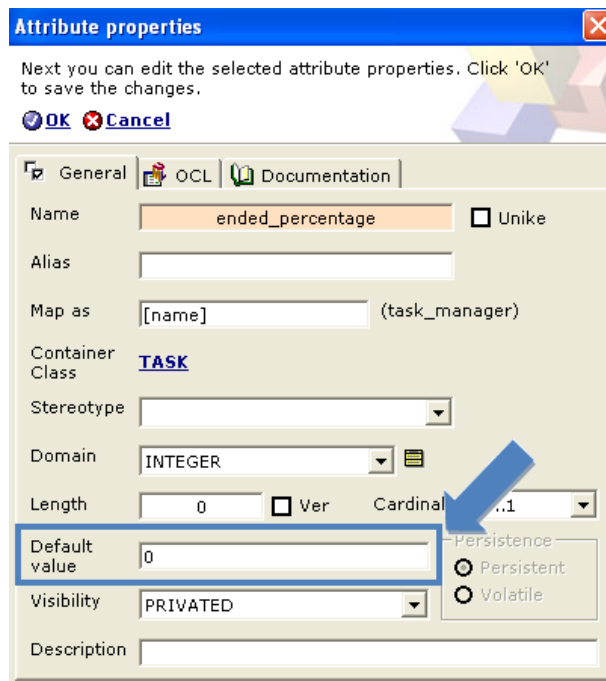


Fig 4.18. Cambios realizados en el modelo de clases

Cambios en el Modelo navegacional: Los cambios en los NADs provienen de los problemas UP002 y UP007. El UP002 recomendaba cambiar los nombres de los enlaces *New* para que resultaran más significativos. Para ello, es suficiente con asignar un alias a los enlaces del modelo navegacional (ver Fig. 4.19 (a)).

Mientras que el UP007 recomendaba definir caminos de navegación hacia estados anteriores que permitan al usuario regresar si ha navegado hacia una operación por error. Para ello es suficiente con crear un enlace de retorno desde la clase navegacional afectada (ver Fig. 4.19 (b)).

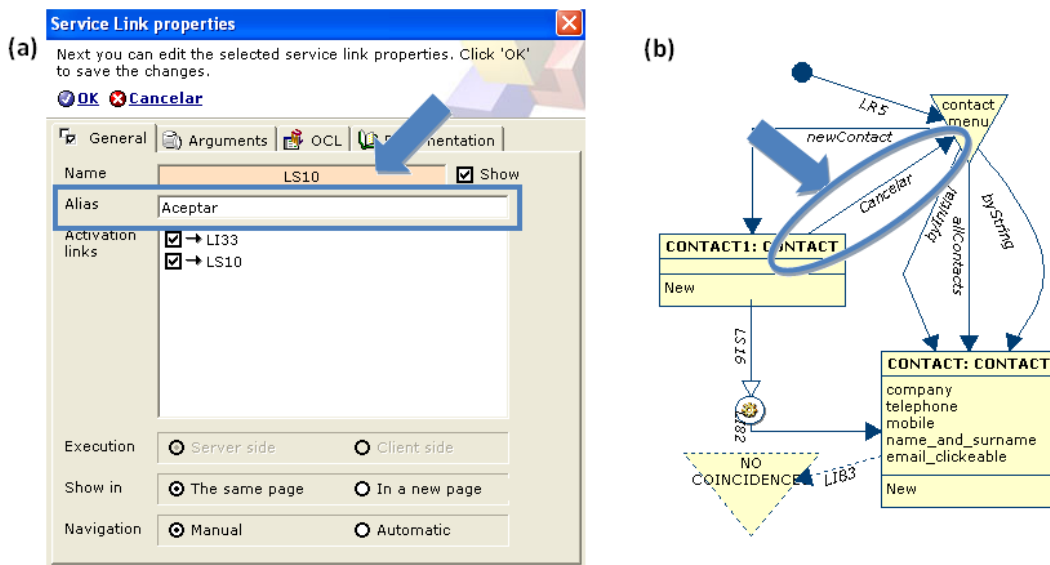


Fig 4.19. Cambios realizados en el modelo de navegación

Cambios en el Modelo de presentación: Los cambios en los APDs provienen de los problemas UP003, UP004, UP006 y UP008. Todos ellos pueden solucionarse modificando las propiedades de los elementos de la interfaz mediante la interfaz de usuario que ofrece VisualWade para el XML subyacente.

El UP003 recomendaba asignar otro valor al atributo *Forecolor* con menos componente de color azul en las etiquetas *Telephone and mail* y *Address* del APD-2 para mejorar el contraste (ver Fig. 4.20 (a)).

El UP004 recomendaba asignar otra tipografía o color a los enlaces y etiquetas problemáticas para diferenciarlos del resto (ver Fig. 4.20 (b)).

El UP008 recomendaba definir en el modelo de presentación que los campos *From* y *To* del APD-3 son del tipo "Fecha" para proveer automáticamente un mecanismo de validación de datos de entrada (ver Fig. 4.20 (c)).

El UP006 recomendaba fijar unos valores máximos y mínimos en todos los marcos para que no haya desajustes en las dimensiones (ver Fig. 4.20 (d)).

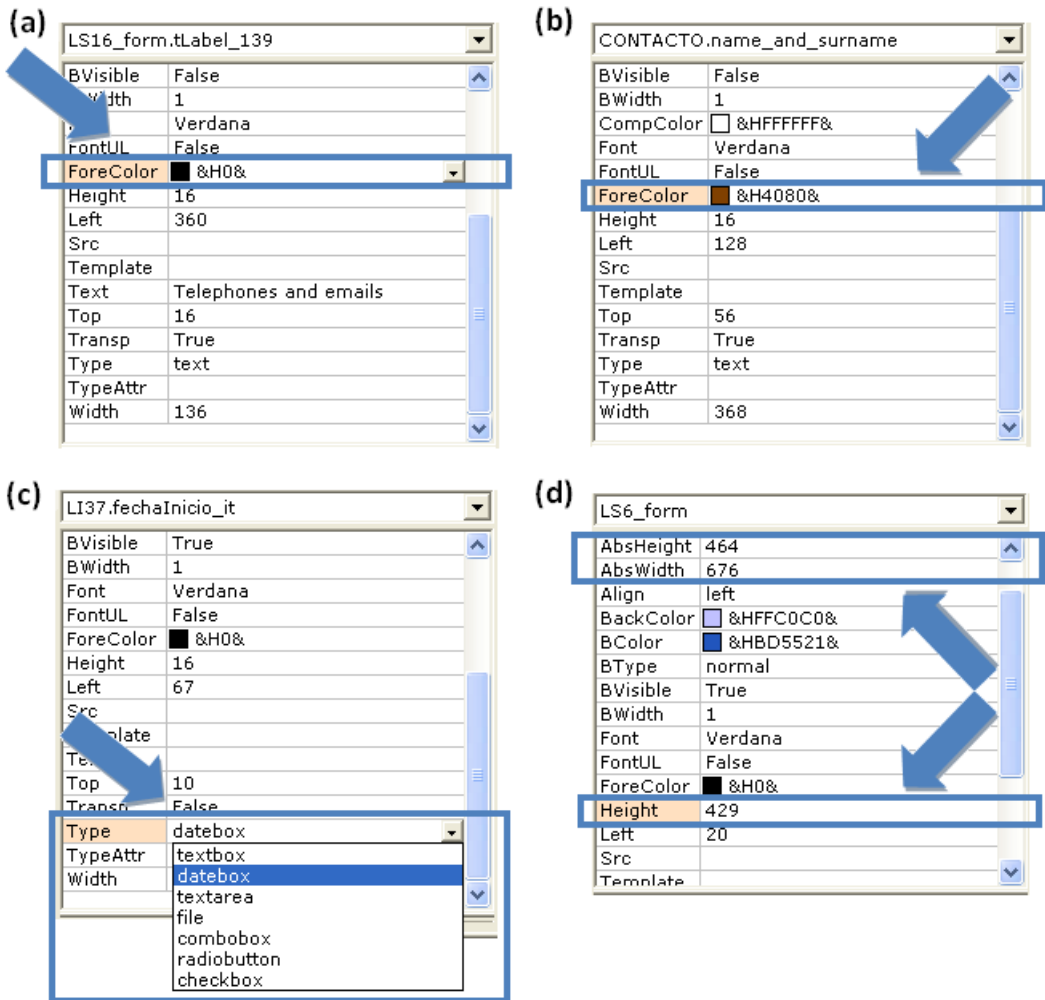


Fig 4.20. Cambios realizados en el modelo de presentación

De esta forma, volviendo a compilar los modelos obtendremos el código fuente de una aplicación Web más usable.

4.4. LECCIONES APRENDIDAS

La instanciación de WUEP en un método de desarrollo Web dirigido por modelos concreto como es OO-H, con el objetivo de evaluar una determinada aplicación Web, ha proporcionado una serie de experiencias de las cuales es posible extraer una serie de lecciones aprendidas.

En primer lugar, se hace patente la factibilidad de evaluar la usabilidad de una aplicación Web en distintas fases del proceso de desarrollo, cuando se proporcionan artefactos intermedios (modelos) que especifican la aplicación Web a distintos niveles de abstracción. Por este motivo, integrando WUEP en procesos de desarrollo Web

dirigido por modelos se cubre una de las necesidades más relevantes que fueron halladas mediante la revisión sistemática de métodos de evaluación de usabilidad Web: la usabilidad se tiene en cuenta durante todo el proceso de desarrollo, especialmente, en fases tempranas del desarrollo permitiendo predecir errores de usabilidad que la mayoría de métodos detecta cuando la aplicación Web está prácticamente finalizada. De esta forma la usabilidad de aplicaciones Web se obtiene mediante construcción, no mediante mantenimiento del código fuente final.

Aunque en la evaluación de la aplicación TaskManager sólo se hayan evaluado un conjunto determinado de atributos, durante el desarrollo de la aplicación Web siguiendo una estrategia de inspección, el Modelo de Usabilidad Web en el que cual se basa WUEP ofrece un catálogo extenso que permite cubrir el concepto de usabilidad Web de forma bastante completa, tanto en su perspectiva orientada hacia el producto software (mediante la aplicación de métricas en una estrategia basada en inspección de artefactos), como en su perspectiva orientada a la calidad en uso (mediante el cálculo de métricas que requieren la participación del usuario final). Esta característica diferencia a WUEP de otros métodos de evaluación de usabilidad Web que sólo tienen en consideración un conjunto muy limitado de atributos. De esta forma, WUEP se convierte en un proceso de evaluación versátil, englobando las ventajas de los métodos de inspección y testeo de usuarios. La completitud de la evaluación recae en la configuración de la misma por parte del diseñador de la evaluación, y vendrá condicionada por la selección de aquel conjunto de atributos de usabilidad que sean prioridad para la aplicación Web a evaluar teniendo en cuenta los perfiles de evaluación definidos: perfil del usuario, tipo de aplicación, contexto de uso y método de desarrollo.

WUEP ofrece al evaluador, una metodología de evaluación guiada y detallada al especificar los pasos a seguir en cada fase de la evaluación mediante la notación SPEM, la cual está basada en el estándar propuesto por OMG. Una de las características más importantes de WUEP reside en ser un proceso genérico el cual puede ser instanciado en cualquier método de desarrollo Web dirigido por modelos, ya que la fase de operacionalización de métricas, permite establecer un mapeo entre los atributos de usabilidad y las primitivas de modelado del método en concreto, empleando para ello la definición genérica de las métricas del Modelo de Usabilidad Web. Como consecuencia añadida, el hecho de poder operacionalizar las métricas a distintos niveles de abstracción, hace que WUEP no sólo sea compatible con los métodos de desarrollo Web dirigido por modelos, sino que además, podría aplicarse para la evaluación de aplicaciones Web finales si las métricas se instanciaran a nivel CM, pese a que de esta forma, la retro-alimentación a las fases más tempranas no sería tan efectivo al no existir tales artefactos intermedios.

