



Universidad de Castilla-La Mancha

**Departamento de Tecnologías y Sistemas de
Información**

Tesis Doctoral

**Entorno de Trabajo para la Gestión de
Requisitos de Calidad de Datos en el
Desarrollo de Aplicaciones Web**

Doctorando: César Arturo Guerra García

Director: Dr. D. Ismael Caballero Muñoz-Reja



Universidad de Castilla-La Mancha

**Departamento de Tecnologías y Sistemas de
Información**

Tesis Doctoral

**Entorno de Trabajo para la Gestión de
Requisitos de Calidad de Datos en el
Desarrollo de Aplicaciones Web**

Ciudad Real, Diciembre de 2011

Doctorando: César Arturo Guerra García

Director: Dr. D. Ismael Caballero Muñoz-Reja

El rápido crecimiento de la Web ha hecho posible que cada vez volúmenes más grandes de datos estén disponibles, permitiendo a los usuarios de aplicaciones Web un acceso cada vez más fácil a ellos. De esta forma, la Web se ha consolidado en sí misma como un importante recurso de datos. Dado el valor estratégico de los datos en la ejecución de procesos de negocio, y teniendo en cuenta que cada vez con más frecuencia estos datos son publicados y consumidos desde Internet, los individuos y las organizaciones necesitan controlar la calidad de los datos, tratando de asegurar niveles aceptables para las aplicaciones que utilizan.

Lamentablemente, y hasta la fecha, no es frecuente encontrar aplicaciones Web que implementan algún tipo de mecanismo para la gestión de la Calidad de los Datos (DQ). Para implementar estos mecanismos, los desarrolladores de aplicaciones Web tienen que conocer primero cómo los usuarios perciben y entienden el concepto de calidad de datos de acuerdo a las tareas que ellos realizan, y entonces obtener y gestionar tales **requisitos de DQ**. Para lograr esto, los desarrolladores necesitan disponer de mecanismos útiles que les permitan capturar, representar e incluir esta percepción en el desarrollo de aplicaciones Web desde la etapa inicial, como si éstos fueran otros requisitos software.

Desafortunadamente, y a pesar de los beneficios que pueden ser objetivamente obtenidos con el uso de tales mecanismos, el área de desarrollo Web no ha abordado aún la *gestión de requisitos específicos de DQ*, habiendo llegado a esta conclusión tras una pertinente revisión sistemática. En esta revisión no se encontró ningún trabajo que presentara algún tipo de propuesta metodológica o tecnológica desarrollada por investigadores en el área de Ingeniería de Software o Calidad de Datos. Por lo tanto, como se puede observar, es un problema que requiere solución tanto a nivel metodológico como tecnológico.

Así, *el principal objetivo de esta tesis es **generar los mecanismos y artefactos necesarios, tanto metodológicos como tecnológicos, que permita a los desarrolladores de aplicaciones Web la gestión de requisitos específicos de calidad de datos (DQR's), estos requisitos de calidad de datos complementarán al resto de los requisitos software para aplicaciones Web, con el objetivo de desarrollar aplicaciones sensibles de calidad de datos.***

Abstract

The rapid growth of the Web has made possible that more and more data resources become available, thus allowing users increasingly easy access to them. Thus, the Web has established itself as an important source of data. Given the strategic value of data in the running of business processes, and taken into account that more frequently this data is published to, and consumed from Internet, individuals and organizations need to monitor the quality of data, trying to assure acceptable levels of quality for the applications that use them.

However, most of the current Web applications do not implement mechanisms being able to manage Data Quality (DQ). To implement these mechanisms, Web application developers must first know how users understand the concept of data quality according to their tasks, and then to gather and manage such data quality requirements. To better achieve this, developers need useful mechanisms that allow them capturing, representing and include these perception in Web applications development from the earliest stages.

Unfortunately, and despite the objective benefits that can be achieved from the use of such mechanisms, the area of Web application development has not still addressed the *management of data quality requirements*, having reached this conclusion once done a pertinent systematic review. In this revision, we have not found any works presenting methodological or technological proposals developed by researchers in Software Engineering or Data Quality. *Therefore, as can be seen, it is a problem requiring solution at both of methodological and technological levels.*

Thus, *the main goal of this thesis is to propose a **framework that permit to generate the necessary mechanism both methodological and technological, to allow Web application developers a correct management of specific Data Quality Requirements (DQR). These DQRs will complement the remainder of the software requirements for the Web Applications, in order to build Data Quality Awareness Applications.***

Agradecimientos

Le doy las gracias al amor de mi vida, por siempre apoyarme y estar en todo momento a mi lado, muchas gracias Anabel.

Agradezco a toda mi familia, especialmente a mis padres, por inculcarme que el trabajo y la dedicación siempre te dará una recompensa a cambio.

Agradezco a mi tutor, el Dr. Ismael Caballero por toda su guía y orientación, pero sobre todo por su gran amistad. Muchas Gracias Ismael.

Agradezco al Dr. Mario Piattini por todos sus comentarios y revisiones para llevar a buen término cada uno de los trabajos realizados. Además de brindarme siempre su apoyo y confianza durante el tiempo que viví en España.

Agradezco a todos mis compañeros y amigos del grupo Alarcos, por recibirme y hacer de mi estancia en España un tiempo mucho más placentero.

Agradezco a la Dra. Laure Berti-Equille por toda su ayuda y orientación en la realización de la estancia doctoral en Francia.

Agradezco a la UPSLP y al programa PROMEP, por el otorgamiento de la beca para llevar a cabo los estudios de doctorado.

A Dios por todo.

Índice de Contenidos

1	INTRODUCCIÓN.....	9
1.1	PLANTEAMIENTO Y JUSTIFICACIÓN DEL TRABAJO DESARROLLADO EN ESTA TESIS.....	10
1.2	HIPÓTESIS Y OBJETIVOS.....	13
1.3	MARCO DE TRABAJO.....	15
1.4	ORGANIZACIÓN DE LA TESIS.....	17
2	ESTADO DEL ARTE.....	19
2.1	CALIDAD DE DATOS.....	20
2.2	INGENIERÍA WEB.....	26
2.2.1	<i>Ingeniería Web Dirigida por Modelos</i>	29
2.3	INGENIERÍA DE REQUISITOS.....	34
3	ENTORNO DE TRABAJO PARA LA GESTIÓN DE REQUISITOS DE CALIDAD DE DATOS EN APLICACIONES WEB	37
3.1	DEFINICIÓN DEL CONCEPTO DE REQUISITO DE CALIDAD DE DATOS	39
3.2	METODOLOGÍA PARA LA GESTIÓN DE PROYECTOS DE ESPECIFICACIÓN DE REQUISITOS DE CALIDAD DE DATOS (DQ-VORD).....	41
3.2.1	<i>Descripción de la Metodología</i>	41
3.2.2	<i>Catálogo de Roles</i>	43
3.2.3	<i>Catálogo de Elementos</i>	47
3.2.4	<i>Descripción de las actividades de la metodología DQ-VORD</i>	51
3.2.5	<i>Modelo propuesto para la Selección de Requisitos de Calidad de Datos según las Funcionalidades a Implementar en una Aplicación Web (MOSCAF)</i>	61
3.2.6	<i>Metamodelo Ampliado para la Especificación de Requisitos de DQ en Aplicaciones Web (DQ_WebRE)</i>	76
3.2.7	<i>Metamodelo para el Diseño de Aplicaciones Web sensibles a DQ (DAQ_UWE)</i>	87
3.2.8	<i>Plantillas de documentos</i>	100
4	CASO DE ESTUDIO DE VALIDACIÓN	103
4.1	CASO DE ESTUDIO	104
4.1.1	<i>Documento de Especificación de Requisitos Software</i>	104
4.1.2	<i>Aplicación de la metodología DQ-VORD</i>	106
4.2	LECCIONES APRENDIDAS.....	125
5	CONCLUSIONS AND FUTURE RESEARCH LINES	129
5.1	ANALYSIS OF ACHIEVED OBJECTIVES.....	130
5.2	MAIN CONTRIBUTIONS OF THIS THESIS.....	132
5.3	CONTRAST OF RESULTS	133
5.4	FUTURE WORK.....	135
6	ANEXOS.....	137
6.1	ANEXO A. REVISIÓN SISTEMÁTICA REALIZADA.....	137

6.1.1	<i>Planificación de la Revisión Sistemática</i>	138
6.1.2	<i>Ejecución de la Selección</i>	141
6.1.3	<i>Análisis de Resultados Obtenidos</i>	142
6.1.4	<i>Conclusiones de la Revisión Sistemática Realizada</i>	143
6.1.5	<i>Extracción de información obtenida de los estudios primarios encontrados.</i>	144
6.2	ANEXO B. PLUGIN IMPLEMENTADO EN LA PLATAFORMA ECLIPSE.	149
6.3	ANEXO C. DESCRIPCIÓN DE RELACIONES ENTRE PROBLEMAS EN UN SI Y FUNCIONALIDADES WEB.....	151
6.4	ANEXO D. LISTA DE ACRÓNIMOS	157
7	REFERENCIAS	159

Índice de Tablas

Tabla 1. Características de DQ según el estándar ISO 25012 (ISO-25012 2008).....	24
Tabla 2. Etapas principales del método VORD.	36
Tabla 3. Productos, técnicas y roles involucrados en la tarea “Identificación de las funcionalidades críticas susceptibles de requerir ser complementadas con requisitos específicos de DQ (IFPW).”	53
Tabla 4. Productos, técnicas y roles involucrados en la tarea “Identificación de los puntos de vista del portal Web (IPVW).”	53
Tabla 5. Productos, técnicas y roles involucrados en la tarea “Identificación de requisitos de DQ (IRDQ)”	54
Tabla 6. Productos, técnicas y roles involucrados en la tarea “Clasificación de los puntos de vista (CPVW)”	55
Tabla 7. Productos, técnicas y roles involucrados en la tarea “Clasificación de los requisitos de calidad de datos (CRDQ)”	56
Tabla 8. Productos, técnicas y roles involucrados en la tarea “Documentación de los Puntos de Vista (DPVW)”	57
Tabla 9. Productos, técnicas y roles involucrados en la tarea “Documentación de los requisitos de calidad de datos (DRDQ)”	58
Tabla 10. Productos, técnicas y roles involucrados en la tarea “Modelado de los requisitos software de calidad de datos (MRDQ)”	60
Tabla 11. Productos, técnicas y roles involucrados en la tarea “Validación de los resultados de la metodología (VM)”	60
Tabla 12. Funcionalidades de una aplicación Web.	62
Tabla 13. Matriz de relaciones de funcionalidades del portal y obstáculos identificados.	66
Tabla 14. Comparativa de dimensiones/características de DQ.	67
Tabla 15. Dimensiones de DQ que son afectadas por cada uno de los obstáculo.	68
Tabla 16. Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web - MOSCAF.....	69

Tabla 17. Gestión de Contenido vs dimensiones de DQ.....	70
Tabla 18. Procesos y Acciones vs dimensiones de DQ.	70
Tabla 19. Capacidades de búsqueda vs dimensiones de DQ.....	70
Tabla 20. Administración vs dimensiones de DQ.....	71
Tabla 21. Seguridad vs dimensiones de DQ.	71
Tabla 22. Integración y Puntos de Datos vs dimensiones de DQ.....	71
Tabla 23. Comunicación y Colaboración vs dimensiones de DQ.....	72
Tabla 24. Presentación vs dimensiones de DQ.....	72
Tabla 25. Taxonomía vs dimensiones de DQ.....	72
Tabla 26. Personalización vs dimensiones de DQ.	72
Tabla 27. Características de Ayuda vs dimensiones de DQ.	73
Tabla 28. Productos de E/S del paso 1 “Entendimiento del dominio”.	74
Tabla 29. Productos de E/S del paso 2 “Determinar los requisitos de DQ”.....	74
Tabla 30. Productos de E/S del paso 3 “Instanciar y particularizar el requisito con la definición de las dimensiones de DQ seleccionadas”.....	75
Tabla 31. Productos de E/S del paso 4 “Definir una priorización de las dimensiones de DQ dentro de la instancia”.....	75
Tabla 32. Elementos del metamodelo WebRE.....	78
Tabla 33. Especificación de estereotipos definidos en el perfil DQ_WebRE.	82
Tabla 34. Estereotipos definidos en UWE.....	90
Tabla 35. Especificación de estereotipos de DQ del perfil DAQ_UWE.....	94
Tabla 36. Propuesta de reglas de mapeo de elementos “DQ_WebRE” a elementos “DAQ_UWE”. ...	95
Tabla 37. Especificación de funcionalidades Web.	100
Tabla 38. Especificación del Punto de Vista.....	100
Tabla 39. Especificación del Requisitos del sistema.	101
Tabla 40. Petición de modificación.	101
Tabla 41. Plantilla para la Gestión de la Configuración.	102
Tabla 42. Ejemplo de Documento de Especificación de Requisitos Software.	105
Tabla 43. Listado de Requisitos Funcionales identificados.	106

Tabla 44. Lista de funcionalidades Web susceptibles a requisitos de DQ (LFWI).....	107
Tabla 45. Lista de puntos de vista identificados capaces de proponer requisitos de DQ (PVI).	107
Tabla 46. Instanciación del modelo de DQ para el problema particular bajo estudio (IMDQ).....	108
Tabla 47. Nivel de importancia de los requisitos funcionales.	109
Tabla 48. Lista jerarquizada de los puntos de vista (LJPV).....	109
Tabla 49. Lista de clasificación de los requisitos de calidad de datos (LCRDQ).....	109
Tabla 50. Especificación del punto de vista "Administrador".	110
Tabla 51. Especificación del punto de vista "Empleado".	110
Tabla 52. Especificación del punto de vista "Empleado por comisión".....	111
Tabla 53. Especificación del punto de vista "Jefe de departamento".	111
Tabla 54. Especificación del punto de vista "Cliente".	111
Tabla 55. Especificación del punto de vista "Usuario".	112
Tabla 56. Especificación de requisito "Login al sistema".	112
Tabla 57. Especificación de requisito "Mantener información de empleados".	112
Tabla 58. Especificación de requisito "Ejecutar reporte de empleados".	112
Tabla 59. Especificación de requisito "Búsqueda de productos".....	113
Tabla 60. Especificación de requisito "Ejecutar pago de nómina".	113
Tabla 61. Especificación de requisito "Escoger método de pago".	113
Tabla 62. Especificación de requisito "Mantener timecard".....	113
Tabla 63. Especificación de requisito "Mantener órdenes de venta".	114
Tabla 64. Especificación de requisito "Solicitar un producto".	114
Tabla 65. Especificación de requisito "Mantener información de productos".	114
Tabla 66. Especificación de requisito "Ejecutar reporte administrativo".	114
Tabla 67. Palabras clave consideradas en la RS.	138
Tabla 68. Lista de recursos.....	139
Tabla 69. Distribución de estudios por recurso.	141
Tabla 70. Tecnología o Modelo utilizado por los estudios seleccionados.	143
Tabla 71. Estudio primario Wang et al. (Wang y Madnick 1993).	144

Tabla 72. Estudio primario Wang et al. (Wang, Reddy et al. 1995).	145
Tabla 73. Estudio primario Becker et al. (Becker, McMullen et al. 2007).	145
Tabla 74. Estudio primario Scannapieco et al. (Scannapieco, Pernici et al. 2002).	146
Tabla 75. Estudio primario Wang (Wang 1998).	146
Tabla 76. Estudio primario Caballero et al. (Caballero, Verbo et al. 2008).....	147
Tabla 77. Estudio primario Missier et al. (Missier, Embury et al. 2006).....	147
Tabla 78. Estudio primario Gomes et al. (Gomes, Farinha et al. 2007).	148

Índice de Figuras

Figura 1. Ejemplo de características de calidad de datos.	22
Figura 2. Perspectiva de Calidad de Datos.	23
Figura 3. Ejemplo de puntos de vista para una aplicación Web de una librería.	34
Figura 4. Perspectiva general de las propuestas de este trabajo de Tesis	38
Figura 5. Convergencia de los Flujos de trabajo del PUD con la metodología DQ-VORD.	41
Figura 6. Roles involucrados en la metodología DQ-VORD.	44
Figura 7. Actividades principales de la metodología DQ-VORD.	51
Figura 8. Tareas definidas en la actividad 1 "IPPW".	52
Figura 9. Tareas definidas en la actividad 2 "EP"	55
Figura 10. Tareas definidas en la actividad 3 "DP"	57
Figura 11. Tareas definidas en la actividad 4 "TPS"	59
Figura 12. Funcionalidad Web: Capacidad de búsqueda.	63
Figura 13. Funcionalidad Web: Comunicación y colaboración.	64
Figura 14. Funcionalidad Web: Personalización.	64
Figura 15. Análisis realizados para obtener el modelo MOSCAF.	65
Figura 16. Pasos de la guía de uso del modelo MOSCAF.	73
Figura 17. Metamodelo WebRE.	78
Figura 18. Metamodelo para especificación de requisitos de calidad de datos DQ_WebRE	80
Figura 19. Perfil UML para la especificación de requisitos de calidad de datos - DQ_WebRE	81
Figura 20. Diagrama de casos de uso especificando requisitos de DQ.	84
Figura 21. Diagrama de actividades con gestión de DQ.	85
Figura 22. Paquete de Navegación de UWE.	88
Figura 23. Paquete de Presentación de UWE.	89
Figura 24. Paquete de Proceso de UWE.	89
Figura 25. Paquete de Presentación con nuevo elemento para gestión de DQ.	92

Figura 26. Paquete de Proceso con elementos para gestión de DQ.	92
Figura 27. Paquete de Contenido con elementos para gestión de DQ	93
Figura 28. Ejemplo de diagrama del Modelo de Contenido con gestión de DQ	96
Figura 29. Modelo de Navegación con gestión de DQ.	97
Figura 30. Modelo de Presentación incluyendo elementos de DQ.....	99
Figura 31. Diagrama de casos de uso.....	106
Figura 32. Diagrama de casos de uso de los puntos de vista “Administrador” y “Jefe de departamento”, especificando requisitos de DQ.	116
Figura 33. Diagrama de casos de uso de los puntos de vista “Empleado” y “Empleado por comisión”, especificando requisitos de DQ.....	117
Figura 34. Diagrama de casos de uso de los puntos de vista “Cliente” y “Usuario”, especificando requisitos de DQ.	118
Figura 35. Diagrama de actividades con gestión específica de DQ.	120
Figura 36. Modelo de Contenido con elementos para la Gestión de DQ.	122
Figura 37. Modelo de Navegación con elementos para la Gestión de DQ.....	123
Figura 38. Modelo de Presentación incluyendo elementos para la Gestión de DQ.	124
Figura 39. Fases principales del protocolo de RS propuesto por Biolchini.....	137

1

INTRODUCCIÓN

*“Comprender las cosas que nos rodean es la mejor preparación
para comprender las cosas que hay más allá”
– Hipatía (aprox. 370 - 415)*

En esta sección se presenta el planteamiento y justificación de la tesis, además de la hipótesis y los objetivos de este trabajo. Asimismo, se describe el marco de trabajo en el que se encuadra esta tesis.

1.1 Planteamiento y Justificación del trabajo desarrollado en esta tesis.

La Web se ha consolidado como una de las fuentes de datos más importantes (Mahdavi, Shepherd et al. 2004; Phan y Vogel 2010). Las aplicaciones Web proporcionan a sus usuarios un entorno de trabajo más sencillo e intuitivo, permitiéndoles encontrar los datos necesarios para desempeñar sus funciones estratégicas y operacionales de forma más rápida (Phan y Vogel 2010); ayudando así a una toma de decisiones correcta e inmediata (Collins 2001).

Sin embargo, la aparición de problemas debido a niveles inadecuados en la calidad de datos (DQ), puede llegar a afectar negativamente el éxito en las tareas realizadas por las personas y en consecuencia el desempeño de las organizaciones (Caro, Calero et al. 2008). Este hecho ha sido observado no solo desde el ámbito académico, sino también empresarial (Friedman y Bitterer 2011).

En (Loshin 2001; Brien 2011) se menciona algunas *“historias de terror”* en el que problemas relacionados con la calidad de datos llegan a generar una gran problemática en las empresas, perjudicando en gran medida el rendimiento de sus procesos de negocio. Como resultado, estos problemas pueden causar un impacto negativo en las organizaciones, con un importante coste (y no sólo en términos económicos (Cappiello, Daniel et al. 2010)), sino también social (Laudon 1986; Eppler y Helfert 2004; Mehmood, Si-Said et al. 2009).

Una encuesta realizada por la empresa *Gartner* (Gartner 2009), reveló que los costes relacionados con aspectos de calidad de datos son estimados en millones de dólares *“una pobre calidad de datos cuesta a las compañías millones de dólares anualmente”* (Thi, Yao et al. 2010). Por otro lado, *Thomson Reuters and Lepus* llevaron a cabo otra encuesta en 2010 (ThomsonReuters y Lepus 2010), en la que se revela que más de tres cuartas (77%) partes de las empresas encuestadas tiene la intención de aumentar el gasto en proyectos que abordan cuestiones y estrategias de mejora en la calidad de datos, porque *“las empresas están cada vez más tratando de explotar la consistencia de los datos para apoyar los procesos que utilizan grandes cantidades de datos para gestionar riesgos, y lograr así una visión más amplia de ellos”* (ThomsonReuters y Lepus 2010).

Por otra parte, Oracle en su informe (Oracle 2008) muestra un análisis de resultados a nivel internacional en una encuesta realizada a trescientas compañías multinacionales que usan aplicaciones de inteligencia de negocios; en este informe, y como conclusión, se demanda la necesidad de que los usuarios de las aplicaciones puedan confiar en los datos como una forma fundamental de asegurar el éxito en sus tareas, y por tanto en sus negocios. Además, en un informe realizado por la organización IDC sobre calidad de datos en empresas españolas (IDC 2009),

se refleja que las organizaciones consideran la calidad de los datos como un factor determinante para el éxito de sus actividades.

Aunado a lo anterior, como lo demuestra el informe presentado en (Lwanga, Walenta et al. 2011), es necesario la figura de un profesional en calidad de datos dentro de una empresa, cuya función principal es la de asegurar el cumplimiento de estándares y regulaciones relacionadas con la gestión de DQ en los distintos proyectos empresariales.

Así, junto a los distintos tipos de aplicaciones Web y las nuevas tecnologías de desarrollo en las que se apoyan (por ejemplo, la *Ingeniería Web Dirigida por Modelos*), se crea la necesidad de que personas y organizaciones deban introducir mecanismos que permitan controlar niveles aceptables de calidad en los datos que publican y usan (*fitness for use*), pues saben del valor estratégico de los datos en el desarrollo de los procesos de negocio (Yang, Cai et al. 2004; Caro, Calero et al. 2008).

Por tanto, llevar a cabo una correcta gestión de calidad de datos, permitirá asegurar la presencia de datos con niveles adecuados de calidad en las aplicaciones Web, convirtiéndose así en una necesidad crítica para las organizaciones. Esto es debido principalmente a que este tipo de aplicaciones se han convertido en un activo estratégico (Baskarada, Gao et al. 2006) como importantes recursos de información, ya que son utilizadas diariamente por una gran cantidad de personas para llevar a cabo sus tareas.

Es una tendencia común abordar la DQ de forma reactiva, esto es aplicando soluciones a errores encontrados en los datos almacenados en las bases de datos, y usados para una determinada aplicación (informe Gartner, 2011). Sin embargo, este enfoque fracasa al resolver problemas organizacionales complejos (Strong, Lee et al. 1997), ya que el concepto de calidad de datos debería no sólo ser entendido como “*cero defectos*” en los datos usados en la aplicación, sino como “*adecuación al uso*” (“*fitness for use*”) del dato para una tarea por un usuario específico, es decir, la habilidad de un conjunto de datos para satisfacer los requisitos de un usuario (Cappiello y Comuzzi 2007). Además, se debe tener en mente la importancia de los usuarios que desempeñan un rol determinado, porque son ellos los que al final determinan el grado de adecuación al uso de los datos que están utilizando (Caballero, Calero et al. 2008), y que la calidad de datos es el principal discriminante de los datos (Eppler 2003).

No obstante, la mayoría de las aplicaciones Web, tal y como están actualmente implementadas no consideran mecanismos que permitan gestionar la calidad de los datos (Bizer y Cyganiak 2009). Para poder implementar estos mecanismos, los desarrolladores de aplicaciones Web deben conocer y capturar

eficientemente cómo los usuarios entienden el concepto de calidad de datos (Collins 2001; Moraga, Calero et al. 2009). Y para ello, los desarrolladores necesitarían mecanismos que permitieran representar ese conocimiento e incluirlo en el desarrollo de aplicaciones Web como si de otros requisitos software se tratase. Desafortunadamente, ninguna de las propuestas existentes se ha preocupado por definir el concepto de *requisito de DQ*, lo cual resultaría de suma importancia al momento de implementar cualquier mecanismo para su gestión durante el desarrollo de una aplicación.

Debido a esta carencia en la definición del concepto de “*Requisito de DQ*”, se decidió acuñar este concepto como “***la especificación de un conjunto de dimensiones de calidad de datos que un conjunto de datos debería cumplir para una tarea específica desempeñada por un determinado usuario***” (Guerra-García, Caballero et al. 2011).

Varios autores (Wang, Reddy et al. 1995; Strong, Lee et al. 1997; Wang 1998; Loshin 2001; Scannapieco, Pernici et al. 2002; Oliveira, Rodrigues et al. 2005; Caballero, Caro et al. 2008) han desarrollado distintas propuestas en el área de calidad de datos enfocadas principalmente a la identificación e inserción de dimensiones de DQ como parte del desarrollo de Sistemas de Información (SI). Sin embargo, y hasta nuestro conocimiento, no existe propuesta alguna en la que se relacione de alguna forma directa la gestión de dimensiones de calidad de datos con el desarrollo de aplicaciones Web (Guerra-García, Caballero et al. 2010).

Desafortunadamente, y a pesar de los beneficios que objetivamente se pueden conseguir, el área de desarrollo de aplicaciones Web no ha tratado como tal los aspectos de gestión calidad de datos, como demuestra el hecho de que no se hayan encontrado en la literatura trabajos con aproximaciones metodológicas ni tecnológicas desarrolladas por investigadores en Ingeniería del Software, ni por los de Bases de Datos, ni siquiera por los de Calidad de Datos (Guerra-García, Caballero et al. 2010). Se puede considerar por tanto, que es un problema abierto que requiere solución, con aproximaciones tanto a nivel *metodológico* como *tecnológico*. Esta solución supone desarrollar un conocimiento que no existe, y al que se piensa que se puede llegar mediante la aplicación de metodologías científicas.

El enfoque de este trabajo de tesis se centra precisamente en introducir requisitos orientados a *gestionar la DQ* en aplicaciones Web. Así es posible afirmar que, dado el hecho de que los SI en general, y aplicaciones Web en particular, son construidos para satisfacer ciertos requisitos de negocio, si pudieran además ser también inicialmente diseñados para satisfacer requisitos específicos de DQ, entonces, podría ser posible *garantizar* tanto a las organizaciones como a los usuarios, que los problemas previamente mencionados puedan ser evitados o al

menos minimizados como se reclama en (Wang 1998; Levis, Helfert et al. 2007).

Así, el objetivo de esta tesis es abordar el estudio de los mecanismos para proporcionar a los desarrolladores de aplicaciones Web un entorno *metodológico* y *tecnológico* que permita complementar los requisitos software con aquellos que puedan identificarse a partir de los requisitos de calidad de datos definidos por el usuario. Como parte del ambiente tecnológico, se ha considerado el estudio del enfoque recientemente surgido como *Ingeniería Web Dirigida por Modelos*.

1.2 Hipótesis y Objetivos.

La principal hipótesis de trabajo para esta tesis es:

“Es posible crear un entorno de trabajo con los mecanismos necesarios (tanto metodológicos como tecnológicos) que permitan a los desarrolladores de aplicaciones Web poder especificar requisitos de calidad de datos y su posterior trazabilidad e implementación”.

Considerando la hipótesis anterior, el objetivo principal de este trabajo de tesis es:

“Definir un ambiente de trabajo que permita la incorporación de requisitos para la gestión de la calidad de datos en aplicaciones Web a través de un entorno de trabajo que incluya tanto aspectos metodológicos como tecnológicos”

Para cubrir este objetivo principal se decidió dividirlo en varios objetivos parciales:

- **O1.** Estudiar el estado del arte de las áreas principales en las que se sustenta esta tesis: Calidad de Datos, Ingeniería Web (Ingeniería Web Dirigida por Modelos) e Ingeniería de Requisitos.
- **O2.** Llevar a cabo un análisis comparativo de las diferentes propuestas publicadas que proponen algún tipo de solución para la gestión de requisitos de DQ en el desarrollo de sistemas de información.
- **O3.** Definir el concepto de *requisito de DQ* y de *requisito Software de Calidad de Datos*, además de la definición de una *metodología para la especificación de requisitos de calidad de datos en aplicaciones Web*. Asimismo, la propuesta de un *modelo para la selección de requisitos de calidad de datos* dependiendo de las funcionalidades a ser

implementadas en aplicaciones Web.

- **O4.** Especificación de un metamodelo y un perfil UML para la especificación y *el modelado* de conceptos de calidad de datos en la etapa de especificación de requisitos.
- **O5.** Especificación de un metamodelo y perfil UML para el modelado de elementos relacionados con calidad de datos, para la etapa de diseño.
- **O6.** Desarrollar las herramientas que permitan especificar y modelar los elementos de calidad de datos propuestos, y aprobar las propuestas anteriores mediante un caso de estudio de validación.

1.3 Marco de Trabajo.

Esta tesis se enmarca principalmente como parte del proyecto **TALES** (HITO-2009-14) y las redes de investigación “**DQNet**” (TIN2008-04951-E/TIN) e “**IQMNet**” (TIN2010-09809-E).

Proyecto TALES

El principal objetivo de este proyecto es la realización de un Entorno de trabajo para el desarrollo de aplicaciones sensibles a los aspectos de Calidad de Datos. Dicho entorno comprende dos elementos principales: uno metodológico y otro tecnológico. Este proyecto fue desarrollado por la Universidad de Castilla-La Mancha, y financiado por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha (HITO-2009-14).

Redes temáticas DQNet (TIN2008-04951-E/TIN) e IQMNet (TIN2010-09809-E)

Estas redes están formadas por catedráticos y profesores pertenecientes a distintas universidades tanto españolas como latinoamericanas, entre ellas: Universidad de Castilla-La Mancha, Universidad de Málaga, Universidad de Sevilla, Universidad de Oviedo, Universidad de Granada, Universidad de Alicante, Universidad del Bio-Bio (Chile), Universidad Politécnica de San Luis Potosí (México). Además de varias empresas como: Alarcos Quality Center, Indra Software Labs, DEYDE, Schober Ibérica y PowerData. Todas estas organizaciones están involucradas en investigación en calidad de datos y existen ya interacciones entre algunas de ellas desde hace varios años.

Los principales objetivos que se pretenden obtener a través de estas redes temáticas, son los siguientes:

- Ayudar a consolidar, unificar y divulgar conocimiento sobre la calidad de los datos en los Sistemas de Información, en el panorama español, enfocándolo desde el punto de vista riguroso, metódico y teórico de las universidades y desde el punto de vista más práctico y pragmático de las empresas.
- Fomentar el desarrollo y uso de nuevas técnicas y metodologías que garanticen la calidad de los datos en los procesos de negocio de las organizaciones, principalmente en aquellas líneas en las que investigan los participantes y que cubren los aspectos más actuales dentro de los Sistemas de Información.
- Conseguir estrechar lazos entre las diversas organizaciones que integran la red temática, así como con otras organizaciones públicas o privadas,

nacionales o internacionales, para colaborar en el desarrollo de publicaciones, proyectos, conferencias, tutoriales, etc.

- Dar a conocer el soporte actual que se está dando a la I+D respecto a la gestión de la calidad de información en entornos empresariales a nivel Internacional.

1.4 Organización de la Tesis.

A continuación se presenta la estructura sintetizada del resto de los capítulos de esta tesis, mencionando el principal contenido en cada uno de ellos:

- **Capítulo 2.** En este capítulo se presentan los aspectos más importantes de las áreas principales consideradas como base en esta tesis. En este capítulo se pretende cubrir el objetivo parcial O1.
- **Capítulo 3.** En este capítulo se describe en profundidad cómo se abordó el objetivo, así como los resultados, tanto metodológicos como tecnológicos que se generaron para satisfacer dichos objetivos. Uno de estos resultados es la propuesta de metodología titulada *“Metodología para la Gestión de Proyectos de Especificación de Requisitos de Calidad de Datos (DQ-VORD)”*. Además, como parte principal de esta metodología, se detalla el *“Modelo propuesto para la selección de Requisitos de DQ según las funcionalidades a implementar en una aplicación Web (MOSCAF)”*. Igualmente, y como parte del enfoque tecnológico, se presenta primeramente el *“Metamodelo ampliado para la especificación de requisitos de DQ en aplicaciones Web (DQ_WebRE)”*. Siguiendo con este enfoque se describe además la propuesta de un *“Metamodelo para el diseño de Aplicaciones Web sensibles a DQ (DAQ_UWE)”*. Ambas propuestas tecnológicas forman parte primordial en algunas de las actividades específicas definidas dentro de la metodología DQ-VORD. Para cada metamodelo propuesto se desarrolló un perfil UML, a fin de hacerlos operativos. En este capítulo se pretende cubrir los objetivos parciales O3, O4 y O5.
- **Capítulo 4.** Aquí se presenta la validación de las propuestas mediante un caso de estudio de validación realizado. En este capítulo se pretende cubrir el objetivo parcial O6.
- **Capítulo 5.** En este capítulo se presentan las conclusiones y principales aportaciones de este trabajo de tesis. Además de mencionar alguna de las líneas de investigación abiertas para trabajo futuro.
- **Anexos.** Se incorpora al final los anexos de esta tesis. El anexo A describe la revisión sistemática realizada de los trabajos relacionados con la especificación de requisitos de calidad de datos, como parte del desarrollo de sistemas de información (SI). En este apartado se pretende cubrir el objetivo parcial O2. El anexo B presenta la descripción de un plugin implementado en la plataforma de desarrollo Eclipse. En el anexo C se muestra la descripción de relaciones entre problemas de un Sistema de Información y funcionalidades Web, como

parte del apartado 3.2.5. Finalmente, el anexo D presenta la lista de acrónimos.

- **Bibliografía.** Referencias bibliográficas mencionadas en la tesis.

2

Estado del Arte

“Sólo vemos lo que conocemos”
– Johann Wolfgang von Goethe (1749-1832)

En este capítulo se describen los conceptos principales de cada una de las áreas consideradas como base para el desarrollo de este trabajo de tesis, los cuales están relacionados principalmente con: Calidad de Datos, Ingeniería Web, Ingeniería Web Dirigida por Modelos e Ingeniería de Requisitos.

2.1 Calidad de Datos

Un sistema de información es tan bueno como los datos que se encuentran dentro de éste (Clement, Ben Hassine Guetari et al. 2010), cada vez más organizaciones están descubriendo esto, una vez que ellas modernizan sus antiguos sistemas heredados en nuevos sistemas de información, por ejemplo, aplicaciones Web o sistemas ERP (*Planificación de Recursos Empresariales*) (Gartner 2009). Las nuevas tecnologías en el mundo actual han jugado un papel muy importante en la complejidad de intercambio de información (Lucas 2010). Las ventajas de Internet y las nuevas tecnologías de diseño de interfaces han dado lugar a que las empresas estén estrechamente relacionadas unas con otras, además de sus clientes. Con la transferencia simultánea de datos (ISO/TS--8000-100 2011) teniendo lugar constantemente, es más importante que nunca para las empresas poder transmitir y mostrar *datos con calidad* (Akoka, Berti-Equille et al. 2007; Bertino, Maurino et al. 2010).

Los datos representan objetos del mundo real, en un formato que puede ser generado, almacenado, recuperado e intercambiado por varios Sistemas de Información (Batini y Scannapieco 2006) para la realización de ciertas operaciones. Debido a la gran diversidad en los datos, éstos pueden ser extremadamente versátiles en su representación (Batini y Scannapieco 2006).

Ya que investigadores en el área de DQ tienen que tratar con un amplio espectro de posibles representaciones de datos, se han propuesto varias clasificaciones para ellos. Varios autores distinguen implícita o explícitamente tres tipos de datos (Batini y Scannapieco 2006):

- ✓ *Estructurados*, cuando cada elemento de dato tiene una estructura fija asociada, las tablas relacionales son el ejemplo más claro de tipo de datos estructurados.
- ✓ *Semiestructurados*, cuando los datos tienen una estructura la cual tiene algún grado de flexibilidad. Este tipo de datos son también “autodescriptivos”. El lenguaje XML (Extensible Markup Language) es comúnmente usado para representar datos semiestructurados.
- ✓ *No estructurados*, cuando los datos son expresados en lenguaje natural sin ninguna estructura específica o algún tipo de dominio definido.

Los problemas en la calidad de los datos se han venido incrementando y son realmente evidentes, particularmente en bases de datos organizacionales. Levis et al. en (Levis, Helfert et al. 2007) mencionan algunos ejemplos de escenarios en los que datos con niveles de calidad inadecuados originan problemas que afectan negativamente a los Sistemas de Información, y por lo tanto, al rendimiento

organizacional. Las causas más comunes de esos niveles inadecuados son una serie de *obstáculos* a lo largo del ciclo de vida de los datos en un Sistema de Información, como los describe Strong en (Strong, Lee et al. 1997).

En este sentido, en (Eppler y Muenzenmayer 2002) se enumeran algunos de los problemas típicos de calidad de datos en aplicaciones Web:

- o Los datos no están actualizados.
- o La navegación y el diseño de la aplicación Web son confusos para los usuarios.
- o Los usuarios no pueden acceder a la aplicación.
- o La aplicación contiene demasiados datos de forma desordenada.
- o Los datos están incompletos.

Todos estos problemas tienen consecuencias negativas para los usuarios que interactúan con este tipo de aplicaciones, perjudicando muchas veces su desempeño dentro de la organización (Caro, Calero et al. 2008).

Debido a lo anterior, las organizaciones tienen que afrontar las consecuencias de estos problemas a distintos niveles dependiendo de su naturaleza (Caballero, Calero et al. 2008):

- ✓ *Técnicos*, por ejemplo, aquellos relacionados con almacenes de datos (Oliveira, Rodrigues et al. 2005).
- ✓ *Organizacionales*, tales como pérdida de clientes (Redman 2000), grandes pérdidas financieras (Loshin 2001; Eppler y Helfert 2004; Cai y Shankaranarayanan 2007) o incluso insatisfacción de trabajadores (Strong, Lee et al. 1997; English 1999).
- ✓ *Legales*, debido a legislaciones nacionales o privadas (como la Ley Orgánica de Protección de Datos, LOPD de 1999).

La calidad de datos es tratada normalmente como un concepto específico, independientemente del contexto en el cual el dato es producido, almacenado o utilizado, es decir, en un enfoque más genérico, los problemas de DQ solo se visualizan como errores en los datos almacenados. Sin embargo, este enfoque fracasa al resolver problemas organizacionales complejos (Strong, Lee et al. 1997), porque el concepto de calidad de datos no debe ser entendido sólo como “*cero defectos*” en los datos, sino como una adecuación al uso de los mismos, entre la que se encuentra esta exigencia.

Cuando las personas piensan acerca de calidad de datos, frecuentemente reducen este concepto a sólo “exactitud” en los datos, sin embargo, la calidad de datos es más que simplemente “datos exactos”. Otras características significativas

(como credibilidad, actualidad, consistencia y completitud entre otras más) son necesarias con el objetivo de caracterizar completamente la calidad de los datos.

En la Figura 1, se muestra un pequeño esbozo de una tabla en una base de datos, donde se observa algunos ejemplos de estas características importantes, las cuales son descritas en más detalle en la Tabla 1.

ID	Equipo	Entrenador	Campeonatos	Año último campeonato	Año de fundación
1	Barcelona	Guardiola	15	2004	1899
2	Real Madrid	Juan Manuel Lillo	31	2008	
3	Almería	Mourinho	0	2000	1989

Este valor no es verdadero afectando la Actualidad

Ausencia de valor es un problema de Completitud

Intercambio de entrenadores afecta la Exactitud

Este valor debería ser nulo, ya que este equipo nunca ha sido campeón, esto afecta la Consistencia

Figura 1. Ejemplo de características de calidad de datos.

Existen varias definiciones de DQ (Batini, Cappiello et al. 2009). No obstante, puede afirmarse que la gran mayoría de los autores coinciden en que ***un dato es de calidad si es válido para el propósito para el que un usuario quiere utilizarlo para una tarea determinada*** (Strong, Lee et al. 1997). Juran (Juran 1988) define el término calidad de datos como “*adecuación al uso*” -*fitness for use*- (Ge y Helfert 2007). Esta perspectiva puede entenderse como: cada uno de los usuarios desempeñando un rol específico dentro de la organización tendrá una percepción diferente de las dimensiones de DQ relacionadas de acuerdo a la función o tarea que realice usando un sistema de información (ver Figura 2).

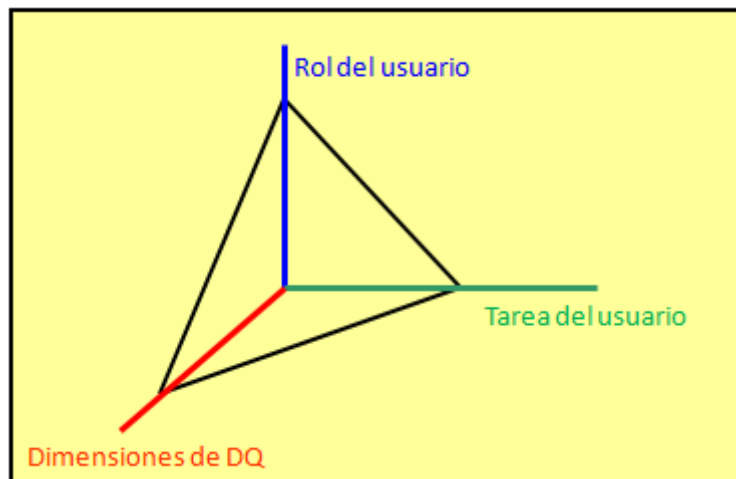


Figura 2. Perspectiva de Calidad de Datos.

La generalidad en la definición “*fitness for use*” (Juran 1988) conlleva dos implicaciones importantes, que a la postre introducirán numerosos aspectos complejos en la gestión de la calidad de datos: por una parte está la *percepción multidimensional* (es necesario evaluar la calidad de los datos usando diferentes características o criterios –denominados *dimensiones de DQ*–), y por otra parte, es preciso tener en cuenta el *carácter subjetivo* de la calidad de datos (cada uno de los usuarios puede tener una visión diferente de cuándo un dato tiene calidad), ya que las mismas acciones no se pueden aplicar a todos los tipos de usuarios (Weber, Boris et al. 2009). Si los usuarios evalúan la calidad de los datos como pobre, su conducta puede llegar a ser influenciada por esta evaluación (Pipino, Lee et al. 2002).

El modelo de calidad propuesto por el estándar internacional ISO/IEC 25012 (ISO-25012 2008) enfocado a Sistemas de Información, identifica quince características de DQ desde dos perspectivas: *inherente* y *dependiente del sistema*.

- ✓ *Inherente*. La calidad de datos inherente se refiere al grado en el cual las características de calidad del dato tienen el potencial intrínseco para satisfacer las necesidades implicadas cuando el dato es usado bajo condiciones específicas.
- ✓ *Dependiente del sistema*. Se refiere al grado en el cual el nivel de calidad de un dato es enriquecida y preservada dentro de un sistema de cómputo cuando el dato es usado bajo condiciones específicas.

Merece la pena resaltar que el estándar utiliza el término “característica” para referirse a lo que tradicionalmente se viene llamando “*dimensiones de calidad de datos*” en la comunidad de investigadores. En la Tabla 1 se describen cada una de las características de DQ propuestas en el estándar ISO 25012 (ISO-25012 2008).

Dimensión	Descripción
Inherentes	
Exactitud	El grado en el cual el dato tiene atributos que representan correctamente el valor verdadero del atributo pretendido de un concepto o evento en un contexto específico de empleo.
Complejidad	El grado al cual el dato del sujeto asociado con una entidad tiene valores para todos los atributos esperados e instancias de entidad relacionadas en un contexto específico de uso.
Consistencia	El grado en el cual el dato tiene los atributos que son libres de contradicción y son coherentes con otros datos en un contexto específico de uso.
Credibilidad	El grado en el cual el dato tiene atributos que son considerados como verdaderos y creíbles por usuarios en un contexto específico de uso.
Actualidad	El grado en el cual el dato tiene los atributos que son del período correcto en un contexto específico de uso.
Inherentes y Dependientes del sistema	
Accesibilidad	El grado en el cual se puede acceder al repositorio donde se encuentra el dato en un contexto específico de uso, en particular por la gente que necesita el soporte de tecnología o una configuración especial debido a alguna inhabilidad (incapacidad).
Conformidad	El grado en el cual el dato tiene atributos que se adhieren a normas, convenciones o regulaciones vigentes y reglas similares relacionadas con la calidad de datos en un contexto específico de uso.
Confidencialidad	El grado en el cual el dato tiene los atributos que aseguran que éste es sólo accesible e interpretable por usuarios autorizados en un contexto específico de uso.
Eficiencia	El grado en el cual el dato tiene los atributos que pueden ser procesados y proporciona los niveles esperados de funcionamiento (desempeño) usando las cantidades y los tipos de recursos apropiados en un contexto específico de uso.
Precisión	El grado en el cual el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso.
Trazabilidad	El grado en el cual el dato tiene atributos que proporcionan un rastro de auditoría de acceso a los datos y de cualquier cambio hecho a los datos en un contexto específico de uso.
Entendibilidad	El grado en el cual el dato tiene atributos que le permiten ser leído e interpretado por usuarios, y es expresado en lenguajes apropiados, símbolos y unidades en un contexto específico de uso.
Dependientes del Sistema	
Disponibilidad	El grado en el cual el dato tiene atributos que le permiten ser recuperados por usuarios autorizados y/o aplicaciones en un contexto específico de uso.
Portabilidad	El grado en el cual el dato tiene los atributos que le permiten ser instalado, substituido o movido de un sistema a otro conservando la calidad existente en un contexto específico de uso.
Recuperabilidad	El grado en el cual el dato tiene atributos que le permiten mantener y conservar un nivel especificado de operaciones y calidad, aún en caso de falla, en un contexto específico de uso.

Tabla 1. Características de DQ según el estándar ISO 25012 (ISO-25012 2008).

Con el objetivo de minimizar el impacto negativo de los problemas ya mencionados (técnicos, organizacionales y legales), es esencial que las empresas puedan evaluar si los niveles de calidad de los datos almacenados en sus sistemas de información son adecuados para las tareas que manejan. Esto primeramente involucra la definición de medidas de DQ sobre los datos, y en segundo lugar el establecimiento de algunos rangos de aceptación válidos para los valores obtenidos mediante un procedimiento de medición (Caballero, Calero et al. 2008). En este contexto, la aplicación de técnicas y herramientas para la gestión de la calidad, hará posible de una forma más eficiente la localización de los datos con niveles

inadecuados de DQ (Wang 1998).

Un componente crucial de la calidad de los datos es que el dato tiene que ser correcto y accesible entre los sistemas. Si los datos no son correctos, las organizaciones se enfrentan a errores, duplicados e inconsistencias, lo que resulta en costes adicionales, re-trabajo y pérdida de tiempo, además de una pérdida de credibilidad y fiabilidad del sistema. Por estas razones y más, diversas organizaciones y empresas siguen gastando grandes cantidades de dinero en iniciativas reactivas de calidad de datos (Eppler y Helfert 2004; Shankaranayanan y Cai 2005; Cai y Shankaranarayanan 2007).

En la literatura se pueden encontrar varias propuestas de técnicas para evaluar y mejorar la calidad de los datos, tales como: *registro de acoplamiento (record linkage)*, *reglas de negocios* y *medidas de similitud*. Con el tiempo, estas técnicas han venido evolucionando haciendo frente a nuevos niveles de complejidad cada vez mayores de DQ (Batini, Cappiello et al. 2009).

Debido a la diversidad y complejidad de estas técnicas, investigaciones recientes se han enfocado a la definición de nuevas metodologías que ayuden a seleccionar, personalizar y aplicar técnicas de evaluación y mejora de calidad de datos en cualquier tipo de sistema de información, entre ellos las aplicaciones Web. Un estudio a profundidad de estas metodologías puede ser visto en (Batini, Cappiello et al. 2009).

Las organizaciones, como se mencionó antes, frecuentemente tienen que tratar con percepciones subjetivas por parte de los individuos que utilizan los datos, así como con mediciones objetivas basadas sobre un conjunto de datos. Un usuario desempeñando un rol puede observar para un conjunto de datos, tantas dimensiones de DQ como sean necesarias en un determinado escenario usando un sistema de información. Por lo que la percepción del nivel de DQ de un conjunto de datos puede llegar a variar para diferentes tareas, aún para el mismo usuario desempeñando distintos roles.

Debido a lo anterior y considerando que la DQ es reconocida como un concepto multidimensional, resulta necesario identificar distintas características de calidad de los datos, y en base a esto definir y especificar *requisitos específicos de DQ*.

Dichos requisitos de DQ deberían ser considerados para su implementación desde la etapa inicial del desarrollo de cualquier tipo de aplicación, de tal forma que las aplicaciones tomarán en cuenta este tipo de requisito específico (además de los requisitos funcionales y no funcionales normalmente considerados), como parte de todo su ciclo de desarrollo.

2.2 Ingeniería Web

Internet ha influido en la vida de millones de personas en varios sentidos (profesional y social), madurando al punto de convertirse en una plataforma tecnológica dominante y atractiva para la implementación de negocios, aplicaciones sociales y sistemas de información organizacional (Murugesan 2008; Bertino, Maurino et al. 2010).

