



## ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO EN INFORMÁTICA

---

**APLICACIÓN WEB PARA LA VISUALIZACIÓN DEL  
CONOCIMIENTO DEL CENTRO AEROESPACIAL ALEMÁN  
(DLR)**

---

EFRAÍN LIMA MIRANDA

CÁDIZ, SEPTIEMBRE DE 2014



## ESCUELA SUPERIOR DE INGENIERÍA

### INGENIERO EN INFORMÁTICA

APLICACIÓN WEB PARA LA VISUALIZACIÓN DEL CONOCIMIENTO DEL  
CENTRO AEROESPACIAL ALEMÁN (DLR)

- Departamento: Lenguajes y Sistemas informáticos
- Director del proyecto: Juan Manuel Dodero
- Autor del proyecto: Efraín Lima Miranda

Cádiz, 15 de septiembre de 2014

Fdo: Efraín Lima Miranda

## ***Agradecimientos***

*Quisiera dar las gracias a todas las personas que, a pesar de la distancia, siempre han estado cerca. Sin su apoyo, su amistad, su paciencia y su confianza este trabajo me habría resultado imposible de realizar.*

# Índice general

<b>I Prolegómeno</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. El proyecto KnowledgeFinder . . . . .	3
1.2. Motivación . . . . .	3
1.2.1. El proyecto STRADA@DLR . . . . .	4
1.3. Alcance . . . . .	5
1.3.1. Mejora de la visualización de estructuras y relaciones complejas de los datos	5
1.3.2. Participación de nuevas fuentes de datos en el DLR relacionadas con el “Transporte y la Movilidad” . . . . .	5
1.4. Organización del documento . . . . .	6
<b>2. Planificación</b>	<b>7</b>
2.1. El Centro Aeroespacial Alemán (DLR) . . . . .	7
2.1.1. “Transporte y la Movilidad” en el DLR . . . . .	7
2.1.2. Equipamiento informático . . . . .	9
2.2. Metodología de desarrollo . . . . .	13
2.3. Planificación del proyecto . . . . .	14
2.4. Costes . . . . .	14
2.4.1. Estimación . . . . .	16
2.5. Riesgos . . . . .	17
2.5.1. Descripción de Riesgos . . . . .	17
2.5.2. Subsanación de los Riesgos . . . . .	19
2.6. Aseguramiento de calidad . . . . .	19
<b>II Desarrollo</b>	<b>20</b>
<b>3. Requisitos del Sistema</b>	<b>22</b>
3.1. Situación actual . . . . .	22
3.1.1. Procesos de Negocio . . . . .	22
3.1.2. Entorno Tecnológico . . . . .	23
3.1.3. Fortalezas y Debilidades . . . . .	26
3.2. Necesidades de Negocio . . . . .	28
3.2.1. Objetivos de Negocio . . . . .	29
3.2.2. Procesos de Negocio . . . . .	29
3.3. Obtención de los Objetivos del Sistema . . . . .	30
3.4. Objetivos del Sistema . . . . .	32

