



Universidad Rey Juan Carlos

Escuela Técnica Superior de Ingeniería Informática

Departamento de Lenguajes y Sistemas Informáticos II

PREMISA: Un PRoceso para la Evolución y Modernización de SistemAs. Aplicación a Red.es

Jorge Moratalla Collado

Directora de Tesis: María Valeria de Castro
Co-director de Tesis: Marcos López Sanz

Móstoles, abril de 2012

La Dra. D^a María Valeria de Castro, Profesora Titular de Universidad Interina del Departamento de Lenguajes y Sistemas Informáticos II de la Universidad Rey Juan Carlos de Madrid y el Dr. D. Marcos López Sanz, Profesor Visitante de Universidad del Departamento de Lenguajes y Sistemas Informáticos II de la Universidad Rey Juan Carlos de Madrid, directores, de la Tesis Doctoral: *“PREMISA: Un proceso para la evolución y modernización de sistemas. Aplicación a Red.es”* realizada por el doctorando Jorge Moratalla Collado,

HACEN CONSTAR QUE:

Esta tesis doctoral reúne los requisitos para su defensa y aprobación

En Madrid, a 4 de abril de 2012

Fdo.: Ma. Valeria de Castro

Fdo.: Marcos López Sanz

A Santi y las “Cármenes” de mi vida

Resumen

En los últimos años, la necesidad de modernización de los sistemas heredados se ha convertido en uno de los principales problemas a los que se enfrentan las organizaciones, ya sea para incorporar nuevas funcionalidades o para adaptarse a diferentes plataformas tecnológicas. En ese sentido, cuestiones como la reducción de tiempos y costes, maximizar el retorno sobre la inversión (*ROI*), la externalización de las tareas de desarrollo y la fragmentación de servicios se han convertido en factores importantes en este proceso de modernización.

En la Administración Pública este aspecto es aún más crítico. La optimización de los recursos se convierte en un factor clave a la hora de realizar los procesos de modernización. Tal es el caso de Red.es, entidad pública vinculada al Ministerio de Industria, Energía y Turismo, cuya estrategia para el impulso de la sociedad de la información en este país hace hincapié en la necesidad de adaptar sus aplicaciones y sistemas de información de cara al marco de la nueva legislación relativa a la interoperabilidad y la administración electrónica, entre otros.

Para superar los retos derivados de esta situación, durante la última década, las empresas y organizaciones han adoptado el paradigma de orientación a servicios (*SOC, Service Oriented Computing*) como una estrategia global de modernización. La idea es basar la evolución de los sistemas en la aplicación de soluciones tecnológicas alineadas con sus dominios de negocios, donde el uso del SOC ha demostrado ser altamente efectivo. Sin embargo, es muy importante tener en cuenta que, por lo general, la mayor parte de los esfuerzos y recursos en los procesos de adopción de SOC se dedican a la modificación e integración de aplicaciones de TI existentes, más que a la implementación de nuevos componentes. En consecuencia, uno de los principales aspectos de la adopción de SOC para los propósitos de modernización es la posibilidad de identificar y reutilizar las funcionalidades actuales a la hora de definir el nuevo objetivo de la aplicación informática.

Para apoyar el proceso de modernización, resulta esencial el uso de técnicas de *gap analysis* para proporcionar una manera de identificar la forma en la que los servicios software disponibles pueden ser ensamblados dentro de otros nuevos, con objeto de satisfacer mejor la estrategia de la organización. El *gap analysis* ha sido ampliamente analizado en diversos ámbitos, que ofrecen diferentes enfoques destinados a ayudar a las organizaciones a obtener una mejor

comprensión de cómo se hacen las cosas hoy en día y cómo se puede hacer de forma más eficaz y eficiente.

Una de las características distintivas del *gap analysis* es que funciona en un alto nivel de abstracción. Sin embargo, como se ha señalado antes, el proceso de modernización comienza normalmente a un nivel más tecnológico. Como consecuencia, el uso exclusivo del gap analysis para los propósitos de modernización no es suficiente. En ese sentido, es necesario contar con herramientas adicionales para llevarla a cabo. En ese contexto, el Object Mangement Group (OMG) ha propuesto ADM (*Architecture Driven Modernization*) como un enfoque para la modernización de los sistemas de información. ADM se basa en la definición y la utilización de un conjunto de modelos y la aplicación de reglas de transformación entre ellos con objeto de: a) obtener modelos de alto nivel de abstracción a partir del análisis de las aplicaciones TI existentes, y b) realizar un análisis *top-down* de las nuevas aplicaciones de TI, definiendo nuevos modelos y definiendo nuevos modelos que pueden ser comparados con los ya existentes.

El propósito de esta Tesis es, pues, proponer un proceso de modernización, para la evolución de sistemas de información, para su aplicación a Red.es, y validarlo con un caso real. Este proceso estará basado en los pilares de la modernización dirigida por modelos, y el gap-analysis.

Abstract

In recent years, the need for modernization of legacy systems has become one of the major problems faced by organizations, either to incorporate new functionality or to adapt to different technological platforms. In that sense, issues such as reducing time and costs, maximizing return on investment (ROI), outsourcing of development tasks and service fragmentation have become important factors in this modernization process.

In Public Administration this subject is even more critical. The optimization of resources becomes a key factor when performing modernization processes. Such is the case of Red.es, a Public Entity linked to the Spanish Ministry of Industry, Energy and Tourism, whose forthcoming strategy for promoting the information society in this country emphasizes the need to adapt their applications and information systems to the new legislation framework aspects on interoperability and eGovernment, among others.

To overcome the challenges derived from this situation, during the last decade companies and organizations have adopted the Service Oriented Computing (SOC) paradigm as global modernization strategy. The idea is to base that modernization in the implementation of technological solutions aligned with their business domains, where the use of SOC has proved to be highly effective. However, it is very important to note that, usually, most of the efforts and resources in SOC adoption processes are devoted to the modification and integration of existing IT applications more than on deploying new ones. Accordingly, one of the main aspects of adopting SOC for modernization purposes is the possibility of to identify and re-use current functionalities when defining the new redefined target IT application.

To support the modernization process, we defend the use of a gap analysis approach to provide a way to identify how available software services (as-is) may be assembled within the newly conceived and redefined business models (to-be) to better meet the organization's goals. Gap analysis has already been extensively scrutinized in several domains offering different approaches aiming to help organizations to obtain a better understanding of how things are done today and how they can be made more effective and efficient.

One of the distinctive features of gap analysis is that it works at a high abstraction level. However, as it has been pointed out before, the modernization process normally starts at a more technological level. As a consequence, the

exclusive use of gap analysis for modernization purposes is not enough. In that sense, it is necessary to count with additional tools to perform it. In that context, the OMG has delivered ADM (Architecture Driven Modernization) as a proposal for the modernization of information systems. It is based on the definition and use of a set of models and the application of transformation rules between them for modernization purposes. This proposal will allow us: a) to obtain high level abstraction models (as-is models) starting from the analysis of existing IT application, and b) to go down to the new IT applications ones we have redefined to-be models and compared them with the existing ones.

So, the objective of this Thesis is to propose a modernization process, in order to apply to the case of Red.es. This process will be based on the model-driven modernization and the gap analysis.

Agradecimientos

En primer lugar, quiero dar las gracias a mis directores Valeria y Marcos. Gracias por vuestra desmesurada paciencia, por vuestra involucración, por vuestro apoyo y por tantas otras cosas sin las cuales este trabajo no hubiera sido posible. Con vosotros he aprendido tantas cosas...

A Esperanza, gracias por haberme dado la oportunidad de formar parte de este gran equipo de profesionales, y por darme el empujoncito final que me ha llevado a escribir estas "líneas".

A mis compañeros de Red.es, en especial a Jesús García Tello. Sin tu apoyo incondicional este proyecto (y tantos otros) hubiera quedado en una simple idea, gracias por confiar en mi (aunque todavía me sigo preguntado por qué lo haces ;)).

Y gracias a ti, Carmina, por estar ahí todos los días, por ser mi incondicional, por soportar mis malos momentos con una sonrisa, por ser mi paraguas en días de tormenta, mi compañera de viaje, mi consejera, mi amiga, mi todo.... Gracias por hacerme sentir importante...y sobre todo por haberte cruzado en mi camino. Este trabajo es tuyo.

Y a Santi y Carmen (cuando podáis leer estas líneas) gracias por ayudarme a levantarme pronto los días de fiesta, por hacerme abrir el portátil a cada momento, por acompañarme mientras vemos los "dibus", sin vosotros este proyecto tampoco hubiera sido posible. Gracias por ser como sois, espero no defraudaros en el camino que os queda por recorrer.

Gracias a todos!

Índice

1. INTRODUCCIÓN	22
1.1 Motivación.....	22
1.1.1 Contexto	22
1.1.2 La necesidad de modernización	24
1.2 Hipótesis y Objetivos.....	28
1.3 Método de investigación.....	29
1.3.1 Método de Resolución y Validación	30
1.4 Estructura de la Tesis.....	33
2. ESTADO DEL ARTE	36
2.1 Conceptos previos.....	36
2.1.1 Modernización dirigida por modelos (ADM)	37
2.1.2 Gap analysis	39
2.2 Propuestas de modernización dirigida por modelos	40
2.2.1 Criterios de evaluación	40
2.2.2 Ulrich y Newcomb	42
2.2.3 SEASAT	43
2.2.4 MARBLE.....	44
2.2.5 Cánovas y García Molina	46
2.2.6 El Proceso de modernización “YoYo”	47
2.2.7 SoSR	48
2.2.8 Baxter y Hendryx.....	48
2.2.9 Comparativa de trabajos.....	49
2.3 Propuestas de gap analysis.....	49
2.3.1 Criterios de evaluación	50
2.3.2 Yang y Papazoglou.....	51
2.3.3 Mantell	52
2.3.4 Papazoglou y van der Heuvel.....	53
2.3.5 Armstrong y Underbakke.....	56

2.3.6	<i>GAMBUSE</i>	57
2.3.7	<i>ARIS PPM</i>	59
2.3.8	<i>El Proceso de modernización “YoYo”</i>	60
2.3.9	<i>Comparativas de trabajos</i>	61
2.4	Conclusiones del capítulo	61
3.	PREMISA: PROCESO DE MODERNIZACIÓN DIRIGIDO POR MODELOS....	64
3.1	Esquema global de la propuesta	64
3.2	Descripción de los artefactos	66
3.2.1	<i>Descripción de los metamodelos</i>	66
3.2.1.1	El metamodelo Knowledge Discovery Metamodel (KDM)	66
3.2.1.2	El metamodelo Semantic Business and Vocabulary Rules (SBVR)68	
3.2.2	<i>Herramientas, frameworks y técnicas de soporte a la modernización</i>	68
3.2.2.1	A nivel lógico	68
3.2.2.2	A nivel de negocio.....	68
3.3	Etapas del proceso	69
3.3.1	<i>Paso 1: Obtención del modelo lógico (inicial)</i>	69
3.3.2	<i>Paso 2: Refinamiento del modelo lógico</i>	70
3.3.3	<i>Paso 3: Obtención de la capa de abstracción KDM</i>	71
3.3.4	<i>Paso 4: Obtención del modelo de negocio as-is</i>	72
3.3.5	<i>Paso 5: Obtención del modelo de negocio to-be</i>	73
3.3.6	<i>Paso 6: Mapeo los modelos de negocio as-is y to-be</i>	73
3.3.6.1	Definir los modelos <i>as-is</i> y <i>to-be</i> utilizando mecanismos formales74	
3.3.6.2	Aplicar operadores para identificar similitudes y diferencias entre los modelos de negocio.....	75
3.3.6.3	Obtener las operaciones a aplicar	77
3.3.7	<i>Paso 7: Obtención del KDM objetivo</i>	78
3.3.8	<i>Paso 8: Obtención de la solución objetivo (final)</i>	78
3.4	Descripción de las transformaciones	79
3.4.1	<i>Refinamiento del KDM</i>	80
3.4.2	<i>Obtención de la capa de abstracción</i>	83
3.4.3	<i>Obtención del modelo de negocio</i>	88

3.4.4	<i>Obtención del modelo lógico to-be</i>	90
3.4.5	<i>Obtención del código fuente</i>	94
4.	VALIDACIÓN: MODERNIZACIÓN EN RED.ES CON PREMISA	98
4.1	Método de Validación	98
4.2	Funcionalidad y ámbito del sistema de gestión de dominios “.es” ..	100
4.2.1	<i>Sistema original</i>	100
4.2.2	<i>Sistema inicial</i>	101
4.2.3	<i>Sistema objetivo</i>	106
4.3	Evolución y modernización del sistema de dominios.es.....	106
4.3.1	<i>Objetivos de modernización</i>	106
4.3.2	<i>Primera iteración</i>	107
4.3.2.1	Paso 1: Obtención del modelo lógico	107
4.3.2.2	Paso 2: Obtención del modelo de negocio <i>as-is</i>	109
4.3.2.3	Lecciones aprendidas de la primera iteración.....	109
4.3.3	<i>Segunda iteración</i>	110
4.3.3.1	Paso 1: Obtención del modelo lógico (inicial).....	110
4.3.3.2	Paso 2: Refinamiento del modelo lógico	110
4.3.3.3	Paso 3: Obtención de la capa de abstracción KDM.....	113
4.3.3.4	Paso 4: Obtención del modelo de negocio <i>as-is</i>	115
4.3.3.5	Paso 5: Obtención del modelo de negocio <i>to-be</i>	115
4.3.3.6	Paso 6: Mapeo los modelos de negocio <i>as-is</i> y <i>to-be</i>	122
4.3.3.7	Paso 7: Obtención del KDM objetivo.....	127
4.3.3.8	Paso 8: Obtención de la solución objetivo (final).....	129
4.3.3.9	Lecciones aprendidas de la segunda iteración	131
4.4	Conclusiones de capítulo	131
5.	CONCLUSIONES Y LÍNEAS FUTURAS	134
5.1	Análisis de la Consecución de Objetivos.....	134
5.2	Principales Aportaciones	135
5.3	Contrastación de Resultados.....	136
5.3.1	<i>Artículos en Congresos Internacionales:</i>	136
5.3.2	<i>Artículos en Congresos Nacionales:</i>	137
5.4	Líneas de Investigación Abiertas.....	137

REFERENCIAS	139
ANEXO I. GLOSARIO	145
ANEXO II. DESCOMPOSICIÓN FUNCIONAL EL SISTEMA ACTUAL.....	152
ANEXO III. ESPECIFICACIÓN DE CASOS DE USO	158
ANEXO IV. METAMODELOS KDM.....	186

Lista de Figuras

Figura 1. Esquema de la propuesta de modernización.....	28
Figura 2. Método de investigación	29
Figura 3. Etapa de Resolución y Validación	32
Figura 4. Aplicación del método de Investigación en Acción	33
Figura 5. Estructura de la Tesis	34
Figura 6. Tipos de modernización en ADM	38
Figura 7. Interfaz web en la propuesta de Armstrong y Underbake	57
Figura 8. Esquema global de PREMISA	65
Figura 9. Etapas del gap analysis.....	74
Figura 10. Formato de ficha para explicar las reglas de transformación	79
Figura 11. Resumen de reglas de transformación.....	80
Figura 12. Proceso de Validación de este trabajo de Tesis Doctoral	99
Figura 13. Arquitectura inicial del Sistema (2003).....	100
Figura 14. Arquitectura complementaria del Sistema (2004).....	101
Figura 15. Infraestructura del sistema de dominios	103
Figura 16. Infraestructura DNS	104
Figura 17. Interfaces de acceso al sistema actual	106
Figura 18. Modelo KDM generado con MoDISCO	108
Figura 19. Modelo KDM refinado para la transmisión de dominios	112
Figura 20. Capa de abstracción KDM para la transmisión de dominios	114
Figura 22. Modelo SBVR para la solución actual de transmisión de dominios ..	116
Figura 23. Análisis DAFO de la herramienta de gestión de dominios .es	116
Figura 24. Modelo de negocio de la solución objetivo para transmisión de .es ..	123
Figura 25. Modelo lógico para la solución objetivo	128
Figura A2- 1. Descomposición funcional para el usuario no registrado.....	152
Figura A2- 2. Descomposición funcional para el usuario del CAU	152
Figura A2- 3. Descomposición funcional para el usuario registrado.....	153
Figura A2- 4. Descomposición funcional para el operador de Red.es.....	154
Figura A2- 5. Descomposición funcional para el agente registrador.....	155

Lista de Tablas

Tabla 1. Criterios de evaluación para las propuestas de modernización.....	42
Tabla 2. Evaluación de las propuestas de modernización.....	49
Tabla 3. Criterios de evaluación para las propuestas de gap analysis.....	51
Tabla 4. Elementos en GAMBUSE.....	58
Tabla 5. Operadores GAMBUSE.....	59
Tabla 6. Evaluación de las propuestas de gap-analysis.....	61
Tabla 7. Artefactos básicos para PREMISA.....	67
Tabla 8. Resumen de artefactos del paso 1 de PREMISA.....	70
Tabla 9. Resumen de artefactos del paso 2 de PREMISA.....	71
Tabla 10. Resumen de artefactos del paso 3 de PREMISA.....	72
Tabla 11. Resumen de artefactos del paso 4 de PREMISA.....	72
Tabla 12. Resumen de artefactos del paso 5 de PREMISA.....	74
Tabla 13. Notación para representar los modelos de negocio en PREMISA.....	75
Tabla 14. Mapeo entre SBVR y notación formal para el gap-analysis.....	76
Tabla 15. Operadores propuestos en PREMISA para el gap-analysis.....	76
Tabla 16. Resumen de artefactos del paso 6 de PREMISA.....	78
Tabla 17. Resumen de artefactos del paso 7 de PREMISA.....	78
Tabla 18. Resumen de artefactos del paso 8 de PREMISA.....	79
Tabla 19. Objetivos para la mejora de procesos del caso de estudio.....	122

Capítulo 1. Introducción

Este capítulo describe la motivación que ha originado la presente Tesis Doctoral, es decir, la modernización de sistemas en Red.es. Se detallará también la hipótesis y objetivos principales que esta Tesis pretende cubrir, así como los pasos seguidos para la definición del proceso de modernización y su validación, todo ello apoyado en un caso de uso real actualmente implantado en un entorno productivo empresarial en Red.es. También se explica en detalle el método de investigación utilizado, así como la descripción global de la estructura del trabajo.

1.1 Motivación

1.1.1 Contexto

Una sociedad de la información es aquella en la cual la creación, distribución y manipulación de la información forman parte importante de sus actividades culturales y económicas. Los sectores relacionados con las tecnologías de la información y la comunicación (TIC) desempeñan un papel particularmente importante dentro de este esquema.

Aunque Internet ha desempeñado un papel muy importante como un medio que facilita el acceso e intercambio de información y datos, la sociedad de la información no está limitada sólo a él. En su origen, las redes conectadas a Internet no eran muy extensas, por lo que se solía trabajar con tablas de conversión manual (archivos secuenciales que asociaban en cada línea un literal a cada dirección IP de la máquina relacionada) para referenciar a los equipos conectados a las mismas, de una forma transparente.

Sin embargo, el anterior sistema de tablas de conversión exigía una actualización manual para la totalidad de los equipos, en caso de incluir o modificar el nombre de una máquina. Por lo tanto, con el aumento en tamaño de las redes y sus interconexiones, fue necesario implementar un sistema de gestión para los nombres, que fuese jerárquico y fácil de administrar. Con tal objetivo, Paul Mockapetris creó en noviembre de 1983 el denominado Sistema de Nombres de Dominio (DNS) [59][60], basado en una estructura jerárquica que se encarga de traducir las direcciones IP de cada nodo activo en la red, a términos memorizables y fáciles de encontrar, denominados nombres de dominio de Internet. Esta abstracción hace posible, entre otros, que cualquier servicio de red

pueda moverse de un lugar geográfico a otro, aun cuando el cambio implique una dirección IP diferente. Sin la ayuda de este sistema, los usuarios de Internet tendrían que acceder a cada servicio web utilizando la dirección IP del nodo.

Este sistema, que ha sido sometido a varias Request For Comments (RFC) [59][60], contempla dentro de su jerarquía, dos grandes categorías:

- **Dominios genéricos de primer nivel (*generic Top-Level Domain, gTLD*):** son aquellos utilizados por una clase particular de organizaciones (por ejemplo, *.com* para organizaciones comerciales, *.gov* para entidades gubernamentales, *.mil* para organismos militares, etc.). La mayoría de los *gTLDs* están disponibles para el uso mundial, pero por razones históricas, algunos de ellos están restringidos para su utilización por las respectivas autoridades estadounidenses.
- **Dominios de nivel superior geográfico, o de nivel superior de código de país (*country code Top-Level Domain, ccTLD*):** son los dominios reservados para un país o territorio dependiente (por ejemplo “.es” para España). Cada país designa gestores para su administración, delegados por IANA¹ (*Internet Assigned Numbers Authority*) e ICANN² (*Internet Corporation for Assigned Names and Numbers*), que determinan las normas y reglas aplicables para cada conjunto de nombres de dominio. Por ejemplo, para el caso de España [10], se ha determinado que únicamente podrán solicitar la asignación de un dominio bajo “.es”, todas aquellas personas físicas o entidades con o sin personalidad jurídica que mantengan intereses o vínculos con España, es decir, aquellas que cumplen algunas de las siguientes premisas: están establecidas en España o dirigen total o parcialmente sus servicios al mercado español u ofrecen información o productos vinculados cultural, histórica o socialmente con España.

En el caso de nuestro país, la Autoridad de Asignación de los dominios de nivel superior geográfico del “.es”, corresponde a la Entidad Pública Empresarial Red.es [10][11][14], organismo dependiente de la Secretaría de Estado de

¹ Autoridad para la Asignación de Números de Internet, responsable de la coordinación global de los protocolos de Raíz DNS, direccionamiento IP y otros recursos del Protocolo de Internet

² La Corporación de Internet para la Asignación de nombres y números de Dominios es una organización que opera a nivel de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de la administración del sistema de servidores raíz.

Telecomunicaciones y para la Sociedad de la Información del Ministerio de Industria, Energía y Turismo, y que nació con el objeto de impulsar la Sociedad de la Información en España.

Actualmente, Red.es, además de ser la responsable de la gestión de los dominios “.es”, es la encargada de ejecutar gran parte de los proyectos que tienen como objetivo el impulso de la Sociedad de la Información [54][53], interactuando para ello con Comunidades Autónomas, Diputaciones, Entidades Locales y el sector privado en materia de Tecnologías de la Información y la Comunicación (TIC).

Para desarrollar la labor que Red.es tiene encomendada (en lo relativo a la asignación y gestión de nombres de dominio), se dispone de un Sistema Informático, así como una infraestructura hardware y de comunicaciones, que se encarga de permitir a todos los *stakeholders* (entendido como todos los implicados en el ciclo de asignación de los “.es”) interactuar entre sí, con el objeto de solicitar, administrar y gestionar los nombres de dominio bajo este indicativo.

Este sistema se concibió inicialmente en un contexto en el cual el parque de los “.es” era reducido. Sin embargo este número ha ido creciendo de forma exponencial hasta llegar actualmente a gestionar más de 1’5 millones de “.es”.

Unido a esta progresión, los actores involucrados han ido evolucionando, tanto en número como en las funciones demandadas, en base a los cambios de Normativa y de Legislación vigente.

Para adaptarse a esta situación, el sistema ha ido evolucionando a demanda, mediante la aplicación de sucesivos parches, para adaptarse a las necesidades operativas de cada momento. Esta evolución no ha sido tan óptima como cabría esperar, debido, sobre todo, a la urgencia con la que era necesario aplicar la mayoría de estas adaptaciones. Así mismo, en cada versión evolutiva, no se aplicaban cambios estructurales ni se aprovechaban las ventajas tecnológicas disponibles en cada momento. El resultado: un sistema heredado difícil de evolucionar y poco documentado, que incurre en altos costes de mantenimiento.

1.1.2 La necesidad de modernización

Para dar respuesta a estas necesidades se ha planteado en Red.es un proceso de modernización [29]. Este proceso se deberá seguir teniendo como hilo conductor el modelo de desarrollo sostenible (integración, estándares, escalabilidad y mantenimiento).

Este enfoque debe estar por lo tanto orientado al mantenimiento y al crecimiento, lo que permitirá a Red.es continuar el desarrollo del sistema, añadiendo nuevas funcionalidades en cada una de las fases, o adaptando la arquitectura sin que estas afecten a las ya existentes.

Con objeto de facilitar la necesidad de modernización comentada anteriormente, sobre todo en lo relativo a la reducción de tiempos y costes, la externalización y fragmentación del desarrollo se convierten en piezas clave dentro del mismo. En este sentido, la adopción de modelos de negocios basados en servicios, evidencia la necesidad de desarrollar sistemas software cuya estructura y comportamiento es susceptible de sufrir cambios.

La herramienta para la gestión de nombres de dominio “.es” es un claro ejemplo de ello: las necesidades operativas han obligado a Red.es a realizar sucesivos cambios en sus sistemas de cara a adaptarlos a las nuevas especificaciones funcionales y normativas entre otras, lo que ha convertido a la herramienta para la gestión de los dominios en un sistema heredado difícil de mantener e incluso de evolucionar.

Si añadimos a este escenario la necesidad de desarrollar servicios software como pieza de intercambio para facilitar la interoperabilidad en la Administración Pública, se hace indispensable contar con un paradigma adecuado. En este sentido, la **orientación a servicios (SOC, *Service Oriented Computing*)** representa un cambio importante en la forma en que el software es analizado, diseñado, construido, entregado y utilizado actualmente, por lo que ha llevado a reconsiderar las técnicas actuales de desarrollo de sistemas de información.

Desde el punto de vista tecnológico, utilizando los principios de este paradigma, es posible crear sistemas débilmente acoplados, dinámicos y que se adaptan a las nuevas necesidades de negocio. Este enfoque parece, por lo tanto, adecuarse bastante bien a uno de los objetivos pretendidos como resultado del proceso de modernización del sistema de gestión de los nombres de dominio “.es”, el de la adaptación al nuevo esquema nacional de interoperabilidad (ENI), que permitirá la interconexión entre las distintas Administraciones Públicas y empresas privadas, como paso previo hacia una eAdministración.

De esta forma, los servicios se erigen como la pieza clave en un desarrollo de este tipo, si lo que se pretende es, además, beneficiarse de las ventajas que este planteamiento puede aportar, entre las cuales destacan la reusabilidad, la flexibilidad y la interoperabilidad del software. Todas ellas redundarán en unos menores costes de mantenimiento y evolución, potenciando además el aumento de la escalabilidad del Sistema o la mejora en los tiempos de respuesta al negocio

(*time-to-market*), sin olvidar que un enfoque de este tipo facilita la externalización de los procesos de negocio, que es una de las piezas clave en la estrategia de Red.es.

Sin embargo, la correspondencia entre los servicios de alto nivel (como parte del proceso económico y financiero) y su implementación mediante las tecnologías de servicios actuales no es sencilla y necesita de un proceso de transformación que dista de ser trivial.

En este sentido, la utilización de **enfoques de desarrollo dirigido por modelos (MDE, *Model Driven Engineering*)** [21] se ha consolidado en los últimos años como una apuesta importante, que cada vez cobra más protagonismo en el desarrollo de sistemas de información, como principal ayuda para solventar el salto desde las especificaciones de servicios de negocio hasta la obtención de sistemas de información basados en servicios.

La utilización de este tipo de enfoques de desarrollo en conjunción con el paradigma orientado a servicios ha demostrado ser de gran utilidad en los últimos años. El alto número de proyectos europeos dedicados a este fin en los que las empresas han estado y están altamente involucradas corroboran esta afirmación.

A pesar de que este enfoque parece ser el más adecuado para abordar los procesos de modernización de sistemas, y para el caso que nos concierne, la reingeniería de la herramienta de gestión de nombres de dominio “.es”, hay que tener en cuenta que, como suele suceder con la mayoría de sistemas empresariales, nos encontramos con una herramienta bastante compleja, heterogénea y enfocada a una misión concreta, lo que provoca que esté fuertemente ligado a unos determinados flujos de negocio. De hecho, en el mundo empresarial raramente se construyen sistemas partiendo de cero (procesos de negocio hardcodeados, sin tener en cuenta la independencia funcional que debe regir el diseño de un sistema), lo que se convierte en una desventaja a la hora de rediseñarlos, o integrarlos en otros flujos más complejos. La mayoría de estos sistemas se convierten, como ha ocurrido con la herramienta para la gestión de los dominios “.es”, en sistemas de información heredados (*Legacy Systems*), debido a los años de mantenimiento, que suelen provocar la acumulación de parches, degradando la arquitectura del mismo e impactando en altos costes de mantenimiento.

Partiendo de esta base, y siguiendo el enfoque orientado a modelos, el *Object Management Group* (OMG) ha establecido un conjunto de soluciones para la modernización de sistemas de información [4], denominado **ADM (*Architecture Driven Modernization*)**, basado en la utilización de modelos y la aplicación de reglas de transformación entre ellos.

La iniciativa ADM define una serie de metamodelos basados en estándares, y una serie de transformaciones entre ellos para facilitar la representación de la solución para cada uno de estos dominios. ADM contempla la modernización como un conjunto de transformaciones para llegar a una solución objetivo (también denominada *to-be*), partiendo de una solución existente (también llamada *as-is*). Estas transformaciones se pueden aplicar a distintos niveles de abstracción:

- **Nivel físico:** la transformación más simple, corta y por lo tanto menos compleja
- **Nivel lógico:** utilizada para proyectos de complejidad media, en los que se busca un cambio o una mejora arquitectónica
- **Nivel de negocio:** implica a todos los niveles (o al menos los dos superiores), y es la más utilizada en los proyectos de Reingeniería. A este nivel se suelen representar las reglas o los procesos de negocio de la solución.

Para apoyar el proceso de modernización, resulta esencial el uso de técnicas de **gap analysis** para proporcionar una manera de identificar la forma en la que los servicios disponibles pueden ser ensamblados dentro de otros nuevos, con objeto de satisfacer la estrategia de la organización. Una de las principales características del gap analysis es que funciona en un alto nivel de abstracción. Sin embargo, como se ha señalado anteriormente, el proceso de modernización comienza normalmente a un nivel más tecnológico.

La unión de ambos, ADM y gap analysis, ayudará por lo tanto a alinear negocio y tecnología; el primero, a representar y transformar, mediante modelos y reglas respectivamente, los distintos niveles de abstracción; el segundo a comparar los modelos de alto nivel entre las soluciones actual y objetivo.

En este sentido, para dar soporte al proceso de modernización en Red.es, se propondrá en la presente Tesis la definición de un marco de trabajo, basado en modelos, que combine técnicas de gap analysis y modernización dirigida por modelos.

Este proceso, al que hemos denominado **PREMISA (PRoceso para la Evolución y Modernización de SistemAs)**, se planteará desde un enfoque práctico, que permita evolucionar la solución actual (*as-is*) hacia la solución objetivo (*to-be*). Todo ello utilizando modelos para representar la solución a cada uno de los niveles de abstracción, aplicando reglas de transformación para pasar de un modelo a otro, y mapeando mediante técnicas de gap analysis los modelos de negocio de ambas soluciones, como muestra la Figura 1.

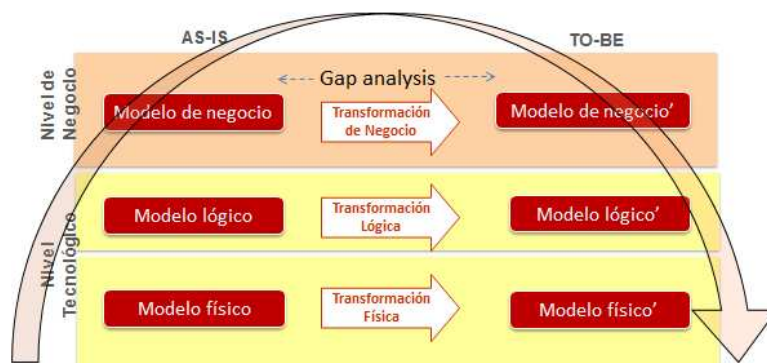


Figura 1. Esquema de la propuesta de modernización

La validación de esta Tesis se llevará a cabo mediante su aplicación al caso real de Red.es. En concreto se trabajará con el sistema de gestión de nombres de dominios “.es”, y se analizarán los resultados reales obtenidos, con el propósito de verificar la hipótesis planteada.

1.2 Hipótesis y Objetivos

La hipótesis planteada en la presente Tesis Doctoral determina que **“es posible definir un proceso que permita la evolución de sistemas empresariales, mediante la combinación de técnicas basadas en gap-analysis y modernización dirigida por modelos”**.

Por lo tanto, el objetivo principal del proyecto de Tesis que se presenta, es definir un proceso que combine técnicas de gap analysis y modernización dirigida por modelos, para la evolución y modernización de sistemas empresariales, centrado en el análisis de las reglas de negocio.

Para conseguir este objetivo principal se desarrollarán los siguientes objetivos parciales:

1. Estudiar, entender y analizar los principios y el estado del arte de la modernización dirigida por modelos y del gap analysis, y cómo éstos pueden aplicarse a Red.es.
2. Proponer un proceso para la modernización de sistemas, basado en la modernización dirigida por modelos y el gap analysis.
3. Validar el proceso en el caso real de Red.es.

1.3 Método de investigación

El método de investigación utilizado en este trabajo de Tesis Doctoral es el resultante de la combinación del método propuesto por Marcos y Marcos [39], para la investigación en el campo de la Ingeniería del Software, y del método denominado Investigación en Acción (Action Research) (Avison *et al.* [7]).

En Marcos y Marcos [39] se propone un método de investigación (Figura 2), basado en el método hipotético-deductivo (Bunge [17]), que consta de una serie de etapas que, por su generalidad, son aplicables a cualquier tipo de investigación. Dichas etapas comprenden: la determinación del problema, la creación de la hipótesis, la definición del método de trabajo, la resolución y validación, el análisis de los resultados y elaboración de conclusiones, y la redacción de un informe final.

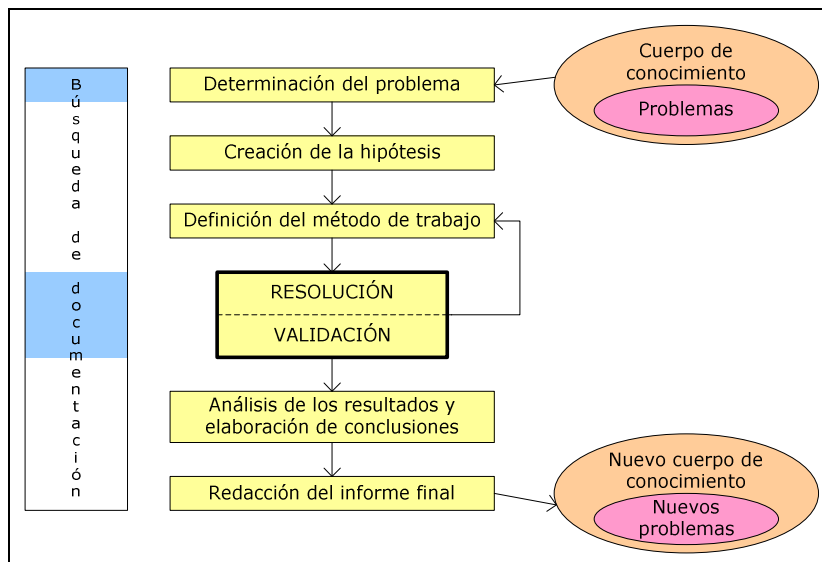


Figura 2. Método de investigación

Como puede apreciarse en la Figura 2, en el método de investigación se incluye una etapa de definición del propio método. Esta etapa es necesaria, puesto que cada investigación posee sus propias características, lo que implica que no exista un método genérico y universal que pueda aplicarse de forma directa y sin modificaciones a cualquier trabajo de investigación. Según sea la naturaleza del problema a resolver, se utilizará un método deductivo, experimental, etc.

Otra de las etapas del método de investigación es la de resolución y validación. En este caso, investigación en el campo de la Ingeniería del Software y

cuyo objetivo es la elaboración de una aproximación basada en modelos y gap analysis, para la etapa de resolución y validación utilizamos una adaptación del método de investigación en acción, trabajando sobre un caso de uso real en Red.es.

El método de investigación en acción utilizado en la resolución y validación de esta Tesis Doctoral se describe en el apartado siguiente.

1.3.1 Método de Resolución y Validación

La Investigación en Acción (Avison *et al.* [7]) es un método de investigación cualitativo utilizado para la validación de los trabajos de investigación mediante su aplicación en proyectos reales, reforzando la interacción entre los investigadores y los participantes de dichos proyectos. Este método, por su carácter de validación práctica, es especialmente apropiado para la investigación en Ingeniería y, específicamente, en Ingeniería del Software.

El método de investigación en acción es “la forma que tienen los grupos de personas para preparar las condiciones necesarias para aprender de sus propias experiencias y hacer estas experiencias accesibles a otros” (Mc Taggar [40]). Más concretamente, puede definirse como “el proceso de recopilar de forma sistemática datos de la investigación acerca de un sistema actual en relación con algún objetivo, meta o necesidad de ese sistema; de alimentar de nuevo con esos datos al sistema; de emprender acciones por medio de variables alternativas seleccionadas dentro del sistema, basándose tanto en los datos como en las hipótesis; y de evaluar los resultados de las acciones, recopilando datos adicionales” (French y Bell [28]).

Una de las características más relevantes de este método es que la investigación se realiza a la vez que se aplican sus resultados, de modo que esta aplicación permite validar y refinar los resultados de la investigación en un proceso iterativo. El proceso de investigación definido en el método de investigación en acción no es por tanto un proceso lineal, sino que va avanzando mediante la realización de ciclos, llamados ciclos de investigación en acción, en cada uno de los cuales se ponen en marcha nuevas ideas, que son puestas en práctica y comprobadas hasta el siguiente ciclo (Wadsworth [78]).

La Figura 3 representa las etapas definidas en (Susman y Evered [67]) para cada ciclo de investigación en acción. Como puede verse en la figura, en este caso, los ciclos de investigación en acción se realizan durante la etapa de resolución y validación de la Figura 2.