Las aplicaciones Web, como uno de los servicios proporcionados a través de Internet, se han convertido en la base para la realización de negocios empresariales, a través de la implementación de una gran variedad de tipos de sistemas de información, bases de datos y sistemas legados. El número de aplicaciones enfocadas para la gestión de contenido en aplicaciones Web ha venido creciendo en los últimos años, basándose principalmente en las necesidades y la experiencia de los distintos tipos de usuarios (Powers, Brown et al. 2011). Además, estas aplicaciones suelen soportar la gestión de documentos, trabajo cooperativo, conocimiento distribuido y la compartición de medios como audio, video, fotos, etc. (Lew, Olsina et al. 2010).

Las aplicaciones Web tienen ciertas características intrínsecas únicas que hacen al desarrollo Web diferente y tal vez más difícil, comparado con el desarrollo de software tradicional (aplicaciones de escritorio). Algunas de las características clave de este tipo de aplicaciones son (Murugesan, Ginige et al. 2005):

- ✓ La mayoría de las aplicaciones Web son evolutivas por naturaleza, requiriendo cambios frecuentes de contenido, funcionalidades, estructura, navegación, presentación e implementación.
- ✓ Deben ser utilizados por una amplia y diversa comunidad remota de usuarios, los cuales tienen diferentes requisitos, expectativas y habilidades. Por lo tanto, la interfaz de usuario y las características de usabilidad tienen que satisfacer un amplio abanico de las necesidades de una diversa y anónima comunidad de usuarios.
- ✓ Demandan una presentación de una gran variedad de contenido multimedia: texto, gráficos, imágenes, audio y video, además de que el contenido debería ser integrado con un procesamiento procedimental. Por lo tanto, el desarrollo de estas aplicaciones incluye la creación y gestión de todo este tipo de contenidos y su presentación de una forma atractiva.
- ✓ En general, demanda una buena apariencia *"look and feel"*, y una navegación fácil.
- ✓ Están especialmente destinadas para audiencias globales, lo que significa que se deben adherir a variados sentimientos sociales y culturales, y

estándares nacionales, incluyendo múltiples lenguajes y diferentes sistemas de unidades.

- ✓ Las necesidades de privacidad y seguridad son en general más exigentes que aquellos de un software tradicional.
- ✓ Necesitan hacer frente con una gran variedad de dispositivos y formatos de visualización, además del soporte de hardware, software y redes con una vasta variedad de velocidades de acceso.
- ✓ Las consecuencias por fallas o insatisfacción de usuarios pueden ser mucho peores que en sistemas de información convencionales.
- ✓ La proliferación de nuevas tecnologías Web y estándares, además de la presión competitiva trae consigo sus ventajas, y también retos adicionales para desarrollar y mantener aplicaciones Web.
- ✓ La naturaleza cambiante de estas aplicaciones requiere un proceso de desarrollo incremental.

Los usuarios de las aplicaciones Web esperan que éstas sean útiles, confiables y seguras, además de personalizables y sobre todo sensibles al contexto. Debido a esto, características como el desempeño, fiabilidad, calidad, mantenibilidad y escalabilidad en estas aplicaciones se han convertido en los principales retos. Como resultado de este incremento de complejidad, actividades como el diseño, desarrollo, implementación y mantenimiento de estas aplicaciones Web se han convertido inherentemente complejas y desafiantes.

Por supuesto el desarrollo de aplicaciones Web no es sólo diseño visual ni interfaz de usuario. Esto involucra planificación, selección de una arquitectura Web o un framework de soporte adecuado a las necesidades del usuario, diseño del sistema, diseño de páginas, codificación, creación de contenido, así como su mantenimiento, pruebas, aseguramiento de la calidad y evaluación del desempeño. Por lo anterior, los desarrolladores Web necesitan adoptar un proceso de desarrollo disciplinado, metodologías de diseño y utilizar las mejores herramientas de desarrollo (Murugesan 2008).

Considerando la complejidad determinada por las características y necesidades anteriores, se hace patente la necesidad de una *Ingeniería Web* (IW) de acuerdo a las percepciones de los desarrolladores y administradores, sus experiencias en la creación de aplicaciones Web y la complejidad de las mismas (Deshpande, Murugesan et al. 2002).

En los últimos años, se ha generado una considerable cantidad de investigación dentro del campo del modelado funcional de aplicaciones Web. Este hecho ha causado una nueva tendencia en investigación dentro de la Ingeniería de Software conocida como *Ingeniería Web* (Meliá y Gómez 2005). La Ingeniería Web es la aplicación cuantificable, sistemática y estructurada de propuestas

metodológicas para el desarrollo, evaluación y mantenimiento de aplicaciones Web (Deshpande, Murugesan et al. 2002).

La disciplina de Ingeniería Web aboga por un enfoque integral y ordenado para desarrollos Web exitosos, tomando en cuenta las características únicas y los requisitos de sistemas basados en Web. La Ingeniería Web *“utiliza principios científicos, de ingeniería y administración, además de enfoques sistemáticos para desarrollar, implementar, y mantener sistemas y aplicaciones Web de alta calidad”* (Murugesan, Deshpande et al. 2001). Por tanto, puede decirse que la esencia de la Ingeniería Web es gestionar con éxito la diversidad y complejidad del desarrollo de aplicaciones Web y, por lo tanto, evitar fallos potenciales que pudieran generar serias implicaciones.

Los principios básicos de la Ingeniería Web pueden ser descritos similarmente a aquellos de la Ingeniería de Software como:

- ✓ Metas y requisitos definidos claramente.
- ✓ Desarrollo sistemático en fases de una aplicación Web.
- ✓ Una planificación cuidadosa de esas fases.
- ✓ Auditoría continua del proceso de desarrollo completo.

La IW hace posible planear e iterar procesos de desarrollo y así también facilita la evolución continua de aplicaciones Web. Esto permite no sólo la reducción de costos y minimización de riesgos durante el desarrollo y mantenimiento, sino también un incremento en la calidad, así como la medición de la calidad de los resultados de cada fase (Ginige y Murugesan 2001; Mendes y Mosley 2006).

Una aplicación basada en Web es un *“sistema vivo”* crece, evoluciona y cambia. Una infraestructura apropiada es necesaria para soportar el crecimiento de una aplicación Web de una forma controlada y flexible. La IW ayuda a crear una infraestructura que permitirá la evolución y el mantenimiento de este tipo de aplicaciones, además de dar un soporte a la creatividad (Murugesan 2008).

El desarrollo Web es un proceso, no es un evento que sucede una sola vez. La Ingeniería Web trata con todos los aspectos del desarrollo de sistemas Web, iniciando desde la concepción y desarrollo, hasta la implementación, desempeño, evaluación y mantenimiento continuo. La IW, por lo tanto, cubre un rango de áreas: elicitación y análisis de requisitos, modelado de sistemas Web, arquitectura Web, diseño de sistemas Web, diseño de páginas, codificación (scripting), interfaz con bases de datos, sistemas ERP (Planificación de Recursos Empresariales) y otros sistemas Web; calidad, usabilidad y seguridad Web, evaluación del desempeño en Web, pruebas Web, metodologías y procesos de desarrollo Web, métricas Web y gestión de proyectos Web.

La naturaleza y características de las aplicaciones Web así como su desarrollo demanda que la Ingeniería Web sea un área multidisciplinaria, que abarca entradas de diversas áreas como: análisis y diseño de sistemas, ingeniería de software, ingeniería de hipermedia/hipertexto, ingeniería de requisitos, interacción humano-computador, interfaz de usuario, ingeniería de información, indexado y recuperación de información, pruebas, modelado y simulación, gestión de proyectos y diseño y presentación gráfica (Deshpande, Murugesan et al. 2002).

Murugesan señala que una aplicación Web bien diseñada debe ser (Murugesan, Ginige et al. 2005):

- ✓ Funcionalmente completa y correcta.
- ✓ Utilizable.
- ✓ Robusta y confiable.
- ✓ Mantenible.
- ✓ Segura.
- ✓ Se desempeña razonablemente bien incluso en picos de cargas.
- ✓ Escalable.
- ✓ Portable.
- ✓ Reusable.
- ✓ Interoperable con otros sistemas.
- ✓ Tiene accesibilidad universal.
- ✓ Bien documentada.

Desde su origen y crecimiento como una nueva disciplina en 1998, la IW ha sido objeto de un interés cada vez mayor entre investigadores, desarrolladores, académicos y usuarios.

2.2.1 Ingeniería Web Dirigida por Modelos

Tradicionalmente, la mayoría de las propuestas de IW han considerado diferentes modelos (contenido, navegación y presentación) como una forma relevante de orientar el diseño de aplicaciones Web. Sin embargo, debido a los nuevos requisitos y tecnologías de implementación en el dominio Web, los enfoques de IW tienen que evolucionar (Moreno, Meliá et al. 2008). El enfoque dirigido por modelos "*model-driven*" da la pauta para esta evolución debido a las ventajas que proporciona durante el proceso de desarrollo, implicando (Moreno, Meliá et al. 2008):

- ✓ La definición de nuevos modelos y elementos de modelado que capturen requisitos y características adicionales.

- ✓ La re-definición del metamodelo para manejar estas características adicionales.
- ✓ La adaptación del proceso de desarrollo para incorporar las nuevas características y la información que éstas representan.
- ✓ La adaptación del proceso de modelado y las herramientas de generación de código que soportan el método.

El paradigma de *Ingeniería Dirigida por Modelos* (MDE) es una solución que ha venido siendo aplicada en el desarrollo Web, obteniendo muy buenos resultados (Moreno y Vallecillo 2008). La aplicación de MDE a la Ingeniería Web es una disciplina llamada *Ingeniería Web Dirigida por Modelos* (MDWE). MDWE propone la representación de conceptos usando metamodelos, soportando el proceso de desarrollo por un conjunto de transformaciones y relaciones entre conceptos, permitiendo desarrollos ágiles y asegurando la consistencia entre los modelos (Escalona y Aragón 2008).

La gran mayoría de las investigaciones en este campo cubren, dentro de las fases del desarrollo, principalmente las fases de análisis y diseño de las aplicaciones Web. En este contexto, diferentes lenguajes, métodos, herramientas y patrones de diseño para el modelado Web han sido propuestos. Estas investigaciones proponen procesos específicos para soportar el desarrollo sistemático o semiautomático de estas aplicaciones; sin embargo, muy pocas de las metodologías existentes inician el ciclo de desarrollo con un análisis de requisitos detallado (Escalona y Koch 2004).

Las propuestas dentro del área de MDWE se pueden clasificar básicamente en dos enfoques principales (Moreno, Romero et al. 2008): primero, unas pocas propuestas están basadas en métodos de diseño de hipermedia, introduciendo los mecanismos y la expresividad requerida para capturar los elementos relevantes específicos Web, por ejemplo la navegación. Mientras que el otro grupo de enfoques más recientes surgido como extensiones de técnicas de desarrollo orientado a objetos, se han adaptado a hacer frente a las características particulares de aplicaciones Web.

Algunas de las principales propuestas dentro del área de MDWE son: **OOWS** (Fons, Pelechano et al. 2003), **NDT** (Escalona y Aragón 2008), **UWE** (Koch y Kraus 2002), **WebML** (Ceri, Fraternali et al. 2000), **WebRE** (Escalona y Koch 2006), **W2000** (Baresi, Garzotto et al. 2002), **WSDM** (De Troyer y Leune 1998), **SOD-M** (De Castro y Marcos 2009) y **WebSA** (Meliá y Gómez 2005). Un estudio comparativo de estas propuestas se muestra en (Escalona y Koch 2004), enfocado principalmente en analizar los tipos de requisitos que gestionan, las técnicas empleadas y el grado de detalle de cada propuesta en términos de su proceso de desarrollo. Enseguida se describen brevemente las características más importantes de cada propuesta.

Fons en (Fons, Pelechano et al. 2003) propone OOWS (Object-Oriented Web Solutions) como una extensión del método de producción de software orientado a objetos “OO-Method” (Pastor, Gómez et al. 2001). OOWS proporciona un método completo que define un conjunto de actividades para especificar de forma adecuada características de tipo funcional, navegacional y presentacional de aplicaciones Web. Este método comprende dos pasos principales: especificación del sistema y desarrollo de la solución. El primer paso está relacionado a la especificación de los requisitos funcionales del sistema. El segundo paso propone una estrategia orientada hacia la generación de componentes software que constituyen la solución (el producto software final).

En (Escalona y Aragón 2008) se presenta una metodología que cubre las fases de requisitos y análisis en desarrollos Web. Esta metodología está basada completamente en el enfoque MDWE, proponiendo metamodelos NDT (*Navigational Development Techniques*) y transformaciones entre ellos. NDT es un enfoque metodológico el cual trata con los requisitos de ambientes Web, fue propuesto con el objetivo de soportar la ingeniería de requisitos y la fase de análisis. Ofrece un ambiente adecuado para la captura, definición, análisis y validación de requisitos Web.

Koch y Kraus en (Koch y Kraus 2002), cubren los aspectos especiales en el diseño de aplicaciones Web (UWE), definen un conjunto de vistas especializadas representadas gráficamente por diagramas UML, tales como modelos de navegación y modelos de presentación, mediante el uso de perfiles de UML. Además muestran cómo utilizar distintos tipos de diagramas estáticos para modelar los aspectos estáticos de las aplicaciones Web, igualmente el uso de diagramas dinámicos para el modelado dinámico, por ejemplo, diagramas de actividad para el modelado de tareas.

La propuesta WebML (Ceri, Fraternali et al. 2000), es un lenguaje de especificación de alto nivel para el diseño de aplicaciones Web intensivas de datos, enfatizando la definición de primitivas de composición y navegación ortogonales. El diseñador puede utilizar dichas primitivas para modelar requisitos complejos. WebML incluye una noción explícita de “vista de sitio”, con el que la misma información puede ser estructurada en diferentes sentidos para cumplir con los intereses y necesidades de diferentes grupos de usuarios. Finalmente contempla también aspectos avanzados de modelado de sitios Web, incluyendo presentación, modelado de usuarios y personalización.

En (Escalona y Koch 2006), los autores se centran en la especificación a través de modelos de requisitos de sistemas Web. Presentan un metamodelo, *WebRE*, que contiene los conceptos clave necesarios para la especificación de

requisitos específicos tales como:

- o casos de uso de navegación y casos de uso de procesos Web;
- o actividades específicas tales como: navegación, búsqueda y transacciones de usuario; y
- o elementos estructurales como nodos, contenido e interfaces de usuario.

Junto con el metamodelo, también se hacen dos aportaciones principales: i) un perfil UML para la especificación de requisitos Web, con los aspectos clave necesarios para el modelado, y, ii) los elementos del metamodelo pueden ser mapeados para la construcción de modelos de diferentes metodologías Web.

La propuesta W2000 de Baresi et al. (Baresi, Garzotto et al. 2002), enriquece la última versión de HDM (*Hypertext Design Model*) con conceptos y notaciones para modelar estructuras de información más ricas, así como también operaciones y servicios accesibles a través de la Web. Esta propuesta considera las ventajas del lenguaje UML y sus características de personalización para atribuir a la propuesta W2000 como una sintaxis gráfica estándar. Además, los autores de esta propuesta desarrollaron una herramienta CASE que asiste al diseñador de aplicaciones Web durante el proceso de desarrollo desde el diseño conceptual hasta la generación semiautomática de la aplicación final.

En la propuesta WSDM (*Web Site Design Method*) (De Troyer y Leune 1998), la gestión de requisitos es llevada a cabo usando técnicas tales como mapeo de conceptos y diccionarios de datos para la definición de requisitos específicos tanto funcionales como de seguridad. Este enfoque considera la especificación mediante una forma textual, por lo que puede producir algunos errores de precisión.

La propuesta de SOD-M (De Castro y Marcos 2009) presenta un enfoque orientado a servicios para el desarrollo de sistemas de información. Inicia mediante el modelado de negocios para identificar los servicios requeridos por los clientes de los negocios, haciendo posible la creación de modelos de composición de servicios Web. Esta propuesta utiliza UML como lenguaje de modelado y define un perfil UML específico para el desarrollo orientado a servicios. SOD-M se enfoca en el desarrollo de aspectos conductuales y define guías con las cuales construir modelos conductuales a partir del modelado de negocios de alto nivel.

En (Meliá y Gómez 2005) se propone un enfoque genérico llamado *WebSA*, basado en el paradigma MDA (Model Driven Architecture, (OMG 2003)). Se propone un desarrollo dirigido por modelos de un conjunto de modelos arquitectónicos de UML y transformaciones QVT (*Query/Views/Transformations*, (OMG 2005)) como mecanismos para integrar aspectos funcionales de las metodologías actuales con los aspectos arquitectónicos. Básicamente, *WebSA* proporciona al diseñador un conjunto de modelos arquitectónicos y modelos de transformación para especificar

aplicaciones Web. Usando estos modelos el diseñador puede integrar los modelos funcionales Web (de dominio, de navegación y de presentación) aplicando un conjunto de transformaciones de modelos mediante la solicitud de propuestas usando el lenguaje QVT. El resultado es un modelo de integración que captura los aspectos funcionales y arquitectónicos de una aplicación Web.

Todas estas propuestas anteriores son dirigidas por modelos (*model-driven*), debido a que dirigen los diferentes *intereses* involucrados en el diseño y desarrollo de una aplicación Web usando modelos separados (tales como, contenido, navegación y presentación). Existen diversos compiladores de modelos que producen la mayoría de páginas y lógica de la aplicación Web a partir de modelos originales. Además, la mayoría de las propuestas cuentan con procesos de desarrollo que soportan sus notaciones y herramientas. Estas propuestas han sido utilizadas satisfactoriamente en ambientes comerciales para la construcción de muchos diferentes tipos de aplicaciones Web (Moreno, Romero et al. 2008).

Las aplicaciones Web actuales necesitan inter-operar con otros sistemas externos. Esto requiere su integración, por ejemplo, con proveedores de servicios Web, portales, y sistemas heredados (Becker, Jaster et al. 2009). Lo que conduce a entre otras cosas, que sus procesos, interfaces y parte de su lógica de negocios tenga que estar explícitamente disponible para la integración con estos sistemas externos (Moreno y Vallecillo 2005). Sin embargo, no todas las propuestas MDWE gestionan este aspecto a nivel de modelo, la integración es más bien realizada a nivel de implementación.

Cabe destacar que todas las propuestas presentadas en esta sección, están especialmente enfocadas a la definición de los aspectos funcionales, relacionados principalmente a los modelos de navegación, conceptual y de presentación. Sin embargo, ninguna de estas propuestas trata con aspectos relacionados con la calidad de los datos que serán gestionados y almacenados en las aplicaciones Web. Solo la propuesta de Ceri et al. en (Ceri, Fraternali et al. 2000) menciona algunos objetivos específicos de datos que deberían ser considerados al momento de diseñar e implementar aplicaciones Web. Sin embargo los autores no profundizan en su estudio ni consideran algún tipo de especificación de características de calidad de datos.

2.3 Ingeniería de Requisitos

La Ingeniería de Requisitos es el proceso sistemático para desarrollar requisitos a través de un proceso iterativo y cooperativo de análisis del problema, documentando las observaciones resultantes en una variedad de formatos de representación, y verificando la precisión del entendimiento ganado (Loucopoulos y Karakostas 1995).

El enfoque orientado a puntos de vista (*Viewpoints*) para la Ingeniería de Requisitos, toma en consideración los diferentes puntos de vista de los diferentes roles y los usa para estructurar y organizar el proceso de elicitación de requisitos.

El punto clave del análisis orientado a puntos de vista es que toma en cuenta la existencia de varias perspectivas y provee un marco de trabajo para descubrir conflictos en los requisitos propuestos por diferentes puntos de vista.

El punto de vista puede ser utilizado como una forma para clasificar los “*stakeholders*” (desarrolladores, clientes y usuarios interesados en el sistema). Existen tres tipos genéricos de puntos de vista (Sommerville 2000):

1. Puntos de vista directos: representan a personas y otros sistemas que interactúan con el sistema a desarrollarse directamente. Este tipo de punto de vista puede dar requisitos detallados del sistema, tales como características, funcionalidades e interfaces.
2. Puntos de vista indirectos: representan aquellos stakeholders que tienen interés en algunos o todos los servicios entregados por el sistema, pero no interactúan directamente con éste. Es más probable que ellos sean capaces de proveer requisitos y restricciones de tipo organizacional de alto nivel.
3. Puntos de vista del dominio: representan las características y restricciones del dominio que tienen influencia en los requisitos del sistema.

Como ilustración de los diferentes tipos de requisitos, una jerarquía de puntos de vista se muestra en la Figura 3. Este es un diagrama relativamente simple de los puntos de vista que pueden ser consultados en el proceso de elicitación de requisitos para una aplicación Web de una librería.

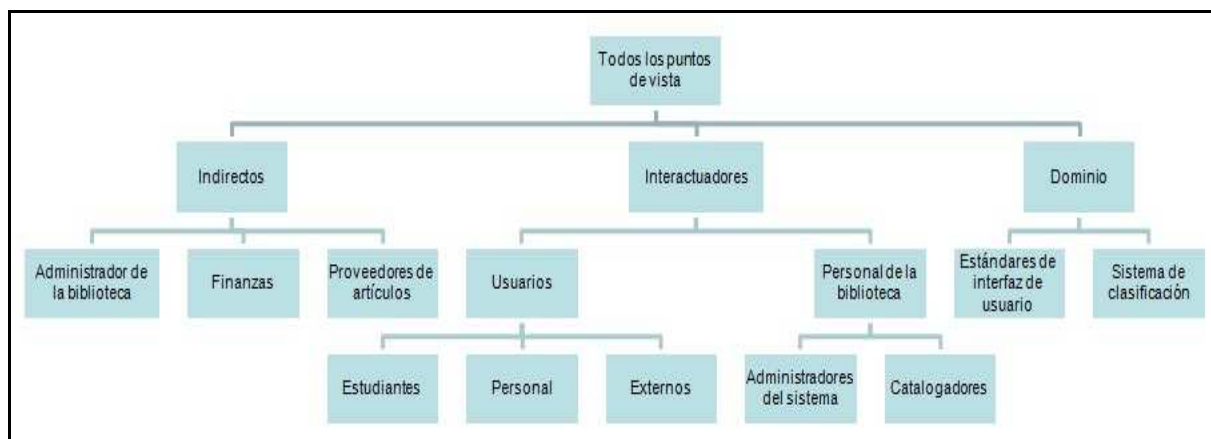


Figura 3. Ejemplo de puntos de vista para una aplicación Web de una librería.

Llevar a cabo una identificación inicial de los puntos de vista relevantes de un sistema puede ser algunas veces difícil. Por lo que los analistas deberían tratar de identificar diferentes tipos de puntos de vista de una forma más sencilla (Sommerville 2000). La siguiente lista pudiera ser usada como una guía para la identificación de puntos de vista generadores de requisitos específicos:

1. Los proveedores de funcionalidades al sistema y los receptores de las funcionalidades del sistema.
2. Los sistemas que tienen que interactuar directamente con el sistema que está siendo especificado.
3. Los estándares y regulaciones que pueden ser aplicados al sistema.
4. Los orígenes de los requisitos no funcionales y de negocios del sistema.
5. Los puntos de vista de ingeniería que reflejan los requisitos de las personas que tienen que desarrollar, administrar o mantener el sistema.
6. Los puntos de vista de mercadotecnia que generan requisitos sobre las características del producto que el cliente espera, y como el sistema debería reflejar la imagen externa de la organización.

Los puntos de vista de ingeniería (aquellos relacionados con la implementación y el diseño del sistema) pueden ser importantes por dos razones: (i) los ingenieros que participan en el desarrollo de aplicaciones Web pueden tener experiencia con aplicaciones similares, por lo que ellos pueden ser capaces de sugerir requisitos a partir de su experiencia, (ii) el personal técnico responsable de la administración y mantenimiento del sistema puede agregar nuevos requisitos que ayuden a simplificar el soporte de éste.

Finalmente, los puntos de vista que proveen algún tipo especial de requisito que puede venir de los departamentos de mercadotecnia o algún otro en la organización. Esto es especialmente cierto para aplicaciones Web, particularmente aplicaciones de comercio electrónico; las cuales tienen que mostrar una imagen favorable de la organización además de proporcionar las funcionalidades al usuario.

Para la mayoría de las aplicaciones una enorme cantidad de posibles puntos de vista pueden existir, por lo que pudiera ser prácticamente imposible obtener una lista completa de requisitos de todos ellos. En consecuencia, es importante organizar y estructurar los puntos de vista jerárquicamente, debido a que probablemente, diferentes puntos de vista en la misma rama puedan compartir requisitos comunes.

El método VORD (*Viewpoints Oriented Requirements Definition*) ha sido diseñado para guiar el proceso de elicitación y análisis de requisitos teniendo en cuenta diferentes puntos de vista (Kotonya y Sommerville 1996). Las etapas principales de este método se muestran en la Tabla 2.

Etapas	Descripción
IP-1. Identificación de puntos de vista.	Implica descubrir a aquellos que reciben las funcionalidades o servicios de la aplicación, e identificar las funcionalidades específicas que proveerá a cada punto de vista.
EP-2. Estructuración de puntos de vista.	Comprende en clasificar los puntos de vista previamente obtenidos en una jerarquía. Las funcionalidades comunes son localizadas en los niveles altos de la jerarquía y se heredan a los puntos de vista de bajo nivel.
DP-3. Documentación de puntos de vista.	Consiste en documentar el nombre de los puntos de vista, requisitos, restricciones sobre sus requisitos y sus recursos.
TP-4. Trazado de los puntos de vista.	Comprende identificar los objetos en un diseño orientado a objetos utilizando la información de la funcionalidad encapsulada en el punto de vista.

Tabla 2. Etapas principales del método VORD.

3 Entorno de trabajo para la Gestión de Requisitos de Calidad de Datos en Aplicaciones Web

“Mientras los hombres sean libres para preguntar lo que deben; libres para decir lo que piensan; libres para pensar lo que quieran; la libertad nunca se perderá y la ciencia nunca retrocederá”
– Julius Robert Oppenheimer (1904-1967)

En este capítulo se describe en profundidad los resultados obtenidos a partir de la investigación en forma de propuestas de trabajo (ver Figura 4) para satisfacer los objetivos presentados en la sección 2. Estas propuestas se encuentran enmarcadas como parte de un *Entorno de trabajo para la Gestión de Requisitos de Calidad de Datos en Aplicaciones Web*. Iniciando con la definición del concepto de **“Requisito de Calidad de Datos”**, y continuando con la propuesta de una *Metodología para la gestión de proyectos de especificación de Requisitos de Calidad de Datos*, como parte del ámbito **metodológico**. Dicha propuesta propone además dos metamodelos, siendo ambos necesarios para dar soporte a la parte del ámbito **tecnológico**. El objetivo es mostrar una aproximación de solución a la problemática planteada, y proporcionar a los desarrolladores de aplicaciones Web las metodologías, herramientas y elementos necesarios para el desarrollo de este tipo de aplicaciones. Involucrando así, los elementos considerados básicos para la *gestión de calidad de datos* durante el proceso de análisis y diseño de aplicaciones Web.

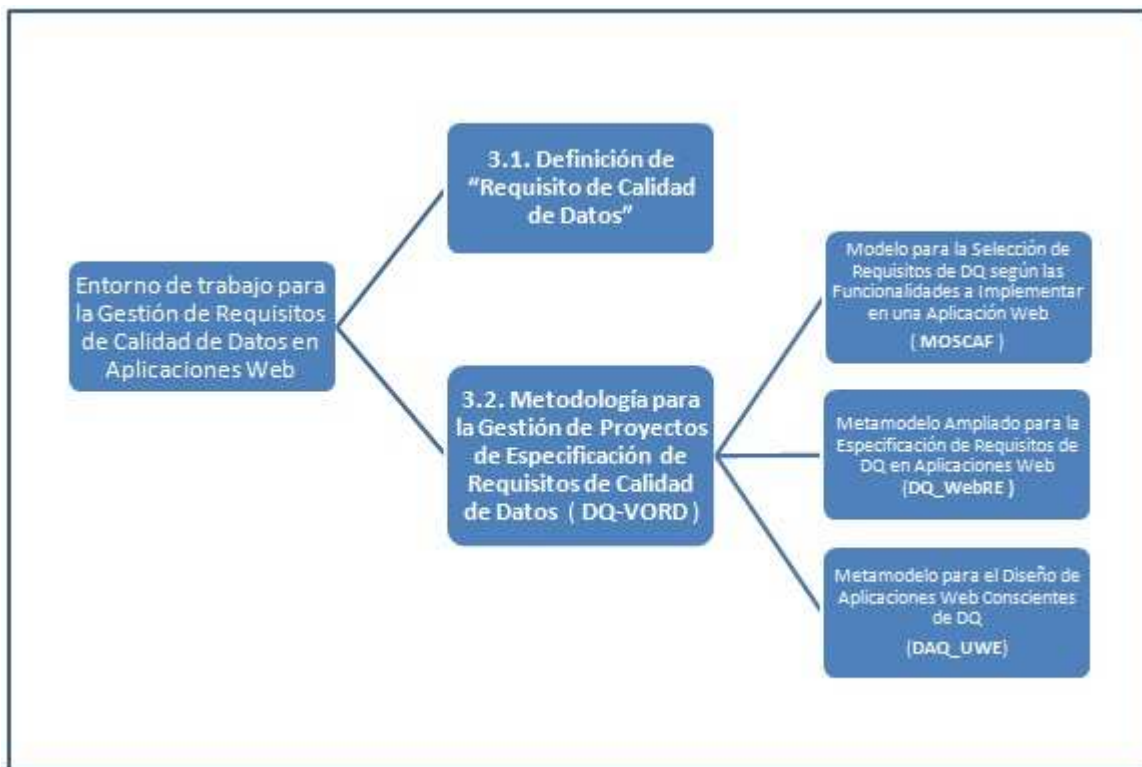


Figura 4. Perspectiva general de las propuestas de este trabajo de Tesis

3.1 Definición del Concepto de Requisito de Calidad de Datos

En los últimos años, el área de la calidad de los datos y de la información, ha empezado a cobrar una importancia cada vez mayor. Datos como los reflejados en (Cappiello, Daniel et al. 2010; ThomsonReuters y Lepus 2010), en donde se muestra que cada vez más organizaciones están invirtiendo grandes cantidades de dinero en iniciativas y proyectos relacionados con calidad de datos, validan su importancia.

Por otra parte, la nueva familia de normas ISO/IEC 25000 Software Product Quality Requirements and Evaluation (SquaRE) (ISO 2005) considera que para que un sistema software tenga globalmente calidad, es necesario, además de la calidad del software, tener en cuenta la *calidad de los datos*. Sin embargo, dada la juventud de esta área específica de calidad de datos, no existen todavía suficientes trabajos significativos que relacionen la calidad de los datos con la calidad del software, y más específicamente algunos que expliquen cómo gestionar *requisitos específicos de calidad de datos*.

Dentro del ámbito de la calidad de datos y hasta donde hemos podido comprobar en las fuentes bibliográficas a nuestro alcance (Guerra-García, Caballero et al. 2010), no existe una definición clara y específica del concepto de “*Requisito de Calidad de Datos*”, por lo que resulta obligatoria buscar una definición consensuada. Cabe mencionar que en (Wang y Madnick 1993), se menciona el término requisito de calidad de datos, sin embargo, éste es utilizado en otro contexto (especificación de “indicadores de DQ” relacionados a datos).

Considerando lo anterior, y como parte de los objetivos parciales de esta tesis (específicamente el objetivo parcial 3), se decidió acotar este concepto. Para ello, se resolvió partir de la definición de requisito software proporcionada por el estándar IEEE 610.12-1990 (IEEE 1990):

- 1) *Una condición o capacidad necesitada por un usuario para resolver un problema o alcanzar un objetivo,*
- 2) *Una condición o capacidad que tiene que ser cumplida o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar o especificación, u otro documento impuesto de manera formal,*
- 3) *Una representación documentada de una condición o capacidad como en 1 ó 2.*

Estas definiciones abarcan tanto el punto de vista del usuario del sistema como el punto de vista de los desarrolladores. Por otra parte, el estándar ISO/IEC 25012 se enfoca sobre la calidad de los datos como parte de un sistema

computacional y define características de calidad para los datos usados por usuarios y otros sistemas software (ISO-25012 2008).

De esta manera, una vez analizado las definiciones anteriores, se propone la siguiente definición de **Requisito de DQ**: ***“la especificación de un conjunto de dimensiones de calidad de datos que un conjunto de datos debería cumplir para una tarea específica desempeñada por un determinado usuario”*** (Guerra-García, Caballero et al. 2011).

El propósito principal al momento de definir un **requisito de DQ**, es por tanto satisfacer las necesidades específicas de calidad en los datos que cada uno de los usuarios requiere en un momento determinado para llevar a cabo sus tareas o funcionalidades específicas con un sistema de información. Esto implica elegir de aquellas dimensiones que mejor se adapten a las recogidas en un modelo de calidad de datos, las que sean de especial interés para el usuario.

Estos *requisitos de calidad de datos (DQR)*, se introducirán normalmente en el proceso de desarrollo, convirtiéndose en **requisitos software de calidad de datos (DQSR)**, los cuales se mezclarán con el resto de requisitos software (funcionales, no funcionales, de seguridad) durante el resto del desarrollo de la aplicación Web.

En la siguiente sección se presenta la aportación principal de este trabajo de tesis, la *Metodología para la Gestión de Proyectos de Especificación de Requisitos de Calidad de Datos (DQ-VORD)*, describiendo en profundidad su objetivo, el catálogo de roles y elementos que involucra, así como sus principales actividades y tareas.

3.2 Metodología para la Gestión de Proyectos de Especificación de Requisitos de Calidad de Datos (DQ-VORD)

Como parte de los objetivos principales (específicamente el objetivo parcial 3) de este trabajo de tesis, se ha elaborado la presente **Metodología para la gestión de proyectos de especificación de requisitos de calidad de datos en aplicaciones Web** (en adelante metodología **DQ-VORD**¹).

3.2.1 Descripción de la Metodología

3.2.1.1 Ámbito de aplicación

La metodología DQ-VORD se ha elaborado para complementar otras metodologías de desarrollo que se usen en proyectos de desarrollo y mantenimiento de aplicaciones Web. De forma concreta, para llevar a cabo el proceso de especificación de requisitos software, implementando características específicas en el sistema de modo que se pueda asegurar la calidad de los datos.

DQ-VORD puede complementar al Proceso Unificado de Desarrollo (Jacobson, Booch et al. 1999), lo ideal es aplicar la metodología desde las primeras fases (especificación, análisis de requisitos y diseño) como un flujo de trabajo más complementario a la captación de requisitos. De esta manera se podrán evolucionar desde las primeras fases del desarrollo los requisitos de DQ (DQR) obtenidos inicialmente, hacia la especificación de requisitos software específicos de DQ (DQSR), pudiendo ser considerados en las fases posteriores de diseño e implementación (ver Figura 5).



Figura 5. Convergencia de los Flujos de trabajo del PUD con la metodología DQ-VORD.

¹ DQ-VORD: Definición de Requisitos Orientados a puntos de Vista de DQ.

3.2.1.2 Objetivo de la Metodología

El principal objetivo de la metodología DQ-VORD es proponer una **guía que sirva a los desarrolladores para la especificación y análisis de requisitos de calidad de datos para aplicaciones Web**. Es una aproximación dirigida por funcionalidades. Para ello habrá que identificar para cada una de las funcionalidades Web a ser implementadas, aquellos requisitos de DQ que tienen que ser asegurados (*tomando como referencia base el modelo **MOSCAF**, explicado posteriormente en la sección 3.2.5*). Es necesario tener en cuenta que cada funcionalidad va a ser utilizada por un tipo de usuario específico que desempeña un rol específico. Por lo que la especificación de estos requisitos de DQ por parte de los distintos tipos de usuarios de la aplicación (a los que se llamará “puntos de vista”) deberán ser adaptados e introducidos en el proceso de desarrollo normal como nuevos requisitos software, teniendo en cuenta las dependencias que se pueden llegar a crear en el modelo de procesos y/o de datos, con otros requisitos de otros puntos de vistas.

En este apartado, se describen las principales características de la metodología DQ-VORD:

- Está formada por un **conjunto estructurado de actividades**.
- Está **orientada a una relación directa con los usuarios y a la obtención de sus requisitos de calidad de datos al momento de implementar las distintas funcionalidades de una aplicación Web**.
- Está pensada para ser una **metodología adaptativa**.
- Está respaldada por un **conjunto de herramientas y técnicas**.

La metodología DQ-VORD está **formada por un conjunto estructurado de actividades**, que fueron desarrolladas tomando como base una de las principales metodologías de elicitación y especificación de requisitos software (VORD), propuesta por Kotonya y Sommerville en (Kotonya y Sommerville 1996). DQ-VORD complementa a los flujos de trabajo en las etapas de captura de requisitos del *Proceso Unificado de Desarrollo (PUD)* (Jacobson, Booch et al. 1999). Así se está proporcionando un marco de trabajo donde se identifica claramente el **qué, cuándo, y el cómo**, de cada una de las actividades y tareas, así como la secuencia de pasos que se deben seguir a la hora de llevar a cabo la metodología para la gestión de proyectos de especificación de requisitos de calidad de datos.

Además, la metodología DQ-VORD está **basada en la relación directa con el usuario** de manera que, en distintos puntos de la metodología, el usuario se encuentra **involucrado en la especificación de sus requisitos de DQ** (en función de los objetivos de la empresa).

Así mismo, la metodología está **pensada para captar todos los requisitos de usuario de cada tipo de usuario** de la aplicación Web, permitiendo modificar el nivel de detalle en el momento de llevar a cabo cada una de las actividades y tareas de la metodología, además de las herramientas y técnicas que se utilizarán.

Finalmente, es interesante afirmar que la metodología DQ-VORD se encuentra **respaldada por un conjunto de técnicas y herramientas de soporte** para agilizar la realización de cada una de las actividades de la metodología, permitiendo especificar y diseñar los distintos requisitos de una manera más sencilla.

Para alcanzar el objetivo principal, la metodología DQ-VORD observa los siguientes objetivos parciales, enumerados a continuación:

- Identificar el catálogo de los roles participantes involucrados en la metodología DQ-VORD, así como las actividades en las que se produce una comunicación entre ellos.
- Identificar los elementos (documentos, informes, catálogos, artefactos, etc.) que se utilizarán a lo largo de las actividades de la metodología.
- Definir el conjunto de actividades necesarias para llevar a cabo la especificación de requisitos software de calidad de datos, así como las tareas en las que se descompone cada una de ellas.
- Concretar los artefactos que se manejan (entradas y salidas de las actividades, entregables, etc.).
- Identificar, para cada actividad, el catálogo de técnicas o herramientas aplicable durante las distintas actividades de la metodología.

3.2.2 Catálogo de Roles

En esta sección se detallan los roles y agrupaciones de éstos que se identifican a lo largo de la metodología de gestión de proyectos de especificación de requisitos de DQ, así como las principales tareas y responsabilidades de cada uno de ellos. Estos roles serán localizados después en cada una de las actividades de la metodología en las que participen.

En la Figura 6 se presentan los roles involucrados en cada una de las actividades de la metodología DQ-VORD, así como la relación que existe entre ellos. Como se aprecia en la Figura 6, existe una comunicación directa a través del Jefe de Desarrollo y el Arquitecto Jefe. Éstos a su vez serán los que se comuniquen directamente con sus equipos internos de trabajo.

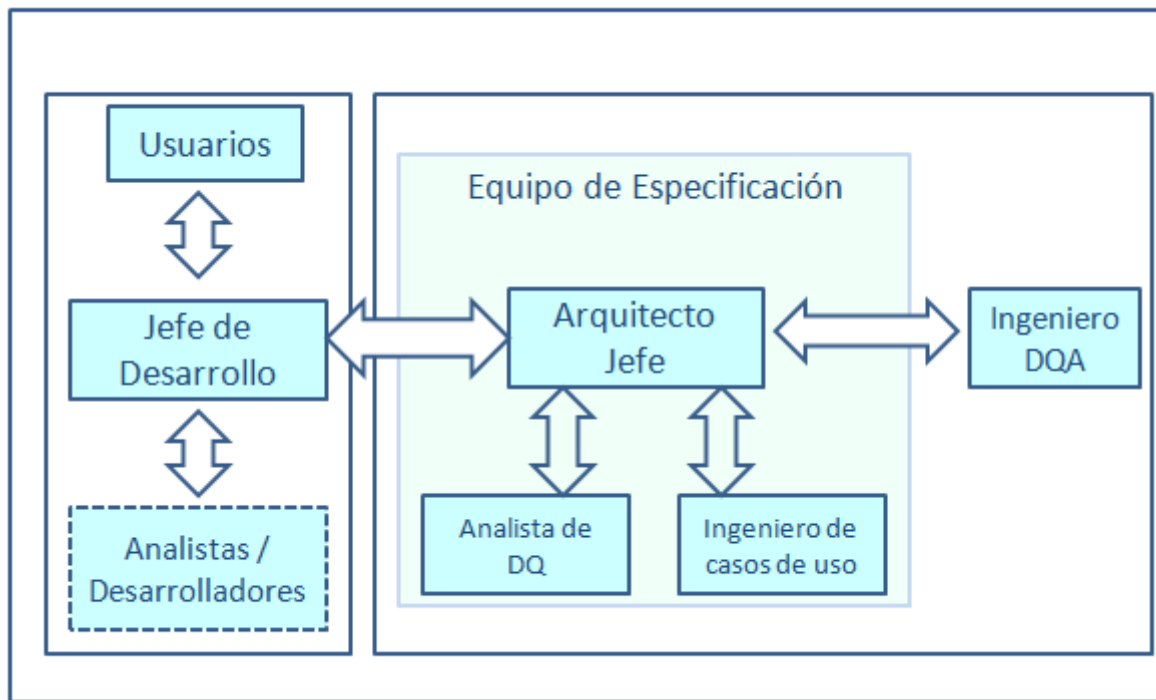


Figura 6. Roles involucrados en la metodología DQ-VORD.

3.2.2.1 Jefe de Desarrollo

Este rol puede estar representado por el Jefe de un proyecto o el analista responsable de la especificación de requisitos software de un sistema.

Normalmente, el Jefe de desarrollo tiene a su cargo un equipo de trabajo formado por analistas y/o desarrolladores (como se señala en la Figura 6 en un recuadro punteado), cuyo objetivo es el de apoyar durante el proceso de especificación de requisitos software, además de analizar los resultados reportados por el equipo de especificación.

Entre las principales responsabilidades del Jefe de desarrollo se encuentran las siguientes:

- Verificar que todos los miembros del equipo de especificación disponen de la experiencia y conocimientos necesarios para llevar a cabo el proceso de especificación de requisitos.
- Asegurar que los recursos necesarios para la evaluación están disponibles y en la cantidad requerida.
- Asegurar que el equipo de Analistas/Desarrolladores tiene acceso a los recursos necesarios durante el proceso de especificación.
- Asegurar que los usuarios estarán disponibles durante el proceso de especificación.
- Mantener la comunicación con el Equipo de Especificación

(específicamente con el Arquitecto Jefe) en lo referente a la evolución y resultados del proceso de especificación de los requisitos de DQ.

3.2.2.2 Analistas / Desarrolladores

El equipo de analistas y desarrolladores de la empresa sirve de apoyo al Jefe de desarrollo. Su principal responsabilidad consiste en analizar los resultados reportados por el Equipo de Especificación para detectar posibles defectos o inconsistencias. Normalmente este equipo de personas es el encargado de desarrollar e implementar las funcionalidades del sistema (a partir de un documento de Especificación de Requisitos Software, ERS).

3.2.2.3 Usuarios (Punto de vista)

Representa el usuario que interactuará directamente con la aplicación a ser desarrollada. Normalmente este rol es el encargado de definir todos los tipos de requisitos software (funcional, no funcional, de calidad de datos, etc.) de la aplicación. Este rol dentro de la metodología DQ-VORD, toma el nombre de “*punto de vista*”. De forma particular, cada punto de vista podrá definir requisitos de calidad de datos relacionados a cada una de las funcionalidades requeridas.

3.2.2.4 Equipo de Especificación

Representa el equipo formado por el Arquitecto Jefe y un conjunto de uno o varios Analistas de calidad de datos e Ingenieros de casos de uso (según los requisitos del proyecto). El principal objetivo del equipo de especificación es cumplir con los objetivos acordados con el jefe de desarrollo dentro de los tiempos planificados. Es muy importante la coordinación y comunicación tanto del arquitecto jefe con los analistas de calidad de datos, como entre los propios analistas de calidad de datos e ingenieros de casos de uso, puesto que de ella dependerá el progreso y éxito final de la metodología de gestión de proyectos de especificación de requisitos de calidad de datos.

3.2.2.5 Arquitecto Jefe

El Arquitecto Jefe es líder del equipo de especificación, su principal objetivo es asegurar el correcto desarrollo de la especificación de requisitos software de calidad de datos. Este rol debe ser ocupado por una persona con suficientes conocimientos en la elicitación, especificación y análisis de requisitos software, y con experiencia en la aplicación práctica en entornos de desarrollo de aplicaciones Web. Además de las principales normas y estándares sobre calidad de datos, especificación de requisitos y procesos software.

Entre las principales responsabilidades del arquitecto jefe se encuentran las siguientes:

- Alcanzar un acuerdo con el Jefe de desarrollo para una correcta ejecución de la aplicación de la metodología DQ-VORD.
- Controlar y supervisar que todas las actividades se desarrollen de acuerdo a lo expuesto en la presente metodología de gestión de proyectos de especificación de requisitos software de DQ.
- Asegurar que todos los involucrados / implicados están informados sobre el objetivo, alcance y enfoque del proyecto que se va a realizar.
- Garantizar que todos los miembros del equipo de trabajo tienen el conocimiento y la experiencia necesaria para llevar a cabo las actividades de especificación que se van a realizar, incluyendo el manejo de las herramientas y técnicas seleccionadas.
- Asegurar que todos los miembros del equipo de trabajo tienen acceso a la presente metodología, así como a la documentación necesaria que les permita realizar sus actividades.
- Confirmar la aceptación por parte del Jefe de desarrollo de los resultados obtenidos al final del proyecto.

3.2.2.6 Analista de Calidad de Datos

El Analista de Calidad de Datos es el rol encargado de ejecutar cada una de las actividades y tareas propias de la metodología de especificación de requisitos software de DQ, como integrante del equipo de especificación. Este rol deberá ser ocupado por una persona con conocimientos avanzados en especificación de requisitos software, calidad de datos y la metodología DQ-VORD. Las principales responsabilidades del analista de calidad de datos son:

- Realizar las actividades de especificación asignadas por el arquitecto jefe, como por ejemplo: detallar la planificación, recolectar la información para especificar los requisitos software de DQ, utilizar las herramientas de elicitación, analizar y agrupar los resultados, etc.
- Recoger información de las distintas actividades y tareas de especificación de requisitos software de DQ.

3.2.2.7 Ingeniero de Casos de Uso

El Ingeniero de Casos de Uso es el rol responsable de la integridad de las realizaciones de cada caso de uso (funcionalidades que proveerá la aplicación Web), garantizando que cumplen los requisitos que recaen sobre ellos (tanto requisitos funcionales, como de calidad de datos).

Su principal actividad es el modelado de los distintos casos de uso y casos de información (Guerra-García, Caballero et al. 2009) definidos en las actividades de la metodología. Para dar soporte a este modelado se llevaron a cabo los metamodelos mostrados posteriormente en los apartados 3.2.6 y 3.2.7).

3.2.2.8 Ingeniero DQA

El Ingeniero DQA (*Data Quality Assurance*) es el rol responsable de todos los procesos del Equipo de Especificación relacionados con la elicitación, análisis, control y especificación de la calidad de los datos del software. Su papel es fundamental en la especificación de requisitos software de DQ sobre los que trata la presente metodología.

El Ingeniero DQA no forma parte directa del equipo de especificación, sin embargo, será a él a quién el Arquitecto Jefe informe sobre los posibles cambios o mejoras detectados para la metodología DQ-VORD durante los proyectos, así como los defectos importantes que puedan ser evitados en el futuro. Él será el encargado de estudiar estas propuestas y asegurar que se realizan las adaptaciones necesarias para la correcta evolución y mejora de la presente metodología de gestión de proyectos de especificación de requisitos de DQ.

3.2.3 Catálogo de Elementos

En esta sección se detallan todos los elementos (documentos, informes, catálogos, artefactos, etc.) que se identifican a lo largo de las actividades de la metodología de gestión de proyectos de especificación de requisitos de DQ. Estos elementos podrán ser de entrada, de salida, y/o de entrada/salida para las actividades de especificación, siendo además algunos de ellos entregables finales una vez llevada a cabo la presente metodología.

3.2.3.1 Modelo de selección de Características de Calidad de Datos Genérico (MOSCAF)

Contiene el conjunto de características (dimensiones) de calidad de datos y las relaciones entre ellas, que se pueden especificar y/o relacionar a cada una de las funcionalidades que serán implementadas en la aplicación Web (*modelo MOSCAF*), además de la guía para su especificación. Para la realización de este modelo se tomó como referencia principalmente tanto el estándar ISO/IEC 25012 (ISO-25012 2008), como la especificación de funcionalidades Web definidas por Collins en (Collins 2001).

El objetivo principal es proveer al analista con una base de la cual partir, para

llevar a cabo la especificación de los requisitos de DQ. En este sentido, un requisito de DQ provisto por cada usuario mostrará las dimensiones de DQ requeridas o necesarias para ciertos datos al momento de ser utilizados en determinadas funcionalidades específicas. Una descripción detallada del modelo MOSCAF se muestra en la sección 3.2.5.