Gracias a la aplicación de métricas operacionalizadas, y a la trazabilidad entre los modelos que definen la aplicación Web, no solo es posible ofrecer una lista de problemas de usabilidad en base a intervalos ofrecidos para cada métrica, sino además podemos deducir el origen del problema y ofrecer recomendaciones (en algunos casos incluso automáticas) que ayuden en su prevención y/o corrección, cubriendo de esta forma, otra de las necesidades halladas en la revisión sistemática: la falta de métodos de evaluación de usabilidad que proporcionen sugerencias al desarrollador. Además, se puede observar la existencia de métricas cuya automatización es factible gracias a lenguajes de restricciones sobre modelos como OCL, como por ejemplo la amplitud, profundidad y densidad de la navegación; o bien mediante analizadores XML para calcular automáticamente métricas que requieren de propiedades asociadas a elementos de presentación si estos se definen mediante plantillas XML. Sin embargo, no todas las métricas pueden ser calculadas automáticamente, sobre todo en aquellas que requieren una interpretación por parte del evaluador, como por ejemplo, la proporción de mensajes de aviso significativos.

Otro de los aspectos observados hace referencia a que la utilidad de WUEP no sólo reside en realizar evaluaciones, sino en generar una base de conocimiento reutilizable, ya que la operacionalización de las métricas es un activo reutilizable en la evaluación de distintas aplicaciones Web que han sido desarrolladas empleando el mismo método de desarrollo Web dirigido por modelos. Además, se observó que algunos atributos de usabilidad, como la uniformidad de los marcos de la interfaz, son soportados automáticamente por el método de desarrollo Web, esto pone de manifiesto la utilidad de WUEP para descubrir limitaciones en la expresividad de los modelos independientes de plataforma, o reglas de transformación, pertenecientes a dicho método de desarrollo concreto, es decir, los modelos podrían incorporar primitivas de modelado que aseguraran, en tiempo de modelado, ciertos atributos de usabilidad, como por ejemplo, incluir un mecanismo que permita agrupar distintas etiquetas de una interfaz abstracta de usuario para asegurar la agrupación cohesiva de la información (ver atributo 1.2.1) o incluir reglas en la generación de código que provea a la aplicación Web de soporte a búsqueda interna (ver atributo 1.6.1) o soporte a la ampliación de texto (ver atributo 5.1).

5. CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se recogen las conclusiones generales de este trabajo, los trabajos futuros propuestos para dejar constancia de la continuidad del trabajo de investigación, y los resultados obtenidos en cuanto a publicaciones.

5.1. CONCLUSIONES

Las aplicaciones Web son consideradas actualmente como un elemento esencial e indispensable en toda actividad empresarial, intercambio de información y motor de redes sociales. La usabilidad de este tipo de aplicaciones es considerada como uno de los factores clave más importantes, puesto que la facilidad o dificultad que los usuarios experimentan con estas aplicaciones determinan en gran medida su éxito o fracaso.

La motivación principal en la que se ubica esta tesina de máster, reside en la problemática existente a la hora de realizar evaluaciones de usabilidad Web, ya que la gran mayoría de propuestas:

- Sólo tienen en cuenta aspectos muy concretos, sin llegar a contemplar una definición completa del concepto usabilidad.
- Plantean la evaluación en las fases finales del desarrollo, donde se requiere invertir más recursos en el mantenimiento de la aplicación Web final.

A todo esto se suma el hecho que los artefactos intermedios producidos en la mayoría de los procesos de desarrollo Web actuales, sirven únicamente para orientar a los desarrolladores y para documentar la aplicación Web. Dado que la trazabilidad entre estos artefactos y la aplicación Web final no está bien definida, evaluar dichos artefactos no resulta eficaz. Este problema no ocurre en el desarrollo de software dirigido por modelos (DSDM) donde los modelos (artefactos intermedios) que especifican una aplicación Web completa se aplican en todas las etapas del proceso de desarrollo, y el código fuente final se genera automáticamente a partir de estos modelos, creando de esta forma un contexto ideal para realizar evaluaciones tempranas de usabilidad.

La problemática anterior es el punto de partida de la contribución principal de la presente tesina de máster, la cual propone el proceso de evaluación WUEP (*Web Usability Evaluation Process*): un proceso de evaluación de usabilidad Web integrado en el desarrollo de software dirigido por modelos.

En el camino hacia la definición de esta propuesta se han ido satisfaciendo las metas planteadas en esta tesina:

1. Analizar en profundidad el empleo de métodos de evaluación de usabilidad: qué tipos de métodos son los más usados, en qué fases se aplican, cuáles han sido

validados, qué información proveen al evaluador y cuáles resultaron más efectivos.

2. Estudiar los estándares de evaluación de calidad para productos software que tratan la usabilidad como una característica de calidad; y analizar otras propuestas de evaluación de usabilidad.
3. Definir un modelo de usabilidad Web que descompone la usabilidad en subcaracterísticas, atributos medibles y métricas. Siendo éste acorde a los estándares más recientes, teniendo en cuenta las distintas definiciones de usabilidad.
4. Definir un proceso de evaluación de usabilidad Web genérico con la capacidad de ser integrado en diferentes métodos de desarrollo Web dirigido por modelos, haciendo uso del modelo de usabilidad como artefacto principal.
5. Instanciar el proceso de evaluación de usabilidad Web a un método concreto de desarrollo Web dirigido por modelos.

Con respecto a la meta 1, se realizó una revisión sistemática sobre métodos de evaluación de usabilidad para la Web, en la cual se incluyeron 163 artículos de entre 847 potenciales. La aplicación de este método científico permitió recopilar información relevante acerca de los aspectos cómo: qué métodos se han creado expresamente para el dominio Web, qué tipos de métodos son los más comunes y qué ventajas aportan, en qué fases del proceso de desarrollo Web suelen aplicarse, qué métodos han sido validados y qué información aportan al evaluador. Tras la revisión se extrajeron una serie de conclusiones que confirmaron la problemática existente en la evaluación de la usabilidad Web y han servido de ayuda para guiar el desarrollo de la contribución de esta tesina:

- Los métodos de evaluación de usabilidad Web evalúan diferentes aspectos del concepto de usabilidad.
- La mayoría de las evaluaciones tienen lugar en la fase de implementación, siendo muy frecuente realizar evaluaciones sólo en una única fase de desarrollo.
- La mayoría de los métodos de evaluación de usabilidad Web sólo proporcionan una lista de problemas de usabilidad, no suelen proporcionar recomendaciones o sugerencias para ayudar a corregirlos.
- Existe una escasez de métodos de evaluación de usabilidad Web que automaticen gran parte de la evaluación y cuya efectividad pueda ser verificada mediante experimentos controlados.

Con respecto a la meta 2 y a fin de obtener una definición amplia del concepto de usabilidad, se estudiaron los estándares de evaluación de la calidad para productos software que tratan la usabilidad como una característica de calidad, ya que en los

estándares se suelen encontrar definiciones consensuadas. Este estudio reveló que la norma ISO SQuaRE constituye la definición más completa de usabilidad existente, permitiendo ser una buena base para definir un modelo de usabilidad Web. De esta forma se analizaron las diferentes propuestas que se basan en estándares, haciendo especial énfasis, en aquellas propuestas de evaluación de usabilidad en el contexto del desarrollo Web dirigido por modelos, ya que estas propuestas constituyen las primeras aproximaciones a realizar evaluaciones de usabilidad tempranas. Los resultados de este análisis mostraron que no existe ningún proceso de evaluación de usabilidad genérico, bien definido, que tenga como base la definición de usabilidad propuesta por la norma SQuaRE y que contemple un amplio conjunto de atributos de usabilidad relacionados con las aplicaciones Web, siendo muy escasos los trabajos desarrollados en el contexto DSDM.

Con respecto a la meta 3, se definió un Modelo de Usabilidad Web tomando como punto de partida el modelo de usabilidad propuesto en Abrahão e Insfran [3] para procesos de desarrollo dirigido por modelos. Este modelo se ha extendido y adaptado a productos orientados a la Web y a la última norma ISO 25000 SQuaRE. Para ello, se ha descompuesto la usabilidad en sub-características y atributos medibles teniendo en cuenta criterios ergonómicos y guías de usabilidad Web, dotándolo además, de métricas genéricas asociadas a dichos atributos, que pueden ser operacionalizadas en base al nivel de abstracción de los artefactos a evaluar y a las primitivas de modelado de distintos métodos de desarrollo Web que siguen el enfoque dirigido por modelos. Este Modelo de Usabilidad Web en el cual se basará la propuesta WUEP, ofrece una de las visiones más completas en cuanto al concepto de usabilidad Web se refiere, al incorporar las dos perspectivas de usabilidad: la orientada al producto software y la orientada a la calidad en uso. De esta forma, se pretende englobar las definiciones propuestas por el campo de la Ingeniería Software y las propuestas por el campo de la Interacción Persona-Ordenador.

Con respecto a la meta 4, se definió el proceso de evaluación de usabilidad WUEP empleando para ello la notación SPEM, contribuyendo a proporcionar una especificación guiada y detallada del método para los evaluadores. WUEP utiliza como artefacto principal el Modelo de Usabilidad Web para realizar evaluaciones de los modelos producidos en el proceso desarrollo Web, proporcionando retro-alimentación a distintas fases de éste proceso de desarrollo, especialmente en aquellas fases tempranas. Como punto de partida se tomaron las fases principales del proceso de evaluación de calidad propuesto en SQuaRE, extendiendo y adaptando las tareas incluidas en dichas fases para su posible integración de evaluaciones de usabilidad Web en un entorno de desarrollo DSDM.

WUEP se considera un proceso de evaluación genérico aplicable a cualquier proceso de desarrollo Web dirigido por modelos, debido a que una de sus fases incide en la operacionalización de las métricas genéricas del Modelo de Usabilidad Web, para ser aplicadas a artefactos pertenecientes a distintos niveles de abstracción y a diferentes métodos de desarrollo Web. Esta operacionalización se basa en el establecimiento de un mapeo entre los conceptos medibles por las métricas y las primitivas de modelado de un método de desarrollo Web concreto.

Una de las características más importantes de WUEP es que engloba las ventajas de los métodos de inspección y testeo de usuarios, convirtiéndolo en un proceso de evaluación versátil, englobando las ventajas de los métodos de inspección (para evaluar los artefactos intermedios y la aplicación Web final) y testeo de usuarios (para evaluar la interacción del usuario en un contexto de uso determinado). Sin embargo, la completitud de la evaluación es responsabilidad del diseñador de la evaluación según la configuración de la misma, y vendrá condicionada por la selección de aquel conjunto de atributos de usabilidad que sean prioridad para la aplicación Web a evaluar, teniendo en cuenta los perfiles de evaluación definidos: perfil del usuario, tipo de aplicación, contexto de uso y método de desarrollo.

Por último y con respecto a la meta 5, WUEP se instanció en OO-H [28], un método de desarrollo Web dirigido por modelos, con el objetivo de mostrar su factibilidad a la hora de realizar evaluaciones de usabilidad. Para ello se realizó una evaluación de usabilidad formativa (durante el proceso de desarrollo Web) de un Gestor de tareas desarrollado por una empresa de Alicante, empleando la herramienta VisualWade, la cual da soporte al método OO-H. De la experiencia se extrajeron varias lecciones aprendidas, entre ellas destacamos:

- La factibilidad de las evaluaciones de usabilidad Web en distintas fases del proceso de desarrollo, especialmente en las tempranas, gracias a las ventajas que aporta el contexto del desarrollo dirigido por modelos. De esta forma la usabilidad de una aplicación Web se obtiene por construcción y no por mantenimiento.
- La posibilidad de ofrecer una automatización de gran parte del método, al estar definido en SPEM y al ofrecer la operacionalización de métricas que pueden calcularse automáticamente mediante lenguajes de restricciones de modelos o analizadores XML.
- Mediante la aplicación de métricas operacionalizadas y la trazabilidad entre los modelos que definen la aplicación Web, no sólo es posible ofrecer una lista de problemas de usabilidad en base a los intervalos ofrecidos para cada métrica, sino además es posible deducir el origen del problema y ofrecer recomendaciones.