A continuación, se describen las etapas de cada ciclo de investigación en acción:

- **Diagnóstico:** es la etapa inicial de cada iteración, involucra la identificación de los aspectos a mejorar y los problemas a resolver propios de cada iteración. Los problemas que se identifican y describen en esta etapa guardan relación con los problemas identificados durante la etapa de determinación del problema, ilustrada en la Figura 2. Sin embargo, es importante tener en cuenta que, según van sucediéndose las iteraciones sobre los diferentes casos de estudio, pueden surgir nuevos problemas y oportunidades de mejora de los resultados de las iteraciones anteriores.
- **Planificación de la Acción:** en esta etapa se consideran los diferentes cursos de acción a tomar para resolver los problemas detectados en la etapa anterior, a partir de los casos de estudios identificados preliminarmente durante la etapa de definición del método de trabajo (ver Figura 2). El resultado de esta actividad es la identificación de una serie de acciones a ejecutar sobre un determinado caso de estudio.
- **Ejecución de la Acción:** implica la implementación del curso de acción elegido en la etapa anterior sobre el caso de estudio seleccionado.
- **Evaluación:** después de completar la acción, se examinan los resultados. La evaluación incluye determinar si se han alcanzado los efectos esperados de la acción, y si esos efectos mitigaron los problemas identificados previamente. Si los resultados obtenidos fueran no satisfactorios, los problemas identificados se trasladan a las siguientes iteraciones del ciclo de investigación en acción.
- **Especificación del aprendizaje:** si bien esta actividad figura al final del ciclo, se trata de una actividad que se lleva a cabo durante todo el ciclo. En ella, los conocimientos adquiridos durante el ciclo de investigación en acción, por más que hayan sido no satisfactorios, se comparten con las personas involucradas en el caso de estudio, para que dicho conocimiento pueda ser asimilado en las tareas que realizan y para que se puedan planificar nuevos ciclos de investigación en acción.

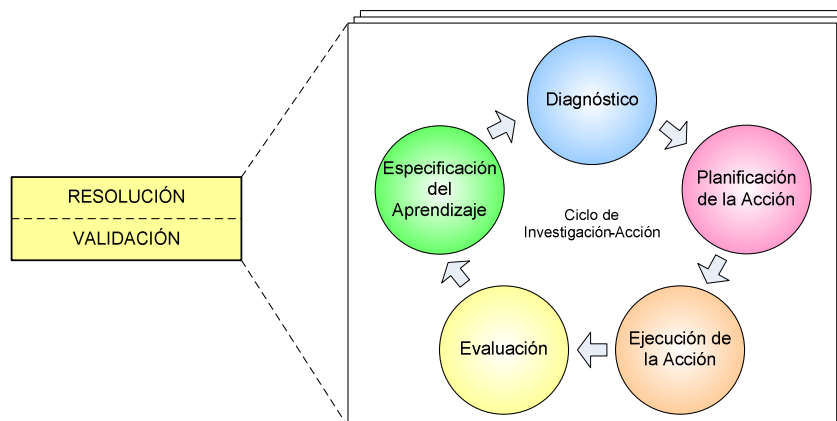


Figura 3. Etapa de Resolución y Validación

En el caso concreto de esta Tesis Doctoral, cada ciclo de investigación en acción se aplicó iterativamente sobre un caso real en Red.es. Desde el punto de vista de las ciencias sociales, los casos de estudio se utilizan para investigar una situación o fenómeno que ocurre en un contexto real (Yin [80]). Sin embargo, cuando se habla de investigación en sistemas software, como es el caso de esta Tesis Doctoral, los casos de estudios pueden abordar casos como el desarrollo o la implantación de un sistema de información en una organización particular (Cornford y Smithson [22]).

El caso de estudio que aquí se presenta es un caso real, que consiste en la modernización del sistema para la gestión de nombres de dominios “.es”. Este caso real se describe detalladamente en el capítulo 4 de esta memoria. A continuación, y a modo de ejemplo, en la Figura 4 se presenta la aplicación del método de Investigación en Acción a este caso de estudio, desarrollado para la validación de esta Tesis Doctoral. En la Figura 4 se reflejan también los diferentes actores que, según Wadsworth [78], intervienen en el proceso:

- El investigador, es aquel que impulsa, como sujeto, el proceso investigador. En este caso, el doctorando.
- El objeto investigado, que es el problema que se desea resolver. En este caso, la problemática de modernización dirigida por modelos unida al gap analysis.
- El grupo crítico de referencia, es aquel para quien se investiga y que tiene un problema por resolver, por lo que participa en la investigación. En este

caso podría definirse como el conjunto de desarrolladores y de analistas de negocio que han participado en la modernización del sistema.

- Beneficiarios de la investigación, son aquellos que pueden beneficiarse del resultado de la investigación, aunque no haya participado directamente en el proceso. En este caso, cualquiera que pueda verse beneficiado por la aplicación del método propuesto. Los principales beneficiados de esta investigación sería Red.es.

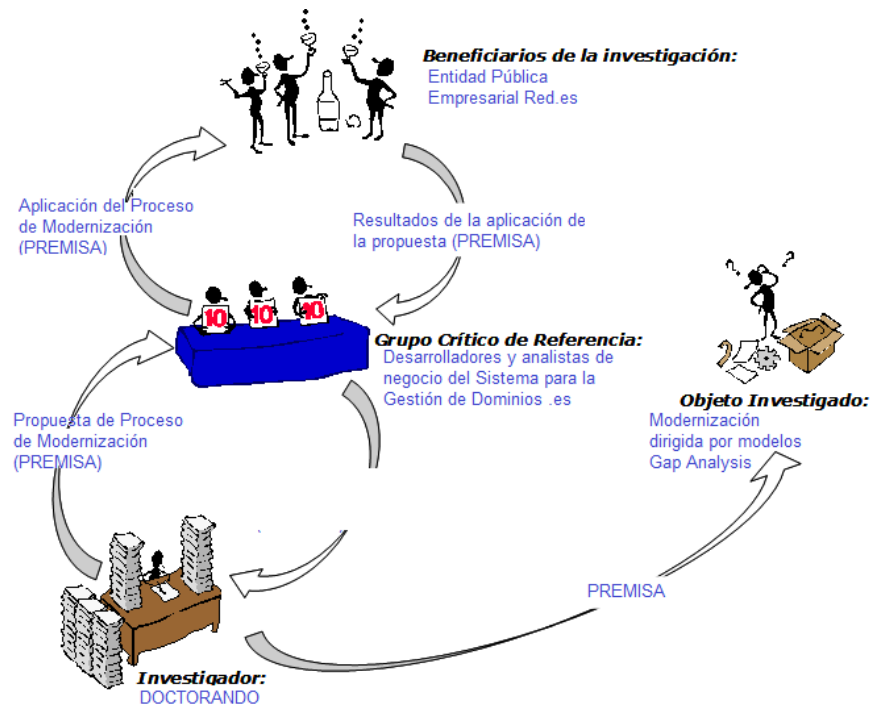


Figura 4. Aplicación del método de Investigación en Acción

1.4 Estructura de la Tesis

Para tener una mejor comprensión de la estructura de la Tesis, la Figura 5 detalla la composición de ésta.

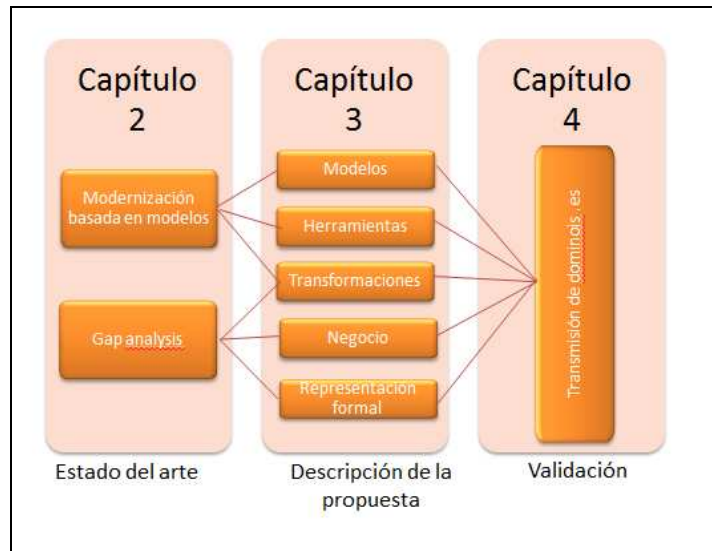


Figura 5. Estructura de la Tesis

La Tesis se compone de los siguientes capítulos:

- **Capítulo 2.** Estado del Arte. Presenta un estudio detallado de los principales trabajos relacionados con a) paradigma orientado a servicios (SOC), b) modernización basada en modelos y c) el gap-analysis
- **Capítulo 3.** El proceso propuesto (PREMISA). Presenta la propuesta del marco de trabajo definido para la modernización de sistemas, siguiendo el enfoque dirigido por modelos, y teniendo en cuenta el gap-analysis. Para ello se presentarán los metamodelos y las reglas de transformación para llegar a la solución objetivo partiendo del sistema actual.
- **Capítulo 4.** Validación. Detalla la aplicación del marco propuesto al caso concreto de Red.es, que permite validar la viabilidad del proceso definido, dentro del marco empresarial de modernización de sistemas heredados.
- **Capítulo 5.** Conclusiones y trabajos futuros. En este capítulo las principales conclusiones se establecen desde el análisis del logro de los objetivos marcados, así como las líneas de continuación derivadas de la Tesis.

Capítulo 2. Estado del Arte

Este capítulo presenta un estudio sobre el estado del arte en las áreas de investigación a las que se refiere la presente Tesis Doctoral. Se considera como una parte fundamental de ésta, ya que ayudará a reconocer las contribuciones que el presente trabajo puede añadir a los conocimientos existentes en el contexto de investigación abordado (de cara a cumplir el primero de los subobjetivos marcados en el capítulo 1) y a posicionar nuestra propuesta, a la que hemos denominado PREMISA (PRoceso para la Evolución y ModernIzacion de SistemAs), en relación a los demás trabajos relacionados.

Con esta idea en mente, en primer lugar, se procederá a definir con exactitud un criterio de clasificación coherente que sirva para realizar un análisis de los trabajos relevantes que pueden tener una influencia en el ámbito de aplicación de la tesis doctoral presentada. Por lo tanto, la sección 2.1 comprende el conjunto de conceptos que permiten la evaluación de cada iniciativa en relación con la modernización de sistemas heredados, teniendo en cuenta el gap analysis. A continuación, en la sección 2.2 se procede a aplicar esos criterios a los trabajos que están relacionados con estas áreas. Por último, este capítulo muestra las conclusiones más importantes del análisis realizado.

2.1 Conceptos previos

Los dos grandes pilares en los que se basa esta Tesis (modernización de sistemas y gap-analysis) tienen una gran relevancia dentro del presente contexto de investigación. Para seleccionar y evaluar aquellas iniciativas, propuestas y trabajos que tienen una influencia directa en la presente propuesta, esta sección realizará una revisión y clasificación de los mismos, de acuerdo al grado de afinidad con la misma. De esta forma, los trabajos considerados deberán necesariamente contemplar:

- Enfoques de modernización de sistemas (sección 2.2.1). Dentro de esta categoría se engloban aquellos trabajos que traten la modernización de sistemas, ya sea desde el punto de vista metodológico o no, siempre que el proceso se aborde desde un enfoque dirigido por modelos, o considere el paradigma orientado a servicios.
- Propuestas que traten el gap-analysis (sección 2.2.2). Esta categoría considera los trabajos, propuestas o estándares que contemplen técnicas

de gap-analysis, ya sea en conjunción con propuestas de modernización o no.

Describiremos a continuación cada uno de estos pilares que conformarán las bases de nuestra propuesta de modernización.

2.1.1 Modernización dirigida por modelos (ADM)

La iniciativa *Architecture Driven Modernization* (ADM) es una propuesta del *Object Management Group* (OMG), destinada a la modernización de sistemas basada en modelos. ADM [4] define una serie de metamodelos basados en estándares, y una serie de transformaciones entre ellos para facilitar la representación de la solución para cada uno de estos dominios. ADM se apoya en el estándar MOF (*Meta Object Facility*) [41] para describir conceptos de forma gráfica.

ADM propone la modernización de sistemas basada en el uso de una serie de transformaciones para conseguir una solución objetivo (*to-be*) partiendo de la solución actual (*as-is*). Estas transformaciones pueden ser aplicadas a cada uno de los niveles definidos por los modelos físico (ASTM), lógico (KDM) o de negocio (SBVR), como se muestra en la Figura 6:

- La transformación más simple, corta y por lo tanto menos compleja, es la correspondiente a una modernización de tipo físico, caracterizada por aportar valor no tanto al negocio, como a la infraestructura empresarial.
- La transformación o modernización lógica correspondería a proyectos de complejidad media, en los que se busca un cambio o una mejora arquitectónica, como puede ser la de transformar un Sistema basado en una Arquitectura Tecnológica de 3-capas a un enfoque SOA.
- Por último, la modernización más compleja, la transformación basada en el Negocio, es aquella que implica a los tres niveles (o al menos los dos superiores), que estaría dirigida por la estrategia de la Empresa, y que busca sobre todo aportar valor al Negocio. Es la más utilizada en los proyectos de Reingeniería.

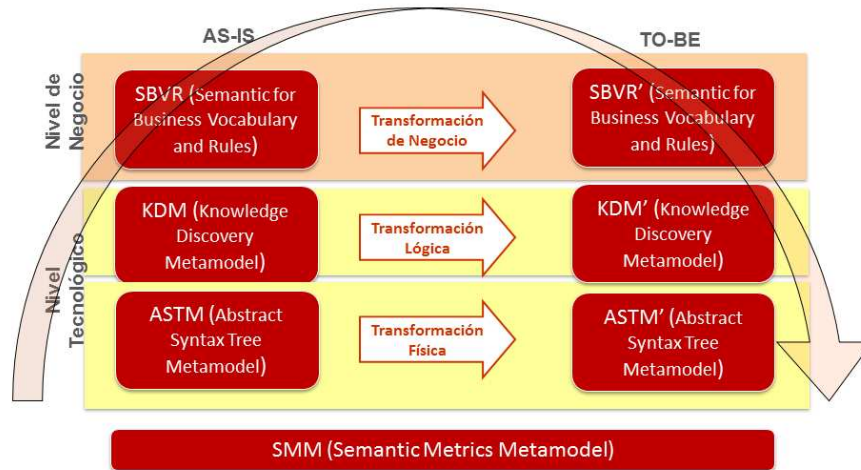


Figura 6. Tipos de modernización en ADM

Los metamodelos contemplados en ADM, son:

- *Abstract Syntax Tree Metamodel (ASTM)*, que representa la vista más granular de la Arquitectura Tecnológica del Sistema y tiene como principal objetivo soportar la transformación del mismo desde el punto de vista de los lenguajes y plataformas existentes. Este metamodelo es descrito por el OMG en [1].
- *Knowledge Discovery Metamodel (KDM)*, descrito dentro de la iniciativa ADM del OMG en [32], representa los aspectos de la solución tecnológica y la estructura lógica del sistema, y tiene como objetivo facilitar la interoperabilidad para cualquier herramienta que dé soporte a la modernización a este nivel..
- *Structured Metrics Metamodel (SMM)*, que es descrito por el OMG en [64], y que define una serie de metamodelos que dan soporte a la representación de métricas necesarias en el proceso de modernización.
- *Semantic for Business Vocabulary and Rules (SBVR)*: si bien la iniciativa ADM no define ningún metamodelo específico para representar la capa de negocio, sí que establece determinados criterios para obtener un mapeo entre los modelos representados en KDM y otro de los estándares

definidos por el OMG, el SBVR, definido en [63]. Su objetivo permite definir la semántica y las reglas de negocio asociadas a la estrategia de la empresa, y permite representar modelos semánticos basados en el vocabulario y las reglas del negocio.

2.1.2 *Gap analysis*

El gap analysis es un concepto que hace referencia a la comparación entre nuevas funcionalidades, servicios o requisitos de negocio que debe cumplir una solución objetivo, y las capacidades de la solución actual, dentro de los sistemas de información.

El gap analysis incluye además técnicas, algoritmos y operadores, que permiten realizar el mapeo entre ambas soluciones (actual y objetivo) de cara a facilitar la transformación de uno a otro, aprovechando al máximo posible el trabajo existente. Este proceso busca, por lo tanto, descubrir similitudes y diferencias para formalizar los pasos a dar, de cara a cumplir la nueva estrategia de la organización.

Las entradas que recibe este proceso serán, según lo especificado, los artefactos de la solución actual y objetivo, a nivel de negocio. Como resultado se obtendrá una serie de especificaciones, a nivel de procesos, reglas o servicios de negocio, que se deberán aplicar incrementalmente a la solución actual para conseguir la solución objetivo.

Los elementos de los que parte el gap analysis son los siguientes:

- **Modelo de negocio:** es el modelo del que se parte para realizar el análisis, ya sea un modelo de procesos (por ejemplo BPM), un modelo de clases o un modelo de reglas de negocio (por ejemplo SBVR).
- **Representación formal:** el modelo de negocio se representa en un formato que permita la aplicación de las técnicas de gap analysis. Se usan clasificadores para representar los elementos y las asociaciones entre ellos
- **Operadores:** los operadores permitirán identificar las similitudes y diferencias entre los modelos de negocio. Además la aplicación de los mismos, permitirá realizar el mapeo entre los modelos, para aplicarlos de forma iterativa hasta conseguir la solución objetivo.

2.2 Propuestas de modernización dirigida por modelos

En la bibliografía revisada se pueden encontrar muchas referencias a los procesos de modernización de sistemas (y más concretamente de sistemas heredados o *legacy system*). Sin embargo, aquellos que tratan este proceso de forma conjunta con el paradigma SOC, o tienen en cuenta enfoques dirigidos por modelos son significativamente menos numerosos. De entre todos ellos destacan los trabajos que se apoyan en la iniciativa ADM (Architecture Driven Modernization) del OMG como estándar de facto en este tipo de procesos. Veamos a continuación los trabajos que, entre todos los revisados, consideramos los más importantes en relación a la propuesta planteada en esta Tesis.

2.2.1 Criterios de evaluación

Los criterios a valorar para cada uno de los trabajos serán:

- **Tipo de enfoque:** Para cada trabajo se evaluará el enfoque que los autores le han dado. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Teórico (T): el trabajo es abordado desde un punto de vista eminentemente teórico, sin ofrecer ejemplos de casos reales.
 - Páctico (P): el trabajo es abordado desde un punto de vista práctico. Para que un trabajo pueda ser considerado como un componente práctico, debe contemplar, al menos, un caso que valide la aplicación del mismo a un ejemplo real.
- **Aproximación:** La aproximación de un trabajo determina si el mismo contempla elementos o mecanismos formales. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Formal (F): contempla algún mecanismo formal para abordar la solución, como por ejemplo la definición de reglas de transformación mediante lenguajes formales, etc.
 - No formal (NF): no contempla ningún mecanismo formal para abordar la solución.
- **Foco de Negocio:** determina el punto de vista que el trabajo da a los modelos de negocio a la hora de abordar la modernización. Los posibles valores que daremos a este criterio para cada trabajo serán determinar si

el trabajo contempla los modelos de negocio desde un punto de vista de reglas (estático) o de procesos (dinámico). Los posibles valores son:

- Procesos (P): contempla los modelos desde un punto de vista orientado a procesos (dinámico)
 - Reglas (R): contempla los modelos desde un punto de vista de reglas de negocio (estático)
 - Ninguno de los dos (N): no contempla modelos de negocio
- **Soportado por herramientas:** Determina si el trabajo viene soportado por alguna herramienta. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Sí (S): está soportado por alguna herramienta
 - No (N): no contempla ninguna herramienta
 - **Soportado por metodología:** determina si el trabajo contempla algún tipo de metodología para soportar el proceso de modernización. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Sí (S): Contempla un enfoque metodológico para abordar la solución
 - No (N): No contempla ningún enfoque metodológico
 - **Ciclo completo:** Determina si el trabajo se aborda desde un punto de vista completo. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Sí (S): el trabajo contempla una aproximación que cubre todo el ciclo de vida de modernización, entendido como un proceso que contempla los modelos físicos y de negocio del *as-is* y del *to-be*
 - No (N): no contempla un ciclo de vida completo para la modernización

La Tabla 1 recoge todas las características y los posibles valores que se utilizarán para evaluar las iniciativas más relevantes dentro de este apartado.

Característica	Descripción de los valores	Acrónimos correspondientes
Tipo de enfoque	Teórico, Práctico	T / P
Aproximación	Formal, No formal	F / NF
Foco de negocio	Orientado a procesos, Orientado a reglas, Otro	P / R / N
Soportado por metodología	Sí, No	S / N
Soportado por herramientas	Sí, No	S / N
Ciclo completo	Sí, No	S / N

Tabla 1. Criterios de evaluación para las propuestas de modernización

2.2.2 Ulrich y Newcomb

Tomando como base la iniciativa ADM, Ulrich y Newcomb proponen en [71] la utilización del proceso de modernización conocido como “en herradura”. Este tipo de transformación, conocida así por la forma que siguen los pasos que se van dando para llegar a los modelos *to-be* partiendo de los modelos *as-is*, toma como base el código fuente para obtener (o lo que es lo mismo, descubrir) un modelo físico, correspondiente a los elementos más cercanos a esta capa. De esta forma, la representación a este nivel correspondería al código fuente de la aplicación para el sistema actual (*as-is*) y los módulos implicados básicamente.

Partiendo de este modelo, se obtendrá un modelo lógico, que correspondería a la arquitectura tecnológica del sistema actual (aplicaciones + datos), y que en el trabajo se representa mediante el modelo KDM.

Utilizando los modelos obtenidos, el siguiente paso consistiría en utilizar el modelado de nivel físico (ASTM), en combinación con el de nivel lógico (KDM), para refinar aquellos aspectos del modelo inicial más cercanos al nivel de programación, sobre todo si la modernización del sistema implica un cambio de plataforma o lenguaje.

A continuación, se utilizarán las reglas de transformación necesarias para obtener un modelo más cercano al negocio, partiendo del modelado de nivel lógico, que será el nexo entre las soluciones actual y objetivo. Este paso entre las capas de negocio del *as-is* y *to-be*, que Ulrich y Newcomb proponen representar mediante los metamodelos SBVR, no es trivial, ya que no existen reglas de

transformación bien definidas para hacer un mapeo automático entre ambos modelos.

En su trabajo, Ulrich y Newcomb proponen solventar este “salto” mediante la definición de distintos escenarios de modernización, que han sido definidos mediante la experiencia de su aplicación en diversos casos de estudio de diferentes ámbitos tecnológicos y funcionales.

Una vez definido el nuevo modelo de negocio, que los autores proponen también modelar siguiendo el metamodelo SBVR utilizado para representar el homólogo del *as-is*, los autores definen una serie de reglas de transformación. Dichas reglas se aplican en sentido “descendente”, recorriendo el camino de forma inversa en la solución objetivo, partiendo de las nuevas reglas de negocio definidas en las capas superiores, hasta llegar a un modelo que sea fácilmente convertible en el código final para la solución del sistema evolucionado.

Esta propuesta, sin embargo, carece de un marco metodológico adecuado, al definir diversas técnicas en base al dominio del sistema, pero sin entrar en detalle en la normalización del ámbito de aplicación. Para cada uno de los escenarios se han propuesto reglas concretas, pero no se describen tareas que permitan obtener un denominador común de cara a automatizar o estandarizar el proceso de modernización, ni tampoco define herramientas para soportar este proceso.

2.2.3 SEASAT

A diferencia del trabajo de Ulrich, Pu *et al.* proponen en [56] un marco no metodológico, basado en una herramienta que denominaron SEASAT (Software Evolution for domain-Specific legacy sysTem). Este trabajo se centra sin embargo, en la definición de una serie de reglas para obtener y actualizar los modelos de forma automática, dentro del contexto de la ingeniería web, para lo cual se propone el uso de UML como notación para representar los distintos modelos obtenidos.

La herramienta propuesta propone partir del código fuente, más concretamente de las etiquetas HTML de las páginas web, para clasificarlas dentro de una serie de grupos:

- Descripción del texto: representan el formato del texto en el navegador y todos aquellos elementos visuales destinados a presentarlo, como por ejemplo `
`, ``, `<div>`, `<style>`, `<hr>`, etc.

- Descripciones de imágenes, ya sean de presentación en el navegador o de localización dentro de él (, <map>, etc.)
- Operaciones de linkado (<a>, <href>)
- Descripción de operaciones relacionadas con los frames de la página (<frame>, <frameset>, etc.)
- Descripción de las operaciones relacionadas con las tablas: <table>, <tr>, <td>, etc.

El siguiente paso de la propuesta es parsear el código de las dichas páginas para obtener de forma automática un modelo estático, denominado *Link Based Model* (LMB), que representa la navegación del sistema, a modo de mapa. A partir de este modelo estático, y tomando el resto de descriptores obtenidos en el paso anterior, la herramienta obtendrá de forma automática los diagramas UML que permiten establecer el modelo estático, el diagrama de clases, etc, que permitirán a los analistas abstraer el código fuente a un nivel de diseño entendible por éstos.

Sin embargo, una de las principales restricciones de esta propuesta, pese a utilizar modelos para abordar el proceso de modernización, es que restringe el mismo a capas de abstracción genéricas, es decir, no entra a descubrir el detalle de la lógica de negocio, ni por lo tanto puede cubrir un ciclo completo de modernización partiendo de la solución actual para obtener un solución objetivo.

Además, los modelos contemplados en esta propuesta se consideran de un nivel demasiado abstracto para poder aplicar reglas lo suficientemente formales, que permitan conseguir unos de otros.

En este mismo sentido, la herramienta definida no contempla una aproximación metodológica, ya que únicamente se centra en la obtención de modelos, no en los pasos necesarios para realizar este proceso de modernización. Del mismo modo, el enfoque es eminentemente teórico, al no describir la aplicación de este marco a un posible caso de estudio.

2.2.4 MARBLE

El trabajo de Perez-Castillo *et al.* [50] viene a solucionar algunos de los problemas vistos en las aproximaciones comentadas anteriormente. En este sentido propone un framework específico, centrado en el modelado del comportamiento del sistema. Este framework, al igual que el trabajo de Ulrich, se basa en la modernización en herradura para completar una transformación a los tres niveles de abstracción definidos por ADM.

KDM es el núcleo de MARBLE, ya que permite la representación y gestión del conocimiento extraída mediante las técnicas de ingeniería inversa. De esta forma, el sistema heredado puede gradualmente ser transformado desde el nivel físico hasta llegar a la definición de los procesos de negocio. Para ello MARBLE se divide en cuatro niveles de abstracción:

- Nivel 0 (N0): representa al sistema físico de origen en el mundo real, del cual se pretende descubrir sus procesos de negocio asociados.
- Nivel 1 (N1): agrupa un conjunto de modelos que representan las diferentes vistas o aspectos de la arquitectura del sistema origen. La transformación N0-a-N1 obtiene modelos PSM desde cada activo software heredado. Para ello se recurre a la extracción de información del sistema origen por medio de técnicas de ingeniería inversa clásicas como análisis estático/dinámico, slicing, etc. Los diferentes modelos PSM se construyen de acuerdo a metamodelos específicos: por ejemplo si el sistema origen es un sistema JAVA se utilizará un metamodelo de JAVA para crear un modelo de código, si la tecnología de bases de datos es la relacional se empleará un metamodelo de SQL para modelar el esquema de la base de datos y así con los demás activos software.
- Nivel 2 (N2): se centra en la representación integrada del sistema origen de acuerdo al metamodelo de KDM. Durante la transformación N1-a-N2 los modelos específicos se transforman hacia un único modelo PIM basado en KDM. Estas transformaciones están desarrolladas en QVT y reducen progresivamente la brecha conceptual entre el sistema origen y los procesos de negocio. El paso desde el LIS (N0) al modelo KDM (N2) no es directo ya que en muchos casos el conocimiento específico a la plataforma puede ser utilizado para inferir los procesos de negocio. Gracias al nivel intermedio N1 se reduce la pérdida semántica común a todo proceso de ingeniería inversa.
- Nivel 3 (N3): este nivel representa los procesos de negocio que son recuperados desde el modelo KDM. Este modelo equivale al CIM y representa un conjunto de diagramas de procesos de negocio. Para la representación de estos diagramas, MARBLE cuenta con un metamodelo ex profeso basado en BPMN. La transformación N2-a-N3 se establece también mediante transformaciones QVT, aunque en este caso están basadas en la detección de patrones de negocio en el

modelo KDM. Adicionalmente, un experto de negocio puede refinar los diagramas de procesos de negocio extraídos mediante las transformaciones.

Todo este marco se ha implementado en una herramienta que se propone desde un punto de vista formal, definiendo las reglas de transformación mediante QVT, y aplicado a un caso de estudio real. Sin embargo esta propuesta no define claramente cómo se solucionaría el problema del gap-analysis.

Así mismo, el nivel de negocio se representa con el modelo Business Process Model (BPM), en lugar de SBVR (metamodelo recomendado por OMG en la propuesta ADM, con objeto de disponer de un paso más directo para pasar del KDM al modelo de negocio).

2.2.5 Cánovas y García Molina

Cánovas y García Molina [18] también basan su trabajo en ADM, más concretamente en la utilización del metamodelo KDM como núcleo para la transformación del sistema.

En su propuesta se propone el uso del lenguaje Gra2MoL para extraer modelos conformes al metamodelo ASTM, también incluido en ADM, y en un segundo paso, estos modelos se transforman en modelos KDM de más alto nivel mediante transformaciones modelo a modelo en RubyTL.

Este proceso se engloba dentro de una herramienta y se ilustra con la aplicación a un caso práctico de extracción de modelos de una plataforma Java, en concreto del framework Struts, enmarcado dentro de un proyecto de migración de Struts a JSF.

El proceso se basa, como hemos comentado, en una cadena de dos transformaciones. En un primer paso se extraen los modelos ASTM a partir del código fuente, para lo que se aplica una transformación código a modelo implementada en Gra2MoL [19]. Este lenguaje permite definir transformaciones código-a-modelo especificando las relaciones (mappings) entre los elementos de la gramática y los elementos del metamodelo. En el segundo paso, los modelos ASTM obtenidos se transforman en modelos KDM mediante una transformación ASTM-KDM implementada en RubyTL [23].

La ventaja que añade la incorporación de modelos ASTM, apoyado en mecanismos formales, es la independencia de la plataforma y del lenguaje fuente,

aunque como añadido tiene la desventaja de que convierte el proceso de modernización en más complejo, sobre todo si únicamente se utiliza una plataforma de desarrollo concreta.

Otro de los aspectos no contemplados en la propuesta es la metodología, o modelos de negocio que permita realizar un ciclo completo de modernización.

2.2.6 El Proceso de modernización “YoYo”

Uno de los pocos trabajos revisados que contemplan aspectos metodológicos es el de Van der Heuvel [73], que define una metodología basada en combinar ingeniería hacia adelante e ingeniería inversa, basado también en modelos.

Estos pasos se basan en operaciones de ajuste (o matching) y adaptación entre las soluciones origen y objetivo.

En lo referente al matching, la propuesta se encarga de definir una serie de algoritmos que devuelven los mapeos entre elementos de ambas soluciones. El solape entre ambos se cuantifica mediante una métrica, que permite identificar los fragmentos reusables y los que habría que redefinir. Estos algoritmos son los siguientes:

- **Semantic Matching:** se encarga de identificar aquellos fragmentos semánticamente relevantes dentro de los nuevos procesos de negocio, comparando su significado lingüístico.
- **Structural Matching.:** se encarga de comparar los tipos de los interfaces, ya sean atómicos o compuestos. Este algoritmo tiene en cuenta también la estructura de la especificación, que se manifiesta mediante el nombre, los tipos de datos, las relaciones, etc.
- **Metamodel-based Matching:** se encarga de detectar correspondencias entre los metamodelos. Este algoritmo se basa en enlazar ambas especificaciones a un modelo unificado, para poder identificar mapeos potenciales y fragmentos reutilizables.
- **Aproximaciones IA:** se basa en el uso de redes neuronales para hacer el mapeo mediante patrones de reconocimiento.

La adaptación es necesaria una vez que el modelo de negocio ha sido mapeado parcialmente. En principio existen varias estrategias para implementar la adaptación, aunque van den Heuvel propone en su trabajo el uso de adaptadores,

que se colocan entre los modelos de negocio y sirven para solventar las incompatibilidades de interfaz entre los wrappers y los nuevos componentes de negocio.

Sin embargo esta propuesta, pese a estar implementada en una herramienta denominada YoYo, no ofrece una visión del ciclo completo de modernización. Incluso para el nivel de negocio, la propuesta trabaja a nivel de wrapper, lejos de definir servicios, procesos o reglas de negocio que permitan modelar esta capa.

2.2.7 SoSR

Otro de los trabajos que abordan la modernización desde un punto de vista metodológico es el de Chung y Byung [20], cuya propuesta, denominada SoSR, contempla una serie de mejores prácticas, enfocadas al paradigma orientado a servicios y dirigido por modelos.

Desde este punto de vista se propone el uso del modelo denominado 4+1, que se define en Kruchten [34], y que consiste en la definición de 4 vistas (diseño, implementación, proceso y despliegue, compartiendo la vista de casos de uso.

Este modelo es definido más ampliamente en terminos de tareas y roles para cada participante, usando la matriz RACI (Responsible, Accountable, Consulted, Keep Informed) [52].

La metodología SoSR consta de dos fases: ingeniería hacia adelante y reingeniería. Puesto que suponemos que el enfoque SOC no fue empleado para el desarrollo del sistema heredado, se debe aplicar una reingeniería. Se propone el uso de cinco matrices RACI para cada modelo 4 +1, en términos de la función principal del desarrollador de software que llevará a cabo varias sub-funciones tales como el análisis, diseño, implementación, prueba y despliegue, etc. Como resultado de la ingeniería inversa, el sistema anterior se modelará en un modelo visual, que se utiliza para el siguiente paso de ingeniería hacia adelante.

2.2.8 Baxter y Hendryx

El ultimo de los trabajos revisados, el de Baxter y Hendryx [9], usa diferentes herramientas y mecanismos formales para extraer la funcionalidad y las reglas de negocio directamente desde el código fuente. Esta estrategia es bastante útil en los procesos de reingeniería, pero sin embargo no propone un enfoque basado en modelos, por lo que este trabajo no será tenido en cuenta.

2.2.9 Comparativa de trabajos

La siguiente tabla muestra un resumen de los trabajos revisados aplicando los valores correspondientes:

	Tipo de Enfoque	Aproximación	Foco de negocio	Soportado por metodologías	Soportado por herramientas	Ciclo Completo
Ulrich y Newcomb	P	NF	R/P	N	N	S
SEASAT	T	NF	N	N	S	N
MARBLE	P	F	P	N	S	N
Cánovas/García Molina	P	F	N	N	S	N
YoYo	P	F	N	S	S	N
SoSR	T	N	N	S	N	N
PREMISA	P	N	R	S	S	S

Tabla 2. Evaluación de las propuestas de modernización

2.3 Propuestas de gap analysis

El gap-analysis focaliza el problema de cómo las aplicaciones existentes pueden ser trazadas a nuevos entornos SOA, yendo un paso más de las simples opciones de mapeo mediante wrappers. Para conseguir este objetivo es necesario empezar capturando los recursos y el portfolio de las aplicaciones actuales, incluyendo los sistemas heredados. A partir de ahí, los nuevos servicios de negocio necesitarán ser provistos dentro de un marco SOA. Esta comparación entre las nuevas funcionalidades o servicios (*to-be*) y las capacidades actuales (*as-is*) es lo que se denomina gap analysis [30].

As-is y *to-be* son elementos dentro de los modelos de negocio. El primero puede ser identificado teniendo en cuenta los sistemas heredados existentes, las bases de datos, etc.. El segundo puede ser el resultado de la aplicación de técnicas como la aplicación de algoritmos, la ejecución de simuladores, etc. El término gap-analysis ha sido utilizado en varios dominios, por lo tanto, surgen varios tipos de enfoques con objeto de ayudar a las organizaciones a obtener una mejor

comprensión de cómo se puede mejorar con el fin de aumentar la eficacia y la eficiencia.

En este apartado se comentarán aquellos trabajos relacionados con el gap-analysis, que tengan relación con el campo de la modernización de sistemas y con el objeto de la propuesta de la presente Tesis.

2.3.1 Criterios de evaluación

Los criterios a valorar para cada uno de los trabajos serán:

- **Tipo de enfoque:** Para cada trabajo se evaluará el enfoque que los autores le han dado. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Teórico (T): el trabajo es abordado desde un punto de vista eminentemente teórico, sin ofrecer ejemplos de casos reales.
 - Páctico (P): el trabajo es abordado desde un punto de vista práctico. Para que un trabajo pueda ser considerado práctico, debe contemplar, al menos un caso que valide la aplicación del mismo a un ejemplo real.
- **Soportado por herramientas:** Determina si el trabajo viene soportado por alguna herramienta. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Sí (S): está soportado por alguna herramienta
 - No (N): no contempla ninguna herramienta
- **Modelo de representación del negocio:** determina el modelo de representación que se sigue a la hora de abordar el gap-analysis. Los posibles valores que daremos a este criterio para cada trabajo serán determina si el trabajo contempla los modelos de negocio desde un punto de vista de reglas (estático) o de procesos (dinámico). Los posibles valores son:
 - Modelo orientado a Procesos (P): contempla los modelos desde un punto de vista orientado a procesos (dinámico)
 - Modelo orientado a Reglas (R): contempla los modelos desde un punto de vista de reglas de negocio (estático)
 - Ninguno de los dos (N): no contempla modelos de negocio

- **Orientación a modelos:** se valorará si los trabajos revisados presentan o no enfoques dirigidos por modelos. Los posibles valores que daremos a este criterio para cada trabajo serán:
 - Sí (S): el trabajo sigue un enfoque de orientación a modelos, entendido como la definición de los metamodelos y las reglas de transformación correspondientes
 - No (N): no contempla un enfoque dirigido por modelos

La Tabla 3 recoge todas las características y los posibles valores que se utilizarán para evaluar las iniciativas más relevantes dentro de este apartado.

Característica	Descripción de los valores	Acrónimos correspondientes
Tipo de enfoque	Teórico, Práctico	T / P
Orientación a modelos	Sí, No	S / N
Soportado por herramientas	Sí, No	S / N
Modelo de negocio	Orientado a procesos, Orientado a reglas, Otro	P / R / N

Tabla 3. Criterios de evaluación para las propuestas de gap analysis

2.3.2 *Yang y Papazoglou*

El trabajo de Yang y Papazoglou [79] destaca el concepto de componente web como un paquete que aúna servicios elementales o complejos, junto con sus interfaces y operaciones, de una forma consistente y uniforme en una definición de clase.

Los componentes web se sintetizan internamente en reutilizados, especializados o extendidos, y externamente se pueden publicar como servicios web que pueden ser consumidos por cualquier aplicación en un entorno SOA.

Los componentes web son, por tanto, un mecanismo para empaquetar el desarrollo de servicios en un entorno distribuido, utilizando un lenguaje de definición estándar, como puede ser WSDL.