3.2.3.2 Catálogo de Técnicas de Especificación

Contiene las técnicas de especificación disponibles para ser seleccionadas en cada una de las actividades y tareas de especificación de requisitos de DQ. Estas técnicas se mencionan en cada una de las actividades de la metodología DQ-VORD. Algunas de ellas pueden ser (Kotonya y Sommerville 2002): entrevistas, estudio de documentación, tormenta de ideas, cuestionarios, sesiones de trabajo, juicio de expertos, etc.

3.2.3.3 Catálogo de Herramientas de Especificación

Describe el conjunto de herramientas disponibles para llevar a cabo el proyecto de especificación. El uso de estas herramientas se explica en cada una de las actividades de la metodología de gestión de proyectos de especificación de requisitos software de DQ. Algunas de ellas son: procesadores de texto, herramientas para el diseño Web, herramientas de modelado UML, etc.

3.2.3.4 Documento de Especificación de Requisitos del Sistema Aumentado con Especificación de Requisitos de DQ (ERSDQ)

Este artefacto es el **primer entregable** de la metodología de gestión de proyectos de especificación de requisitos software de DQ, y como tal, debe estar disponible para todos los involucrados en el proyecto.

Este documento será generado como resultado final una vez llevada a cabo todas las actividades y tareas de la metodología (DQ-VORD), y estará formado por un documento donde se recogen todos los requisitos de calidad de datos asociados a cada una de las funcionalidades que serán implementadas en la aplicación.

Una característica importante de este documento es la necesidad de retroalimentación por parte de los usuarios. Una vez que el documento se ha entregado al Jefe de Desarrollo, éste debería disponer de un tiempo para estudiar los resultados e informar al Arquitecto Jefe cualquier defecto o incoherencia detectada, con el objetivo de aclarar las posibles dudas y perfeccionar el documento de especificación de requisitos software. Para ello utilizará el documento de “Petición de Modificación”, que se describe posteriormente.

El documento de Especificación de requisitos del sistema aumentado con especificación de requisitos de DQ, es un documento sujeto a cambios. Por tanto, será almacenado en un sistema de control de versiones dentro de la empresa.

3.2.3.5 Documento del Modelo de Diseño Sensible a Calidad de Datos (MDSDQ)

Este artefacto representa el **segundo entregable** de la metodología de gestión de proyectos de especificación de requisitos software de DQ, y como tal, es público para todo el personal involucrado en el proyecto.

Este documento será generado como resultado final una vez llevada a cabo todas las actividades y tareas de la metodología DQ-VORD, y estará formado por un documento donde se modelan principalmente el modelo de datos y el modelo de procesos (por ejemplo: *modelo de contenido, modelo de navegación, modelo de presentación, etc.*) de la aplicación Web.

Una característica importante de este documento es la necesidad de retroalimentación. Una vez que el documento se ha entregado al Jefe de Desarrollo, dispondrá de un tiempo para estudiar los resultados y reportar al Arquitecto Jefe cualquier defecto o incoherencia detectada, con el objetivo de aclarar las posibles dudas y perfeccionar el documento.

Este documento estará sujeto a cambios. Por lo que podrá ser almacenado en el sistema de control de versiones de la empresa, dentro del repositorio habilitado para dicho proyecto.

3.2.3.6 Petición de Modificación

Este artefacto representa el documento mediante el cual el jefe de desarrollo, una vez habiendo revisado alguno de los documentos entregables 1 o 2 (ERSDQ, MDSDQ) entregados por el arquitecto jefe, manifiesta sus opiniones y posibles cambios respecto a los resultados obtenidos en dichos documentos.

Este documento será entregado al equipo de especificación y éste lo revisará y actualizará si fuese necesario de acuerdo a los cambios sugeridos. Una vez modificado el documento en cuestión, será de nuevo entregado al jefe de desarrollo, quién podrá solicitar una nueva reunión o reportar una nueva petición de modificación, actuándose de igual manera que la primera vez. El resultado de esta petición de modificación será el refinamiento de dicho documento así como la mejora del propio proceso de gestión.

3.2.3.7 Información Interna del Proyecto de Especificación de Requisitos de Calidad de Datos

Durante la ejecución de la metodología de gestión de proyectos de especificación de requisitos de DQ se genera, además de los propios entregables, un conjunto de documentación relacionada con la propia aplicación de la metodología. Esta documentación estará principalmente formada por los productos de salida obtenidos en cada una de las actividades y tareas de la presente metodología DQ-VORD.

3.2.4 Descripción de las actividades de la metodología DQ-VORD

La metodología para la gestión de proyectos de especificación de requisitos de calidad de datos (DQ-VORD) se encuentra compuesta por **cuatro actividades principales**, tal y como se observa en la Figura 7:

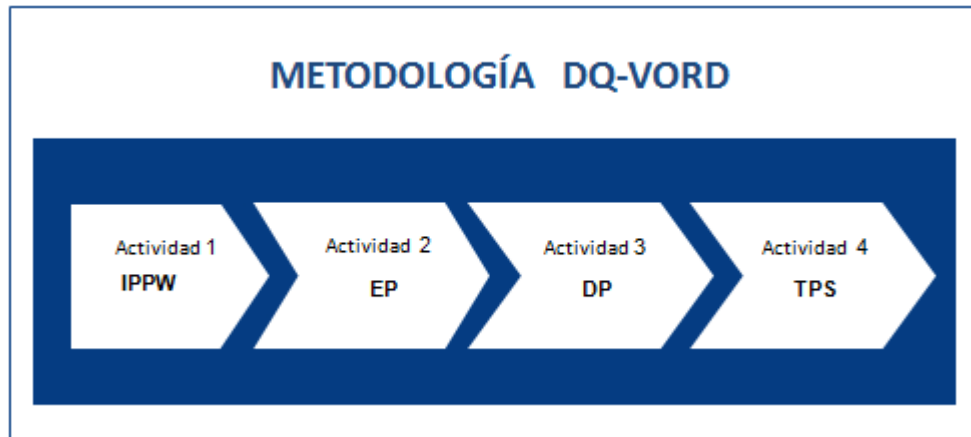


Figura 7. Actividades principales de la metodología DQ-VORD.

- **IPPW. Identificación de los Puntos de Vista del Portal Web:** esta actividad está enfocada a descubrir los diferentes puntos de vista (usuarios) que usarán/ejecutarán las funcionalidades de la aplicación, las cuales serán susceptibles de asociarles requisitos de calidad de datos.
- **EP. Estructuración de los Puntos de Vista:** esta actividad consiste en agrupar las funcionalidades y puntos de vista relacionados en una jerarquía, con el objetivo de organizarlos de forma apropiada y resolver posibles conflictos entre requisitos de calidad de datos para las funcionalidades establecidas por los diferentes puntos de vista.
- **DP. Documentación de los Puntos de Vista:** en esta actividad se refinará la descripción de los puntos de vista y las funcionalidades identificadas, además de agregar los requisitos de DQ mediante el modelado de las distintas dimensiones de DQ a través de distintos diagramas.
- **TPS. Trazado de los Puntos de Vista del Sistema:** comprende identificar los objetos principales en un diseño orientado a objetos usando la información de la funcionalidad encapsulada en los puntos de vista. Asimismo, se lleva a cabo el modelado de diseño a través de distintos diagramas específicos para ello.

3.2.4.1 IPPW. Identificación de los Puntos de Vista del Portal Web

Esta actividad está enfocada a identificar las funcionalidades críticas desde el punto de vista de calidad de datos, así como los diferentes actores del sistema (puntos de vista) (usuarios) que usarán dichas funcionalidades del portal Web, con la idea de capturar los requisitos de calidad de datos asociados.

Esta actividad se compone de tres tareas bien diferenciadas (Figura 8), en cada una de las cuales se indica el objetivo que se pretende, los productos de entrada y salida, así como las técnicas y herramientas utilizadas, además de los roles involucrados.

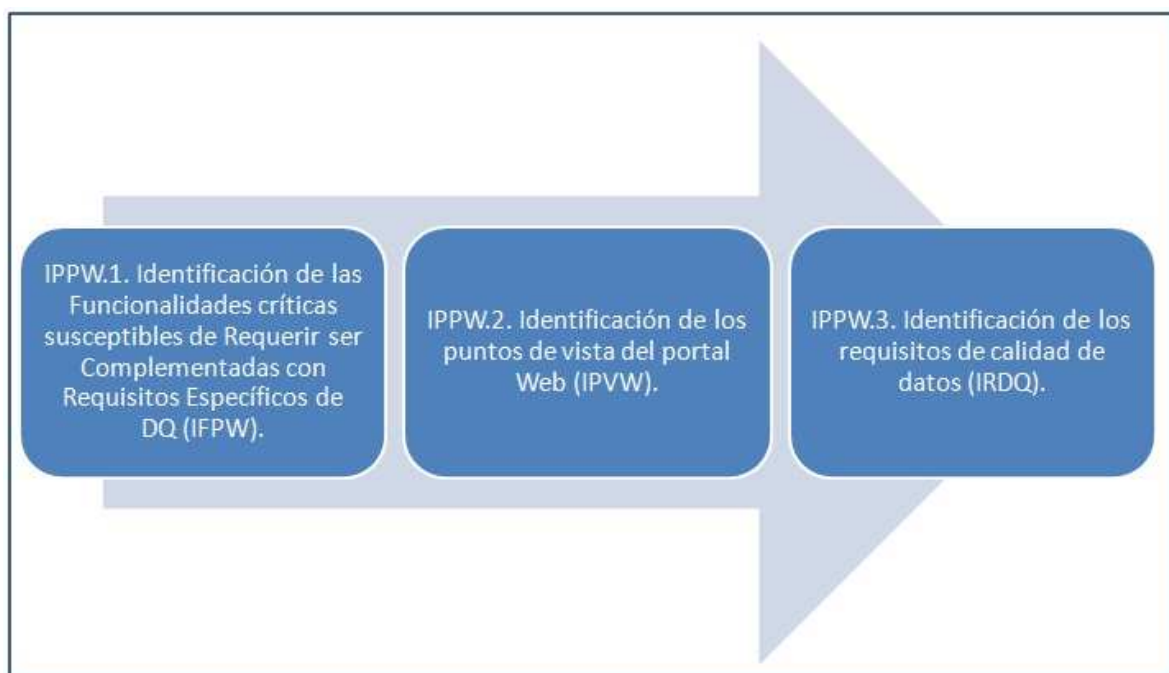


Figura 8. Tareas definidas en la actividad 1 "IPPW".

3.2.4.1.1 TAREA IPPW.1: Identificación de las Funcionalidades críticas susceptibles de Requerir ser Complementadas con Requisitos Específicos de DQ (IFPW)

El objetivo de esta tarea es identificar las funcionalidades críticas específicas susceptibles a ser complementadas con requisitos de DQ, que serán provistas para cada punto de vista. Una vez identificadas estas funcionalidades críticas, deberán ser catalogadas en base a las funcionalidades definidas por Collins en (Collins 2001). Lo anterior, con el objetivo de facilitar posteriormente la especificación de los requisitos específicos de DQ.

El principal producto de salida de esta tarea es una *lista de funcionalidades*

críticas susceptibles de necesitar requisitos específicos de DQ de cada punto de vista (LFWI).

A continuación se enumeran cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 3).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Documento de Especificación de Requisitos Software (ERS). - Lista de todas las funcionalidades del portal Web definidas por Collins en (Collins 2001) (FWC).	- Lista de funcionalidades críticas susceptibles de necesitar requisitos específicos de calidad de datos de cada punto de vista (LFWI).	- Entrevistas. - Estudio de documentación. - Tormenta de ideas. - Observación.	- Arquitecto Jefe. - Analista de calidad de datos. - Usuarios. - Analistas / Desarrolladores.

Tabla 3. Productos, técnicas y roles involucrados en la tarea "Identificación de las funcionalidades críticas susceptibles de requerir ser complementadas con requisitos específicos de DQ (IFPW)."

3.2.4.1.2 TAREA IPPW.2: Identificación de los puntos de vista del portal Web (IPVW)

El objetivo de esta tarea es descubrir los diferentes puntos de vista (usuarios) que ejecutarán las funcionalidades críticas de la aplicación Web susceptibles a requisitos de calidad de datos.

El principal producto de salida de esta tarea es una *lista de puntos de vista identificados capaces de proponer requisitos software de DQ para la aplicación (PVI).*

A continuación se mencionan cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 4).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de funcionalidades críticas susceptibles de necesitar requisitos específicos de calidad de datos de cada punto de vista (LFWI). - Documento de Especificación de Requisitos Software (ERS).	- Lista de los puntos de vista identificados capaces de proponer requisitos de calidad de los datos para la aplicación (PVI).	- Entrevistas. - Estudio de documentación. - Cuestionarios. - Tormenta de ideas. - Observación.	- Arquitecto Jefe. - Analista de calidad de datos. - Usuarios. - Analistas / Desarrolladores.

Tabla 4. Productos, técnicas y roles involucrados en la tarea "Identificación de los puntos de vista del portal Web (IPVW)."

3.2.4.1.3 TAREA IPPW.3: Identificación de los Requisitos de DQ (IRDQ)

El objetivo de esta tarea es identificar para cada uno de las funcionalidades consideradas como críticas, aquellos distintos requisitos de DQ relacionados a cada una de las funcionalidades identificadas previamente y descritas para cada punto de vista. Para facilitar esta labor se recomienda el uso del “*Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web – MOSCAF*” desarrollado como parte de esta tesis y descrito en profundidad en la sección 3.2.5 (Guerra-García, Caballero et al. 2009).

Para un mayor entendimiento acerca del concepto de **requisito de DQ**, se siguió la definición propuesta (específicamente en el apartado 3.1), y publicada en (Guerra-García, Caballero et al. 2010).

A continuación se listan cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 5).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de los puntos de vista identificados capaces de proponer requisitos de calidad de los datos para la aplicación (PVI). - Lista de funcionalidades críticas susceptibles de necesitar requisitos específicos de calidad de datos de cada punto de vista (LFWI).	- Instanciación del modelo de Calidad de datos para el problema particular bajo estudio (IMDQ).	- MOSCAF.	- Arquitecto Jefe. - Analista de calidad de datos.

Tabla 5. Productos, técnicas y roles involucrados en la tarea “Identificación de requisitos de DQ (IRDQ)”

3.2.4.2 EP. Estructuración de los Puntos de Vista

Esta actividad está enfocada a llevar a cabo una clasificación tanto de los puntos de vista identificados previamente como de los requisitos de calidad de datos obtenidos.

Esta actividad se compone de dos tareas (Figura 9), en las cuales se indican los objetivos que se pretenden, los productos de entrada y salida, así como las técnicas y herramientas utilizadas.

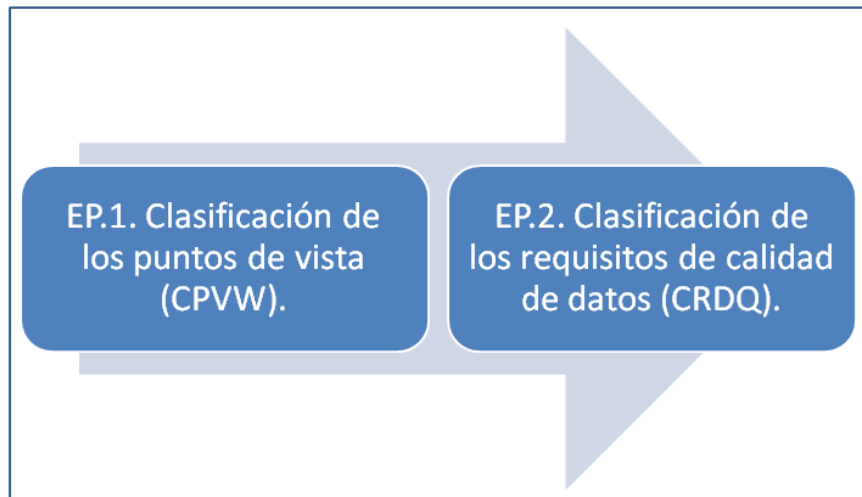


Figura 9. Tareas definidas en la actividad 2 "EP"

3.2.4.2.1 TAREA EP.1: Clasificación de los Puntos de Vista (CPVW)

Esta tarea consiste en agrupar los puntos de vista relacionados en una jerarquía, con el objetivo de minimizar posibles dependencias y diferencias entre sus requisitos de DQ. Para llevar a cabo esta jerarquización, primeramente se identifican las funcionalidades más comunes e importantes, ubicándose en los niveles más altos de la jerarquía, y heredándose a los puntos de vista de bajo nivel.

El principal producto de salida de esta tarea es una *lista jerarquizada de los puntos de vista y funcionalidades Web relacionadas (LJPV)*.

A continuación se mencionan cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 6).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de los puntos de vista identificados capaces de proponer requisitos de calidad de los datos para la aplicación (PVI). - Instanciación del modelo de Calidad de datos para el problema particular bajo estudio (IMDQ).	- Lista jerarquizada de los puntos de vista y funcionalidades Web relacionadas (LJPV).	- Sesiones de trabajo. - Juicio y opiniones de expertos.	- Arquitecto Jefe. - Analista de calidad de datos. - Ingeniero de casos de uso.

Tabla 6. Productos, técnicas y roles involucrados en la tarea "Clasificación de los puntos de vista (CPVW)"

3.2.4.2.2 TAREA EP.2: Clasificación de los Requisitos de Calidad de Datos (CRDQ)

El objetivo de esta tarea consiste en clasificar los requisitos de calidad de datos de acuerdo a una jerarquía, considerando el nivel de criticidad que tendrían con respecto a las funcionalidades de la aplicación Web sobre las que se definen. Esto permitirá al analista de DQ, el darse cuenta del nivel de importancia que tiene cada uno de los requisitos de DQ, y decidir si así fuese el caso, cuál de ellos debería implementarse en primera instancia.

El principal producto de salida de esta tarea es una *lista de clasificación de los requisitos de calidad de datos (LCRDQ)*.

Enseguida se muestran cada uno de los productos de entrada y salida que se pueden utilizar, así como las técnicas o herramientas y los roles involucrados (véase Tabla 7).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de los puntos de vista identificados capaces de proponer requisitos de calidad de los datos para la aplicación (PVI). - Instancia del modelo de Calidad de datos para el problema particular bajo estudio (IMDQ). - Lista jerarquizada de los puntos de vista y funcionalidades Web relacionadas (LJPV).	- Lista de clasificación de los requisitos de calidad de datos (LCRDQ).	- Sesiones de trabajo. - Juicio de expertos.	- Arquitecto Jefe. - Analista de calidad de datos. - Ingeniero de casos de uso.

Tabla 7. Productos, técnicas y roles involucrados en la tarea “Clasificación de los requisitos de calidad de datos (CRDQ)”

3.2.4.3 DP. Documentación de los Puntos de Vista

Esta tercera actividad consiste en refinar la descripción de los puntos de vista y las funcionalidades identificadas, además de agregar una descripción más detallada de los distintos requisitos de DQ previamente obtenidos.

Esta actividad se compone de dos tareas principales (véase Figura 10), en las cuales se indican los objetivos que se pretenden cumplir, los productos de entrada y salida, así como las técnicas y herramientas utilizadas.

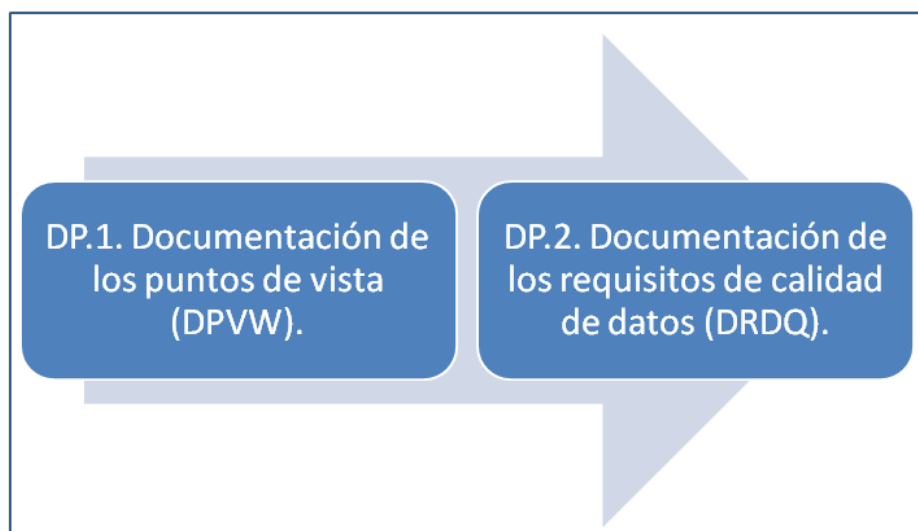


Figura 10. Tareas definidas en la actividad 3 "DP"

3.2.4.3.1 TAREA DP.1: Documentación de los Puntos de Vista (DPVW)

El objetivo de esta tarea de la metodología, consiste en refinar la descripción de los puntos de vista y las funcionalidades identificadas, a fin de mostrar una descripción más detallada y completa tanto de los puntos de vista como de las funcionalidades seleccionadas. Para ello, se han de utilizar algunas de las plantillas propuestas en la sección 3.2.8, de forma específica las mostradas en la Tabla 38 y Tabla 39.

A continuación se mencionan cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 8).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de funcionalidades críticas susceptibles de necesitar requisitos específicos de calidad de datos de cada punto de vista (LFWI). - Lista jerarquizada de los puntos de vista y funcionalidades Web relacionadas (LJPV). - Lista de clasificación de los requisitos de calidad de datos (LCRDQ). - Documento de Especificación de Requisitos Software (ERS).	- Primera versión del "Documento de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)".	- Herramientas software: procesadores de texto, herramientas CASE, etc. - Sesiones de trabajo. - Juicio de expertos. - Herramientas de modelado de diagramas UML.	- Arquitecto Jefe. - Analista de calidad de datos. - Ingeniero de casos de uso.

Tabla 8. Productos, técnicas y roles involucrados en la tarea "Documentación de los Puntos de Vista (DPVW)".

3.2.4.3.2 TAREA DP.2: Documentación de los Requisitos de Calidad de Datos (DRDQ)

El objetivo de esta tarea consiste en documentar y modelar en la medida de lo posible, los requisitos de calidad de datos identificados (por ejemplo, mediante la extensión de los diagramas de casos de uso o diagramas de casos de información², modelados a través del metamodelo propuesto “DQ_WebRE”, descrito en profundidad en la sección 3.2.6).

A continuación se muestran cada uno de los productos de entrada y salida que se pueden utilizar, así como las técnicas o herramientas y los roles involucrados (véase Tabla 9).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Lista de clasificación de los requisitos de calidad de datos (LCRDQ). - Lista jerarquizada de los puntos de vista y funcionalidades Web relacionadas (LJPV). - Documento de Especificación de Requisitos Software (ERS). - Primera versión del “Documento de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)”.	- Documento completo de “Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)”.	- Herramientas software: procesadores de texto, herramientas CASE, etc. - Herramientas de modelado de diagramas UML. - Perfil UML para la especificación de requisitos de DQ en aplicaciones Web (DQ_WebRE) implementado en la herramienta Enterprise Architect. - Metamodelo “DQ_WebRE” desarrollado como plugin en Eclipse para hacerlo operativo (descrito en Anexo B). - Sesiones de trabajo. - Juicio de expertos.	- Arquitecto Jefe. - Analista de calidad de datos. - Ingeniero de casos de uso.

Tabla 9. Productos, técnicas y roles involucrados en la tarea “Documentación de los requisitos de calidad de datos (DRDQ)”

3.2.4.4 TPS. Trazado de los Puntos de Vista del Sistema

Esta es la cuarta actividad de la metodología, consiste en identificar los objetos

² Diagrama de Caso de Información (DCI). Metamodelo UML (DQ_WebRE) que describe un conjunto de estereotipos definidos para llevar a cabo la especificación de requisitos software de calidad de datos en aplicaciones Web, publicado en Guerra-García, C., I. Caballero, et al. (2011). Capturing Data Quality Requirements for Web Applications by means of DQ_WebRE. 2nd International Workshop on Business intelligence and the WEB, BEWEB 2011, Uppsala, Sweden, ACM 978-1-4503-0610-2/11/03.

principales en un modelo de diseño (el cual contiene clases de diseño) orientado a objetos usando la información de la funcionalidad encapsulada en los puntos de vista.

Para lograr el objetivo de esta actividad se ejecutarán dos tareas bien diferenciadas (véase Figura 11), en cada una de las cuales se indica el objetivo que se pretende, los productos de entrada y salida, así como las técnicas y herramientas utilizadas, además de los roles involucrados.

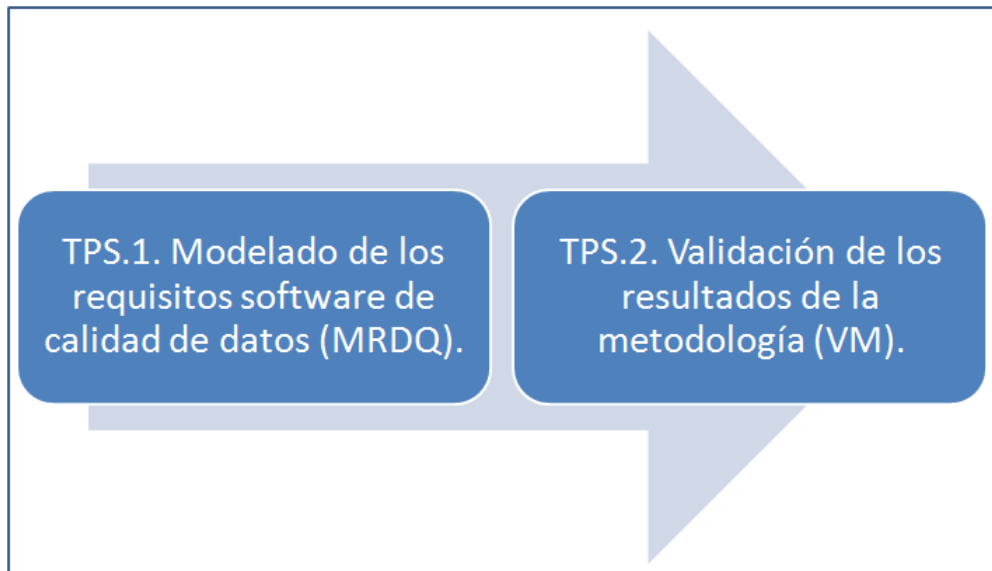


Figura 11. Tareas definidas en la actividad 4 “TPS”.

3.2.4.4.1 TAREA TPS.1: Modelado de los Requisitos Software de Calidad de Datos (MRDQ)

El objetivo de esta tarea es llevar a cabo una interpretación clara de las *dimensiones de DQ* hacia sus correspondientes *requisitos software de calidad de datos*. Permitiendo modelar los diferentes requisitos (dimensiones) de DQ en un modelo de datos y posteriormente en un modelo de procesos (haciendo uso del metamodelo para el diseño de aplicaciones Web sensibles a aspectos de calidad de datos - *DAQ_UWE* – el cual permite modelar distintos tipos de diagramas como: de contenido, de presentación, de navegación, etc.). El metamodelo *DAQ_UWE* se describe en la sección 3.2.7.

A continuación se mencionan cada uno de los productos de entrada y salida que se pueden utilizar, las técnicas o herramientas, además de los roles involucrados (véase Tabla 10).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Documento de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ).	- Documento del Modelo de Diseño con conciencia de calidad de datos, incluyendo tanto el modelo de datos como el modelo de procesos (MDSDQ).	- Perfil UML para el Diseño de aplicaciones Web sensibles a aspectos de calidad de datos (DAQ_UWE) implementado en la herramienta Enterprise Architect. - Herramientas de modelado orientado a objetos (Rational Rose, Visual Paradigm, Poseidon, ArgoUML, etc.).	- Analista de calidad de datos. - Ingeniero de casos de uso. - Analistas / Desarrolladores.

Tabla 10. Productos, técnicas y roles involucrados en la tarea “Modelado de los requisitos software de calidad de datos (MRDQ)”

3.2.4.4.2 TAREA TPS.2: Validación de los Resultados de la Metodología (VM)

El objetivo de esta tarea consiste principalmente en la validación de los dos entregables obtenidos mediante el uso de la metodología (“ERSDQ” y “MDSDQ”) entre los principales implicados, de forma específica el Arquitecto jefe y el Jefe de desarrollo.

Enseguida se muestran cada uno de los productos de entrada y salida que se pueden utilizar, así como las técnicas o herramientas y los roles involucrados (véase Tabla 11).

Productos de Entrada	Productos de Salida	Técnicas y Herramientas	Roles
- Documento de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ). - Documento del Modelo de Diseño con conciencia de calidad de datos (MDSDQ).	- Documento Final aprobado de “Especificación de Requisitos del Sistema aumentado con Especificación de Requisitos Software de DQ (ERSDQ)”. - Documento Final aprobado del “Modelo de Diseño con conciencia de calidad de datos (MDSDQ)”.	- Sesiones de trabajo. - Técnicas de negociación interpersonal.	- Arquitecto Jefe. - Analista de calidad de datos. - Jefe de desarrollo. - Ingeniero DQA.

Tabla 11. Productos, técnicas y roles involucrados en la tarea “Validación de los resultados de la metodología (VM)”

3.2.5 Modelo propuesto para la Selección de Requisitos de Calidad de Datos según las Funcionalidades a Implementar en una Aplicación Web (MOSCAF)

Varios autores (Wang, Reddy et al. 1995; Strong, Lee et al. 1997; Wang 1998; Loshin 2001; Scannapieco, Pernici et al. 2002; Oliveira, Rodrigues et al. 2005; Caballero, Caro et al. 2008) han desarrollado distintas propuestas en el área de calidad de datos enfocadas principalmente a la identificación e inclusión de características y elementos de DQ como parte del desarrollo de sistemas de información. Sin embargo, y hasta nuestro conocimiento (véase Anexo A sobre la revisión sistemática realizada) no existe propuesta alguna en el que se relacione de alguna forma directa la gestión de características de calidad de datos con el desarrollo de aplicaciones Web, y de forma más específica en la especificación y análisis de requisitos.

Así, como parte de los objetivos parciales de esta tesis (específicamente el objetivo parcial 3) en este punto se presenta un modelo que permite identificar los requisitos de calidad de datos para cada uno de los tipos de funcionalidades que se podrán incorporar en una aplicación Web. La estrategia de trabajo que se siguió, fue la de realizar un análisis que incluyó un doble mapeo: por un lado se intentó de forma genérica identificar los requisitos de calidad de datos asociados a cada una de las funcionalidades Web, y por otro se identificaron las dimensiones de calidad de datos que componen cada uno de estos requisitos.

Para afrontar la primera parte del trabajo (identificación de requisitos) se realizó un mapeo entre las funcionalidades de un portal Web dadas en (Collins 2001) y los obstáculos identificados en (Strong, Lee et al. 1997) que pueden existir en un sistema de información. Esto nos permitió identificar las dimensiones de calidad de datos que se podrían ver implicadas como posibles requisitos de calidad de datos exigibles para una funcionalidad a implementar en una aplicación Web. En un segundo mapeo de las funcionalidades versus las dimensiones de calidad de datos mostradas por Strong et al. en (Strong, Lee et al. 1997), se obtuvo una versión inicial del conjunto de dimensiones de calidad de datos que está asociada a cada uno de los requisitos de calidad de datos. No obstante, esta versión inicial fue complementada con un análisis más profundo llevado a cabo en función de la experiencia del doctorando tras varios años trabajando en el desarrollo de aplicaciones Web y en la aplicación de modelos de calidad de datos. Finalmente se incluyó una guía para que el usuario pueda trabajar mejor con el modelo.

3.2.5.1 Funcionalidades de una aplicación Web

Como se mencionó previamente en la introducción, y como parte de uno de

nuestros objetivos parciales de investigación, es necesario especificar los requisitos de DQ asociados a cada funcionalidad que será implementada en una aplicación Web. Por lo que, primero es preciso conocer cuáles serán los distintos tipos de funcionalidades que pueden existir en una aplicación Web. En esta tesis se toma como base la enumeración propuesta en (Collins 2001). Además, a la hora de identificar cuáles pueden ser las más críticas, se ha tomado el siguiente criterio: *“A mayor probabilidad de utilización de una funcionalidad de un portal Web, mayor probabilidad de que ocurran no conformidades en los niveles aceptables de calidad de los datos que se utilizan en dicha funcionalidad”*.

Es necesario resaltar que el orden que se ha propuesto según el criterio se ha establecido en base a la experiencia y conocimiento del doctorando en ambas áreas, tanto el desarrollo Web como calidad de datos. En la Tabla 12 se muestra la lista de funcionalidades ordenadas según el criterio establecido junto con una pequeña descripción desde el punto de vista del desarrollador.

Funcionalidad	Descripción
Gestión de Contenido	La publicación y distribución de información debe soportar la creación de contenido, autorización e inclusión (o exclusión) de colecciones de contenido. Las organizaciones tienen volúmenes de información y material de referencia que se dejan disponible desde el portal.
Procesos y Acciones	Permiten a los usuarios del portal iniciar y participar en procesos de negocios de la organización.
Capacidades de búsqueda	Se deben proporcionar servicios de búsqueda de información para los usuarios, que soporten búsqueda a lo largo de la empresa, en el World Wide Web, y en motores de búsqueda de catálogos e índices.
Administración	Esta funcionalidad proporciona dos servicios: el primero, correspondiente a actividades de mantenimiento y utilización de las tareas asociadas con el Sistema del Portal Corporativo. El segundo es un servicio que puede ser configurado únicamente por el administrador con el Sistema del Portal Corporativo, así como por cada usuario a través de la personalización.
Seguridad	Proporciona una descripción de los niveles de acceso de cada usuario o grupo de usuarios, para cada aplicación y función de software incluidas en el portal.
Integración y puntos de datos	Permite la capacidad de acceder a información desde un amplio rango de fuentes de información internas y externas, y mostrar la información resultante como un simple punto de acceso en el escritorio.
Comunicación y Colaboración	Estos elementos facilitan la discusión, la identificación de ideas innovadoras y soluciones ingeniosas o emprendedoras. Estos elementos dan la posibilidad a los empleados de trabajar juntos de forma más cualitativa mediante la creación de un entorno virtual compartido (colaboración), soportando mensajería electrónica (comunicación) y agregando características de comunicación y colaboración al proceso de negocios (coordinación).
Presentación	Provee la experiencia visual para el usuario, encapsulando todas las funcionalidades del portal.
Taxonomía	La taxonomía proporciona información de contexto incluyendo categorías específicas de la organización que soportan y reflejan los negocios de ésta. Provee terminología común usada en la organización de reconocimiento rápido y mejora la semántica para los usuarios del portal (empleados, socios, clientes, etc.).
Personalización	Éste es un componente crítico para crear un ambiente de trabajo organizado y configurado especialmente para cada empleado en la organización. La clave está en lograr un balance entre la información y los atributos que se necesitan para ser consistentes y constantes en la personalización del portal y las características únicas requeridas por los empleados individuales.
Características de Ayuda	Asistencia en el uso del portal. La ayuda debe estar referida tanto a características del portal como a características específicas de la organización.

Tabla 12. Funcionalidades de una aplicación Web.

En las siguientes figuras se muestran ejemplos típicos de algunas de las funcionalidades implementadas dentro de una aplicación Web. En la Figura 12 se

observa la funcionalidad de “*Capacidad de búsqueda*”, en la cual se muestra la capacidad de una aplicación Web para proporcionar servicios de búsqueda para los usuarios.

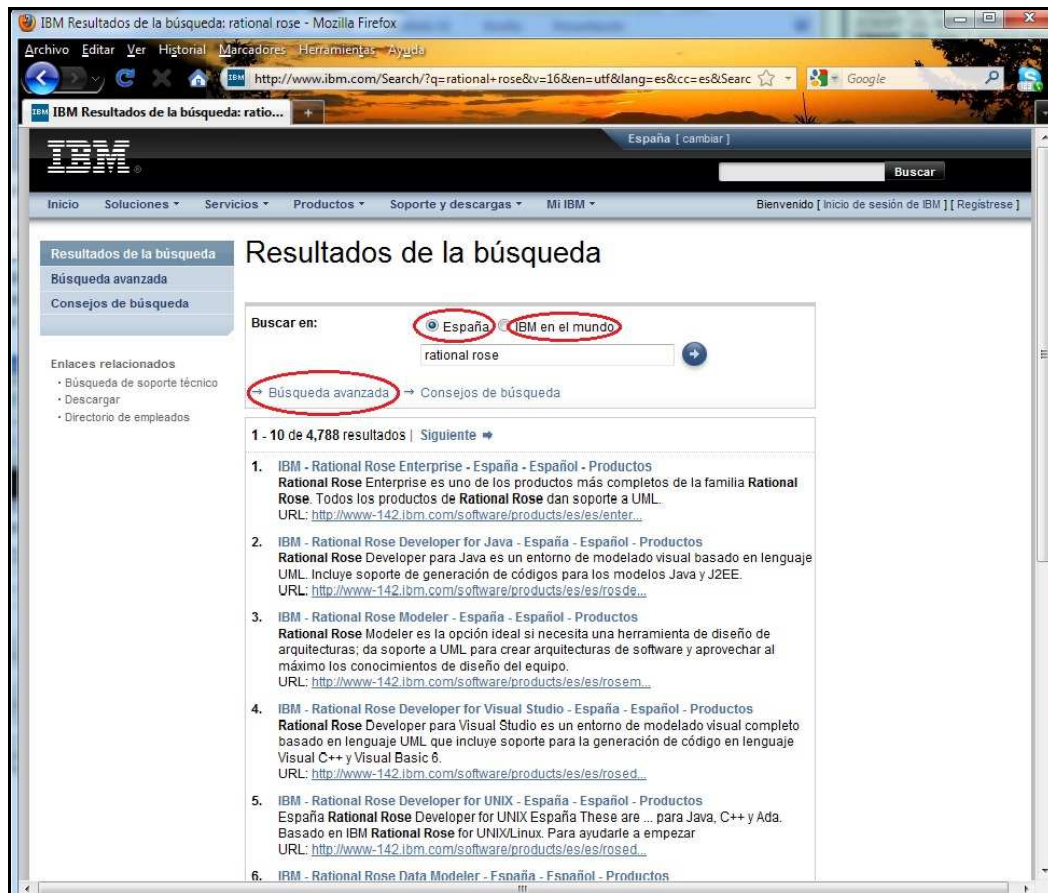


Figura 12. Funcionalidad Web: Capacidad de búsqueda.

La Figura 13 ejemplifica la funcionalidad “*Comunicación y colaboración*”, en la cual se muestran algunos elementos que facilitan la discusión entre los usuarios, además de la identificación de ideas y soluciones.

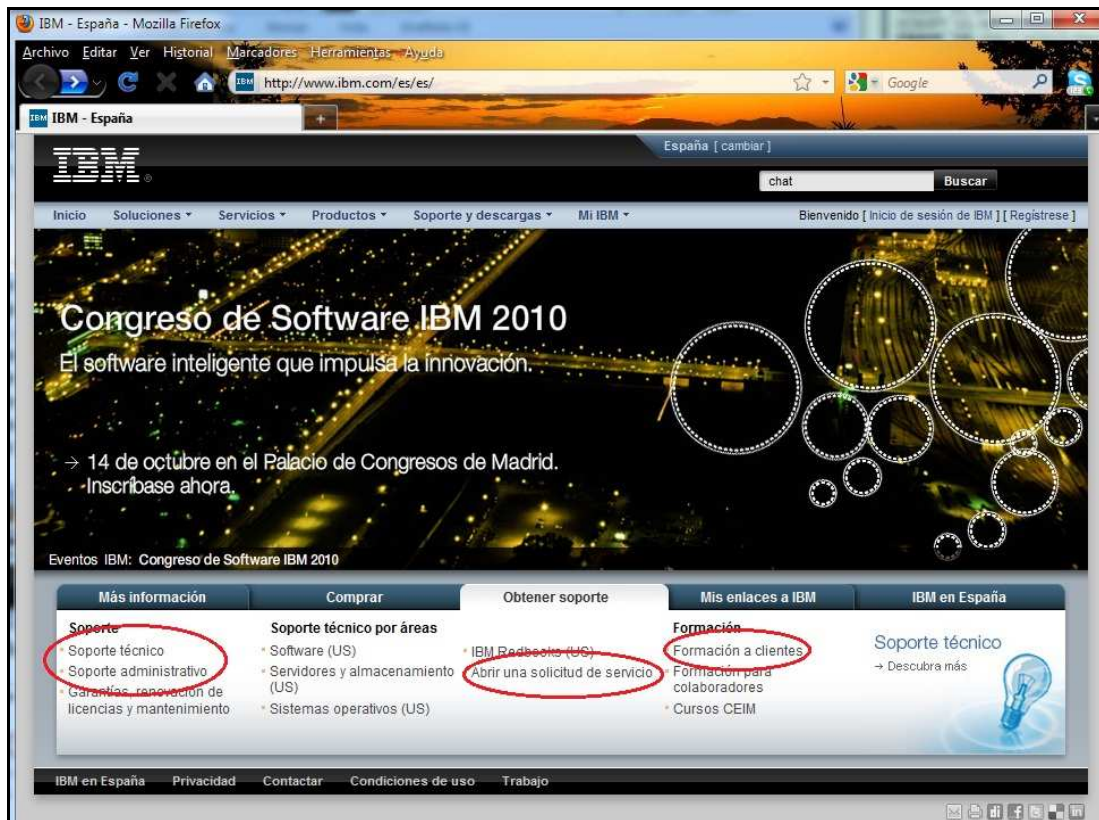


Figura 13. Funcionalidad Web: Comunicación y colaboración.

Finalmente, la Figura 14 muestra la funcionalidad Web de “Personalización” que contiene ciertos elementos que ayudan a mantener un ambiente de trabajo organizado y configurado para cada tipo de usuario.

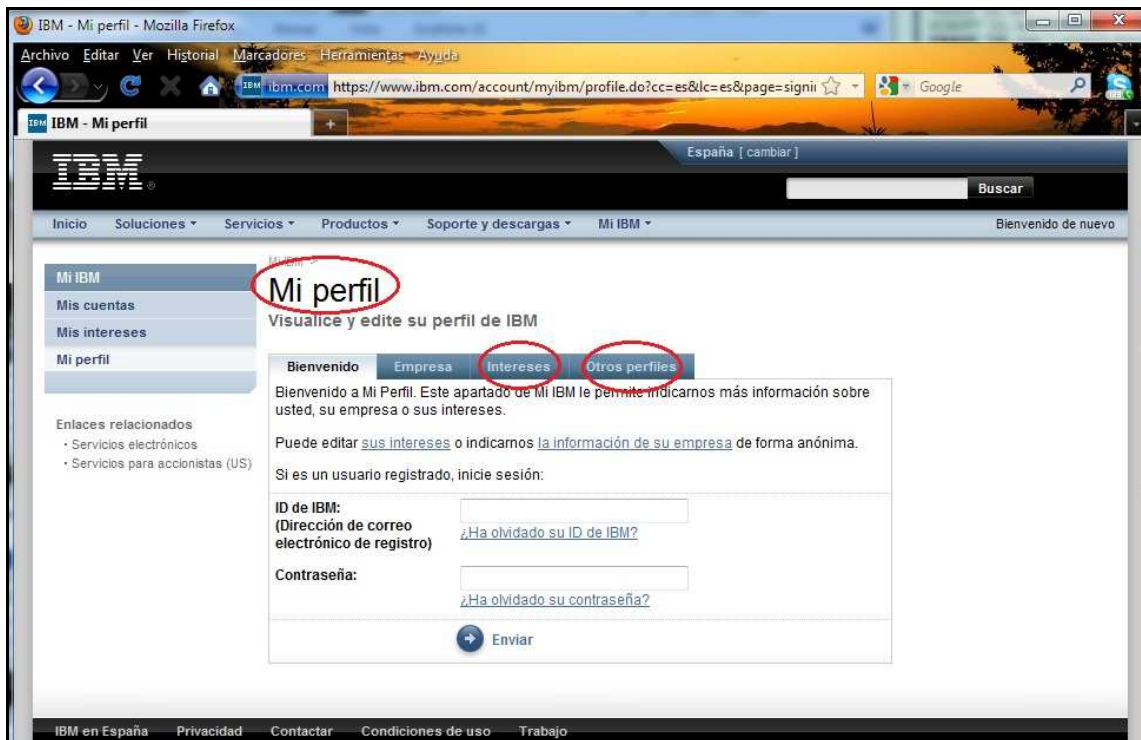


Figura 14. Funcionalidad Web: Personalización.

3.2.5.2 MOSCAF - Modelo para la Selección de Requisitos de DQ según las Funcionalidades a Implementar en una Aplicación Web

Una vez descrito el significado del concepto de Requisito de Calidad de Datos (*“la especificación de un conjunto de dimensiones de calidad de datos que un conjunto de datos debería cumplir para una tarea específica desempeñada por un determinado usuario”*) además de cada una de las dimensiones de DQ (Tabla 1), y listado las principales funcionalidades de una aplicación Web. El siguiente paso de la investigación estuvo enfocado en realizar un análisis acerca de qué dimensiones de calidad de datos pueden formar parte de un requisito de DQ al momento de implementar una funcionalidad determinada. En este sentido, sería posible asegurar que los datos que serán usados por las funcionalidades tienen un nivel de calidad aceptable para cada usuario de la aplicación para cada tarea.

Una vista completa de los análisis realizados para obtener el modelo MOSCAF se muestra en la Figura 15. Posteriormente se describe a detalle cada uno de ellos.

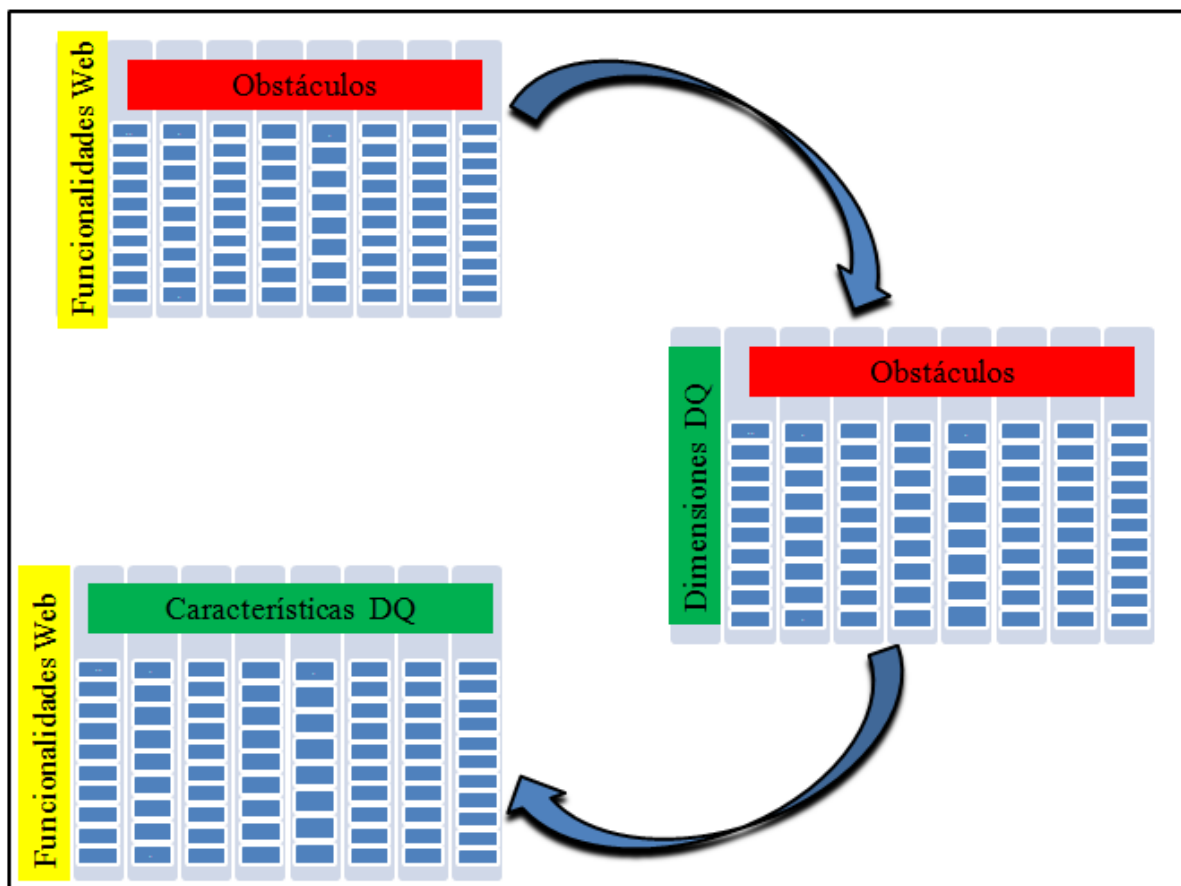


Figura 15. Análisis realizados para obtener el modelo MOSCAF.

Como parte fundamental de la investigación, se llevó a cabo un análisis acerca de qué *problemas u obstáculos* que normalmente se presentan en un SI

(definidos por Strong et al. en (Strong, Lee et al. 1997)) pudieran estar en un momento dado inmiscuidos o relacionados con cada una de las *funcionalidades* de una aplicación Web, como resultado de esta fase inicial se obtuvo la siguiente matriz de relaciones mostrada en la Tabla 13.

Obstáculos Funcionalidades del portal web	Múltiples fuentes	Producción subjetiva	Errores en la producción	Demasiada información	Sistemas distribuidos	Información no numérica	Análisis automatizado	Necesidad de cambio en datos	Requisitos de seguridad	Falta de recursos
Gestión de contenido	√	√	√	√		√	√	√	√	√
Procesos y Acciones				√				√		
Capacidades de búsqueda				√	√	√			√	√
Administración								√		
Seguridad									√	
Integración y puntos de datos				√	√				√	
Comunicación y Colaboración				√	√		√			√
Presentación								√		
Taxonomía					√					
Personalización								√	√	
Características de Ayuda				√						

Tabla 13. Matriz de relaciones de funcionalidades del portal y obstáculos identificados.

La Tabla 13 resume el proceso de análisis llevado a cabo y que se detalla en el Anexo C.