- Las métricas operacionalizadas pueden reutilizarse en futuras evaluaciones de aplicaciones Web que hayan sido desarrolladas siguiendo el mismo método de desarrollo.
- La operacionalización de las métricas a distintos niveles de abstracción, hace que WUEP no sólo sea compatible con los métodos de desarrollo Web dirigido por modelos, sino que además, con otros métodos de desarrollo si las métricas se instanciaran a nivel CM.
- Algunos atributos de usabilidad se soportan automáticamente por las primitivas de modelado, lo cual pone de manifiesto la utilidad de WUEP para descubrir limitaciones en la expresividad de los modelos y reglas de transformación asociados a un método de desarrollo Web concreto, permitiendo que los métodos ofrezcan primitivas de modelado que tengan mejoren ciertos atributos de usabilidad en fase de modelado.

5.2. TRABAJOS FUTUROS

El trabajo presentado constituye los primeros pasos hacia un proceso de evaluación de usabilidad Web genérico con la capacidad de ser integrado en cualquier proceso de desarrollo Web dirigido por modelos. Sin embargo, existe la necesidad de diversos trabajos futuros para mejorar esta contribución, ya que todavía no se han tenido en cuenta algunas de las conclusiones extraídas de la revisión sistemática, como la automatización del método de evaluación de usabilidad o su validación mediante experimentos controlados. A continuación se presenta la lista de trabajos futuros a desarrollar, ordenados según el orden de aparición de las partes afectadas.

Con respecto a la revisión sistemática, pese a que los resultados obtenidos han sido bastante significativos, éstos se han extraído de artículos orientados al ámbito académico. Por este motivo, se pretende analizar también artículos pertenecientes al sector industrial, para obtener una evidencia más real de las evaluaciones de usabilidad fuera del ámbito académico.

Con respecto al Modelo de Usabilidad Web, éste se pretende mejorar mediante:

- La propuesta de nuevas métricas y adaptación de métricas existentes para ser operacionalizadas en diferentes niveles de abstracción, especialmente en modelos PIM y PSM. Para demostrar la validez y utilidad de las nuevas métricas incorporadas, se utilizarán aproximaciones basadas en la teoría de la medición como el marco DISTANCE [71] y aproximaciones basadas en propiedades [14] para validar teóricamente las métricas, además, se realizarán experimentos controlados para validarlas empíricamente y demostrar su utilidad.
- La extensión del Modelo de Usabilidad Web para incorporar una nueva perspectiva asociada a la calidad de los datos, basándonos en la estructura de la

norma SQuaRE, que permita representar como ciertas características referentes a los datos, que se manejan en la aplicación Web, contribuyen a alcanzar cierto grado de usabilidad.

Con respecto al proceso WUEP, existen diversas tareas futuras que se han considerado para realizar en un futuro:

- Elaboración de una herramienta para dar soporte a todas las fases de WUEP con la capacidad de incorporar el Modelo de Usabilidad Web a modo de repositorio. Una de las posibilidades que se han empezado a estudiar es la transformación de la especificación SPEM en otros lenguajes de procesos orientados a automatizar flujos de trabajo. Se pretende que la herramienta ofrezca además, los mecanismos necesarios para operacionalizar las distintas métricas en distintos niveles de abstracción y en diferentes métodos de desarrollo Web. Esto se pretende conseguir haciendo uso de los meta-modelos que definen los distintos modelos (PIM, PSM, CM) de un método de desarrollo concreto y representando la fórmula de las métricas en base a los elementos de estos meta-modelos, para su cálculo automático en las instancias de dichos meta-modelos (modelos).
- Análisis de umbrales e intervalos de aceptación para las métricas suministradas por el Modelo de Usabilidad Web, para poder ofrecer al evaluador una guía sobre que intervalos serían más adecuados a utilizar teniendo en cuenta los perfiles de la evaluación.
- Análisis de distintas propuestas acerca de mecanismos de agregación para combinar los distintos valores individuales obtenidos tras la aplicación de las métricas, para poder ofrecer al evaluador una guía sobre qué mecanismos ayudan a realizar comparaciones fiables de atributos de usabilidad entre distintas aplicaciones Web evaluadas.
- Definición de un mecanismo para analizar cómo los atributos de usabilidad pertenecientes al Modelo de Usabilidad Web impactan, (positivamente o negativamente) entre ellos, especialmente, relacionando atributos pertenecientes a la vista de la calidad del producto software, con atributos pertenecientes a la vista de la calidad en uso.
- Instanciación de WUEP en otros métodos de desarrollo Web dirigido por modelos como WebML o UWE, para ofrecer evaluaciones de usabilidad a los desarrolladores que emplean dichos métodos, proporcionando un conjunto de métricas operacionalizadas, evaluando distintos tipos de aplicaciones Web.
- Experimentos controlados para medir la efectividad y la facilidad de uso de WUEP comparándolo con otros métodos de evaluación de usabilidad. Cubriendo de esta forma una de las necesidades detectadas en la revisión sistemática.

- Definición de mecanismos para el análisis de la trazabilidad e impacto de los problemas de usabilidad detectados de cara a la evolución de los cambios propuestos en los modelos de la aplicación Web.
- Extensión y adaptación de WUEP para la evaluación de usabilidad en próximas generaciones de aplicaciones Web (Cloud-Computing y Rich Internet Applications – RIA) que sigan un desarrollo basado en el enfoque de líneas de producto Software.

Los trabajos futuros estarán contextualizados en el marco del proyecto CICYT denominado MULTIPLE (Multimodeling Approach For Quality-Aware Software Product Lines - TIN2009-13838)

5.3. PUBLICACIONES RELACIONADAS

Durante el desarrollo de esta tesina de máster, se han realizado diferentes contribuciones en modo de publicaciones. A continuación se recopilan dichas publicaciones ordenadas de mayor a menor nivel, y detallando qué parte de la tesina está vinculado a su contenido:

Conferencias internacionales relevantes (Conferencias de nivel A según CORE ³)

- Fernández, A.; Insfrán, E.; Abrahão, S.: *Integrating a Usability Model into a Model-Driven Web Development Process*. 10th International Conference on Web Information Systems Engineering (WISE 2009), Poznan, Poland, Lecture Notes in Computer Science 5802, Springer (2009), pp. 497-510.

En este artículo se presentó la idea principal acerca de la integración de evaluaciones de usabilidad en procesos de desarrollo Web dirigidos por modelos (Sección 3.1) y el Modelo de Usabilidad Web (Sección 3.2).

Workshops internacionales relevantes (pertenecientes a conferencias CORE A):

- Insfran, E.; Fernández, A.: *A Systematic Review of Usability Evaluation in Web Development*. Proceedings of the International Workshop on Web Usability and Accessibility (IWWUA 2008) co-located with the 9th International Conference on Web Information Systems Engineering (WISE 2008), Auckland, New Zealand, Lecture Notes in Computer Science 5176, Springer (2008), pp. 81-91.

Los proceedings de este Workshop aparecen en la lista CiteseerX⁴ [358/581], con impacto estimado de 0.02 basado en el factor tradicional de impacto Garfield.

En este artículo se presentó la primera versión de la revisión sistemática (Sección 2.1) que sirvió de punto de partida para el trabajo a desarrollar en esta tesina.

³ <http://www.core.edu.au/rankings/Conference%20Ranking%20Main.html>

⁴ <http://citeseerx.ist.psu.edu/stats/venues/>

- Fernández, A.; Insfrán, E.; Abrahão, S.: *Towards a Usability Evaluation Process for Model-Driven Web Development*. Proceedings of the 2nd International Workshop on the Interplay between Usability Evaluation and Software Development (I-USED 2009), co-located with the 12th IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2009), Upsala, Sweden. CEUR-WS.org, ISSN 1613-0073, (2009).

En este artículo se presentó una versión preliminar del proceso de evaluación de usabilidad.

Capítulos de libro:

- Fernández, A.; Insfran, E.; Abrahão, S.: *Modelo de Usabilidad Web Alineado con SQUARE para Procesos de Desarrollo Dirigido por Modelos*. In: Calidad del Producto y del Proceso Software, Editorial RA-MA (2009).

En este capítulo de libro en castellano se presentaron los primeros resultados de la operacionalización de métricas provenientes del Modelo de Usabilidad Web en un método de desarrollo Web concreto. El capítulo es fruto de una invitación destinada a la recopilación de resultados obtenidos durante la participación en la red CALIPSO (Calidad del Producto y Proceso Software - TIN2005-24055-E) y se sometió a un proceso de revisión externa por pares para su aceptación final.

Posters:

- Abrahão, S.; Ramos, I.; Cabello, M.E.; Insfran, E.; Fernandez, A.: *A multimodeling approach for cloud computing*. Presentado en Google Faculty Summit Software Engineering Track, Zurich, (2009).

En este póster se introduce una de líneas futuras que se pretenden seguir, la incorporación de la evaluación de usabilidad en aplicaciones basadas en Cloud-Computing que han sido desarrolladas siguiendo el enfoque de líneas de producto Software. Esta idea está basada en el contexto del proyecto MULTIPLE (Multimodeling Approach For Quality-Aware Software Product Lines - TIN2009-13838)

REFERENCIAS

- [1] Abrahão, S.; Condori-Fernandez, N.; Olsina, L.; Pastor, O.: *Defining and Validating Metrics for Navigational Models*. Proc. of the 9th International Software Metrics Symposium (METRICS'03), IEEE. pp. 200-210, (2003).
- [2] Abrahão, S.; Iborra, E.; Vanderdonck J.: *Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool*. Maturing Usability. Springer HCI series, Vol. 10, pp. 3-32, (2007).
- [3] Abrahão, S.; Insfran, E.: *Early Usability Evaluation in Model-Driven Architecture Environments*. Proc. of the 6th IEEE International Conference on Quality Software (QSIC), Beijing, China. IEEE Computer Society, ISBN 0-7695-2718-3, pp. 287-294, (2006).
- [4] Abran A.; Khelifi A.; Suryan W.; Seffah A.: *Consolidating the ISO Usability Models*. Proceedings of 11th International Software Quality Management Conference (Springer), Glasgow, Scotland, UK, (2003).
- [5] Atterer, R.; Schmidt, A.: *Adding Usability to Web Engineering Models and Tools*. 5th International Conference on Web Engineering (ICWE), Australia. Springer Verlag, pp. 36-41, (2005).
- [6] Bastien, J. M.; Scapin, D. L.: *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*, version 2.1, (1993).
- [7] Becker, S. A.; Berkemeyer, A.: *Rapid Application Design and Testing of Web Usability*. IEEE MultiMedia archive, Vol. 9, Issue 4, pp. 38-46, (2002).
- [8] Bevan, N.: *Web Usability and the New ISO Quality Model*. Keynote 1st International Workshop on Web Usability and Accessibility (IWWUA), (2007).
- [9] Blackmon, M. H.; Polson, P. G.; Kitajima, M.; Lewis, C.: *Cognitive walkthrough for the Web*. Proc. of the ACM conference on human factors in computing systems (CHI'02), pp. 463-470, (2002).
- [10] Blackmon, M.H.; Kitajima, M.; Polson, P.G.: *Tool for accurately predicting Website navigation problems, non-problems, problem severity, and effectiveness of repairs*. Proc. of the SIGCHI conference on Human factors in computing systems, USA, pp.31-40, (2005).
- [11] Bolchini, D.; Garzotto, F.: *Quality of Web Usability Evaluation Methods: An Empirical Study on MILE+*. Proceedings of 1st International Workshop on Web Usability and Accessibility (IWWUA), France, pp. 481-492, (2007).
- [12] Bolchini, D.; Paolini, P.; Randazzo, G.: *Adding hypermedia requirements to goal-driven analysis*. Proc. of the 11th IEEE International Requirements Engineering Conference, pp 127-137, (2003).
- [13] Botafogo, R.; Rivlin, E.; Shneiderman, B.: *Structural analysis of hypertexts: Identifying hierarchies and useful metrics*. ACM Trans. Inform. Systems, 10(2), pp. 142-180, (1992).
- [14] Briand, L.; El-Elman, K.; Morasca, S.: *Theoretical and Empirical Validation of Software Product Measures*. International Software Engineering Research Network Technical Report ISERN-95-03, (1995).
- [15] Calero, C.; Ruiz, J.; Piattini, M.: *Classifying Web metrics using the Web Quality Model*. Emerald Group Publishing Limited. ISSN: 1468-4527. Vol.29, nº 3, pp. 227-248, (2005).
- [16] Casteleyn, S.; Daniel, F.; Dolog, P.; Matera, M.: *Engineering Web Applications*. Libro editado por Springer Verlag, (2009)
- [17] Ceri, S.; Fraternali, P.; Bongio, A.: *Web Modeling Language (WebML): A Modeling Language for Designing Web Sites*. Proc. of the 9th WWW Conference, pp. 137-157, (2000).
- [18] Conte, T.; Massollar, J.; Mendes, E.; Travassos, G.H.: *Usability Evaluation Based on Web Design Perspectives*. Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 146-155, (2007).
- [19] Costabile, M.F.; Matera, M.: *Guidelines for hypermedia usability inspection*. IEEE Multimedia, Vol. 8, Issue 1, pp. 66-69, (2001).
- [20] Chi, E.H.: *Improving Web usability through visualization*. IEEE Internet Computing, Vol. 6, Issue 2, pp. 64-71, (2002).