Normalmente, los servicios compuestos son hardcodeados en la lógica de negocio de las aplicaciones, lo que suele crear problemas de mantenimiento. La propuesta revisada propone una serie de constructores para utilizar como base en estas composiciones:

- Composición de servicio secuencial: los servicios son invocados de forma sucesiva. La ejecución de un servicio es dependiente del anterior.
- Composición secuencial alternativa: indica que puede haber alternativas en la ejecución de los servicios, que pueden ser ordenados siguiendo determinados criterios de coste, tiempo, etc.
- Composición paralela de servicios: en este caso, todos los servicios compuestos se pueden ejecutar de forma independiente. Puede ser de dos formas:
 - (a) sincronización paralela con resultado: esta situación se da cuando los servicios pueden ejecutarse de forma concurrente, pero su ejecución necesita ser combinada.
 - (b) composición paralela alternativa: esta situación se da cuando los servicios compiten por su ejecución

Un servicio se puede considerar como un triplete $\langle C, A, P \rangle$ correspondiente a contenidos, actividades y propiedades respectivamente. Los contenidos se refieren al contexto del servicio, las actividades son el conjunto de operaciones que el servicio proporciona y las propiedades se refieren a la información que permite su configuración.

Por lo tanto, el framework propuesto en este trabajo abstrae estas formas de composición y sus características principales. Basado en este framework, se propone un enfoque para las distintas fases dentro del ciclo de vida del servicio (planificación, definición, implementación y ejecución).

Este framework permite así mismo identificar posibles candidatos de cara a reutilizar los servicios al comparar los componentes web de distintas soluciones.

Sin embargo esta propuesta no permite la definición de los modelos de negocio que permitan identificar posibles soluciones para automatizar el salto entre el *as-is* y el *to-be*.

2.3.3 Mantell

BPEL es un lenguaje que provee una notación basada en XML, y permite especificar la semántica para el comportamiento de los procesos de negocio basados en servicios web. BPER4WS (Business Process Execution Language for Web Services) se puede considerar una implementación concreta de este estándar

En esta línea, Mantell propone en [38] una propuesta para la utilización del perfil UML para la automatización de procesos de negocio [72], que permite a los procesos BPEL4WS ser modelados gráficamente mediante herramientas basadas en notación UML. Este perfil se basa en el mapeo entre conceptos de BPEL y perfiles UML, como podemos ver a continuación:

Profile Construct	BPEL4WS Concept
<<process>> class	BPEL process definition
Activity graph on a <<process>> class	BPEL activity hierarchy
<<process>> class attributes	BPEL variables
Hierarchical structure and control flow	BPEL sequence and flow activities
<<receive>>, <<reply>>, <<invoke>> activities	BPEL activities

Siguiendo este mapeo, se ha construido un ejemplo sobre un caso de estudio práctico, correspondiente a un escenario en el cual se ha utilizado Rational XDE mediante un motor BPEL4WS, para desarrollar un plugin en Eclipse, que utiliza XMI como formato de intercambio para los modelos UML. Dentro de este caso de estudio se han contemplado las siguientes fases:

1. Construir y exportar el modelo UML a la notación XMI (usando Rose o XDE)
2. Generar los ficheros BPEL, WSDL, y XSD
3. Desarrollar y ejecutar en el motor BPWS4J runtime

2.3.4 Papazoglou y van der Heuvel

El trabajo de Papazoglou y van den Heuvel incluye en [48] una dimension metodológica, al contemplar la definición de un ciclo de vida compuesto por los siguientes pasos: planificar, analizar y diseñar, realizar, ejecutar y desplegar.

La fase de planificación es una tarea preliminar que sirve para organizar las siguientes fases dentro de la metodología. El análisis y diseño especifica los procesos de negocio paso a paso, para que la siguiente fase de realización proceda a transformar las especificaciones de análisis y diseño en la implementación. La fase de realización busca desarrollar los servicios y los procesos para publicar los

interfaces en un repositorio. La fase de ejecución soporta la invocación de los servicios desarrollados.

Uno de los pasos más importantes en este marco metodológico corresponde a la fase de diseño, en la cual hay que buscar la solución que minimice el grado de acoplamiento. Se proponen tres formas de conseguirlo:

- **Acoplamiento de representación:** Los procesos de negocio no deben depender de representación específica o los detalles de implementación. Esto quiere decir que, no es necesario conocer el lenguaje de scripting que se utiliza para componer sus servicios subyacentes. El acoplamiento de representación es útil para dar soporte a servicios existentes que pueden ser intercambiados con un nuevo servicio, implementaciones proporcionadas por un proveedor diferente que ofrece un mejor rendimiento sin interrumpir la funcionalidad de procesos de negocios en general o versiones de servicios múltiples, correspondientes a diferentes versiones de un servicio que pueden funcionar mejor en algunas partes de un negocio
- **Acoplamiento de identidad:** Los canales de conexión entre los servicios deben ser conscientes de que está proporcionando el servicio. No es conveniente hacer un seguimiento de los objetivos (beneficiarios) de mensajes de servicio, especialmente cuando es probable que cambie
- **Acoplamiento de comunicación:** El número de mensajes intercambiados entre un emisor y un destinatario con el fin de lograr un cierto objetivo debe ser mínimo, teniendo en cuenta la comunicación modelo, por ejemplo, de un solo sentido, de solicitud / respuesta, y solicitar respuesta.

Otro aspecto relacionado es la cohesión, entendida por el grado de la fuerza de la relación funcional entre las operaciones en un proceso de servicio o negocio. El agregador de servicios debe esforzarse para ofrecer cohesión (autónomo) en los procesos cuyos servicios y operaciones de servicio están fuertemente relacionados entre sí.

Las pautas mediante las cuales se pretende aumentar la cohesión son las siguientes: proceso de servicio o negocio. El agregador de servicios debe esforzarse para conseguir:

- **Cohesión funcional:** Un proceso de negocio funcionalmente cohesivo debe realizar uno y sólo uno de los servicios relacionados.

- Cohesión de comunicación: Un proceso de negocio cohesivo desde el punto de vista de la comunicación es aquel cuyas actividades y servicios utilizan el mismo conjunto de mensajes de entrada y de salida
- Cohesión lógica: Una cohesión lógica de procesos de negocio es aquella que ofrece una independencia lógica, pero de funciones similares

El paso fundamental de esta propuesta, y por el cual este trabajo ha sido tenido en cuenta en el estado del arte, es la inclusión del proceso de gap-analysis como un paso clave dentro del marco metodológico.

Esta fase comienza con la comparación entre la implementación de los servicios disponibles y los planificados en la primera etapa, para identificar aquellos que pueden ser ensamblados dentro de la construcción de nuevos procesos.

La metodología distingue cuatro tipos de servicios:

- Sistemas heredados
- COTS
- Paquetes ERP
- Servicios web existentes

El gap analysis busca el ajuste entre las descripciones de alto nivel de los servicios a incluir contra los recursos en el contexto de web-service y se entrelaza con un análisis de escenarios, que considera los costes, riesgos, beneficios y retorno de la inversión de las siguientes opciones para el desarrollo de nuevos procesos de negocio:

- Enfoque “Green” del desarrollo: sigue un modelo de desarrollo clásico que cubre las especificaciones secuenciales para el desarrollo. Con esta opción se asume que los procesos de negocio tienen que ser desarrollados desde cero sin ningún tipo de reutilización de los trozos existentes en el proceso o especificaciones de servicio
- El desarrollo top-down: por lo general un modelo de casos de uso de negocio proporciona la especificación para los servicios empresariales. El proceso top-down descompone el dominio del negocio en sus áreas funcionales y subsistemas, incluyendo su flujo o la descomposición de procesos en los sub-procesos y casos de uso de negocio.

- Desarrollo bottom-up: parte de las aplicaciones empresariales existentes y los transforma a los nuevos servicios mediante wrappers.
- Desarrollo out-of-the-middle: combina el desarrollo top-down y bottom-up permitiendo los agregadores de servicios para seleccionar los fragmentos de recursos de la empresa que mejor satisfagan nuevos requerimientos del proceso de negocio. Los fragmentos son luego reforzados usando los servicios. Este enfoque puede ser útil en los casos donde los procesos de los sistemas heredados están ampliamente solapados con los requerimientos de servicios existentes

Cada una de estas opciones de desarrollo puede realizarse según las siguientes estrategias:

- (a) Reutilizar servicios o procesos de negocio existentes
- (b) Desarrollar nuevos servicios o procesos de negocio desde cero
- (c) Comprar o subcontratar los nuevos procesos de negocio
- (d) Modernizar componentes (COTS) o sistemas en servicios o procesos de negocio

2.3.5 *Armstrong y Underbakke*

Al igual que la anterior propuesta, el trabajo de Armstrong y Underbakke [5] no presenta un enfoque dirigido por modelos (únicamente se apoya en diagramas UML para soportar la transformación), si bien contempla una aproximación metodológica para abordar el gap analysis.

En concreto, las tareas a abordar dentro de esta propuesta tienen como objetivo comparar la funcionalidad actual con las necesidades de modernización (las nuevas entidades de negocio), a alto nivel, utilizando para ello modelos y diagramas UML. Estas tareas son básicamente las siguientes:

1. Seleccionar un paquete de entidades de negocio del *as-is*
2. Seleccionar una o varias entidades de negocio dentro del paquete
3. Revisar qué representa la entidad de negocio
4. Examinar su descripción (documentación)
5. Examinar los diagramas de clases en los que la entidad aparece
6. Examinar los diagramas de colaboración en los que aparecen los objetos de las entidades de negocio

7. Examinar los mensajes en los que aparecen los objetos de dicha entidad
8. Mapear la entidad de negocio *as-is* con la correspondiente *to-be* en base a la información obtenida.

Para soportar este proceso, este trabajo propone el uso de una herramienta que permite facilitar la trazabilidad entre las entidades (ver Figura 7).

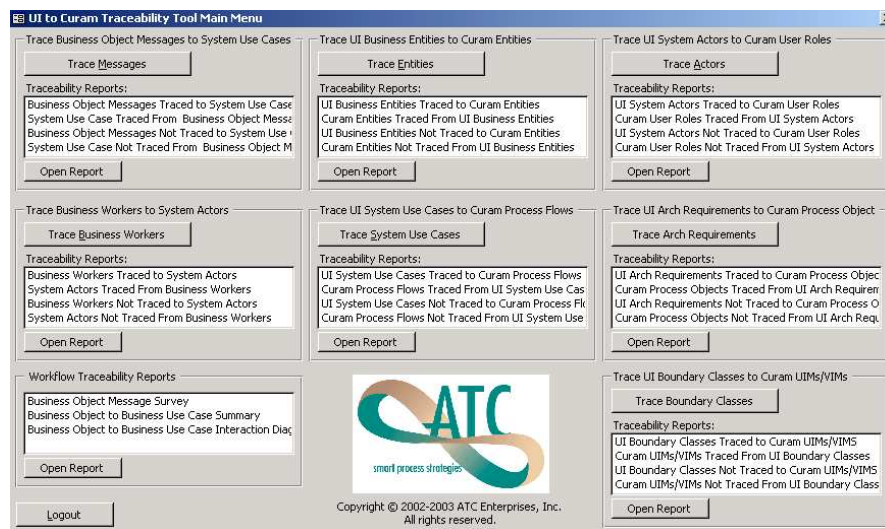


Figura 7. Interfaz web en la propuesta de Armstrong y Underbake

Como conclusión de este trabajo, destacar que no serviría para proyectos no documentados, al no realizar un descubrimiento de los servicios o los procesos partiendo del código, sino de la documentación existente del proyecto.

2.3.6 GAMBUSE

Uno de los pocos trabajos que se plantea en detalle el gap-analysis desde el punto de vista de los modelos es la orientación de De Castro *et al.* [25], que define una herramienta llamada GAMBUSE, que plantea el uso de una notación específica para representar modelos, cuyo objetivo es la utilización de una serie de operadores para pasar de un modelo a otro. Esta notación propuesta por GAMBUSE adopta un enfoque orientado a la gestión de modelos que implica la identificación y manipulación de asignaciones entre los modelos de negocio del

as-is y el *to-be*. GAMBUSE utiliza definiciones formales de los elementos de los modelos con el propósito de compararlos (ver Tabla 4).

Elemento	Descripción
<i>as-is</i> (M_a)	Representa el modelo <i>as-is</i>
<i>to-be</i> (M_w)	Representa el modelo <i>to-be</i>
Service Classifier (C)	Un element o colección de elementos del modelo
Signature Interface (σ)	Interfaz de servicio, método o procedimiento
Input message (i)	Parámetro de entrada de interfaz
Output message (o)	Parámetro de salida de un interfaz
Type (τ)	Tipo de un parámetro de entrada/salida

Tabla 4. Elementos en GAMBUSE

Una vez que los modelos *as-is* y *to-be* han sido formalmente definido, se hace una comparación entre ellos para identificar las correspondencias entre ambos; la identificación de nuevos elementos o diferencias entre ellos surgen también en esta comparación.

GAMBUSE define:

- tres operadores básicos de modelos para la detección de correspondencias modelo y la definición de los mapeos (la intersección, la unificación y la disparidad)
- dos operadores que se aplican con el fin de pasar de la especificación inicial a la nueva, como resultado de la adaptación de los procesos de negocio (la resta y la extracción).

Los operadores contemplados en GAMBUSE se muestran en la Tabla 5.

GAMBUSE propone el uso de una serie de algoritmos para medir la distancia entre los distintos elementos representados en el modelo, con el objetivo de determinar como de parecido es un elemento del modelo de negocio *as-is* con su correspondiente en el *to-be*.

La aplicación de estos operadores a los modelos de negocio representados en el *as-is* y el *to-be* está automatizada en una herramienta, que en el trabajo es acompañada de un caso de estudio en el que se ha aplicado de forma exitosa.

GAMBUSE, en cambio, no representa modelos de negocio basados en procesos o reglas, sino más bien define los mecanismos para pasar de un modelo genérico representado en el *as-is* a otro modelo de negocio objetivo (*to-be*) de una manera formal.

Operador	Descripción
Intersection ($c_\alpha \approx c_\omega$)	Devuelve pares de clasificadores equivalentes en los modelos comparados
Unification ($c_\alpha c_\omega$)	Devuelve un Nuevo clasificador compuesto por los clasificadores pasados como parámetro
Disparity (M_α, M_ω)	Devuelve los clasificadores c_ω del modelo <i>to-be</i> , que no existen en M_α
Subtraction (M_ω, c_ω)	Devuelve el modelo M_ω sin los clasificadores c_ω
Extraction (M_ω, c_ω)	Devuelve los clasificadores c_ω del modelo M_ω

Tabla 5. Operadores GAMBUSE

2.3.7 ARIS PPM

Aunque no existen demasiadas herramientas que aborden desde un punto de vista global el gap analysis, la utilización de ARIS Process Performance Manager (ARIS PPM) para cubrir este aspecto desde un enfoque de mejora de procesos ha sido propuesta en varios trabajos, entre los cuales destaca [77].

En concreto ARIS PPM es una herramienta que permite la detección automática de procesos extremo a extremo y el análisis de los históricos para encontrar posibles debilidades.

ARIS PPM genera automáticamente una representación gráfica de los flujos de trabajo empresariales. Para conseguirlo, los datos se extraen de la operativa de TI, con lo que es posible reconstruir cada transacción de principio a fin.

A diferencia de las representaciones en el modelado de procesos de otro tipo de herramientas, los modelos de proceso resultantes son visualizados no como deberían ser, sino como son en realidad.

La visualización del modelo descubierto es la base para un análisis estructural del proceso, ya que muestra claramente cuales son la senda y actividades más importantes en el proceso.

Además ofrece una variedad de capacidades de monitorización para descubrir las debilidades del proceso (como los largos periodos de procesamiento o los elevados costes) y las posibles causas del problema, que se pueden analizar mediante indicadores de procesos clave de rendimiento (KPI) que ofrece la herramienta

ARIS se basa en extractores, que se ubican en los orígenes de datos de las aplicaciones para hacer un análisis que permita extrapolar patrones de comportamiento. Los analistas de negocio deberán sondear dichos patrones para detectar posibles mejoras, para lo cual ARIS ofrece una serie de funcionalidades que permiten filtrar y operar con los mismos para detectar/proponer mejoras en los procesos.

Sin embargo ARIS, pese a representar gráficamente la información, no utiliza modelos entendidos como tales, es decir, no define metamodelos ni reglas para realizar las transformaciones.

2.3.8 El Proceso de modernización “YoYo”

El trabajo de Van der Heuvel [73], que ha sido descrito en la sección correspondiente al estudio de los trabajos relacionados con la modernización de sistemas, también ha sido tenido en cuenta en este apartado, al incluir un enfoque también dirigido al gap-analysis

Este trabajo que, como expusimos anteriormente, define una metodología basada en modelos, utilizar operaciones de ajuste (o matching) y adaptación entre las soluciones origen y objetivo para pasar de una a otra de forma más o menos automática.

En lo referente al gap-analysis, la propuesta presenta una serie de algoritmos que se encargan de obtener los mapeos entre elementos de ambas soluciones. En este sentido, la propuesta va muy en línea con el trabajo de De Castro *et al.* [25]. Van den Heuvel ofrece, al igual que GAMBUSE, una serie de métricas para medir el grado de solapamiento entre ambos, que permite identificar los fragmentos reusables y los que habría que redefinir. Estos algoritmos, como se ha comentado anteriormente, son los siguientes:

- **Semantic Matching:** se encarga de identificar aquellos fragmentos semánticamente relevantes dentro de los nuevos procesos de negocio, comparando su significado lingüístico.
- **Structural Matching:** se encarga de comparar los tipos de los interfaces, ya sean atómicos o compuestos. Este algoritmo tiene en cuenta también la estructura de la especificación, que se manifiesta mediante el nombre, los tipos de datos, las relaciones, etc.
- **Metamodel-based Matching:** se encarga de detectar correspondencias entre los metamodelos. Este algoritmo se basa en enlazar ambas

especificaciones a un modelo unificado, para poder identificar mapeos potenciales y fragmentos reutilizables.

- Aproximaciones IA: se basa en el uso de redes neuronales para hacer el mapeo mediante patrones de reconocimiento.

Sin embargo, y a diferencia de GAMBUSE, el trabajo de Van den Heuvel se centra en la aplicación de estos algoritmos a los modelos para determinar los pasos que habría que dar para mapear o adaptar el modelo existente de cara a la solución objetivo, pero no define la aplicación de operadores concretos que habría que aplicar al modelo de negocio *as-is* para llegar al modelo de negocio *to-be*.

2.3.9 Comparativas de trabajos

La siguiente tabla muestra un resumen de los trabajos revisados aplicando los valores correspondientes:

	Tipo de Enfoque	Orientación a modelos	Nivel de Negocio	Soportado por herramientas
Yang y Papazoglou	T	S	N	N
Mantell	P	S	P	S
Papazoglou y van der Heuvel	T	N	N	N
Armstrong y Banderbakke	P	N	N	S
GAMBUSE	P	S	N	S
ARIS PPM	P	N	P	S
YoYo	P	S	N	S
PREMISA	P	S	R	S

Tabla 6. Evaluación de las propuestas de gap-analysis

2.4 Conclusiones del capítulo

Una de las principales conclusiones que podemos establecer del estudio de los trabajos realizados es que, pese a que la literatura contempla una variedad ingente de trabajos, tanto referente a modernización como al gap-analysis,

únicamente un reducido grupo de ellos pueden ser considerados de interés dentro del contexto de la presente Tesis.

En este sentido, los trabajos referentes a la modernización basada en modelos suelen contemplar un enfoque eminentemente teórico en su mayoría, o presentan ejemplos no reales que no permiten validar el enfoque de sus propuestas. Los que contemplan casos reales lo hacen desde un punto de vista metodológico, que no permite ver un ciclo completo de modernización dentro del contexto del objeto de la presente Tesis.

Si bien algunas propuestas proponen el uso de modelos basados en la definición de procesos (BPM), el modelo de negocio que se precisa en Red.es, debe ser basado en reglas. El motivo es doble: por un lado las reglas del negocio permiten definir de una forma más sencilla la estrategia de la empresa, y por otro lado permite un modelo estático más simple de entender por los analistas de negocio y de sistemas, que les permite compartir información y alinear la estrategia con la tecnología.

Los trabajos relacionados con el gap-analysis, además de ser mucho más escasos que los referentes a la modernización, suelen abordar el problema desde un punto de vista teórico o no contemplan modelos de negocio apropiados para obtener un resultado óptimo en conjunción con el proceso de modernización.

Como punto a destacar, comentar que no hemos encontrado ningún trabajo que contemple ambos pilares (modernización y gap analysis) de forma conjunta, según los criterios establecidos.

Conocidos todos estos trabajos, se pretende cubrir los huecos analizados, para proponer un proceso de modernización basado en modelos, que contemple el gap analysis, y que plantee un ciclo completo, que se pueda aplicar al caso de Red.es. Dicho proceso será descrito en el siguiente capítulo.

***Capítulo 3. PREMISA:
Proceso de modernización
dirigido por modelos***

En esta sección, se especificarán los elementos básicos que ha sido necesario combinar para conseguir definir PREMISA (Proceso para la Evolución y Modernización de SistemAs), de cara a cumplir el segundo de los subobjetivos marcados en el capítulo 1.

Esta propuesta se fundamenta en dos pilares básicos, la modernización dirigida por modelos y el gap analysis. En este capítulo se describirá el esquema global de la propuesta, aplicado a cada uno de ellos, detallando los artefactos utilizados en la misma, así como las etapas y transformaciones necesarias para su aplicación a un caso real.

3.1 Esquema global de la propuesta

Una vez estudiados los trabajos previos, y establecidas las bases de nuestra propuesta, veremos a continuación como combinar todo ello de forma óptima para definir la propuesta de modernización a aplicar en Red.es. La Figura 8 muestra el esquema global de PREMISA, cuyos elementos pasaremos a desgranar en las siguientes secciones.

PREMISA se basa en los metamodelos definidos por ADM para representar los modelos para cada uno de los niveles de abstracción. Así pues, se utilizará KDM para representar el nivel lógico y SBVR para el nivel de negocio (que será completado utilizando técnicas de tipo DAFO). A este nivel se utilizarán técnicas de gap analysis para mapear las soluciones actual y objetivo. La representación gráfica de estos modelos se realizará utilizando UML, y la notación XMI (*XML Interchange*) para el intercambio de información. En el caso de Red.es no se ha utilizado una representación a nivel físico (como puede ser ASTM), ya que la plataforma tecnológica que se utiliza para todas las aplicaciones es J2EE, y el IDE de desarrollo es Eclipse (lo cual permite el uso de herramientas como MoDISCO para el descubrimiento del modelo lógico). Todos estos artefactos serán descritos en detalle en la sección 3.2.

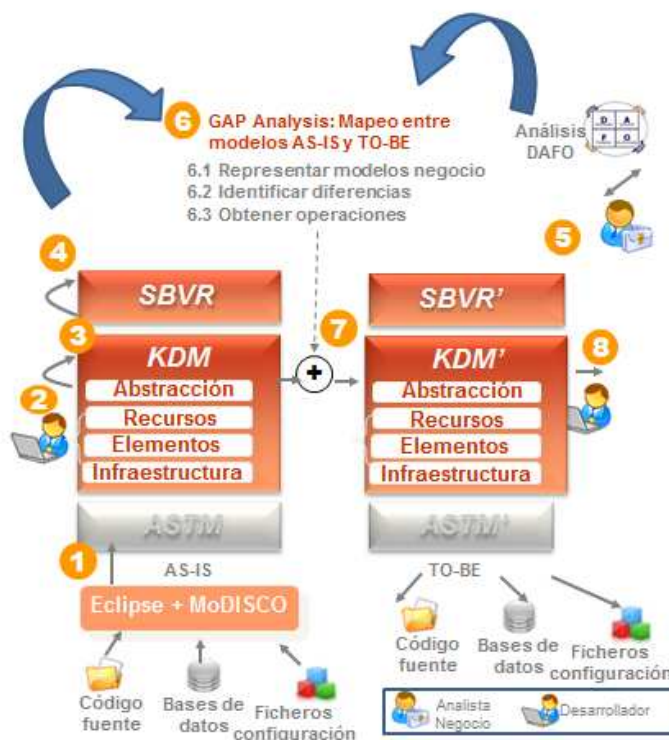


Figura 8. Esquema global de PREMISA

Una vez obtenidos los modelos, PREMISA se basa en la transformación denominada “en herradura” [68], para conseguir el sistema objetivo partiendo del sistema actual mediante la ejecución de una serie de pasos (y cuyo detalle se muestra en la sección 3.3):

1. El primer paso consiste en conseguir descubrir un modelo lógico, partiendo del código fuente, las bases de datos y los ficheros de configuración.
2. Se aplicarán las reglas necesarias para refinar este modelo,
3. y obtener el paquete de transformación, que permitirá obtener...
4. ... un modelo de negocio para el sistema actual.
5. A continuación, en base a este modelo, se diseñará el correspondiente al de la solución objetivo para, a continuación,
6. utilizar técnicas de gap analysis que permitan salvar el salto entre ambos.

7. La aplicación de estas diferencias al modelo KDM inicial, permitirá obtener un modelo KDM objetivo, que es el paso previo a...
8. ... la obtención del sistema final (8).

Las transformaciones necesarias para iterar entre estos pasos se realizarán mediante la ejecución de reglas que se aplicarán a los modelos obtenidos en cada fase. Todas ellas serán descritas en detalle en la sección 3.4.

La combinación de todo ello conforma la propuesta central de esta Tesis, que pretende dar respuesta a la hipótesis planteada en la sección 1.2.

3.2 Descripción de los artefactos

Describiremos en esta sección los distintos artefactos utilizados por PREMISA, los cuales se dividen en: metamodelos utilizados para representar los modelos, herramientas que dan soporte a la obtención de los mismos y técnicas a utilizar para depurar y mapear los modelos de negocio.

La Tabla 7 muestra un resumen de los artefactos en los que se fundamenta PREMISA, para cubrir cada uno de los aspectos del proceso de modernización comentado.

3.2.1 Descripción de los metamodelos

La representación de los modelos que utiliza PREMISA, se basa en los metamodelos propuestos por la iniciativa ADM. Veamos a continuación cada uno de ellos.

3.2.1.1 El metamodelo Knowledge Discovery Metamodel (KDM)

El metamodelo KDM [32] define la representación para cualquier tipo de elemento software, las relaciones entre ellos y los entornos en los que se ejecutan. El metamodelo de KDM se ha organizado en cuatro capas o niveles que a su vez están formadas por varios dominios o vistas arquitecturales del sistema. Cada dominio está formado por un único paquete y contiene las metaclases para los conceptos propios del aspecto considerado. Esta organización favorece la modularidad y separación de aspectos. Estas capas, de menor a mayor nivel de abstracción, son:

3. PREMISA: Proceso de modernización dirigido por modelos 67

- **Capa de infraestructura:** proporciona los elementos básicos para la construcción de los modelos KDM, como las entidades, relaciones entre elementos y trazabilidad.
- **Capa de elementos de programa:** permite representar los artefactos software del sistema a nivel de implementación.
- **Capa de recursos runtime:** representa elementos de mayor nivel de abstracción ya que están ligados a la plataforma de ejecución, como por ejemplo la interfaz de usuario.
- **Capa de abstracciones:** trabaja a un nivel de abstracción más cercano al negocio, representando el conocimiento específico del dominio, como las reglas de negocio o el conocimiento arquitectural.

Aspecto	Artefacto
Modernización (global)	ADM
Descubrimiento de modelos lógicos	MoDISCO
Marco de desarrollo	Eclipse
Modelo lógico	KDM
Modelo de negocio	SBVR
Análisis de negocio	DAFO
Mapeo de negocio	Gap-analysis
Representación gráfica de modelos	UML
Intercambio de modelos	XMI

Tabla 7. Artefactos básicos para PREMISA

El uso de KDM no requiere considerar todos los modelos sino aquellos que sean de interés para el usuario [68]. El anexo IV muestra un detalle completo de los metamodelos definidos para cada una de estas capas contempladas en KDM.

3.2.1.2 El metamodelo Semantic Business and Vocabulary Rules (SBVR)

La iniciativa ADM se centra en los metamodelos más cercanos a la tecnología (KDM y ASTM). No obstante, establece determinados criterios para obtener un mapeo entre los modelos representados en KDM y otro de los estándares definidos por el OMG, el *Semantic Business and Vocabulary Rules* (SBVR) [63].

SBVR tiene como objetivo definir la semántica y las reglas de negocio asociadas a la estrategia de la empresa, y permite desarrollar modelos semánticos basados en el vocabulario y las reglas del negocio.

3.2.2 Herramientas, frameworks y técnicas de soporte a la modernización

3.2.2.1 A nivel lógico

La arquitectura tecnológica de Red.es pasa por el uso de plataformas basadas en J2EE para contemplar el desarrollo de los sistemas. De este modo, Eclipse es el IDE utilizado para la construcción de las aplicaciones en Red.es.

Dentro de este entorno de desarrollo existe un componente denominado MoDISCO (*Model DISCO*vering) [43], que es propuesto por PREMISA como soporte a la obtención del modelo lógico (KDM) del sistema actual.

Este Framework de la iniciativa Eclipse, proporciona una serie de herramientas que facilitan el descubrimiento de los modelos lógicos. A partir del código fuente de un sistema basado en J2EE, MoDISCO devuelve automáticamente una estructura basada en XMI (*XML Model IN*terchange), que posteriormente puede ser visualizada como un diagrama UML a través del conversor que también ofrece este Framework.

3.2.2.2 A nivel de negocio

PREMISA propone el uso de técnicas DAFO [44] para refinar el modelo de negocio, de acuerdo a la estrategia de la empresa. Mediante estas técnicas se puede hacer un análisis de las debilidades y fortalezas de la organización, en combinación con las amenazas y oportunidades del contexto actual, para proponer una estrategia de negocio adecuada.

Mediante las técnicas de gap analysis se podrá entonces realizar un mapeo entre el modelo de negocio actual y el refinado, para aplicar las operaciones que permiten transformar uno en otro.

3.3 Etapas del proceso

Pasaremos a continuación a describir cada uno de los pasos de PREMISA, teniendo en cuenta los artefactos descritos anteriormente.

3.3.1 Paso 1: Obtención del modelo lógico (inicial)

El primer paso de nuestra propuesta consiste en representar el modelo lógico del sistema actual (*as-is*), utilizando la notación del metamodelo KDM. Como se ha comentado anteriormente, PREMISA propone la utilización de la herramienta MoDISCO [43] para la extracción semiautomática del modelo KDM.

Puesto que la plataforma utilizada en Red.es está basada en J2EE [11], la utilización del IDE de Eclipse permite integrar el uso de MoDISCO en un único interfaz.

De entre todos los trabajos de modernización revisados, no hemos encontrado ninguno que aplique este framework en el descubrimiento del modelo lógico de una forma completa.

La principal dificultad, por lo tanto, en la realización de este paso, consiste en cómo integrar MoDISCO dentro de PREMISA en un marco uniforme. En este sentido, el resultado del descubrimiento de este modelo viene representado en la notación XMI, utilizada como intercambio de información entre las distintas herramientas CASE de análisis y diseño.

Esto nos permite visualizar el modelo obtenido en un formato gráfico similar al de un diagrama de clases, utilizando visores y herramientas UML, para facilitar a los desarrolladores la comprensión del modelo. Del mismo modo, este formato permite disponer de una representación formal que permitirá aplicar las reglas de transformación para iterar hasta los niveles de abstracción superiores.

La Tabla 8 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Código fuente (proyecto en Eclipse)
Salida	Modelo KDM (en formato XMI)
Herramientas/técnicas	MoDISCO/Eclipse
Reglas de transformación	n/a (basado en MoDISCO)

Tabla 8. Resumen de artefactos del paso 1 de PREMISA

3.3.2 Paso 2: Refinamiento del modelo lógico

El principal objeto de esta fase es completar el modelo lógico obtenido en el paso anterior, debido a que MoDISCO sólo permite el descubrimiento automático para las capas más cercanas a la tecnología.

Como dificultad añadida, este descubrimiento tampoco es completo, ya que los paquetes de cada una de estas clases no son suficientes para obtener un nivel de abstracción necesario para pivotar hasta el modelo de negocio de forma intuitiva.

Yendo desde un nivel de abstracción superior (cercano al nivel de negocio) hacia niveles inferiores del modelo lógico, podemos determinar qué clases/paquetes son necesarios en el modelo de abstracción del modelo lógico, que permitan obtener el modelo de negocio. Del mismo modo, será necesario completar las clases/paquetes de la capa de recursos del modelo lógico que serán necesarias para obtener las clases de la capa de abstracción que a su vez nos permitirán obtener el modelo de negocio. Y por último, deberemos completar las clases de la capa de elementos de programa del modelo lógico no descubiertas por MoDISCO, que nos permita una trazabilidad completa entre todas las clases del modelo.

El anexo IV muestra la lista de todas las clases y paquetes, para cada una de las capas del modelo lógico KDM, indicando aquellas que son descubiertas automáticamente por MoDISCO.

Esto no quiere decir que haya que completar todas las clases no descubiertas por MoDISCO para obtener un modelo de negocio válido, sino que habrá que completar únicamente aquellas clases que permitan, dentro del caso de Red.es, realizar el proceso de forma completa.

Por ejemplo, si bien MoDISCO descubre de forma automática los elementos de tipo *ActionElement* (encargados, entre otras cosas, de representar los flujos condicionales del código fuente), las bifurcaciones o ramas condicionales

verdadero y falso (*TrueFlow* y *FalseFlow*) no son descubiertas de forma automática.

A su vez, las clases de tipo *ActionElement*, correspondientes a la capa de elementos de programa dentro de KDM, permitirá obtener las clases *TermUnit* y *FactUnit* correspondientes a la capa de abstracción, dentro del modelo lógico (ver reglas de transformación R10 y R12 respectivamente en el apartado 3.4.2). Igualmente las clases *TermUnit* y *FactUnit* permitirán obtener las clases *Fact* y *Keyword* del modelo SBVR (ver reglas de transformación en el apartado 3.4.3).

Para completar este modelo, el desarrollador deberá tener en cuenta el código fuente de la aplicación, aplicando las reglas de transformación necesarias.

La Tabla 9 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Modelo KDM + código fuente
Salida	Modelo KDM refinado
Herramientas/técnicas	n/a
Reglas de transformación	R1, R2, R3, R4

Tabla 9. Resumen de artefactos del paso 2 de PREMISA

3.3.3 Paso 3: Obtención de la capa de abstracción KDM

Mediante la aplicación de reglas de transformación al modelo lógico refinado, se obtendrá una representación de un mayor nivel de abstracción, cercana al nivel de negocio del modelo KDM (capa de abstracción). De esta forma se facilitará, en el siguiente paso, la aplicación de las reglas de transformación, para obtener el modelo de negocio del sistema existente.

La Tabla 10 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Modelo KDM refinado
Salida	Paquete de abstracción de KDM
Herramientas/técnicas	n/a
Reglas de transformación	R5, R6, R7, R8, R9, R10, R11, R12

Tabla 10. Resumen de artefactos del paso 3 de PREMISA

3.3.4 Paso 4: Obtención del modelo de negocio as-is

Si bien existe un grupo de trabajo para automatizar el proceso de transformación entre el nivel de abstracción de KDM y el modelo SBVR [63], en el momento de escribir esta Tesis no se dispone de ningún Framework que permita esta obtención de forma automática. Basándonos sin embargo en la capa de abstracción del modelo KDM, se proponen en el presente trabajo las reglas de transformación necesarias para obtener de forma automática el modelo SBVR, que permitirá derivar las reglas de negocio (*Business Rules*) inherentes al sistema. De esta forma podremos extrapolar el conocimiento del sistema de una forma más inteligible para los expertos de negocio y la Alta Dirección.

Este será por lo tanto, el punto de partida para establecer la estrategia de la modernización basada en las políticas y la visión de la Organización, que se encargará de dirigir el resto del proceso.

La Tabla 11 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Paquete de abstracción de KDM
Salida	Modelo SBVR del <i>as-is</i>
Herramientas/técnicas	n/a
Reglas de transformación	R13, R14, R15

Tabla 11. Resumen de artefactos del paso 4 de PREMISA

3.3.5 Paso 5: Obtención del modelo de negocio to-be

Una vez obtenido el modelo de negocio de sistema actual, PREMISA propone pasar esta representación a los analistas de negocio, para que lo completen o actualicen de acuerdo a la estrategia de la empresa.

Este no es un paso directo, sino que debe realizarse de forma manual, contando con analistas de negocio que conozcan esta representación (para lo cual puede ser necesario formar a los mismos en este tipo de modelado). El modelo obtenido en este paso es clave para obtener un sistema que satisfaga las necesidades de la organización.

Aunque existen diversas metodologías para analizar los puntos fuertes y débiles de la organización, y su posicionamiento en el sector según las oportunidades del mismo, las técnicas de tipo DAFO [44] son las que mejor se adecúan a las necesidades de Red.es.

Mediante este tipo de técnicas se identificarán las debilidades, amenazas, fortalezas y oportunidades a nivel de negocio, para establecer una estrategia que responda a las preguntas: ¿Es necesario cambiar la Tecnología actual? ¿Qué reglas de negocio deben añadirse, cambiarse o eliminarse? ¿Qué enfoque se va a dar a la modernización?, etc. Todo ello estableciendo una priorización y una planificación que permita a Red.es contar con una hoja de ruta clara, y añadir valor al proceso. Este valor puede venir dado como ROI (Retorno de Inversión), satisfacción del cliente/proveedor, posicionamiento en el mercado, etc.

Las nuevas reglas de negocio deberán ser consensuadas entre los analistas del negocio y la dirección de la empresa. Este proceso no es automático y de él dependerá el éxito del proyecto. Además, en un enfoque SOC, estas reglas serán las mejores candidatas para publicar los correspondientes servicios de valor añadido para la entidad.

La Tabla 12 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

3.3.6 Paso 6: Mapeo los modelos de negocio as-is y to-be

Para aplicar las técnicas de gap-analysis entre el modelo de negocio de la solución actual (*as-is*) y el correspondiente a la solución objetivo (*to-be*), PREMISA propone la división del paso en tres etapas, que se resumen en la Figura 9.

Entrada	Modelo SBVR <i>as-is</i> + nuevas reglas
Salida	Modelo SBVR <i>to-be</i>
Herramientas/técnicas	DAFO
Reglas de transformación	n/a

Tabla 12. Resumen de artefactos del paso 5 de PREMISA

3.3.6.1 Definir los modelos *as-is* y *to-be* utilizando mecanismos formales

Esta primera etapa busca representar los modelos de negocio de una manera formal. Esto permitirá en la siguiente etapa la aplicación de los operadores necesarios con objeto de establecer las similitudes y diferencias entre ambos modelos.



Figura 9. Etapas del gap analysis

Para ello, PREMISA propone una extensión a la notación propuesta en GAMBUSE [25], que permite definir las relaciones entre los clasificadores (ver Tabla 13), necesaria para la representación de modelos SBVR.