Teniendo en cuenta que los dos modelos considerados como estándares, el propuesto por Wang y Strong (Wang y Strong 1996) vs el estándar ISO/IEC 25012 (ISO-25012 2008), introducen distintos significados para las características de calidad de datos, merece la pena realizar un esfuerzo en trazar la analogía entre los significados de la dimensión frente a la característica limitando el alcance de cada uno de ellos. Para ello se estudió el significado de cada dimensión, plasmándose los resultados en la siguiente tabla comparativa (véase Tabla 14).

El objetivo de esta comparación fue resolver posibles conflictos en la descripción de las diferentes dimensiones de DQ, ya sea la existencia de dimensiones/características con el mismo nombre y diferente significado, o dimensiones con diferente nombre pero el mismo significado.

Modelo Wang & Strong	Estándar ISO/IEC 25012
Exactitud	Exactitud
Compleitud	Compleitud
Representación concisa	Compleitud
Representación consistente	Consistencia
Objetividad	Consistencia
Confiabilidad	Credibilidad
Reputación	Credibilidad
Puntualidad	Actualidad
Accesibilidad	Accesibilidad
Valor-agregado	Conformidad
Seguridad	Confidencialidad
Cantidad de información	Eficiencia
Cantidad de información	Precisión
Trazabilidad	Trazabilidad
Fácil de entender	Entendibilidad
Interpretabilidad	Entendibilidad
Variedad de recursos de datos	Disponibilidad
Facilidad de operación	Portabilidad
Flexibilidad	Portabilidad
	Recuperabilidad

Tabla 14. Comparativa de dimensiones/características de DQ.

En la parte final del análisis realizado, la idea es describir qué dimensiones de calidad de datos forman cada uno de los requisitos específicos que se pueden mostrar. Ello requería hacer un análisis que identificara para las funcionalidades Web las dimensiones de calidad de datos a partir de los mostrados en la Tabla 13. Como punto de partida, se tomó nuevamente como referencia el trabajo presentado por Strong et al. en (Strong, Lee et al. 1997), en donde se clasifican en base a su modelo (Wang y Strong 1996) las dimensiones de DQ que afectan a cada uno de los obstáculos (ver Tabla 15).

A continuación, y teniendo en cuenta que el estándar ISO 25012 es más apropiado para el trabajo que se está desarrollando, (debido a que éste es un estándar internacional aceptado por la comunidad del área de DQ, además de que las aplicaciones Web pueden considerarse Sistemas de Información), manteniendo el significado de la dimensión, pero teniendo en cuenta el cambio de alcance descrito en la Tabla 14, se reescribió la Tabla 15 obteniendo la matriz presentada en la Tabla 16 donde se pone de manifiesto la relación (señalada por el símbolo “√”) entre las funcionalidades de un portal Web y las dimensiones de DQ identificadas por el estándar internacional ISO/IEC 25012.

Obstáculos Dimensiones	Múltiples fuentes	Producción subjetiva	Errores en la producción	Demasiada información	Sistemas distribuidos	Información no numérica	Análisis automatizado	Necesidad por cambio en datos	Requerimientos de seguridad y privacidad	Falta de recursos computacionales
Consistencia	x									
Confiabilidad	x	x								
Objetividad		x								
Exactitud			x							
Complejidad			x					x		
Relevancia			x				x	x		
Representación concisa				x		x				
Puntualidad				x	x					
Valor agregado				x	x	x	x	x	X	x
Accesibilidad				x		x			X	x
Representación Consistente					x		x			
Requisitos de Análisis							x			
Seguridad									X	

Tabla 15. Dimensiones de DQ que son afectadas por cada uno de los obstáculos.

Con el objetivo de complementar los resultados obtenidos, se decidió realizar un análisis más exhaustivo acerca de qué otras dimensiones pudieran estar en un momento dado presuntamente relacionadas con alguna otra de las funcionalidades, obteniendo de esta manera un conjunto de nuevas relaciones (mostradas en Tabla 16, con el símbolo “α”).

En la Tabla 16 se muestra el resultado final del **Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web (MOSCAF)**, a través del mapeo de relaciones entre las características de DQ y las funcionalidades a implementar en la aplicación Web.

En este sentido se puede decir que un requisito de DQ podrá ser especificado como un subconjunto de cada una de las filas de la matriz para cada funcionalidad requerida a ser implementada en la aplicación Web.

Características de DQ Funcionalidades de la aplicación web	Exactitud	Compleitud	Consistencia	Credibilidad	Actualidad	Accesibilidad	Conformidad	Confidencialidad	Eficiencia	Precisión	Trazabilidad	Entendibilidad	Disponibilidad	Portabilidad	Recuperabilidad
Gestión de contenido	α	√	√	√	√	√	√	√			α	α		α	
Procesos y acciones		√			√	√	√								
Capacidades de búsqueda		√	√		√	√	√	√			α		α		
Administración		√					√		α	α		α		α	α
Seguridad						√	√	√			α				
Integración y puntos de datos		√	√		√	√	√	√					α		
Comunicación y Colaboración	α	√	√	α	√	√	√						α		
Presentación		√					√								
Taxonomía	α		√		√		√								
Personalización		√				√	√	√					α		
Características de ayuda		√			√	√	√					α			

Tabla 16. Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web - MOSCAF.

De esta forma se puede observar, por ejemplo, que cuando una aplicación desea implementar la funcionalidad Web “*Gestión de contenido*”, y con el fin de evitar los problemas que podrían generar los obstáculos identificados, la aplicación Web debería implementar mecanismos para asegurar que los datos que se usan en dicha funcionalidad cumplan con las dimensiones de Exactitud, Compleitud, Consistencia, Credibilidad, Actualidad, Accesibilidad, Conformidad, Confidencialidad, Trazabilidad, Entendibilidad y Portabilidad. Lo que significaría que si se quisiera implementar esta funcionalidad, en los requisitos de usuario para calidad de datos se debería considerar dichas dimensiones, y que el equipo de desarrollo debería valorar si es o no viable dotar a la aplicación Web de los mecanismos necesarios para garantizarlas.

Para un mayor entendimiento del modelo (MOSCAF), en las siguientes Tablas (17-27) se describen cada una de las distintas relaciones (“√”, “α”) especificadas:

Gestión de Contenido	
Exactitud	Los datos gestionados para su inclusión en la aplicación deberán representar un valor correcto de acuerdo al contexto específico de uso.
Completitud	Todos los datos gestionados dentro de la aplicación deberán estar completos en cada uno de sus atributos.
Consistencia	Todos los datos gestionados en la aplicación deberán ser coherentes en un mismo ambiente de uso.
Credibilidad	Los datos gestionados en la aplicación deberán ser creíbles para los usuarios.
Actualidad	Los datos gestionados en la aplicación deberán estar actualizados de acuerdo al contexto de su utilización para cada usuario.
Accesibilidad	Los datos gestionados en la aplicación deberán siempre estar accesibles para los usuarios.
Conformidad	Los datos gestionados en la aplicación deberán adherirse a las normas o regulaciones especificadas por el encargado de la gestión del contenido.
Confidencialidad	Los datos deberían ser clasificados para su gestión en distintos niveles de confidencialidad, asegurando que sólo son accesibles e interpretables por usuarios autorizados.
Trazabilidad	Como parte de la gestión de los datos, se deberá proporcionar información de cuándo y quién publicó o distribuyó éstos y quién los podrá acceder.
Entendibilidad	Los datos deberían ser gestionados en el lenguaje apropiado, utilizando los símbolos o unidades convenientes para ser entendidos por cada usuario.
Portabilidad	Los datos podrán ser instalados o movidos dentro de cualquier otra aplicación en la organización.

Tabla 17. Gestión de Contenido vs dimensiones de DQ.

Procesos y Acciones	
Completitud	Todos los datos deberán estar completos para que los usuarios puedan llevar a cabo un proceso de negocio con la organización.
Actualidad	Los datos deberán ser actuales para que un usuario inicie y participe en un proceso de negocio.
Accesibilidad	El usuario podrá iniciar y participar en un proceso de negocio una vez que los datos estén accesibles para su uso.
Conformidad	Los datos deberán cumplir con ciertas normas o convenciones vigentes, de acuerdo al contexto en que el usuario los utilizará para poder participar en un proceso de negocio.

Tabla 18. Procesos y Acciones vs dimensiones de DQ.

Capacidades de búsqueda	
Completitud	Los datos obtenidos al realizar una búsqueda deberán estar completos con los valores para todos sus atributos esperados.
Consistencia	Los datos obtenidos al efectuar una búsqueda deberán ser coherentes con otros datos en un contexto de uso definido.
Actualidad	Los datos obtenidos al efectuar una búsqueda deberán ser actuales para ser utilizados en un contexto específico.
Accesibilidad	Los datos deberían estar accesibles en la aplicación al momento de realizar una búsqueda.
Conformidad	Los datos conseguidos como resultado de una búsqueda deberían estar acorde a las normas o regulaciones establecidas en los procesos del negocio.
Confidencialidad	El usuario deberá obtener sólo los datos disponibles para él, de acuerdo a su nivel o rol en la aplicación.
Trazabilidad	El sistema debería proporcionar información acerca de la cantidad de veces que ha sido accedido y/o modificado un determinado dato o conjunto de datos.
Disponibilidad	Los datos deberían poder ser recuperados al momento de ejecutar algún tipo de búsqueda específica. Ello implica que el dato esté accesible, que los caminos de acceso estén disponibles y que el usuario tenga los permisos adecuados para su lectura.

Tabla 19. Capacidades de búsqueda vs dimensiones de DQ.

Administración	
Compleitud	El administrador de la aplicación deberá asegurar que los datos gestionados en la aplicación permanecen completos, una vez habiendo realizado alguna actividad de mantenimiento y/o tareas relacionadas con la aplicación.
Conformidad	Los datos deberán adherirse a normas o convenciones definidas por la organización o el Administrador de la aplicación.
Eficiencia	El administrador de la aplicación deberá asegurar que los datos pueden ser accedidos con un nivel aceptable de funcionamiento.
Precisión	El administrador de la aplicación deberá asegurar que los datos están definidos acorde a la precisión o exactitud requerida para tarea específica.
Entendibilidad	El administrador deberá mantener los datos en el lenguaje, símbolo y unidad apropiada para cada funcionalidad o tarea de la aplicación.
Portabilidad	Los datos podrán ser instalados o substituidos por el Administrador del portal durante actividades de mantenimiento.
Recuperabilidad	Los datos podrán ser mantenidos y recuperados por el Administrador del portal durante actividades de mantenimiento, o en caso de algún tipo de falla del sistema.

Tabla 20. Administración vs dimensiones de DQ.

Seguridad	
Accesibilidad	Los datos deberían ser clasificados de acuerdo a distintos niveles de acceso, para ser accesibles a cada tipo de usuario.
Conformidad	Los datos deberían estar acorde a normas o convenciones y de esta manera mantener la política de seguridad en ellos.
Confidencialidad	Los datos serán sólo asequibles por usuarios autorizados de acuerdo a su nivel de acceso.
Trazabilidad	Los datos deberían proporcionar información de cuándo y quién lo utilizó.

Tabla 21. Seguridad vs dimensiones de DQ.

Integración y puntos de datos	
Compleitud	Todos los datos deberán estar completos para cada uno de sus atributos, independientemente de la fuente de información de donde se obtiene.
Consistencia	Los datos deberán ser consistentes sin importar la fuente de información de donde se adquiere.
Actualidad	Los datos deberían estar actualizados sin importar la fuente de información de donde son obtenidos.
Accesibilidad	Los datos deberían estar siempre accesibles al usuario, preferentemente como un solo punto de acceso en la aplicación.
Conformidad	Los datos deberán estar de acuerdo a normas o regulaciones, independientemente de la fuente de información de donde se obtienen.
Confidencialidad	Los datos sólo deberían ser asequibles por usuarios autorizados, sin importar la fuente de donde provienen.
Disponibilidad	Los datos deberían de estar siempre al alcance de los usuarios en el momento que sean requeridos.

Tabla 22. Integración y Puntos de Datos vs dimensiones de DQ.

Comunicación y Colaboración	
Exactitud	Los datos deberán representar un valor correcto en el mismo contexto por todos los usuarios, mejorando con esto la comunicación y el desempeño de ellos.
Compleitud	Los datos deberían estar completos en todos sus atributos esperados, ayudando con ello en la comunicación y colaboración al momento de realizar sus tareas los usuarios.
Consistencia	Los datos deberían no ser contradictorios para los usuarios al momento de ejecutar alguna actividad, generando una buena colaboración entre ellos.
Credibilidad	Los datos deberán ser considerados como verídicos por todos los usuarios, mejorando la colaboración entre cada uno de ellos.
Actualidad	Los datos deberán estar actualizados, ayudando con ello a realizar soluciones ingeniosas e innovadoras de los procesos de negocio por parte de los usuarios.
Accesibilidad	Los datos deberán ser accedidos en todo momento, permitiendo mejorar la comunicación y colaboración de todos los usuarios.
Conformidad	Los datos deberán adecuarse a regulaciones vigentes en la organización, generando una mejor comunicación entre los distintos usuarios.
Disponibilidad	Los datos deberán estar disponibles para los usuarios autorizados, ayudándolos a discutir y solucionar problemas, así como la toma de decisiones.

Tabla 23. Comunicación y Colaboración vs dimensiones de DQ.

Presentación	
Compleitud	Todos los datos deberán mostrarse al usuario de manera completa en cada una de las instancias de presentación.
Conformidad	Los datos mostrados a los usuarios deberían cumplir con las normas y regulaciones especificadas en la organización.

Tabla 24. Presentación vs dimensiones de DQ.

Taxonomía	
Exactitud	Los datos intrínsecamente deberían de ser sintáctica y semánticamente correctos, mejorando la terminología común utilizada en la organización.
Consistencia	Los datos deberán ser mostrados de forma consistente en cada categoría específica de la organización, mejorando con esto en una terminología común para los usuarios de la aplicación.
Actualidad	Los datos deberán ser del período correcto, contribuyendo a mejorar los negocios de la empresa en cada una de sus categorías específicas.
Conformidad	Los datos deben cumplir con ciertas normas específicas definidas por la organización, contribuyendo hacia una terminología común y mejorando la semántica y el reconocimiento rápido entre los usuarios.

Tabla 25. Taxonomía vs dimensiones de DQ.

Personalización	
Compleitud	Todos los datos deberán visualizarse de forma completa, con el objetivo de mantener un ambiente de trabajo organizado a los usuarios. Contribuyendo a mantener una consistencia en la personalización de la aplicación para cada usuario de la aplicación.
Accesibilidad	Los datos podrán ser accedidos y configurados especialmente para cada usuario, colaborando a alcanzar un balance en el ambiente de trabajo de la aplicación.
Conformidad	Los datos deberán estar de acuerdo con ciertas regulaciones establecidas en la organización y aplicadas en la personalización y las características únicas requeridas por cada usuario de la aplicación.
Confidencialidad	Los datos sólo serán presentados y configurados para cada ambiente de trabajo de cada usuario con autorización.
Disponibilidad	Los datos deberán estar disponibles para los usuarios autorizados, creando un ambiente de trabajo configurado especialmente para cada usuario.

Tabla 26. Personalización vs dimensiones de DQ.

Características de Ayuda	
Compleitud	Todos los datos mostrados al usuario para asistirlo en las características propias de la aplicación deberían ser completos en el contexto específico de uso.
Actualidad	Los datos presentados al usuario como ayuda para la realización de alguna tarea específica deberían ser actuales.
Accesibilidad	Los datos deberían siempre estar disponibles para ser expuestos a los usuarios como ayuda en alguna actividad específica de la aplicación o de la organización.
Conformidad	Los datos exhibidos como ayuda a los usuarios deberían mantener las convenciones vigentes establecidas por la organización.
Entendibilidad	Los datos mostrados al usuario como asistencia del portal deberían ser perfectamente leídos e interpretados en el lenguaje apropiado.

Tabla 27. Características de Ayuda vs dimensiones de DQ.

Una vez mostrado el *Modelo para la selección de Requisitos de calidad de datos según las funcionalidades a implementar en una aplicación Web - MOSCAF* (Tabla 16), enseguida se muestra la guía de uso.

3.2.5.3 Guía de Uso del Modelo MOSCAF

La guía para el uso del modelo toma como referencia el trabajo de Franch y Carvalho (Franch y Carvalho 2003). Comprende 4 pasos (véase Figura 16) descritos a continuación:

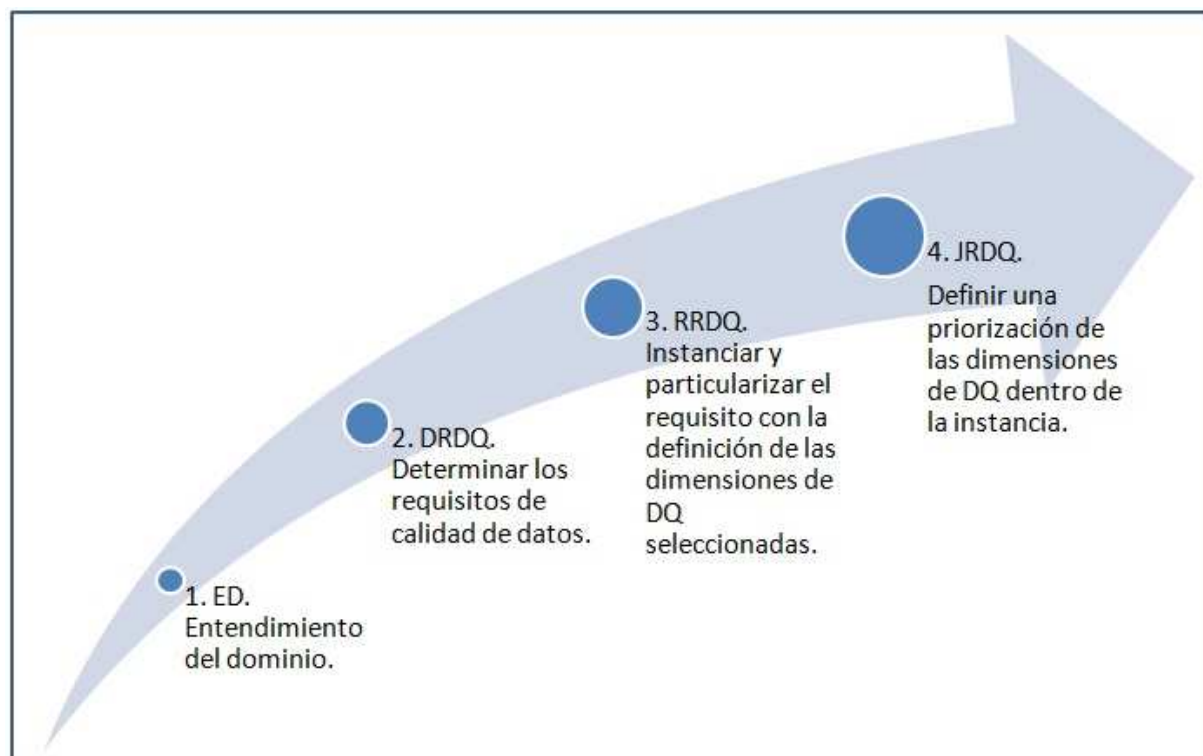


Figura 16. Pasos de la guía de uso del modelo MOSCAF.

Paso 1. ED. Entendimiento del dominio.

Primero, es indispensable describir el dominio de la aplicación, y el contexto en el que el usuario ejecutará la correspondiente funcionalidad de la aplicación Web. Todos los involucrados deben tener un pleno conocimiento tanto de las funcionalidades Web como de las dimensiones de DQ, es decir, deberían conocer y entender el significado de cada una de ellas.

El objetivo es generar como producto de salida un documento con la descripción completa y detallada del dominio o ambiente en el cual se generará la aplicación (DED) (véase Tabla 28). La descripción detallada de los roles involucrados en cada uno de los pasos se muestra en la sección 3.2.2.

Producto de Entrada	Producto de Salida	Rol involucrado
- Modelo del dominio de la aplicación Web (el cual abarca tanto las dimensiones de DQ como las distintas funcionalidades Web).	- Documento con la descripción del contexto de aplicación (DED).	- Jefe de desarrollo. - Usuarios. - Analista de Calidad de datos. - Arquitecto Jefe.

Tabla 28. Productos de E/S del paso 1 "Entendimiento del dominio".

Paso 2. DRDQ. Determinar los requisitos de calidad de datos.

Este paso consiste en la selección de las dimensiones de DQ elegidas al momento de implementar cada una de las funcionalidades Web. Esto pudiera ser a partir de una observación propuesta por una persona experta en el dominio de la aplicación.

Como producto de entrada se debería tener tanto el modelo MOSCAF como el documento con la descripción del dominio de la aplicación (DED), obteniendo como producto de salida un documento con los requisitos de DQ necesarios (DRDQ), es decir, la selección de las dimensiones de DQ seleccionadas para cada funcionalidad (véase Tabla 29).

Producto de Entrada	Producto de Salida	Rol involucrado
- Modelo MOSCAF. - Documento con la descripción del ambiente de aplicación (DED).	- Documento con los requisitos de DQ necesarios (DRDQ) para la funcionalidad.	- Analista de Calidad de datos. - Usuarios.

Tabla 29. Productos de E/S del paso 2 "Determinar los requisitos de DQ".

Paso 3. RRDQ. Instanciar y particularizar el requisito con la definición de las dimensiones de DQ seleccionadas.

En este paso es necesario particularizar la definición de las distintas dimensiones de DQ seleccionadas de acuerdo al contexto en que serán implementadas como parte de las diferentes funcionalidades Web, el objetivo es definir de forma clara el significado de cada dimensión de DQ de acuerdo al

contexto de cada funcionalidad, de tal forma que todos los involucrados entiendan el significado y estén de acuerdo con él.

El producto de entrada de este paso es un documento con los requisitos de DQ (DRDQ), obteniendo como producto de salida un documento extendido con la definición de cada dimensión seleccionada (DDRDQ) (véase Tabla 30) para cada funcionalidad según las necesidades de cada usuario.

Producto de Entrada	Producto de Salida	Rol involucrado
- Documento con los requisitos de DQ necesarios (DRDQ).	- Documento refinado con la definición de cada dimensión seleccionada (DDRDQ).	- Jefe de desarrollo. - Analista de Calidad de datos. - Arquitecto Jefe.

Tabla 30. Productos de E/S del paso 3 "Instanciar y particularizar el requisito con la definición de las dimensiones de DQ seleccionadas".

Paso 4. JRDQ. Definir una priorización de las dimensiones de DQ dentro de la instancia.

Dado que no todas las dimensiones de calidad de datos elegidas pueden tener la misma importancia, en este paso es posible definir una priorización, de acuerdo al grado de importancia de las distintas dimensiones de DQ que fueron seleccionadas previamente, considerando los distintos comentarios de expertos involucrados en la definición. Para ayudarnos en esta tarea, es posible llevar a cabo entrevistas con los expertos y usuarios de la aplicación.

El principal producto de entrada es el documento generado en el paso anterior (DDRDQ), obteniendo a su vez como producto de salida un documento especificando la priorización de dichas dimensiones de DQ (DJRDQ) (véase Tabla 31).

Producto de Entrada	Producto de Salida	Rol involucrado
- Documento refinado con la definición de cada dimensión seleccionada (DDRDQ).	- Documento especificando la jerarquización (priorización) de las dimensiones de DQ (DJRDQ).	- Jefe de desarrollo. - Analista de Calidad de datos.

Tabla 31. Productos de E/S del paso 4 "Definir una priorización de las dimensiones de DQ dentro de la instancia".

Con este modelo, se ha pretendido facilitar la identificación y elección de los requisitos de calidad de datos para una aplicación Web. Puede entenderse como *un camino que los analistas pueden seguir para escribir un Documento de Especificación de Requisitos complementado con gestión de DQ, siempre teniendo en mente las dimensiones de DQ que deberían ser implementadas para cada funcionalidad durante el proceso de desarrollo de aplicaciones Web.*

3.2.6 Metamodelo Ampliado para la Especificación de Requisitos de DQ en Aplicaciones Web (DQ_WebRE)

Una vez mostrado la propuesta principal enfocada en la parte *metodológica* de este trabajo de tesis, la metodología **DQ-VORD**. En las secciones 3.2.6 y 3.2.7 se describen las propuestas relacionadas con el enfoque tecnológico, para de esta forma cumplir con los objetivos parciales O4 y O5, previamente establecidos. Dichas propuestas forman parte fundamental de las actividades 3 y 4 (“*DP. Documentación de los Puntos de Vista*” y “*TPS. Trazado de los Puntos de Vista del Sistema*”) de la metodología DQ-VORD.

Las dos propuestas, como parte del **enfoque tecnológico** son:

- ✓ Metamodelo Ampliado para la Especificación de Requisitos de DQ en Aplicaciones Web (DQ_WebRE).
- ✓ Metamodelo para el Diseño de Aplicaciones Web Conscientes de DQ (DAQ_UWE).

Con ambas propuestas de metamodelos se cumple el objetivo principal de este trabajo de tesis definido inicialmente como “*Definir un ambiente de trabajo que permita la incorporación de requisitos para la gestión de la calidad de datos en aplicaciones Web a través de un entorno de trabajo que incluya tanto aspectos metodológicos como tecnológicos*”.

3.2.6.1 Contexto y Motivación.

Una de las motivaciones más importantes de este trabajo, es la necesidad de dotar a los analistas y desarrolladores de aplicaciones Web, de las herramientas necesarias para dar soporte al desarrollo de aplicaciones Web con aspectos de calidad de datos, haciendo especial énfasis en cómo especificar e implementar de forma clara e intuitiva los requisitos software de DQ mediante distintos elementos de modelado. Para capturar todo el conocimiento relacionado, se consideró imprescindible elaborar un metamodelo.

Como se mencionó previamente (en sección 2.2.1), las principales metodologías de Ingeniería Web dirigida por Modelos se centran en los aspectos funcionales (Moreno, Romero et al. 2008). Pero ninguna de estas propuestas ha tratado de integrar en sus métodos o herramientas una consideración explícita las características de calidad en los datos usados por estas aplicaciones.

Algunas de las propuestas como (Ceri, Fraternali et al. 2000; Escalona y Koch

2006; Escalona y Aragón 2008) mencionan ciertos objetivos específicos de información que deberían ser considerados al momento de diseñar la aplicación Web, sin embargo, no profundizan en su estudio ni consideran ningún tipo de requisito o especificación de características de DQ.

Habiendo hecho un análisis a fondo de las diferentes propuestas, se decidió tomar como base el trabajo realizado por Escalona y Koch - **WebRE** (Escalona y Koch 2006), en el cual se propone un metamodelo para la representación de conceptos y relaciones de *Ingeniería de Requisitos Web*.

Una de las razones por la que se optó considerar esta propuesta como plataforma para la incursión de aspectos de calidad de datos, fue la flexibilidad que provee su propuesta de metamodelo y el perfil para la especificación de requisitos Web, lo que nos permite de alguna manera realizar una especificación y/o modelado de requisitos de DQ específicos necesarios y relacionarlos a cada uno de los elementos Web mencionados en el metamodelo. Cabe mencionar que *WebRE* fue concebido a partir de un análisis exhaustivo de diferentes propuestas Web que tratan con requisitos (Escalona y Koch 2006), uniendo criterios y definiendo un metamodelo unificado, ajustándose a estándares como MDA, UML y OCL.

3.2.6.2 Metamodelo WebRE

Este metamodelo propuesto por Escalona (Escalona y Koch 2006) (ver Figura 17) muestra los elementos principales para el modelado de requisitos Web mediante un diagrama de clases. Las metaclases representan los conceptos sin ninguna información acerca de su representación, y son agrupadas en dos paquetes siguiendo la estructura del metamodelo de UML “*WebRE Structure*” y “*WebRE Behavior*”.

La funcionalidad de un sistema Web (*WebRE Behavior*) es modelada por un conjunto de instancias de dos tipos de casos de uso específicos “*Navigation*” y “*WebProcess*”, y actividades específicas como “*Browse*”, “*Search*” y “*UserTransaction*”.

El segundo paquete del metamodelo “*Structure Package*”, contiene las metaclases usadas para describir los elementos estructurales del sistema Web: contenido (*Content*), nodo (*Node*) e interfaz de usuario Web (*WebUI*).

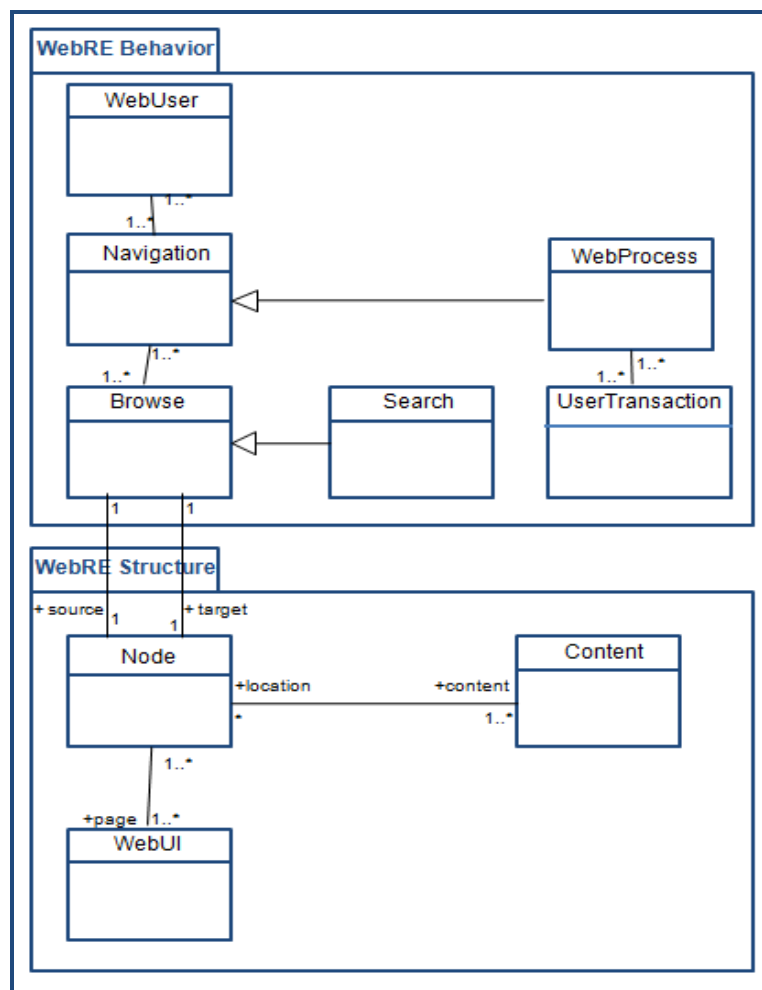


Figura 17. Metamodelo WebRE.

Una breve descripción de cada elemento del metamodelo se muestra en la Tabla 32.

Elemento	Descripción
WebUser	Representa cualquier usuario que interactúa con la aplicación Web.
Navigation	Representa un caso de uso específico, el cual incluye un conjunto de actividades de tipo “Browse” que el usuario web será capaz de desempeñar para alcanzar un nodo destino.
WebProcess	Modela las funcionalidades principales (normalmente procesos de negocio) de la aplicación Web. Representa otro caso de uso el cual puede ser refinado por diferentes actividades de tipo <i>Browse</i> , <i>Search</i> y <i>UserTransaction</i> .
Browse	Representa una actividad normal de navegación dentro del sistema, la cual puede ser mejorada por una actividad de búsqueda “Search”.
Search	Esta actividad tiene un conjunto de parámetros, la cual permite realizar consultas sobre los datos almacenados en la metaclass “Content”. Los resultados pueden ser mostrados en el nodo destino.
UserTransaction	Representa actividades más complejas que pueden ser expresadas en términos de transacciones iniciadas por los usuarios.
Node	Representa un punto de navegación donde el usuario puede encontrar información. Cada instancia de una actividad <i>Browse</i> inicia en un nodo (origen) y termina en otro nodo (destino). Los nodos se muestran al usuario como páginas Web.
Content	Representa el lugar donde se almacenan las piezas de información.
WebUI	Representa el concepto de página Web.

Tabla 32. Elementos del metamodelo WebRE.

3.2.6.3 Metamodelo Extendido para especificar Requisitos de DQ para Aplicaciones Web - DQ_WebRE.

Una vez mostrado en la sección anterior las características y elementos principales del metamodelo tomado como base, en esta sección se describe la propuesta de un metamodelo ampliado (*DQ_WebRE*) para la integración de los elementos considerados básicos para la gestión de requisitos de calidad de datos.

De forma específica la utilización de este metamodelo es fundamental en la actividad 3 (“*DP. Documentación de los Puntos de Vista*”) de la metodología DQ-VORD.

Habiendo realizado previamente una revisión de las principales propuestas para la especificación y modelado de requisitos de DQ presentado en (Guerra-García, Caballero et al. 2010), se decidió incorporar los siguientes elementos clave (Guerra-García, Caballero et al. 2011) etiquetados como “*estereotipos*” (ver Figura 18), los cuales fueron inferidos principalmente de (Caballero, Verbo et al. 2007; Becker, Jaster et al. 2009):

- o Para el paquete conductual (*Behavior*): “*DQRequirement*”, “*InformationCase*”, “*DQ_Req_Specification*” y “*Add_DQ_Metadata*”.
- o Para el paquete estructural (*Structure*): “*DQ_Validator*”, “*DQConstraint*” y “*DQ_Metadata*”.

Teniendo en cuenta el objetivo de modelar requisitos de DQ, se introdujeron estos nuevos elementos, permitiendo a un usuario controlar el nivel de detalle con respecto a los requisitos de calidad de los datos usados en una aplicación Web a desarrollarse.

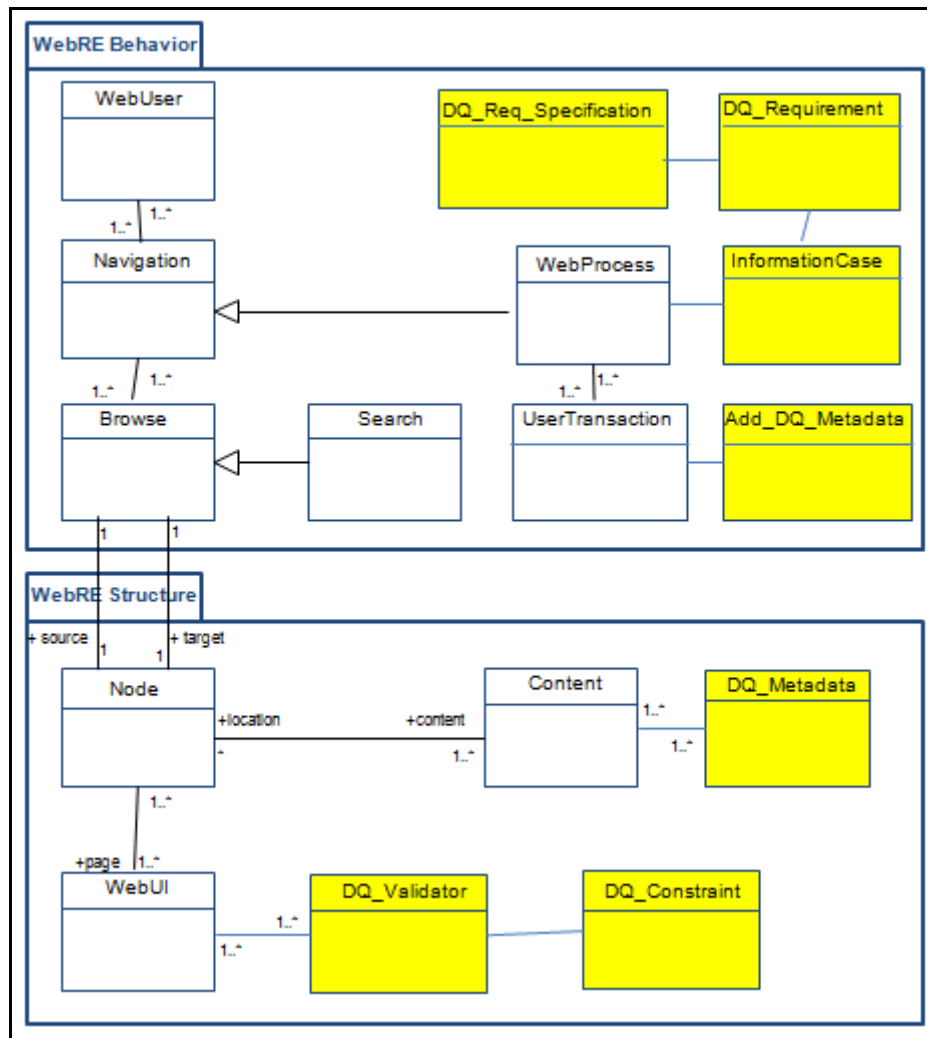


Figura 18. Metamodelo para especificación de requisitos de calidad de datos DQ_WebRE

Con el objetivo de hacer nuestra propuesta operativa, se creó un perfil UML a partir del metamodelo para la especificación de requisitos de calidad de datos. Dicho perfil fue implementado en la herramienta comercial Enterprise Architect (ver Figura 19), lo que proporcionó un ambiente de trabajo adecuado que entre otras cosas, permite realizar diagramas de modelado (*casos de uso y actividades*) con los elementos definidos.

En la parte derecha de la herramienta (ver Figura 20), se puede observar un “*toolbox*” especial con los elementos definidos en el perfil “DQ_WebRE”.

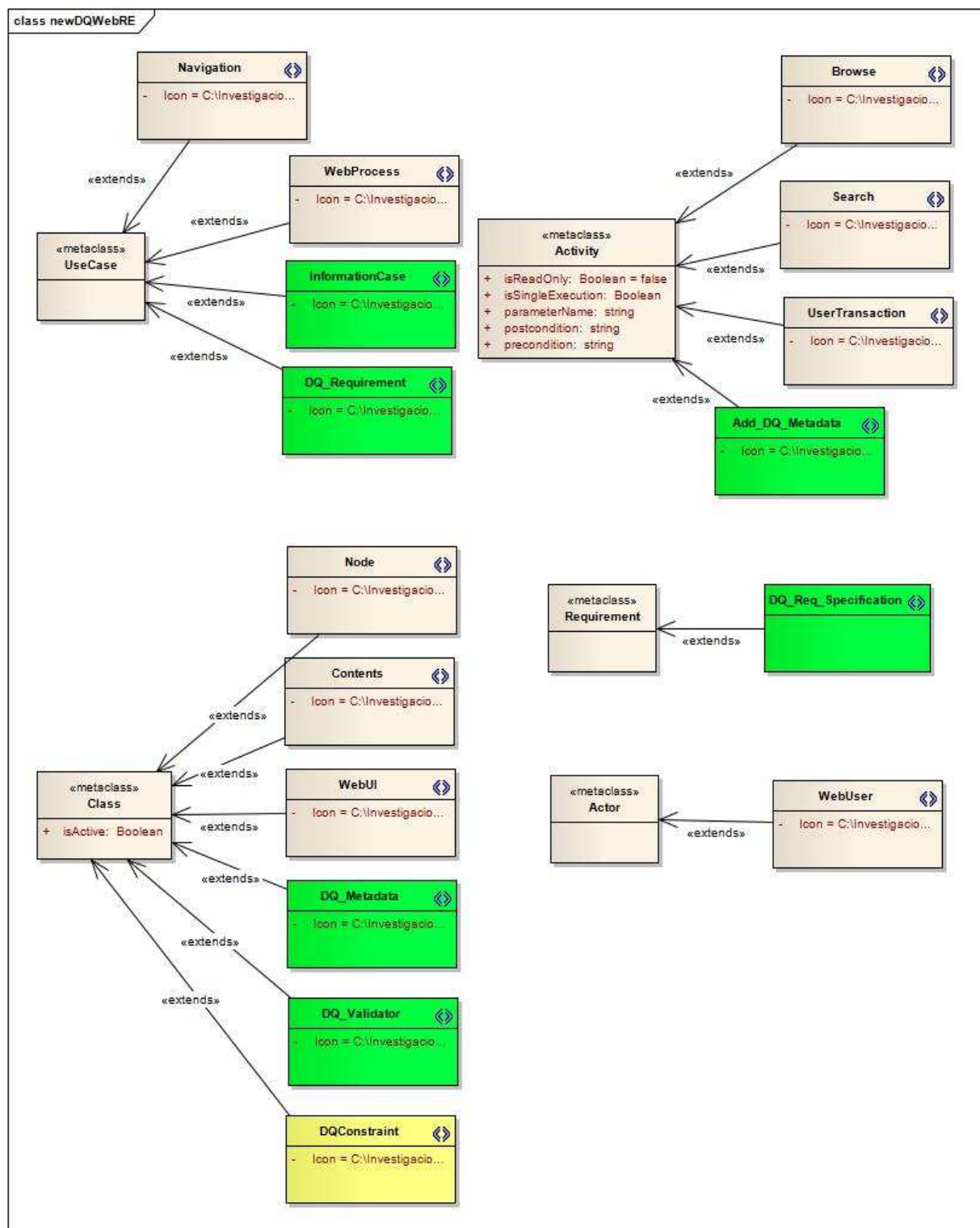


Figura 19. Perfil UML para la especificación de requisitos de calidad de datos - DQ_WebRE

La especificación de cada uno de los nuevos estereotipos propuestos se describe en la Tabla 33.

Nombre	Clase Base	Descripción	Restricciones	Valor etiquetado
InformationCase	UseCase	Los <i>InformationCase</i> (IC), a diferencia de los casos de uso normales, tiene como principal función la de gestionar y almacenar los datos involucrados con las funcionalidades de tipo "WebProcess". Estos datos estarán relacionados a requisitos específicos de DQ (<i>DQ_Requirement</i>), se considera que el mejor camino para relacionarlos es a través de una relación de tipo "include".	Deberá estar relacionado al menos con un elemento de tipo "WebProcess".	Ninguno.
DQ_Requirement	UseCase	Representa para un caso de uso específico, necesario para modelar los requisitos de DQ (dimensiones de DQ) que estarán relacionados a los casos de uso "InformationCase" (IC).	Deberá estar relacionado ("include") al menos con un elemento del tipo "InformationCase".	Ninguno.
DQ_Req_Specification	Element	Clase abstracta que representa un elemento particular (tipo "Requirement"). Esta es usada para especificar en detalle (<i>mediante diagramas de requisitos</i>), cada uno de los requisitos de DQ añadidos.		ID: Integer. Text: String.
Add_DQ_Metadata	Activity	Representa una <i>actividad</i> concreta relacionada a las diferentes actividades "UserTransaction". Esta metaclass se encarga de agregar y/o verificar las operaciones y la información asociada con cada uno de los atributos (metadatos de DQ) pertenecientes a cada una de las instancias de las metaclasses "DQ_Metadata" y "DQ_Validator".	No obligatorio.	None.
DQ_Metadata	Class	Representa un elemento estructural de la aplicación Web. Aquí serán gestionados y almacenados los metadatos de DQ. Estos metadatos estarán asociados a los elementos de tipo "Content". En este sentido, será posible especificar diferentes requisitos de DQ (<i>dimensiones de DQ</i>) ligados directamente a los datos almacenados por los elementos de tipo "Content".	No obligatorio.	DQ_metadata:set (String)
DQ_Validator	Class	Representa un elemento estructural de la aplicación Web, esta metaclass será la responsable de implementar funciones específicas de DQ, con el objetivo de validar o restringir los elementos de interfaz de usuario (WebUI) que proveerá la aplicación.	No obligatorio.	Ninguno.
DQConstraint	Class	Representa un elemento estructural de la aplicación Web. Aquí se almacenan los datos específicos de las distintas restricciones, las cuales estarán relacionadas a elementos del tipo <i>DQ_Validator</i> .	No obligatorio	Ninguno.

Tabla 33. Especificación de estereotipos definidos en el perfil DQ_WebRE.

3.2.6.4 Ejemplo de aplicación usando el perfil DQ_WebRE

En esta sección se demuestra cómo usar el perfil propuesto por medio de un breve ejemplo de aplicación. Como desarrolladores estamos interesados en resaltar como capturar las principales funcionalidades de una aplicación, además de los principales requisitos de DQ para los datos usados durante la ejecución del software.

Teniendo en cuenta el enfoque del Proceso Unificado de Desarrollo (PUD) (Jacobson, Booch et al. 1999), el analista primero modelará los casos de uso principales de la aplicación, para posteriormente, por medio de diagramas de actividades proporcionar una descripción más detallada de cada uno de los casos

de uso identificados.

El ejemplo ilustrativo describe un proceso de negocios típico de una aplicación Web, que permite llevar a cabo la reserva y pago de entradas para conciertos y eventos.

El flujo de eventos es descrito como sigue: el cliente primero será capaz de navegar entre los eventos disponibles, sin embargo, con el objetivo de hacer una reserva, el cliente tendrá que estar registrado para poder ingresar al sistema. Una vez entrado al sistema, el cliente podrá seleccionar el evento y verificar su disponibilidad y costo. Si el cliente está de acuerdo con esos datos, procederá a introducir los datos específicos para la reserva. Una vez hecho esto, el cliente efectuará el pago por la entrada, y el sistema le enviará un billete electrónico por email. Para satisfacer este requisito funcional, el analista introduce el caso de uso *“Hacer reserva de entrada”* (ver Figura 20).

En este diagrama el analista incluye el caso de uso *“Hacer reserva y pago de entradas”* (de tipo *“InformationCase”*), cuyo principal objetivo es gestionar los siguientes datos: *“numero_factura”*, *“id_cliente”*, *“cliente”*, *“dirección”*, *“costo”*, etc. (ver comentario de la Figura 21).

Una vez que los datos han sido identificados, el siguiente paso es capturar e introducir los requisitos de calidad de datos. Es sabido que si la funcionalidad definida por este caso de uso es exitosa, los datos usados tendrán que ser creíbles, exactos, completos y confidenciales.

En este contexto un conjunto de datos puede decirse que es creíble cuando ha sido provista por un usuario autorizado (con un determinado nivel de credibilidad). En este ejemplo no queremos profundizar si un usuario es o no autorizado, por lo que se asume que existe una base de datos con todos los usuarios autorizados.

Debido a que se tiene la necesidad de verificar la credibilidad de los datos (ver Figura 20), la especificación de un requisito específico de DQ tendrá que llevarse a cabo <<DQ_Requirement>> [Credibilidad] *“Comprobar que los datos han sido provistos por un usuario autorizado”*. Por otra parte, con el objetivo de asegurar el nivel de exactitud de una pieza de datos, el caso de uso denominado <<DQ_Requirement>> [Exactitud] *“Asegurar/comprobar que los datos introducidos satisfacen un formato estándar específico”*, deberá ser ejecutado para asegurar que los datos gestionados cumplen este requisito de DQ.

De manera similar, el analista será capaz de garantizar el nivel de confidencialidad de los datos, a través del caso de uso <<DQ_Requirement>>

[Confidencialidad] “Comprobar que los datos son solo mostrados a usuarios permitidos”. Finalmente, mediante el caso de uso denominado <<DQ_Requirement>> [Compleitud] “Verificar que los datos introducidos están completos” el analista podrá garantizar el requisito de DQ relacionado.

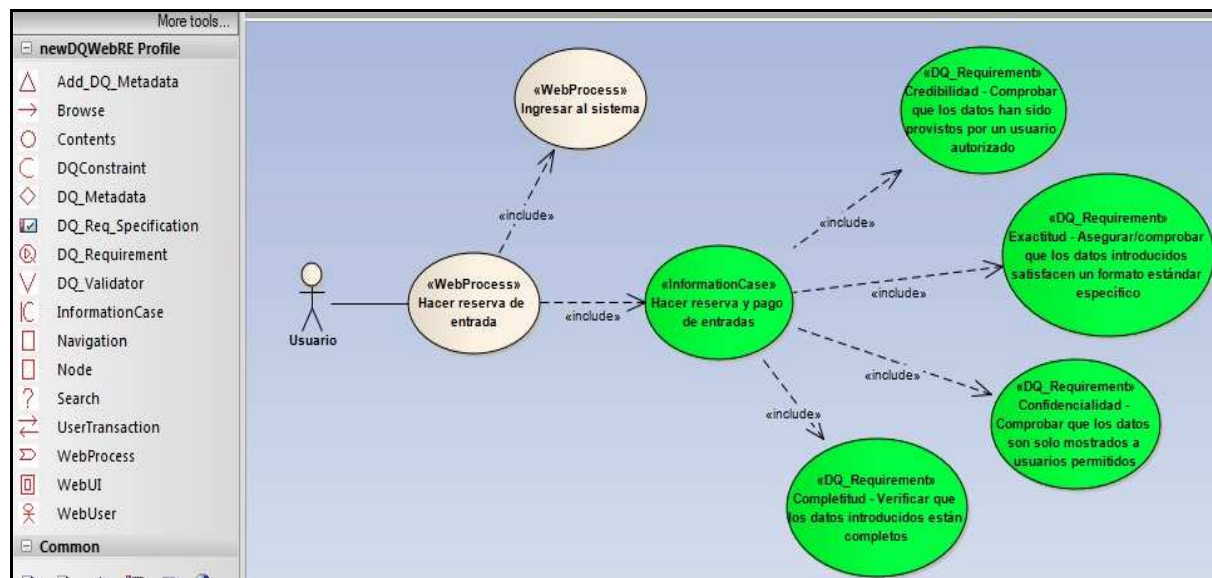


Figura 20. Diagrama de casos de uso especificando requisitos de DQ.

Haciendo uso de los elementos definidos en el perfil, es posible también realizar los correspondientes diagramas de actividad (ver Figura 21). En esta figura, el diagrama modela las actividades principales llevadas a cabo con el objetivo de describir de mejor manera el caso de uso “*Hacer reserva de entrada*”.

En este diagrama de actividades (ver Figura 21), el analista será capaz de modelar las actividades específicas para cumplir con los requisitos de DQ, dichas actividades estarán relacionadas a los distintos elementos propios del modelado de la aplicación Web. Estas actividades de DQ específicas son derivadas a partir de los requisitos de DQ que cada usuario define para los datos que serán gestionados en cada elemento del tipo “*InformationCase*”.