- [21] Chi, E.H.; Rosien, A.; Supattanasiri, G.; Williams, A.; Royer, C.; Chow, C.; Robles, E.; Dalal, B.; Chen, J.; Cousins, S.: *The bloodhound project: Automating discovery of Web usability issues using the infoscent simulator*. ACM Conference on Human Factors in Computing Systems, CHI Letters, Vol. 5, Issue 1, pp. 505-512, (2003).
- [22] Dieste, O.; López, M.; Ramos, F.: *Updating Empirical Evidence to Derive Well-founded Practices in Software Requirements Elicitation Techniques Selection*. 11th Workshop on Requirements Engineering, Spain, (2008).
- [23] Dromey, R.G.: *Software Product Quality: Theory, Model and Practice*. Software Quality Institute, Griffith University, Nathan, Brisbane, Australia, (1998).
- [24] Fenton, N.; Pfleeger, S. L.: *Software Metrics: A Rigorous and Practical Approach*, Second Edition. International Thomson Computer Press, (1996).
- [25] Fleiss, J.L.: *Statistical Methods for Rates and Proportions*, 2nd ed. (New York: John Wiley) pp. 38-46 (1981).
- [26] Freire, A. P.; Goularte, R; Fortes, R. P. M.: *Techniques for Developing more Accessible Web Applications: a Survey Towards a Process Classifications*. Proc. of the 25th ACM International Conference on Design of communication, pp. 162-169, (2007).
- [27] Ginige, A.; Murugesan, S.: *Web engineering: an introduction*. IEEE Multimedia, Vol. 8, Issue 1, pp. 14-18. (2001).
- [28] Gómez, J.; Cachero, C; y Pastor, O.: *On conceptual Modeling of Device-Independent Web Applications*. IEEE MultiMedia 8(2), pp. 26-39, (2001).
- [29] Granollers, T.: *MPlu+a. Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares*. Tesis Doctoral, Departament de Llenguatges i Sistemes Informàtics. Universitat de Lleida, (2004).
- [30] Hvannberg, E.T.; Law, E.L.; Lárusdóttir, M.K.: *Heuristic evaluation: Comparing ways of finding and reporting usability problems*. Interacting with Computers, Vol. 19, Issue 2, pp 225-240, (2007).
- [31] Hornbæk, K.: *Current practice in measuring usability: Challenges to usability studies and research*. International Journal of Human-Computer Studies, 64, pp. 79-102, (2006).
- [32] Hornbæk, K.; Frøkjær, E.: *Two psychology-based usability inspection techniques studied in a diary experiment*. Proceedings of the 3rd Nordic conference on Human-computer interaction, Finland, pp. 3-12, (2004).
- [33] Hornbæk, K.; Frøkjær, E.: *Comparing Usability Problems and Redesign Proposals as Input to Practical Systems Development*. Proceedings of the SIGCHI conference on Human factors in computing systems, USA, pp. 391-400, (2005).
- [34] IBM. IBM Web Design Guidelines. Disponible en <https://www-01.ibm.com/software/ucd/> [Último acceso: Octubre 2009].
- [35] ISO/IEC 9126-1 Standard, *Software engineering – Product quality - Part 1: Quality model*, (2001).
- [36] ISO/IEC 9126-2-3 Standard, *Software engineering – Product quality - Part 2-3: External & Internal Metrics*, (2002).
- [37] ISO/IEC 9241-10, *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 10: Dialogue Principles*, (1996).
- [38] ISO/IEC 9241-11, *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability*, (1998).
- [39] ISO/IEC 12207: *Standard for Information Technology – Software Lifecycle Processes*, (1998).
- [40] ISO/IEC 13407: *Human-Centred Design Processes for Interactive Systems*, (1999).
- [41] ISO/IEC 14598-1, *Information technology - Software product evaluation - Part 1: General overview*, (1999).
- [42] ISO/IEC 25000, *Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*, (2005).

- [43] Ivory, M.Y.: *An Empirical Foundation for Automated Web Interface Evaluation*. PhD Thesis, University of California, Berkeley, Computer Science Division, (2001).
- [44] Ivory, M.Y.; Hearst, M.A.: *The State of the Art in Automating Usability Evaluation of User Interfaces*. ACM Computing Surveys, Vol. 33, Issue 4, pp. 470-516, (2001).
- [45] Ivory, M.Y.; Megraw, R.: *Evolution of Web Site Design Patterns*. ACM Transactions on Information Systems, Vol. 23, Issue 4, pp 463-497, (2005).
- [46] Jung, H.; Lee, M.; Lee, S; Sung, W.: *Development of an Academic Research Information Service through Repeated Usability Evaluations*. 1st International Conference on Advances in Computer-Human Interaction, pp. 195-199, (2008).
- [47] Kitchenham, B.: *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, Version 2.3, EBSE Technical Report, Keele University, UK, (2007).
- [48] Kraemer, E.; Ummelen, N.: *Thinking about thinking aloud: a comparison of two verbal protocols for usability testing*. IEEE Transactions on Professional Communication, Vol. 47, Issue 2, pp. 105-117, (2004).
- [49] Kraus, A.; Knapp, A.; Koch, N.: *Model-Driven Generation of Web Applications in UWE*. 3rd Inter Workshop on Model-Driven Web Engineering, (2006).
- [50] Leavit, M.; Shneiderman, B.: *Research-Based Web Design & Usability Guidelines*. U.S. Government Printing Office. (2006), Web: <http://www.usability.gov/pdfs/guidelines.html> [Último acceso: Octubre 2009]
- [51] Li, C.; Kit, C.: *Web structure mining for usability analysis*. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 309-312, (2005).
- [52] Lorés, J.; Granollers, T.: *La Ingeniería de la Usabilidad y de la Accesibilidad aplicada al diseño y desarrollo de sitios Web*, (2007). Capítulo del libro "Tendencias Actuales en la interacción Persona Ordenador: Accesibilidad, Adaptabilidad y Nuevos Paradigmas", pp. 119-144, Ediciones de la Universidad Castilla-La Mancha (Albacete, España), (2003).
- [53] Lynch, P.; Horton, S.: *Web Style Guide: Basic Design Principles for Creating Web Sites*, 2nd ed. New Haven: Yale University Press, (2002).
- [54] Mariage, C.; Vanderdonckt, J.; Chevalier, A.: *Using the MetroWeb tool to improve usability quality of Web sites*. Proc. of the 3rd Latin American Web Congress (LA-WEB), pp. 10, (2005).
- [55] Matera, M.; Rizzo, F.; Toffetti Carughi, G.: *Web Usability: Principles and Evaluation Methods*. En Mendes, E. and Mosley, N. (eds.), Web Engineering. Springer Verlag, pp. 143-180, (2006).
- [56] McCall, J. A.; Richards, P. K.; Walters, G. F.: *"Factors in software quality"*. Vols. III, Rome Aid Defense Centre, Italy, (1977).
- [57] McNeile, A.: MDA: The Vision with the Hole? (2003). Disponible en <http://www.metamaxim.com/download/documents/MDAv1.pdf>, 2003. [Último acceso: Octubre 2009]
- [58] Mendes, E.: *A Systematic Review of Web Engineering Research*. Proc. of the International Symposium on Empirical Software Engineering, pp. 498-507, (2005).
- [59] Microsoft corporation. Microsoft Web Design Guidelines. Disponible en <http://www.microsoft.com/spain/empresas/guias/usabilidad/home.mspx> [Último acceso: Octubre 2009]
- [60] Molina, F.; Toval, J.A.: *Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems*. Advances in Engineering Software 40(12), pp. 1306-1317, (2009).
- [61] Moraga, M.A; Calero, C.; Piattini, M.; Diaz, O.: *Improving a portlet usability model*. Software Quality Control, v.15 n.2, pp. 155-177, (2007).
- [62] Nakamichi, N.; Sakai, M.; Shima, K., Hu, J.; Matsumoto, K.: *WebTracer: A new Web usability evaluation environment using gazing point information*. Electronic Commerce Research and Applications, Vol. 6, Issue 1, pp. 63-73, (2007).