<i>Elemento</i>	<i>Descripción</i>
<i>as-is</i> (M_{α})	Representa el modelo <i>as-is</i>
<i>to-be</i> (M_{ω})	Representa el modelo <i>to-be</i>
Clasificador	Un elemento o colección de elementos de un modelo
Interfaz (σ)	Interfaz de un servicio, método o procedimiento
Mensaje de entrada (i)	Parámetro de entrada de un interfaz
Mensaje de salida (o)	Parámetro de salida de un interfaz
Tipo (τ)	Tipo de un parámetro de entrada o salida
Contains ($C_1 \subseteq C_2$)	Representa una asociación entre dos clasificadores C_1 y C_2 .
Inherits ($C_1 \downarrow C_2$)	Representa una herencia entre dos clasificadores C_1 y C_2 .
Compose ($C_1 \equiv C_2$)	Representa una composición entre dos clasificadores C_1 y C_2 .

Tabla 13. Notación para representar los modelos de negocio en PREMISA

Partiendo del modelo de negocio, PREMISA propone una serie de correspondencias y mapeos, que permiten representar formalmente sus elementos, de acuerdo a la Tabla 13 (ver Tabla 14).

3.3.6.2 Aplicar operadores para identificar similitudes y diferencias entre los modelos de negocio

Una vez representados de una manera formal los modelos de negocio de las soluciones actual y objetivo (*as-is* y *to-be* respectivamente), la siguiente etapa consiste en identificar las similitudes y diferencias entre ambos.

Para ello, PREMISA propone una extensión a los operadores planteados en GAMBUSE, que permitan el mapeo entre ambos modelos de negocio (ver Tabla 15).

Elemento en el SBVR	Elemento en notación formal
Term	Clasificador con parámetro de salida de tipo term
Condition(Y)	Clasificador con parámetro de entrada de tipo <i>condicion_Y</i>
Condition(O)	Clasificador con parámetro de entrada de tipo <i>condicion_O</i>
Condition(equals)	Clasificador con parámetro de entrada de tipo <i>condicion_equals</i>
Fact	Clasificador sin parámetros de entrada/salida
Keyword	Clasificador con parámetro de salida de tipo keyword
Relación de composición	Relación de composición entre los clasificadores correspondientes
Relación de herencia	Relación de herencia entre los clasificadores relacionados (candidato a ser servicio a exponer)
Servicio/método	Clasificador con parámetros de entrada/salida (i/o) de los tipos correspondientes, que normalmente va unido al operador de compatibilidad
Parámetros de entrada	Clasificador de tipo i
Parámetros de salida	Clasificador de tipo o

Tabla 14. Mapeo entre SBVR y notación formal para el gap-analysis

Operador	Descripción
Intersection ($c_{\alpha} \approx c_{\omega}$)	Devuelve pares de clasificadores equivalentes de los modelos comparados
Unification ($c_{\alpha} c_{\omega}$)	Devuelve un Nuevo clasificador compuesto por los clasificadores pasados como parámetro
Disparity (M_{α}, M_{ω})	Devuelve los clasificadores c_{ω} del modelo <i>to-be</i> que no existen en M_{α}
Subtraction (M_{ω}, c_{ω})	Devuelve el modelo M_{ω} sin el clasificador c_{ω}
Extraction (M_{ω}, c_{ω})	Devuelve el clasificador c_{ω} de M_{ω}
Insert (M_{ω}, c_{ω})	Inserta el clasificador c_{ω} en el modelo M_{ω}

Tabla 15. Operadores propuestos en PREMISA para el gap-analysis

El algoritmo para establecer este mapeo pasa por la ejecución de las siguientes actividades en orden secuencial:

1. Aplicar el operador *intersection* entre los modelos M_{α} y M_{ω} . Los clasificadores obtenidos como consecuencia de esta operación

corresponden a aquellos clasificadores en el modelo de negocio *as-is* que no sufrirán cambios en el modelo de negocio *to-be*.

2. Aplicar el operador *Disparity* entre los modelos $M\alpha$ y $M\bar{\omega}$. Los clasificadores obtenidos como consecuencia de esta operación representarán los clasificadores que aparecen en el modelo *as-is* pero no en el *to-be*.
3. Aplicar el operador *Disparity* entre los modelos $M\bar{\omega}$ y $M\alpha$. Los clasificadores obtenidos como consecuencia de esta operación representarán los clasificadores que aparecen en el modelo *to-be* pero no en el *as-is*.

Para determinar si un clasificador del modelo *as-is* es compatible con otro clasificador del modelo *to-be*, estos operadores utilizarán la comparación de sus nombres; es decir, si dos clasificadores se llaman igual en ambos modelos, se considerará que es el mismo, o compatible.

3.3.6.3 Obtener las operaciones a aplicar

La última etapa de este paso tiene como objetivo identificar las operaciones necesarias que deben ser aplicadas al modelo de negocio *as-is*, para obtener su homólogo *to-be*.

De acuerdo al algoritmo ejecutado en la etapa anterior, los resultados obtenidos marcarán los operadores que deben ser aplicados al modelo de negocio *as-is* para obtener el correspondiente modelo de negocio *to-be*:

1. Los clasificadores obtenidos al aplicar el operador *intersection* entre $M\alpha$ y $M\bar{\omega}$, determinan los clasificadores del *as-is* a los que no hay que aplicar ninguna regla de transformación para obtener el *to-be*.
2. Los clasificadores obtenidos al aplicar el operador *Disparity* entre los modelos $M\alpha$ y $M\bar{\omega}$, determinan los clasificadores del *as-is* que deben eliminarse (mediante operaciones del tipo *Subtraction*) para obtener el *to-be* correspondiente.
3. Los clasificadores obtenidos al aplicar el operador *Disparity* entre los modelos $M\bar{\omega}$ y $M\alpha$, determinan los clasificadores que habrá que añadir mediante operadores de tipo insert al *as-is* para obtener el *to-be* correspondiente

La Tabla 16 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Modelos SBVR <i>as-is</i> y <i>to-be</i>
Salida	Operaciones a ejecutar
Herramientas/técnicas	Gap-analysis
Reglas de transformación	n/a

Tabla 16. Resumen de artefactos del paso 6 de PREMISA

3.3.7 Paso 7: Obtención del KDM objetivo

Una vez definidas las nuevas reglas de negocio y determinada la estrategia a seguir, el siguiente paso consistirá en aplicar las reglas obtenidas en el gap-analysis al modelo KDM original, para obtener el correspondiente a la solución objetivo.

Siguiendo las reglas obtenidas en el paso anterior, se definirán las operaciones equivalentes para aplicar al nivel lógico del modelo.

La Tabla 17 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Modelo KDM + operaciones
Salida	KDM <i>to-be</i>
Herramientas/técnicas	n/a
Reglas de transformación	R16, R17, R18, R19, R20, R21

Tabla 17. Resumen de artefactos del paso 7 de PREMISA

3.3.8 Paso 8: Obtención de la solución objetivo (final)

Partiendo del modelo KDM de la solución objetivo, el último paso consiste en obtener el código fuente de la solución final, utilizando las mismas reglas de transformación que se utilizaron para obtener el modelo original, pero en sentido inverso.

La Tabla 18 resume las entradas, salidas, herramientas y reglas de transformación de este paso.

Entrada	Modelo KDM <i>to-be</i>
Salida	Código fuente
Herramientas/técnicas	Eclipse
Reglas de transformación	R22, R23, R24, R25

Tabla 18. Resumen de artefactos del paso 8 de PREMISA

3.4 Descripción de las transformaciones

En este apartado se describen las reglas para realizar las transformaciones necesarias entre los modelos contemplados en PREMISA. Dichas reglas se describirán siguiendo el lenguaje natural para simplificar el proceso, si bien se podrían representar mediante lenguajes formales en trabajos futuros.

Para describir estas reglas, utilizaremos un formato de ficha para cada una, cuya plantilla se muestra en la Figura 10.

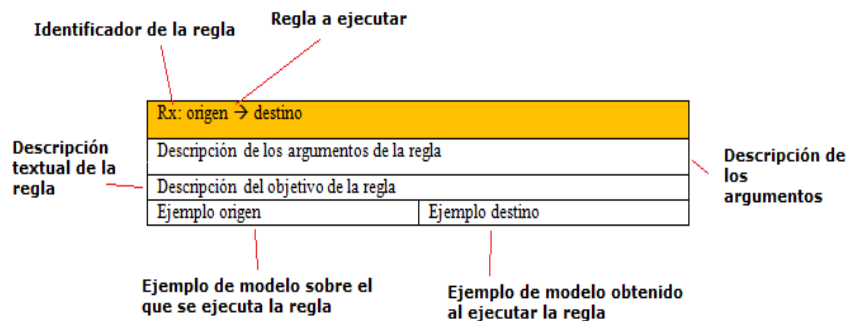


Figura 10. Formato de ficha para explicar las reglas de transformación

El esquema mostrado en la Figura 11 resume las reglas de transformación que se aplican a cada una de las fases en PREMISA.

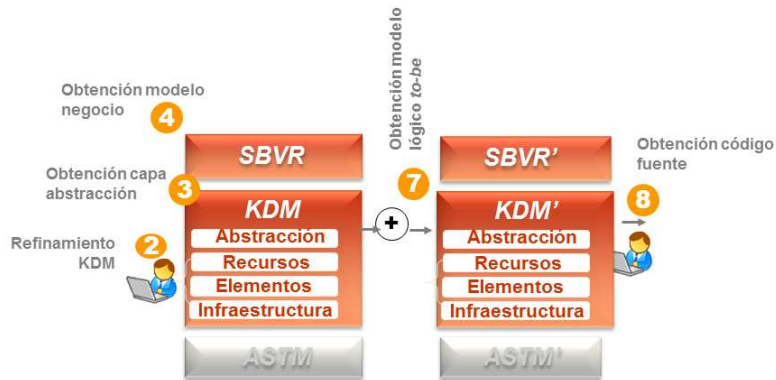


Figura 11. Resumen de reglas de transformación

Pasaremos a describir cada una de las reglas, agrupadas por las fases en las que se ejecuta cada una.

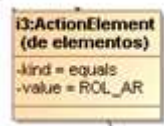
3.4.1 Refinamiento del KDM


En esta sección describiremos las reglas de transformación destinadas a completar las capas de recursos y elementos de programa del modelo lógico obtenido con MoDISCO. Estas reglas tomarán como origen el código fuente de la aplicación y como destino se obtendrá el KDM refinado.

Estas reglas se aplican en el paso 2 de PREMISA, y el principal rol implicado es el desarrollador, que deberá analizar el código fuente para aplicarlas y completar el modelo. Sólo serán necesarias cuatro reglas en este paso, ya que en el caso real de Red.es son no necesarias más reglas para refinar el modelo.

La principal aportación en este paso es la definición de las reglas de transformación que permiten completar el modelo lógico obtenido con MoDISCO.

3. PREMISA: Proceso de modernización dirigido por modelos 81

R1: condicional (expr1, tipo, expr2) → nuevo (ActionElement); kind = tipo; value = expr2	
<p>Expr1: expresión izquierda de la comparación Expr2: expresión derecha de la comparación Tipo: tipo de comparación (equals, minor, mayor, etc.)</p>	
<p>Un condicional en el código genera un elemento de tipo <i>ActionElement</i> (elemento que representa una acción) en el modelo KDM cuyos atributos son: <i>kind</i> (operador condicional utilizado) y <i>value</i> (expresión derecha a comparar).</p>	
<p>if (rol_actual == ROL_AR)</p>	

R2: Mensaje (M) → nuevo (ActionElement); value = M	
M: Mensaje por pantalla	
<p>Una instrucción de tipo “mensaje por pantalla” genera un <i>ActionElement</i> en el modelo cuyo atributo <i>value</i> corresponde al mensaje a mostrar.</p>	
<p>Msgbox ('error')</p>	

<p>R3: Rama_verdadera_if (bloque_instr) → ae2 = nuevo (ActionElement); value = bloque_instr → link (ae1; ae2); kind = true</p>	
<p>Bloque_instr: bloque de instrucciones que se ejecuta si se cumple la condición ae1: ActionElement del condicional origen</p>	
<p>La rama “verdadero” de un condicional genera una asociación entre el <i>ActionElement</i> en el modelo que lo origina y el <i>ActionElement</i> que se ejecuta si se cumple la condición lógica.</p>	
<pre> if (rol_actual == ROL_AR) dominio.titular = titular_actual; else msgbox ('error') </pre>	<pre> classDiagram class i1["i1:ActionElement (de elementos)"] { -kind = assign } class i3["i3:ActionElement (de elementos)"] { -kind = equals -value = ROL_AR } class i4["i4:ActionElement (de elementos)"] { -kind = message -value = error } i3 --> i1 : TrueFlow i3 --> i4 : FalseFlow </pre>

R4: Rama_falsa_if (bloque_instr) → ae2 = nuevo (ActionElement); value = bloque_instr → link (ae1; ae2); kind = false	
Bloque_instr: bloque de instrucciones que se ejecuta si no se cumple la condición ae1: ActionElement del condicional origen	
La rama “verdadero” de un condicional genera una asociación entre el <i>ActionElement</i> en el modelo que lo origina y el <i>ActionElement</i> que se ejecuta si no se cumple la condición lógica.	
<pre> if (rol_actual == ROL_AR) dominio.titular = titular_actual; else msgbox ('error') </pre>	

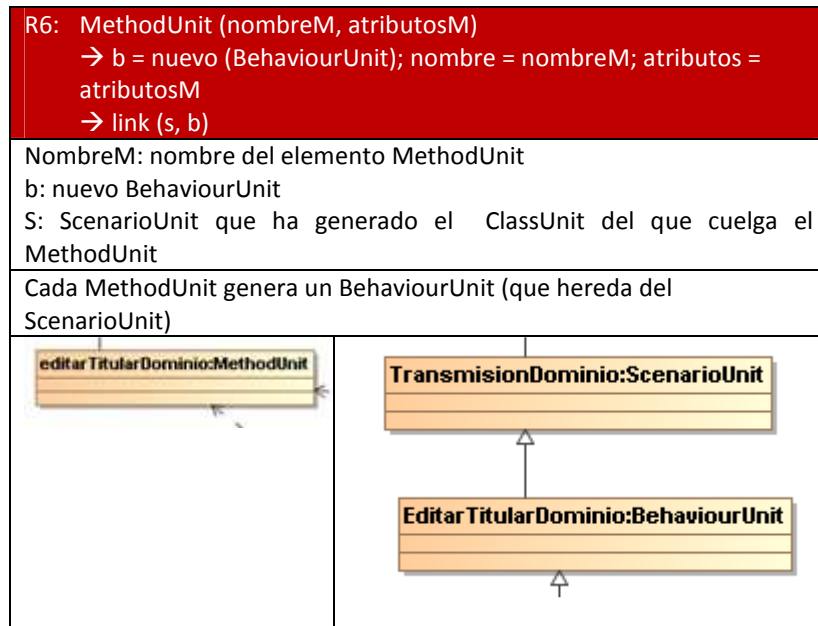
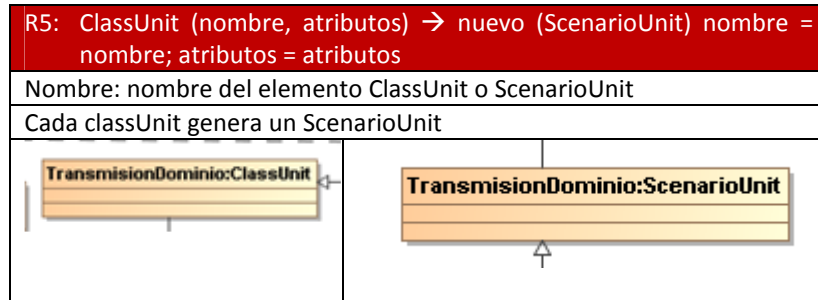
3.4.2 Obtención de la capa de abstracción

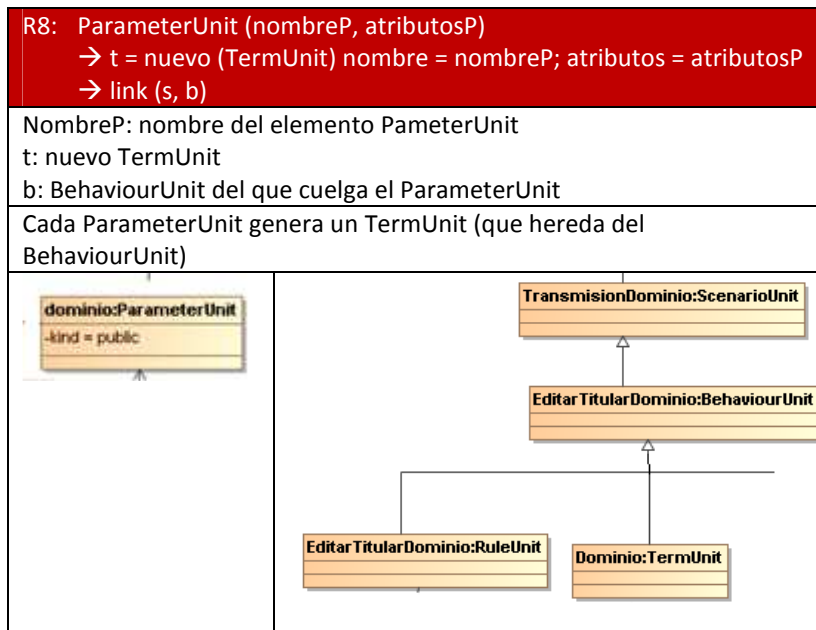
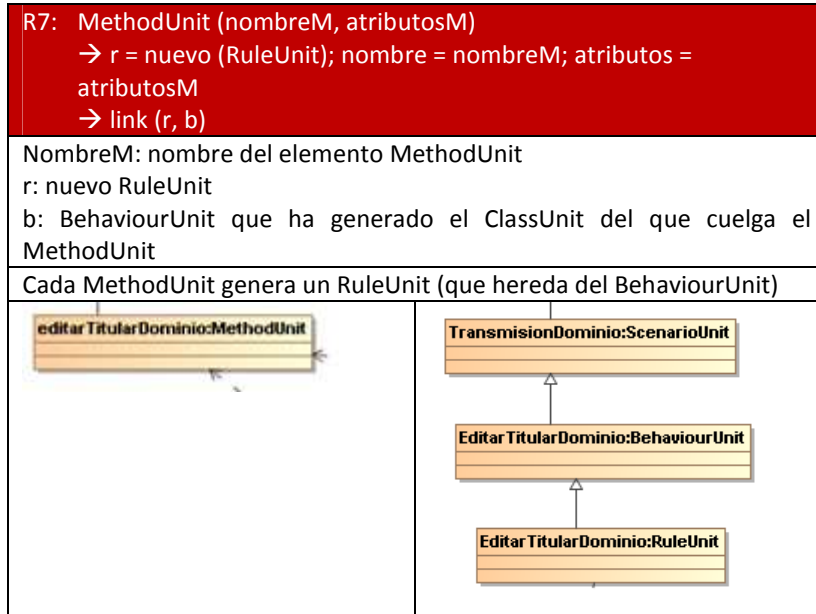
En esta sección describiremos las reglas de transformación destinadas a obtener la capa de abstracción del modelo lógico refinado en el paso 2 de PREMISA. Estas reglas tomarán como origen las capas de elementos de programa y recursos de dicho modelo y como salida se obtendrá la capa de abstracción del KDM refinado.

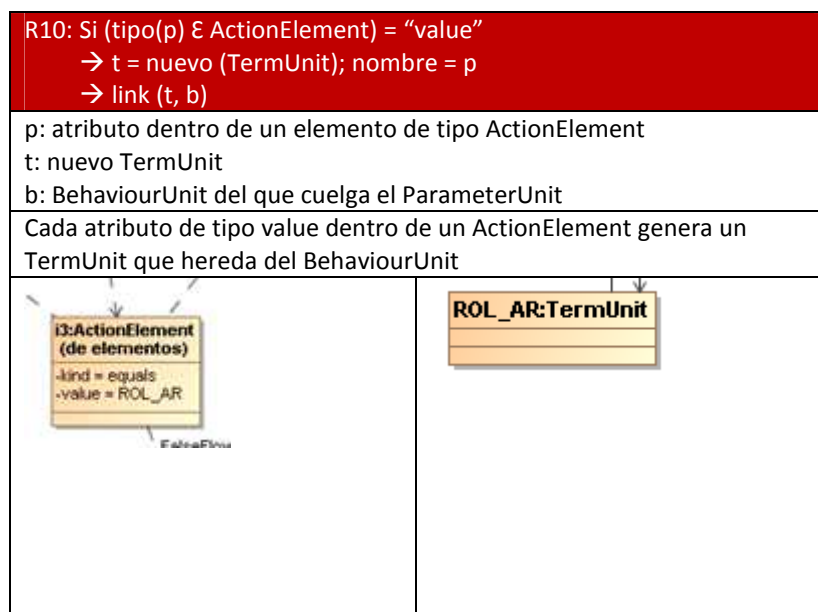
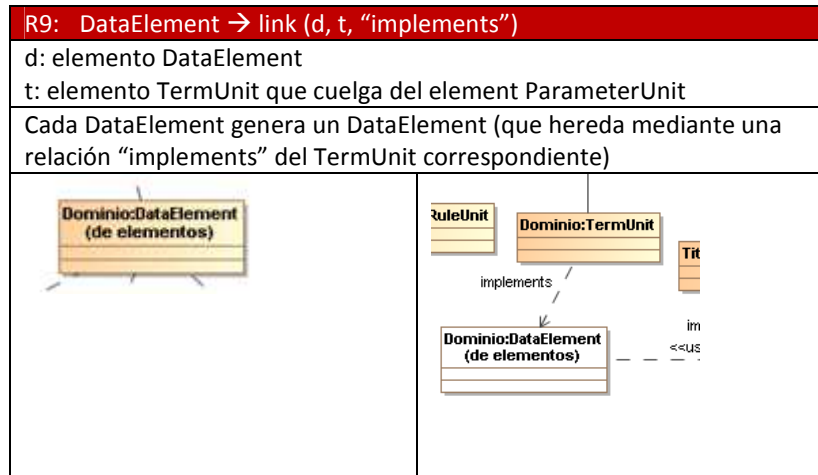
Estas reglas se aplican en el paso 3 de PREMISA, y el principal rol implicado es el desarrollador, que deberá analizar el KDM refinado para aplicarlas y obtener la capa de abstracción correspondiente.

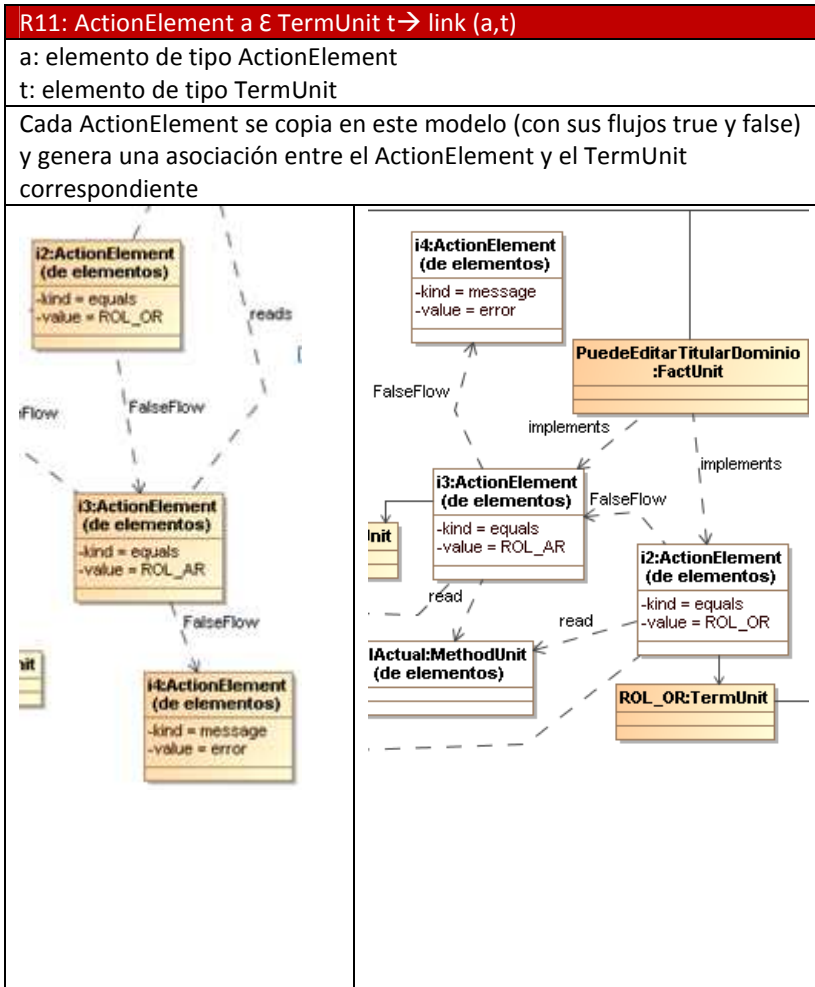
El motivo de aplicar estas reglas en dos pasos, es lograr una mayor granularidad del proceso y facilitar al desarrollador la aplicación de las mismas.

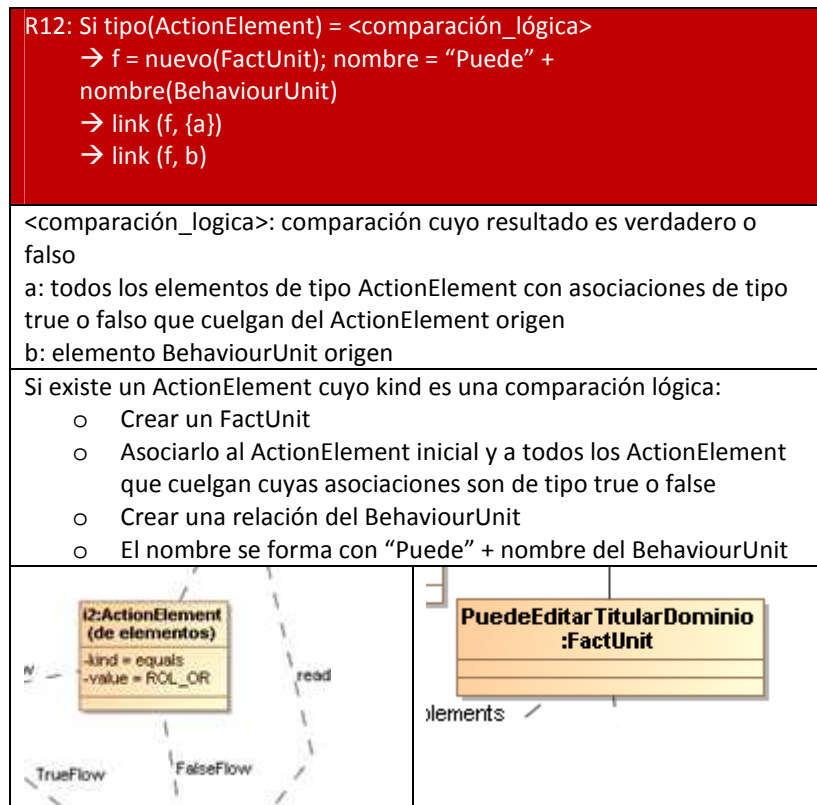
La principal aportación en este paso es la definición de unas reglas de transformación no contempladas por ADM, y que permiten pivotar de forma más fácil hacia el modelo de negocio.









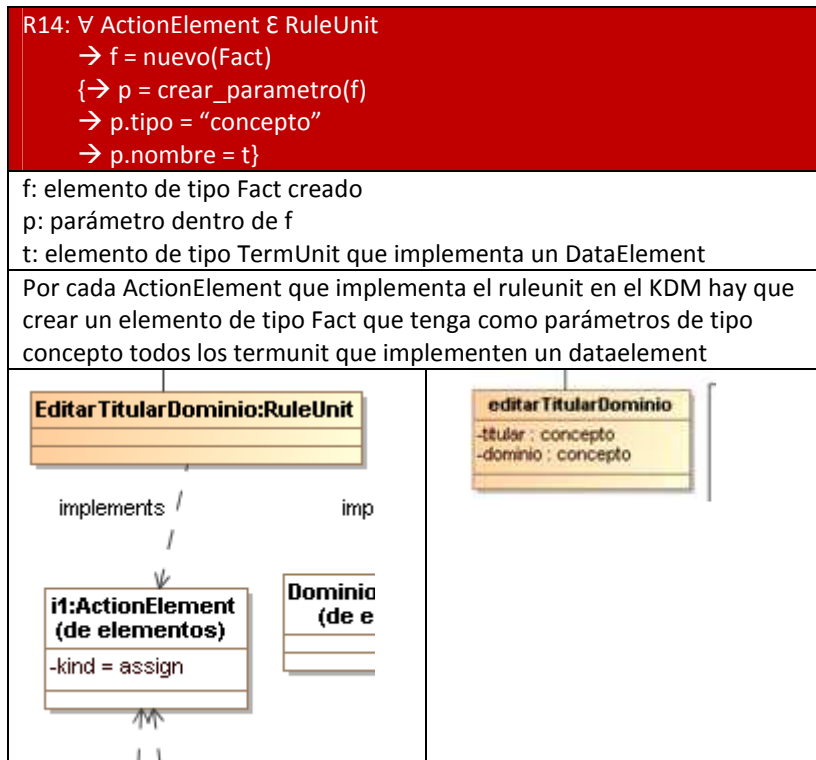
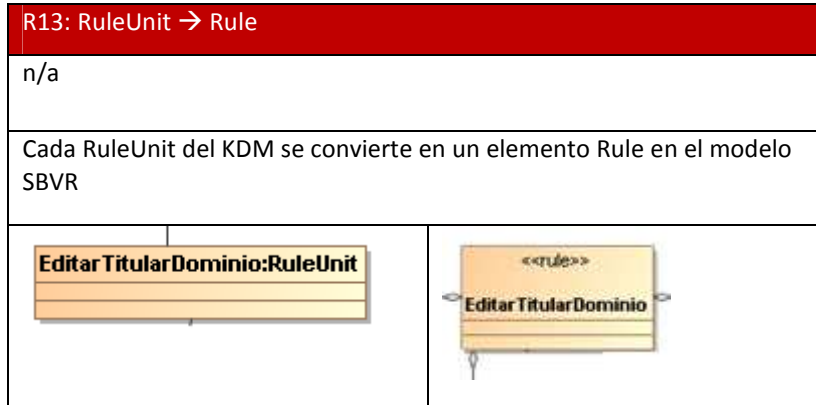


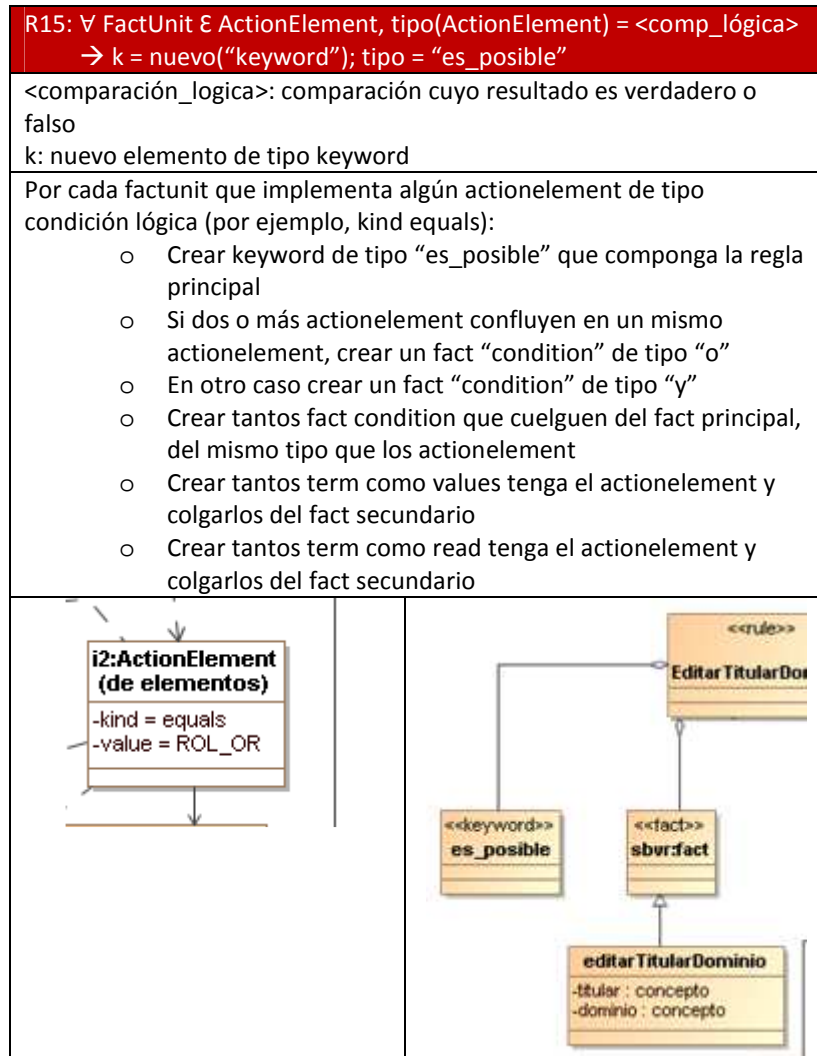
3.4.3 Obtención del modelo de negocio

En esta sección describiremos las reglas de transformación destinadas a obtener el modelo de negocio de la solución actual (*as-is*). Estas reglas tomarán como origen la capa de abstracción del modelo KDM, y como salida se obtendrá el modelo de negocio *as-is*.

Estas reglas se aplican en el paso 4 de PREMISA, y el principal rol implicado es el desarrollador, que deberá analizar la capa de abstracción del modelo KDM para aplicarlas y obtener el modelo de negocio correspondiente.

La principal aportación en este paso es la definición de unas reglas de transformación no contempladas por ADM, y que permiten obtener el modelo de negocio partiendo del modelo KDM.



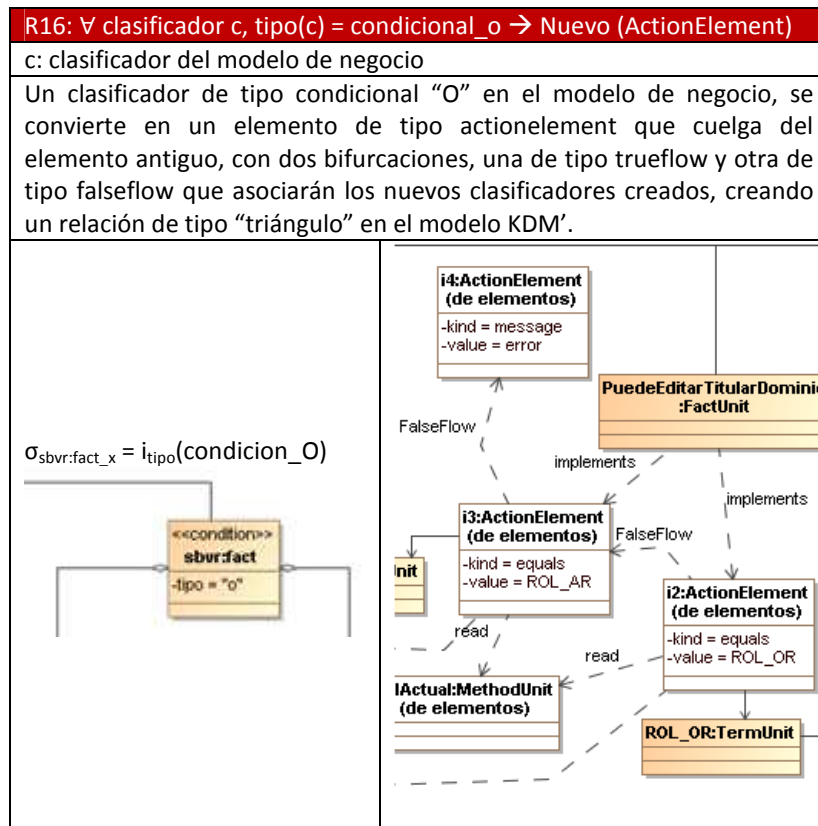


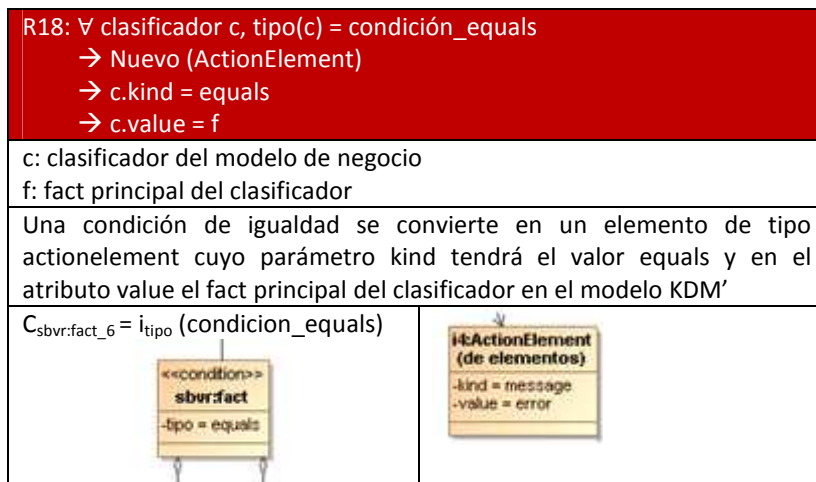
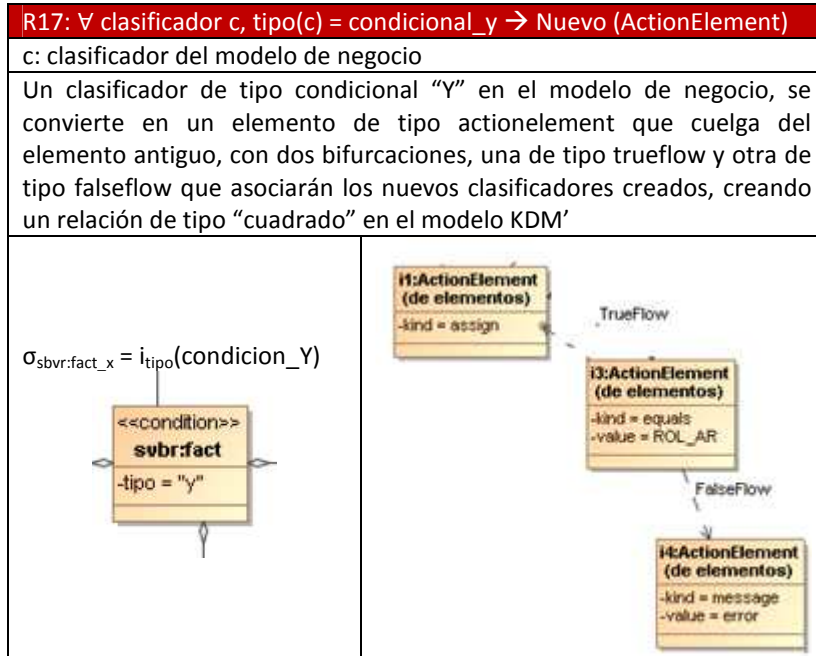
3.4.4 Obtención del modelo lógico to-be

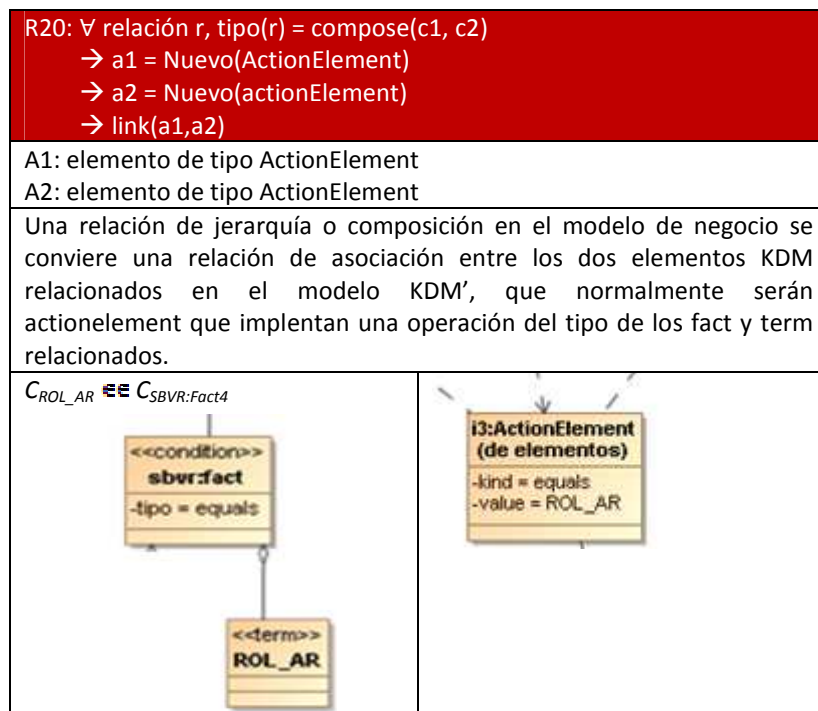
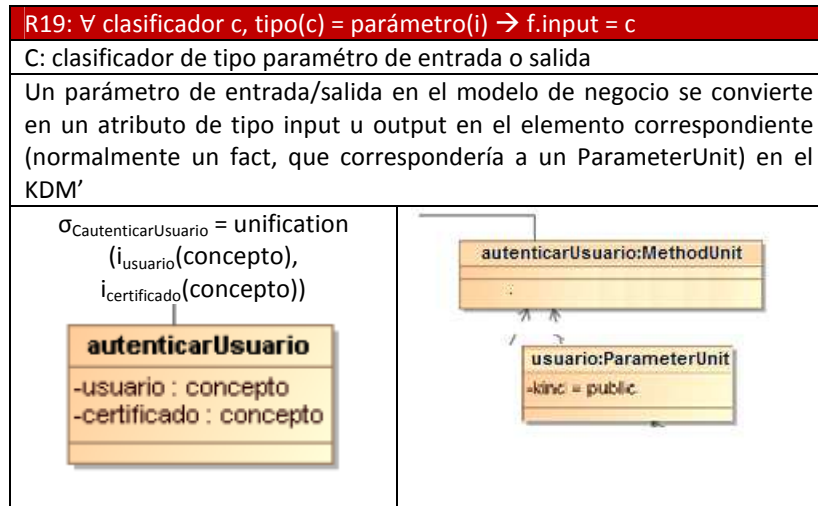
En esta sección describiremos las reglas de transformación destinadas a obtener el modelo lógico de la solución objetivo (*to-be*). Estas reglas tomarán como origen el modelo lógico de la solución actual y las operaciones descubiertas en el gap analysis, y como salida se obtendrá el modelo lógico de la solución objetivo.