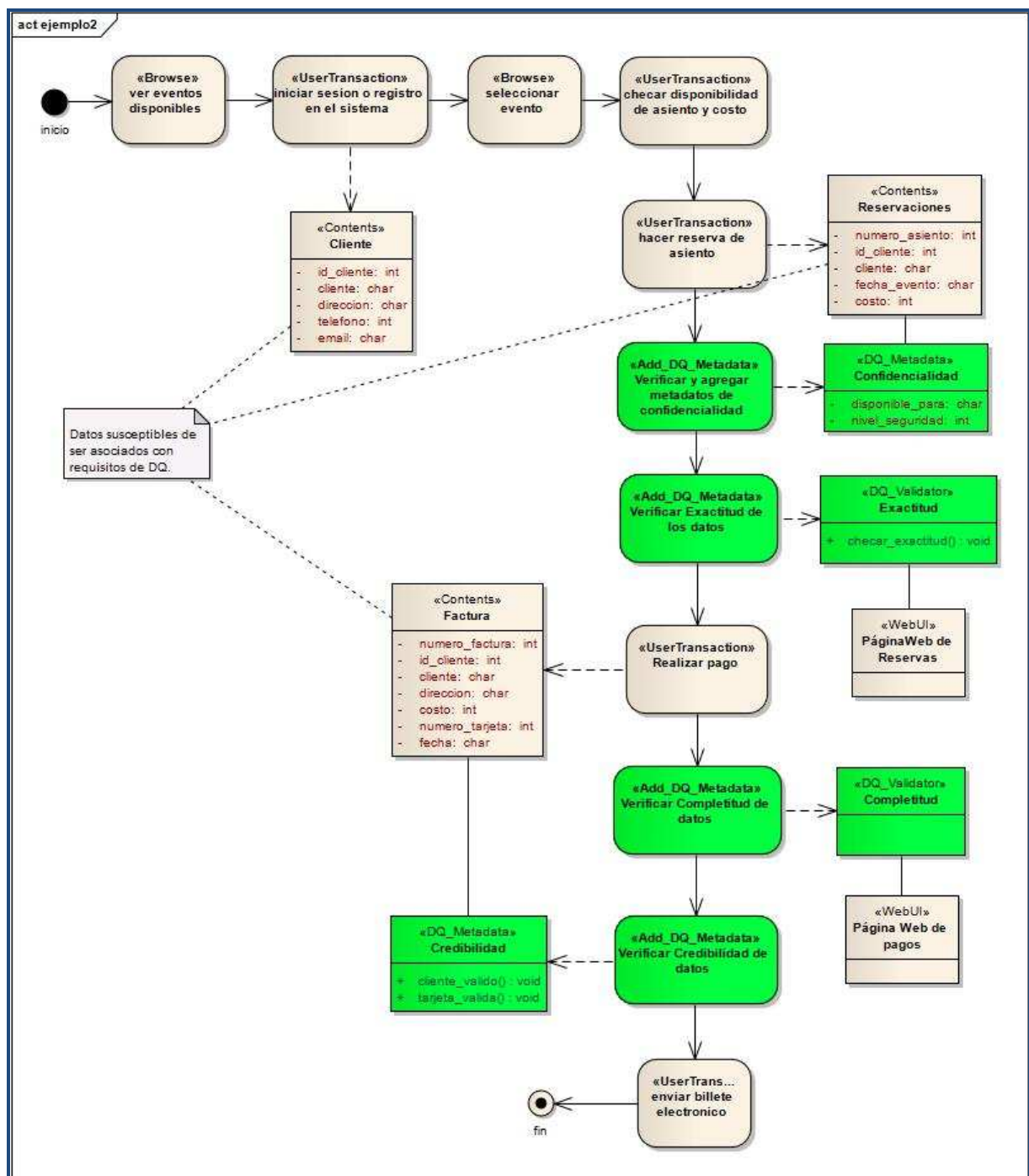


Figura 21. Diagrama de actividades con gestión de DQ.

En este ejemplo, la actividad “Verificar y agregar metadatos de confidencialidad” será la responsable de cotejar y anexar los metadatos de confidencialidad (“disponible_para” y “nivel_seguridad”), los cuales serán almacenados en una instancia de la clase “DQ_Metadata”, cumpliendo de esta manera con el requisito de DQ de “Checar que los datos son solo mostrados a usuarios permitidos”. En el diagrama también se observa la relación existente de los metadatos de Confidencialidad con los datos de la reservación.

La actividad “Verificar Exactitud de los datos”, será la responsable de agregar las funciones específicas (almacenadas en una instancia de la clase “DQ_Validator”)

con el fin de verificar la exactitud de los datos gestionados en el elemento “*PáginaWeb de Reservas*” del tipo “*WebUI*”.

De manera similar, la actividad “*Verificar Completitud de datos*” estará encargada de agregar las funciones específicas para verificar la *Completitud* de cada uno de los elementos gestionados dentro de la “*Página Web de pagos*” (tipo “*WebUI*”).

Finalmente, la actividad “*Verificar Credibilidad de datos*” será la responsable de gestionar y agregar los metadatos de DQ (“*cliente_valido*” y “*tarjeta_valida*”) almacenados en una instancia de la clase “*DQ_Metadata*”; con el objetivo de garantizar el requisito de DQ de “*Checar que los datos han sido provistos por un usuario autorizado*”, el cual estará relacionado con los datos respectivos de *Facturación* (de tipo “*Content*”).

3.2.7 Metamodelo para el Diseño de Aplicaciones Web sensibles a DQ (DAQ_UWE)

3.2.7.1 Metamodelo y perfil UWE

UWE (*UML-based Web Engineering*) es un proceso de desarrollo para aplicaciones Web, con especial énfasis en el diseño sistemático, la personalización y la generación semiautomática (Koch y Kraus 2002). UWE describe una metodología de diseño sistemático usando exclusivamente técnicas de UML. Además prescribe cómo construir diversos modelos de acuerdo a diferentes intereses (enfoques): *contenido*, *navegacional*, *presentación* y *procesos*.

El enfoque UWE proporciona un conjunto de elementos de modelado específico de dominio para el modelado de los distintos enfoques. Estos elementos de modelado y sus relaciones son especificados por un metamodelo.

Es posible mapear el metamodelo UWE a un *perfil UML*. Así, UWE es compatible con MOF (OMG 2002), lo que significa que cualquier herramienta CASE para el modelado UML que permita la gestión de perfiles, podrá ser utilizada para crear modelos UWE para el desarrollo de aplicaciones Web.

UWE define vistas especiales gráficamente representadas por diagramas de clase UML, organizadas en cuatro paquetes: *Contenido* (Content), *Navegacional* (Navigation), *Presentación* (Presentation) y *Proceso* (Process). El *modelo de Contenido* dentro de UWE no difiere del modelado de contenido para cualquier otro tipo de aplicación no Web. El *paquete de Navegación* es representado en la Figura 22. Una descripción detallada de cada uno de sus elementos se muestra en (Koch y Kroib 2008).

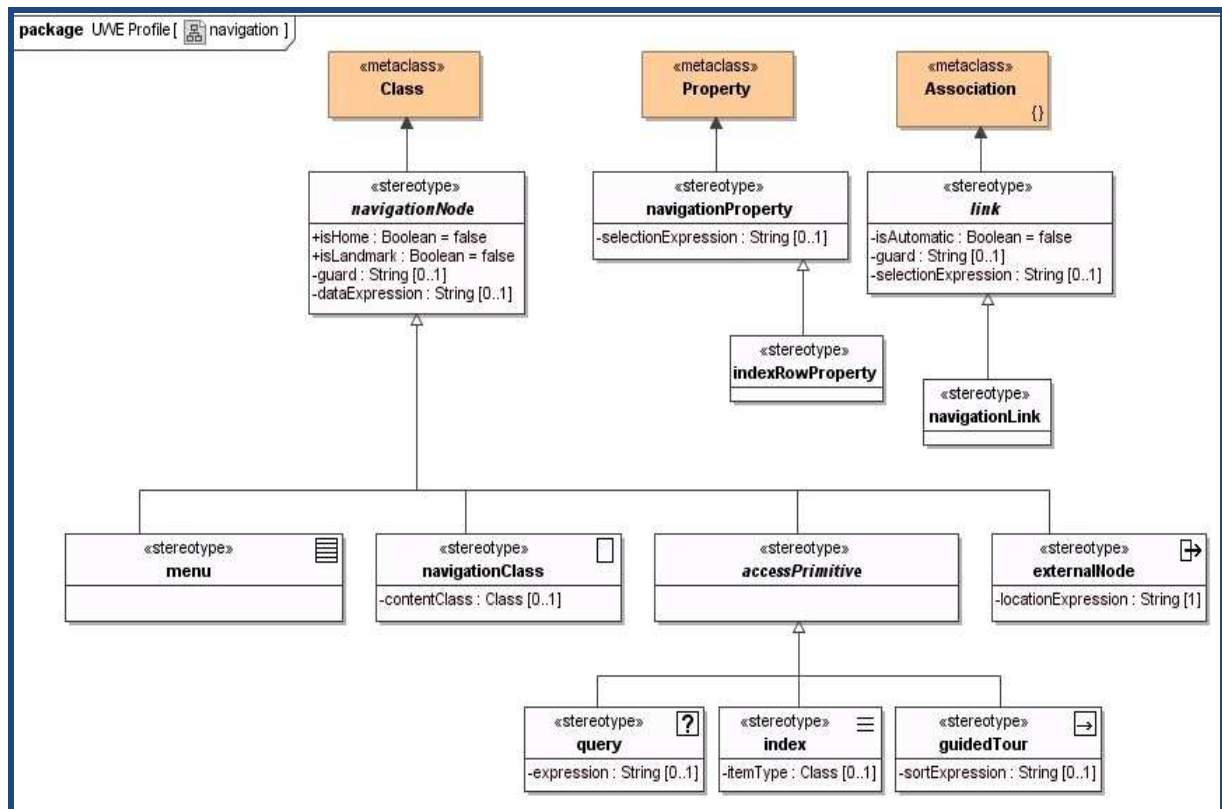


Figura 22. Paquete de Navegación de UWE.

El *modelo de presentación* (*paquete de Presentación*) provee una vista abstracta sobre la interfaz de usuario (UI) de una aplicación Web. Este modelo describe la estructura básica de la UI, por ejemplo, texto, imágenes, formas y anclas (anchors). Los elementos de interfaz de usuario no representan componentes concretos de ninguna tecnología de presentación, simplemente describen la funcionalidad requerida en un punto en particular de la interfaz de usuario. Los elementos contenidos en este paquete se muestran en la Figura 23.

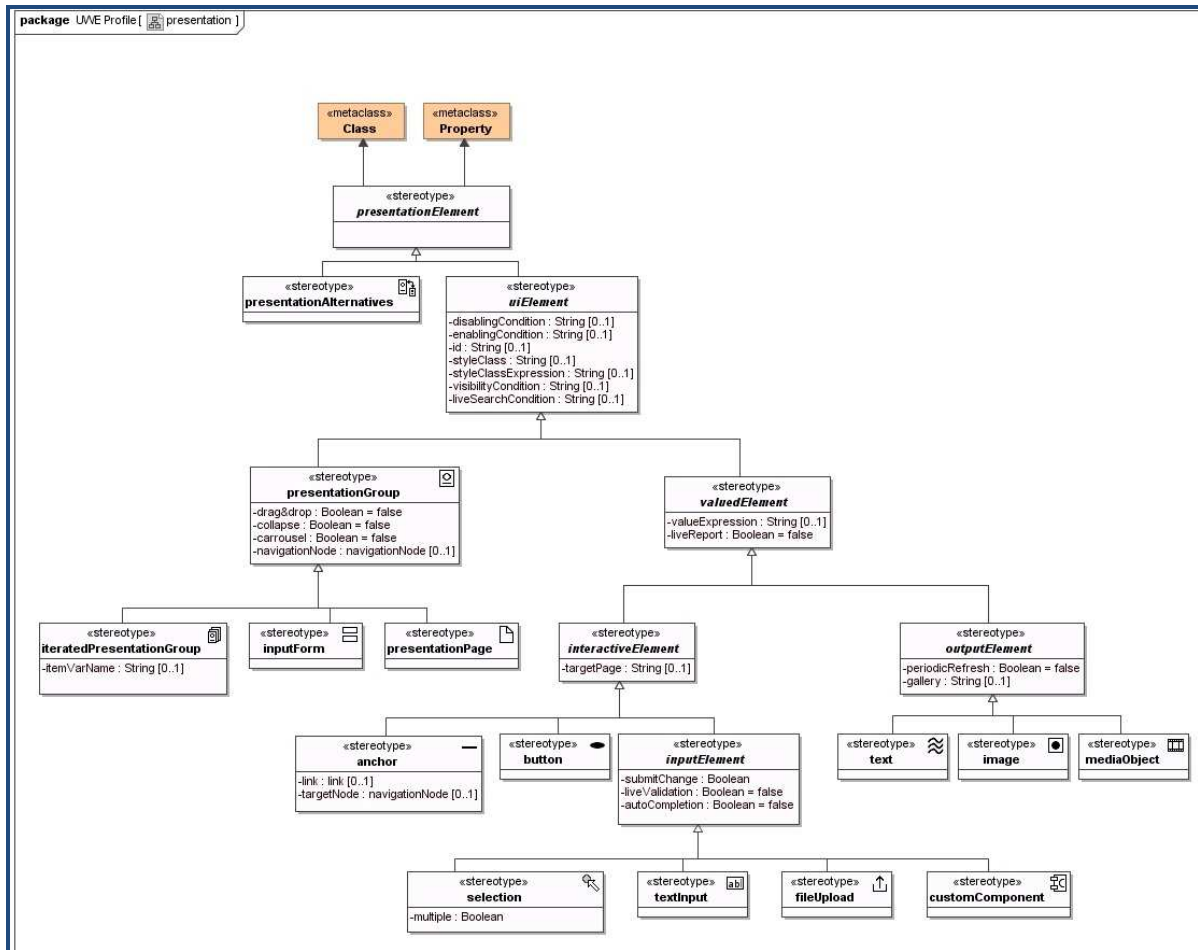


Figura 23. Paquete de Presentación de UWE.

El *paquete de Proceso* provee elementos de modelado para la integración de procesos de negocio dentro de un modelo de aplicación Web UWE. Los elementos del modelo y sus relaciones se muestran en la Figura 24. Estos elementos son descritos de forma extensiva en (Koch y Kroib 2008).

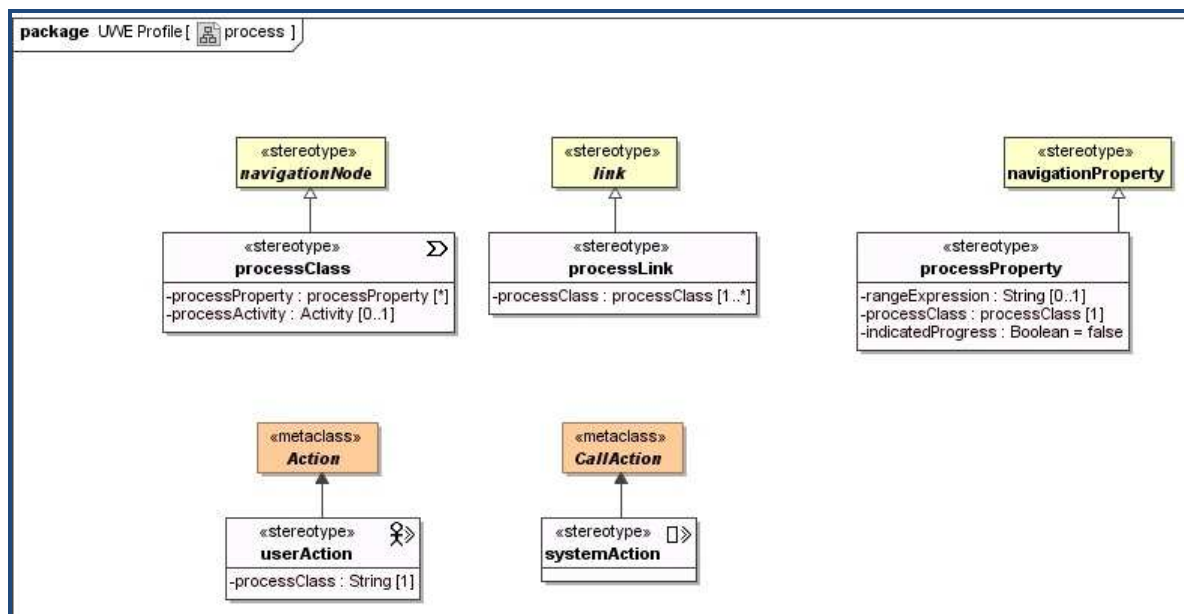


Figura 24. Paquete de Proceso de UWE.

El perfil UWE especifica cómo los conceptos del metamodelo UWE son relacionados y representados en UML estándar usando estereotipos y restricciones. La Tabla 34 muestra la representación gráfica de este perfil, mostrando cómo los estereotipos definidos para cada clase del metamodelo extienden una metaclase de UML (OMG 2005).

UWE stereotype	UML base class	Used in	Icon
menu	Class	navigation model	
navigationClass	Class	navigation model	
externalNode	Class	navigation model	
query	Class	navigation model	
index	Class	navigation model	
guidedTour	Class	navigation model	
navigation property	Property	navigation model	
userAction	Class	process model	
systemAction	Class	process model	
processClass	Class	process model	
process property	Property	navigation/process model	
presentationAlternatives	Class	presentation model	
presentationGroup	Class	presentation model	
iteratedPresentationGroup	Class	presentation model	
inputForm	Class	presentation model	
presentationPage	Class	presentation model	
anchor	Class	presentation model	
button	Class	presentation model	
text	Class	presentation model	
image	Class	presentation model	
mediaObject	Class	presentation model	
selection	Class	presentation model	
textInput	Class	presentation model	
fileUpload	Class	presentation model	
customComponent	Class	presentation model	
presentation property	Property	presentation model	
navigationLink	association	navigation model	
processLink	association	navigation model	

Tabla 34. Estereotipos definidos en UWE.

3.2.7.2 Propuesta de Metamodelo ampliado y Perfil UML para el Diseño de Aplicaciones Web con conciencia de calidad de datos – DAQ_UWE

Una de las motivaciones para el desarrollo de este metamodelo ampliado, es proveer tanto a diseñadores como a desarrolladores de aplicaciones Web, con los *elementos de DQ* necesarios para llevar a cabo el diseño de este tipo de aplicaciones.

De forma específica la utilización de este metamodelo es fundamental en la actividad 4 (*“TPS. Trazado de los Puntos de Vista del Sistema”*) de la propuesta metodológica *DQ-VORD*.

Asimismo, en la sección 3.2.7.3, se describen un conjunto de reglas de mapeo, cuyo objetivo es llevar a cabo la transformación directa de elementos modelados con el metamodelo *DQ_WebRE* a elementos del metamodelo *DAQ_UWE*.

Habiendo realizado un análisis de las diferentes propuestas existentes (ya mencionadas en apartados anteriores), se decidió tomar como base la propuesta del metamodelo *UWE*, publicada en (Koch y Kraus 2002), y actualizada en (Koch y Kraus 2003; Koch y Kroib 2008), en la cual los autores proponen un metamodelo para la representación de conceptos y relaciones dentro de Ingeniería Web dirigida por Modelos. Este metamodelo constituye la base para la definición de un *perfil UML para el diseño de aplicaciones Web (UWE)*.

Se decidió considerar la propuesta *UWE* como plataforma para agregar elementos de *DQ*, ya que *UWE* abarca una gran cantidad de elementos para el diseño de aplicaciones Web, considerados esenciales al momento de modelar este tipo de aplicaciones, además de que la implementación de su metamodelo provee una gran flexibilidad para la incorporación de nuevos elementos de modelado. Todo ello nos facilita la introducción de los nuevos elementos relacionados con *DQ* y su relación con los elementos ya definidos en el metamodelo *UWE*.

Una vez mostrado en el apartado previo las principales características y elementos del metamodelo y el perfil *UWE*, en este apartado se describe la propuesta *DAQ_UWE*, que consiste de un metamodelo ampliado para la integración de elementos considerados esenciales para la gestión de *DQ* en la etapa de diseño de aplicaciones Web.

Se puede observar la incursión de un nuevo elemento (resaltado en color amarillo) como parte del paquete de Presentación *“UIE_DQVerifier”* (ver Figura 25), tres nuevos elementos dentro del paquete de Proceso: *“DQProcessClass”*, *“DQLink”* y *“DQProcessProperty”* (ver Figura 26), y finalmente tres nuevos elementos como parte del paquete de Contenido: *“DQ_Constraint”*, *“DQDimProperty”* y *“DQDim”* (ver Figura 27).

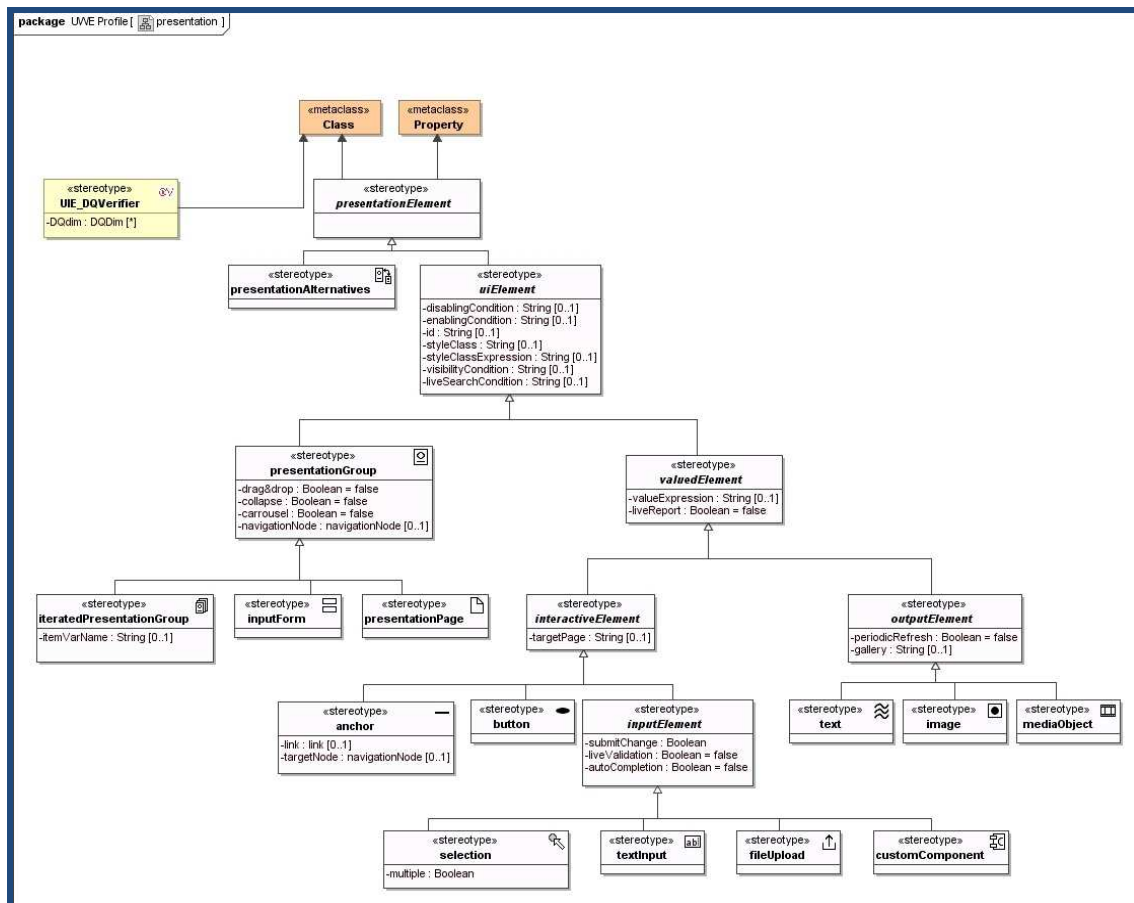


Figura 25. Paquete de Presentación con nuevo elemento para gestión de DQ.

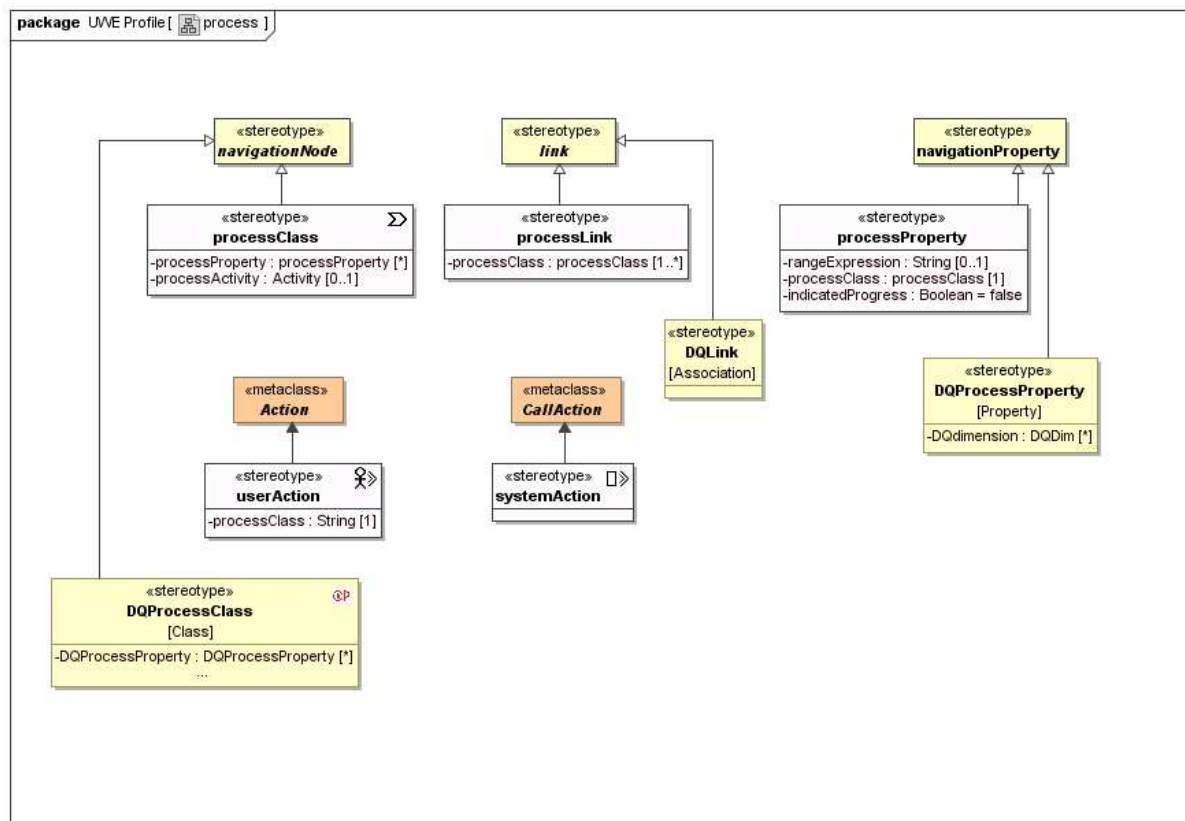


Figura 26. Paquete de Proceso con elementos para gestión de DQ.

Nombre	UIE_DQVerifier	Icon	
Clase Base	Class		
Descripción	Esta metaclassa representa una clase responsable para especificar y verificar que los datos gestionados por cada elemento en la interfaz de usuario satisfacen las dimensiones de DQ (<i>DQDim</i>) especificadas.		
Restricciones	- Esta clase tiene que estar relacionada al menos con un elemento de tipo “ <i>UIElement</i> ”.		
Valor etiquetado	Ninguno.		
Usado en	Modelo de Presentación.		
Nombre	DQProcessClass	Icon	
Clase Base	navigationNode::Class		
Descripción	Representa un proceso específico responsable por la gestión de los metadatos de DQ asociados a los datos que serán manejados en cada “ <i>ProcessClass</i> ”. En otras palabras, el elemento “ <i>DQProcessClass</i> ” será capaz de utilizar/usar la clase “ <i>DQDim</i> ”, la cual especifica los metadatos de DQ de cada una de las dimensiones de DQ definidas.		
Restricciones	- Esta clase tiene que estar relacionada al menos con un elemento de tipo “ <i>ProcessClass</i> ”.		
Valor etiquetado	Ninguna.		
Usado en	Modelos de <i>Navegación y Proceso</i>		
Nombre	DQProcessProperty		
Clase Base	navigationProperty::Property		
Descripción	Un elemento de tipo <i>DQProcessProperty</i> es propiedad de la clase “ <i>DQProcessClass</i> ”, y es utilizado para especificar las dimensiones de DQ que deberían ser satisfechas por cada uno de los datos gestionados en los elementos del tipo “ <i>ProcessClass</i> ” relacionados.		
Valor etiquetado	Ninguno.		
Usado en	Modelos de <i>Navegación y Proceso</i>		
Nombre	DQLink		
Clase Base	link::Association		
Descripción	Este representa un tipo de relación (link) especial, y es utilizado para conectar elementos de tipo “ <i>ProcessClass</i> ” con elementos de tipo “ <i>DQProcessClass</i> ”.		
Valor etiquetado	Ninguno.		
Usado en	Modelo de <i>Navegación</i> .		
Nombre	DQDim	Icon	
Clase Base	Class		
Descripción	Este representa un elemento de contenido, esta metaclassa corresponde a cada una de las dimensiones de DQ, y es responsable del almacenado de los diferentes metadatos de DQ (<i>DQDimProperty</i>).		
Valor etiquetado	None.		
Usado en	Modelo de <i>Contenido</i> .		
Nombre	DQDimProperty		
Clase Base	Property		
Descripción	Un <i>DQDimProperty</i> es propiedad (atributo) de una dimensión de DQ, y es usada para definir los metadatos de DQ relacionados a cada dimensión de DQ (<i>DQDim</i>).		
Valor etiquetado	Ninguno.		
Usado en	Modelo de <i>Contenido</i> .		
Nombre	DQ_Constraint	Icon	
Clase Base	Class		
Descripción	Este representa un elemento de contenido, esta metaclassa corresponde a cada una de las restricciones (<i>Constraints</i>) que deberían ser definidas y asociadas a las diferentes dimensiones de DQ, además de sus correspondientes límites (por ejemplo, “ <i>upper_bound</i> ” and “ <i>lower_bound</i> ”).		
Valor etiquetado	Ninguno.		
Usado en	Modelo de <i>Contenido</i> .		

Tabla 35. Especificación de estereotipos de DQ del perfil DAQ_UWE.

3.2.7.3 Propuesta de Reglas de mapeo de elementos definidos en DQ_WebRE a elementos definidos en DAQ_UWE

Considerando las ventajas del desarrollo de software dirigido por modelos (DSDM) y con el objetivo de obtener un mejor entendimiento de cómo un diseñador puede usar cada uno de los nuevos elementos estereotipados de DQ (tanto a nivel de análisis – especificado por DQ_WebRE, como a nivel de diseño – especificado por DAQ_UWE), se proponen un conjunto de reglas de mapeo. Esta propuesta de reglas de mapeo (ver Tabla 36) deberían ser tomadas en cuenta por el diseñador al momento de modelar los correspondientes diagramas de diseño (*contenido, navegación y presentación*). Todo ello como parte de la actividad 4 (“TPS. Trazado de los Puntos de Vista del Sistema”) de la metodología DQ-VORD.

Cabe mencionar que todos estos elementos estereotipados definidos en DAQ_UWE pueden ser obtenidos directamente a partir de los diferentes elementos modelados en el diagrama de actividades usando el perfil “DQ_WebRE” (utilizado en la actividad 3 de la metodología DQ-VORD).

Elemento DQ_WebRE	Elemento DAQ_UWE	Descripción	Usado en
Add_DQ_Metadata	DQProcessClass	Cada elemento “Add_DQ_Metadata” será transformado en un elemento de tipo “DQProcessClass”. La <u>relación</u> de un elemento de tipo “UserTransaction” a otro elemento de tipo “Add_DQ_Metadata” será convertida a una relación específica del tipo “DQLink”, entre los elementos de tipo “ProcessClass” y “DQProcessClass”. El nombre de los elementos de tipo “DQ_Metadata” y “DQ_Validator” relacionados a “Add_DQ_Metadata”, serán convertidos a atributos de tipo “DQDim” dentro de los elementos “DQProcessClass”.	Modelo de Navegación
DQ_Metadata	DQDim	Cada elemento “DQ_Metadata” será transformado en un elemento del tipo “DQDim”, además de copiar sus respectivos atributos definidos (<i>DQDim properties</i>). Cada <u>relación</u> entre un elemento de tipo “DQ_Metadata” y un elemento “Contents”, será convertida en una relación entre un elemento “DQDim” y un elemento “ContentClass” (ésta es una clase normal para el almacenado de datos).	Modelo de Contenido
DQ_Validator	UIE_DQVerifier	Cada elemento “DQ_Validator” será transformado en un elemento “UIE_DQVerifier”. Una <u>relación</u> entre los elementos del tipo “DQValidator” y “WebUI” será convertida en una relación entre elementos del tipo “UIE_DQVerifier” y “PresentationPage”; esta relación pudiera llegar a ser más específica a elementos particulares dentro de <i>PresentationPage</i> , por ejemplo: <i>presentationGroup</i> , <i>textInput</i> , etc.	Modelo de Presentación
DQ_Validator	DQDim	Cada elemento “DQ_Validator” será transformado a un elemento “DQDim”. Este elemento (DQDim) estará a cargo de almacenar las operaciones de validación y representarlas en el modelo de contenido (en caso que este elemento <i>DQDim</i> haya sido definido previamente, solo será necesario agregar la operación de validación).	Modelo de Contenido
DQConstraint	DQ_Constraint	Cada elemento “DQConstraint” será transformado en otro elemento de tipo “DQ_Constraint”. Este elemento será encargado de almacenar los datos específicos a los cuales las restricciones estarán relacionadas, además de las variables para la definición de los límites máximos y mínimos (“ <i>upper_bound</i> ” y “ <i>lower_bound</i> ”). Una <u>relación</u> entre elementos “DQ_Validator” y “DQConstraint” será convertida en una relación entre elementos del tipo “DQDim” y “DQ_Constraint” dentro del modelo de contenido.	Modelo de Contenido

Tabla 36. Propuesta de reglas de mapeo de elementos “DQ_WebRE” a elementos “DAQ_UWE”.

3.2.7.4 Ejemplo de aplicación del perfil DAQ_UWE

Una vez implementado el perfil en la herramienta Enterprise Architect, se obtiene un ambiente adecuado para trabajar y realizar diagramas de modelado con los nuevos elementos definidos. En este apartado se muestra un breve ejemplo de aplicación usando el nuevo perfil propuesto “DAQ_UWE”. El ejemplo ilustrativo describe el diseño de una simple aplicación Web para gestionar una libreta de direcciones de contactos. Cada contacto contiene información específica almacenada en diferentes clases.

Continuando con el enfoque de separación de intereses de la propuesta UWE, a continuación se muestran los diferentes modelos de diseño (*Contenido*, *Navegación* y *Presentación*) de la aplicación Web de manera separada.

El *modelo de Contenido* de la aplicación Web se muestra en la Figura 28, en este modelo se observa las distintas clases definidas para almacenar los datos de cada contacto: *AddressBook*, *Contact*, *Phone* y *Address*. En este diagrama, además podemos ver las clases relacionadas a la definición de cada una de las dimensiones de DQ (*Complejidad*, *Conformidad*, *Confidencialidad* y *Exactitud*), así como también sus atributos (metadatos) y sus operaciones. Este modelo de Contenido está representado como un simple diagrama de clases UML.

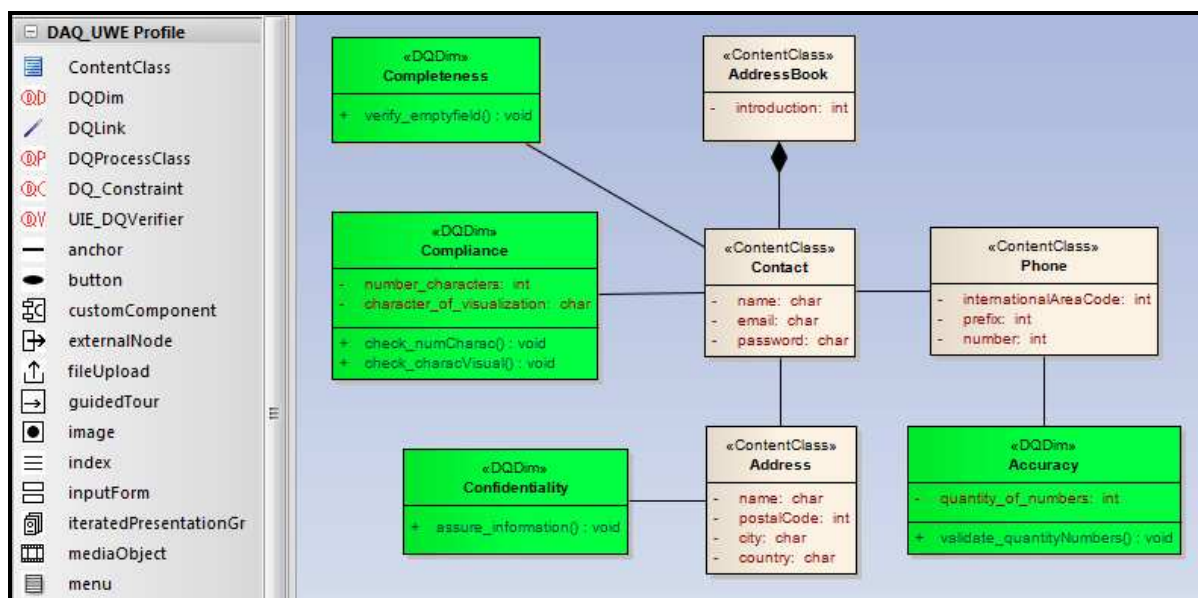


Figura 28. Ejemplo de diagrama del Modelo de Contenido con gestión de DQ

Haciendo uso de los elementos estereotipados definidos en el nuevo perfil, es posible también modelar la estructura de hipertexto usando el *modelo de Navegación* (ver Figura 29). En este diagrama se puede observar el conjunto de nodos y ligas (links) de navegación de la aplicación Web. UWE distingue entre diferentes tipos de nodos, tales como clases de *Navegación*, *Menús*, *Index* y

Consultas (Query). Este modelo de Navegación presenta dos clases de Navegación (*AddressBook* and *Contact*) y una clase de tipo Index (*ContactList*). El actualizado de los datos, por ejemplo, “*Update Info*”, “*Update Address Info*” y “*Update Phone Info*”, es modelado con clases adicionales (*ProcessClass*) en el modelo de navegación, también llamados “*nodos de proceso*”, e integrados en el flujo de navegación a través de links de proceso (*processLink*).

De forma similar, la gestión de DQ es modelada con elementos de tipo “*DQProcessClass*” (por ejemplo, *VerifyDQ*, *CheckDQ* y *ValidateDQ*) relacionados a las diferentes clases de tipo “*ProcessClass*” a través de “*DQLink*”. Estos elementos de tipo “*DQProcessClass*” son responsables de indicar las dimensiones de DQ que deberían satisfacer los datos involucrados (almacenados) en cada elemento de tipo “*ProcessNode*”.

En este ejemplo específico, se observa que los datos (*name*, *email* y *password*) gestionados en el nodo de proceso “*Update Info*”, deberían satisfacer las dimensiones de DQ de Completitud (Completeness) y Conformidad (Compliance). De forma similar, los datos (*address*, *postalCode*, *city* y *country*) gestionados en el nodo “*Update Address Info*” deberán satisfacer la dimensión de DQ de Confidencialidad (Confidentiality). Finalmente, los datos “*internationalAreaCode*”, “*prefix*” y “*number*”, gestionados en el nodo de proceso “*Update Phone Info*” deberán cumplir con la dimensión de DQ de Exactitud (Accuracy).

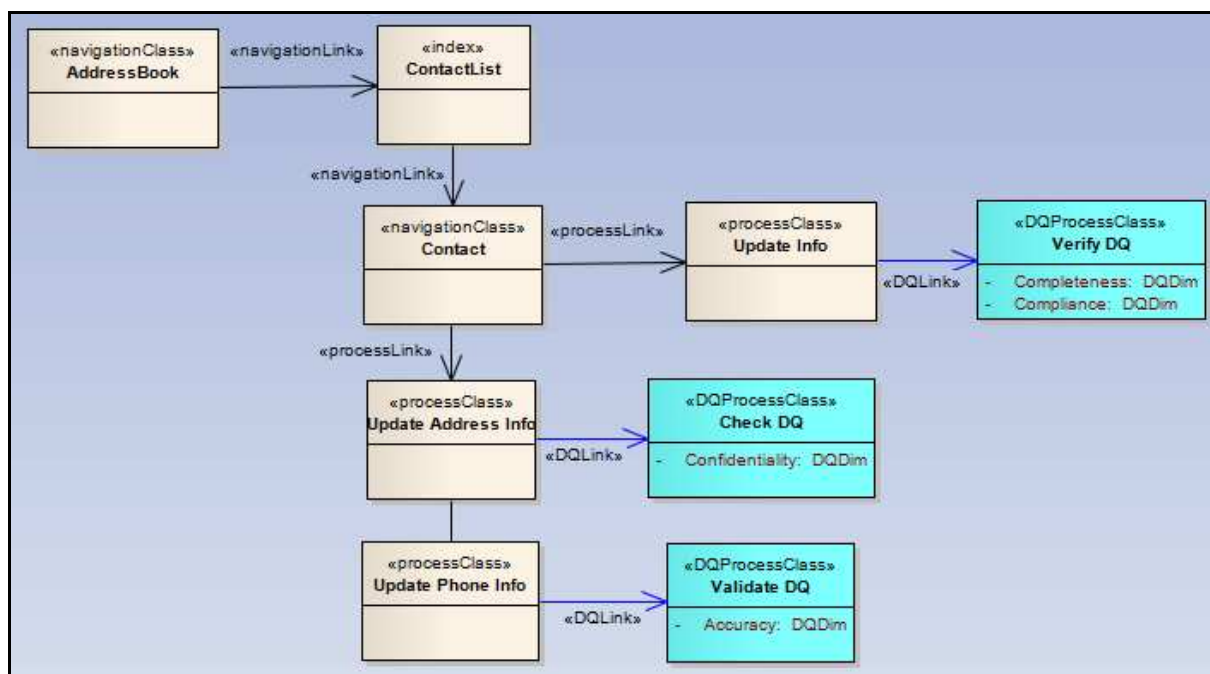


Figura 29. Modelo de Navegación con gestión de DQ.

La Figura 30 muestra el modelo de Presentación al momento de ejecutarse la aplicación Web. De forma similar al modelado anterior, es posible utilizar los elementos definidos en el perfil a través de diagramas de clases para la

representación del modelo de Presentación. La página de presentación “*Page of Updated*” contiene tres grupos de presentación (de tipo *presentation class*): “*Contact Information*”, “*Address Information*” y “*Phone Information*”. Por cada grupo de presentación diferentes campos (de tipo “*text input*”) son desplegados, por ejemplo: *name*, *email*, *password*, *address*, *city*, etc.

En este modelo de Presentación, la clase “*Verify Completeness*” (de tipo *UIE_DQVerifier*) está relacionada al grupo de presentación “*Contact Information*”, esto significa que esta clase será responsable de verificar la Completitud a través de la operación “*verify_emptyfield()*” de todos los elementos contenidos en este grupo de presentación (*name*, *email* y *password*), y así satisfacer la dimensión de DQ de Completitud (*Completeness*).

La clase “*Check Compliance*” está relacionada sólo al elemento “*password*” (de tipo “*text input*”) lo que significa que esta clase será responsable de ejecutar dos operaciones específicas (“*check_numCharac()*” y “*check_characVisual()*”) con el objetivo de verificar la *Conformidad* del dato almacenado en este elemento específico.

De manera similar, la clase “*Validate Accuracy*” será la encargada de ejecutar la operación específica de “*validatequantityNumbers()*”, con el objetivo de verificar la Exactitud (*Accuracy*) de los distintos elementos gestionados dentro del grupo de presentación “*Phone Information*”.

Finalmente, la clase “*Assure Confidentiality*” será responsable de asegurar que los datos contenidos en el grupo de presentación “*Address Information*” cumplen con la dimensión de DQ de Confidencialidad (*Confidentiality*), a través de la operación “*assure_information()*”.

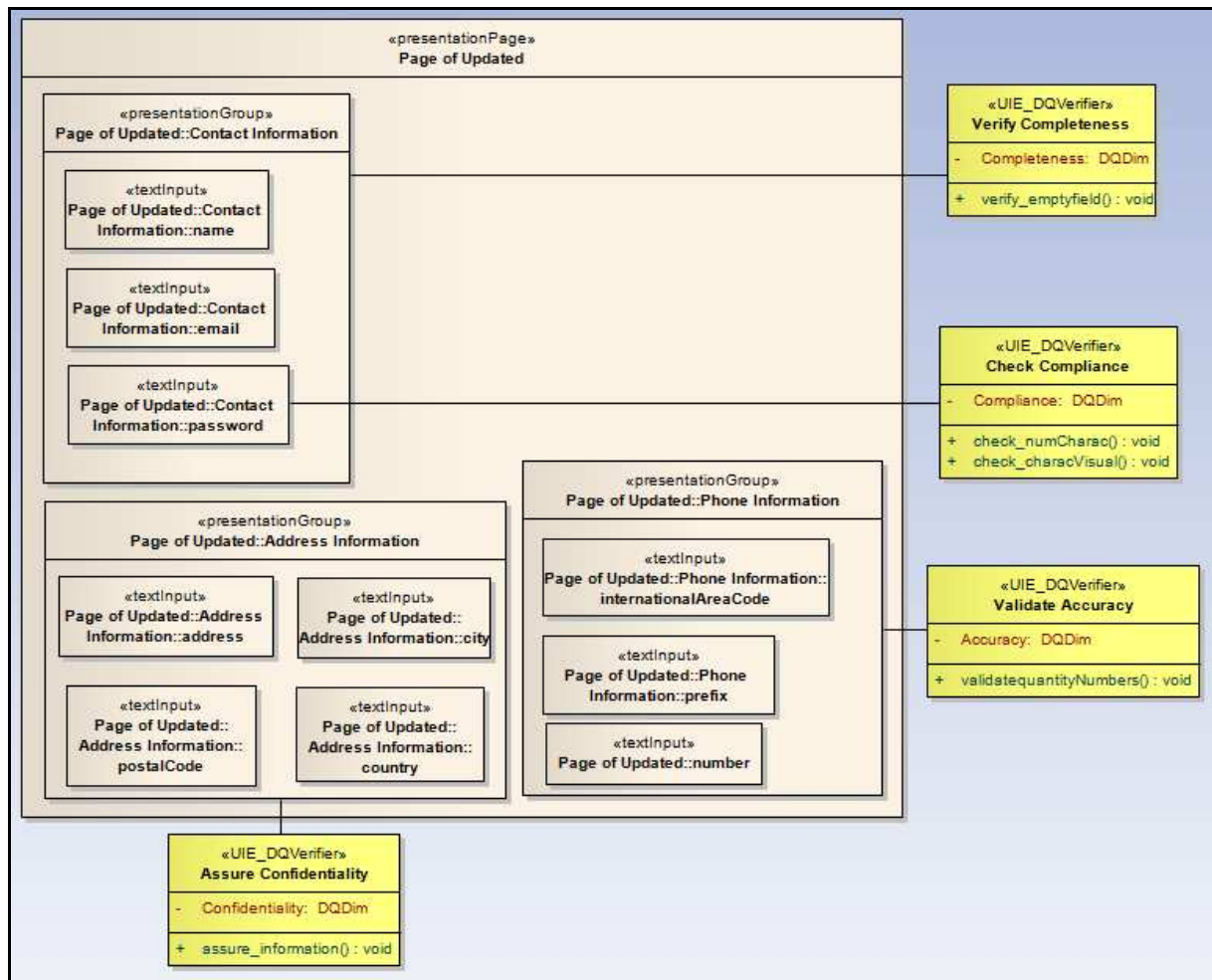


Figura 30. Modelo de Presentación incluyendo elementos de DQ.

3.2.8 Plantillas de documentos

En este apartado se presentan un conjunto de plantillas para los distintos documentos que se manejan a lo largo de la ejecución de las distintas actividades y tareas de la metodología de gestión de proyectos de especificación de requisitos de calidad de datos *DQ-VORD*. Para cada documento se detallarán cuáles son los campos necesarios, así como una breve descripción de lo que debe contener cada uno de ellos.

3.2.8.1 Plantilla de especificación de funcionalidades Web

Esta plantilla (véase Tabla 37) podrá utilizarse para llevar a cabo la especificación (relación) de cada uno de los requisitos funcionales de la aplicación con las distintas funcionalidades Web descritas por Collins (Collins 2001).

ESPECIFICACIÓN DE FUNCIONALIDADES WEB		
Punto de Vista	Requisito Funcional	Funcionalidad Web descrita por Collins
Nombre del punto de vista identificado.	Nombre del requisito funcional a ser implementado en la aplicación Web.	Nombre de la funcionalidad Web relacionada con el requisito funcional.

Tabla 37. Especificación de funcionalidades Web.

3.2.8.2 Plantilla de especificación de puntos de vista

La plantilla debe contener al menos los siguientes campos relacionados para la especificación de los diferentes puntos de vista definidos (véase Tabla 38).

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Nombre del punto de vista.
Enfoque	Este muestra la definición desde la perspectiva tomada por el punto de vista.
Atributos	Descripción de las características propias del punto de vista.
Requisitos	Referencia a un conjunto de escenarios describiendo cómo reacciona el sistema a eventos específicos del punto de vista.
Funcionalidades Web	Referencia a un conjunto de descripciones de las funcionalidades Web.
Excepciones	Descripción de las excepciones que el sistema será capaz de manejar.
Historia	Este almacena las alteraciones de la información del punto de vista a través del tiempo.