- [63] Neuwirth, C. M.; Regli, S. H.: *IEEE Internet Computing Special Issue on Usability and the Web*, Vol. 6, No. 2, (2002).
- [64] Nielsen, J.: *Usability Engineering*. Academic Press, London, (1993).
- [65] Nielsen, J.: *Usabilidad. Diseños de sitios Web*. Prentice Hall, (2000).
- [66] Offutt, J.: *Quality Attributes of Web Software Applications*. IEEE Software: Special Issue on Software Engineering of Internet Software, pp. 25-32, (2002).
- [67] Olsina, L.; Rossi, G.: *Measuring Web Application Quality with WebQEM*. IEEE Multimedia, Vol. 9, nº 4, pp. 20-29, (2002).
- [68] OMG, Software Process Engineering Meta-model v2.0 (SPEM 2), formal/2008-04-01, (2008). Disponible en <http://www.omg.org/technology/documents/formal/spem.htm> [Último acceso: Octubre 2009]
- [69] Panach, I., Condori, N., Valverde, F., Aquino, N., Pastor, O.: *Towards an Early Usability Evaluation for Web Applications*. En: Alain Abran, R.D., Antonia Mas (ed.): MENSURA 2007. LNCS 4895, pp. 32-45, (2007).
- [70] Petticrew, M.; Roberts, H.: *Systematic Reviews in the Social Sciences: A Practical Guide*, Blackwell Publishing, (2005).
- [71] Poels, G.; Dedene, G.: *DISTANCE: A Framework for Software Measure Construction*. Research Report 9937, Department of Applied Economics, Katholieke Universiteit Leuven, (1999).
- [72] Seffah, A.; Donyaee, M.; Kline, R.B.; Padda, H.K.: *Usability measurement and metrics: A consolidated model*. Software Quality Journal 14(2), pp. 159-178, (2006).
- [73] Ssemugabi, S.; De Villiers, R.: *A comparative study of two usability evaluation methods using a Web-based e-learning application*. Proc. of the annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries SAICSIT. pp 132-142, (2007).
- [74] Steuer, J.: *Defining Virtual Reality: Dimensions Determining Telepresence*. Journal of Communication, 42(4), pp. 73-93, (1992).
- [75] Sun Microsystems. *Sun Guide to Web Style*. Disponible en <http://www.sun.com/Webdesign/> [Último acceso: Octubre 2009]
- [76] Tonn-Eichstädt, H.: *Measuring Website usability for visually impaired people-a modified GOMS analysis*. Proc. of the 8th international ACM SIGACCESS conference on Computers and accessibility, USA, (2006).
- [77] Triacca, L.; Inversini, A.; Bolchini, D.: *Evaluating Web usability with MiLE+*. Proceedings of the 7th IEEE International Symposium on Web Site Evolution, pp. 22-29, 2005.
- [78] Usability.gov Site; Disponible en <http://www.usability.gov/> [Último acceso: Octubre 2009]
- [79] Van Velsen, L.; Van der Geest, T.; Klaassen, R.: *Testing the usability of a personalized system: comparing the use of interviews, questionnaires and thinking-aloud*. IEEE International Professional Communication Conference IPCC, pp.1-8, (2007).
- [80] Van Waes, L.: *Thinking aloud as a method for testing the usability of Websites: the influence of task variation on the evaluation of hypertext*. IEEE Transactions on Professional Communication. Vol. 43, Issue 3, pp. 279-291, (2000).
- [81] Vanderdonckt, J.; Beirekdar, A.; Noirhomme-Fraiture, M.: *Automated Evaluation of Web Usability by Guideline Review*. Proceedings of the 4th International Conference on Web Engineering (ICWE). Vol. 3140, pp. 17-30, (2004).
- [82] World Wide Web Consortium W3C: *Web Content Accessibility Guidelines (WCAG) 2.0*. Disponible en <http://www.w3.org/TR/WCAG/> [Último acceso: Octubre 2009]
- [83] Zaharias, P.: *A usability evaluation method for e-learning: focus on motivation to learn*. Proceedings of the Conference on Human Factors in Computing Systems, Canada. pp. 1571-1576, (2006).

ANEXO A: LISTA DE ARTÍCULOS INCLUIDOS EN LA REVISIÓN SISTEMÁTICA.

La lista de artículos incluidos sigue el siguiente formato:

id. autores. "título" (año) [fuente bibliográfica]

1. Aberg, J. and Shahmehri, N. "An empirical study of human Web assistants: implications for user support in Web information systems" (2001) [ACM]
2. Ahn, J.; Brusilovsky, P.; He, D.; Grady, J. and Li, Q. "Personalized Web Exploration with Task Models" (2008) [WWW]
3. Allen, M.; Currie, L.; Bakken, S.; Patel, V. and Cimino, J. "Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology" (2006) [ScienceDirect]
4. Alshamari, M. and Mayhew, P. "Task Design: Its Impact on Usability Testing" (2008) [IEEEExplore]
5. Anandhan, A.; Dhandapani, S.; Reza, H. and Namasivayam, K. "Web Usability Testing - CARE Methodology" (2006) [IEEEExplore]
6. Ardito, C.; Lanzilotti, R.; Buono, P. and Piccinno, A. "A tool to support usability inspection" (2006) [ACM]
7. Atterer, R. "Where Web engineering tool support ends: building usable Websites" (2005) [ACM]
8. Atterer, R. and Schmidt, A. "Tracking the interaction of users with AJAX applications for usability testing" (2007) [ACM]
9. Atterer, R. and Schmidt, A. "Adding Usability to Web Engineering Models and Tools" (2005) [ICWE]
10. Atterer, R.; Wnuk, M. and Schmidt, A. "Knowing the user's every move: user activity tracking for Website usability evaluation and implicit interaction" (2006) [WWW]
11. Bachiochi, D.; Berstene, M.; Chouinard, E.; Conlan, N.; Danchak, M.; Furey, T.; Neligon, C. and Way, D. "Usability studies and designing navigational aids for the World Wide Web" (1997) [ScienceDirect]
12. Badre, A. and Jacobs, A. "Usability, aesthetics, and efficiency: an evaluation in a multimedia environment" (1999) [IEEEExplore]
13. Bartell, A.L. "Using content analysis and Web design heuristics to evaluate informational Web sites: an exploratory study" (2005) [IEEEExplore]
14. Basu, A. "Context-driven assessment of commercial Web sites" (2003) [IEEEExplore]
15. Becker, S.A. and Berkemeyer, A. "Rapid application design and testing of Web usability" (2002) [IEEEExplore]
16. Becker, S.A. and Mottay, F.E. "A global perspective on Web site usability" (2001) [IEEEExplore]
17. Bednarik, R.; Gerdt, P.; Miraftebi, R. and Tukiainen, M. "Development of the TUP model - evaluating educational software" (2004) [IEEEExplore]
18. Beier, B. and Vaughan, M.W. "The bull's-eye: a framework for Web application user interface design guidelines" (2003) [ACM]
19. Bevis, K.J. and Henke, K.A. "Evaluating usability in an information product" (2008) [IEEEExplore]
20. Blackmon, M.H.; Kitajima, M. and Polson, P.G. "Repairing usability problems identified by the cognitive walkthrough for the Web" (2003) [ACM]
21. Blackmon, M.H.; Kitajima, M. and Polson, P.G. "Tool for accurately predicting Website navigation problems, non-problems, problem severity, and effectiveness of repairs" (2005) [ACM]
22. Blackmon, M.H.; Polson, P.G.; Kitajima, M. and Lewis, C. "Cognitive walkthrough for the Web" (2002) [ACM]
23. Blake, C.T. and Rapanotti, L. "Usability evaluation of distributed groupware in distance learning" (2004) [IEEEExplore]

24. Bolchini, D. and Garzotto, F. "Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+" (2007) [IWUUA]
25. Bolchini, D.; Paolini, P. and Randazzo, G. "Adding hypermedia requirements to goal-driven analysis" (2003) [IEEEExplore]
26. Brajnik, G.; Cancila, D.; Nicoli, D. and Pignatelli, M. "Do text transcoders improve usability for disabled users?" (2005) [ACM]
27. Burmeister, O.K. "Usability testing: revisiting informed consent procedures for testing internet sites" (2000) [ACM]
28. Burton, C. and Johnston, L. "Will World Wide Web user interfaces be usable?" (1998) [IEEEExplore]
29. Burton, M.C. and Walther, J.B. "A survey of Web log data and their application in use-based design" (2001) [IEEEExplore]
30. Cao, J.; Crews, J.M.; Nunamaker, J.F., Jr.; Burgoon, J.K. and Lin, M. "User experience with Agent99 Trainer: a usability study" (2004) [IEEEExplore]
31. Chadwick-Dias, A.; McNulty, M. and Tullis, T. "Web usability and age: how design changes can improve performance" (2003) [ACM]
32. Chandrashekar, S.; Stockman, T.; Fels, D. and Benedyk, R. "Using think aloud protocol with blind users: a case for inclusive usability evaluation methods" (2006) [ACM]
33. Chang, W.; Hon, S. and Chu, C. "A systematic framework for evaluating hyperlink validity in Web environments" (2003) [IEEEExplore]
34. Chatley, R.; Kramer, J.; Magee, J. and Uchitel, S. "Visual methods for Web application design" (2003) [IEEEExplore]
35. Chi, E. "Improving Web Usability Through Visualization" (2002) [IEEE IC]
36. Chi, E.; Rosien, A.; Supattanasiri, G.; Williams, A.; Royer, C.; Chow, C.; Robles, E.; Dalal, B.; Chen, J. and Cousins, S. "The bloodhound project: automating discovery of Web usability issues using the InfoScent simulator" (2003) [ACM]
37. Chou, E. "Redesigning a large and complex Website: how to begin, and a method for success" (2002) [ACM]
38. Clark, J.; Van Oorschot, P.C. and Adams, C. "Usability of anonymous Web browsing: an examination of Tor interfaces and deployability" (2007) [ACM]
39. Clayton, N.; Biddle, R. and Tempero, E. "A study of usability of Web-based software repositories" (2000) [IEEEExplore]
40. Conte, T.; Massollar, J.; Mendes, E. and Travassos, G.H. "Usability Evaluation Based on Web Design Perspectives" (2007) [IEEEExplore]
41. Cooke, L. "Improving usability through eye tracking research" (2004) [IEEEExplore]
42. Cooke, L. and Cuddihy, E. "Using eye tracking to address limitations in think-aloud protocol" (2005) [IEEEExplore]
43. Costabile, M.F. and Matera, M. "Guidelines for hypermedia usability inspection" (2001) [IEEEExplore]
44. Cuddihy, E.; Wei, C.; Barrick, J.; Maust, B.; Bartell, A.L. and Spyridakis, J.H. "Methods for assessing Web design through the internet" (2005) [ACM]
45. Cugini, J. and Scholtz, J. "VISVIP: 3D visualization of paths through Web sites" (1999) [IEEEExplore]
46. De Marsico, M. and Levialdi, S. "Evaluating Web sites: exploiting user's expectations" (2004) [ScienceDirect]
47. De Wet, L.; Bignaut, P. and Burger, A. "Comprehension and usability variances among multicultural Web users in South Africa" (2002) [ACM]
48. Douglas, I. "Collaborative International Usability Testing: Moving from Document-based Reporting to Information Object Sharing" (2006) [IEEEExplore]
49. Duan, J. and Zhang, N. "Research on Visualization Techniques for Web Usability Analysis" (2007) [IEEEExplore]

50. El Helou, S.; Gillet, D.; Salzmann, C. and Yu, C.M. "A Study of the Acceptability of a Web 2.0 Application by Higher-Education Students Undertaking Collaborative Laboratory Activities" (2009) [IEEExplore]
51. Faulkner, X. and Culwin, F. "The internet as a resource for teaching non internet topics" (2001) [IEEExplore]
52. Fisher, J. and Burstein, F. "Usability + usefulness = trust: an exploratory study of Australian health Web sites" (2008) [IR]
53. Foglia, P.; Prete, C.A. and Zanda, M. "Relating GSR Signals to traditional Usability Metrics: Case Study with an anthropomorphic Web Assistant" (2008) [IEEExplore]
54. Fraternali, P. and Tisi, M. "Identifying Cultural Markers for Web Application Design Targeted to a Multi-Cultural Audience" (2008) [ICWE]
55. Fuhrmann, S.; Bosley, J.; Li, B.; Crawford, S.; MacEachren, A.; Downs, R. and Gahegan, M. "Assessing the usefulness and usability of online learning activities: MapStats for kids" (2003) [ACM]
56. Gee, K. "The ergonomics of hypertext narrative: usability testing as a tool for evaluation and redesign" (2001) [ACM]
57. Go, K.; Takahashi, T. and Imamiya, A.; "A case study on participatory redesign of Web site with scenario-based techniques" (2000) [IEEExplore]
58. Grady, H.M. "Web site design: a case study in usability testing using paper prototypes" (2000) [IEEExplore]
59. Granic, A. and Glavinic, V. "Evaluation of interaction design in Web-based intelligent tutoring systems" (2006) [IEEExplore]
60. Granic, A.; Mitrovic, I. and Marangunic, N. "Usability evaluation of Web portals" (2008) [IEEExplore]
61. Habuchi, Y.; Kitajima, M. and Takeuchi, H. "Comparison of eye movements in searching for easy-to-find and hard-to-find information in a hierarchically organized information structure" (2008) [ACM]
62. Harper, S.; Yesilada, Y.; Goble, C. and Stevens, R. "How much is too much in a hypertext link?: investigating context and preview -- a formative evaluation" (2004) [ACM]
63. Hattori, G.; Hoashi, K.; Matsumoto, K. and Sugaya, F. "Robust Web page segmentation for mobile terminal using content-distances and page layout information" (2007) [WWW]
64. Hatzilygeroudis, I.; Koutsojannis, C. and Papachristou, N. "Evaluation of Usability and Assessment Capabilities of an e-Learning System for Nursing Radiation Protection" (2007) [IEEExplore]
65. Hijon-Neira, R.; Urquiza-Fuentes, J.; Dominguez-Mateos, F. and Velazquez-Iturbide, J.A. "Assessing the Usability of a Cookies-Based Access Monitoring Tool for a CMS" (2007) [IEEExplore]
66. Hornbæk, K. and Frøkjær, E. "Two psychology-based usability inspection techniques studied in a diary experiment" (2004) [ACM]
67. Hornbæk, K. and Frøkjær, E. "Comparing usability problems and redesign proposals as input to practical systems development" (2005) [ACM]
68. Hu, J.; Zhao, J.; Shima, K.; Takemura, Y. and Matsumoto, K. "Comparison of Chinese and Japanese in designing B2C Web pages toward impressional usability" (2001) [IEEExplore]
69. Hvannberg, E.T.; Law, E.; Lárusdóttir, M. "Heuristic evaluation: Comparing ways of finding and reporting usability problems" (2007) [ScienceDirect]
70. Iahad, N.; Dafoulas, G.A.; Kalaitzakis, E. and Macaulay, L.A. "Evaluation of online assessment: the role of feedback in learner-centered e-learning" (2004) [IEEExplore]
71. Ivory, M.Y. and Hearst, M.A. "The state of the art in automating usability evaluation of user interfaces" (2001) [ACM]
72. Ivory, M.Y. and Hearst, M.A. "Improving Web Site Design" (2002) [IEEE IC]
73. Ivory, M.Y. and Megraw, R. "Evolution of Web Site Design Patterns" (2005) [ACM]
74. Jinling, C. and Huan, G. "Measuring Website Usability of Chinese Enterprise with a Heuristic Procedure" (2007) [IEEExplore]