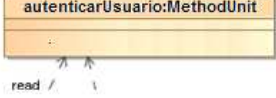
Estas reglas se aplican en el paso 7 de PREMISA, y el principal rol implicado es el desarrollador (ayudado por el analista de negocio), que deberá aplicar los operadores obtenidos en la fase de gap analysis (6) al modelo lógico de la solución actual.

La principal aportación en este paso es la definición de unas reglas de transformación que utiliza las técnicas de gap analysis para transformar el modelo lógico de la solución actual al correspondiente modelo de la solución objetivo.







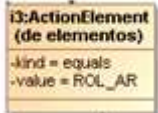
R21: \forall relación r , $\text{tipo}(r) = \text{inherits}(c1, c2)$ $\rightarrow a1 = \text{Nuevo}(\text{ActionElement})$ $\rightarrow a2 = \text{Nuevo}(\text{actionElement})$ $\rightarrow \text{link}(a1, a2)$	
A1: elemento de tipo ActionElement A2: elemento de tipo ActionElement	
Una relación de herencia en el modelo de negocio se convierte en elemento de tipo methodunit, que normalmente será candidato a ser un servicio a exponer en el modelo KDM'	
$\sigma_{\text{autenticarUsuario}} \downarrow \text{sbvr:fact}$ 	

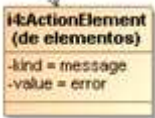
3.4.5 Obtención del código fuente

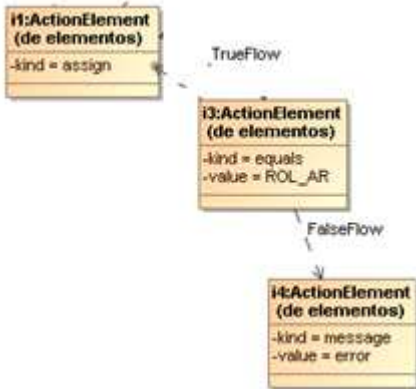
En esta sección describiremos las reglas de transformación destinadas a obtener el código fuente de la solución objetivo. Estas reglas tomarán como origen el modelo lógico de la solución objetivo, y como salida se obtendrá el código fuente final.

Estas reglas se aplican en el paso 8 de PREMISA, y el principal rol implicado es el desarrollador, que deberá tomar el modelo lógico de la solución objetivo y el código fuente de la aplicación origen, para aplicar las reglas y obtener el código fuente actualizado.

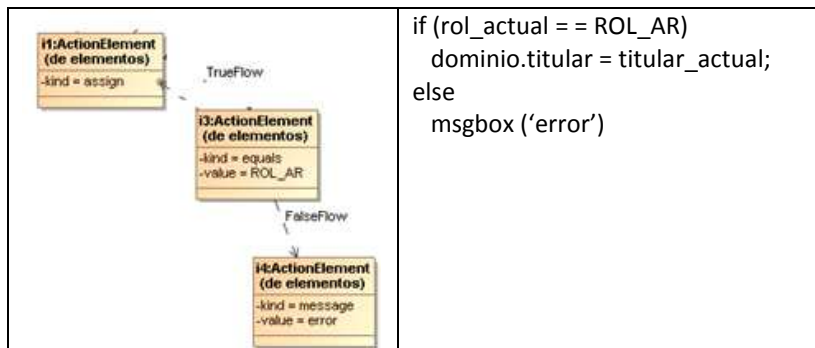
La principal aportación en este paso es la definición de unas reglas de transformación no contempladas por ADM, y que permiten obtener el código fuente final, partiendo del modelo lógico de la solución objetivo.

R22: ActionElement; kind = tipo; value = expr2 \rightarrow condicional (expr1, tipo, expr2)	
Expr1: ActionElement del que cuelga el elemento raíz	
Un elemento de tipo <i>ActionElement</i> (elemento que representa una acción) en el modelo KDM' cuyos atributos son <i>kind</i> (operador condicional utilizado) y <i>value</i> (expresión derecha a comparar), genera un condicional	
	if (rol_actual == ROL_AR)

R23: ActionElement; value = M → Mensaje (M)	
M: Mensaje por pantalla	
Un <i>ActionElement</i> en el modelo cuyo atributo <i>value</i> corresponde al mensaje a mostrar genera una instrucción de tipo “mensaje por pantalla”.	
	Msgbox ('error')

R24: ActionElement; value = bloque_instr → condicional Rama_verdadera_if (bloque_instr)	
Bloque_instr: bloque de instrucciones que se ejecuta si se cumple la condición	
ae1: ActionElement del condicional origen	
Un <i>ActionElement</i> genera un condicional cuya rama “verdadero” representa una asociación entre el <i>ActionElement</i> en el modelo que lo origina y el <i>ActionElement</i> que se ejecuta si se cumple la condición lógica.	
	<pre>if (rol_actual == ROL_AR) dominio.titular = titular_actual; else msgbox ('error')</pre>

R25: ActionElement; value = bloque_instr → condicional Rama_verdadera_if (bloque_instr)	
Bloque_instr: bloque de instrucciones que se ejecuta si se cumple la condición	
ae1: ActionElement del condicional origen	
La rama “verdadero” de un condicional genera una asociación entre el <i>ActionElement</i> en el modelo que lo origina y el <i>ActionElement</i> que se ejecuta si no se cumple la condición lógica.	



Los ejemplos concretos para cada una de estas secciones (obtención de modelos, transformaciones entre ellos, etc.), serán vistas en el siguiente capítulo, aplicadas al caso real de Red.es.

***Capítulo 4. Modernización en
Red.es con PREMISA***

Con el fin de validar el subobjetivo 3 comentado en la sección 1.2, y una vez propuesto el proceso de modernización, este capítulo describe el resultado de su aplicación al contexto de Red.es. Para comprender mejor el ámbito de aplicación, este capítulo describirá en primer lugar el método de validación utilizado, para luego detallar el contexto de aplicación (el sistema para la gestión de nombres de dominios “.es”, y en concreto la funcionalidad para la transmisión de dominios) como base para la aplicación de PREMISA.

Por último se describirá el resultado de la aplicación del proceso de modernización al caso concreto de Red.es, comentando las conclusiones y lecciones aprendidas de cada una de las iteraciones realizadas.

4.1 Método de Validación

El método de investigación utilizado en este trabajo de Tesis Doctoral, que ha sido presentado en el capítulo 1, es el resultante de la combinación del método propuesto en Marcos y Marcos [39] para investigación en el campo de la Ingeniería del Software, y de Investigación en Acción (*Action Research*) propuesto en Avison *et al.* [7].

En el método de Investigación en Acción, la validación se realiza junto con la etapa de resolución, como parte de un proceso iterativo y cíclico de realimentación, lo que permite ir resolviendo el problema al mismo tiempo que se va validando. El proceso de investigación definido en el método de investigación en acción no es por tanto un proceso lineal, sino que va avanzando mediante la realización de ciclos, llamados ciclos de investigación en acción, proporcionando así la oportunidad de ir fortaleciendo los resultados y el aprendizaje obtenido de cada ciclo de la investigación en acción. Este fortalecimiento de los resultados puede obtenerse, también, expandiendo el alcance de la investigación; por ejemplo, aplicando la investigación a diferentes campos relacionados, aumentando, de este modo, la generalidad de los resultados obtenidos en cada ciclo de la Investigación en Acción.

En este trabajo de Tesis Doctoral, el proceso de validación se ha ido realizando de forma continua y progresiva junto con la etapa de resolución. Así, cada ciclo de investigación en acción se ha aplicado iterativamente sobre el caso real de Red.es.

La Figura 12 muestra, de forma completa, el proceso de validación de este trabajo de investigación.

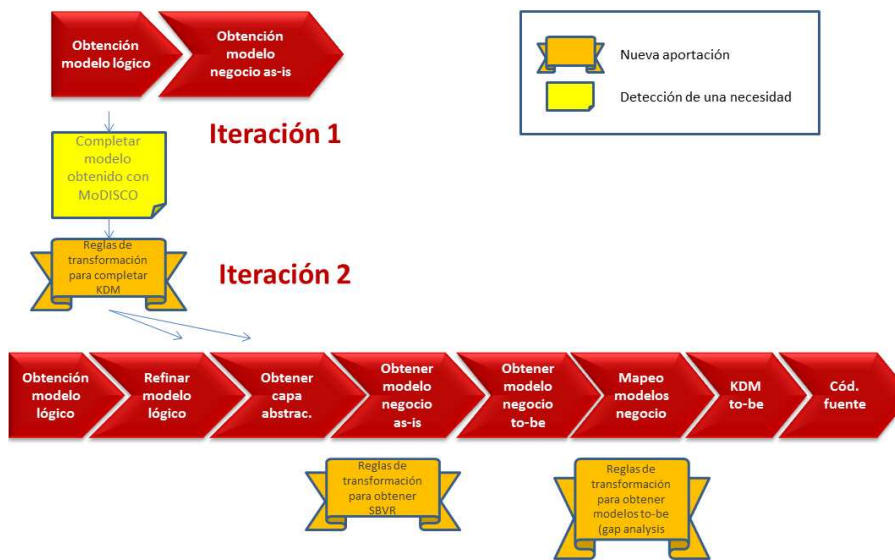


Figura 12. Proceso de Validación de este trabajo de Tesis Doctoral

Como puede verse en la Figura 12, partiendo de la transmisión de dominios “.es”, se han realizado dos iteraciones.

La primera iteración consistió en la obtención del modelo lógico, utilizando PREMISA, y se obtuvo como resultado un modelo KDM no completo que impidió obtener el modelo de negocio de la solución actual.

La segunda iteración partía de los resultados de la primera, y la principal actividad consistió en actualizar el proceso de modernización para poder aplicarlo de forma completa al caso real de Red.es.

De este modo, y una vez detectadas las primeras necesidades del método se ha definido una propuesta actualizada, que ha sido validada a través de la realización de un nuevo ciclo de investigación. De este modo, gracias al continuo análisis de necesidades, se han ido integrando las diferentes propuestas que han dado forma a este trabajo de Tesis Doctoral.

A continuación se presenta el contexto de validación, y principal ámbito objeto de trabajo de esta Tesis Doctoral, indicando tanto las necesidades detectadas, como las nuevas aportaciones que fueron surgiendo a partir de ellos.

4.2 Funcionalidad y ámbito del sistema de gestión de dominios “.es”

Red.es como Autoridad de Asignación delegada por IANA e ICANN para la gestión de los dominios del indicativo “.es”, cuenta con un sistema de información para la gestión operativa de los mismos.

Este sistema ha ido evolucionando a lo largo del tiempo. Veamos a continuación el detalle de las distintas etapas por las que ha ido pasando: el sistema original, el inicial (que tomaremos como base para esta validación) y el objetivo.

4.2.1 Sistema original

El sistema para la gestión de dominios “.es” surge en 2003 [51], como soporte a la asignación y gestión de los “.es”, según la disposición que habilita a Red.es como Autoridad de Asignación para el *ccTLD* del indicativo de España [13].

Este sistema se basa en una arquitectura monolítica (una única capa) como se muestra en la Figura 13. La principal desventaja que presenta esta tecnología es que no se tienen bien definidas las diferentes capas del modelo, por lo que se mezclan entre sí el acceso a datos con la lógica de negocio y de presentación.

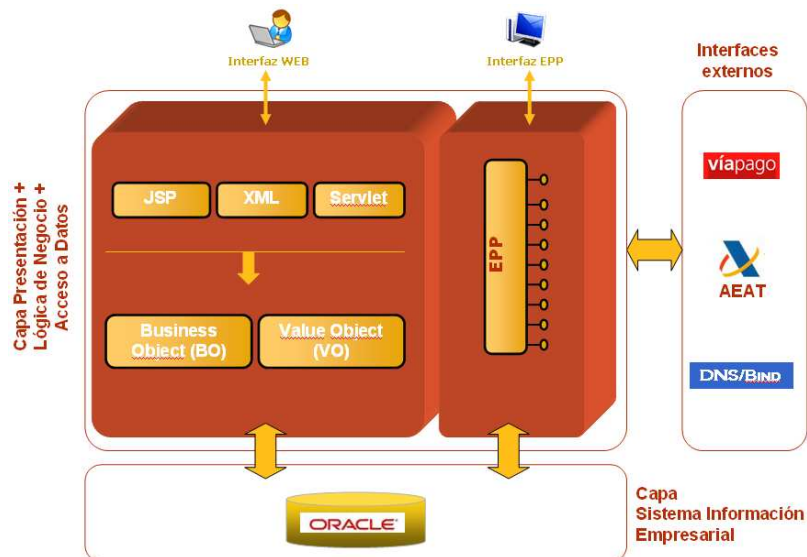


Figura 13. Arquitectura inicial del Sistema (2003)

4.2.2 Sistema inicial

El sistema del cual partimos para evolucionar, se apoya en la anterior arquitectura de 2003 sobre la que se fueron añadiendo nuevas funcionalidades, separando el acceso a la información de la presentación y la lógica de negocio, para paliar las desventajas del anterior enfoque. Sin embargo, aunque las capas de este modelo están mejor definidas, no se presenta un grado de independencia suficiente, como se puede observar en la Figura 14, al existir accesos directos a los datos sin pasar por la capa intermedia. Asimismo las dependencias entre ambas arquitecturas complica la escalabilidad del Sistema. Esta arquitectura, que no evoluciona desde 2004, se considera actualmente obsoleta para absorber las nuevas funcionalidades y la creciente carga que soporta el sistema [29].

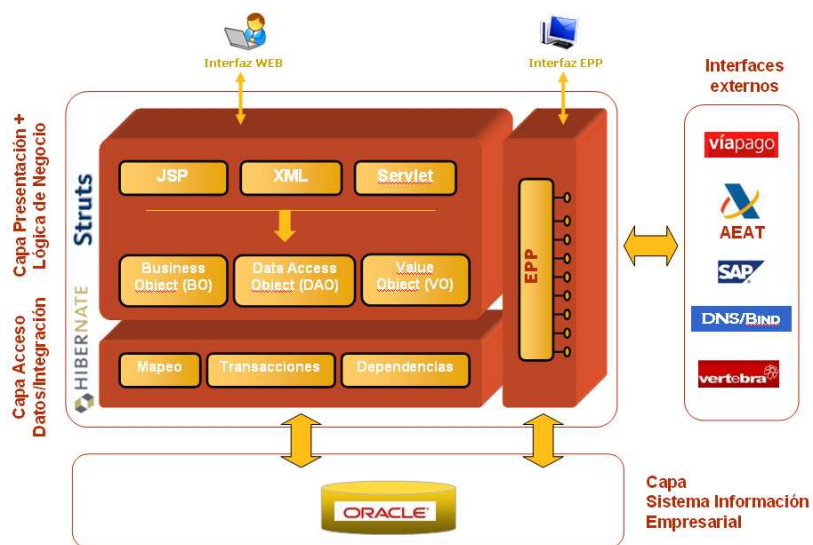


Figura 14. Arquitectura complementaria del Sistema (2004)

Como se puede observar en la definición de la arquitectura, el Sistema contempla dos formas de acceso a través de sendos interfaces:

- Interfaz WEB: es el interfaz visual web, a través de HTTP [29], utilizado por todos los implicados en el proceso de asignación y gestión de nombres de dominio.
- Interfaz EPP: Es el interfaz utilizado por los Agentes Registradores (entidades intermediarias en la gestión de dominios), en modo servicio, para solicitar la asignación de un nombre de dominio o su gestión de forma automática. El protocolo EPP es un estándar de facto, comúnmente aceptado por la mayoría de Registros a nivel internacional para la gestión de dominios, pero que, pese a sufrir numerosas revisiones, no ofrece un protocolo único de intercambio de información. Trabaja bajo el protocolo HTTP y define el formato XML para el intercambio de datos, pero cada Registro establece sus propias extensiones para trabajar de acuerdo a la legislación de cada país.

La infraestructura de este sistema se encuentra desplegada en el CPD de Red.es, y cuenta con dos partes bien diferenciadas:

- La primera es la necesaria para el correcto funcionamiento del Sistema, y que se muestra en la figura Figura 15. Aquí se puede observar como el sistema se basa en alta disponibilidad, y el acceso seguro a la información.
- La segunda es la encargada de la propagación de los nombres de dominio en los servidores DNS, tal como muestra la Figura 16. En ella se puede observar la estructura global de la red, y el esquema para la propagación de los nombres de dominio.

La Figura 17 muestra un resumen de los distintos interfaces y tipos de acceso que el sistema ofrece a los diferentes perfiles autorizados, cada uno con sus respectivas funciones y permisos (para más detalle, consultar los anexos II y III).

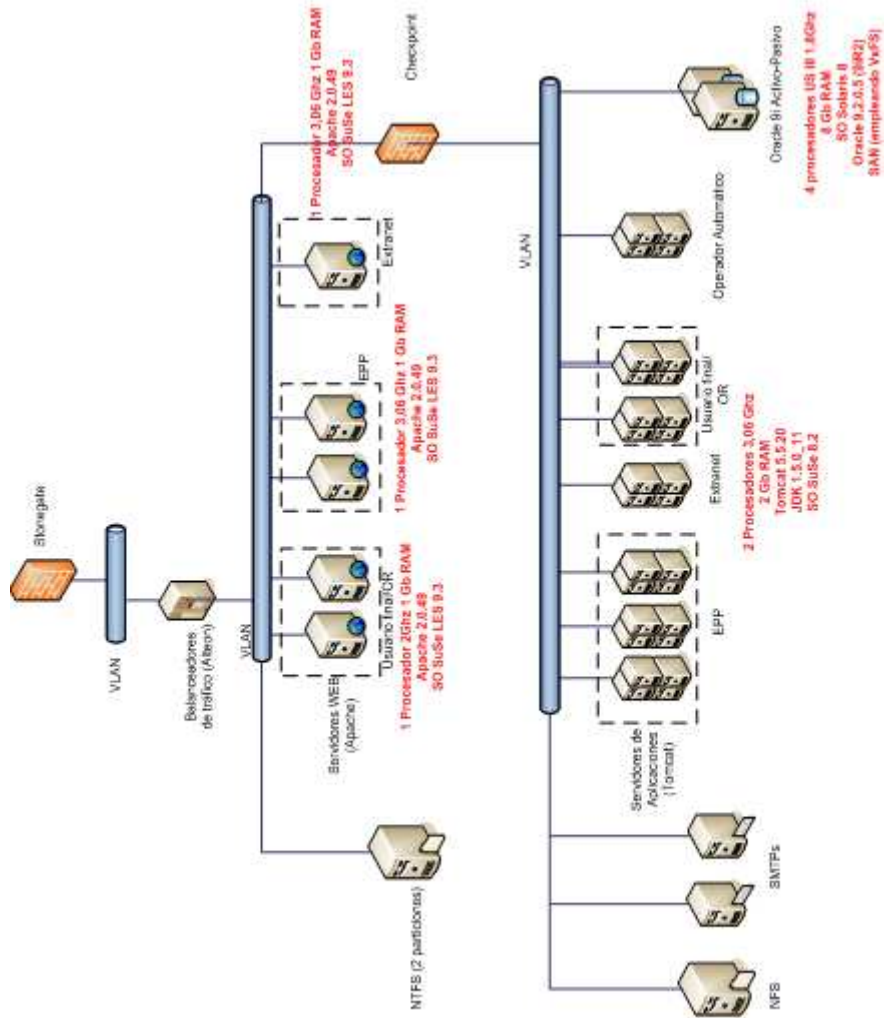


Figura 15. Infraestructura del sistema de dominios

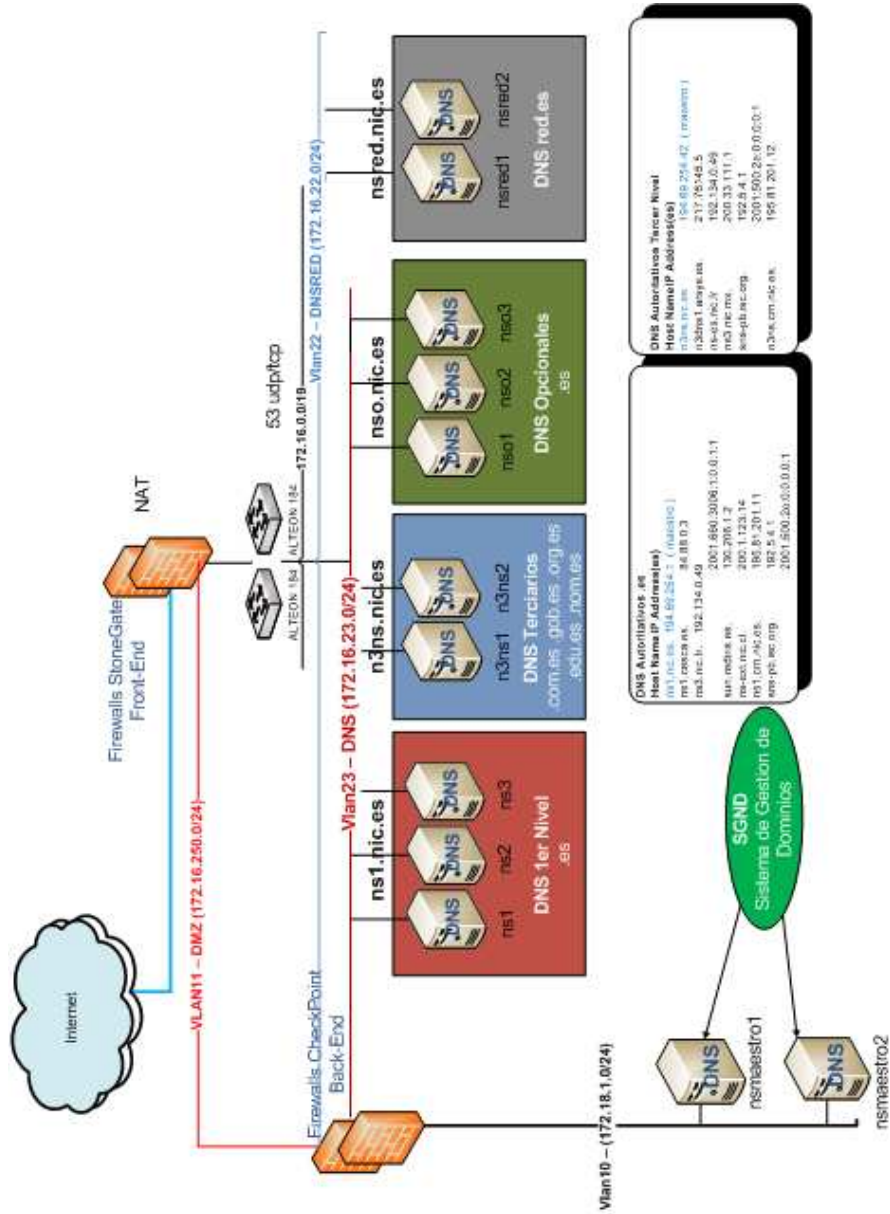


Figura 16. Infraestructura DNS

Los procesos que cada uno de estos perfiles puede gestionar son los siguientes:

- **Usuario no registrado:** Un usuario no registrado, únicamente podrá consultar, a través del portal, los datos públicos correspondientes a un nombre de dominio (whois), siguiendo la normativa de la LOPD. El detalle de la funcionalidad se muestra en la Figura A2- 1.
- **Usuario registrado:** El usuario registrado o usuario final del Sistema, es el perfil del público en general, que accede al sistema mediante un registro previo, y al que se le permite solicitar nuevos dominios y gestionarlos, entre otros. El detalle de la funcionalidad se muestra en la Figura A2- 3.
- **Operador del centro de atención al usuario (CAU):** Este perfil permite a sus usuarios la resolución de dudas de las personas que contactan al servicio de atención al usuario, así como la solución de incidencias de primer nivel. Las funcionalidades propias de este perfil comprenden aspectos relativos a la información de datos de un dominio o de sus contactos asociados, la consulta del estado de una solicitud o de los servidores DNS asociados a un dominio. El detalle de la funcionalidad se muestra en la Figura A2- 2.
- **Operador de Red:** El Operador de Red (OR) es un usuario con perfil avanzado, que recae en la propia Entidad Red.es, con objeto de realizar las labores encomendadas a la Autoridad de Asignación, al que se le permite administrar y gestionar el sistema, así como intervenir como figura principal en los flujos básicos operativos (aprobar solicitudes de alta que requieren verificación previa, etc.). El detalle de la funcionalidad se muestra en la Figura A2- 4.
- **Agente Registrador:** Los Agentes registradores (AR) son entidades encargadas de gestionar los dominios en nombre del usuario final. El detalle de la funcionalidad se muestra en la Figura A2- 5.

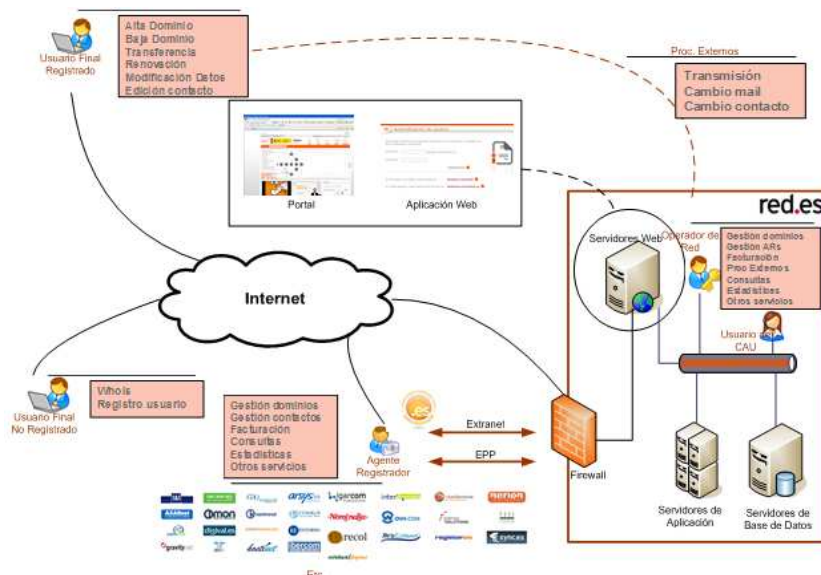


Figura 17. Interfaces de acceso al sistema actual

4.2.3 Sistema objetivo

La propuesta de modernización será aplicada, dentro de Red.es, al sistema de gestión de nombres de dominios “.es”. Más concretamente, se utilizará el módulo correspondiente a la transmisión de dominios como base para esta evolución (disponible en los entornos productivos en la URL <http://www.dominios.es>).

Esta evolución consistirá en la implementación de los mecanismos necesarios para modernizar el sistema, de cara a integrar los procesos relacionados con la gestión de dominios “.es” en la eAdministración, hito marcado por el Gobierno de España para antes del año 2015.

4.3 Evolución y modernización del sistema de dominios.es

4.3.1 Objetivos de modernización

El módulo para la transmisión de dominios es uno de los módulos más utilizados, e incorpora flujos lo suficientemente representativos para la aplicación

y validación de la propuesta de modernización. Actualmente la transmisión de nombres de dominio, que consiste en el cambio de titular del mismo, se realiza de forma manual, desde la solicitud por parte del ciudadano o de terceros interesados, hasta la finalización del proceso, pasando por todas las fases del ciclo de vida del mismo.

Este tipo de tramitación incide en los tiempos de respuesta, que en la mayoría de pasos requieren de validaciones manuales, al no incluir, por ejemplo, la utilización de certificados digitales que permitan asegurar aspectos como la autenticidad, integridad, confidencialidad o no repudio de las solicitudes y los trámites. Todos estos puntos impactan en aspectos como costes, insatisfacción del usuario, etc.

La modernización de este módulo busca, además, cumplir con la legislación vigente en materia de eAdministración, como puede ser el acceso telemático y garantizado de los ciudadanos, según la Ley 11/2007 de acceso electrónico a los servicios de la Administración Pública [12] o el Esquema Nacional de Interoperabilidad (ENI) [58], que permitirá estandarizar los servicios necesarios para la interconexión entre los diferentes Organismos, cumpliendo con los hitos marcados como prioritarios para la primera fase del proceso de modernización.

Pasaremos a continuación a describir las iteraciones realizadas dentro de la aplicación de PREMISA para la evolución del módulo de transmisión de dominios dentro del sistema informático para la gestión de los nombres de dominios “.es”. Para cada una de estas iteraciones se comentarán las conclusiones y las lecciones aprendidas que deberán ser tenidas en cuenta para la realización de una nueva iteración.

Cabe destacar que el proceso de modernización tal cual se ha planteado en el capítulo 3, es la propuesta final que se ha obtenido tras realizar las modificaciones necesarias al aplicarlo de forma incompleta en la primera iteración.

4.3.2 Primera iteración

4.3.2.1 Paso 1: Obtención del modelo lógico

El primer paso del proceso de modernización consiste en obtener el modelo KDM automáticamente a partir del código fuente del sistema actual. Para ello, y tal cual propone PREMISA, se ha utilizado la herramienta MoDISCO, obteniendo como resultado el modelo mostrado en la Figura 18, en la que se muestra una

captura de pantalla correspondiente a la representación visual del KDM obtenido con esta herramienta.

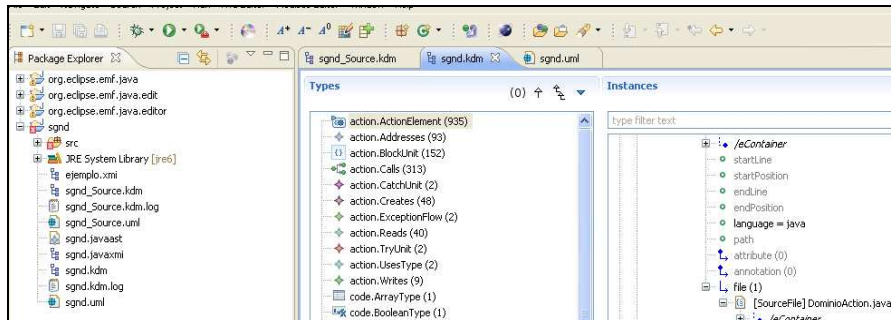


Figura 18. Modelo KDM generado con MoDISCO

El modelo completo en el formato de intercambio utilizado por MoDISCO (XMI) es el mostrado a continuación:

```
<model xmi:id="id.0" xmi:type="code:CodeModel">
  <codeElement xmi:id="id.1" xmi:type="code:StringType" name="char"/>
  <codeElement xmi:id="id.2" xmi:type="code:IntegerType" name="int">
    <codeElement xmi:id="id.3" xmi:type="code:RecordType" name="TDominio">
      <itemUnit xmi:id="id.4" name="Cod_Dominio" type="id.1" ext="Char 50" size="50"/>
      <itemUnit xmi:id="id.5" name="Nombre_Dominio" type="id.1" ext="Char 50"
        size="50"/>
      <itemUnit xmi:id="id.6" name="Cod_Titular" type="id.2"/>
    </codeElement>
    <codeElement xmi:id="id.7" xmi:type="code:RecordType" name="TContacto">
      <itemUnit xmi:id="id.8" name="Cod_Contacto" type="id.2"/>
      <itemUnit xmi:id="id.9" name="Nic_Handle" type="id.1" ext="Char 10" size="10"/>
      <itemUnit xmi:id="id.10" name="Nom_Contacto" type="id.1" ext="Char 50" size="50"/>
    </codeElement>
    <codeElement xmi:id="id.11" xmi:type="action:ActionElement" name="i1" kind="Assign">
      <actionRelation xmi:id="id.12" xmi:type="action:Reads" to="id.3" from="id.7"/>
      <actionRelation xmi:id="id.13" xmi:type="action:Writes" to="id.7" from="id.3"/>
    </codeElement>
    <codeElement xsi:type="code:MethodUnit" xmi:id="id.14" name="setTitular"
      export="public">
      <parameterUnit name="titular" xmi:id="id.15" type="id.7" kind="public"/>
    </codeElement>
    <codeElement xsi:type="code:MethodUnit" xmi:id="id.16" name="editarDominio"
      export="public">
      <parameterUnit name="dominio" xmi:id="id.17" type="id.3" kind="public"/>
      <parameterUnit name="titular" xmi:id="id.18" type="id.7" kind="public"/>
    </codeElement>
    <codeElement xsi:type="code:MethodUnit" xmi:id="id.35" name="getRol"
      export="public"/>
    <codeElement xmi:id="id.19" xmi:type="action:ActionElement" name="i2" kind="Equals"/>
  </model>
```

```

<codeElement xmi:id="id.20" xmi:type="code:Value" name="ROL_AR" type="id.1">
  <actionRelation xmi:id="id.21" xmi:type="action:Reads" to="id.35" from="id.19"/>
  <actionRelation xmi:id="id.22" xmi:type="action:TrueFlow" to="id.11" from="id.19"/>
</codeElement>
<codeElement xmi:id="id.23" xmi:type="action:ActionElement" name="i3" kind="Equals"/>
<codeElement xmi:id="id.24" xmi:type="code:Value" name="ROL_OR" type="id.1">
  <actionRelation xmi:id="id.25" xmi:type="action:Reads" to="id.35" from="id.23"/>
  <actionRelation xmi:id="id.26" xmi:type="action:TrueFlow" to="id.11" from="id.23"/>
</codeElement>
</model>

```

En este modelo podemos observar la jerarquía de elementos y las relaciones entre ellos. Así, por ejemplo, el objeto *TContacto* de tipo *RecordType*, haría referencia al tipo de registros en una tabla en base de datos, que a su vez tiene los atributos (elementos de tipo *ItemUnit*) *Cod_Contacto*, *Nic_handle* y *Nom_Contacto*. Cada uno de estos atributos son de tipo entero y carácter.

Este modelo será pasado al desarrollador para que, siguiendo las reglas marcadas en PREMISA, se pueda obtener el modelo transformado correspondiente.

4.3.2.2 Paso 2: Obtención del modelo de negocio *as-is*

La propuesta inicial de PREMISA contemplaba el paso directo del modelo lógico obtenido con MoDISCO al modelo de negocio del sistema actual. Sin embargo, echando un vistazo al modelo KDM obtenido podemos observar que faltan las clases correspondientes a la capa de abstracción, que son las que permitirán pivotar hasta el modelo SBVR.

Por lo tanto, al intentar aplicar las reglas de transformación, descubrimos que no podemos obtener un modelo SBVR. Al no disponer de un modelo de negocio, la iteración finaliza aquí, sin poder realizar las transformaciones necesarias para llegar a la solución objetivo.

4.3.2.3 Lecciones aprendidas de la primera iteración

A la vista de los resultados arrojados, podemos concluir que la utilización de la herramienta MoDISCO no es suficiente para la obtención de un modelo completo, por lo que deberá ser completado mediante reglas basadas en la inspección del código.