Tabla 38. Especificación del Punto de Vista.

3.2.8.3 Plantilla de especificación de requisitos del sistema

La plantilla debe contener al menos los siguientes campos relacionados para la especificación de los requisitos funcionales del sistema (véase Tabla 39).

ESPECIFICACIÓN DEL REQUISITO FUNCIONAL	
Referencia	Nombre/Código del requisito.
Descripción	Justificación o descripción del requisito del sistema Web.
Datos	Referencia a una lista o conjunto de datos que gestionará el requisito, pudiendo estar en distintas notaciones.
Punto de Vista	Lista de puntos de vista relacionados con este requisito.
Requisitos no funcionales	Descripción de los requisitos no funcionales relacionados (en caso de existir).
Requisitos de DQ	Descripción de los requisitos (dimensiones) de DQ relacionados.

Tabla 39. Especificación del Requisitos del sistema.

3.2.8.4 Plantilla de petición de modificación

El documento con las peticiones de modificación para alguno de los documentos entregables de las fases finales de la metodología DQ-VORD (*“Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)”*, *“Modelo de Diseño sensible a de calidad de datos (MDSDQ)”*), deben contener una tabla por cada una de las peticiones de cambio solicitadas. Cada una de estas tablas tendrá el formato y contenido reflejado en la Tabla 40.

PETICIÓN DE MODIFICACIÓN	
Identificador de petición	Identificador para la petición de modificación con el formato <i>“nombreproyecto_MOD_Nº”</i> . Donde Nº es el número de la petición.
Fecha de realización de la petición	Fecha en la que se realiza la petición de modificación (normalmente realizada por el Jefe de desarrollo).
Estado de la petición	La petición podrá estar: ➤ <i>Pendiente</i> de atender. ➤ <i>Realizada</i>: cuando se haya atendido y se haya contestado al solicitante. ➤ <i>Rechazada</i>: cuando después de estudiar la petición, esta ha sido denegada por no considerarse oportuna.
Tipo de cambio	La petición podrá suponer una: ➤ Inclusión de algo que falta. ➤ Modificación de algo existente. ➤ Borrado de algo que no aplica.
Fecha de finalización de la petición	Fecha en la que se comunica al solicitante (normalmente Jefe de desarrollo) la realización o rechazo de la petición.
Documento afectado	Nombre del documento que se ve afectado por la solicitud de modificación.
Sección afectada por el cambio	Lugar del documento final que se ve afectado por la petición de modificación.
Descripción y justificación de la modificación	Donde se detalla la modificación que se requiere en alguno de los documentos finales y la justificación de dicha modificación

Tabla 40. Petición de modificación.

3.2.8.5 Plantilla para la gestión de la configuración

La plantilla debe contener al menos los siguientes campos, necesarios para llevar a cabo el control de las versiones de los documentos generados durante la ejecución del proyecto de especificación de requisitos de DQ (véase Tabla 41).

NOMBRE DEL DOCUMENTO GENERADO				
Nombre del Fichero:	Realizado por:	Fecha de realización:	Último revisor:	Fecha de revisión:

Tabla 41. Plantilla para la Gestión de la Configuración.

4

Caso de Estudio de Validación

“Pensar es el trabajo más difícil que existe. Quizá sea ésta la razón por la que haya pocas personas que lo practiquen”
– Henry Ford

En esta sección se describe en profundidad un caso de estudio de validación, en el cual se aplicó cada una de las actividades y tareas descritas en la Metodología para la Gestión de Proyectos de Especificación de Requisitos de Calidad de Datos (DQ-VORD). Asimismo, se muestran los productos de entrada y salida de cada una de las tareas propuestas en la metodología.

4.1 Caso de Estudio

En este capítulo se describe con detalle un caso de estudio en el que se ha aplicado la metodología DQ-VORD. En primer lugar se presenta un documento típico de especificación de requisitos software (ERS), que sirvió como punto de partida para la implementación de la metodología. En segundo lugar se describe la aplicación y desarrollo de cada una de las actividades y tareas definidas en la metodología. Mostrando de esta forma los productos de salida obtenidos al final de cada tarea.

4.1.1 Documento de Especificación de Requisitos Software

En la Tabla 42 se describe un ejemplo de documento de especificación de requisitos software (ERS), para llevar a cabo el desarrollo de una aplicación Web.

Documento de Especificación de Requisitos Software (ERS)
Como encargado del área de Tecnologías de Información de la empresa Acme, se nos ha encomendado el desarrollo de un portal Web corporativo que muestre en otras cosas, la imagen de la empresa, los productos que vende, que permita realizar una búsqueda por algún producto en específico, así como la incorporación de un mecanismo de comunicación que permita a cualquier persona (cliente potencial) poder solicitar un determinado producto a la empresa; además como una de las funcionalidades principales, el portal deberá proveer al Administrador un conjunto de funcionalidades específicas para el cálculo de nómina de los empleados de la empresa. La descripción más a detalle del subsistema de cálculo de nómina vía Web se menciona a continuación: Como una de las partes fundamentales de la aplicación Web, se ha de implementar un conjunto de funciones para llevar a cabo el cálculo de la nómina de todos los empleados de la empresa, dicho subsistema vendrá a reemplazar al existente. Acme necesita un nuevo sistema que permita a los empleados registrar la información del tiempo que trabajan de manera electrónica y automáticamente generar los cheques de pago basados sobre el número de horas trabajadas y la cantidad total de ventas (para empleados por comisión). Este tipo de empleados envía sus órdenes de venta en donde se refleja la fecha y la cantidad de la venta. El nuevo sistema permitirá a los empleados ingresar información de horas de trabajo (timecard), ingresar órdenes de venta, cambiar las preferencias del empleado (como método de pago), y crear varios reportes. Por razones de seguridad y auditoría, los empleados sólo podrán acceder y editar sus propias timecard y órdenes de venta, por lo que el acceso al sistema será mediante un usuario y una clave de acceso para cada empleado. El sistema guardará la información de todos los empleados de la compañía. A los empleados por comisión se les paga por un porcentaje de horas. Ellos envían sus timecard que registra la fecha y el número de horas trabajadas por un número de cargo particular. Si alguien trabaja por más de 8 horas, Acme le pagará al empleado 1.5 veces su cargo normal por las horas extras. A este tipo de empleados se les paga cada viernes. A algunos empleados se les paga con un salario fijo (asalariado). Sin embargo, ellos también deberán almacenar sus timecard para registrar la fecha y horas trabajadas. Esto es para que el sistema pueda realizar un seguimiento de las horas trabajadas en contra de un número de cargo en particular. Estos empleados son pagados el último día laboral de cada mes. Una de las características más importantes del nuevo sistema es el reporte para empleados. Ellos serán capaces de consultar el número de horas trabajadas, el total de horas facturadas para un proyecto en particular (número de cargo), total del pago recibido en el transcurso del año, tiempo restante de vacaciones, etc. Los empleados podrán escoger su método de pago. Ellos podrán recibir su cheque de pago directamente en su domicilio, o ellos podrán solicitar que se les deposite en una cuenta de banco de su elección. Los empleados podrán también seleccionar pasar a recoger su cheque de pago directamente en la oficina encargada de la empresa. El Jefe de Departamento será responsable de mantener actualizada toda la información de los empleados de su departamento, ejecutar reportes de sus empleados y actualizar la información relacionada a todos los productos asignados a su departamento. El Administrador será el responsable de agregar nuevos empleados, eliminar empleados y actualizar toda su información como nombre, dirección y clasificación de pago (<i>asalariado o por comisión</i>), además de ejecutar los reportes administrativos. El Administrador de la aplicación ejecutará automáticamente el pago de nómina cada viernes y el último día de cada mes.

Tabla 42. Ejemplo de Documento de Especificación de Requisitos Software.

Analizando el documento de Especificación de Requisitos Software (Tabla 42), se identifican los siguientes requisitos funcionales de la aplicación Web (ver Tabla 43):

Requisitos Funcionales Identificados
RF1. Ejecutar reporte administrativo.
RF2. Mantener información de empleados.
RF3. Ejecutar pago de nómina.
RF4. Login al sistema.
RF5. Búsqueda de productos.
RF6. Solicitar un producto.
RF7. Escoger método de pago.
RF8. Ejecutar reporte de empleado.
RF9. Mantener timecard.
RF10. Mantener información de productos.
RF11. Mantener órdenes de venta.

Tabla 43. Listado de Requisitos Funcionales identificados.

En la Figura 31, se muestran los casos de uso identificados para la aplicación.

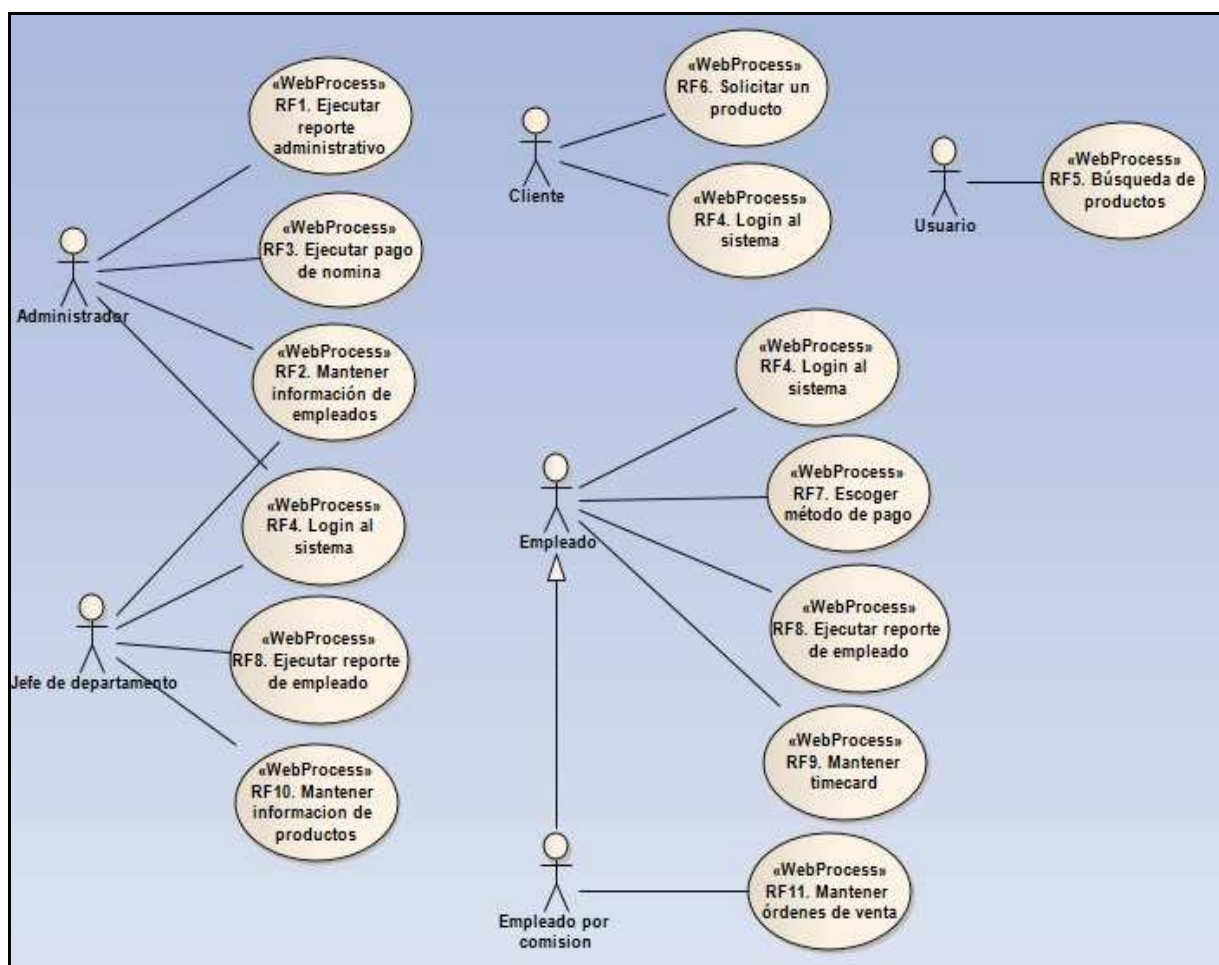


Figura 31. Diagrama de casos de uso.

4.1.2 Aplicación de la metodología DQ-VORD

Una vez descrito en el apartado anterior el documento de especificación de requisitos (ERS), en este apartado se describe la implementación de cada una de las actividades de la metodología DQ-VORD.

IPPW. Identificación de los Puntos de Vista del Portal Web.

- IPPW.1. Identificación de las Funcionalidades críticas susceptibles de requerir ser Complementadas con Requisitos específicos de DQ (IFPW).

El producto de salida de esta tarea en base a la Tabla 3, es una lista de las funcionalidades Web susceptibles de necesitar requisitos de DQ de cada punto de vista (LFWI) (véase Tabla 44). El objetivo de esta tarea es llevar a cabo una *clasificación* de cada requisito funcional con las funcionalidades descritas por Collins (Collins 2001). Para realizar esta tarea, se utilizó el formato mostrado en la Tabla 37.

Punto de Vista	Requisito Funcional	Funcionalidad Web descrita por Collins
Administrador	RF1. Ejecutar reporte administrativo.	Administración
	RF2. Mantener información de empleados.	Administración
	RF3. Ejecutar pago de nómina.	Administración
	RF4. Login al sistema.	Seguridad
Cliente	RF5. Búsqueda de productos.	Capacidades de búsqueda
	RF6. Solicitar un producto.	Procesos y acciones
	RF4. Login al sistema.	Seguridad
Empleado	RF4. Login al sistema.	Seguridad
	RF7. Escoger método de pago.	Administración
	RF8. Ejecutar reporte de empleado.	Seguridad
	RF9. Mantener timecard.	Procesos y acciones
Usuario	RF5. Búsqueda de productos.	Capacidades de búsqueda
Jefe de departamento	RF2. Mantener información de empleados.	Administración
	RF8. Ejecutar reporte de empleado	Seguridad
	RF4. Login al sistema.	Seguridad
	RF10. Mantener información de productos.	Gestión de contenido
Empleado por comisión	RF11. Mantener órdenes de venta.	Procesos y acciones

Tabla 44. Lista de funcionalidades Web susceptibles a requisitos de DQ (LFWI).

- IPPW.2. Identificación de los Puntos de Vista del portal Web (IFPW).

El producto de salida de esta tarea considerando la Tabla 4, consiste en una lista con los *puntos de vista* identificados capaces de proponer requisitos de calidad de datos (PVI) (ver Tabla 45):

Puntos de vista identificados
Administrador
Cliente
Empleado
Usuario
Jefe de departamento
Empleado por comisión

Tabla 45. Lista de puntos de vista identificados capaces de proponer requisitos de DQ (PVI).

➤ IPPW.3. Identificación de los requisitos de calidad de datos DQ (IRDQ).

Una vez obtenidas las distintas funcionalidades Web, el objetivo de esta tarea es obtener una lista de los requisitos de DQ (dimensiones) relacionados a cada funcionalidad. Para ello se tomó como base el modelo MOSCAF (*Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web*) previamente descrito (Tabla 16).

El producto de salida de esta tarea (en base a la Tabla 5) es la *Instanciación del modelo de DQ para el problema particular bajo estudio (IMDQ)* (Tabla 46).

Requisito Funcional	Funcionalidad Web descrita por Collins	Requisito de DQ
RF1. Ejecutar reporte administrativo.	Administración	Compleitud, Precisión.
RF2. Mantener información de empleados	Administración	Compleitud, Conformidad, Precisión.
RF3. Ejecutar pago de nómina	Administración	Precisión.
RF4. Login al sistema	Seguridad	Conformidad, Confidencialidad, Trazabilidad.
RF5. Búsqueda de productos	Capacidades de búsqueda	Consistencia, Conformidad, Trazabilidad.
RF6. Solicitar un producto	Procesos y acciones	Compleitud, Conformidad, Actualidad.
RF7. Escoger método de pago	Administración	Compleitud, Precisión.
RF8. Ejecutar reporte de empleado	Seguridad	Confidencialidad, Trazabilidad.
RF9. Mantener timecard	Procesos y acciones	Compleitud, Conformidad, Actualidad.
RF10. Mantener información de productos	Administración	Confidencialidad, Exactitud, Compleitud, Actualidad, Consistencia, Trazabilidad.
RF11. Mantener órdenes de venta	Procesos y acciones	Compleitud, Conformidad, Actualidad.

Tabla 46. Instanciación del modelo de DQ para el problema particular bajo estudio (IMDQ).

EP. Estructuración de los Puntos de Vista.

➤ EP.1. Clasificación de los puntos de vista (CPVW).

El *nivel de importancia de cada requisito funcional*, considerando la cantidad de veces que cada uno de ellos está relacionado con cada punto de vista es el siguiente (Tabla 47):

Requisito Funcional
1. Login al sistema
2. Mantener información de empleados
3. Ejecutar reporte de empleado
4. Búsqueda de productos
5. Ejecutar pago de nómina
6. Escoger método de pago
7. Mantener timecard
8. Mantener órdenes de venta
9. Solicitar un producto
10. Mantener información de productos
11. Ejecutar reporte administrativo

Tabla 47. Nivel de importancia de los requisitos funcionales.

Tomando como base el nivel de importancia de cada requisito funcional, es posible obtener como producto de salida de esta tarea (en base a la Tabla 6), una lista jerarquizada de los puntos de vista (LJPV), mostrados en la Tabla 48:

Lista jerarquizada de puntos de vista
1. Jefe de departamento
2. Administrador
3. Empleado
4. Cliente
5. Empleado por comisión (éste hereda las funcionalidades del punto de vista "Empleado")
6. Usuario

Tabla 48. Lista jerarquizada de los puntos de vista (LJPV).

➤ EP.2. Clasificación de los requisitos de calidad de datos (CRDQ).

Debido al nivel de criticidad que tienen los requisitos funcionales de la aplicación Web, se determina la clasificación de los distintos requisitos de calidad de datos. Esto permite al analista de DQ, el darse cuenta del nivel de importancia que tiene cada uno de los requisitos de DQ, ayudándolo a una correcta toma de decisión futura, en dado caso que exista algún tipo de conflicto al momento de la implementación de algunos de los requisitos software de DQ.

El producto de salida de esta tarea (tomando como base la Tabla 7) es una *lista de clasificación de los requisitos de calidad de datos (LCRDQ)* (ver Tabla 49).

Requisitos de DQ
1. Conformidad
2. Trazabilidad
3. Confidencialidad
4. Completitud
5. Precisión
6. Consistencia
7. Actualidad
8. Exactitud

Tabla 49. Lista de clasificación de los requisitos de calidad de datos (LCRDQ).

DP. Documentación de los Puntos de Vista.

➤ DP.1. Documentación de los puntos de vista (DPVW).

Para completar esta tarea se utilizó las plantillas mostradas en Tabla 38 y Tabla 39, con el objetivo de documentar de forma conveniente tanto los puntos de vista identificados (ver Tablas 50 a 55) como los requisitos funcionales (ver Tablas 56 a 66). Esta documentación es una parte fundamental de una primera versión del *“documento de especificación de requisitos del sistema aumentado con especificación de requisitos de DQ (ERSDQ)”*.

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Administrador.
Enfoque	Punto de vista del <i>Administrador</i> de la aplicación. Persona encargada principalmente de administrar y controlar las funciones principales de la empresa.
Atributos	Ninguno en específico.
Requisito funcional	RF1. Ejecutar reporte administrativo, RF2. Mantener información de empleados, RF3. Ejecutar pago de nómina, RF4. Login al sistema.
Funcionalidades Web	Administración, Seguridad.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 50. Especificación del punto de vista "Administrador".

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Empleado.
Enfoque	Encargado de desempeñar las funcionalidades básicas dentro de la aplicación Web.
Atributos	ID_empleado, nombre, dirección, teléfono, tipo de empleado, método de pago.
Requisito funcional	RF4. Login al sistema, RF7. Escoger método de pago, RF8. Ejecutar reporte de empleado, RF9. Mantener timecard.
Funcionalidades Web	Seguridad, Administración, Procesos y acciones.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 51. Especificación del punto de vista "Empleado".

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Empleado por comisión.
Enfoque	Este hereda las funcionalidades Web asociadas al punto de vista “Empleado”, además de realizar una de las funcionalidades principales de la empresa “Mantener órdenes de venta”.
Atributos	Mismos atributos que el punto de vista “Empleado” (véase tabla 51)
Requisito funcional	RF11. Mantener órdenes de venta, RF4. Login al sistema, RF7. Escoger método de pago, RF8. Ejecutar reporte de empleado, RF9. Mantener timecard.
Funcionalidades Web	Procesos y acciones, Seguridad, Administración.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 52. Especificación del punto de vista “Empleado por comisión”.

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Jefe de departamento.
Enfoque	Este punto de vista se encarga de mantener actualizada la información de los empleados pertenecientes a su departamento, además de mantener actualizada una de las funciones principales de la empresa, “mantener información de productos”.
Atributos	Ninguno en específico.
Requisito funcional	RF2. Mantener información de empleados, RF8. Ejecutar reporte de empleado, RF4. Login al sistema, RF10. Mantener información de productos.
Funcionalidades Web	Administración, Seguridad, Gestión de contenido.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 53. Especificación del punto de vista “Jefe de departamento”.

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Cliente.
Enfoque	Este punto de vista consiste en toda aquella persona ajena a la empresa, la cual está interesada en llevar a cabo una compra a través de la aplicación Web.
Atributos	ID_cliente, nombre, dirección, teléfono.
Requisito funcional	RF5. Búsqueda de productos, RF6. Solicitar un producto, RF4. Login al sistema.
Funcionalidades Web	Capacidades de búsqueda, Procesos y acciones, Seguridad.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 54. Especificación del punto de vista “Cliente”.

ESPECIFICACIÓN DEL PUNTO DE VISTA	
Referencia	Usuario.
Enfoque	Punto de vista interesado en buscar algún producto específico mediante la aplicación Web.
Atributos	Ninguno.
Requisito funcional	RF5. Búsqueda de productos.
Funcionalidades Web	Capacidades de búsqueda.
Excepciones	Ninguna.
Historia	Sin alteración.

Tabla 55. Especificación del punto de vista "Usuario".

Enseguida se documentan los distintos requisitos funcionales de la aplicación Web (Tablas 57 a 67).

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF4. Login al sistema.
Descripción	Garantizar un acceso seguro de los distintos puntos de vista a la aplicación Web.
Datos	Usuario, password.
Punto de Vista	Administrador, Cliente, Empleado, Jefe de departamento, Empleado por comisión.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Conformidad, Confidencialidad, Trazabilidad.

Tabla 56. Especificación de requisito "Login al sistema".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF2. Mantener información de empleados.
Descripción	Administrar y mantener la información de cada uno de los empleados.
Datos	ID_empleado, nombre, dirección, teléfono, tipo de empleado, método de pago, tipo de empleado (asalariado, por comisión).
Punto de Vista	Administrador, Jefe de departamento.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Conformidad, Precisión.

Tabla 57. Especificación de requisito "Mantener información de empleados".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF8. Ejecutar reporte de empleado.
Descripción	Se encarga de ejecutar un reporte con toda la información de los empleados. Esta información debería sólo ser observada por cada empleado específico y el Jefe de departamento.
Datos	ID_empleado.
Punto de Vista	Empleado, Jefe de departamento.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Confidencialidad, Trazabilidad.

Tabla 58. Especificación de requisito "Ejecutar reporte de empleados".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF5. Búsqueda de productos.
Descripción	Este requisito consiste en llevar a cabo una búsqueda de los productos que vende la empresa.
Datos	nombre del producto, ID_producto.
Punto de Vista	Cliente, Usuario.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Consistencia, Conformidad, Trazabilidad.

Tabla 59. Especificación de requisito "Búsqueda de productos".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF3. Ejecutar pago de nómina.
Descripción	Este requisito se encarga de ejecutar un procedimiento específico para realizar el pago de la nómina de todos los empleados de la empresa, calculando el pago dependiendo del tipo de empleado (asalariado o por comisión).
Datos	Pago Mensual / Semanal.
Punto de Vista	Administrador.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Precisión.

Tabla 60. Especificación de requisito "Ejecutar pago de nómina".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF7. Escoger método de pago.
Descripción	El empleado será capaz de seleccionar su método de pago: envío de cheque a domicilio, depósito en cuenta bancaria o recoger cheque en oficina.
Datos	ID_empleado, método de pago.
Punto de Vista	Empleado, Empleado por comisión.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Precisión.

Tabla 61. Especificación de requisito "Escoger método de pago".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF9. Mantener timecard.
Descripción	La aplicación proveerá la funcionalidad al empleado de ingresar sus horas de trabajo.
Datos	fecha, horas de trabajo, número de cargo.
Punto de Vista	Empleado, Empleado por comisión.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Conformidad, Actualidad.

Tabla 62. Especificación de requisito "Mantener timecard".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF11. Mantener órdenes de venta.
Descripción	Este requisito consiste en mantener actualizada toda la información de cada una de las órdenes de venta de los productos vendidos por empleados específicos .
Datos	fecha, ID_producto, ID_empleado, total de venta.
Punto de Vista	Empleado por comisión.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Conformidad, Actualidad.

Tabla 63. Especificación de requisito "Mantener órdenes de venta".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF6. Solicitar un producto.
Descripción	Este requisito proveerá la funcionalidad al cliente de solicitar un producto específico para llevar a cabo una compra.
Datos	ID_producto, fecha, costo, ID_cliente.
Punto de Vista	Cliente.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Conformidad, Actualidad.

Tabla 64. Especificación de requisito "Solicitar un producto".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF10. Mantener información de productos.
Descripción	Este requisito consiste en mantener actualizada la información relacionada a cada uno de los productos que la empresa vende.
Datos	ID_producto, nombre, categoría, precio.
Punto de Vista	Jefe de departamento.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Confidencialidad, Exactitud, Compleitud, Actualidad, Consistencia, Trazabilidad.

Tabla 65. Especificación de requisito "Mantener información de productos".

ESPECIFICACIÓN DEL REQUISITO	
Referencia	RF1. Ejecutar reporte administrativo.
Descripción	Esta funcionalidad proveerá al Administrador de la aplicación para llevar a cabo un reporte administrativo, en el cual podrá observarse información como: total de ventas por semana/mes/año, total de ventas de cada empleado por comisión, total de horas facturadas a cada proyecto.
Datos	fecha inicial, fecha final, ID_empleado, ID_proyecto.
Punto de Vista	Administrador.
Requisitos no funcionales	Ninguno.
Requisitos de DQ	Compleitud, Precisión.

Tabla 66. Especificación de requisito "Ejecutar reporte administrativo".

➤ DP.2. Documentación de los requisitos de calidad de datos (DRDQ).

Como parte del producto de salida de esta tarea (basándose en la Tabla 9), y con el objetivo de documentar y modelar de una forma más clara y específica cada uno de los requisitos de DQ, se utilizó el *metamodelo DQ_WebRE* (descrito en la sección 3.2.6). Este metamodelo nos permite la incorporación y el modelado de los distintos requisitos de DQ asociados a los diferentes requisitos funcionales que proveerá la aplicación Web.

Haciendo uso del metamodelo *DQ_WebRE* se obtuvo los siguientes *diagramas de casos de uso* (ver Figura 32, Figura 33 y Figura 34). En estos diagramas se muestran los diferentes requisitos funcionales asociados a cada punto de vista (Actor).

Además, se observa la especificación de distintos casos de uso específicos, los cuales son estereotipados como “**InformationCase**” (*Casos de Información*). Estos “*Casos de información*” son los encargados de gestionar todos los *datos* involucrados en los distintos casos de uso (requisitos funcionales) definidos para cada punto de vista.

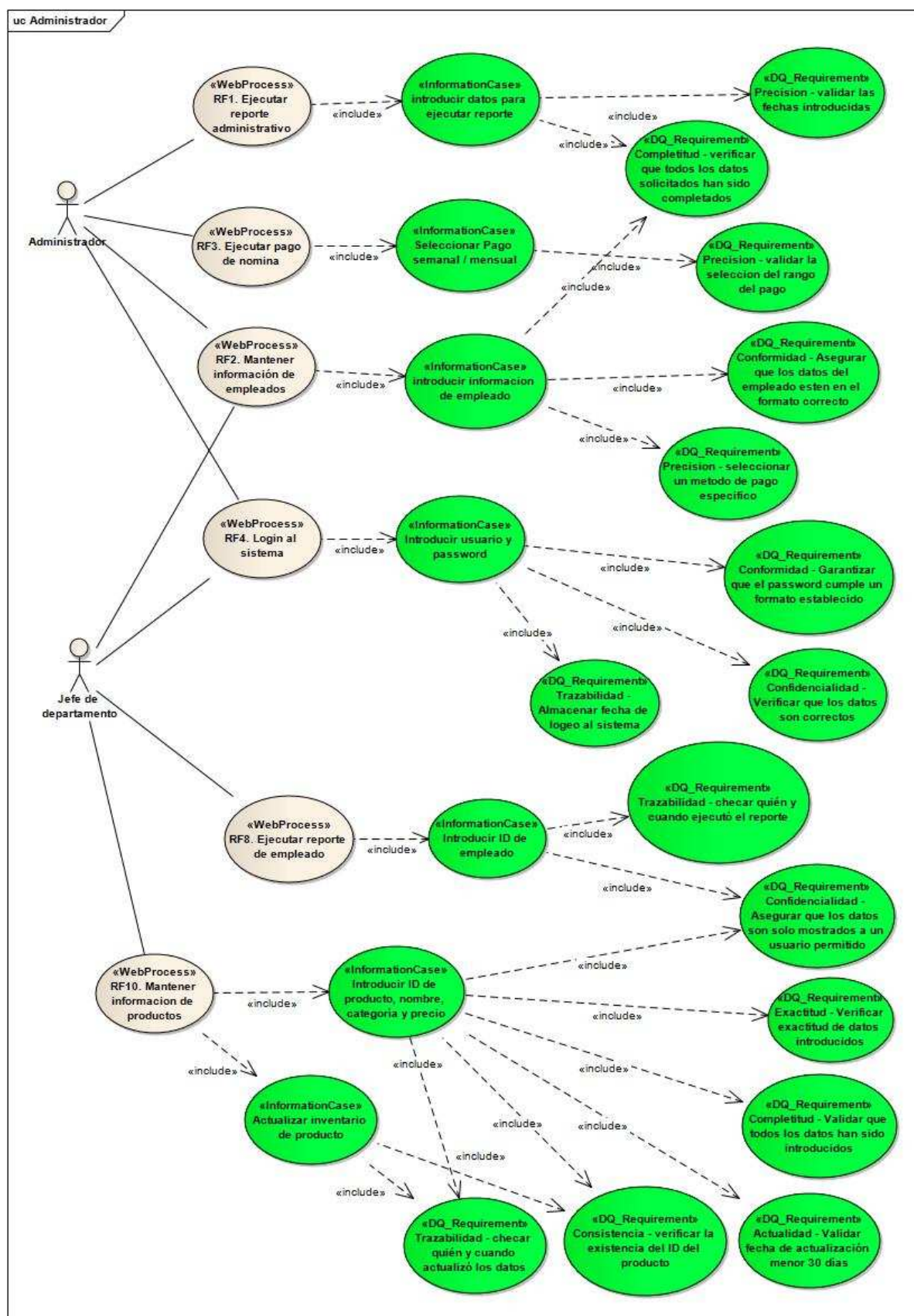


Figura 32. Diagrama de casos de uso de los puntos de vista “Administrador” y “Jefe de departamento”, especificando requisitos de DQ.

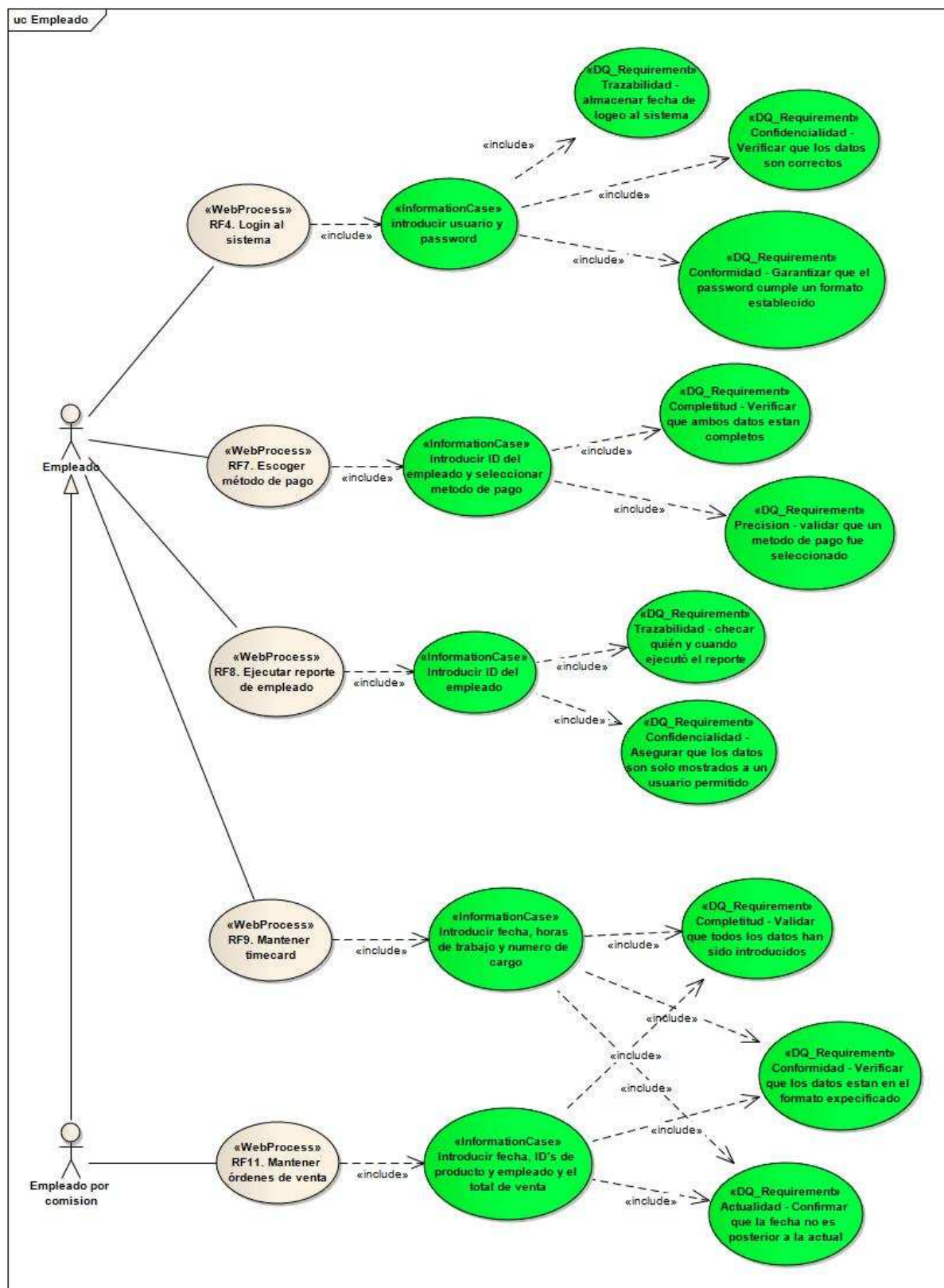


Figura 33. Diagrama de casos de uso de los puntos de vista "Empleado" y "Empleado por comision", especificando requisitos de DQ.

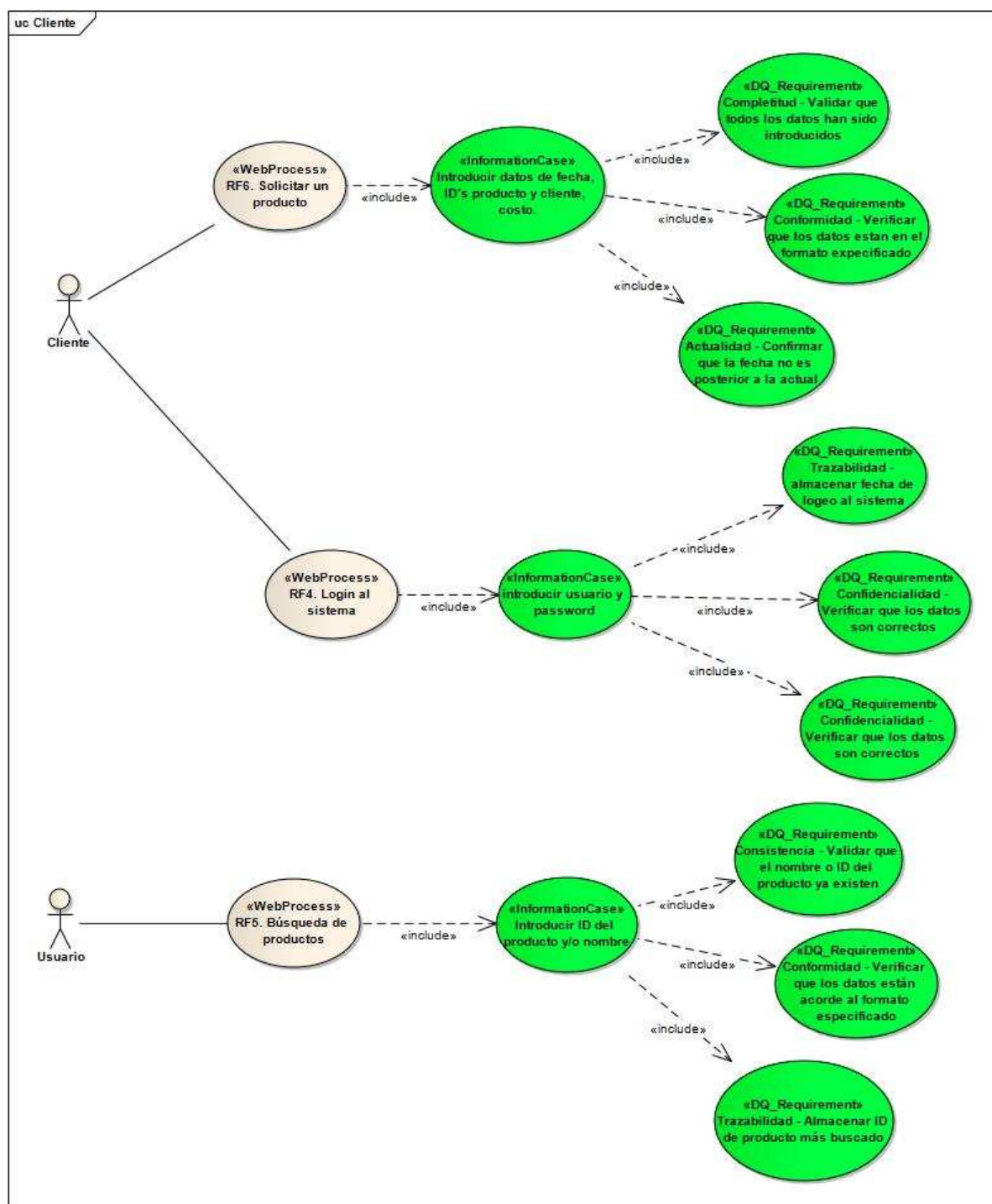


Figura 34. Diagrama de casos de uso de los puntos de vista "Cliente" y "Usuario", especificando requisitos de DQ.

Con el objetivo de mostrar de mejor manera la especificación de los requisitos de DQ, enseguida se describe ampliamente el requisito funcional **"RF10. Mantener información de productos"**, el cual tiene asociado la mayor cantidad de requisitos de calidad de datos.

En este caso de uso **"RF10. Mantener información de productos"** (asociado al punto de vista **"Jefe de departamento"**), se observa que incluye dos **casos de información** específicos: **"Introducir ID de producto, nombre, categoría y precio"** y

“Actualizar inventario de producto”.

Habiendo especificado los diferentes casos de información, se procedió a la especificación y el modelado de los requisitos de DQ relacionados a cada uno de los elementos del tipo *“InformationCase”*.

Como se puede observar, los siguientes **requisitos de DQ** fueron especificados: *“Confidencialidad-Asegurar que los datos son solo mostrados a un usuario permitido”*, *“Exactitud-Verificar exactitud de datos introducidos”*, *“Compleitud-Validar que todos los datos han sido introducidos”*, *“Actualidad-Validar fecha de actualización menor 30 días”*, *“Consistencia-Verificar la existencia del ID del producto”* y *“Trazabilidad-Checar quién y cuándo actualizó los datos”*.

Una vez completado el diagrama de casos de uso *“RF10. Mantener información de productos”*, modelando además sus requisitos específicos de DQ, y con el afán de llevar a cabo una descripción más detallada de ellos, enseguida se muestra el diagrama de actividades, usando los elementos correspondientes (estereotipados) definidos en el metamodelo *DQ_WebRE*.

En este diagrama de actividades (ver Figura 35), el Analista modela las actividades específicas para cumplir los requisitos de DQ previamente definidos. Estas actividades están relacionadas a los diferentes elementos los cuales son específicos para el desarrollo de una aplicación Web.

Las primeras actividades *“Agregar metadatos relacionados a confidencialidad”* y *“Agregar metadatos relacionados a actualidad”* (estereotipados como *“Add_DQ_Metadata”*), son los responsables de la captura de los metadatos relacionados a la Confidencialidad (*“disponible_para”* y *“nivel_seguridad”*) y Actualidad (*“fecha_de_actualizacion”*). Estos metadatos serán almacenados en una instancia de la clase *“DQ_Metadata”*, la cual será posteriormente utilizada para satisfacer los requisitos de DQ de *Confidencialidad* y *Actualidad* respectivamente.

Por otra parte, la actividad *“Agregar metadato relacionado a Consistencia”*, capturará el correspondiente metadato (*“producto_registrado”*) el cual es almacenado en una clase específica de tipo *“DQ_Metadata”*. El valor de este metadato indicará si el producto ya existe al momento de actualizar el inventario. Es válido mencionar que todas estas clases de tipo *“DQ_Metadata”* están relacionadas a los *datos* (almacenados en instancias de tipo *“Contents”*) gestionados en las actividades previas de *“Introducir información de producto”* y *“Actualizar inventario de producto”* estereotipadas como *“UserTransaction”* (ver comentario en Figura 35).

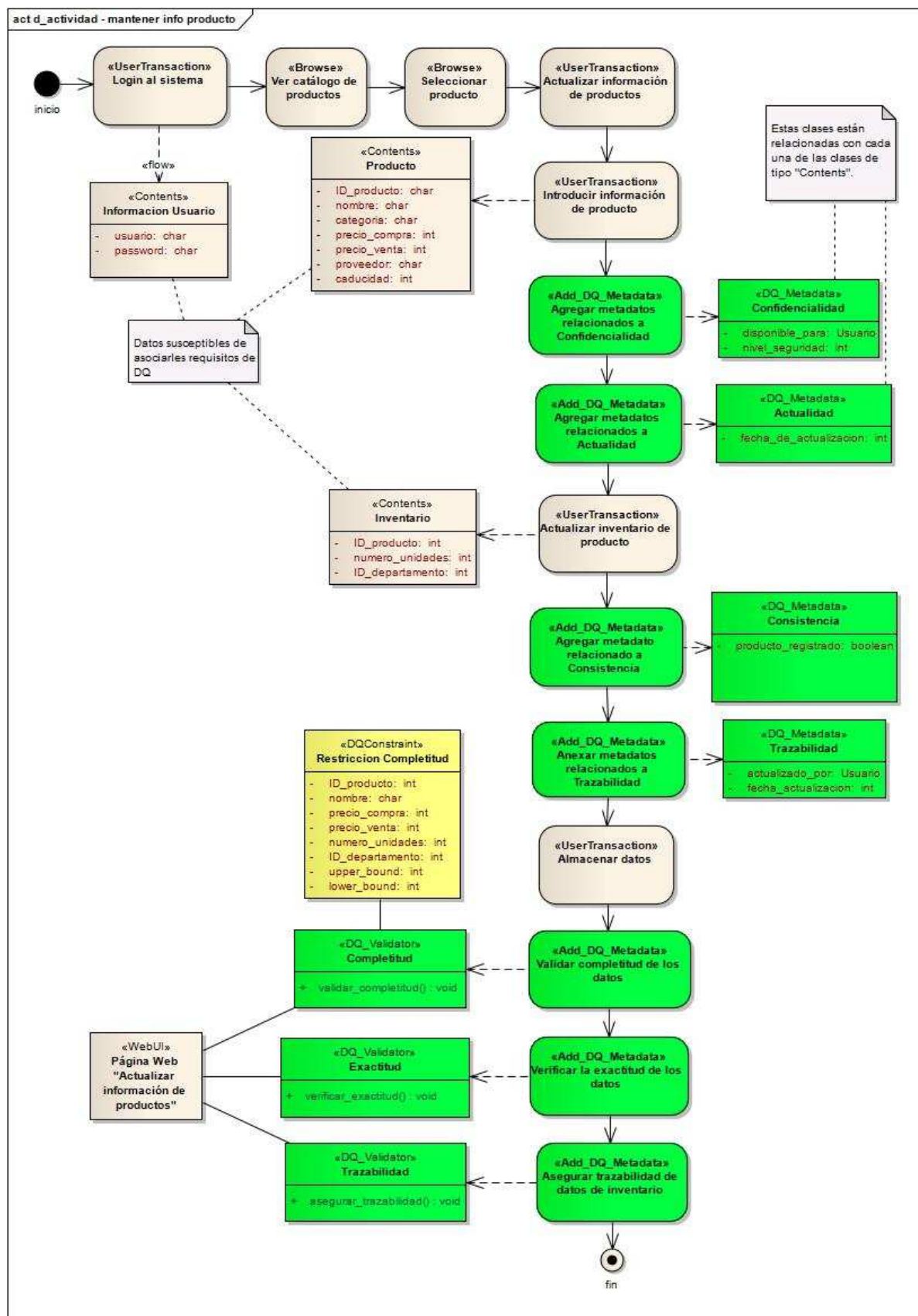


Figura 35. Diagrama de actividades con gestión específica de DQ.

La actividad "Anexar metadatos relacionados a Trazabilidad" está a cargo de capturar los metadatos "actualizado_por" y "fecha_actualizacion" (almacenados en una instancia de tipo "DQ_Metadata"), relacionados para el cumplimiento del

requisito de DQ específico de Trazabilidad.

Finalmente, las actividades “*Validar completitud de los datos*”, “*Verificar la exactitud de los datos*” y “*Asegurar trazabilidad de datos de inventario*”, serán las responsables de agregar las operaciones específicas “*validar_completitud()*”, “*verificar_exactitud()*” y “*asegurar_trazabilidad()*”, con el objetivo de verificar y comprobar la Completitud, Exactitud y Trazabilidad de los datos gestionados en el elemento “*Página Web-Actualizar información de productos*” del tipo “*WebUI*”.

Los datos gestionados dentro del elemento “*Restriccion Completitud*” (estereotipado como “*DQConstraint*”) son usados para especificar los datos que son forzosamente requeridos de ser completados, además de especificar los límites de completitud que se deben satisfacer, a través de los valores especificados en “*upper_bound*” y “*lower_bound*”. La operación “*validar_completitud*” es la encargada de verificar que todos estos datos sean introducidos a través del elemento “*Página Web-Actualizar información de productos*” (estereotipado como “*WebUI*”).

Al término de esta tarea, obtenemos el **Documento completo de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)**.

TPS. Trazado de los Puntos de Vista del Sistema

➤ TPS.1. Modelado de los requisitos software de calidad de datos (MRDQ).

Como parte del producto de salida de esta tarea (basándose en la Tabla 10) y con el objetivo de llevar a cabo una trazabilidad de los requisitos de DQ previamente modelados hacia un modelo de diseño y un modelo de datos (considerando las reglas de transformación propuestas en la Tabla 36), en este punto se muestran los diagramas de diseño obtenidos (*modelo de contenido*, *modelo de navegación* y *modelo de presentación*) específicos del caso de uso “*RF10. Mantener información de productos*”. Estos diagramas de diseño fueron modelados haciendo uso del metamodelo *DAQ_UWE* (descrito en la sección 3.2.7).

Tomando como base los diferentes elementos estereotipados modelados en el diagrama de actividad anterior (Figura 35) y considerando las reglas de transformación, se obtienen distintos elementos modelados en los siguientes diagramas de diseño.

El *Modelo de Contenido* es mostrado en la Figura 36, con las clases definidas para el almacenado de la información principal relacionada con “*Producto*” e “*Inventario*” (estereotipadas como “*ContentClass*”). Este modelo también muestra las

clases relacionadas a la definición de cada una de las características (requisitos software) de DQ (estereotipadas como “DQDim”): *Actualidad*, *Confidencialidad*, *Consistencia*, *Completitud*, *Exactitud* y *Trazabilidad*, así como de sus respectivos atributos (del tipo “DQDimProperty”) y operaciones definidas. Asimismo, es modelada la clase “*Restricción Completitud*” del tipo “DQ_Constraint”, la cual es encargada de almacenar los datos usados para almacenar las restricciones y límites definidos para satisfacer el requisito de Completitud.