75. Johnson, J. and Marshall, C. "Convergent usability evaluation: a case study from the EIRS project" (2005) [ACM]
76. Johnson, T.; Zhang, J.; Tang, Z.; Johnson, C. and Turley, J. "Assessing informatics students' satisfaction with a Web-based courseware system" (2004) [ScienceDirect]
77. Jung, H.; Lee, M.; Lee, S. and Sung, W. "Development of an Academic Research Information Service through Repeated Usability Evaluations" (2008) [IEEExplore]
78. Kakasevski, G.; Mihajlov, M.; Arsenovski, S. and Chungurski, S. "Evaluating usability in learning management system moodle" (2008) [IEEExplore]
79. Kao, H.Y. "Usability Testing of AMC Hospital's Website for Home Users: Case Study for On-Site Registration Design" (2007) [IEEExplore]
80. Katsanos, C.; Tselios, N.K. and Avouris, N.M "InfoScout evaluator: a semi-automated tool to evaluate semantic appropriateness of hyperlinks in a Web site" (2006) [ACM]
81. Kazienko, P. and Pilarczyk, M. "Hyperlink assessment based on Web usage mining" (2006) [ACM]
82. Keilson, S.; King, E. and Sapnar, M. "Learning science by doing science on the Web" (1999) [IEEExplore]
83. Kemp, E. and Setungamudalige, D.T. "A resource support toolkit (R-IDE): supporting the DECIDE framework" (2006) [ACM]
84. Kim, E.; Wang, M.; Lau, C. and Kim, Y. "Application and Evaluation of Personal Health Information Management System" (2004) [IEEExplore]
85. Kotelly, C. "World wide Web as usability tester, collector, recruiter" (1997) [ACM]
86. Krahmer, E. and Ummelen, N. "Thinking about thinking aloud: a comparison of two verbal protocols for usability testing" (2004) [IEEExplore]
87. Lee, K.B. and Grice, R.A. "An adaptive viewing application for the Web on personal digital assistants" (2003) [ACM]
88. Li, C. and Kit, C. "Web structure mining for usability analysis" (2005) [IEEExplore]
89. Li, Q.; Sun, L. and Duan, J. "Web page viewing behavior of users: an eye-tracking study" (2005) [IEEExplore]
90. Lister, M. "Usability testing software for the internet" (2001) [ACM]
91. Mariage, C.; Vanderdonckt, J. and Chevalier, A. "Using the MetroWeb tool to improve usability quality of Web sites" (2005) [IEEExplore]
92. Massey, A.P.; Khatri, V. and Montoya-Weiss, M.M. "Online Services, Customer Characteristics and Usability Requirements" (2008) [IEEExplore]
93. Matera, M.; Rizzo, F. and Toffetti Carughi, G. "Web Usability: Principles and Evaluation Methods". In E. Mendes, N. Mosley (eds.), *Web Engineering*. Springer Verlag. (2006) [Book]
94. Mayuzumi, Y.; Jin, H. and Choh, I. "Study of finding solutions to problems between environmental information side and user side on Web" (2003) [IEEExplore]
95. Milic-Frayling, N.; Jones, R.; Rodden, K.; Smyth, G.; Blackwell, A. and Sommerer, R. "Smartback: supporting users in back navigation" (2004) [WWW]
96. Millen, D.R. "Remote usability evaluation: user participation in the design of a Web-based email service" (1999) [ACM]
97. Moraga, M.A.; Calero, C. and Piattini, M. "Ontology driven definition of a usability model for second generation portals" (2006) [ICWE]
98. Moreno, L.; Martínez, P. and Ruiz, B. "Inclusive Usability Techniques in Requirements Analysis of Accessible Web Applications" (2007) [IWWUA]
99. Morgan, M. and Borns, L. "360 degrees of usability" (2004) [ACM]
100. Nakamichi, N.; Sakai, M.; Shima, K.; Hu, J. and Matsumoto, K. "WebTracer: A new Web usability evaluation environment using gazing point information" (2007) [ScienceDirect]
101. Nakamichi, N.; Shima, K.; Sakai, M. and Matsumoto, K. "Detecting low usability Web pages using quantitative data of users' behavior" (2006) [ACM]

102. Nijland, N.; Seydel, E.R.; van Gemert-Pijnen, J.E.W.C.; Brandenburg, B.; Kelders, S.M. and Will, M. "Evaluation of an Internet-Based Application for Supporting Self-Care of Patients with Diabetes Mellitus Type 2" (2009) [IEEExplore]
103. Norman, K.L and Panizzi, E. "Levels of automation and user participation in usability testing" (2006) [ScienceDirect]
104. Obendorf, H.; Weinreich, H. and Hass, T. "Automatic support for Web user studies with SCONE and TEA" (2004) [ACM]
105. Oehler, M. and Biffignandi, S. "Web Sites Communication Performance: A Methodological Approach" (2008) [IEEExplore]
106. Olsina, L.; Rossi, G.; Garrido, A.; Distanto, D. and Canfora, G. "Incremental Quality Improvement in Web Applications Using Web Model Refactoring" (2007) [IWWUA]
107. Orii, Y.; Nozawa, T. and Kondo, T. "Web-Based Intelligent Photo Browser for Flood of Personal Digital Photographs" (2008) [IEEExplore]
108. Paganelli, L. and Paterno, F. "Automatic reconstruction of the underlying interaction design of Web applications" (2002) [ACM]
109. Paganelli, L. and Paterno, F. "Intelligent analysis of user interactions with Web applications" (2002) [ACM]
110. Panach, J.I.; Valverde, F. and Pastor, O. "Improvement of a Web Engineering Method through Usability Patterns" (2007) [IWWUA]
111. Paolini, P. "Hypermedia, the Web and Usability issues" (1999) [IEEExplore]
112. Pascual, V. and Dursteler, J.C. "WET: a prototype of an Exploratory Search System for Web Mining to assess Usability" (2007) [IEEExplore]
113. Paterno, F. and Paganelli, L. "Remote automatic evaluation of Web sites based on task models and browser monitoring" (2001) [ACM]
114. Perlman, G. "Achieving Universal Usability by Designing for Change" (2002) IEEE IC
115. Piller, M.J. and Miller, M.S. "Dissociation of subjective Web site usability evaluation and performance: effect of user experience" (2001) [ACM]
116. Plaza, I.; Arcega, F.; Castro, M. and Tovar, E. "Work in Progress: Proposed Usability Model for the Analysis of Digital Electronic Educational Web Pages" (2006) [IEEExplore]
117. Qi, Y.; Reynolds, C. and Picard, R.W. "The Bayes Point Machine for computer-user frustration detection via PressureMouse" (2001) [ACM]
118. Ramli, R. and Jaafar, A. "e-RUE : A cheap possible solution for usability evaluation" (2008) [IEEExplore]
119. Rosenthal, A. "Redesign solution for civicinfo bc Web site" (2007) [ACM]
120. Roy, M.C.; Dewit, O. and Aubert, B.A. "The impact of interface usability on trust in Web retailers" (2001) [IR]
121. Saito, N.A.; Ogata, H.; Paredes, R.G.J.; Yano, Y. and San Martin, G.A. "Supporting classroom activities with the BSUL environment" (2005) [IEEExplore]
122. Santos, B.S.; Zamfir, F.; Ferreira, C.; Mealha, O. and Nunes, J. "Visual application for the analysis of Web-based information systems usage: a preliminary usability evaluation" (2004) [IEEExplore]
123. Saward, G.; Hall, T. and Barker, T. "Assessing usability through perceptions of information scent" (2004) [IEEExplore]
124. Scholtz, J. "Adaptation of traditional usability testing methods for remote testing" (2001) [IEEExplore]
125. Selvidge, P. "Reservations about the usability of airline Web sites" (1999) [ACM]
126. Seva, R.; Wu, J. and Yi, X. "Evaluation of Cinema Website" (2006) [IEEExplore]
127. Skov, M.B. and Stage, J. "Supporting problem identification in usability evaluations" (2005) [ACM]
128. Spalteholz, L. "KeySurf: A Character Controlled Browser for People with Physical Disabilities" (2008) [WWW]
129. Sperry, R.A. and Fernandez, J.D. "Usability testing using physiological analysis" (2008) [ACM]