Tras plantearnos la eliminación de MoDISCO, se concluye que esta herramienta es necesaria para obtener un modelo preliminar que, de otra forma,

sería muy costoso. El anexo V muestra los componentes que MoDISCO obtiene de forma automática.

Por otro lado no hemos encontrado ninguna herramienta, basada en J2EE, que permita el descubrimiento del modelo lógico, que cubra las necesidades de Red.es.

4.3.3 Segunda iteración

4.3.3.1 Paso 1: Obtención del modelo lógico (inicial)

El primer paso dentro de esta segunda iteración consiste, al igual que en la primera, en obtener el modelo KDM automáticamente a partir del código fuente del sistema actual. Utilizando MoDISCO se obtiene el mismo modelo que en la primera iteración (ver apartado 4.3.2.1).

4.3.3.2 Paso 2: Refinamiento del modelo lógico

Tomando como base el modelo KDM obtenido, se ha realizado el refinamiento del mismo. Se han incorporado, tras realizar una inspección del código fuente, aquellos elementos no contemplados por MoDISCO.

Partiendo del código de la aplicación de gestión de nombres de dominios, correspondiente al módulo de transmisión, del que mostramos un extracto a continuación, y aplicando las reglas de PREMISA, se ha obtenido el modelo KDM mostrado en la Figura 19.

```
public class DominioAction extends SgndBaseAction implements
ModelDriven<DominioFormView> {
...

    public String editarDominio(TDominio dominio, TContacto titular) throws Exception
    {

        if( getRol().equals(SgndConstantes.ROL_AR) ||
            getRol().equals(SgndConstantes.ROL_OR) {

            dominio.setTitular(titular);

        }

    }

}
```

```

public class DominioFormView extends SgndBaseFormView {
private TContacto titular;

    public void setTitular(TContacto titular) {
        this.titular = titular;
    }
}

public class TDominio implements Serializable {
    @Id
    @Column(name = "COD_DOMINIO")
    private Int codDominio;

    @Column(name = "NOMBRE_DOMINIO")
    private String nombreDominio;
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "COD_TITULAR")
    private TContacto tCodContacto;
}

public class TContacto implements Serializable {
    @Id
    @Column(name="COD_CONTACTO")
    private Int codContacto;
    @Column(name="NOM_CONTACTO")
    private String nombre;

    @Column(name="NIC_HANDLE")
    private String nichHandle;
}

```

Cabe destacar en este paso la obtención de las relaciones entre los elementos de tipo *ActionElement*, que en el paso anterior no han sido descubiertas por MoDISCO. Dichas relaciones son necesarias para obtener la capa de abstracción, que permitirá pivotar hasta el modelo de negocio de la solución actual.

Por ejemplo, la aplicación de la regla de transformación R3 permite convertir el condicional:

```

if( getRol().equals(SgndConstantes.ROL_AR) ||
    getRol().equals(SgndConstantes.ROL_OR)

```

en dos *ActionElement* en el modelo refinado: i2, que se encarga de comprobar si el rol actual del usuario de la aplicación es de tipo ROL_OR y, en caso negativo (asociación falseflow) realizará la misma comprobación con i3 para validar si el rol del usuario actual es de tipo ROL_AR.

4.3.3.3 Paso 3: Obtención de la capa de abstracción KDM

Una vez obtenido el modelo KDM refinado, el siguiente paso consiste en generar la capa de abstracción de dicho modelo, mediante la aplicación de las reglas de transformación propuestas. Así, para el caso del sistema de gestión de dominios “.es”, el elemento editarTitularDominio (de tipo *MethodUnit*) genera un elemento *BehaviourUnit* que hereda de *TransmisionDominio*, (que a su vez es un *ScenarioUnit* derivado del *ClassUnit* del modelo KDM).

A partir de ahí, se generan los elementos de tipo *TermUnit* *ROL_AR*, *ROL_OR* (correspondientes a atributos de tipo value dentro de los *ActionElement* obtenidos en el modelo KDM refinado), dominio y contacto (derivados de los elementos de tipo *ParameterUnit* dentro de este modelo). Se generará también el elemento *PuedeEditarTitularDominio*, de tipo *FactUnit*, que representa el hecho principal del escenario, y que deriva del *MethodUnit* del KDM refinado que se está refinando, y que centraliza todas las asociaciones de tipo *TrueFlow*, *FalseFlow*, *read* y *write* con los elementos de tipo *ActionElement* y/o *MethodUnit* del modelo KDM refinado. Todo ello derivará del elemento raíz *BehaviourUnit*. La Figura 20 muestra el modelo final obtenido.

Mostramos a continuación esta capa de abstracción obtenida, en formato XMI:

```
<model xmi:id="id.27" xmi:type="conceptual:ConceptualModel" name="SGND">
  <conceptualElement xmi:id="id.28" xmi:type="conceptual:ScenarioUnit"
name="Transmision">
  <conceptualElement xmi:id="id.29" xmi:type="conceptual:BehaviorUnit"
name="Editar_dominio">
  <conceptualElement xmi:id="id.30" xmi:type="conceptual:TermUnit" name="Dominio"
implementation="id.3"/>
  <conceptualElement xmi:id="id.31" xmi:type="conceptual:TermUnit" name="Contacto"
implementation="id.7"/>
  <conceptualElement xmi:id="id.32" xmi:type="conceptual:FactUnit"
name="PuedeEditarTitular" implementation="id.19 id.23"/>
  <conceptualElement xmi:id="id.33" xmi:type="conceptual:ConceptualRole"
name="Titular de dominio" conceptualElement="id.31"/>
  <conceptualElement xmi:id="id.34" xmi:type="conceptual:RuleUnit" name="Asignar
titular" implementation="id.11">
  </conceptualElement>
</conceptualElement>
</model>
```

Este modelo será el punto de entrada para el siguiente paso, que será parseado por el desarrollador para aplicar las reglas correspondientes al mismo.

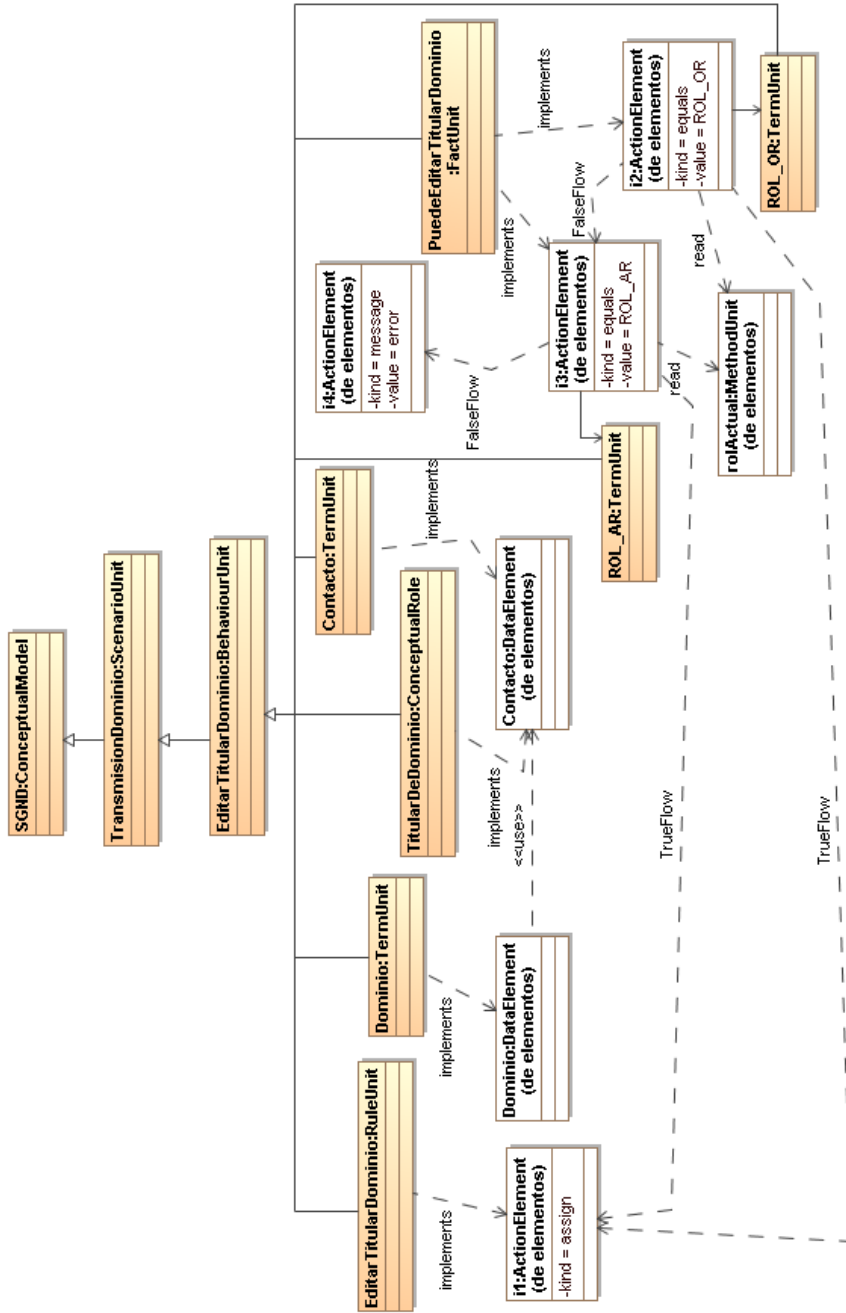


Figura 20. Capa de abstracción KDM para la transmisión de dominos

4.3.3.4 Paso 4: Obtención del modelo de negocio *as-is*

Partiendo de la capa de abstracción del paso anterior, el objetivo de esta fase consiste en aplicar las reglas de transformación que permitan obtener el modelo SBVR para la solución actual en Red.es. Este modelo es relativamente fácil de conseguir, ya que los elementos de tipo fact marcan las condiciones de la regla de negocio, como se puede observar en la Figura 21.

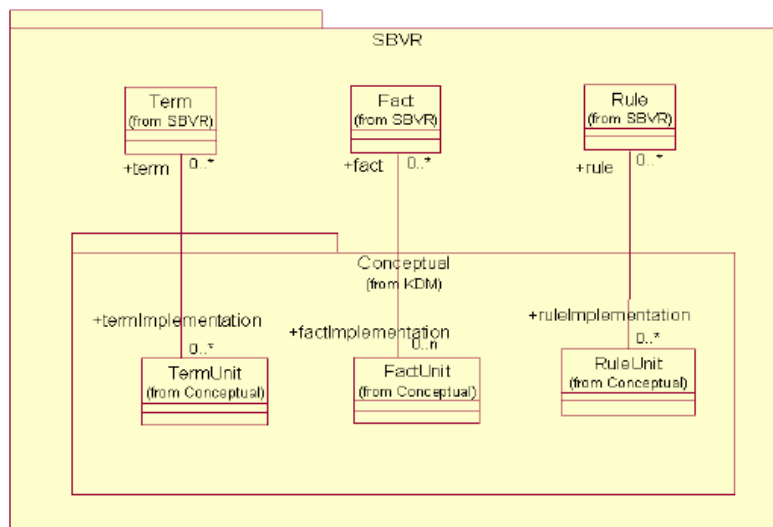


Figura 21. Mapeo entre la capa de abstracción KDM y el modelo SBVR

El modelo SBVR obtenido para el módulo de la transmisión de dominios, aplicando las reglas correspondientes a este paso, es el mostrado en la Figura 22.

4.3.3.5 Paso 5: Obtención del modelo de negocio *to-be*

Llegados a este punto, y tomando como partida el modelo de negocio obtenido en el paso anterior, los analistas de negocio deberán modelar la capa de negocio de la solución objetivo. Tras realizar un análisis DAFO, encargado por la Alta Dirección y llevado a cabo por los analistas de negocio, se ha realizado un estudio de los sistemas de distintos Registros europeos. Como principal debilidad (que a su vez se constituye como una oportunidad), se ha detectado la necesidad de adaptar el sistema a las nuevas leyes y normas de interoperabilidad [58] y de acceso electrónico de los Ciudadanos a los Servicios Públicos [12]. El resultado de este análisis de muestra en la Figura 23.

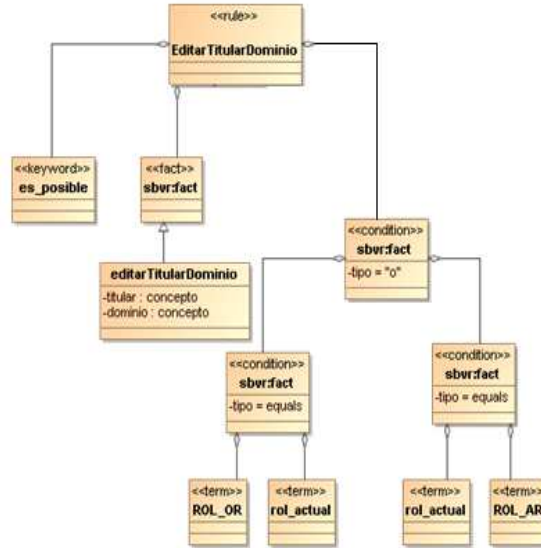


Figura 22. Modelo SBVR para la solución actual de transmisión de dominios

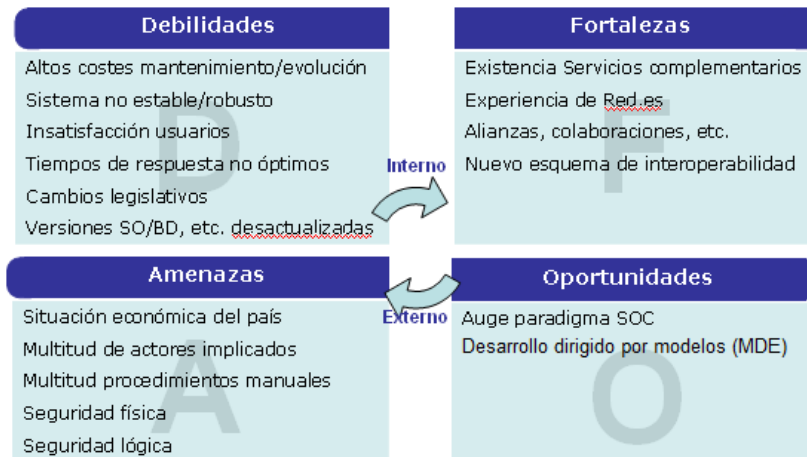


Figura 23. Análisis DAFO de la herramienta de gestión de dominios .es

Con este análisis podemos establecer las siguientes conclusiones:

Debilidades

Existen en la actualidad distintos estilos arquitectónicos implementados que están mezclados entre sí, redundando en un alto acoplamiento, que hace que existan numerosas dependencias entre los distintos módulos de la aplicación. Todo esto viene a dificultar la escalabilidad del sistema, provocando en consecuencia unos elevados costes de mantenimiento y evolución.

Precisamente es este diseño el que impide abordar la mayoría de necesidades demandadas por los diferentes *stakeholders*, incidiendo en un grado de insatisfacción mayor del deseado. Del mismo modo este planteamiento hace que el sistema no sea del todo estable ni robusto, reaccionando negativamente ante determinados picos de carga, produciendo en algunos casos caídas que afectan al servicio y a la imagen que la Entidad debe prestar a todos los implicados.

Uno de los motivos de este comportamiento se debe a los complejos *workflows* de la mayoría de los procesos del sistema, algunos de los cuales son obsoletos o pueden ser mejorables pero que, por motivos del diseño de la Arquitectura Tecnológica, no pueden ser abordados.

Otro de los principales motivos que han producido esta situación han sido los constantes cambios operativos y legislativos, que han propiciado las continuas adaptaciones del sistema mediante el despliegue de parches, la mayoría de forma urgente, perdiendo de esta forma la visión estratégica del producto. Así mismo, en cada versión evolutiva, no se aplicaban cambios estructurales ni se aprovechaban las ventajas tecnológicas disponibles en cada momento, por lo que no se ha realizado ningún análisis coste-beneficio de la actualización de las versiones del Framework, del Sistema Operativo, del SGBD, etc.

Amenazas

Existen actualmente varias amenazas catalogadas como críticas, que ponen en riesgo los objetivos para los cuales el sistema ha sido construido.

Por un lado encontramos multitud de actores implicados en los procedimientos de asignación y registro de los “.es”. Si tenemos en cuenta, por ejemplo, el papel que los Agentes Registradores³ juegan en este ámbito, el

³ Los Agentes Registradores son aquellas empresas u organismos, acreditados por Red.es como autoridad de registro, que se encargan de la asignación y gestión de los dominios en nombre de un usuario final que le encarga su tramitación. Actúan por lo tanto como intermediarios entre Red.es y el usuario final

principal objetivo a cumplir en cualquier correctivo o evolutivo del Sistema es la transparencia, ya que cualquier cambio en el interfaz que éste ofrece podría provocar un efecto en cadena incidiendo en el servicio que presta al cliente final.

Otra de las amenazas que se ha detectado en este análisis es la existencia de numerosos procedimientos que se están gestionando de forma manual por fuera del sistema. La excesiva dedicación de recursos a este tipo de tareas incide en un coste elevado de gestión, sin contar con la existencia del riesgo del error humano o el elevado tiempo de respuesta ante determinados picos de trabajo. Sobre todo en tiempos de crisis económica como la que estamos atravesando actualmente, esta amenaza puede poner en peligro el servicio que el Registro debe prestar a todos aquellos implicados en la gestión del “.es”.

Esta amenaza añade un riesgo adicional, que va encaminado a la dificultad para cumplir uno de los objetivos encomendados a la Entidad, el de promover la Sociedad de la Información en nuestro país. En este sentido Red.es debe dinamizar el uso de las TIC, fomentando la presencia de las mismas en actividades cotidianas de los ciudadanos y promocionando el acercamiento entre la Administración Pública al servicio del usuario final mediante la utilización de las Nuevas Tecnologías. Este es precisamente el objetivo perseguido por la Ley 11/2007 [12], que confronta directamente con la tramitación manual de algunos trámites relacionados con la gestión de los dominios “.es”.

Por último pero no menos importante, destacar la seguridad del Sistema como otra de las amenazas del mismo. Por un lado la seguridad física, al no existir planes de contingencia claramente detallados, ni mecanismos de recuperación, ponen en peligro la fortaleza de un sistema tan importante como este. Por otro, la seguridad lógica muestra algunos agujeros de seguridad, detectados por INTECO⁴ en su última auditoría realizada sobre el código que, aun no siendo del todo críticos, podrían dejar al descubierto mecanismos para promover ataques de denegación de servicio, fuga de información o inyección de código entre otros.

Fortalezas

Los puntos fuertes que permitirían solventar en gran parte las debilidades encontradas, se fundamentan sobre todo en la gran experiencia que Red.es dispone para abordar grandes proyectos y soluciones de este tipo.

⁴ **El Instituto Nacional de Tecnologías de la Comunicación (INTECO)**, promovido por el Ministerio de Industria, Turismo y Comercio, es una plataforma para el desarrollo de la Sociedad del Conocimiento a través de proyectos del ámbito de la innovación y la tecnología

Buena muestra de ello es la existencia de numerosos servicios desarrollados en torno a la dinamización de la Sociedad de la Información, la mayoría de los cuales podrían ser aprovechados para ofrecer un valor añadido al actual sistema, como por ejemplo la pasarela de pago SPT [65], o la implantación del sistema de identificación y firma electrónica basado en ASF [3].

Fruto de estos servicios, Red.es dispone de un gran conjunto de alianzas con otras Entidades Públicas, empresas privadas y otros actores dentro de la Administración General del Estado, que le permiten obtener colaboraciones en distintos ámbitos del sector. Como resultado de estas uniones surgen nuevas formas de crecimiento, como por ejemplo la adopción del estándar de la WEB 2.0 o la liberación de código (en colaboración con CENATIC⁵), así como la posibilidad de contar con Organismos como RedIris para la implantación de DNSSec o implantar IPv6 entre otros.

Pero sobre todo, la principal fortaleza es el papel que Red.es juega en el reciente Real Decreto sobre el nuevo esquema de interoperabilidad [58], que regula y promueve los mecanismos para compartir información y servicios como objetivo único para establecer sinergias entre los diferentes actores de la Administración General del Estado, mediante la puesta en común de servicios que permiten eliminar redundancias y aprovechar las ventajas del intercambio de información y conocimiento entre éstos. Este nuevo esquema le permitirá a Red.es jugar un papel importante como actor clave dentro de la Administración Pública y como nexo de unión entre ésta, los ciudadanos y el sector privado. En concreto, con el Plan Avanza [54] (cuyo principal precursor fue Red.es), se consiguió recuperar el retraso de España respecto de la Unión Europea, especialmente en cobertura y conectividad.

Desde el 2005 hasta el 2010, se realizó la primera fase de este Plan, que supuso el inicio de las políticas digitales españolas. Sus buenos resultados han sido avalados por la OECD (*Organization for Economic Co-Operation and Development*) en su informe “Buena Gobernanza en las Políticas Digitales: Cómo Maximizar el Potencial de las TIC. El Caso del Plan Avanza” [45]. Así, gracias a los esfuerzos realizados durante estos años, se ha conseguido que el mercado de

⁵ **CENATIC** es el Centro Nacional de Referencia de Aplicación de las Tecnologías de la Información y la Comunicación (TIC) basadas en fuentes abiertas. Es una Fundación Pública Estatal, promovida por el Ministerio de Industria, Turismo y Comercio (a través de la Secretaría de Telecomunicaciones y para la Sociedad de la Información y la entidad pública Red.es) y la Junta de Extremadura, que además cuenta en su Patronato con las comunidades autónomas de Andalucía, Asturias, Aragón, Cantabria, Cataluña, Illes Balears y País Vasco. También forma parte del Patronato de CENATIC las empresas Atos Origin, Bull, Telefónica y Gpex

Comercio Electrónico crezca un 16% en cifra de negocio, que España lidere en UE la tasa de empresas con banda ancha (98%), que superemos en 9 puntos a la media de la UE el porcentaje de hogares con Banda Ancha, que seamos uno de los 3 líderes en el mundo en cuanto a uso de redes sociales o que seamos uno de los países de la UE que mejor cobertura y penetración tenga de la TDT.

En la segunda fase de este plan (Plan Avanza 2) [53], se pretende situar a España en una posición de liderazgo en el desarrollo y uso de productos y servicios TIC avanzados. Para ello, se centrarán los esfuerzos, no tanto en dinamizar la oferta, sino en fomentar la demanda, ayudar a la PYME y consolidar una industria TIC propia especializada en sectores estratégicos. Así, una de las principales bazas del Plan será implantar la eAdministración antes del 2015, como ha fijado el Gobierno de España. Para ello, se digitalizarán todos los trámites administrativos, se finalizará la incorporación de la firma electrónica (DNI electrónico) y se hará una migración de todos los registros hacia el formato digital. En este sentido el Sistema de gestión de los “.es” deberá estar preparado para adaptarse a este nuevo contexto y aprovechar la posición de Red.es en el ámbito de las Administraciones Públicas para hacer de esta ventaja no sólo una fortaleza, sino también contar con la experiencia en el desarrollo de este tipo de proyectos para impulsar la eAdministración como parte de la Sociedad de la Información.

De esta forma, el Gobierno quiere que las empresas TIC españolas sean protagonistas en la construcción del “Internet del Futuro”, impulsando el capital semilla y el capital riesgo TIC, fortaleciendo el sector de los contenidos digitales, promoviendo las TIC verdes y desarrollando proyectos tractores que fomenten el ahorro y la eficiencia en España mediante el desarrollo de las aplicaciones TIC en ámbitos de futuro.

Oportunidades

Dentro del Nuevo Esquema de Interoperabilidad [58] que se está construyendo como forma de potenciar la Sociedad de la Información, parece que el paradigma de Orientación a servicios [6][26][66] es la mejor manera de encajar la demanda de servicios que todos los agentes están demandando dentro y fuera del sector TIC. Si tenemos en cuenta que este paradigma está cobrando cada vez más fuerza en el ámbito tecnológico, y en particular en el desarrollo de Sistemas de Información, se convierte en una oportunidad de crecimiento para aprovechar y sacar el máximo partido a ese nuevo modelo.

Teniendo en cuenta lo anteriormente expuesto y el análisis DAFO realizado, se establecen los siguientes criterios de priorización que permitan abordar el proyecto en fases evolutivas:

1. Abordar aquellos puntos de evolución para los que existe una fecha determinada de implantación, sobre todo los derivados de decisiones políticas y aquellos que puedan impactar en el desarrollo de la Sociedad de la Información.

¿Cómo?: **modernizando** el sistema para integrar aquellos aspectos relacionados con la eAdministración (ley 11/2007 [12], nuevo esquema de interoperabilidad [58], etc.)

2. Incidir en los puntos **débiles** para ofrecer una solución de base, robusta y tecnológicamente eficiente.

¿Cómo?: **redefiniendo** los procesos de negocio mediante una evolución tecnológica y funcional

3. Minimizar el impacto de las posibles **amenazas**, para conseguir un Sistema seguro y económicamente rentable.

¿Cómo?: **incorporando** los procesos manuales en el Sistema y los mecanismos para aumentar la seguridad del mismo

4. Aprovechar las **fortalezas** de la Entidad para ofrecer Servicios de Valor añadido.

¿Cómo?: **integrando** servicios existentes

5. Aprovechar las **oportunidades** del sector para conseguir alinear el Sistema con la estrategia y la tecnología

¿Cómo?: **ofreciendo** nuevos servicios a través de un Bus de Integración [31].

Cabe destacar que este proceso, además, debe cumplir los criterios establecidos por la Alta Dirección, y cuyas necesidades de mejora se muestran en la Tabla 19.

<i>Transmisión dominios .es (AS-IS)</i>	<i>Transmisión dominios .es (TO-BE)</i>
Tramitación manual	Tramitación automática
Tiempos de gestión altos	Tiempos de gestión menores
Validación manual	Validación automática
Costes de gestión inherentes	Retorno de inversión (ROI)
Comunicación manual con interesados	Comunicación vía servicios
No interconectado	Interconectado (ENI)
Riesgo de error humano	Facilitar decisiones
Riesgo seguridad	Esquema seguro (validaciones)

Tabla 19. Objetivos para la mejora de procesos del caso de estudio

Con esta priorización, los analistas de negocio han actualizado el modelo de negocio actual, de forma manual, para incorporar una actualización de una de las reglas, para incluir una nueva rama que permita al ciudadano solicitar la transmisión de un dominio siempre que se cumpla la condición de que dicho usuario sea el titular del mismo, y que éste se haya autenticado con certificado digital (nuevos elementos fact en el modelo). La Figura 24 muestra el modelo SBVR correspondiente a la solución objetivo.

4.3.3.6 Paso 6: Mapeo los modelos de negocio *as-is* y *to-be*

Una vez obtenidos ambos modelos de negocio, PREMISA propone la utilización de técnicas de gap analysis para mapear el sistema actual con los nuevos requisitos del negocio. Para ello, el primer paso consiste en la representación de los modelos mediante mecanismos formales que permita esta transición.

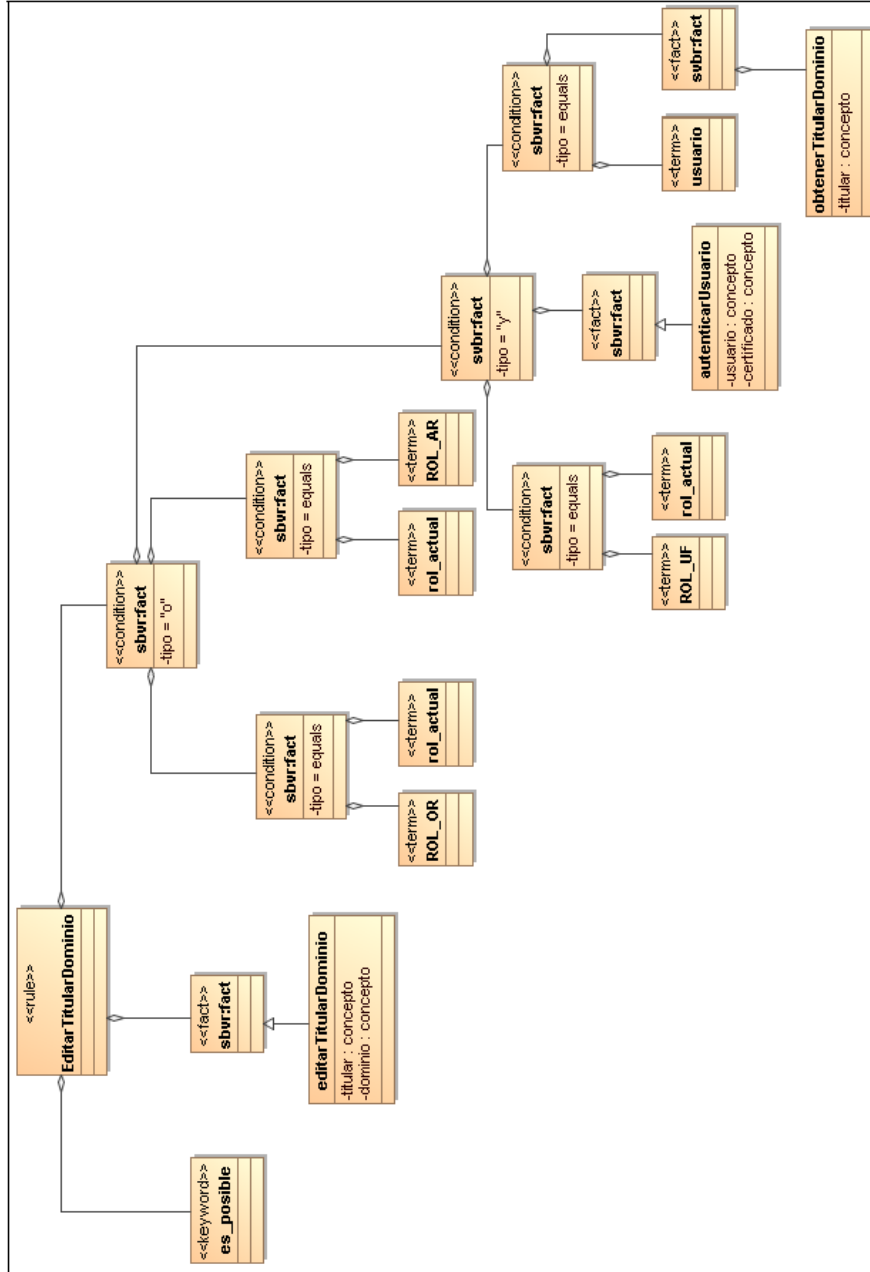


Figura 24. Modelo de negocio de la solución objetivo para transmisión de .es

La representación del modelo de negocio *as-is* de la herramienta de gestión de dominos “.es”, es la mostrada a continuación:

M_α	\subseteq	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact1}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact2}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{es posible}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact1}}$	\downarrow	$C_{\text{EditarTitular}}$
$\sigma_{\text{CSBVR:Fact2}}$	$=$	$i_{\text{tipo}}(\text{condicion_O})$
$BVR:Fact3$	$=$	$i_{\text{tipo}}(\text{condicion_equals})$
$\sigma_{\text{CSBVR:Fact4}}$	$=$	$i_{\text{tipo}}(\text{condicion_equals})$
$C_{\text{ROL OR}}$	\equiv	$C_{\text{SBVR:Fact3}}$
$ROL \text{ Actual}$	\equiv	$C_{\text{SBVR:Fact3}}$
$C_{\text{ROL AR}}$	\equiv	$C_{\text{SBVR:Fact4}}$
$C_{\text{ROL Actual}}$	\equiv	$C_{\text{SBVR:Fact4}}$
$\sigma_{\text{Crol or}}$	$=$	$o(\text{term})$
$\sigma_{\text{Crol actual}}$	$=$	$o(\text{term})$
$\sigma_{\text{Crol ar}}$	$=$	$o(\text{term})$
$\sigma_{\text{EditarTitularDominio}}$	$=$	$o(\text{term})$
$\sigma_{\text{Ces posible}}$	$=$	$O(\text{keyword})$
$\sigma_{\text{EditarTitular}}$	\subseteq	$i_{\text{titular}}(\text{concepto})$
$\sigma_{\text{EditarTitular}}$	\subseteq	$i_{\text{dominio}}(\text{concepto})$
$C_{\text{SBVR:Fact3}}$	\equiv	$C_{\text{SBVR:Fact2}}$
$C_{\text{SBVR:Fact4}}$	\equiv	$C_{\text{SBVR:Fact2}}$

Para el modelo de negocio *to-be* obtenemos la siguiente representación:

M_σ	\subseteq	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact1}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact2}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{es posible}}$	\equiv	$C_{\text{EditarTitularDominio}}$
$C_{\text{SBVR:Fact1}}$	\downarrow	$C_{\text{EditarTitular}}$
$\sigma_{\text{CSBVR:Fact2}}$	$=$	$i_{\text{tipo}}(\text{condicion_O})$
$BVR:Fact3$	$=$	$i_{\text{tipo}}(\text{condicion_equals})$

$\sigma_{CSBVR:Fact4}$	=	i_{tipo} (condicion_equals)
$C_{ROL OR}$	\subseteq	$C_{SBVR:Fact3}$
$ROL Actual$	\subseteq	$C_{SBVR:Fact3}$
$C_{ROL AR}$	\subseteq	$C_{SBVR:Fact4}$
$C_{ROL Actual}$	\subseteq	$C_{SBVR:Fact4}$
$\sigma_{Crol or}$	=	$o(term)$
$\sigma_{Crol actual}$	=	$o(term)$
$\sigma_{Crol ar}$	=	$o(term)$
$\sigma_{CEditarTitularDominio}$	=	$o(term)$
$\sigma_{Ces posible}$	=	$O(keyword)$
$\sigma_{CeditarTitular}$	\subseteq	$i_{titular}(concepto)$
$\sigma_{CeditarTitular}$	\subseteq	$i_{dominio}(concepto)$
$C_{SBVR:Fact3}$	\subseteq	$C_{SBVR:Fact2}$
$C_{SBVR:Fact4}$	\subseteq	$C_{SBVR:Fact2}$
$C_{sbvr:fact 6}$	\subseteq	$C_{sbvr:fact 5}$
$C_{sbvr:fact 7}$	\subseteq	$C_{sbvr:fact 5}$
$C_{sbvr:fact 8}$	\subseteq	$C_{sbvr:fact 5}$
$C_{rol actual}$	\subseteq	$C_{sbvr:fact 6}$
$C_{rol uf}$	\subseteq	$C_{sbvr:fact 6}$
$C_{usuario actual}$	\subseteq	$C_{sbvr:fact 8}$
$C_{sbvr:fact 9}$	\subseteq	$C_{sbvr:fact 8}$
$\sigma_{Crol uf}$	=	$o(term)$
$\sigma_{Crol or}$	=	$o(term)$
$\sigma_{Crol actual}$	=	$o(term)$
$\sigma_{Crol ar}$	=	$o(term)$
$\sigma_{CEditarTitularDominio}$	=	$o(term)$
$C_{sbvr:fact 5}$	=	$i_{tipo}(condicion_Y)$
$C_{sbvr:fact 6}$	=	$i_{tipo}(condicion_equals)$
$C_{sbvr:fact 8}$	=	$i_{tipo}(condicion_equals)$
$\sigma_{CobtenerTitularDominio}$	\subseteq	$i_{titular}(concepto)$
$\sigma_{CautenticarUsuario}$	\subseteq	$i_{usuario}(concepto)$
$\sigma_{CobtenerTitularDominio}$	\subseteq	$o(term)$
$\sigma_{CautenticarUsuario}$	\subseteq	$i_{certificado}(concepto)$

Una vez representados los modelos de negocio, el siguiente paso es aplicar los operadores intersección y disparity, para obtener las similitudes y diferencias entre ambos modelos. Para el caso de la aplicación de dominios “.es”, mostramos a continuación los resultados obtenidos.

a) Resultado de la operación *intersection* (M_α, M_β)

Este operador devolverá los elementos que son compatibles en ambos modelos

```

CeditarTitularDominio
  Csbvr:fact_1
  Ces_posible
  CeditarTitular
  Csbvr:fact_2
  Csbvr:fact_3
  Csbvr:fact_4
  Csbvr:fact_4
  Crol_or
  Crol_actual
  Crol_ar
    
```

b) Resultado de la operación *disparity* (M_α, M_β)

Esta operación devuelve los elementos de M_β que no están en M_α . El resultado de esta operación es el siguiente:

```

Csbvr:fact_5
Csbvr:fact_6
Csbvr:fact_7
Csbvr:fact_8
Csbvr:fact_9
CautenticarUsuario
CobtenerTitularDominio
Crol_uf
    
```

c) Resultado de la operación *disparity* (M_β, M_α)

Esta operación devuelve los elementos de M_α que no están en M_ω . En el caso de la aplicación de dominios, el resultado es vacío, ya que se han añadido elementos sobre el *as-is*, pero no se han eliminado.

Por lo tanto las operaciones que habrá que ejecutar para conseguir el modelo objetivo a partir del actual, serán las siguientes (se han obviado aquí las inserciones de las asociaciones correspondientes):

```

Insert( $M_\alpha$ ,  $C_{SBVR:Fact5}$ )
Insert( $M_\alpha$ ,  $C_{SBVR:Fact6}$ )
Insert( $M_\alpha$ ,  $C_{SBVR:Fact7}$ )
Insert( $M_\alpha$ ,  $C_{SBVR:Fact8}$ )
Insert( $M_\alpha$ ,  $C_{SBVR:Fact9}$ )
Insert( $M_\alpha$ ,  $C_{autenticarUsuario}$ )
Insert( $M_\alpha$ ,  $C_{obtenerTitularDominio}$ )
Insert( $M_\alpha$ ,  $C_{usuario\_actual}$ )
Insert( $M_\alpha$ ,  $C_{ROL\_UF}$ )

```

4.3.3.7 Paso 7: Obtención del KDM objetivo

Partiendo del modelo KDM de la solución objetivo, el siguiente paso consiste en aplicar las reglas de transformación, basándonos en las operaciones obtenidas en el paso anterior.

En el caso del sistema de gestión de dominios “.es”, únicamente existen operaciones de inserción, ya que el modelo de negocio únicamente ha añadido elementos con respecto a la solución actual.

La aplicación de estas reglas, ha dado lugar al modelo KDM correspondiente a la solución objetivo para la transmisión de dominios mostrado en la Figura 25 (en verde aparecen los nuevos elementos añadidos con respecto al KDM original):

4.3.3.8 Paso 8: Obtención de la solución objetivo (final)

Una vez obtenido el KDM correspondiente al modelo lógico de la solución objetivo, se tomará como base para actualizar el código fuente, en base a las reglas propuestas en PREMISA.