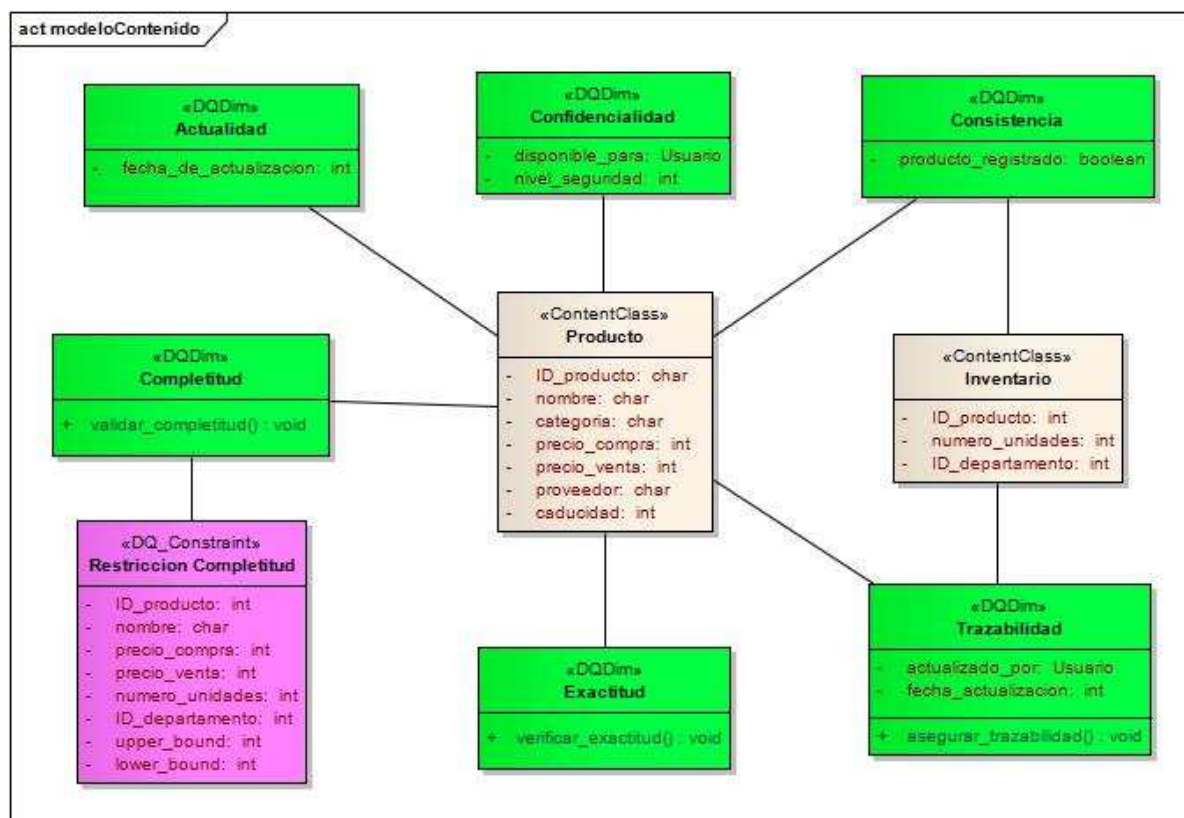


Figura 36. Modelo de Contenido con elementos para la Gestión de DQ.

Haciendo también uso de los elementos estereotipados definidos en el metamodelo *DAQ_UWE*, es posible modelar la estructura de hipertexto, obteniendo de esta manera el siguiente *Modelo de Navegación* (ver Figura 37). Este diagrama muestra una clase de navegación “*Ver catálogo de productos*”, una clase index “*Seleccionar producto*” y una clase menú “*Actualizar información de productos*”. Todos los datos introducidos en la aplicación son modelados mediante las clases “*Introducir información de producto*” y “*Actualizar inventario de producto*”, de tipo “*ProcessClass*”.

La gestión de la calidad de datos es análogamente modelada mediante los siguientes elementos estereotipados como “*DQProcessClass*”: “*Agregar metadatos relacionados a confidencialidad*”, “*Agregar metadatos relacionados a actualidad*”, “*Agregar metadato relacionado a consistencia*”, “*Anexar metadatos relacionados a trazabilidad*”, “*Validar completitud de los datos*”, “*Verificar la exactitud de los datos*” y

“Asegurar trazabilidad de datos de inventario”. Todos estos elementos se encuentran relacionados a las distintas clases del tipo “processClass” a través de relaciones específicas del tipo “DQLink”. Cabe recordar que todos estos elementos surgieron a partir de los elementos del tipo “Add_DQ_Metadata” modelados en el diagrama de actividad definido previamente.

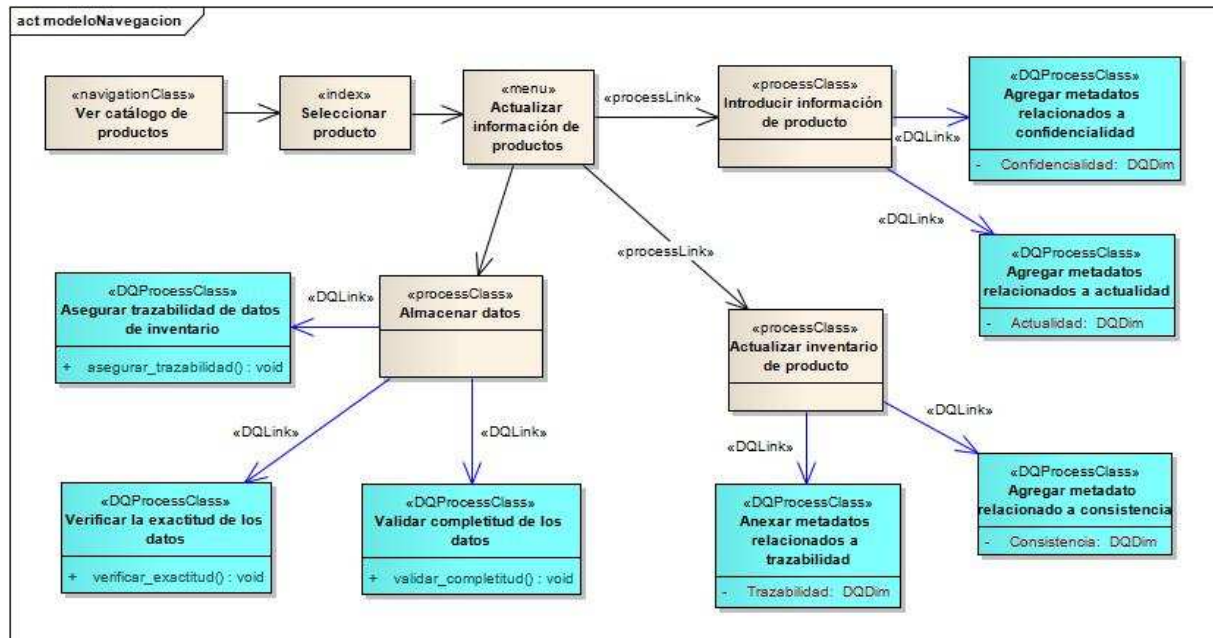


Figura 37. Modelo de Navegación con elementos para la Gestión de DQ.

Finalmente, la Figura 38 muestra el *Modelo de Presentación* obtenido. La forma tipo “Container” fue seleccionada con el objetivo de proveer una representación más intuitiva de la página Web. La página de presentación “Actualizar información de productos” (del tipo “presentationPage”) contiene dos grupos de presentación (del tipo “presentationGroup”): “Introducir información de producto” y “Actualizar inventario de producto”.

Diferentes elementos son mostrados en cada grupo de presentación, por ejemplo, los siguientes elementos de tipo “textInput” son modelados en el grupo de presentación “Introducir información de producto”: *ID_producto*, *nombre*, *categoría*, *precio_compra*, *precio_venta*, *proveedor* y *caducidad*. Asimismo, en el grupo de presentación “Actualizar inventario de producto” se modelan los elementos “ID_producto” y “numero_unidades” del tipo “textInput”, y el elemento “ID_departamento” del tipo “selection”.

En este modelo de presentación, también está modelado los requisitos de DQ mediante distintas clases estereotipadas como “UIE_DQVerifier”. Se puede observar que la clase “Completitud” está relacionada a la página de presentación (“presentationPage”) “Actualizar información de productos”. Esto significa que esta clase está encargada de validar la completitud (a través de la operación

“*validar_completitud()*”) de todos los datos introducidos en cada uno de los elementos contenidos en este grupo, representando de esta forma el cumplimiento del requisito de DQ de Completitud.

La clase “*Exactitud*” (estereotipada como “*UIE_DQVerifier*”), de forma similar estará a cargo de ejecutar la operación específica “*verificar_exactitud()*”, con el fin de verificar que todos los datos introducidos en cada uno de los elementos de la página sean exactos, asegurando de esta manera el requisito de calidad de datos relacionado con la Exactitud.

Finalmente, la clase “*Trazabilidad*” es la responsable de comprobar que la trazabilidad de los datos introducidos en los elementos del grupo de presentación (presentationGroup) “*Actualizar inventario de producto*”, mediante la operación definida “*asegurar_trazabilidad()*”. En este caso específico, el requisito de trazabilidad verifica que la información acerca de “quién” y “cuándo” modificó los datos del inventario hayan sido almacenados correctamente.

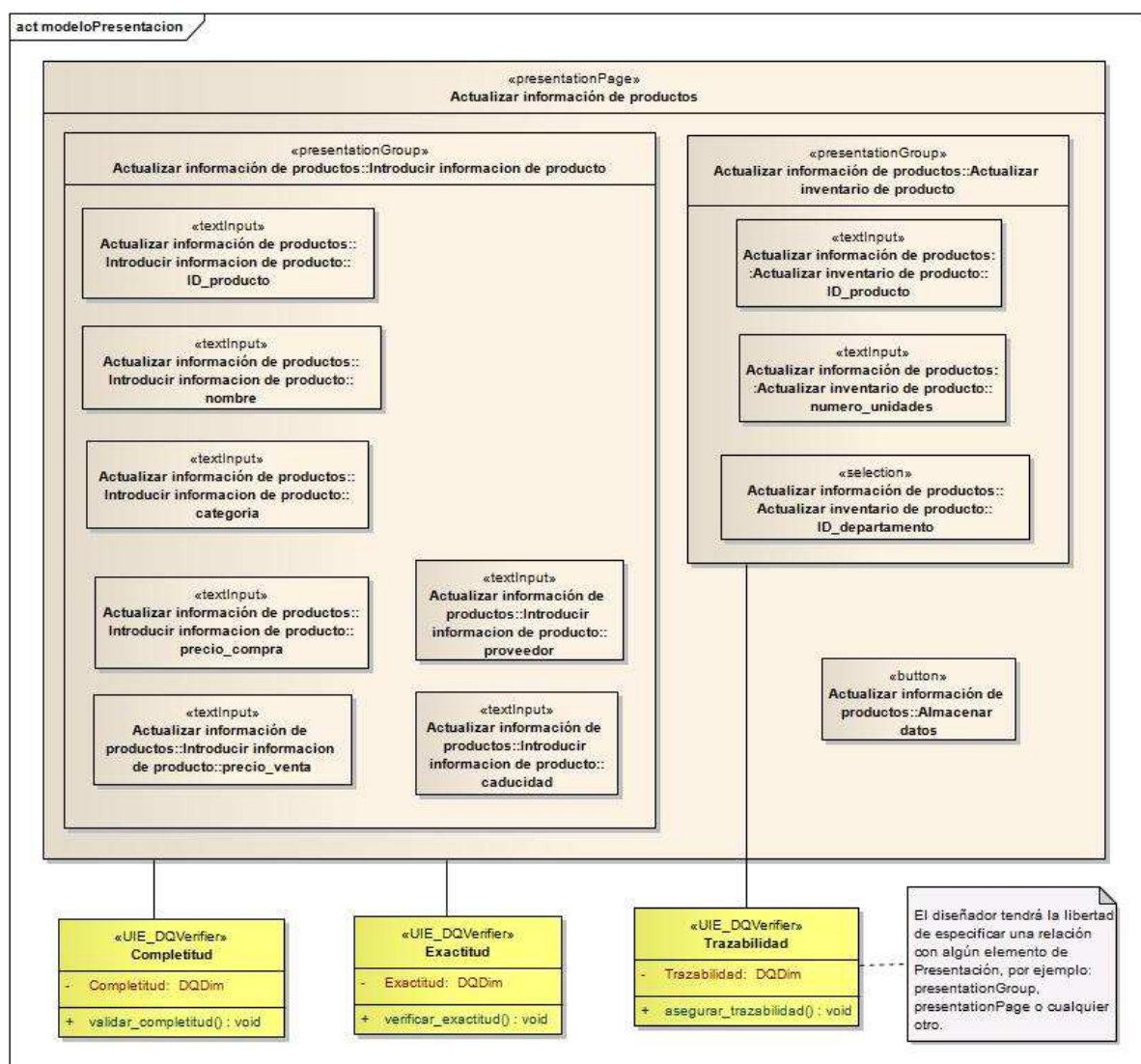


Figura 38. Modelo de Presentación incluyendo elementos para la Gestión de DQ.

Es válido resaltar que en un momento dado, la persona encargada de modelar este modelo de presentación puede tener la libertad de asociar un requisito específico de DQ a un determinado elemento específico, como pudiera ser en este caso al elemento “*ID_producto*” o “*numero_unidades*” de tipo *textInput*.

De manera similar que en los modelos anteriores (*Contenido y Navegación*), todos los elementos del tipo “*UIE_DQVerifier*” fueron derivados de los elementos del tipo “*DQ_Validator*” modelados en el diagrama de actividades (Figura 35).

Al término de esta tarea, obtenemos el **Documento del Modelo de Diseño sensible a calidad de datos (MDSDQ)**. Este documento contiene todo el conjunto de diagramas de diseño obtenidos a partir de los distintos diagramas de casos de uso y actividades modelados previamente.

➤ TPS.2. Validación de los resultados de la metodología (VM).

El resultado de esta última tarea de la metodología es la validación de los dos documentos principales obtenidos: “*Documento completo de Especificación de Requisitos del Sistema aumentado con especificación de requisitos de DQ (ERSDQ)*” y “*Documento del Modelo de Diseño con conciencia de calidad de datos (MDSDQ)*”. Dichos documentos son validados y aceptados de forma específica tanto por el Arquitecto jefe como por el Jefe de desarrollo.

4.2 Lecciones Aprendidas

En esta sección se mencionan un listado de lecciones aprendidas una vez aplicada la metodología DQ-VORD en el caso de estudio de validación.

- Inicialmente la metodología DQ-VORD fue concebida con 6 actividades principales, sin embargo, una vez aplicada la metodología a un primer caso de validación, nos dimos cuenta de que dos de ellas podrían unirse, basándonos fundamentalmente en dos hechos importantes: el primero, dos de las tareas definidas en la primera versión de la metodología perseguían objetivos similares, lo que podría llegar a ocasionar algún tipo de conflicto al momento de ejecutarlas; el segundo hecho era que el número de roles que había que involucrar hacía que la aplicabilidad de la metodología en un contexto empresarial decayera ostensiblemente, debido a la cantidad de recursos requeridos para su ejecución.
- Inicialmente la metodología no proponía ningún tipo de trazabilidad de los

diagramas de análisis hacia modelos de diseño, y por eso surgió la necesidad de proponer un conjunto de reglas de mapeo entre ellos.

- Una vez aplicando el conjunto de reglas de mapeo, nos dimos cuenta de la *reducción del esfuerzo* llevado a cabo por la persona encargada, al momento de obtener los distintos modelos de diseño, con lo que constatamos su necesidad.
- El seguimiento de las reglas de mapeo, genera una mayor confiabilidad en los diagramas de diseño obtenidos. Así mismo, esta tarea puede generar una reducción en el posible número de errores al modelar los correspondientes diagramas de diseño.
- Originalmente en la metodología, no se contemplaba ninguna herramienta para llevar a cabo el modelado de los diferentes diagramas, sin embargo, al momento de proponer una primera versión de la metodología en un ejemplo de aplicación, se recibió como comentario la necesidad de desarrollar una herramienta que facilitara la realización de los diagramas tanto de análisis como de diseño.
- Una vez implementado los perfiles presentados en la tesis (*DQ_WebRE* y *DAQ_UWE*) como extensión de la herramienta comercial, y considerando que posiblemente alguna persona no tuviera el acceso a una herramienta comercial para llevar a cabo el uso de los perfiles desarrollados, se consideró la opción de desarrollar un plugin en la plataforma de Eclipse, bajo el término EPL (Eclipse Public License).
- Resulta necesario que todas las personas involucradas en la ejecución de las distintas actividades y tareas de la metodología, tengan un conocimiento pleno de ellas. Lo cual generaría un ahorro en el tiempo de formación de las personas.
- Es imprescindible determinar el tiempo y costo necesario para aplicar la metodología DQ-VORD, lo cual permitirá al encargado de un proyecto de desarrollo software, decidir su aplicabilidad en un entorno real dentro de una empresa. Asimismo, es necesario cuantificar de alguna manera los aspectos relacionados a riesgos y recursos utilizados al aplicar la metodología.
- Al momento de emplear la metodología, resulta básico un soporte que pueda ser capaz de automatizar todas las actividades y tareas de la metodología. Esto permitiría llevar un mejor seguimiento y control en cada una de ellas por parte de todas las personas involucradas.

- Debería existir un compromiso por parte de la empresa interesada en aplicar la metodología DQ-VORD, debido a que su aplicación requerirá la asignación de recursos humanos por un periodo de tiempo a las distintas tareas.

5 Conclusions and Future Research Lines

“Un hombre verdaderamente rico, es aquel cuyos hijos corren a sus brazos aún cuando tiene las manos vacías”
– Anónimo

This chapter lists and discusses the final results of work done in this thesis. Initially, an analysis of achieved objectives is shown, in addition to mentioning the main obtained results and their contrast through scientific publication. Finally, it describes possible lines for future work.

5.1 Analysis of achieved objectives

In the first chapter is showed the main objective of this thesis, which was specified as:

Define a work environment that allows the incorporation of requirements for the management of data quality in Web applications, through a work environment that includes both methodological and technological aspects.

The following is an analysis of each of the objectives achieved:

Objective 1. Study the state of the art of the key areas that supports this thesis: Data Quality, Web Engineering (Model Driven Web Engineering) and Requirements Engineering.

It was carried out extensive research in these areas, in order to obtain an overview of them, so it that would possible to obtain the necessary background and expertise. It is presented in concrete form in section 2.

Objective 2. Conducting a comparative analysis of the different published works which proposed some kind of solution for DQ requirements management in information systems development.

We conducted a systematic literature review of all related published works, based on the review protocol proposed by Biolchini et al. in (Biolchini, Mian et al. 2007). The obtained result was both a description as a comparative analysis of the main features found in each proposal. Both the description of each stage of the systematic review and analysis of each proposal are shown in Annex A.

Objective 3. Defining the concept of DQ requirement and the DQ software requirement, as well as the definition of a Methodology for Specifying Data Quality Requirements in Web applications. Also, the proposal of a model for the selection of data quality requirements depending on the functionalities to be implemented in Web applications.

It was carried out the definition of DQ requirement concept, because after reviewing the literature, there was no definition for this concept, although this is one of the key concepts within the area of data quality (section 3.1).

As core part of this research work, it was proposed a methodology (DQ_VORD), which main objective was to propose a guide to be used as the basis for Web application developers, in the specification of specific DQ requirements. This

methodology consists of four main activities, each of which is broken down into different specific tasks, on which are involved different roles, and are defined some input and output products, also are mentioned various techniques and tools. In addition, a set of templates for different documents (input and output products) that are managed throughout the implementation of various activities and tasks of the methodology were proposed. The DQ_VORD methodology is described in depth in section 3.2.

Moreover, and as a technique usable in the application of DQ-VORD, the model (MOSCAF) was developed, through this model is specified a set of relationships between different DQ dimensions (proposed in standard ISO 25012) and the Web functionalities proposed by Collins. This model has as main purpose facilitating to analyst, the selection of DQ dimensions that should be incorporated as part of the DQ requirements specification, these DQ requirements should be considered while implementing the different Web functionalities. This model is presented in depth in section 3.2.5.

Objective 4. Specification of a metamodel and a UML profile for the specification and the modeling of data quality concepts in the stage of requirements specification.

It was defined a set of elements considered essential for DQ requirements specification, in addition to the specification of an UML profile. All this based on the features and benefits proposed by the approach of model-driven software development (MDSD). Each of these elements allows proper management of DQ requirements. The detailed description of both the metamodel and the profile proposed (DQ_WebRE) is presented in section 3.2.6.

Objective 5. Specification of a metamodel and a UML profile for modeling of elements related with data quality in design stage.

It was carried out the definition of different specific elements for managing aspects of data quality during the design stage, considering the model-driven software development approach. These elements are essentials when modeling different design diagrams as: *Navigation*, *Presentation* and *Content*. The definition and specification of the metamodel proposed (DAQ_UWE) is showed in section 3.2.7.

Objective 6. Developing the tools that allows specifying and modeling the data quality elements proposed, and validate the above proposals through a validation case study.

It was carried out the implementation of both metamodels (“DQ_WebRE” and

"DAQ_UWE") and their respective UML profiles in the tool "Enterprise Architect", with the objective of making them operatives. Examples of the implementation can be found in sections 3.2.6.4 and 3.2.7.3, also of section 4.

Besides, a plug-in was developed using the approaches of EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework) for its implementation. This plug-in allows the modeling of use case and activity diagrams, it allows modeling DQ requirements through using specific elements related with managing of data quality (DQ_WebRE).

On the other hand, validation of proposals for this thesis work was carried out by applying the DQ-VORD methodology in a validation case study. In section 4 is described in depth, the implementation of each one of activities and tasks defined in the methodology, getting output products in each one. This allowed to get a first approximation of applicability of the methodology proposed. Thus achieving a favorable outcome at the time to be evaluated by a panel of academic experts in the area.

It is worth to highlight at this point, that both methodology and the validation case study were published in international conferences focused in the data quality area, besides of indexed journals, it is showed in section 5.3.

5.2 Main contributions of this thesis

The main contributions of this work thesis are:

- o The definition of the concept of data quality requirement, it is based in the state of art, and the definition of concept of data quality software requirement based in standard ISO 25012.
- o The definition of a Methodology for managing of projects of data quality requirements specification in Web applications, DQ_VORD. This methodology includes:
 - The Specification of a Model for the selection of data quality requirements depending on the functionalities to be implemented in a Web application - MOSCAF.
 - Developing and implementation of a extended metamodel for the specification of DQ requirements in Web applications, DQ_WebRE.
 - Developing and implementation of a metamodel for the design of data quality awareness Web applications, DAQ_UWE.

5.3 Contrast of results

The work of this thesis has been contrasted with the research community through the publication of the results in leading conferences and journals both national and international.

JCR Journals.

- o “Capturing Data Quality Requirements for Web Applications by means of DQ_WebRE”. *Guerra-García César, Caballero Ismael, Piattini M.* In Information Systems Frontiers Journal. Aceptado
- o “MOSCAF: A Model for the Selection of Data Quality Requirements for the Functionalities in a Web Application”. *Guerra-García César, Caballero Ismael, Ayala Claudia, Piattini M.* Enviado
- o “A Methodology for Managing Data Quality Requirements in Information Systems Development”. *Guerra-García César, Caballero Ismael, Piattini M.* PENDIENTE DE ENVIAR

Other International Journals.

- o “Calidad de Datos en Aplicaciones Web, Conceptos y Enfoques”. *Guerra-García César, Caballero Ismael.* In Revista Software Guru. Vol. 30. 2010. ISSN: 1870-0888.
- o "Introducing Data Quality Requirements into the Web Portal Development Process". *Guerra-García César, Caballero Ismael, Piattini M.* In Research in Computing Science Journal. Advances in Computer Science and Applications. 2009. Vol. 43: p. 81-95.

International Conferences.

- o “Designing Data Quality Aware Web Applications”. *Guerra-García César, Caballero Ismael, Berti-Equille Laure, Piattini M.* In International Conference on Information Quality, **ICIQ 2011**. Adelaide, Australia.
- o “A Systematic Literature Review of How to Introduce Data Quality Requirements into a Software Product Development”. *Guerra-García César, Caballero Ismael, Piattini M.* In 5th. International Conference on Evaluation of

Novel Approaches to Software Engineering, ENASE. 2010. Athens, Greece.

Publicado también como: "A Survey on How to Manage Specific Data Quality Requirements during Information System Development". *Guerra-García César, Caballero Ismael, Piattini M.* In Evaluation of Novel Approaches to Software Engineering. Lecture Notes in Computer Science, 2011.

- o "DQ-VORD: A Methodology for Managing and Integrating Data Quality Requirements into Software Requirement Specification". *Guerra-García César, Caballero Ismael, Piattini M.* In IADIS International Conference WWW/INTERNET 2009. 2009. Rome, Italy.
- o "Inserción de Requerimientos de Calidad de Datos en el Proceso de Desarrollo de Portales Web". *Guerra-García César, Caballero Ismael, Piattini M.* In Simposio de Ingeniería de Software 2009 (SIS '09). 2009. México.

International Workshops.

- o "Sentando las bases para definir un modelo de Calidad de Datos para los artefactos usados en el Proceso de Planificación de un Proyecto de Desarrollo de Software". *Bernabé Oscar, Caballero Ismael, Guerra-García César, Piattini M.* In Second Ibero-American Workshop on Data Quality, IAWDQ 2011. Chaves, Portugal. Publisher: IEEE Computer Society. "Building the pillars for the definition of a data quality model to be applied to the artifacts used in the Planning Process of a Software Development Project".
- o "Capturing Data Quality Requirements for Web Applications by means of DQ_WebRE". *Guerra-García César, Caballero Ismael, Piattini M.* In 2nd. International Workshop on Business Intelligence and the Web, BEWEB 2011. ACM 978-1-4503-0610-2/11/03. Publisher: ACM.
- o "Managing Data Quality Requirements for Web Portals Development: an Example of Application". *Guerra-García César, Caballero Ismael, Piattini M.* In First Ibero-American Workshop on Data Quality, IAWDQ 2010. 2010. Santiago de Compostela, España. Publisher: IEEE Computer Society.
- o "Developing Data Quality Aware Applications". *Pérez Ricardo, Guerra-García César, Caballero Ismael, Piattini M, Moraga, M^a. Ángeles.* In 1st. International Workshop on Data Quality for Software Engineering (DQ4SE 2009). 2009. Jeju, Korea. Publisher: IEEE Computer Society.

National Conferences.

- o “Entorno de Trabajo para la Gestión de Requisitos de Calidad de Datos en Aplicaciones Web”. *Guerra García César Arturo*. In 1as. Jornadas Doctorales de Castilla-La Mancha. ISBN: 978-84-693-9605-6. 2011.
- o “Modelado de Requisitos de Calidad de Datos en Ingeniería Web”. *Guerra-García César, Caballero Ismael, Piattini M.* In XV Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2010. 2010. Valencia, España.
- o “*Modelado de Requisitos de Calidad de Datos en el Proceso de Desarrollo de Portales Web*”. Guerra-García César, Caballero Ismael, García-Rodríguez de Guzmán Ignacio, Piattini M. In VI Taller sobre Desarrollo de Software Dirigido por Modelos (DSDM'09) 2009. San Sebastián, España.

5.4 Future Work

We can say that the areas considered in this dissertation are very recent. They have been evolving considerably in recent years. In this way, some of the proposed research as future work are shown below:

- o Investigate new international standards related both to data quality (for instance, (ISO/TS--8000-100 2011)) and Web Engineering. In order to include them to the metamodels proposed. The objective is to propose new elements that allow to improve the managing of DQ requirements during the complete process of Web application development.
- o The development of a tool that allows to receive as an input element to the different diagrams generated with the metamodels proposed, thus allowing to execute the corresponding transformations between them, and finally the semiautomatic generation of code. These transformations could be developed through the QVT language (Query/View/Transformation) (OMG 2008).
- o Pursue case studies for implementing the DQ-VORD methodology on real projects of requirements specification, within software development companies. In order to obtain more feedback from users and Developers, and to propose improvements and/or changes in the various activities or tasks of the methodology.

Some possible limitations and/or weaknesses that shows the DQ-VORD

methodology proposed in this thesis are:

- The methodology does not include all stages of the full development cycle of an application.
- Increased training time by the persons (roles) involved in the execution of the activities of the methodology.
- The lack of a complete tool, capable of automating the automatic generation of code, starting from requirements and design diagrams generated by the metamodels proposed.
- The validation of the methodology in a real environment within a company.

6.1 Anexo A. Revisión sistemática realizada.

La Revisión Sistemática (RS) consiste en una metodología científica específica que va un paso delante de una simple búsqueda de bibliografía. La RS se enfoca en integrar investigación empírica con el objetivo de crear generalizaciones. Este reto de integración involucra objetivos específicos, que permiten al investigador analizar de forma crítica los datos obtenidos, resolver conflictos detectados en el material de literatura, e identificar aspectos para planificar posibles investigaciones futuras (Biolchini, Mian et al. 2007).

Las principales fases (Figura 39) del protocolo de Revisión Sistemática propuesto por Biolchini et al. en (Biolchini, Mian et al. 2007) son:

- o La *planificación* se centra principalmente en la definición de los objetivos de la investigación, la selección de las fuentes de información y la definición de los criterios de inclusión y exclusión de los estudios.
- o La fase de *ejecución* se enfoca en la selección y evaluación de los estudios encontrados, así como en la extracción de la información de los estudios seleccionados.
- o La última fase llamada de *análisis* se encarga de analizar y presentar los resultados, de acuerdo a diferentes criterios y perspectivas que faciliten su comprensión y uso posterior.

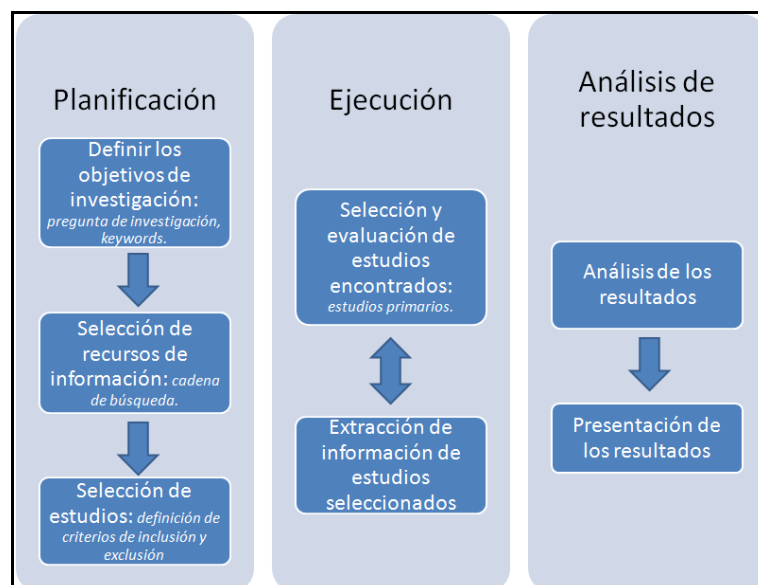


Figura 39. Fases principales del protocolo de RS propuesto por Biolchini.

6.1.1 Planificación de la Revisión Sistemática

6.1.1.1 Formalización de la Pregunta

En este apartado siguiendo el protocolo de la RS, se presentan los objetivos de la investigación, los cuales se enfocan tanto en los aspectos metodológicos como tecnológicos para la especificación y modelado de requisitos de DQ. El foco de la pregunta de investigación se centra en *“la identificación de trabajos que propongan mecanismos (tanto metodológicos como tecnológicos) para la especificación, representación e incorporación de requisitos de calidad de datos en el proceso de desarrollo de un producto software”*.

Para definir de una mejor manera la calidad y el alcance de la pregunta de investigación, primero se analizó el problema, el cual es posible resumirlo concretamente como, *“No existen en la bibliografía relacionada descripciones de mecanismos para plasmar específicamente los requisitos de calidad de datos”*.

Una vez definido el problema, es posible definir la pregunta de investigación como, *¿Qué metodologías y/o propuestas para la representación o modelado de requisitos de calidad de datos han sido descritas en la literatura y cuáles son sus principales características?*

Las palabras clave y algunos de los conceptos que se consideraron para llevar a cabo las consultas se muestran en la Tabla 67.

Palabras clave
Data Quality Dimension
Data Quality Requirement
Data Quality Metamodel
Data Quality Modeling
Data Quality Metadata
Data Quality Framework
Data Quality Methodology
Data Quality Dimensions Modeling

Tabla 67. Palabras clave consideradas en la RS.

El resultado esperado al término de la revisión sistemática es, la presentación del estado del arte sobre propuestas de investigación existentes para la especificación y modelado de requisitos de DQ, publicadas en los recursos disponibles y accesibles desde nuestra universidad (véase Tabla 68). Una vez

obtenidos los resultados, los principales beneficiarios de este trabajo serán aquellas personas relacionadas con el desarrollo de sistemas software como analistas, diseñadores, programadores y administradores de proyectos, además de académicos e investigadores relacionados con el área de calidad de datos, y otras áreas afines como calidad de Sistemas de Información e Ingeniería de Requisitos.

6.1.1.2 Selección de Recursos Bibliográficos

En este apartado, se enuncian los criterios de selección de los recursos, los métodos de búsqueda, así como la especificación de las cadenas de búsqueda utilizadas en el protocolo de la revisión sistemática.

En base a la recomendación de expertos en el área, las búsquedas fueron realizadas en bases de datos electrónicas de las principales editoriales, además se seleccionaron las actas de la conferencia internacional de calidad de datos e información más importante, la “*International Conference on Information Quality*” (ICIQ, <http://mitiq.mit.edu/>) por ser el evento especializado más importante del área, en el cual se presentan anualmente desde 1996 los principales trabajos e investigaciones en el área de calidad de información.

Asimismo, se amplió la búsqueda a las dos revistas más importantes dentro de esta área: *International Journal of Information Quality* “IJIQ” y *Journal of Data and Information Quality* “JDIQ” (cabe señalar que la búsqueda en ambas revistas se llevó a cabo de forma manual). Todos estos recursos presentan artículos con una gran importancia y la mayoría de ellos ofrecen motores de búsqueda. La lista completa de recursos se presenta en la Tabla 68.

Recursos bibliográficos disponibles
ACM Digital Library
IEEE Computer Society
Information Quality at MIT (ICIQ)
Science Direct
Wiley InterScience
IJIQ
JDIQ

Tabla 68. Lista de recursos.

Considerando la lista de palabras clave mencionadas anteriormente, y haciendo una combinación mediante los conectores lógicos “AND” y “OR” se obtuvo la siguiente cadena de búsqueda: (“Data Quality”) AND (“requirements” OR “dimensions” OR “Metamodel” OR “Modeling” OR “Model” OR “metadata” OR “framework” OR “Modelling” OR “MDA” OR “representation” OR “accuracy” OR “completeness” OR “consistency” OR “credibility” OR “currentness” OR “accessibility”

OR “compliance” OR “confidentiality” OR “efficiency” OR “precision” OR “traceability” OR “understandability” OR “availability” OR “portability” OR “recoverability”). Cabe mencionar que la sintaxis de la cadena varía en base a los distintos motores de búsqueda de los recursos disponibles.

6.1.1.3 Selección de los Estudios

Una vez elegidos los recursos para llevar a cabo las búsquedas, se define el procedimiento de selección de los estudios, en donde además se definen principalmente los criterios de inclusión y exclusión que se aplicarán durante este proceso de revisión.

El procedimiento utilizado para la selección de los estudios consistió básicamente en lo siguiente: inicialmente un investigador lee sólo el título y el resumen de los artículos encontrados para seleccionar los *estudios relevantes*, después se analiza el artículo completo, identificándose una lista de estudios *primarios potenciales*. Posteriormente otro investigador experto en el área llevó a cabo una verificación de esta lista para comprobar que los estudios cumplieran con los criterios acordados.

El procedimiento para la selección de los estudios primarios consiste en un proceso iterativo e incremental; *iterativo* debido a que algunas de las actividades principales como búsqueda, lectura, y extracción de la información se lleva a cabo en cada uno de los recursos seleccionados, además de que en cada iteración en base a los comentarios de los expertos, es posible en un momento dado ir refinando la cadena de búsqueda y con ello obtener resultados más precisos; e *incremental*, porque conforme se van realizando las búsquedas y extrayendo la información, se va conformando un subconjunto de estudios potenciales, el cual va creciendo hasta la terminación de la revisión sistemática obteniendo un documento completo.

Los criterios de inclusión y exclusión definidos para este trabajo se mencionan a continuación.

Criterios de Inclusión:

- Los artículos deberán describir propuestas o estrategias para la especificación y/o modelado de requisitos de calidad de datos.
- Los artículos deberán estar escritos en inglés.
- Se llevará a cabo un análisis sobre el título, palabras clave y resumen de cada uno de los estudios encontrados.
- No existe ningún tipo de restricción en cuanto a la fecha de publicación de los estudios.

De acuerdo a estos criterios se decidió cuáles estudios eran considerados

como estudios *relevantes* en la revisión sistemática.

Criterio de exclusión:

- Artículos que no proponen ningún tipo de metodología, estrategia o modelo (o metamodelo) para la especificación de requisitos de calidad de datos.

De acuerdo a la aplicación de este criterio, se decidió cuáles estudios de la revisión sistemática fueron considerados como *estudios primarios*.

6.1.2 Ejecución de la Selección

Después de aplicar el procedimiento de la revisión sistemática para la obtención de estudios primarios, un total de 820 estudios fueron encontrados, 42 de ellos se consideraron como relevantes y sólo 8 estudios primarios fueron obtenidos, el desglose de los estudios encontrados por cada recurso se muestra en la Tabla 69.

Recursos	Estudios			
	Fecha de búsqueda	Encontrados	Relevantes	Primarios
ACM Digital Library	13 – Septiembre - 09	164	6	4
IEEE Computer Society	24 – Septiembre - 09	169	9	1
ICIQ	31 – Agosto - 09	44	7	2
JDIQ	26 – Agosto - 09	12	0	0
IJIQ	24 – Agosto - 09	34	0	0
Science Direct	11 – Agosto - 09	100	16	1
Wiley InterScience	1 – Agosto - 09	297	4	0
	Totales	820	42	8

Tabla 69. Distribución de estudios por recurso.

A continuación se enumeran los estudios primarios obtenidos una vez realizada la ejecución de la revisión sistemática.

1. *Toward quality data: An attribute-based approach* (Wang, Reddy et al. 1995).
2. *Data Quality Requirements Analysis and Modeling* (Wang y Madnick 1993).
3. *A flexible and generic data quality metamodel* (Becker, McMullen et al. 2007).
4. *IP-UML: Towards a Methodology for Quality Improvement Based on the IP-MAP Framework* (Scannapieco, Pernici et al. 2002).
5. *A Product Perspective on Total Data Quality Management* (Wang 1998).
6. *DQRDFS: Towards a Semantic Web Enhanced with Data Quality* (Caballero, Verbo et al. 2008).
7. *Quality Views: Capturing and Exploiting the User Perspective on Data Quality* (Missier, Embury et al. 2006).
8. *A Data Quality Metamodel Extension to CWM* (Gomes, Farinha et al. 2007).

Como parte de la ejecución de la selección de los estudios y con el objetivo de hacer una revisión, un investigador experto verificó todos los estudios primarios seleccionados.

6.1.2.1 Extracción de la Información

Una vez identificados los estudios primarios, el siguiente paso fue extraer la información relevante para la revisión sistemática de cada uno de ellos, para lo cual se definió un formulario con el objetivo de mostrar la información más importante, además de una descripción de la propuesta de cada uno de los estudios referente a la especificación o modelado de requisitos de calidad de datos. Toda la información de los estudios se muestra en las tablas mostradas en la sección 6.1.5.

La evaluación de la ejecución de la revisión sistemática se llevó a cabo mediante reuniones con los expertos en el área, con el objetivo de validar que los estudios encontrados satisfacen la meta buscada.

6.1.3 Análisis de Resultados Obtenidos

Una vez extraída la información de todos los estudios primarios, el objetivo principal fue abordar la utilidad de los estudios primarios identificados de acuerdo a nuestro interés en descubrir propuestas que traten tanto los aspectos metodológicos como tecnológicos.

En este apartado se muestran los resultados de este análisis correspondiente. Cabe resaltar que el número de propuestas son significativamente pocas en relación al grado de interés que ha venido creciendo en el área de calidad de datos y de información en los últimos años. Nuestra preocupación por esto nos llevó a preguntar a varios investigadores y profesionales de DQ de diferentes países y organizaciones por qué no se encontró más trabajos. La mayoría de ellos coincidieron que la calidad de datos es tratada como un área específica y no una cuestión de organización, por lo que muchas organizaciones aún no son conscientes de los posibles beneficios de nuestro tema de investigación.

Por otra parte, la mayoría de ellos también están de acuerdo que el tema es sumamente relevante debido a que los resultados pudieran ayudar a las organizaciones a mejorar su desempeño a un costo relativamente bajo.

En la Tabla 70 se resume la información extraída de cada propuesta: la tecnología o modelo de datos utilizado, la existencia de una herramienta o prototipo que lo soporte, la inclusión de un ejemplo o caso de estudio, e informes respecto a si los resultados propuestos ya han sido probados en un entorno real.

Estudio	Modelo utilizado	Herramienta	Ejemplo	Probado
Data Quality Requirements Analysis and Modeling	Relacional	No	Si	No
Toward quality data: An attribute-based approach	Relacional	No	Si	No
A Product Perspective on Total Data Quality Management	Relacional	Si	Si	No
IP-UML: Towards a Methodology for Quality Improvement Based on the IP-MAP Framework	Orientado a objetos	Si	Si	No
Quality Views: Capturing and Exploiting the User Perspective on Data Quality	XML	No	Si	No
A flexible and generic data quality metamodel	Relacional	No	Si	No
A Data Quality Metamodel Extension to CWM	Orientado a objetos	No	No	Si
DQRDFS: Towards a Semantic Web Enhanced with Data Quality	XML	No	Si	No

Tabla 70. Tecnología o Modelo utilizado por los estudios seleccionados.

Al estudiar el análisis en mayor profundidad, hemos observado que ninguno de los trabajos existentes muestra una metodología para la obtención y gestión de requisitos de DQ. Teníamos la esperanza de encontrar una metodología que pudiera, en algún momento, guiar a los analistas y desarrolladores a implementar una correcta gestión de los requisitos de calidad de datos desde las primeras etapas, y en todo el proceso de desarrollo de un sistema de información. Esta falta motiva al reto de la meta de investigación de desarrollar una metodología para la gestión y combinación de requisitos software de calidad de datos, junto con los otros tipos de requisitos.

Por otra parte, y con respecto a la tecnología utilizada, se concluye que, dado que muchos tipos de aplicaciones pueden ser desarrolladas mediante el uso de diferentes tipos de tecnologías, una especie de generalización se debería utilizar, con el objetivo de hacer posibles diferentes tipos de desarrollo. Esta generalización puede ser lograda trabajando con modelos y metamodelos. Por lo que, la conclusión más importante con respecto a este tema, es que se debería trabajar sobre los cimientos de la Ingeniería Dirigida por Modelos (MDE) (Bézivin 2004), y el Desarrollo de Software Dirigido por Modelos (DSDM), a fin de hacer posible la implementación de diferentes tipos de desarrollos usando los mismos conceptos relacionados a requisitos de DQ.

6.1.4 Conclusiones de la Revisión Sistemática Realizada

Llevar a cabo una revisión sistemática es una labor considerablemente intensa, en comparación a realizar una búsqueda convencional de bibliografía. Sin embargo, el efectuar paso a paso el protocolo completo de la revisión sistemática nos permite validar los resultados obtenidos. Una vez realizado el análisis de extracción de la información de cada uno de los trabajos encontrados, enfocándose

principalmente en las ventajas y desventajas que cada uno de ellos provee, se pudo constatar la falta de nuevas metodologías, definición de procesos, herramientas y artefactos que son necesarios para los desarrolladores, con el objetivo de mejorar la implementación de sistemas de información capaces de gestionar de forma adecuada la calidad de los datos, mejorando con ello el desempeño de las tareas realizadas por los distintos usuarios.

En objetivo principal de esta RS, fue mostrar estudios con propuestas para la especificación y/o representación de requisitos de calidad de datos, tanto a nivel metodológico como tecnológico. Todo esto con la intención de situarse en un punto de partida en el estado del arte dentro de esta área de investigación, y de esta manera plantear propuestas encaminadas a solucionar las deficiencias en la especificación de requisitos de DQ. Tomando como referencia los resultados obtenidos, se hace evidente la necesidad de nuevas propuestas dentro del área de calidad de datos e ingeniería del software, debido a la escasez de iniciativas existentes enfocadas a esta área de la calidad en particular, a pesar de la gran relevancia e interés que ha venido surgiendo en la última década por una gran cantidad de organizaciones.

6.1.5 Extracción de información obtenida de los estudios primarios encontrados.

Descripción de los trabajos encontrados como parte de la RS realizada.

Extracción de Datos del estudio	
Nombre	Data Quality Requirements Analysis and Modeling
Publicación	Proceedings of the Ninth International Conference of Data Engineering
Fecha	Abril, 1993
Autores	Richard Wang, Henry Kon, and Stuart Madnick
Resultados objetivos del estudio	
Propuesta	El artículo está enfocado en: (1) establecer un conjunto de premisas, términos y definiciones para la gestión de la DQ, y (2) desarrollar una metodología paso a paso para la definición y documentación de los parámetros de DQ importantes para los usuarios. Dicha metodología de análisis de requisitos propuesta por los autores se basa en dos enfoques principales: - la especificación de <i>etiquetas</i> necesarias para los usuarios con el objetivo de estimar, determinar o mejorar la calidad de los datos. - la obtención desde el punto de vista del usuario de los aspectos generales de DQ no sensibles al etiquetado, por ejemplo, las características de completitud y tiempo de respuesta. Además, se propone una serie de vistas (vista de aplicación, vista de parámetros y vista de calidad), las cuales deberían ser incluidas como parte de la documentación de especificación de requisitos de calidad; conjuntamente mencionan una lista de candidatos posibles de atributos de calidad de datos.
Resultados	Metodología para la recolección y documentación de requisitos de calidad de datos.
Modelo	Utiliza un modelo de tipo "Relacional".
País	Austria
Dificultad mencionada por el autor	No existe aún una definición y estandarización de las dimensiones de calidad.

Tabla 71. Estudio primario Wang et al. (Wang y Madnick 1993).

Extracción de Datos del estudio	
Nombre	Toward quality data: An attribute-based approach
Publicación	Journal of Decision Support Systems
Fecha	Marzo, 1995
Autores	Wang, Richard Y., Reddy, M., Kon, H.
Resultados objetivos del estudio	
Propuesta	Los autores proponen una perspectiva de calidad mediante datos etiquetados a nivel de celda con indicadores de calidad, los cuales son características objetivas de los datos y su proceso de manufactura. Basados sobre esos indicadores, el usuario puede evaluar la calidad de los datos para una aplicación específica. Además los autores investigan cómo estos indicadores de calidad pueden ser especificados, almacenados, recuperados y procesados. Se propone un modelo de datos basado en atributos, algebra de consulta y reglas de integridad que facilitan el etiquetado a nivel de celda, así como también el procesamiento de los datos de la aplicación que está aumentada con los indicadores de calidad.
Resultados	Metodología de análisis de requisitos de calidad de datos basada en un modelo entidad-relación, para la especificación de los tipos de indicadores de calidad a ser modelados.
Modelo	Utiliza un modelo de tipo "Relacional".
País	U.S.A.
Dificultad mencionada por el autor	Estudian e investigan un enfoque orientado a objetos, debido a que el modelo relacional que representa el esquema de calidad, puede llegar a ser restrictivo. Un enfoque orientado a objetos parecería más sencillo para modelar los datos y sus indicadores de calidad, debido a que muchos de los mecanismos de control de calidad son orientados a procedimientos y éste enfoque los manejaría sin problema.

Tabla 72. Estudio primario Wang et al. (Wang, Reddy et al. 1995).

Extracción de Datos del estudio	
Nombre	A flexible and generic data quality metamodel
Publicación	Proceedings of the 12 th . International Conference on Information Quality, ICIQ 2007
Fecha	Noviembre, 2007
Autores	David Becker, William McMullen y Kevin Hetherington-Young.
Resultados objetivos del estudio	
Propuesta	Analizan y describen tres metamodelos genéricos mencionando algunas de sus capacidades más importantes: Common Warehouse Metamodel (CWM), Data Warehouse Quality (DWQ) y Universal Meta Data Model. Los autores proponen una <i>arquitectura</i> y un <i>metamodelo</i> básico para DQ, el cual cumple con los objetivos de proveer adecuadamente una flexibilidad, generalidad y facilidad de uso de los requisitos en situaciones de uso potenciales. Este metamodelo representa de forma adecuada los productos de información (IP), objetos de datos, así como las métricas, mediciones, requisitos, evaluaciones y acciones de DQ.
Resultados	Propuesta de una arquitectura y un metamodelo genérico para DQ.
Modelo	Utiliza un modelo de tipo "Relacional".
País	U.S.A.
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 73. Estudio primario Becker et al. (Becker, McMullen et al. 2007).

Extracción de Datos del estudio	
Nombre	IP-UML: Towards a Methodology for Quality Improvement Based on the IP-MAP Framework
Publicación	Proceedings of the Seventh International Conference on Information Quality, ICIQ02
Fecha	Noviembre, 2002
Autores	Monica Scannapieco, Barbara Pernici y Elizabeth Pierce
Resultados objetivos del estudio	
Propuesta	Se propone un perfil UML para calidad de datos con la finalidad de sustentar la mejora de la calidad dentro de una organización. Este perfil esta basado en el Framework IP-MAP (Shankaranarayan, Wang et al. 2000), aunque difiere de éste debido principalmente a: (1) especifica los artefactos para producción durante el proceso de mejora en términos de diagramas dibujados usando elementos de UML definidos en el perfil de calidad de datos; (2) éste usa el IP-MAP no sólo con el objetivo de evaluar la calidad y pensar en acciones de mejora, sino también como un camino esquemático para diseñar y realizar acciones de mejora. <i>El IP-MAP es una extensión de un Sistema de Manufactura de Información (IMS) propuesto por (Ballou, Wang et al. 1998), este Framework tiene la ventaja de combinar tanto el análisis de datos como el análisis de procesos, con el objetivo de evaluar la calidad de los datos.</i> El perfil de calidad de datos consiste de tres diferentes modelos: Modelo de Análisis de Datos, Modelo de Análisis de Calidad y Modelo de Diseño de Calidad. El modelo de análisis de datos, especifica cuales datos son importantes para los consumidores, ya que su calidad es crítica para el éxito de las organizaciones. El modelo de análisis de calidad consiste en modelar los elementos que permiten la representación de los <i>requisitos de calidad de los datos</i> , un requisito de calidad puede ser relacionado a una dimensión de calidad o características que son típicamente definidas para la calidad de los datos. El modelo de diseño de calidad incorpora la perspectiva de IP-MAP, el cual ayuda a entender los detalles asociados con el proceso de manufactura de los productos de información.
Resultados	Muestra un perfil y una metodología para la producción de artefactos UML diseñados por el perfil de calidad de datos.
Modelo	Utiliza un modelo "Orientado a objetos".
País	U.S.A.
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 74. Estudio primario Scannapieco et al. (Scannapieco, Pernici et al. 2002).