130. Spool, J. and Schroeder, W. "Testing Web sites: five users is nowhere near enough" (2001) [ACM]
131. Ssemugabi, S. and De Villiers, R. "A comparative study of two usability evaluation methods using a Web-based e-learning application" (2007) [ACM]
132. Stry, C. and Eberle, P. "Building up usability-engineering capability by improving access to automated usability evaluation" (2008) [ScienceDirect]
133. Stolz, C.; Viermetz, M.; Barth, M. and Wilde, K. "Searchstrings revealing User Intent - A better Understanding of User Perception" (2006) [ICWE]
134. Sullivan, T. and Matson, R. "Barriers to use: usability and content accessibility on the Web's most popular sites" (2000) [ACM]
135. Sutcliffe, A. "Assessing the reliability of heuristic evaluation for Web site attractiveness and usability" (2002) [IEEEExplore]
136. Tao, Y.; Guo, S. "The design of a Web-based training system for simulation analysis" (2001) [IEEEExplore]
137. Taylor, M.; Wade, S. and England, D. "Informing IT system Web site design through normalization" (2003) [IR]
138. Thimbleby, H. "Gentler: a tool for systematic Web authoring" (1997) [IJHCS]
139. Thompson, K.E.; Rozanski, E.P. and Haake, A.R. "Here, there, anywhere: remote usability testing that works" (2004) [ACM]
140. Tirapat, T. and Achalakul, T. "Usability Assessment for Hyperlink Methods" (2006) [IEEEExplore]
141. Toleman, M.A. and Toleman, J.M. "User experiences and a usability inspection of an electronic services environment for students" (1998) [IEEEExplore]
142. Tonn-Eichstädt, H. "Measuring Website usability for visually impaired people-a modified GOMS analysis" (2006) [ACM]
143. Triacca, L.; Inversini, A. and Bolchini, D. "Evaluating Web usability with MiLE+" (2005) [IEEEExplore]
144. Tsiaousis, A.S. and Giaglis, G.M. "Evaluating the Effects of the Environmental Context-of-Use on Mobile Website Usability" (2008) [IEEEExplore]
145. Van den Haak, M.J. and De Jong, M.D.T. "Exploring two methods of usability testing: concurrent versus retrospective think-aloud protocols" (2003) [IEEEExplore]
146. Van Velsen, L.; Van der Geest, T. and Klaassen, R. "Testing the usability of a personalized system: comparing the use of interviews, questionnaires and thinking-aloud" (2007) [IEEEExplore]
147. Van Waes, L. "Thinking aloud as a method for testing the usability of Websites: the influence of task variation on the evaluation of hypertext" (2000) [IEEEExplore]
148. Vanderdonck, J.; Beirekdar, A. and Noirhomme-Fraiture, M. "Automated Evaluation of Web Usability and Accessibility by Guideline Review" (2004) [ICWE]
149. Wang, L.; Bretschneider, S. and Gant, J. "Evaluating Web-Based E-Government Services with a Citizen-Centric Approach" (2005) [IEEEExplore]
150. Wang, X. and Liu, J. "Usability Evaluation of B2C Web Site" (2007) [IEEEExplore]
151. Weir, C.; Anderson, J. and Jack, M. "On the role of metaphor and language in design of third party payments in eBanking: Usability and quality" (2006) [ScienceDirect]
152. West, R. and Lehman, K. "Automated summative usability studies: an empirical evaluation" (2006) [ACM]
153. Wilson, R.; Shortreed, J. and Landoni, M. "A study into the usability of e-encyclopaedias" (2004) [ACM]
154. Winckler, M.; Freitas, C. and Valdeni de Lima, J. "Usability remote evaluation for WWW" (2000) [ACM]
155. Wood, F.B.; Siegel, E.R.; LaCroix, E.-M.; Lyon, B.J.; Benson, D.A.; Cid, V. and Fariss, S. "A practical approach to e-government Web evaluation" (2003) [IEEEExplore]
156. Xu, L. and Xu, B. "Applying Agent into Intelligent Web Application Testing" (2007) [IEEEExplore]
157. Xu, L.; Xu, B. and Jiang, J. "Testing Web applications focusing on their specialties" (2005) [ACM]
158. Ytikseitiirk, E. (2004) "Usability evaluation of an online certificate program" [IEEEExplore]

159. Zaharias, P. "A usability evaluation method for e-learning: focus on motivation to learn" (2006) [ACM]
160. Zhang, P.; Small, R.V.; Von Dran, G.M. and Barcellos, S. "Websites that satisfy users: a theoretical framework for Web user interface design and evaluation" (1999) [IEEExplore]
161. Zhao, L. and Deek, F.P. "Exploratory inspection: a learning model for improving open source software usability" (2006) [ACM]
162. Zimmerman, D. and Stapel, L. "Strategies for Integrating Research and Evaluation in Web Site Development" (2006) [IEEExplore]
163. Zimmerman, D.; Slater, M. and Kendall, P. "Risk communication and usability case study: implications for Web site design" (2001) [IEEExplore]

ANEXO B: MODELO DE USABILIDAD WEB

Usabilidad: Perspectiva de Producto Software

Sub-característica		Atributo	Métrica
1. Facilidad de entendimiento	1.1 Legibilidad visual	1.1.1 Adecuación de fuente	Tamaños de fuente adecuados a cada contexto
		1.1.2 Adecuación de la visualización textual	Contraste de color
			Número de clusters textuales
			Número de palabras enfatizadas
		1.1.3 Disposición	Número de scrolls horizontales
	1.2 Facilidad de lectura	1.2.1 Agrupación Cohesiva de la Información	Proporción de información agrupada por tipo
			Centros de asociación semántica
			Cohesión
			Acoplamiento
		1.2.2 Densidad de información	Número de componentes
			Número de secciones
			Número de palabras
			Núm. de animaciones flash
			Núm. total de controles
			Duración media de clips de audio
			Duración media de clips de video
			Número de páginas
			Núm. de elementos multimedia
	Espacio físico usado		
			Número de imágenes
	1.3 Familiaridad	1.3.1 Consistencia de formato	Número de formatos distintos para datos
		1.3.3 Metáfora	Núm. de metáforas reconocidas
		1.3.2 Internacionalización	Núm. de comandos estandarizados empleados
	1.4 Ahorro de esfuerzo	1.4.1 Acciones mínimas	Disponibilidad de valores por defecto
			Disponibilidad de demostraciones
			Entradas y salidas fáciles de entender
		1.4.2 Auto-descripción	Compleitud de descripciones
Claridad de los elementos de la interfaz			
1.4.3 Complejidad de la información		Estructura	
		Complejidad de la página	
		Complejidad del audio	
		Complejidad del video	
		Complejidad de las animaciones	
		Complejidad ciclomática	
1.5 Orientación al usuario	1.5.1 Calidad de los mensajes de actualización	Proporción de mensajes de actualización significativos	
	1.5.2 Calidad de los mensajes de aviso	Proporción de mensajes de aviso significativos	
	1.5.3 Retroalimentación inmediata de los controles	Proporción de elementos que muestran el estado actual	
		Retro-alimentación orientada al sujeto	

Sub-característica		Atributo	Métrica
	1.6 Navegabilidad	1.6.1 Soporte a búsqueda interna	Disponibilidad de búsqueda interna
		1.6.2 Clickabilidad	Enlaces distinguibles
		1.6.3 Interconectividad	Compactibilidad
			Prestigio
			Estratos
			Entradas de navegación
			Salidas de navegación
			Distancia media de conexión
			Número total de enlaces
			Número total de nodos
			Distancia de salida convertida
			Distancia de entrada convertida
		1.6.4 Alcanzabilidad	Amplitud de la navegación
			Profundidad de la navegación
Densidad de la navegación			
Número de enlaces rotos			
	Número de nodos huérfanos		
2. Facilidad de aprendizaje	2.1 Predictibilidad	2.1.1 Nombres de enlaces significativos	Proporción de nombres adecuados para enlaces
			Ángulo de distinción Latent Semantic Analysis
		2.1.2 Etiquetas significativas	Proporción de nombres adecuados para etiquetas
		2.1.3 Controles significativos	Proporción de controles escogidos adecuadamente para cada función
		2.1.4 Contenido multimedia significativo	Proporción de nombres adecuados para enlaces
	2.2 Potencialidad	2.2.1 Determinación de acciones posibles	Número de enlaces en la misma sección
		2.2.2 Determinación de acciones prometedoras	Número de enlaces distintivos de cada sección
	2.3 Retroalimentación informativa	2.3.1 Progreso explícito de las transacciones	Proporción de elementos que muestran progreso de una transacción
		2.3.2 Contexto explícito del usuario	Proporción de elementos indicadores de páginas seguras
	3. Facilidad de uso	3.1 Compatibilidad	3.1.1 Compatibilidad con navegadores y plugins
Número de plugins necesarios			
3.1.2 Compatibilidad con sistemas operativos			Diferencias de comportamiento de los controles entre sistemas operativos
3.1.3 Compatibilidad con velocidades de conexión			Tiempo de carga entre velocidades
		3.1.4 Compatibilidad con la resolución de pantalla	Número de resoluciones de pantalla soportadas
3.2 Gestión de los datos		3.2.1 Validez de los datos de entrada	Proporción de mecanismos de validación de datos de entrada
		3.2.2 Visibilidad de los datos	Proporción de mecanismos que protegen los datos
3.3 Controlabilidad		3.3.1 Edición posterior	Operaciones de usuario editables
			Permisibilidad de corrección de errores

Sub-característica	Atributo	Métrica	
	3.3.2 Soporte a operaciones de cancelación	Operaciones de usuario cancelables	
	3.3.3 Soporte a la interrupción	Operaciones de usuario interrumpibles	
	3.3.4 Soporte a operaciones de deshacer	Proporción de operaciones con capacidad de ser deshechas	
	3.3.5 Soporte a operaciones de rehacer	Proporción de operaciones con capacidad de ser rehechas	
	3.3.6 Soporte a la impresión	Número de páginas imprimibles adecuadamente	
	3.4 Capacidad de adaptación	3.4.1 Adaptabilidad	Personalización
		3.4.2 Adaptativo	Reducción de procesos operativos
	3.5 Consistencia	3.5.1 Comportamiento constante de los enlaces	Proporción de enlaces con los mismos destinos
		3.5.2 Comportamiento constante de los controles	Proporción de controles con el mismo comportamiento
		3.5.3 Permanencia de los enlaces	Proporción de enlaces que permanecen en cada interfaz de usuario
		3.5.4 Permanencia de los controles	Proporción de controles que permanecen en cada interfaz de usuario
		3.5.5 Consistencia en el orden de los enlaces	Número de variaciones en el orden de los enlaces
		3.5.6 Consistencia en el orden de los controles	Número de variaciones en el orden de los controles
		3.5.7 Consistencia en las etiquetas	Número de etiquetas que se corresponden con el campo que hacen referencia
	3.6 Gestión de errores	3.6.1 Prevención de errores	Proporción de mecanismos de validación de datos de entrada
		3.6.2 Recuperación ante errores	Proporción de mecanismos de retorno a estado consistente.
		3.6.3 Calidad de los mensajes de error	Proporción de mensajes de error significativos.
	4. Facilidad de ayuda	4.1 Completitud de la ayuda online	Proporción de operaciones que han sido contempladas en la ayuda
			Idiomas disponibles
4.2 Documentación Multi-usuario		Proporción de usuarios que tengan todas sus acciones contempladas	
4.3 Completitud del mapa de sitio		Proporción de funcionalidades que aparecen en el mapa.	
4.4 Calidad de los mensajes de asesoramiento	Proporción de mensajes de asesoramiento significativos.		
5. Accesibilidad técnica	5.1 Soporte a la ampliación/reducción del texto	Incorporación de la función ampliar y reducir para el texto	
	5.2 Independencia de dispositivos de control	Número de dispositivos de entrada de datos operables.	
	5.3 Soporte a texto alternativo	Proporción de imágenes con texto alternativo asociado	
	5.4 Colores seguros	Número de colores propensos a la	

Sub-característica	Atributo	Métrica	
		epilepsia	
	5.5 Grado de conformidad con WCAG	Ratio de conformidad cubierto	
6. Grado de atracción	6.1 Uniformidad del color de fondo	Estilo de fondo	
	6.2 Uniformidad de la fuente	Estilo de fuente	
	6.3 Uniformidad en la posición de las secciones de la interfaz		Número de elementos no alineados o desencuadrados
			Variación en la composición de los marcos
	6.4 Personalización de la estética de la interfaz de usuario	Número de opciones de personalización estética	
6.5 Grado de interactividad	Tasa de información intercambiada entre usuario e interfaz.		
7. Adherencia a normas o convenciones	7.1 Grado de conformidad a la ISO/IEC 25000 SQuaRE [42]	Ratio de conformidad cubierto	
	7.2 Grado de conformidad a “Research-Based Web Desing & Usability Guidelines” [50].	Ratio de conformidad cubierto	
	7.3 Grado de conformidad a “Web Style Guide” [53].	Ratio de conformidad cubierto	
	7.4 Grado de conformidad a “Microsoft Web Design Guidelines” [59]	Ratio de conformidad cubierto	
	7.5 Grado de conformidad a “Sun Guide to Web Style” [75]	Ratio de conformidad cubierto	
	7.6 Grado de conformidad a “IBM Web Design Guidelines” [34]	Ratio de conformidad cubierto	