Mostramos a continuación un fragmento del código fuente obtenido para la solución objetivo del sistema de gestión de nombres de dominios “.es”.

```
public class DominioAction extends SgndBaseAction implements
ModelDriven<DominioFormView> {
...

    public String editarDominio(TDominio dominio, TContacto titular) throws Exception
    {

        if( getRol().equals(SgndConstantes.ROL_AR) ||
            getRol().equals(SgndConstantes.ROL_OR) ||
            (getRol().equals(SgndConstantes.ROL-UF) &&
            obtenerTitularDominio == = autenticarUsuario) {

            dominio.setTitular(titular);

        }

    }

}

public class DominioFormView extends SgndBaseFormView {
private TContacto titular;

    public void setTitular(TContacto titular) {
        this.titular = titular;
    }
}

public class TDominio implements Serializable {
    @Id
    @Column(name = "COD_DOMINIO")
    private Int codDominio;

    @Column(name = "NOMBRE_DOMINIO")
    private String nombreDominio;
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "COD_TITULAR")
    private TContacto tCodContacto;
```

```
}  
  
public class TContacto implements Serializable {  
    @Id  
    @Column(name="COD_CONTACTO")  
    private Int codContacto;  
    @Column(name="NOM_CONTACTO")  
    private String nombre;  
  
    @Column(name="NIC_HANDLE")  
    private String nicHandle;  
}
```

La principal dificultad encontrada a la hora de transformar el modelo lógico de la solución objetivo en el código fuente de la solución final, ha sido la falta de especificaciones técnicas para los nuevos elementos añadidos.

Por ejemplo, para el nuevo elemento de tipo *MethodUnit* que ha surgido como consecuencia de este proceso (autenticarUsuario), se tienen identificadas las entradas, salidas, los parámetros y una especificación a alto nivel de lo que deberá hacer este método (correspondiente a un *MethodUnit* en el KDM objetivo).

No obstante, el detalle de los tipos de datos para los nuevos atributos surgidos (usuario final), así como el detalle de la implementación del método no vienen determinadas desde el nivel de negocio, ni pueden ser descubiertas al no disponer de estos elementos en el modelo lógico origen. De este modo, la implementación de las instrucciones necesarias para desarrollar la funcionalidad de validar el usuario (por ejemplo, mediante invocaciones a servicios que se encarguen de validar si el certificado del usuario coincide con las credenciales presentadas y su validación) no podrán ser generadas automáticamente.

Sin embargo, este proceso sí ayuda al desarrollador a determinar la estructura de los métodos a utilizar, las condiciones a implementar, los atributos a manejar, identificar las relaciones con el resto de elementos en el código, etc. Una vez determinados estos aspectos, el desarrollador únicamente deberá preocuparse de la forma más eficiente de invocar la funcionalidad requerida.

Además la actualización de los elementos (métodos, atributos, etc.) existentes en la solución actual, sí puede ser realizada de forma más directa, al disponer de información adicional, y del proceso de evolución de la misma, determinado a lo largo del proceso de modernización.

4.3.3.9 Lecciones aprendidas de la segunda iteración

Como lecciones aprendidas de esta iteración podemos mencionar que, aunque se ha definido un conjunto de reglas de transformación para iterar de un modelo a otro, este proceso es muy propenso a errores si no se realiza automáticamente. Por lo tanto, nos dimos cuenta de que el desarrollo de un conjunto de herramientas de apoyo sería de gran utilidad.

Del mismo modo, el enfoque metodológico ofrecido por PREMISA está orientado al proceso más que al producto. Esto quiere decir que determinadas actividades, sobre todo derivadas de la documentación (por ejemplo, la gestión de los requisitos del sistema), no está contemplada en pasos concretos del proceso. Derivado de esto, en la segunda iteración, hemos encontrado diversos problemas para la actualización de los requisitos. Se optó por realizar una versión inicial de los mismos, una vez obtenido el modelo de negocio de la solución actual, que posteriormente fueron actualizados tras completar el modelo de negocio de la solución objetivo.

Otro de los problemas encontrados es esta iteración ha sido la falta de especificaciones técnicas a la hora de incluir nuevos elementos en los modelos de la solución objetivo. Al no disponer de información en los modelos de la solución actual, estas especificaciones (tipo de datos, invocaciones a servicios, etc.) han debido ser consensuadas entre los desarrolladores y los analistas de negocio, una vez obtenido el modelo de negocio de la solución objetivo. Para minimizar este impacto, se tratará en futuros trabajos de completar este marco metodológico.

4.4 Conclusiones de capítulo

Como conclusiones de la aplicación de PREMISA al caso de Red.es, tras dos iteraciones, es que hemos sido capaces de aplicarlo de forma satisfactoria a un caso real, además de establecer puntos de mejora para futuros trabajos.

La principal ventaja de la aplicación de este marco de modernización, es la institucionalización del proceso en Red.es. No sólo permite que cada rol sepa exactamente lo que tiene que hacer, sino que además se establecen los pasos necesarios para modernizar un sistema heredado, lo cual es muy útil para el resto de proyectos de modernización que Red.es tiene en cartera.

Como mejora adicional, el control de este proceso establece los mecanismos necesarios para la mejora continua.

*Capítulo 5. Conclusiones y
líneas futuras*

En este capítulo se presentan las conclusiones de este trabajo de Tesis Doctoral y las futuras líneas de investigación. En primer lugar, en la sección 5.1.1, se analiza la consecución de los objetivos planteados al inicio de este trabajo, y que han sido presentados en el capítulo 1. Posteriormente, en la sección 5.2, se indican las principales aportaciones de esta Tesis Doctoral. En la sección 5.3, se contrastan los resultados obtenidos a través de la presentación de las diferentes publicaciones realizadas. Finalmente, en la sección 5.4 se exponen las líneas de investigación que permanecen abiertas.

5.1 Análisis de la Consecución de Objetivos

En el capítulo 1 de esta memoria de Tesis Doctoral, se planteó el objetivo principal de este trabajo de investigación:

“Definir un proceso que combine técnicas de gap analysis y modernización dirigida por modelos, para la evolución de sistemas empresariales, centrado en el análisis de las reglas de negocio”.

Para la consecución de este objetivo, se plantearon un conjunto de objetivos parciales. A continuación, se pasa a analizar el resultado obtenido para cada uno de ellos:

- **Objetivo 1:** Estudiar, entender y analizar los principios de la modernización dirigida por modelos y del gap analysis, y cómo éstos pueden aplicarse a Red.es.

Se ha realizado un estudio sobre los principales trabajos relacionados con la modernización dirigida por modelos y el gap analysis. A través de este estudio se ha comprobado, como aspecto más importante, la ausencia de propuestas que utilicen ambos pilares de forma combinada. Además, si bien se han encontrado trabajos que abordan la modernización desde un punto de vista MDE, como norma general éstos son demasiado teóricos, o no plantean un ciclo completo de modernización, o no presentan herramientas que den soporte al proceso. Desde el punto de vista del gap analysis, los trabajos son escasos, y se abordan desde un punto de vista teórico o muy centrados en el negocio. Este estudio se ha presentado en las secciones 2.2 y 2.3 de esta memoria.

- **Objetivo 2:** Proponer un proceso para la modernización de sistemas, basado en la modernización dirigida por modelos y el gap analysis.

En esta Tesis Doctoral se ha propuesto un proceso basado en ADM y gap analysis, llamado PREMISA. Dicha propuesta, presentada en el capítulo 3, se centra en la definición de los modelos y las reglas de transformación para llegar a la solución objetivo partiendo de la solución actual.

- **Objetivo 3:** Validar el proceso en el caso real de Red.es

Como parte de esta Tesis Doctoral, se ha analizado la situación actual de Red.es, desde un punto de vista estratégico, para proponer un objetivo de modernización que ha sido aplicado mediante la aplicación del proceso definido. Las funcionalidades del sistema a modernizar han sido modeladas siguiendo también este proceso.

La definición y validación del método propuesto en esta Tesis Doctoral, que ha sido presentada en el capítulo 4, se ha ido realizando mediante un proceso iterativo e incremental basado en el método de investigación en acción. Para ello se han aplicado al caso real de Red.es.

5.2 Principales Aportaciones

El trabajo realizado ha supuesto un conjunto de aportaciones relativas al tema de investigación abordado en esta Tesis Doctoral, que se exponen a continuación:

- En cuanto a la orientación a servicios: PREMISA es proceso basado en modelos que cubre todos los pasos de un proceso de modernización en el que todos los artefactos involucrados se entiende desde un punto de vista orientado a servicios. De esta manera es posible utilizar un lenguaje común que permite la alineación de negocio y tecnología.
- En cuanto a los enfoques de modernización: PREMISA está fuertemente basada en el uso de técnicas basadas en modelos, no sólo por el uso de éstos en cualquier fase del proceso de modernización, sino también teniendo en cuenta la definición de reglas de transformación específicas en ese contexto. Después de estudiar en detalle el enfoque de ADM, se detectó que carecía de cualquier norma concreta para la transformación de modelos, como ya lo hace PREMISA.

- En cuanto a la definición de modelos de negocio: PREMISA define tanto un modelo de negocio específico, que no se incluye explícitamente como parte del formato original de ADM, y las reglas de transformación para obtener automáticamente dicho modelo (SBVR) partiendo del modelo lógico (KDM).
- En cuanto a la modificación (modernización) de las reglas de negocio: PREMISA propone seguir la modernización de las reglas de negocio a partir del uso del gap analysis para la conversión del *as-is* en los modelos *to-be*.
- Extensión de la notación utilizada en el gap analysis: Aparte de las contribuciones concretas anteriores, nos gustaría hacer hincapié en el hecho de que también hemos proporcionado con PREMISA una extensión a la notación utilizada en el gap analysis. Este aspecto se requiere para transformar los modelos *as-is* basados en SBVR a modelos KDM en a *to-be*.
- Aplicación de la propuesta a un caso real: A partir de la experiencia adquirida por el equipo de la modernización, durante el uso de este proceso hemos sido capaces de percibir una reducción de costes y tiempos de ejecución y una mejor alineación con la estrategia de la Entidad (debido a una mejor la comunicación entre los analistas de negocio y desarrolladores). Aunque no es una consecuencia directa de la utilización de PREMISA como marco de la modernización, ésta ha terminado en un aumento de la satisfacción del usuario, permitiendo la transmisión de los dominios "sin papeles", cuatro años antes de los hitos de la administración electrónica establecido por el Gobierno de España.

5.3 Contrastación de Resultados

Además de la aplicación a un caso real [29], algunas de las contribuciones realizadas como parte de este trabajo de investigación han sido contrastadas en diferentes foros:

5.3.1 Artículos en Congresos Internacionales:

- J. Moratalla, V. de Castro, M. López Sanz, E. Marcos. A Gap-Analysis-based Framework for Evolution and Modernization: Modernization of

Domain Management at Red.es. SRII Global Conference. San Jose, California 2012. Aceptado para publicación.

5.3.2 Artículos en Congresos Nacionales:

- J. Moratalla, V. de Castro, M. López Sanz, E. Marcos. Marco para la Modernización de Sistemas de Información basado en Modelos: Aplicación a un Caso Práctico (transmisión de dominios .es). Actas de las Jornadas Científico-Técnicas en Ingeniería de Servicios (JCIS). A Coruña 2011.

5.4 Líneas de Investigación Abiertas

Como continuación de este trabajo de Tesis Doctoral, existen diversas líneas de investigación que quedan abiertas y en las que es posible continuar trabajando. Algunas de ellas, están más directamente relacionadas con este trabajo de Tesis Doctoral y son el resultado de cuestiones que han ido surgiendo durante la realización del mismo. Otras, sin embargo, son líneas más generales y proceden de problemas que se conocían pero que, sin embargo, no son objeto de esta Tesis Doctoral; tales líneas están siendo abordadas por otros miembros del grupo de investigación Kybele y, en algunos casos, en colaboración con la Entidad Pública Empresarial Red.es:

- En primer lugar, una de las principales líneas de investigación abiertas está relacionada con el desarrollo de un conjunto de herramientas de apoyo a la propuesta. De esta manera será posible automatizar (o semiautomatizar) el proceso de modernización mediante la inclusión de soporte para la ejecución de los operadores de análisis de las diferencias en las reglas de negocio.
- Otra línea de continuación pasa por independizar el marco de trabajo de la plataforma, incorporando el modelo ASTM para la representación del código de la aplicación. De esta forma, el proceso podría ser aplicado a otras plataformas distintas de J2EE.
- También se propone la utilización del lenguaje natural para la representación del modelo SBVR, así como diseñar las herramientas necesarias para contemplar gramáticas que permitan la transformación automática entre ambas representaciones.

- Y por último, la incorporación de indicadores relevantes en el marco como parte de un SMM (Metamodelo de Métricas de semántica) que se utiliza para comprobar la calidad de los modelos obtenidos en un proceso de mejora continua.

Referencias

- [1] Abstract Syntax Tree Metamodel specification of the OMG, available at <http://www.omg.org/spec/ASTM/1.0/>
- [2] Acuña, C. J. (2007). PISA – Arquitectura de integración de portales Web: un enfoque dirigido por modelos y basado en servicios Web semánticos. Tesis Doctoral, Universidad Rey Juan Carlos, Madrid
- [3] Advanced Signature Framework (ASF). Disponible en https://asf.demo.red.es/asf_demo/
- [4] Architecture Driven Modernization specification of the OMG, available at <http://adm.omg.org/>
- [5] Armstrong, C., Underbakke, B.: MDA Legacy Modernization Case Study: State of Wisconsin Unemployment Insurance Division, Architecture-Driven Modernization. Accesible on: http://www.omg.org/news/meetings/workshops/adm_2004.htm, (2004).
- [6] Arsanjani, A.; Ghosh, S.; Allam, A.; Abdollah, T.; Ganapathy, S.; Holley, K. SOMA: a method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) (published online August 6, 2008).
- [7] Avison et al., 1999. Action Research. Avison, D., Lan, F., Myers, M. y Nielsen, A. Communications of the ACM, 42(1), pp. 94-97.
- [8] Backvanski, V.; Graff, P. “Domain Driven Modernization of Legacy Systems” Architecture Driven Modernization Workshop, Chicago, Marzo 2004. Disponible en http://www.omg.org/news/meetings/workshops/adm_2004.htm
- [9] Baxter, I.; “A Standards- Based Approach to Extracting Business Rules” Architecture Driven Modernization Workshop, Alexandria. Octubre 2005. Disponible en <http://www.omg.org/news/meetings/workshops/adm-2005.htm>
- [10] BOE 129 de 31 de mayo de 2005. Orden Ministerial ITC/1542/2005, de 19 de mayo, por la que se aprueba el Plan Nacional de Nombres de dominio de Internet bajo el Código correspondiente a España (".es")
- [11] BOE número 129 de 31/05/2005. Orden ITC/1542/2005, de 19 de mayo (Apartado Segundo, Disposición Primera)
- [12] BOE número 150 de 23/06/2007. Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos. Páginas 27150 a 27166

- [13] BOE número 166 de 12/07/2002. Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico (Disposición Adicional Sexta)
- [14] BOE número 99 de 25/4/1998. Ley 11/1998, de 24 de abril, General de Telecomunicaciones (Disposición Adicional Sexta). Páginas 13909 a 13940
- [15] Bollati V. A. (2011). MeTAGeM: un Entorno de Desarrollo de Transformaciones de Modelos Dirigido por Modelos. Tesis Doctoral, Universidad Rey Juan Carlos, Madrid
- [16] Bolstorff, P., Rosenbaum, R.: Supply Chain Excellence: A Handbook for Dramatic Improvement Using the Scor Model. Ed. AMACOM, 2nd edition (2007).
- [17] Bunge, M. (1979). La Investigación Científica. Barcelona: Ariel
- [18] Cánovas Izquierdo, J., García Molina, J. Extracción de Modelos en una Modernización basada en ADM. Actas de los talleres de las Jornadas de Ingeniería del Software y BBDD. Vol. 3, No. 2 (2009)
- [19] Cánovas, J; García Molina, J.. A Domain Specific Language for Extracting Models in Software Modernization". ECMDA-09
- [20] Chung, S., Byung, J. Service-Oriented Software Reengineering: SoSR. IEEE Proceedings of th 40th Hawaii International Conference on Systems Science (2007)
- [21] Clavreul, M.; Barais, O.; Jezequel, J.; "Integrating Legacy Systems with MDE". ACM Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, May 2010
- [22] Cornford y Smithson, 1996. Project Research in Information Systems: A Student's Guide Cornford, T., Smithson, S. Macmillan Press.
- [23] Cuadrado, J. S.; Molina, J. C.; Tortosa, M. M. "Rubytl: A practical, extensible transformation language" in ECMDA-FA, L. N. in Computer Science, vol. 4066/2006, pp. 158, 172 (2006).
- [24] De Castro, V. (2007). Aproximación MDA para el desarrollo orientado a servicios de sistemas de información web: del modelo de negocio al modelo de composición de servicios web. Tesis Doctoral, Universidad Rey Juan Carlos, Madrid
- [25] De Castro, V.; Marcos, E.; van den Heuvel, W.; Papazoglou, M. "Applying a Model-Driven Framework for Gap Analysis: Towards Business Service Engineering". Kybele Research Group & European Research Institute in Service Sciencie (ERISS)
- [26] Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Upper Saddle River: Prentice Hall PTR.

- [27] Favre, J. M. "Foundations of Model (driven) (Reverse) Engineering - Episode I: Story of the Fidus Papyrus and the Solarus", post-proceedings of Dagstuhl Seminar on Model Driven Approaches for Language Engineering, May 2004
- [28] French y Bell, 1996. Desarrollo organizacional (quinta edición). French, W.L., Bell, C.H. Jr. Editorial Prentice-Hall, Naucalpán de Juárez, México.
- [29] Instrucción del Director General de red.es, de 2 de enero de 2010, por la que se desarrollan los procedimientos aplicables a la asignación y a las demás operaciones asociadas al registro de nombres de dominio bajo el ".es". Disponible en <http://www.nic.es/media/2010-02/1265027277414.pdf>
- [30] Jeston, J.; Nelis, J. "Business Process Management: Practical Guidelines to Successful Implementations" Ed. Butterworth-Heinemann, 3rd edition (2006)
- [31] Keen, M., et al.: Patterns: Implementing an SOA using an Enterprise Service Bus. IBM Redbook, 22 July 2004
- [32] Knowledge Description Metamodel specification of the OMG, available at www.omg.org/technology/kdm/index.htm
- [33] Krafzig, D.; Banke K., Slama D. Enterprise SOA Service Oriented Architecture Best Practices. Upper Saddle River: Prentice Hall PTR. 2004
- [34] Kruchten, P. B. (1995) The 4+1 View Model of Architecture. IEEE Software, 12 (6), pp. 42 – 50.
- [35] López, M. (2011) ArchiMeDeS: A Service-Oriented Framework for Model-Driven Development of Software Architectures Tesis Doctoral, Universidad Rey Juan Carlos, Madrid
- [36] Lublinsky, B. "Defining SOA as an architectural style: Align your business model with technology" IBM Developer Works site. Octubre 2008. Disponible en <http://www.ibm.com/developerworks/webservices/library/arsostyle/index.html>
- [37] Malhotra, Yogesh. "Business Process Redesign: An Overview," IEEE Engineering Management Review, vol. 26, no. 3, Fall 1998
- [38] Mantell, K.: From UML to BPEL: MDA in a Web Services World, DeveloperWorks, available at: <http://www.ibm.com/developerworks/webseVICES/library/ws-uml2bpel/>
- [39] Marcos, E. and Marcos, A. (1998). An Aristotelian Approach to the Methodological Research: a Method for Data Models Construction. In: Information Systems- The Next Generation. Ed. L. Brooks y C. Kimble. Mc Graw-Hill, pp. 532-543
- [40] McTaggar, 1991. Principles of Participatory Action Research. McTaggart, R. Adult Education Quarterly 41(3).

- [41] MetaObject Facility specification of the OMG, available at <http://www.omg.org/mof/>
- [42] Model Driven Architecture specification of the OMG, available at <http://www.omg.org/mda/specs.htm>
- [43] MoDISCO user guide, available at <http://help.eclipse.org/helios/index.jsp?nav=/38>
- [44] Nakajima, S.; “Semi-automated diagnosis of FODA feature diagram”. Proceedings of the 2010 ACM Symposium on Applied Computing. Marzo 2010
- [45] OECD (2010) Informe “Buena Gobernanza en las Políticas Digitales: Cómo Maximizar el Potencial de las TIC. El Caso del Plan Avanza”. Disponible en <http://www.planavanza.es/InformacionGeneral/EvaluacionSeguimiento/Documents/Plan%20Avanza%20peer%20review%20draft%20Final%20Report%20July%2008.pdf>
- [46] OMG. “Business Process Modeling Notation (BPMN) Specification” The Object Management Group, febrero 2006. Disponible en http://www.bpmn.org/Documents/OMG_Final_Adopted_BPMN_1-0_Spec_06-02-01.pdf
- [47] Palmer, N.; Mooney, L. “Building a business case for BPM – a fast path to real result” White paper. Accesible on: <http://whitepaper.insurancetech.com/cmpinsurancetech/search/viewabstract/89568/index.jsp> (2007)
- [48] Papazoglou, M. P. and WJ van den Heuvel: Business Process Development Lifecycle Methodology. Communications of ACM, October 2007.
- [49] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.; “Service-Oriented Computing: A research roadmap”. INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS, ISSN 0218-8430, 06/2008, Volumen 17, Número 2, pp. 223 – 255
- [50] Pérez-Castillo et al., Business process archeology using MARBLE, Inform. Softw. Technol. Mayo 2011
- [51] Perfil del contrante de Red.es. Disponible en www.red.es
- [52] Petit, M.; Dubois, E.; Strengthening employee's responsibility to enhance governance of IT: COBIT RACI chart case study. WISG '09 Proceedings of the first ACM workshop on Information security governance. Nueva York 2009.
- [53] Plan Avanza 2. Disposición 6339 del BOE número 96 de 2010
- [54] Plan Avanza. Disposición 7239 del BOE número 110 de 2010

- [55] Pressman, S.; *Manifesto for agile software development*. McGraw-Hill, 2009
- [56] Pu, J, Yang, H., Xu, B., Xu, L., Cheng-Chung, W. Combining MDE and UML to Reverse Engineer Web Based Legacy Systems. Annual IEEE International Computer Software and Applications Conference (2008)
- [57] Rajala, M. "Minimizing the Risks of Architecture Driven Modernization Projects" Architecture Driven Modernization Workshop, Chicago, Marzo 2004. Disponible en http://www.omg.org/news/meetings/workshops/adm_2004.htm
- [58] Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica
- [59] RFC 1034 "Domain Names. Concepts and Facilities". Disponible en <http://tools.ietf.org/html/rfc1034>
- [60] RFC 1035 "Domain Names. Implementation and specification". Disponible en <http://tools.ietf.org/html/rfc1035>
- [61] Rosemann, M.. "Process Management as a Service". BPTrends, octubre 2010. Disponible en <http://www.bptrends.com/>
- [62] Sánchez, O., Sánchez J., García J.; "Model Driven Reverse Engineering of legacy graphical user interfaces". Proceedings of the IEEE/ACM international conference on Automated software engineering. Septiembre 2010
- [63] Semantic Business and Vocabulary Rules specification of the OMG, available at <http://www.omg.org/spec/SBVR/1.0/>
- [64] Semantic Metrics Metamodel specification of the OMG, available at <http://www.omg.org/spec/SMM/>
- [65] Sistema de Pago Telématico (SPT). Disponible en <http://pago.red.es/documentacion/516>
- [66] SOA: Principles of Service Design. Prentice Hall
- [67] Susman y Evered, 1978. An Assesment of the Scientific Merits of Action Research. Susman, G, Evered, R. Administrative Science Quarterly 23(4), pp. 582-603.
- [68] Ulrich, W. "A Model - driven Approach to Modernizing Existing Systems" Architecture Driven Modernization Workshop, Chicago, Marzo 2004. Disponible en http://www.omg.org/news/meetings/workshops/adm_2004.htm
- [69] Ulrich, W. "Aligning Modern Business Processes and Legacy Applications" MIT Press (2007)

- [70] Ulrich, W. "Architecture- Driven Modernization (ADM) Task Force: Overview, Scenarios & Roadmap". Architecture Driven Modernization Workshop, Alexandria. Octubre 2005. Disponible en <http://www.omg.org/news/meetings/workshops/adm-2005.htm>
- [71] Ulrich, W.; Newcomb, P. Information Systems Transformation: Architecture-Driven Modernization Case Studies. Morgan Kaufmann
- [72] UML 1.4 Profile for Automated Business Processes with a mapping to BPEL 1.0. Amsden, J; Gardner, T; Griffin, C.; Igengar S. Recuros IBM (2003), disponible en http://www.ibm.com/developerworks/rational/library/content/04April/3103/3103_UMLProfileForBusinessProcesses1.1.pdf
- [73] Van den Heuvel, W. "Matching and Adaptation: Core Techniques for MDA-(ADM)-driven Integration of new Business Applications with Wrapped Legacy Systems". Model Driven Evolution of Legacy systems. Septiembre 2004. Disponible en <http://www.cis.uab.edu/EDOC-MELS/Papers/Willem-Jan.pdf>
- [74] Van den Heuvel, W.J., *Aligning Modern Business Processes and Legacy Applications*, MIT Press, (2007).
- [75] Vara, J. M. (2009). M2DAT: a technical solution for Model-Driven development of Web Information Systems. PhD Thesis, Rey Juan Carlos University, Madrid
- [76] Vela B. (2003). MIDAS/BD: Una Metodología basada en Modelos para el Desarrollo de la Dimensión Estructural de Sistemas de Información Web. Tesis Doctoral, Universidad Rey Juan Carlos, Madrid
- [77] Von den Driesch, M; Blickle, T.; "Operational, Tool-Supportes Corporate Performance Management with ARIS Process Performance Manager". Springer, 2005
- [78] Wadsworth, 1998. What is Participatory Action Research?. Wadsworth, Y. Action Research International. Disponible en: <http://www.scu.edu.au/schools/gcm/ar/ari/pywadsworth 98.html>
- [79] Yang, J.; Papazoglou, M. "Web Component: A Substrate for Web Service Reuse and Composition". CAiSE 2002, pp. 21-36 (2000)
- [80] Yin, 2003. *Case Study Research Design and Methods*. Yin, R.K. Sage Publications. 3ª Edición.

Anexo I:Glosario

Término	Descripción
Reingeniería	Proceso mediante el cual se reorganiza y/o modifica un Sistema Software con el objetivo de optimizar el mismo, de tal manera que se incremente su facilidad de mantenimiento, reutilización y/o evolución. Durante este proceso se reestructura y reescriben partes o la totalidad de un Sistema pero sin cambiar su funcionalidad
Autoridad de Asignación o Registro	La entidad pública empresarial Red.es, entidad a la que se le ha encomendado la asignación de nombres de dominio bajo el código de país correspondiente a España (".es"), de acuerdo con lo dispuesto en la Disposición Adicional Sexta de la Ley 11/1998, de 24 de abril, General de Telecomunicaciones, en la Disposición Adicional Sexta de la Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico y en el apartado Segundo de la Disposición Primera de la Orden ITC/1542/2005, de 19 de mayo, por la que se aprueba el Plan Nacional de nombres de dominio de Internet bajo el código de país correspondiente a España (".es"), (en adelante, "el Plan de Nombres de Dominio").
Beneficiario o titular	La persona física o entidad con o sin personalidad jurídica a quien la Autoridad de Asignación confiere el derecho de uso y utilización de un nombre de dominio bajo ".es", a los efectos de direccionamiento en el Sistema de Nombres de Dominio de Internet en los términos señalados en el Plan de Nombres de Dominio
Persona de Contacto administrativo	La persona física que el beneficiario designa ante la Autoridad de Asignación como representante en relación con los derechos de uso del nombre de dominio bajo ".es", y que cuenta con poder de representación suficiente

Término	Descripción
Persona de Contacto de facturación	La persona física que el beneficiario designa ante la Autoridad de Asignación como contacto en relación con los aspectos relacionados con el pago de las cantidades correspondientes a las tasas de asignación y renovación del nombre de dominio bajo “.es”.
Persona de Contacto técnico	La persona física que el beneficiario designa ante la Autoridad de Asignación como contacto en relación con los aspectos técnicos relacionados con la configuración y operativa del nombre de dominio bajo “.es”.
Agente registrador	Entidad debidamente acreditada ante la Autoridad de Asignación, que, en nombre y por cuenta de sus usuarios, tramita las solicitudes de asignación y realiza la gestión de los nombres de dominio bajo “.es”, de acuerdo con lo establecido en el Plan Nacional de Nombres de Dominio, en el contrato de Agente Registrador, en la presente Instrucción y en el resto de disposiciones aplicables.
Asignación de nombre de dominio bajo “.es”	El procedimiento por medio del cual la Autoridad de Asignación asigna un nombre de dominio bajo “.es” a un titular en virtud de la normativa aplicable
Asignación provisional de nombre de dominio bajo “.es”	La asignación provisional se realizará en el momento en el que se solicite un nombre de dominio, siempre y cuando éste se encuentre disponible y la solicitud cuente con prioridad temporal. En tanto en cuanto la asignación provisional no se convierta en definitiva, el nombre de dominio no será visible en Internet

Término	Descripción
Asignación definitiva de nombre de dominio bajo “.es”	La asignación provisional se convierte en definitiva una vez se ha realizado el pago efectivo del nombre de dominio “.es”. El pago efectivo se deberá efectuar en un plazo máximo de CINCO días naturales. En el caso de que el pago no se realice, la asignación provisional será cancelada y el nombre de dominio estará disponible
Modificación directa de datos asociados a un nombre de dominios baja “.es”	La modificación de datos los recogidos a continuación se podrá realizar: (i) de forma directa por parte del usuario final en el caso de que el nombre de dominio sea gestionado directamente por la Autoridad de Asignación, (ii) por parte del Agente Registrador en el caso de que el nombre de dominio se gestionado por medio de Agente Registrador. El proceso será automático y no requerirá verificación previa
Modificación de datos con verificación previa, asociados a un nombre de dominios baja “.es”	La modificación de los datos que no sean susceptibles de modificación directa, únicamente podrá ser realizada por parte de la Autoridad de Asignación previa verificación
Baja de nombre de dominio bajo “.es”	La operación por medio de la cual se solicita a la Autoridad de Asignación la baja definitiva de un nombre de dominio bajo “.es”, quedando éste disponible, en su caso, para una posterior asignación
Transferencia de la gestión del nombre de dominio bajo “.es”	La operación por medio de la cual se acepta la transferencia de la gestión del nombre de dominio bajo “.es” a favor de un Agente Registrador, o por la que el nombre de dominio bajo “.es” pasa de ser gestionado por Agente Registrador a ser gestionado directamente por el beneficiario ante la Autoridad de Asignación.

Término	Descripción
Transmisión de un nombre de dominio bajo “.es”	La operación por medio de la cual la Autoridad de Asignación gestiona la transmisión de la titularidad de un nombre de dominio bajo “.es”, a los efectos de direccionamiento en Internet
Solicitud Telemática de Nombre de Dominio	El documento en soporte electrónico a través del cual se inicia el procedimiento de registro de un nombre de dominio
Formulario web de asignación	El formulario compuesto por los datos necesarios para el registro de nombres de dominio bajo “.es” accesible a través de la dirección web, http://www.dominios.es
Sistema de Nombres de Dominio de Internet	El sistema construido sobre una base de datos distribuida y jerárquica que permite acceder a una dirección IP mediante un nombre de dominio
Sistema de Registro de Nombres de Dominio bajo “.es”	El sistema informático software y hardware a través del cual se registran y gestionan los nombres de dominio bajo “.es”.
Nombre de Dominio de segundo nivel bajo “.es”	Dominios de extensión simple que terminen en .es, como por ejemplo red.es
Nombre de Dominio de tercer nivel bajo “.es”	Dominios con extensión compuesta que terminen en .es, como por ejemplo red.com.es, red.edu.es, red.gob.es, red.nom.es, red.org.es
Nombre de Dominio seguro bajo “.es”	Requiere aprobación expresa de Red.es, y en general son los dominios bajo extensión .edu.es y .gob.es
Nombre de Dominio no seguro bajo “.es”	Son los dominios cuya asignación no requiere aprobación de Red.es

Término	Descripción
Nombre de Dominio bloqueado	Nombre de dominio que no puede ser utilizado, al estar inmerso en algún trámite especial, en algún procedimiento judicial o en un procedimiento de resolución extrajudicial de conflictos (DRP)
Nombre de Dominio prohibido	Nombre de dominio que no puede ser utilizado, al contener términos considerados no válidos (palabras relacionadas con el terrorismo, etc.)
Nombre de Dominio reservado	Nombre de dominio que requiere una verificación previa por parte del Registro antes de su asignación (por ejemplo, topónimos, términos relacionados con la Casa Real o el Gobierno, etc.)

***Anexo II: Descomposición
funcional del sistema atual***

Mostramos a continuación los distintos diagramas de descomposición funcional por cada uno de los roles del sistema actual.

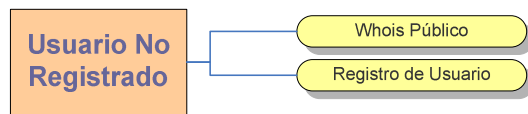


Figura A2- 1. Descomposición funcional para el usuario no registrado

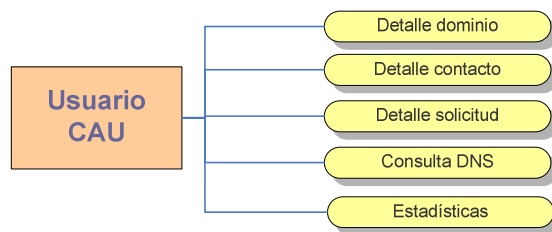


Figura A2- 2. Descomposición funcional para el usuario del CAU

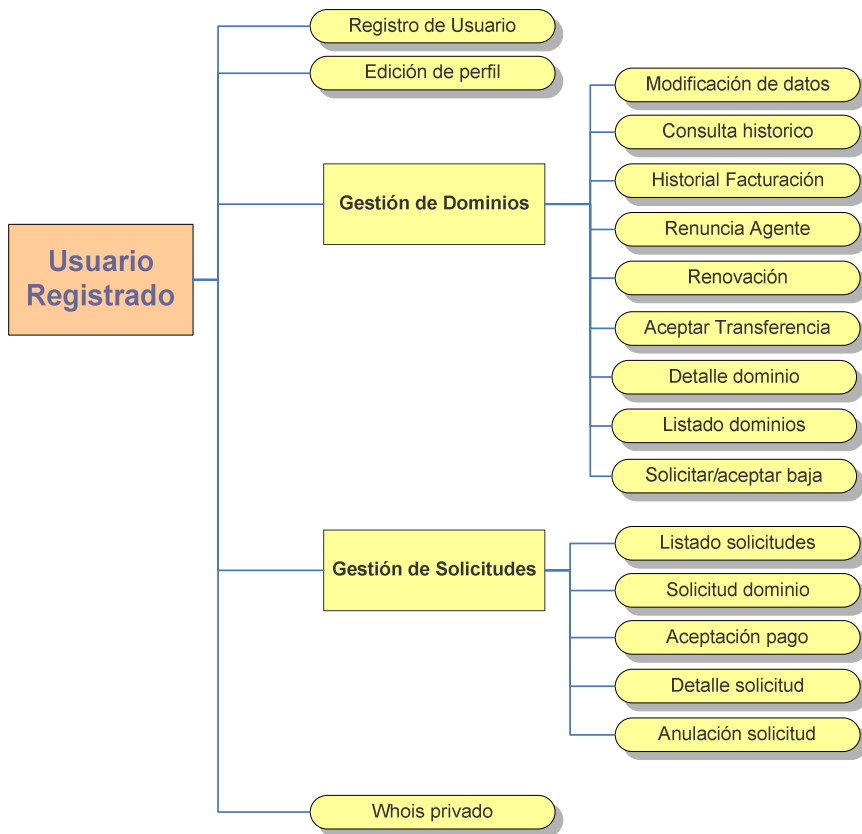


Figura A2- 3. Descomposición funcional para el usuario registrado

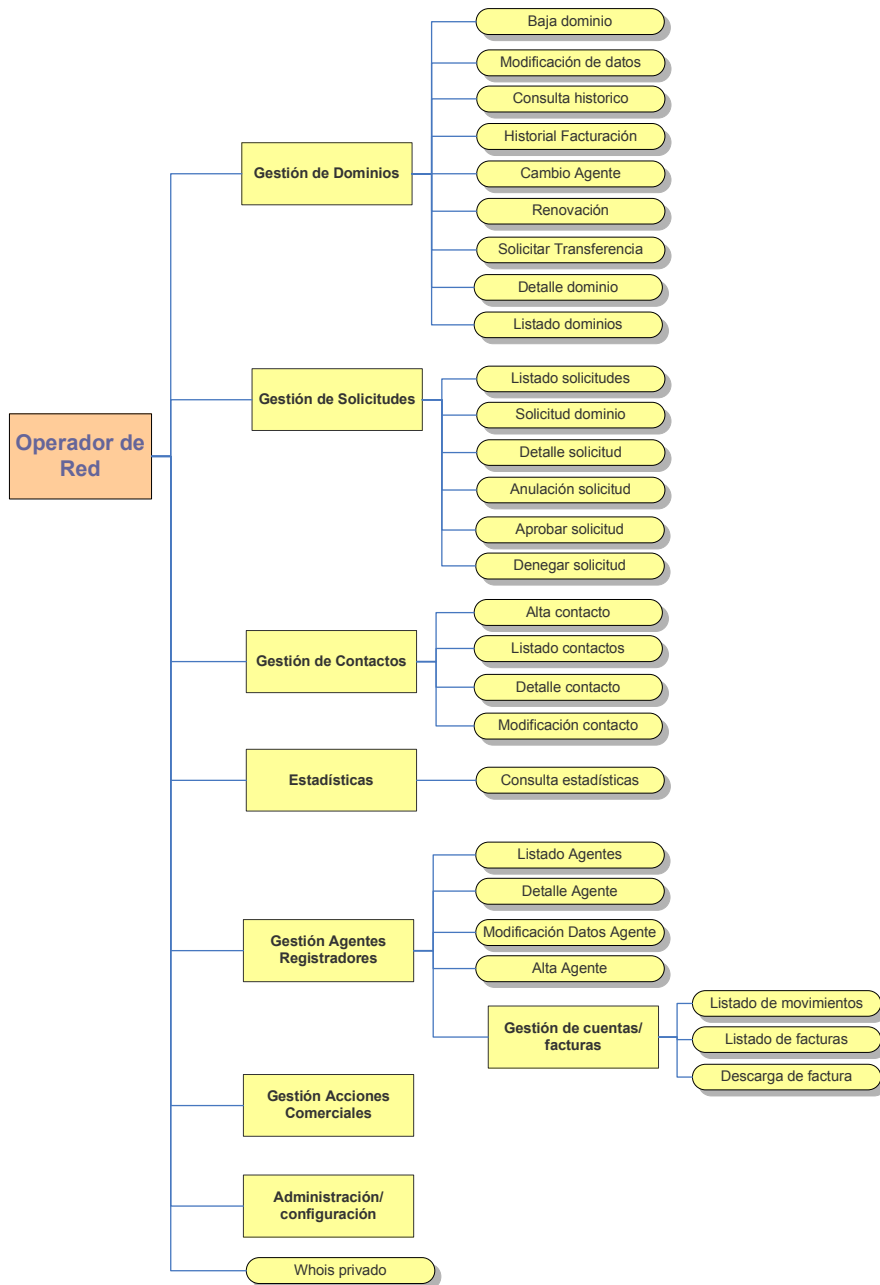


Figura A2- 4. Descomposición funcional para el operador de Red.es

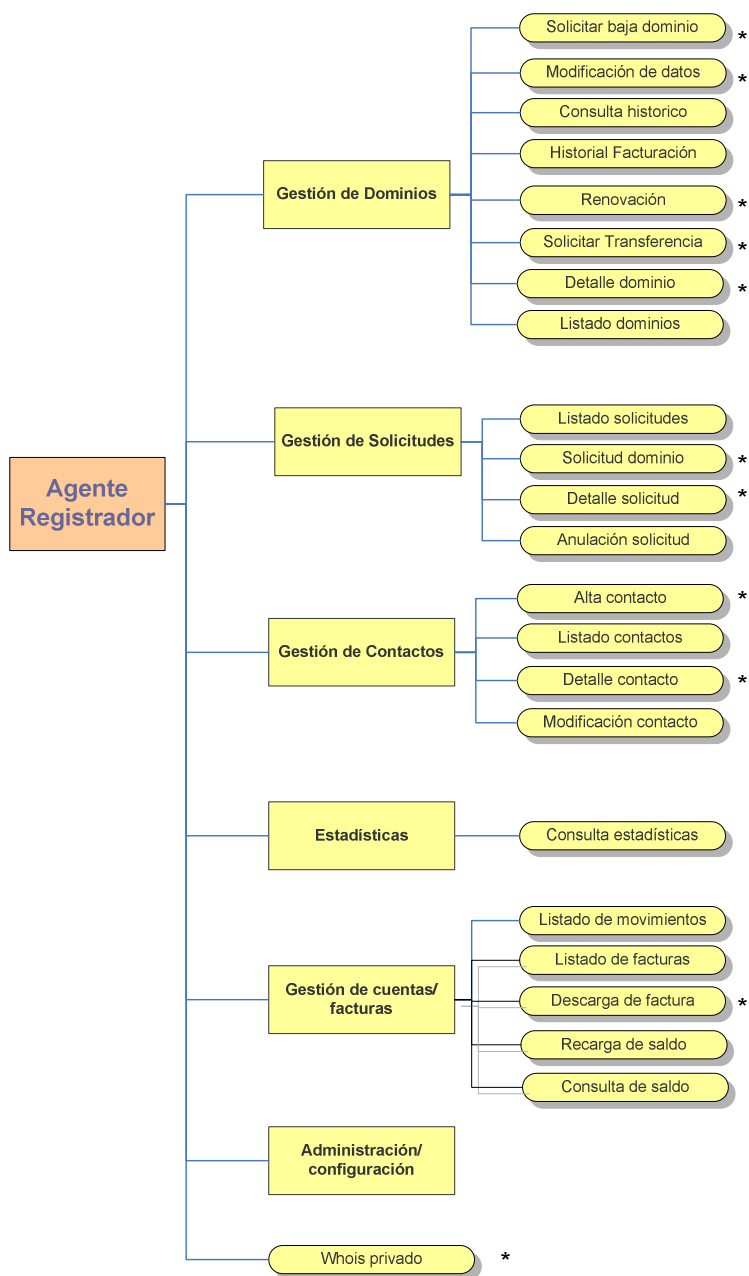


Figura A2- 5. Descomposición funcional para el agente registrador

***Anexo III: Especificación de
Casos de Uso***

A continuación se detallan los casos de uso en formato de alto nivel, del sistema origen.