Extracción de Datos del estudio	
Nombre	A Product Perspective on Total Data Quality Management
Publicación	Communications of the ACM
Fecha	Febrero, 1998
Autores	Richard Wang
Resultados objetivos del estudio	
Propuesta	Este artículo presenta la metodología <i>Total Data Quality Management</i> (TDQM), cuyo propósito principal es entregar productos de información (IP) de alta calidad a los consumidores de información, además de introducir los conceptos del ciclo de TDQM y de productos de información. Explica las etapas del ciclo de TDQM relacionadas a los productos de información: <i>Definición, Medición, Análisis y Mejora</i> , haciendo un especial énfasis en la definición de las características de los productos de información y los requisitos de calidad de la información. Además el autor también muestra un instrumento de software para llevar a cabo encuestas para la evaluación de la calidad de información, en donde se evalúa una lista de dimensiones de calidad definidas por el autor.
Resultados	Muestra la metodología TDQM e ilustra como ésta puede ser llevada a la práctica en un amplio rango de organizaciones.
	Utiliza un modelo de tipo "Relacional".
País	U.S.A.
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 75. Estudio primario Wang (Wang 1998).

Extracción de Datos del estudio	
Nombre	DQRDFS: Towards a Semantic Web Enhanced with Data Quality
Publicación	4th. International Conference on Web Information Systems and Technologies, WEBIST 08.
Fecha	Mayo, 2008
Autores	Ismael Caballero, Eugenio Verbo, Coral Calero y Mario Piattini
Resultados objetivos del estudio	
Propuesta	Este artículo introduce una nueva vista acerca de la Web semántica, basada en el concepto de cantidad de calidad de datos (QDQ), en los cuales los aspectos de DQ son usados como base para habilitar a las máquinas el procesar los documentos de la Web semántica para diferentes actividades como recuperación de información o filtrado de documentos. <i>La Web semántica es una extensión de la Web actual en la cual la información es dada con un sentido bien definido, habilitando a las personas y computadoras a trabajar en cooperación</i> (Bernes-Lee, Hendler et al. 2001). Este artículo tiene una doble meta: (1) mostrar a los lectores una breve introducción de DQ, y (2) mostrar cómo se han aplicado los fundamentos de la DQ con el objetivo de resaltar la calidad de documentos Web para la Web semántica. El primer paso con el objetivo de habilitar la DQ en la Web semántica, consiste en identificar desde la Especificación de Requisitos de Usuario de la DQ (DQ-URS) el conjunto de elementos necesarios que necesitan ser estudiados. El segundo paso consiste en identificar las dimensiones de DQ y los metadatos relacionados. El tercer paso es la obtención y anotación de los valores para los metadatos. La representación de ésta información se hace mediante documentos de tipo XML.
Resultados	Muestra una propuesta del concepto de QDQ orientado a la Web semántica.
	Utiliza el lenguaje "XML" (Extensible Markup Language) para su representación.
País	Portugal
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 76. Estudio primario Caballero et al. (Caballero, Verbo et al. 2008).

Extracción de Datos del estudio	
Nombre	Quality Views: Capturing and Exploiting the User Perspective on Data Quality
Publicación	International Conference on Very large databases, VLDB '06
Fecha	Septiembre, 2006
Autores	Paolo Missier, Suzanne Embury, Mark Greenwood, Alun Preece y Binling Jin
Resultados objetivos del estudio	
Propuesta	Este artículo presenta un modelo de calidad centrado en el usuario además de un ambiente de software, en el cual expertos en el dominio pueden de una manera fácil y rápida codificar y probar sus propios criterios de calidad heurísticos. Como parte medular del modelo, proponen el concepto de "vista de calidad" ("quality view"), similar a " <i>lentes</i> " personalizados, a través de los cuales los datos pueden ser observados. Las principales contribuciones de este trabajo son: • Un modelo semántico extensible al usuario para conceptos de calidad de información en e-science. • Un modelo de procesos y un lenguaje declarativo para la especificación de vistas de calidad abstractas en términos de unos pocos operadores lógicos. • Una arquitectura para la implementación de vistas de calidad, dentro de varios ambientes de procesamiento de datos.
Resultados	Propone un marco de trabajo para la especificación de requisitos de procesamiento de calidad por parte del usuario, llamadas " <i>vistas de calidad</i> ".
Modelo	Utiliza el lenguaje "XML" (Extensible Markup Language) para su representación.
País	Corea
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 77. Estudio primario Missier et al. (Missier, Embury et al. 2006).

Extracción de Datos del estudio	
Nombre	A Data Quality Metamodel Extension to CWM
Publicación	4 th . Asia-Pacific Conference on Conceptual Modelling, APCCM 2007
Fecha	Febrero, 2007
Autores	Pedro Gomes, José Farinha, Maria José Trigueiros
Resultados objetivos del estudio	
Propuesta	Este artículo propone un metamodelo para calidad de datos y limpieza de datos, ambos conceptos aplicables al contexto de almacenes de datos. El metamodelo presentado esta integrado con el “Metamodelo de Almacenes de Datos Común” (CWM (OMG 2003)), ofreciendo una extensión de este estándar hacia la calidad de datos. Además provee una serie de directrices de modelado para el almacenado de especificaciones formales de reglas de DQ. El propósito principal del metamodelo es proveer soporte a las actividades de perfilado y limpieza de datos, con reglas que pueden ser establecidas con el objetivo de detectar problemas de calidad de datos, así como establecer soluciones de limpieza de datos. En relación a la limpieza de los datos, un soporte de “ <i>metadato</i> ” es dado, con el objetivo de habilitar la meta final de lograr el nivel más alto de automatización posible. No obstante, un soporte de <i>metadato</i> también es provisto cuando la participación del usuario es requerida dentro del proceso de limpieza.
Resultados	Muestra un metamodelo para calidad y limpieza de datos, ambos conceptos aplicables en el contexto de almacenes de datos.
Modelo	Utiliza un modelo “Orientado a objetos”.
País	Australia
Dificultad mencionada por el autor	No menciona ninguna.

Tabla 78. Estudio primario Gomes et al. (Gomes, Farinha et al. 2007).

6.2 Anexo B. Plugin implementado en la plataforma Eclipse.

El metamodelo *DQ_WebRE* se implementó en la plataforma Eclipse (ver Figura 40), haciendo uso del marco de trabajo *EMF* (*Eclipse Modeling Framework*) (Eclipse). Una de las razones principales por la que se decidió utilizar *EMF* para la implementación del metamodelo *DQ_WebRE*, fue debido a las características y ventajas que provee, ya que los modelos generados a partir de este metamodelo pueden ser almacenados en formato *XMI* (*XML Metadata Interchange*), el cual puede considerarse como un modelo de entrada para posteriormente implementar las transformaciones necesarias y convertir los modelos definidos (diagramas de casos de uso y actividades) a modelos de diseño (por ejemplo, diagramas de clases) y posteriormente código.

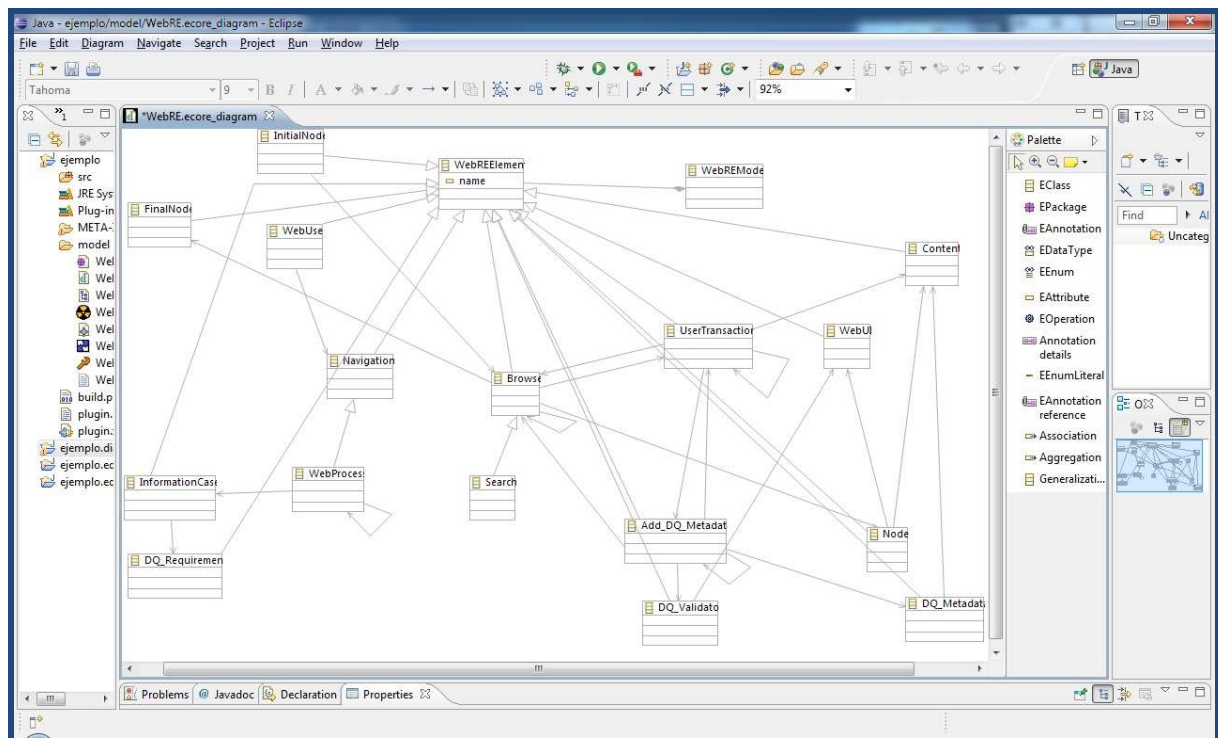


Figura 40. Metamodelo *DQ_WebRE* implementado en EMF de Eclipse.

Con el objetivo de soportar nuestro enfoque, se desarrolló un *plugin* usando *GMF* (*Graphical Modeling Framework*) (Eclipse) proporcionado por la plataforma *Eclipse*. Dicho *plugin* permite modelar los distintos elementos para la gestión de requisitos de calidad de datos mediante diagramas de casos de uso y diagramas de actividades.

El *plugin* se desarrolló para proporcionar un ambiente de trabajo adecuado que entre otras cosas, permitiera realizar diagramas de modelado con los elementos definidos.

Tanto en la Figura 41 como en la Figura 42 se puede observar el uso del plugin desarrollado, en el cual se muestra un diagrama de casos de uso y otro de actividades, como resultado del ejemplo de aplicación descrito en la sección 3.2.6.4.

En la parte derecha de la herramienta (ver Figura 41), se puede observar una “toolbox” especial con los elementos definidos en el metamodelo “DQ_WebRE”.

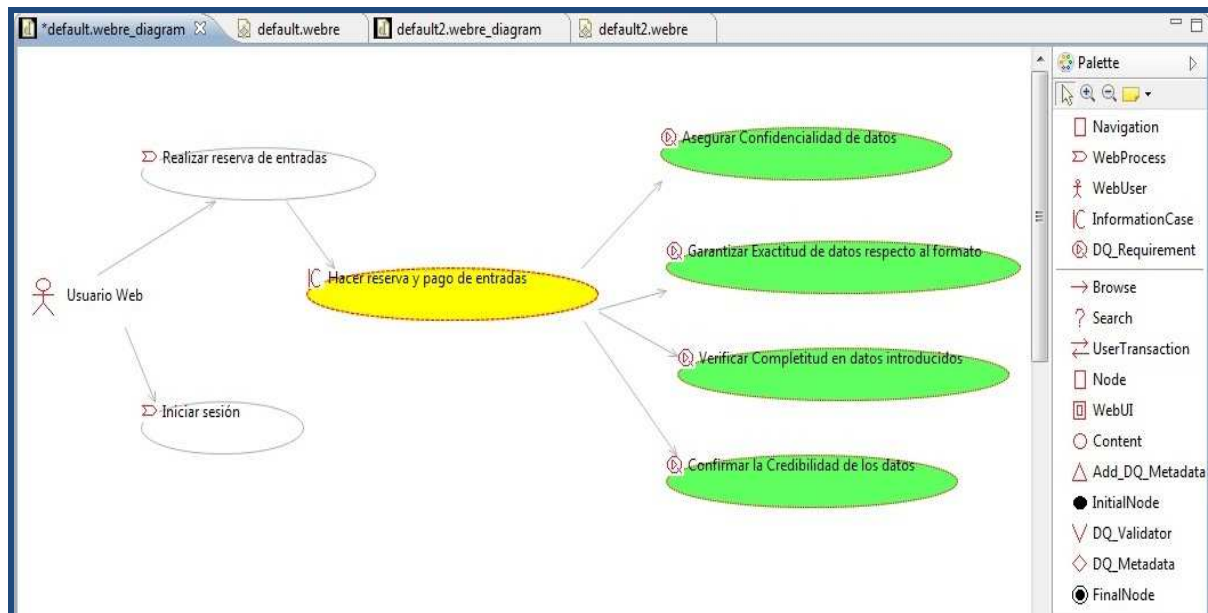


Figura 41. Diagrama de casos de uso utilizando plugin desarrollado en Eclipse.

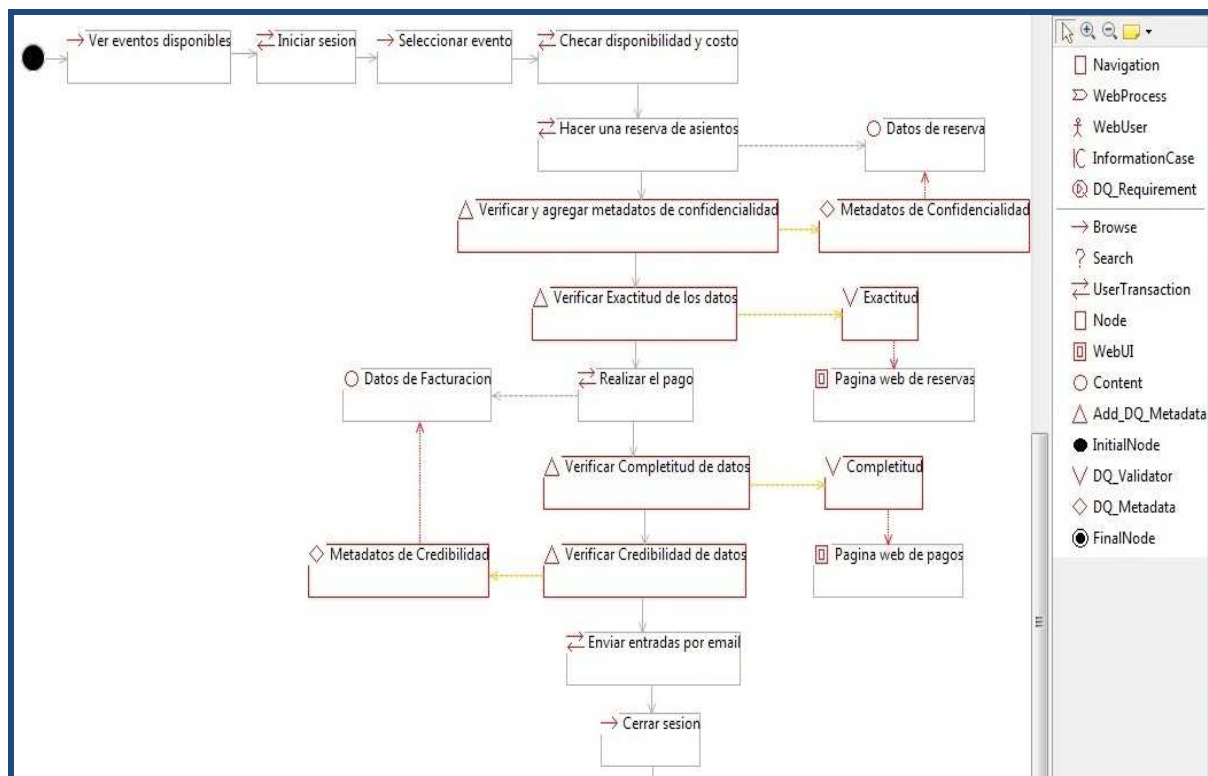


Figura 42. Diagrama de actividades utilizando plugin desarrollado en Eclipse.

6.3 Anexo C. Descripción de relaciones entre problemas en un SI y funcionalidades Web.

Enseguida se describe a profundidad cada una de las relaciones obtenidas en la Tabla 13, entre los obstáculos o problemas en un SI identificados por Strong en (Strong, Lee et al. 1997) y las distintas funcionalidades propuestas por Collins (Collins 2001).

Gestión de contenido

Múltiples fuentes

Para la gestión de contenidos, la existencia de múltiples procesos o recursos diferentes que generan valores de los datos puede provocar el problema de no saber cuál es la que realmente tiene el mayor grado de calidad. Por ejemplo, al generarse distintos valores para los “mismos” datos, la elección de la fuente de información debe realizarse de forma minuciosa, verificando que los datos obtenidos son los mismos, o que su combinación puede dar lugar a unos valores de mayor calidad.

Producción subjetiva

La creación de valores para los datos de forma subjetiva o con connotaciones, puede ocasionar problemas al momento de autorizar su inclusión o exclusión en el portal. La gestión de todo el contenido de la información debería de realizarse a través de la generación de “reglas” para reducir la varianza en los datos producida por esa subjetividad.

Errores sistemáticos en la producción

Al momento de llevar a cabo la creación de colecciones de valores pueden generarse errores en la producción de los valores para los datos. Una correcta gestión de todo el contenido debería abarcar el proceso desde la generación de los datos (fuente de información) hasta su correcto almacenado en la aplicación.

Demasiada cantidad de información

Grandes cantidades de datos almacenados puede provocar la inclusión o exclusión de cierta información reservada. Además, el tener demasiados datos almacenados puede provocar un tiempo de acceso a ellos exagerado, provocando una mala gestión al momento de extraer o resumir cierta información.

Información no numérica que no puede ser indexada

La gestión y recuperación de la información en formato de imágenes hace más difícil su manejo e indexación. De ser posible la gestión de los contenidos, la información manejada debería estar en un formato establecido y entendible para todos los procesos de la organización (preferentemente “textual”), de tal forma que su acceso, gestión y almacenado se facilite dentro de la aplicación.

Análisis automatizado

Se requiere un análisis del contenido de forma automatizada para proveer al usuario un valor agregado en los datos. De ser posible la gestión de la información debería llevarse a cabo en un formato adecuado (“textual”) con el propósito de realizar diferentes tipos de análisis de los datos de forma automatizada y en tiempos razonablemente cortos. Realizar una gestión de información en formato no textual, por ejemplo en imágenes, resulta sumamente complicado y de difícil análisis e indexación.

Necesidad de cambio en datos

Es necesario cumplir con las necesidades de información adecuada para los distintos consumidores/usuarios/clientes, aún cuando éstas cambien en el transcurso del tiempo. Una correcta gestión del contenido mostrado en la aplicación deberá proveer datos actualizados en el momento preciso de realizar alguna tarea por parte de los usuarios, además de revisar de forma periódica los distintos procesos de información.

Requisitos de seguridad

Los datos deben estar accesibles para los usuarios, sin embargo, habrá que asegurar la privacidad y seguridad de la información que se provee mediante una correcta gestión, posiblemente definiendo niveles de acceso entre usuarios, y/o políticas y procedimientos consistentes de seguridad.

Falta de recursos de cómputo

La falta de recursos de cómputo suficientes para gestionar contenidos puede llegar a provocar una falta de calidad en los datos gestionados y almacenados en la aplicación. Algunos de los problemas más comunes se manifiestan al no disponer de esos recursos, como tiempos de espera demasiado grandes o resultados de análisis de datos incompletos.

Procesos y acciones

Demasiada información

Grandes cantidades de datos pueden provocar conflictos con los usuarios, generando una mala comunicación al momento de ejecutar algún proceso de

negocio con la organización. Proporcionar la cantidad idónea de información mejora la toma de decisiones al iniciar o participar en un proceso de negocio.

Necesidad de cambio en datos

Resulta imprescindible proveer datos adecuados y actuales a los usuarios, cumpliendo sus necesidades de información en el momento justo que lo requieran para participar en un proceso de negocio. Proporcionar datos no actualizados puede generar un impacto negativo para la organización, generando a su vez pérdida de clientes con los cuales iniciar o continuar un proceso de negocio.

Capacidades de búsqueda

Demasiada información

Grandes cantidades de datos puede dificultar la realización de búsquedas dentro y fuera de la aplicación, impidiendo el acceso a la información en un tiempo moderado, y mostrándola a los usuarios de la manera adecuada. Mostrar demasiada información como resultado de una búsqueda, puede llegar a generar insatisfacción en los usuarios de la aplicación, ya que lejos de ayudarlo le obligaría a realizar más esfuerzos para entender y discriminar aquella que no le interese.

Sistemas distribuidos

Los datos almacenados en varios sistemas distribuidos pueden provocar problemas para su acceso, a parte de un tiempo excesivo para su localización. Además de que cierta información puede estar almacenada en otros sistemas en distinto tipo de formato o código de representación. Todo esto resulta en una búsqueda de información con una baja calidad de los datos encontrados.

Información no numérica

La recuperación de datos en formatos no numéricos es muy difícil de indexar, y como consecuencia más difícil de acceder, provocando la recuperación de información no adecuada y/o errónea para su uso por parte de los usuarios. Todo ello puede generar insatisfacción en el usuario y una falta de confiabilidad en los datos.

Requisitos de seguridad

Los usuarios deberían acceder solo a la información permitida, asegurándose la privacidad y seguridad de cierta información reservada. Llevar a cabo una búsqueda de información debería estar limitado sólo a los datos “públicos” o “permitidos” para los distintos tipos de usuarios de la aplicación.

Falta de recursos

La falta de suficientes recursos de cómputo puede ocasionar un acceso a los datos limitado. Siendo los principales problemas a enfrentarse: la cantidad de información en donde se realizará la búsqueda y el formato en el que se encuentra almacenada dicha información. Todo esto conlleva a obtener información errónea e incompleta al realizar una búsqueda por parte del usuario.

Administración

Necesidad de cambio en datos

El administrador del portal deberá proveer la información necesaria al usuario dando mantenimiento al portal mediante sus tareas asociadas, con el objetivo de cumplir con las necesidades de información solicitadas por cada usuario en el momento oportuno. De la misma manera, los procesos de negocio deberían ser actualizados dentro de la aplicación, permitiendo a los usuarios realizar sus tareas con la aplicación, utilizando datos confiables.

Seguridad

Requisitos de seguridad

Asegurar la privacidad y confidencialidad de los datos, mediante niveles de acceso a cada tipo de usuario. De ser posible también, desarrollar políticas y procedimientos de seguridad para cada tarea en la aplicación. El objetivo final es mantener los datos accesibles sólo para los usuarios autorizados.

Integración y puntos de datos

Demasiada información

Una gran cantidad de volúmenes de datos pueden dificultar su acceso, sobre todo si éstos se encuentran localizados en varios sistemas de información (por ejemplo, bases de datos distintas, archivos con distinto formato de almacenado, diversos almacenes de datos como Data Warehouse, etc.), generando en consecuencia problemas al mostrarlos al usuario mediante un solo punto de acceso.

Sistemas distribuidos

Pueden surgir problemas al acceder a los datos almacenados en varios sistemas o servidores de forma distribuida, además de que los mismos datos pueden llegar a estar almacenados en diferentes formatos o bajo una codificación distinta, ocasionando problemas para su indexación al ser mostrados al usuario a través de un solo punto de acceso.

Requisitos de seguridad

El acceso fácil a los datos desde varias fuentes, puede ocasionar conflicto con requisitos de seguridad y confidencialidad de la información. Para eliminar o al menos minimizar este tipo de problemas, se deberá generar requisitos de verificación y permisos de acceso por cada tipo de usuario en la aplicación, protegiendo con ello la seguridad y confidencialidad de los datos a ser mostrados mediante un solo punto de acceso.

Comunicación y colaboración

Demasiada información

Grandes cantidades de datos pueden hacer más difícil la transmisión eficiente de información, provocando una mala comunicación y colaboración entre los empleados y usuarios de la aplicación, entorpeciendo así su colaboración al efectuar o verificar un proceso de negocio.

Sistemas distribuidos

El tener los datos distribuidos en varios sistemas y en formatos distintos, puede interferir en la comunicación y coordinación de empleados y usuarios. Al igual que en punto anterior, una mala gestión de la información puede llegar a generar conflictos entre los usuarios para poder acceder a ella de forma oportuna, perjudicando con esto la colaboración entre ellos. Un ejemplo típico, es la caída de conexión cuando se intentan recuperar datos desde sistemas remotos, lo que ocasiona retrasos en el acceso de información.

Análisis automatizado

A falta de un análisis de contenido automatizado, los datos serán almacenados en formatos, nombres y definiciones inconsistentes, interfiriendo en la comunicación y colaboración de empleados y usuarios. Uno de los problemas más comunes en este sentido, se refiere al formato en que se encuentran almacenados determinados datos (tipo: int, char, String), el almacenar datos en distinto formato ocasiona que no se puedan realizar análisis de ellos de forma automatizada, lo que perjudicará la comunicación entre los usuarios al momento de tomar decisiones de negocio.

Falta de recursos

La carencia del suficiente equipo de cómputo y la falta de acceso a la información por parte de los empleados, impedirá una buena comunicación y colaboración disminuyendo su productividad. Un ejemplo característico resulta al tener un equipo de cómputo obsoleto, insuficiente para realizar procesos específicos de la aplicación.

Presentación

Necesidad de cambio en datos

Es necesario proveer y presentar los datos actualizados y de forma adecuada al usuario, de tal forma que cumpla con sus necesidades específicas de información en todas las funcionalidades disponibles en la aplicación.

Taxonomía

Sistemas distribuidos

Los datos son almacenados en sistemas heterogéneos con diferentes valores o formatos, lo que no permite una terminología de uso común en las diferentes categorías de la organización. Este problema puede generar un desconcierto en la semántica de los conceptos usados por los usuarios, al ejecutar diversos procesos de negocio haciendo uso de la aplicación.

Personalización

Necesidad de cambio en datos

El portal debe proveer un ambiente de trabajo organizado y configurado para cada usuario, de acuerdo a sus necesidades específicas de información en el momento oportuno. La personalización es un aspecto muy importante, ya que ayuda a mantener los datos consistentes para cada uno de los usuarios.

Requisitos de seguridad

Se deberá proveer al usuario la opción de personalizar su ambiente de trabajo en la aplicación, respetando ciertos requisitos de seguridad y privacidad de la información. El usuario podrá configurar su ambiente de trabajo, mostrando sólo los datos a los cuales tenga acceso de acuerdo a su rol dentro de la aplicación. Además, esto permitirá mantener los datos de forma consistente y constante mediante un ambiente de trabajo organizado.

Características de Ayuda

Demasiada información

Grandes cantidades de datos almacenados requerirán de opciones de ayuda para el usuario, en caso de llevar a cabo algún tipo de funcionalidad específica con un conjunto de datos. Las características de ayuda también se reflejarán al momento de mostrar información detallada relacionada con algún proceso específico de la organización, explicando la forma de utilizar y gestionar ciertos datos del negocio.

6.4 Anexo D. Lista de Acrónimos

CWM	Common Warehouse Metamodel
DAQ_UWE	Metamodelo para el diseño de aplicaciones Web conscientes de calidad de datos
DCI	Diagrama de Casos de Información
DQ	Data Quality
DQ-VORD	Metodología para la Gestión de Proyectos de Especificación de Requisitos de calidad de datos
DQ_WebRE	Metamodelo ampliado para la especificación de requisitos de DQ en aplicaciones Web
DQA	Data Quality Assurance
DQR	Data Quality Requirement
DQSR	Data Quality Software Requirement
DSDM	Desarrollo de Software Dirigido por Modelos
DWQ	Data Warehouse Quality
EMF	Eclipse Modeling Framework
ERP	Planificación de Recursos Empresariales
ERS	Especificación de Requisitos Software
GMF	Graphical Modeling Framework
HDM	Hypertext Design Model
ICIQ	International Conference on Information Quality
IEEE	Institute of Electrical and Electronics Engineers
IJIQ	International Journal of Information Quality
IMS	Sistema de Manufactura de Información
IP	Information Product
ISO	International Organization for Standardization
IW	Ingeniería Web
JDIQ	Journal of Data and Information Quality
LOPD	Ley Orgánica de Protección de Datos
MDA	Model Driven Architecture
MDE	Ingeniería Dirigida por Modelos
MDWE	Ingeniería Web Dirigida por Modelos
MOSCAF	Modelo para la selección de requisitos de DQ según las funcionalidades a implementar en una aplicación Web
NDT	Navigational Development Techniques
OOWS	Object Oriented Web Solutions
PUD	Proceso Unificado de Desarrollo
QDQ	Quantity of Data Quality
QVT	Query View Transformation
RS	Revisión Sistemática
SI	Sistema de Información

TDQM	Total Data Quality Management
UI	User Interface
UML	Unified Modeling Language
UWE	UML-based Web Engineering
WebRE	Ingeniería de Requisitos Web
WSDM	Web Site Design Method
XMI	XML Metadata Interchange
XML	Extensible Markup Language

-
- Akoka, J., L. Berti-Equille, et al. (2007). A Framework for Quality Evaluation in Data Integration Systems. International Conference on Enterprise Information Systems, ICEIS.
- Alkharboush, N. and Y. Li (2010). A DECISION RULE METHOD FOR DATA QUALITY ASSESSMENT. 15th. International Conference on Information Quality, Little Rock, Arkansas.
- Avison, D., F. Lan, et al. (1999). "Action Research." Communications of the ACM. **42**(1): 94-97.
- Ballou, D. P., R. Y. Wang, et al. (1998). "Modelling Information Manufacturing Systems to Determine Information Product Quality." Management Science **44**(4): 462-484.
- Baresi, L., F. Garzotto, et al. (2002). Meta-modeling Techniques Meet Web Application Design Tools. Fundamental Approaches to Software Engineering, Springer Berlin / Heidelberg. **Volume 2306/2002**: 182-206.
- Baskarada, S., J. Gao, et al. (2006). Process Areas Impacting on the Quality of Asset Management Information in Engineering Organisations. 1st Australasian Workshop on Information Quality (AusIQ'2006), Adelaide, Australia.
- Batini, C., C. Cappiello, et al. (2009). "Methodologies for data quality assessment and improvement." ACM Computing Surveys **Vol. 41, No. 3**.
- Batini, C. and M. Scannapieco (2006). Data Quality: Concepts, Methodologies and Techniques. Berlin, Springer-Verlag Berlin Heidelberg.
- Becker, D., J. Jaster, et al. (2009). Flexible and Generic Data Quality Metadata Exchange. International Conference on Information Quality, ICIQ '09.
- Becker, D., W. McMullen, et al. (2007). A Flexible and Generic Data Quality Metamodel. International Conference on Information Quality.
- Bernes-Lee, T., J. Hendler, et al. (2001). "The Semantic Web." Scientific American.
- Bertino, E., A. Maurino, et al. (2010). Guest Editors' Introduction: Data Quality in the Internet Era. **14**: 11-13.
- Bézivin, J. (2004). "In Search of a Basic Principle for Model Driven Engineering." UPGRADE, Novática. **Vol. 2**(No.2): 21-24.

- Biolchini, J. C. d. A., P. G. Mian, et al. (2007). "Scientific research ontology to support systematic review in software engineering." Advanced Engineering Informatics **21**(2): 133-151.
- Bizer, C. and R. Cyganiak (2009). "Quality-driven information filtering using the WIQA policy framework." Web Semant. **7**(1): 1-10.
- Brien, D. O. (2011). "A Website Dedicated to Information/Data Quality Disasters from Around the World." from <http://www.iqtrainwrecks.com/>.
- Caballero, I., C. Calero, et al. (2008). MMPRO: A Methodology Based on ISO/IEC 15939 to Draw Up Data Quality Measurement Processes. ICIQ.
- Caballero, I., A. Caro, et al. (2008). "IQM3: Information Quality Maturity Model." Journal of Universal Computer Science **14**: 1-29.
- Caballero, I., E. M. Verbo, et al. (2007). A Data Quality Measurement Information Model based on ISO/IEC 15939. 12th International Conference on Information Quality, MIT, Cambridge, MA.
- Caballero, I., E. M. Verbo, et al. (2008). DORDFS: Towards a Semantic Web Enhanced with Data Quality. Web Information Systems and Technologies, Funchal, Madeira, Portugal.
- Cai, Y. and G. Shankaranarayanan (2007). "Managing data quality in inter-organisational data networks." International Journal of Information Quality **1**(3): 254 - 271.
- Cappiello, C. and M. Comuzzi (2007). Efficient Allocation of Quality Improvement Efforts to Support the Definition of Data Service Offerings. 12th International Conference on Information Quality, Cambridge, MA.
- Cappiello, C., F. Daniel, et al. (2010). "Information Quality in Mashups." Internet Computing, IEEE **14**: 14-22.
- Caro, A., C. Calero, et al. (2008). "A proposal for a set of attributes relevant for Web Portal Data Quality." Software Quality Journal.
- Ceri, S., P. Fraternali, et al. (2000). "Web Modeling Language (WebML): a modeling language for designing Web sites." Computer Networks **33**(1-6): 137-157.
- Clement, D., S. Ben Hassine Guetari, et al. (2010). Data Quality as a key success factor for migration projects. The 15th. International Conference on Information Quality, ICIQ 2010, Little Rock, Arkansas.
- Collins, H. (2001). Corporate Portal Definitions and Features. New York, NY, USA, Amacom Books.
- De Castro, V. and E. Marcos (2009). "Towards a Service-Oriented MDA-Based Approach to the Alignment of Business Process with IT Systems: from the Business Model to a Web Service Composition Model." International Journal of Cooperative Information Systems **18**, No. 2: 225-

- De Troyer, O. M. F. and C. J. Leune (1998). "WSDM: a user centered design method for Web sites." Computer Networks and ISDN Systems **30**(1-7): 85-94.
- Deshpande, Y., S. Murugesan, et al. (2002). "Web Engineering." Journal of Web Engineering, Rinton Press, US **1**(1): 61-73.
- Eclipse. "Eclipse Modeling Framework Project (EMF)." from <http://www.eclipse.org/modeling/emf/>.
- Eclipse. "Graphical Modeling Framework (GMF)." from <http://www.eclipse.org/modeling/gmf/>.
- English, L. (1999). Improving Data Warehouse and Business Information Quality: Methods for reducing costs and increasing Profits. New York, NY, USA, Willey & Sons.
- Eppler, M. (2003). Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes, Springer.
- Eppler, M. and M. Helfert (2004). A Classification and Analysis of Data Quality Costs. International Conference on Information Quality, MIT, Cambridge, MA, USA.
- Eppler, M. and P. Muenzenmayer (2002). Measuring Information Quality in the Web Context: A Survey of State-of-the-Art Instruments and an Application Methodology. Proceeding of the Seventh International Conference on Information Quality.
- Escalona, M. J. and G. Aragón (2008). "NDT. A Model-Driven Approach for Web Requirements." IEEE Trans. Softw. Eng. **34**(3): 377-390.
- Escalona, M. J. and N. Koch (2004). "Requirements Engineering for Web Applications: A Comparative Study." Journal on Web Engineering **2**: 193-212.
- Escalona, M. J. and N. Koch (2006). Metamodeling the Requirements of Web Systems. Web Information Systems and Technologies. S. B. Heidelberg. **Volume 1**: 267-280.
- Fons, J., V. Pelechano, et al. (2003). Development of Web Applications from Web Enhanced Conceptual Schemas. Conceptual Modeling - ER 2003. I.-Y. Song, S. Liddle, T.-W. Ling and P. Scheuermann, Springer Berlin / Heidelberg. **2813**: 232-245.
- Franch, X. and J. P. Carvallo (2003). "Using Quality Models in Software Package Selection." IEEE Software. **20**(1): 34-41.
- French, W. L. and C. H. Bell (1996). Organizational Development: Behavioral Science Interventions for Organization Improvement. London, Prentice Hall.
- Friedman, T. and A. Bitterer. (2011). "Magic Quadrant for Data Quality Tools."

- Gartner. (2009). "Survey: Poor data quality most common business intelligence problem." from <http://searchdatamanagement.techtarget.com/>.
- Ge, M. and M. Helfert (2007). A Review of Information Quality Research. International Conference on Information Quality, MIT, Cambridge, MA, USA.
- Ginige, A. and S. Murugesan (2001). "The essence of web engineering - managing the diversity and complexity of web application development." Multimedia, IEEE **8**(2): 22-25.
- Gomes, P., J. Farinha, et al. (2007). A data quality metamodel extension to CWM Proceedings of the fourth Asia-Pacific conference on Conceptual modelling - Volume 67 Ballarat, Australia Australian Computer Society, Inc.: 17-26
- Guerra-García, C., I. Caballero, et al. (2009). Modelado de Requisitos de Calidad de Datos en el Proceso de Desarrollo de Portales Web. VI Taller sobre Desarrollo de Software Dirigido por Modelos (DSDM'09) San Sebastián, España.
- Guerra-García, C., I. Caballero, et al. (2009). DQ-VORD: A Methodology for Managing and Integrating Data Quality Requirements into Software Requirement Specification. IADIS International Conference on WWW/INTERNET 2009, Rome, Italy.
- Guerra-García, C., I. Caballero, et al. (2010). Modelado de Requisitos de Calidad de Datos en Ingeniería Web. XV Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2010, Valencia, España.
- Guerra-García, C., I. Caballero, et al. (2010). A Systematic Literature Review of How to Introduce Data Quality Requirements into a Software Product Development. 5th. International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE, Athens, Greece.
- Guerra-García, C., I. Caballero, et al. (2011). Capturing Data Quality Requirements for Web Applications by means of DQ WebRE. 2nd International Workshop on Business intelligence and the WEB, BEWEB 2011, Uppsala, Sweden, ACM 978-1-4503-0610-2/11/03.
- Guerra-García, C., I. Caballero, et al. (2011). "A Survey on How to Manage Specific Data Quality Requirements during Information System Development." Lecture Notes in Computer Science(Evaluation of Novel Approaches to Software Engineering): To be published.
- Hult, M. and S. Å. Lennung (1980). "TOWARDS A DEFINITION OF ACTION RESEARCH: A NOTE AND BIBLIOGRAPHY." Journal of Management Studies **17**(2): 241-250.
- IDC (2009). Estudio sobre la Calidad de Datos en España, IDC, España.
- IEEE (1990). IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology - Description.
- ISO-25012 (2008). "ISO/IEC 25012: Software Engineering-Software product Quality Requirements and

Evaluation (SQuaRE)-Data Quality Model."

ISO (2005). ISO/IEC 25000 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) –Guide to SQuaRE. International Organization for Standardization. Ginebra, Suiza.

ISO/TS--8000-100 (2011). "ISO/TS 8000-100:2011 Data Quality - Part 100:Overview."

Jacobson, I., G. Booch, et al. (1999). The Unified Software Development Process, Reading (MA): Addison-Wesley.

Juran, J. M. (1988). Juran on Planning For Quality. New York, NY, USA, McMillan.

Kock, N. and F. Lau (2001). "Information Systems Action Research: Serving Two Demanding Masters." Information Technology & People (special issue on Action Research in Information Systems) **14**(1): 6-11.

Koch, N. and A. Kraus (2002). The Expressive Power of UML-based Web Engineering. Second Int. Workshop on Web-oriented Software Technology (IWWOST '02). Málaga, Spain.: 105-119.

Koch, N. and A. Kraus (2003). Towards a Common Metamodel for the Development of Web Applications. Web Engineering. L. N. i. C. Science, Springer Berlin / Heidelberg: 419-422.

Koch, N. and C. Kroib (2008). UWE Metamodel and Profile. User Guide and Reference. T. Report, Institute for Informatics. Ludwig-Maximilians-Universitat Munchen (LMU), Germany.

Kotonya, G. and I. Sommerville (1996). "Requirements engineering with viewpoints." Software Engineering Journal **11**(1).

Kotonya, G. and I. Sommerville (2002). Requirements Engineering. Processes and Techniques.

Laudon, K. C. (1986). "Data Quality and Due Process in Large Interorganizational Record System." Communications of the ACM **29**(1): 4-11.

Levis, M., M. Helfert, et al. (2007). Information Quality Management: Review of an Evolving Research Area. ICIQ'07, MIT, Cambridge, MA, USA.

Lew, P., L. Olsina, et al. (2010). Quality, Quality in Use, Actual Usability and User Experience as Key Drivers for Web Application Evaluation. Web Engineering. B. Benatallah, F. Casati, G. Kappel and G. Rossi, Springer Berlin / Heidelberg. **6189**: 218-232.

Lewin, K. (1947). "Frontiers in Group Dynamics." Human Relations. **1**(1): 5-41.

Loshin, D. (2001). Enterprises Knowledge Management: The Data Quality Approach. San Francisco, CA, USA, Morgan Kauffman.

- Loucopoulos, P. and V. Karakostas (1995). System Requirements Engineering. New York, USA, McGraw-Hill.
- Lucas, A. (2010). Corporate Data Quality Management in Context. 15th International Conference on Information Quality, ICIQ 2010, Little Rock, Arkansas.
- Lwanga, Y., C. Walenta, et al. (2011). The Job of the Information/Data Quality Professional. I. International Association for Information and Data Quality.
- Mahdavi, M., J. Shepherd, et al. (2004). A Collaborative Approach for Caching Dynamic Data in Portal Applications. Proceedings of the fifteenth conference on Australian database.
- McNiff, J., P. Lomax, et al. (2003). You and Your Action Research Project, RoutledgeFalmer.
- McTaggart, R. (1991). "Principles of Participatory Action Research." Adult Education Quarterly. **41**(3): 168-187.
- Mehmood, K., S. Si-Said, et al. (2009). Data Quality Through Conceptual Model Quality - Reconciling Researchers and Practitioners through a Customizable Quality Model. International Conference on Information Quality, ICIQ 2009, Potsdam, Germany.
- Meliá, S. and J. Gómez (2005). Applying Transformations to Model Driven Development of Web applications. Perspectives in Conceptual Modeling. S. B. Heidelberg. **Volume 3770/2005**: 63-73.
- Mendes, E. and N. Mosley (2006). Web Engineering, Springer Berlin Heidelberg.
- Missier, P., S. Embury, et al. (2006). "Quality views: capturing and exploiting the user perspective on data quality." Proceedings of the 32nd international conference on Very large data bases- Volume 32.
- Moraga, M. A., C. Calero, et al. (2009). "Assessment of portlet quality: Collecting real experience." Comput. Stand. Interfaces **31**(2): 336-347.
- Moreno, N., S. Meliá, et al. (2008). Addressing New Concerns in Model-Driven Web Engineering Approaches. Web Information Systems Engineering - WISE 2008. J. Bailey, D. Maier, K.-D. Schewe, B. Thalheim and X. Wang, Springer Berlin / Heidelberg. **5175**: 426-442.
- Moreno, N., J. R. Romero, et al. (2008). An Overview Of Model-Driven Web Engineering and the Mda. Web Engineering: Modelling and Implementing Web Applications. G. Rossi, O. Pastor, D. Schwabe and L. Olsina, Springer London: 353-382.
- Moreno, N. and A. Vallecillo (2005). A Model-Based Approach for Integrating Third Party Systems with Web Applications. Web Engineering. D. Lowe and M. Gaedke, Springer Berlin / Heidelberg. **3579**: 441-452.

- Moreno, N. and A. Vallecillo (2008). "Towards interoperable Web engineering methods." Journal of American Society for Information Science and Technology **59**(7): 1073-1092.
- Murugesan, S. (2008). Web Application Development: Challenges and the Role of Web Engineering. Web Engineering: Modelling and Implementing Web Applications. G. Rossi, O. Pastor, D. Schwabe and L. Olsina.
- Murugesan, S., Y. Deshpande, et al. (2001). Web Engineering: a New Discipline for Development of Web-Based Systems. Web Engineering. S. Murugesan and Y. Deshpande, Springer Berlin / Heidelberg. **2016**: 3-13.
- Murugesan, S., A. Ginige, et al. (2005). Web engineering : introduction and perspectives, Hershey, Pa. Idea Group.
- Oliveira, P., F. Rodrigues, et al. (2005). A Taxonomy of Data Quality Problems. Second International Workshop on Data and Information Quality (in conjunction with CAISE'05), Porto, Portugal.
- Oliveira, P., F. t. Rodrigues, et al. (2005). A formal Definition of Data Quality Problems. Tenth International Conference on Information Quality (ICIQ'05), MIT, Cambridge, MA, USA.
- OMG. (2002). "Meta Object Facility (MOF) Specification." from <<http://www.omg.org/docs/formal/02-04-03.pdf>> [Accessed: 11-05-2005].
- OMG. (2003). "Common Warehouse Metamodel (CWM) Specification v1.1." October, 2008, from <<http://www.omg.org/docs/formal/03-03-02.pdf>> [Consulted: 29-09-2008].
- OMG (2003). MDA Guide Version 1.0.1., Object Management Group: 62.
- OMG. (2005). "MOF QVT Final Adopted Specification." from <<http://www.omg.org/docs/ptc/05-11-01.pdf>> [Accessed in February, 2005].
- OMG. (2005). "Unified Modeling Language: Superstructure. Versión 2.0." from <<http://www.omg.org/docs/formal/05-07-04.pdf>>.
- OMG. (2008). "MOF QVT Final Adopted Specification." from <http://www.omg.org/spec/QVT/1.0/> [Accessed in January, 2011].
- Oracle (2008). State of the Data Integration Market 2009-2009: 1-29.
- Padak, N. and G. Padak. (1994). "Research to Practice: Guidelines for Planning Action Research Projects." Retrieved September 27, 2010, from <http://archon.educ.kent.edu/Oasis/Pubs/0200-08.html>.
- Pastor, O., J. Gómez, et al. (2001). "The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming." Information Systems **26**: 507-534.

- Phan, D. D. and D. R. Vogel (2010). "A model of customer relationship management and business intelligence systems for catalogue and online retailers." Information & Management **47**(2): 69-77.
- Pipino, L., Y. Lee, et al. (2002). "Data Quality Assessment." Communications of the ACM **45**(4): 211-218.
- Powers, S., M. Brown, et al. (2011) "The Forrester Wave: Web Content Management for Online Customer Experience."
- Redman, T. (2000). Data Quality: The field guide. Boston, Digital Press.
- Scannapieco, M., B. Pernici, et al. (2002). IP-UML: Towards a Methodology for Quality Improvement Based on the IP-MAP Framework. International Conference on Information Quality, ICIQ-02.
- Shankaranarayan, G., R. Y. Wang, et al. (2000). IP-MAP: Representing the Manufacture of an Information Product. Fifth International Conference on Information Quality (ICIQ'2000), MIT, Cambridge, MA, USA.
- Shankaranayanan, G. and Y. Cai (2005). A Web Services Application for the Data Quality Management in the B2B Networked Environment. 38th Hawaii International Conference on System Sciences (HICSS-38 2005), Big Island, HI, USA, IEEE Computer Society.
- Sommerville, I. (2000). "Software Engineering. 6th Edition." from <http://www.comp.lancs.ac.uk/computing/resources/lanS/SE6/PDF/SEGlossary.pdf> [Accessed:05-04-2005].
- Strong, D., Y. Lee, et al. (1997). "Ten Potholes in the Road to Information Quality." IEEE Computer: 38-46.
- Strong, D. M., Y. W. Lee, et al. (1997). "Data Quality in Context." Communications of the ACM **40**(5): 103-110.
- Thi, T. T. P., J. Yao, et al. (2010). InfoGuard: A Process-Centric Rule-Based Approach for Managing Information Quality. European Research Consortium for Informatics and Mathematics ERCIM. **82**: 55-56.
- ThomsonReuters and Lepus (2010) "Thomson Reuters and Lepus Survey reveals Data Quality and Consistency Key to Risk Management and Transparency."
- Wadsworth, Y. (1998). "What is participatory Action Research? Action Research Internation, paper 2." Retrieved 20-marzo-2002, from <http://www.scu.edu.au/schools/sawd/ari/ari-wadsworth.html>.
- Wang, R. and D. Strong (1996). "Beyond accuracy: What data quality means to data consumers." Journal of Management Information Systems; Armonk; Spring 1996 **12**(4): 5-33.

- Wang, R. Y. (1998). "A Product Perspective on Total Data Quality Management." Communications of the ACM **41**(2): 58-65.
- Wang, R. Y. and S. Madnick (1993). Data Quality Requirements: Analysis and Modelling. Ninth International Conference on Data Engineering (ICDE'93), Vienna, Austria, IEEE Computer Society.
- Wang, R. Y., M. Reddy, et al. (1995). "Towards quality data: An attribute-based approach." Journal of Decision Support Systems **13**(3-4): 349-372.
- Weber, K., O. Boris, et al. (2009). "One Size Does Not Fit All---A Contingency Approach to Data Governance." J. Data and Information Quality **1**(1): 1-27.
- Yang, Z., S. Cai, et al. (2004). "Development and validation of an instrument to measure user perceived service quality of information presenting Web portals." Information and Management **42**(4): 575-589.