Usabilidad: Perspectiva de Calidad en Uso

Sub-característica	Atributo	Métrica	
8. Efectividad en uso	8.1. Facilidad de ayuda	8.1.1 Efectividad de la ayuda online	Legibilidad del tutorial
			Efectividad del sistema de ayuda
			Facilidad de uso del sistema de ayuda
		8.1.2 Completitud de la ayuda online	Proporción de funcionalidades no cubiertas en las consultas del usuario
		8.1.3 Frecuencia de consulta de ayuda	Número de veces que el usuario accede a la ayuda por tarea
	8.2 Rendimiento de las tareas del usuario	8.2.1 Completado de las tareas	Número de tareas completadas
8.2.2 Exactitud de las tareas		Número de tareas completadas de forma adecuada	
9. Eficiencia en uso	9.1 Eficiencia de las tareas del usuario	9.1.1 Tiempo para completar las tareas	Tiempo medio necesario para cumplir una tarea
		9.1.2 Carga de la tarea	Índice User Task Load (UTLindex)
	9.2 Esfuerzo cognitivo	9.2.1 Esfuerzo mental subjetivo	Tasa Subjective Mental Effort (SME ratio)
		9.2.2 Facilidad de recordar (memorabilidad)	
	9.3 Limitaciones del contexto	9.3.1 Carga del sistema	Memoria consumida durante el uso de la aplicación Web
		9.3.2 Adaptación a las habilidades del usuario	

Sub-característica		Atributo	Métrica
10. Satisfacción en Uso	10.1 Satisfacción cognitiva	10.1.1 Utilidad percibida	Número de funcionalidades que el usuario encuentra útiles
		10.1.2 Calidad de los resultados	Número de funcionalidades que el usuario esperaba encontrar
	10.2 Satisfacción emocional	10.2.1 Atracción subjetiva percibida	Número de comentarios positivos del usuario
		10.2.2 Frustración percibida	Número de interrupciones en una tarea
		10.2.3 Riesgos del contenido	Número de comentarios negativos acerca del contenido
	10.3 Satisfacción física		Número de comentarios positivos del usuario
	10.4 Confianza	10.4.1 Aparición de errores	Número de errores entre operaciones.
		10.4.2 Credibilidad del sitio	Calidad de las impresiones del usuario
		10.4.3 Riesgos económicos	Número de incidentes involucrando pérdidas económicas
11. Usabilidad en uso Adherida a normas o convenciones	11.1 Grado de conformidad a la ISO/IEC 25000 SQuaRE [42]		Ratio de conformidad cubierto
	11.2 Grado de conformidad a criterios ergonómicos [6].		Ratio de conformidad cubierto
	11.3 Grado de conformidad con cuestionarios SUMI, SUS y QUIS		Ratio de conformidad cubierto

ANEXO C: WUEP DEFINIDO EN SPEM 2 USANDO EPF-COMPOSER

Vista Consolidada de Eclipse Process Framework Composer para un proceso de desarrollo Web genérico que sigue el enfoque DSDM.

Presentation Name	I.	P.	Model Info	Type
[-] Desarrollo Web DSDM	0			Capability Pattern
[-] Elicitación de Requisitos	1			Task Descriptor
Cliente			Primary Performer	Role Descriptor
Ing. Requisitos			Primary Performer	Role Descriptor
Requerimientos			Mandatory Input	Artifact Descriptor
Modelos CIM			Output	Artifact Descriptor
Modelos Requisitos Funcionales			Output	Artifact Descriptor
Modelos Requisitos No-Funcionales			Output	Artifact Descriptor
[-] Análisis	2	1		Task Descriptor
Analista			Primary Performer	Role Descriptor
Modelos CIM			Mandatory Input	Artifact Descriptor
Modelos Requisitos Funcionales			Mandatory Input	Artifact Descriptor
Modelos Requisitos No-Funcionales			Mandatory Input	Artifact Descriptor
Informe de Mejora en Analisis			Optional Input	Artifact Descriptor
Modelos PIM de Análisis			Output	Artifact Descriptor
[-] Diseño	3	2		Task Descriptor
Diseñador			Primary Performer	Role Descriptor
Modelos PIM de Análisis			Mandatory Input	Artifact Descriptor
Informe de Mejora en Diseño			Optional Input	Artifact Descriptor
Modelos PIM de Diseño			Output	Artifact Descriptor
[-] Transformación de Modelos	4	3		Task Descriptor
Compilador			Primary Performer	Role Descriptor
Modelos PIM de Análisis			Mandatory Input	Artifact Descriptor
Modelos PIM de Diseño			Mandatory Input	Artifact Descriptor
Informe de Mejora en Compilación de Modelos			Optional Input	Artifact Descriptor
Modelos PSM			Output	Artifact Descriptor
[-] Generación de Código	5	4		Task Descriptor
Compilador			Primary Performer	Role Descriptor
Modelos PSM			Mandatory Input	Artifact Descriptor
Informe de Mejora en Generación de Código			Optional Input	Artifact Descriptor
Codigo Fuente CM			Output	Artifact Descriptor

Nota: En la captura no se han mostrado las columnas referentes a los flags: Planned, Repeatable, Multiple Occurrences, Ongoing, Event-Driven, Optional, Team. Estos flags hacen referencia a características de las tareas para ser exportadas a un plan de proyecto.

Vista Consolidada de Eclipse Process Framework Composer para WUEP (Web Usability Evaluation Process).

Presentation Name	I.	P..	Model Info	Type	F
WUEP	0			Capability Pattern	
Establecimiento de los Requisitos Evaluación	1			Activity	
Establecimiento del propósito de la evaluación	2			Task Descriptor	
Diseñador Evaluación			Primary Performer	Role Descriptor	
Modelos Requisitos No-Funcionales			Mandatory Input	Artifact Descriptor	
Propósito de Evaluación			Output	Outcome Descriptor	
Especificación de Perfiles	3			Task Descriptor	
Diseñador Evaluación			Primary Performer	Role Descriptor	
Modelos Requisitos No-Funcionales			Mandatory Input	Artifact Descriptor	
Perfil de Evaluación			Output	Outcome Descriptor	
Selección de atributos de usabilidad	4	2,3		Task Descriptor	
Diseñador Evaluación			Primary Performer	Role Descriptor	
Modelos Requisitos No-Funcionales			Mandatory Input	Artifact Descriptor	
Perfil de Evaluación			Mandatory Input	Outcome Descriptor	
Propósito de Evaluación			Mandatory Input	Outcome Descriptor	
Atributos a Evaluar			Output	Outcome Descriptor	
Elaboración del documento "Requisitos de evaluación"	5	4		Task Descriptor	
Diseñador Evaluación			Primary Performer	Role Descriptor	
Atributos a Evaluar			Mandatory Input	Outcome Descriptor	
Perfil de Evaluación			Mandatory Input	Outcome Descriptor	
Propósito de Evaluación			Mandatory Input	Outcome Descriptor	
Requisitos de Evaluación			Output	Artifact Descriptor	

Nota: En la captura no se han mostrado las columnas referentes a los flags: *Planned*, *Repeatable*, *Multiple Occurrences*, *Ongoing*, *Event-Driven*, *Optional*, *Team*. Estos flags hacen referencia a características de las tareas para ser exportadas a un plan de proyecto.

Presentation Name	I.	P..	Model Info	Type
[-] [Icon] Especificación de la Evaluación	6			Activity
[-] [Icon] Selección de Artefactos a Evaluar	7			Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Artefactos a evaluar			Output	Outcome Descriptor
[-] [Icon] Selección de Metricas a Emplear	8	7		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Artefactos a evaluar			Mandatory Input	Outcome Descriptor
[Icon] Atributos a Evaluar			Mandatory Input	Outcome Descriptor
[Icon] Métricas a emplear			Output	Outcome Descriptor
[-] [Icon] Operacionalización de métricas	9	8		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Métricas a emplear			Mandatory Input	Outcome Descriptor
[Icon] Métricas operacionalizadas			Output	Outcome Descriptor
[-] [Icon] Establecimiento de umbrales e indicadores	10	9		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Métricas operacionalizadas			Mandatory Input	Outcome Descriptor
[Icon] Umbrales e indicadores			Output	Outcome Descriptor
[-] [Icon] Establecimientos de criterios de normalización y agrega	11	10		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Métricas operacionalizadas			Mandatory Input	Outcome Descriptor
[Icon] Criterios de agregación			Output	Outcome Descriptor
[-] [Icon] Definición de la plantilla para informes de usabilidad	12			Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Plantilla Informe Usabilidad			Output	Artifact Descriptor
[-] [Icon] Elaboración del documento "Especificación de evaluació	13	1...		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Artefactos a evaluar			Mandatory Input	Outcome Descriptor
[Icon] Atributos a Evaluar			Mandatory Input	Outcome Descriptor
[Icon] Criterios de agregación			Mandatory Input	Outcome Descriptor
[Icon] Métricas operacionalizadas			Mandatory Input	Outcome Descriptor
[Icon] Umbrales e indicadores			Mandatory Input	Outcome Descriptor
[Icon] Especificación de la Evaluación			Output	Artifact Descriptor

Presentation Name	I.	P..	Model Info	Type
[-] [Icon] Diseño de la Evaluación	14			Activity
[-] [Icon] Definición de restricciones	15			Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Especificación de la Evaluación			Mandatory Input	Artifact Descriptor
[Icon] Restricciones evaluación			Output	Outcome Descriptor
[-] [Icon] Elaboración Plan de Evaluación	16	15		Task Descriptor
[Icon] Diseñador Evaluación			Primary Performer	Role Descriptor
[Icon] Restricciones evaluación			Mandatory Input	Outcome Descriptor
[Icon] Plan de Evaluación			Output	Artifact Descriptor

Presentation Name	I.	P	Model Info	Type
Ejecución de la Evaluación	17			Iteration
Evaluación de Artefactos	18			Activity
Evaluación a nivel PIM	19			Task Descriptor
Ejecutor Evaluación			Primary Performer	Role Descriptor
Especificación de la Evaluación			Mandatory Input	Artifact Descriptor
Modelos PIM de Análisis			Mandatory Input	Artifact Descriptor
Modelos PIM de Diseño			Mandatory Input	Artifact Descriptor
Plan de Evaluación			Mandatory Input	Artifact Descriptor
Plantilla Informe Usabilidad			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad PIM			Output	Artifact Descriptor
Evaluación a nivel PSM	20			Task Descriptor
Ejecutor Evaluación			Primary Performer	Role Descriptor
Especificación de la Evaluación			Mandatory Input	Artifact Descriptor
Modelos PSM			Mandatory Input	Artifact Descriptor
Plan de Evaluación			Mandatory Input	Artifact Descriptor
Plantilla Informe Usabilidad			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad PSM			Output	Artifact Descriptor
Evaluación a nivel CM	21			Task Descriptor
Ejecutor Evaluación			Primary Performer	Role Descriptor
Codigo Fuente CM			Mandatory Input	Artifact Descriptor
Especificación de la Evaluación			Mandatory Input	Artifact Descriptor
Plan de Evaluación			Mandatory Input	Artifact Descriptor
Plantilla Informe Usabilidad			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad CM			Output	Artifact Descriptor
Evaluación en Uso	22			Task Descriptor
Ejecutor Evaluación			Primary Performer	Role Descriptor
Usuario Final			Primary Performer	Role Descriptor
Especificación de la Evaluación			Mandatory Input	Artifact Descriptor
Experiencia en uso			Mandatory Input	Outcome Descriptor
Plan de Evaluación			Mandatory Input	Artifact Descriptor
Plantilla Informe Usabilidad			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad en Uso			Output	Artifact Descriptor
Análisis de cambios	23			Task Descriptor
Ejecutor Evaluación			Primary Performer	Role Descriptor
Informe de Problemas de Usabilidad CM			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad en Uso			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad PIM			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad PSM			Mandatory Input	Artifact Descriptor
Informe de Problemas de Usabilidad en Analisis			Output	Artifact Descriptor
Informe de Problemas de Usabilidad en Transformación de Mod.			Output	Artifact Descriptor
Informe de Problemas de Usabilidad en Diseño			Output	Artifact Descriptor
Informe de Problemas de Usabilidad en Generación de Código			Output	Artifact Descriptor