Código	CU.001
Nombre	Acceso usuario/contraseña
Actores	UF, OR, CAU
Tipo	Primario, Real
Descripción	Autenticación de las credenciales de acceso del Usuario mediante usuario y contraseña
Precondiciones	El usuario debe acceder a la aplicación desde un navegador y estar registrador previamente en SGND mediante el CU.003
Postcondiciones	Permiso o denegación de acceso a la aplicación.

Código	CU.002
Nombre	Acceso certificado digital AR
Actores	AR
Tipo	Primario, Real
Descripción	Autenticación de las credenciales de acceso del Usuario mediante certificado digital.
Precondiciones	El Agente registrador debe acceder a la aplicación mediante un navegador con un certificado emitido por un organismo oficial, este debe estar instalado en el navegador.
Postcondiciones	Permiso o denegación de acceso a la aplicación.

Código	CU.003
Nombre	Alta Contaco.
Actores	UF
Tipo	Primario, Real
Descripción	Alta de contacto en aplicación SGND.
Precondiciones	El caso de uso se inicia cuando un Usuario accede a la aplicación y pincha en la opción de menú Registrarse en dominios “.es”. Los agentes registradores y los operadores de red podrán dar de alta contactos mediante la opción Contacto --> Alta contacto de su menú personal en la aplicación.
Postcondiciones	El usuario obtiene acceso a la aplicación mediante el usuario registrado.

Código	CU.004
Nombre	Consulta Whois, disponibilidad de dominio.
Actores	UF, AR, OR
Tipo	Primario, Real
Descripción	Consulta de dominio en Whois, mediante esta consulta el usuario comprueba la disponibilidad de un domino así como el estado y titulares en caso de estar ya activado.
Precondiciones	El caso de uso se inicia: <ul style="list-style-type: none"> · Un usuario público accede a la aplicación y pincha en el enlace Burcar dominio. · Un usuario registrado accede al alta dominio y se comprueba su disponibilidad antes de proceder a su registro
Postcondiciones	Estado del dominio ofreciendo la posibilidad de su registro.

Código	CU.005
Nombre	Alta dominio vía Web
Actores	UF, AR, OR
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la solicitud de alta de un dominio. Este proceso tiene finalidad adquirir los derechos de un dominio por un periodo de tiempo determinado.
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de alta dominio desde el menú de la aplicación de UF,AR,OR.
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede activado mediante la ejecución correcta de la solicitud o sea cancelada por cualquier de acción controlada.

Código	CU.006
Nombre	Validación pago solicitud de alta
Actores	Sistema
Tipo	Primario, Real
Descripción	Este caso reproduce la ejecución de un job programado de fondo que comprueba si el pago ha sido liquidado en las solicitudes en estado aprobado (APRB), el periodo máximo de tiempo (parametrizable) son 15 días, 10 para pagar y 5 para conciliar.
Precondiciones	La aplicación debe estar en ejecución y el job de validación de solicitudes de alta activo.
Postcondiciones	El caso de uso finalizará correctamente cuando la comprobación y posible cambio de estado sea efectuado.

Código	CU.007
Nombre	Proceso validación documentación Presentada solicitud alta (edu/gob/reservado)
Actores	Sistema
Tipo	Primario, Real
Descripción	Este caso reproduce la ejecución de un job programado de fondo que comprueba si una solicitud de alta en estado en trámite (ETRM) lleva más de 10 días(parametrizable) en este estado. En caso afirmativo el sistema procede a su cancelación.
Precondiciones	La aplicación debe estar en ejecución y el job de validación dedocumentación presentado de dominio edu/gob/reservado activo.
Postcondiciones	El caso de uso finalizará correctamente cuando la comprobación y posible cambio de estado sea efectuado.

Código	CU.008
Nombre	Baja dominio vía Web
Actores	UF, AR
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la solicitud de baja de un dominio. Este proceso tiene como finalidad rechazar los derechos de propiedad sobre el dominio.
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de alta dominio desde la extranet de UF o AR.
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede desactivado mediante la ejecución correcta de la solicitud o sea cancelada por cualquier de acción controlada.

Código	CU.009
Nombre	Renovación dominio
Actores	UF, AR, Sevidor correo, Pasarela pago
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la solicitud de renovación de un dominio. Este proceso tiene como finalidad ampliar los derechos de propiedad sobre el dominio.
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de renovación de dominio desde la extranet de UF o AR.
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede renovado mediante la ejecución correcta de la solicitud o sea cancelada por cualquier de acción controlada.

Código	CU.010
Nombre	JOB – Validación caducidad de dominio
Actores	Sistema
Tipo	Primario, Real
Descripción	Este caso reproduce la ejecución de un job programado de fondo que comprueba si el dominio está próximo a su fecha de caducidad
Precondiciones	El usuario que lanza el job debe estar registrado en la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la comprobación y posible operación sea realizada correctamente

Código	CU.011
Nombre	Transferencia dominio
Actores	UF, AR, Servidor de correo
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la solicitud de transferencia de un dominio. Este proceso tiene como finalidad cambiar el gestor del dominio
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de transferencia de dominio desde la extranet de AR.
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede transferido

Código	CU.012
Nombre	Anulación de solicitud de alta
Actores	principales: UF
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la cancelación de una solicitud de alta
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando el UF accede a su consola y seleccionada una solicitud de alta para anular.
Postcondiciones	El caso de uso finalizará correctamente cuando la solicitud de alta quede cancelada.

Código	CU.013
Nombre	Módulo Consultas
Actores	principales: OR, AR
Tipo	Primario, Real
Descripción	Este caso de uso reproduce la cancelación de una solicitud de alta
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando el AR/OR accede a su consola y escoge una búsqueda
Postcondiciones	El caso de uso finalizará correctamente cuando la búsqueda muestre por pantalla el listado deseado.

Código	CU.014
Nombre	Gestión de movimientos y Facturas agregadas de un agente registrador
Actores	AR, OR, Sistema
Tipo	Primario, Real
Descripción	Búsqueda y visualización de movimientos Facturas agregadas de un AR
Precondiciones	
Postcondiciones	El OR o AR debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando accede a su consola y escoge la opción: Movimientos: · Agentes/Movimientos de Agentes, en el caso del OR · Otros Servicios / gestión de Movimientos, en el caso del AR Facturas: · Agentes/ Facturas Agregadas de Agente, en el caso del OR · Otros Servicios / Gestión de Facturas, en el caso del AR.El caso de uso finalizará correctamente cuando la búsqueda muestre por pantalla el listado deseado.

Código	CU.015
Nombre	Mant. Lista negra
Actores	principales: OR
Tipo	Primario, Real
Descripción	Este caso de uso reproduce el mantenimiento de la lista negra, dominios que no pueden ser dados de alta.
Precondiciones	El operado debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando el OR accede a su consola y escoge la opción gestión de lista negra.
Postcondiciones	El caso de uso finalizará correctamente cuando el mantenimiento deseado finalice.

Código	CU.016
Nombre	Parametrización Whois
Actores	principales: OR
Tipo	Primario, Real
Descripción	Este caso de uso reproduce cuando el usuario OR configura los parámetros para el acceso a los datos mostrados por el Whois.
Precondiciones	El operado debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando el OR accede a su consola y escoge la opción gestión de Whois
Postcondiciones	El caso de uso finalizará correctamente cuando la parametrización del Whois finalice.

Código	CU.017
Nombre	Estadísticas de dominios dados de alta por el AR
Actores	AR, Sistema
Tipo	Primario, Real
Descripción	Posibilita ver el Histórico de gestión de dominio.
Precondiciones	.
Postcondiciones	<p>El AR debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede mediante el menú Otros servicios/estadísticas de la aplicación del AR.</p> <p>El caso de uso finalizará correctamente cuando se muestren la estadística por pantalla</p>

Código	CU.018
Nombre	Descargas de estadísticas predeterminadas
Actores	AR, OR, Sistema
Tipo	Primario, Real
Descripción	Se realiza una descarga de un fichero en formato Excel de los datos de la estadística que se seleccione.
Precondiciones	El OR o AR debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede mediante el menú Otros servicios/Descargas de estadísticas.
Postcondiciones	El caso de uso finalizará correctamente cuando el fichero sea descargado con éxito en la máquina cliente.

Código	CU.019
Nombre	Listado de solicitudes
Actores	UF, AR, OR, CAU, Sistema
Tipo	Primario, Real
Descripción	Posibilita realizar búsquedas y visualizar solicitudes.
Precondiciones	El AR, OR o el UF debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Solicitudes/Listado de Solicitudes de las distintas aplicaciones.
Postcondiciones	El caso de uso finalizará correctamente cuando se muestre la información de una solicitud por pantalla.

Código	CU.020
Nombre	Gestión de lista de reserva
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita visualizar la lista de reserva además de añadir, editar, borrar e importar dominios a la lista de reserva.
Precondiciones	.
Postcondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Gestión de Lista de Reserva de la aplicación del OR. El caso de uso finalizará correctamente cuando se visualice la Lista de Reserva.

Código	CU.021
Nombre	Gestión de dominios bloqueados
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita visualizar la lista de reserva además de añadir, editar, borrar e importar dominios a la lista de reserva.
Precondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Gestión de Dominios Bloqueados de la aplicación del OR.
Postcondiciones	El caso de uso finalizará correctamente cuando encuentre el dominio buscado.

Código	CU.022
Nombre	Gestión de lista de acciones comerciales
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita visualizar las asignaciones de agentes a Acciones Comerciales
Precondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Gestión de lista de Acciones Comerciales del OR.
Postcondiciones	El caso de uso finalizará correctamente cuando encuentre la Acción Comercial buscada

Código	CU.023
Nombre	Gestión de usuarios CAU, OR, AR
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita visualizar y gestionar los usuarios CAU y OR .
Precondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Gestión de Usuarios CAU y OR del OR
Postcondiciones	El caso de uso finalizará correctamente cuando se realice una búsqueda exitosa de un usuario.

Código	CU.024
Nombre	Modificación de dominio
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita gestionar un dominio
Precondiciones	.
Postcondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Dominio/Modificación de Dominio. El caso de uso finalizará correctamente cuando se realice una modificación exitosa de un dominio.

Código	CU.025
Nombre	Gestión de dominios
Actores	OR, AR, CAU, Sistema
Tipo	Primario, Real
Descripción	Posibilita gestionar un dominio
Precondiciones	El OR/AR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Dominio/Consulta de Dominios.
Postcondiciones	El caso de uso finalizará correctamente cuando se visualice el histórico del Gestor del Dominio.

Código	CU.026
Nombre	Gestión de contactos
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita gestionar un contacto
Precondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Dominio/Gestión de contactos.
Postcondiciones	El caso de uso finalizará correctamente cuando se realice una modificación exitosa de un contacto

Código	CU.027
Nombre	Consulta a Dominios.es
Actores	AR, Sistema
Tipo	Primario, Real

Descripción	Posibilita al AR enviar una consulta a Dominios.es
Precondiciones	El AR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Consulta a Dominios.
Postcondiciones	El caso de uso finalizará correctamente cuando se envíe exitosamente el formulario.
Código	CU.028
Nombre	Gestión de Referencias
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Posibilita gestionar una referencia
Precondiciones	El OR debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú Otros Servicios/Obtener Referencias.
Postcondiciones	El caso de uso finalizará correctamente cuando se consulte una referencia.
Código	CU.029
Nombre	Conciliador
Actores	Sistema, sistema externo (banco)
Tipo	Primario, Esencial
Descripción	Actualiza el estado de los dominios al comprobar el pago de los mismos
Precondiciones	El job "Conciliador" debe estar activo y funcionando
Postcondiciones	El caso de uso finalizará correctamente cuando se actualicen los estados de los dominios.

Código	CU.030
Nombre	Proceso DNSDump
Actores	Sistema Jobs, Sistema
Tipo	Primario, Esencial
Descripción	Proceso que genera el fichero DNSDump. Este fichero contiene las direcciones DNS de todos los dominios activos en la aplicación que no estén marcados como “phishing”. Este fichero proporcionará a los servidores de DNS de Red.es la información necesaria para resolver las IP destino de cada dominio “.es”
Precondiciones	La aplicación debe estar en ejecución y el job de generación del fichero activo, este job se ejecuta varias veces al día y lanzará el servicio DNSDump.
Postcondiciones	El caso de uso finalizará correctamente cuando la aplicación genere el fichero de DNSs.

Código	CU.031
Nombre	alta dominio EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de alta de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.032
Nombre	Anulación alta dominio EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de anulación de solicitud de alta de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.033
Nombre	Baja dominio EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de baja de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.034
Nombre	Transferencia dominio EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de transferencia de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.035
Nombre	Renovación dominios EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de renovación de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.036
Nombre	Gestión/Edición dominio EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de gestión y modificación de datos de dominio mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.037
Nombre	Gestión de contactos EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de gestión de contactos asociados a dominio (alta, modificación y baja) mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.038
Nombre	Bandeja de tareas EPP
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Este método proporciona el acceso directo a la funcionalidad de consultar la bandeja de tareas de un AR mediante EPP
Precondiciones	El agente registrador debe ejecutar el servicio EPP enviándole un fichero XML con la y parámetros de la aplicación.
Postcondiciones	El caso de uso finalizará correctamente cuando la operaciónseleccionada este completada y el servicio EPP devuelva la respuesta con el estado de la misma.

Código	CU.039
Nombre	Proceso Facturación
Actores	Sistema Jobs, Esencial
Tipo	Primario, Real
Descripción	Proceso que genera los ficheros de facturación. Estos ficheros contienen las facturas con los movimientos de los ARs del mes vencido
Precondiciones	Referencias
Postcondiciones	La aplicación debe estar en ejecución y el job de generación de facturas activo, este job se ejecutará el día 2 de cada mes El caso de uso finalizará correctamente cuando la aplicación los ficheros de facturación.

Código	CU.040
Nombre	Validación solicitudes de baja
Actores	Sistema
Tipo	Primario, Real
Descripción	Este caso reproduce la ejecución de un job programado de fondo que comprueba si existen solicitudes de baja aprobadas y ha transcurrido un periodo de 10 días (parametrizable). Este proceso ejecuta las bajas dejando el dominio desactivado.
Precondiciones	La aplicación debe estar en ejecución y el job de validaciones solicitudes de baja en ejecución.
Postcondiciones	El caso de uso finalizará correctamente cuando la comprobación y posible cambio de estado sea efectuado

Código	CU.041
Nombre	Validación solicitudes de baja
Actores	Sistema
Tipo	Primario, Real
Descripción	Este caso reproduce la ejecución de un job programado de fondo que comprueba si existen correos en cola. Este proceso se ejecuta dejando los correos enviados y almacenados en las tablas de histórico
Precondiciones	La aplicación debe estar en ejecución y el job de envío correos en cola en ejecución.
Postcondiciones	El caso de uso finalizará correctamente cuando la tabla de correos en cola quede completamente vacía y liberada.

Código	CU.042
Nombre	Histórico modificaciones de dominio
Actores	Sistema
Tipo	Primario, Real
Descripción	Esta funcionalidad permite a los perfiles de OR,AR y UF acceder a un listado donde muestra todas las modificaciones realizadas sobre un dominio determinado.
Precondiciones	El usuario debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú al link “Histórico de dominios”
Postcondiciones	El caso de uso finalizará correctamente cuando el sistema muestre al usuario un listado con un resumen de todas las Emodificaciones realizadas sobre el dominio solicitado.

Código	CU.043
Nombre	Alta Entidad Externa.
Actores	OR, Sistema
Tipo	Primario, Real
Descripción	Alta de una entidad externa en aplicación SGND.
Precondiciones	El caso de uso se inicia cuando un Usuario accede a la aplicación, identificándose como OR, y pincha en la opción de menú entidades externas.
Postcondiciones	Se crea una nueva entidad externa en el sistema

Código	CU.044
Nombre	Consulta WHOIS de entidades externas
Actores	Entidad Externa, Sistema
Tipo	Primario, Real
Descripción	Esta funcionalidad permite a un usuario autenticado como
Precondiciones	Entidad Externa realizar un WHOIS para un dominio determinado
Postcondiciones	<p>El usuario debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se pulsa sobre el de un dominio o sobre el linke “ver datos”.</p> <p>El caso de uso finalizará correctamente cuando el usuario acepte los términos legales y el sistema muestre al usuario el detalle del dominio solicitado.</p>

Código	CU.045
Nombre	Bloqueo de solicitud por un OR
Actores	Operador de Red, Sistema
Tipo	Primario, Real
Descripción	Esta funcionalidad permite a un usuario autenticado como Operador de Red bloquear una solicitud para que sólo sea modificada por él.
Precondiciones	El usuario debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se realiza una búsqueda de solicitudes.
Postcondiciones	El caso de uso finalizará correctamente cuando el sistema informe al usuario de que la solicitud ha sido bloqueada

Código	CU.046
Nombre	Desbloqueo de solicitud por un OR
Actores	Operador de Red, Sistema
Tipo	Primario, Real
Descripción	Esta funcionalidad permite a un usuario autenticado como
Precondiciones	Operador de Red desbloquear una solicitud para que pueda ser modificada por cualquiera.
Postcondiciones	<p>El usuario debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se realiza una búsqueda de solicitudes.</p> <p>El caso de uso finalizará correctamente cuando el sistema informe al usuario de que la solicitud ha sido desbloqueada</p>

Código	CU.047
Nombre	Histórico modificaciones de contactos
Actores	Sistema
Tipo	Primario, Real
Descripción	Esta funcionalidad permite a los perfiles de OR y AR acceder a un listado donde muestra todas las modificaciones realizadas sobre un contacto determinado.
Precondiciones	El usuario debe estar registrado y logado en su correspondiente aplicación. La ejecución de este caso comienza cuando se accede desde el menú al link "Histórico de contacto"
Postcondiciones	El caso de uso finalizará correctamente cuando el sistema muestre al usuario un listado con un resumen de todas las Emodificaciones realizadas sobre el contacto solicitado.

Código	CU.48
Nombre	Cancelación de dominio ordinaria
Actores	UF, AR, Jurídico, OR
Tipo	Primario, Real
Descripción	Este caso de uso permite a un usuario realizar una solicitud de demanda de cancelación o cancelación con reasignación de un nombre de dominio de otro titular
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de cancelación de dominio desde la extranet de UF.
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede resuelto.

Código	CU.49
Nombre	Transmisión ordinaria de dominio
Actores	UF, AR, OR
Tipo	Primario, real.
Descripción	Permite a un titular de un dominio traspasar el mismo a otro titular
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de transmisión de dominio desde la extranet de UF(AR
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede transmitido.

Código	CU.50
Nombre	Transmisión especial de dominio
Actores	UF, AR, Servidor de correo
Tipo	Primario, real.
Descripción	Permite a un titular de un dominio quedarse con la titularidad del mismo en casos especiales (transmisión intevivos)
Precondiciones	El usuario debe estar registrado y logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de transmisión de dominio desde la extranet de UF(AR
Postcondiciones	El caso de uso finalizará correctamente cuando el dominio quede transmitido.

Código	CU.51
Nombre	Procedimiento DRP
Actores	OR, Proveedor DRP, Demandante, Demandado, Servidor de correo
Tipo	Primario, real.
Descripción	Permite a un proveedor de arbitraje dirimir un conflicto entre dos titulares que reclaman el mismo dominio
Precondiciones	El usuario jurídico debe estar logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado de Inicio procedimiento DRP desde la extranet de Jurídico.
Postcondiciones	El caso de uso finalizará correctamente cuando el procedimiento DRP quede totalmente cerrado.

Código	CU.52
Nombre	Modificación Datos contacto mediante aprobación
Actores	OR, Usuario, Sevidor de correo
Tipo	Primario, real.
Descripción	Permite a un usuario solicitar el cambio del nombre y el nif de su perfil, para que el OR determine la autenticidad de sus datos antes de proceder a la aprobación del mismo
Precondiciones	El usuario debe estar logado en la aplicación. La ejecución de este caso comienza cuando se accede al apartado editar perfil desde la extranet de Usuario
Postcondiciones	El caso de uso finalizará correctamente cuando se haya efectuado la modificación del contacto.

Anexo IV. Elementos KDM

Este anexo detalla los elementos contemplados en KDM, descompuesta por capas, paquetes dentro de cada una de las capas, y clases implementadas entre de ellos (metamodelos contemplados en KDM). Se especifica además los metamodelos que son descubiertos, ya sea de forma completa o parcial, en MoDISCO.

Capa	Paquete	Clase	MoDISCO
Abstracción	Build	AbstracBuildRelationship	
Abstracción	Build	AbstractBuildElement	
Abstracción	Build	AbstractBuildRelationship	
Abstracción	Build	BuildComponent	
Abstracción	Build	BuildDescription	
Abstracción	Build	BuildElement	
Abstracción	Build	BuildModel	
Abstracción	Build	BuildProduct	
Abstracción	Build	BuildReslationship	
Abstracción	Build	BuildResource	
Abstracción	Build	BuildStep	
Abstracción	Build	Consumes	
Abstracción	Build	DescribedBy	
Abstracción	Build	Library	
Abstracción	Build	LinksTo	
Abstracción	Build	Produces	
Abstracción	Build	SuppliedBy	
Abstracción	Build	Supplier	
Abstracción	Build	SupportedBy	
Abstracción	Build	SymbolicLink	
Abstracción	Build	Tool	
Abstracción	Conceptual	AbstractConceptualElement	
Abstracción	Conceptual	AbstractConceptualRelationship	
Abstracción	Conceptual	BehaviorUnit	
Abstracción	Conceptual	ConceptualContainer	
Abstracción	Conceptual	ConceptualElement	
Abstracción	Conceptual	ConceptualFlow	
Abstracción	Conceptual	ConceptualModel	
Abstracción	Conceptual	ConceptualRelationship	
Abstracción	Conceptual	ConceptualRole	

Abstracción	Conceptual	FactUnit	
Abstracción	Conceptual	RuleUnit	
Abstracción	Conceptual	ScenarioUnit	
Abstracción	Conceptual	TermUnit	
Abstracción	Structure	AbstractStructureElement	
Abstracción	Structure	AbstractDataElement	
Abstracción	Structure	AbstractRelationship	
Abstracción	Structure	AbstractStructureElement	
Abstracción	Structure	AbstractStructureRelationship	
Abstracción	Structure	ArchitectureView	
Abstracción	Structure	Component	
Abstracción	Structure	DataModel	
Abstracción	Structure	Layer	
Abstracción	Structure	SoftwareSystem	
Abstracción	Structure	StructureElement	
Abstracción	Structure	StructureModel	
Abstracción	Structure	StructureRelationship	
Abstracción	Structure	Subsystem	
Elementos de Programa	Action	AbstractActionRelationship	x
Elementos de Programa	Action	ActionElement	x
Elementos de Programa	Action	ActionRelationship	x
Elementos de Programa	Action	Addresses	x
Elementos de Programa	Action	BlockUnit	x
Elementos de Programa	Action	Calls	x
Elementos de Programa	Action	CatchUnit	x
Elementos de Programa	Action	CompliesTo	x
Elementos de Programa	Action	ControlFlow	x
Elementos de Programa	Action	Creates	x
Elementos de Programa	Action	Dispatches	x
Elementos de Programa	Action	EntryFlow	x
Elementos de Programa	Action	ExceptionFlow	x
Elementos de Programa	Action	ExceptionUnit	x
Elementos de Programa	Action	ExitFlow	x
Elementos de Programa	Action	FalseFlow	x
Elementos de Programa	Action	FinallyUnit	x
Elementos de Programa	Action	Flow	x
Elementos de Programa	Action	GuardedFlow	x
Elementos de Programa	Action	Reads	x
Elementos de Programa	Action	Throws	x
Elementos de Programa	Action	TrueFlow	x

Elementos de Programa	Action	TryUnit	x
Elementos de Programa	Action	UsesType	x
Elementos de Programa	Action	Writes	x
Elementos de Programa	Code	AbstractCodeElement	x
Elementos de Programa	Code	AbstractCodeRelationship	x
Elementos de Programa	Code	ArrayType	x
Elementos de Programa	Code	BagType	x
Elementos de Programa	Code	BinstringType	x
Elementos de Programa	Code	BitType	x
Elementos de Programa	Code	BooleanType	x
Elementos de Programa	Code	CallableKind	
Elementos de Programa	Code	CallableUnit	x
Elementos de Programa	Code	ClassUnit	x
Elementos de Programa	Code	CodeAssembly	x
Elementos de Programa	Code	CodeElement	x
Elementos de Programa	Code	CodeItem	x
Elementos de Programa	Code	CodeModel	x
Elementos de Programa	Code	CodeRelationship	x
Elementos de Programa	Code	CommentUnit	x
Elementos de Programa	Code	CompilationUnit	x
Elementos de Programa	Code	ComputationalObject	x
Elementos de Programa	Code	ConditionalDirective	x
Elementos de Programa	Code	ControlElement	x
Elementos de Programa	Code	CharType	x
Elementos de Programa	Code	ChoiceType	x
Elementos de Programa	Code	DataElement	x
Elementos de Programa	Code	Datatype	x
Elementos de Programa	Code	DefinedType	x
Elementos de Programa	Code	DerivedType	x
Elementos de Programa	Code	EnumeratedType	x
Elementos de Programa	Code	Expands	x
Elementos de Programa	Code	ExportKind	
Elementos de Programa	Code	Extends	x
Elementos de Programa	Code	FloatType	x
Elementos de Programa	Code	GeneratedFrom	x
Elementos de Programa	Code	HasType	x
Elementos de Programa	Code	HasValue	x
Elementos de Programa	Code	ImplementationOf	x
Elementos de Programa	Code	Implements	x
Elementos de Programa	Code	Imports	x

Elementos de Programa	Code	IncludeDirective	x
Elementos de Programa	Code	Includes	x
Elementos de Programa	Code	IndexUnit	x
Elementos de Programa	Code	InstanceOf	x
Elementos de Programa	Code	IntegerType	x
Elementos de Programa	Code	InterfaceUnit	x
Elementos de Programa	Code	ItemUnit	x
Elementos de Programa	Code	LanguageUnit	x
Elementos de Programa	Code	MacroDirective	x
Elementos de Programa	Code	MacroKind	x
Elementos de Programa	Code	MacroUnit	x
Elementos de Programa	Code	MemberUnit	x
Elementos de Programa	Code	MethodKind	
Elementos de Programa	Code	MethodUnit	x
Elementos de Programa	Code	Module	
Elementos de Programa	Code	NamespaceUnit	x
Elementos de Programa	Code	OctectstringType	x
Elementos de Programa	Code	OrdinalType	x
Elementos de Programa	Code	Package	
Elementos de Programa	Code	ParameterKind	
Elementos de Programa	Code	ParameterTo	x
Elementos de Programa	Code	ParameterUnit	x
Elementos de Programa	Code	PointerType	x
Elementos de Programa	Code	PreprocessorDirective	x
Elementos de Programa	Code	PrimitiveType	x
Elementos de Programa	Code	RangeType	x
Elementos de Programa	Code	RecordType	x
Elementos de Programa	Code	Redefines	x
Elementos de Programa	Code	ScaledType	x
Elementos de Programa	Code	SequenceType	x
Elementos de Programa	Code	SetType	x
Elementos de Programa	Code	SharedUnit	x
Elementos de Programa	Code	Signature	x
Elementos de Programa	Code	StorableKind	
Elementos de Programa	Code	StorableUnit	x
Elementos de Programa	Code	SynonymType	x
Elementos de Programa	Code	TemplateParameter	x
Elementos de Programa	Code	TemplateType	x
Elementos de Programa	Code	TemplateUnit	x
Elementos de Programa	Code	TimeType	x

Elementos de Programa	Code	TypeUnit	x
Elementos de Programa	Code	Value	x
Elementos de Programa	Code	ValueElement	x
Elementos de Programa	Code	ValueList	x
Elementos de Programa	Code	VaritanTo	x
Elementos de Programa	Code	VisibleIn	x
Elementos de Programa	Code	VoidType	x
Infraestructura	Core	AggregatedRelationship	x
Infraestructura	Core	Boolean	
Infraestructura	Core	Element	x
Infraestructura	Core	Integer	
Infraestructura	Core	KDMElement	
Infraestructura	Core	KDMEntity	x
Infraestructura	Core	KDMRelationship	x
Infraestructura	Core	ModelElement	x
Infraestructura	Core	String	
Infraestructura	KDM	Annotation	x
Infraestructura	KDM	Attribute	x
Infraestructura	KDM	Audit	x
Infraestructura	KDM	ExtendedValue	x
Infraestructura	KDM	ExtensionFamily	x
Infraestructura	KDM	KDMFramework	x
Infraestructura	KDM	KDMModel	x
Infraestructura	KDM	Segment	x
Infraestructura	KDM	Stereotype	x
Infraestructura	KDM	TagDefinition	x
Infraestructura	KDM	TaggedRef	x
Infraestructura	KDM	TaggedValue	x
Infraestructura	Source	AbstractInventoryElement	x
Infraestructura	Source	AbstractInventoryRelationship	x
Infraestructura	Source	DependsOn	x
Infraestructura	Source	InventoryElement	x
Infraestructura	Source	InventoryModel	x
Infraestructura	Source	InventoryRelationship	x
Infraestructura	Source	SourceFile	x
Infraestructura	Source	SourceRef	x
Infraestructura	Source	SourceRegion	x
Recursos runtime	Data	AbstractContainerElement	
Recursos runtime	Data	AbstractContentElement	
Recursos runtime	Data	AbstractDataElement	

Recursos runtime	Data	AbstractDataRelationShip	
Recursos runtime	Data	AllContent	
Recursos runtime	Data	Catalog	
Recursos runtime	Data	CimpleContentType	
Recursos runtime	Data	ColumnSet	
Recursos runtime	Data	ColumnSet	
Recursos runtime	Data	ComplexContentType	
Recursos runtime	Data	ContentAttribute	
Recursos runtime	Data	ContentElement	
Recursos runtime	Data	ContentItem	
Recursos runtime	Data	ContentReference	
Recursos runtime	Data	ContentRestriction	
Recursos runtime	Data	ChoiceContent	
Recursos runtime	Data	DataAction	
Recursos runtime	Data	DataContainer	
Recursos runtime	Data	DataEvent	
Recursos runtime	Data	DataModel	
Recursos runtime	Data	DataRelationship	
Recursos runtime	Data	DataResource	
Recursos runtime	Data	DataSegment	
Recursos runtime	Data	DataTypeOf	
Recursos runtime	Data	ExtendedDataElement	
Recursos runtime	Data	ExtensionTo	
Recursos runtime	Data	GroupContent	
Recursos runtime	Data	HasContent	
Recursos runtime	Data	Index	
Recursos runtime	Data	IndexElement	
Recursos runtime	Data	KeyRelation	
Recursos runtime	Data	ManagesData	
Recursos runtime	Data	MixedContent	
Recursos runtime	Data	ProducesDataEvent	
Recursos runtime	Data	ReadsColumnSet	
Recursos runtime	Data	RecordFile	
Recursos runtime	Data	ReferenceKey	
Recursos runtime	Data	ReferenceTo	
Recursos runtime	Data	RelationalSchema	
Recursos runtime	Data	RelationalTable	
Recursos runtime	Data	RelationalView	
Recursos runtime	Data	RestrictionOf	
Recursos runtime	Data	SeqContent	

Recursos runtime	Data	TypedBy	
Recursos runtime	Data	UniqueKey	
Recursos runtime	Data	WritesColumnSet	
Recursos runtime	Data	XMLSchema	
Recursos runtime	Event	AbstractEventElement	
Recursos runtime	Event	AbstractEventRelationship	
Recursos runtime	Event	ConsumesEvent	
Recursos runtime	Event	Event	
Recursos runtime	Event	EventAction	
Recursos runtime	Event	EventElement	
Recursos runtime	Event	EventModel	
Recursos runtime	Event	EventRelationship	
Recursos runtime	Event	EventResource	
Recursos runtime	Event	HasState	
Recursos runtime	Event	InitialState	
Recursos runtime	Event	NextState	
Recursos runtime	Event	OnEntry	
Recursos runtime	Event	OnExit	
Recursos runtime	Event	ProducesEvent	
Recursos runtime	Event	ReadsState	
Recursos runtime	Event	State	
Recursos runtime	Event	Transition	
Recursos runtime	Platform	AbstractPlatformElement	
Recursos runtime	Platform	AbstractPlatformRelationship	
Recursos runtime	Platform	BindsTo	
Recursos runtime	Platform	DataManager	
Recursos runtime	Platform	DefinedBy	
Recursos runtime	Platform	DeployedComponent	
Recursos runtime	Platform	DeployedResource	
Recursos runtime	Platform	DeployedSoftwareSystem	
Recursos runtime	Platform	ExecutionResource	
Recursos runtime	Platform	ExternalActor	
Recursos runtime	Platform	FileResource	
Recursos runtime	Platform	Loads	
Recursos runtime	Platform	LockResource	
Recursos runtime	Platform	Machine	
Recursos runtime	Platform	ManagesResource	
Recursos runtime	Platform	MarshaledResource	
Recursos runtime	Platform	MessagingResource	
Recursos runtime	Platform	NamingResource	

Recursos runtime	Platform	PlatformAction	
Recursos runtime	Platform	PlatformElement	
Recursos runtime	Platform	PlatformEvent	
Recursos runtime	Platform	PlatformModel	
Recursos runtime	Platform	PlatformRelationship	
Recursos runtime	Platform	Process	
Recursos runtime	Platform	ReadsResource	
Recursos runtime	Platform	Requires	
Recursos runtime	Platform	ResourceType	
Recursos runtime	Platform	RuntimeResource	
Recursos runtime	Platform	Spawns	
Recursos runtime	Platform	StreamResource	
Recursos runtime	Platform	Thread	
Recursos runtime	Platform	WritesResource	
Recursos runtime	UI	AbstractElement	
Recursos runtime	UI	AbstractUIElement	
Recursos runtime	UI	AbstractUIRelationship	
Recursos runtime	UI	Displays	
Recursos runtime	UI	DisplaysImage	
Recursos runtime	UI	ManagesUI	
Recursos runtime	UI	ReadsUI	
Recursos runtime	UI	Report	
Recursos runtime	UI	Screen	
Recursos runtime	UI	UIAction	
Recursos runtime	UI	UIDisplay	
Recursos runtime	UI	UIElement	
Recursos runtime	UI	UIEvent	
Recursos runtime	UI	UIField	
Recursos runtime	UI	UIFlow	
Recursos runtime	UI	UILayout	
Recursos runtime	UI	UIModel	
Recursos runtime	UI	UIModel	
Recursos runtime	UI	UIRelationship	
Recursos runtime	UI	UIResource	
Recursos runtime	UI	WritesUI	