3.4.1. Importación y adaptación configurable . . . . .	32
3.4.2. Representación de los (Meta-)datos en un grafo . . . . .	32
3.5. Catálogo de Requisitos . . . . .	32
3.5.1. Requisitos de interfaces externas . . . . .	32
3.5.2. Requisitos funcionales . . . . .	34
3.5.3. Requisitos no funcionales . . . . .	35
3.5.4. Reglas de negocio . . . . .	37
3.5.5. Requisitos de información . . . . .	37
3.5.6. Restricciones técnicas del sistema . . . . .	37
3.6. Alternativas de Solución . . . . .	37
3.6.1. Elección de la tecnología de visualización . . . . .	37
3.6.2. Elección del índice de búsqueda . . . . .	41
3.6.3. Elección del tipo de aplicación . . . . .	42
3.7. Solución Propuesta . . . . .	43
3.7.1. Elección de la tecnología de visualización . . . . .	43
3.7.2. Elección del índice de búsqueda . . . . .	44
3.7.3. Elección del tipo de aplicación . . . . .	44
<b>4. Análisis del Sistema</b> . . . . .	<b>46</b>
4.1. Modelo Conceptual . . . . .	46
4.2. Modelo de Casos de Uso . . . . .	47
4.2.1. Diagramas de Casos de Uso . . . . .	47
4.2.2. Actores . . . . .	47
4.2.3. Especificación de los Casos de Uso . . . . .	47
4.3. Modelo de Comportamiento . . . . .	52
4.4. Modelo de Interfaz de Usuario . . . . .	57
<b>5. Diseño del Sistema</b> . . . . .	<b>59</b>
5.1. Arquitectura del Sistema . . . . .	59
5.1.1. Ámbito y Contexto del Sistema . . . . .	59
5.1.2. Arquitectura Física . . . . .	59
5.1.3. Arquitectura Lógica . . . . .	61
5.1.4. Modelo Vista Controlador en KnowledgeFinder II . . . . .	61
5.1.5. Desglose de Componentes . . . . .	61
5.2. Diseño Físico de Datos . . . . .	66
5.3. Vista de Procesos . . . . .	68
5.3.1. Iniciación de la Visualización . . . . .	68
5.3.2. Ejecución de Consulta . . . . .	68
5.3.3. Modificación del Grafo de Exploración . . . . .	69
5.3.4. Modificación de la Lista de Resultados . . . . .	70
5.4. Diseño detallado de la Interfaz de Usuario . . . . .	71
5.4.1. Pantalla Principal . . . . .	71
5.4.2. Pantalla de Detalles de Documento . . . . .	73

<b>6. Construcción del Sistema</b>	<b>75</b>
6.1. Entorno de Construcción . . . . .	75
6.2. Código Fuente . . . . .	76
6.2.1. Índice Solr . . . . .	76
6.2.2. Servicio Web . . . . .	78
6.2.3. Visualización . . . . .	79
<b>7. Pruebas del Sistema</b>	<b>85</b>
7.1. Estrategia . . . . .	85
7.2. Entorno de Pruebas . . . . .	85
7.3. Roles . . . . .	86
7.4. Niveles de Pruebas . . . . .	86
7.4.1. Pruebas Unitarias . . . . .	86
7.4.2. Pruebas de Integración . . . . .	86
7.4.3. Pruebas de Sistema . . . . .	86
7.4.4. Pruebas de Aceptación . . . . .	86
<b>III Epílogo</b>	<b>87</b>
<b>8. Manual de implantación y explotación</b>	<b>89</b>
8.1. Introducción . . . . .	89
8.2. Requisitos generales previos . . . . .	89
8.2.1. Java . . . . .	89
8.2.2. Eclipse . . . . .	90
8.2.3. Maven 3 . . . . .	90
8.2.4. Cliente SVN . . . . .	90
8.2.5. MantisBT y Jenkins . . . . .	90
8.3. Índice Solr . . . . .	91
8.3.1. Requisitos previos . . . . .	91
8.3.2. Procedimientos de instalación . . . . .	92
8.3.3. Pruebas de implantación . . . . .	95
8.3.4. Procedimientos de operación y nivel de servicio . . . . .	96
8.4. Servicio web en Liferay . . . . .	96
8.4.1. Requisitos previos . . . . .	96
8.4.2. Procedimientos de instalación . . . . .	97
8.4.3. Pruebas de implantación . . . . .	99
8.4.4. Procedimientos de operación y nivel de servicio . . . . .	100
8.5. Visualización . . . . .	100
8.5.1. Requisitos previos . . . . .	100
8.5.2. Procedimientos de instalación . . . . .	100
8.5.3. Pruebas de implantación . . . . .	102
8.5.4. Procedimientos de operación y nivel de servicio . . . . .	102
<b>9. Manual de usuario</b>	<b>103</b>
9.1. Introducción . . . . .	103
9.2. Características . . . . .	103
9.3. Requisitos previos . . . . .	104

---

9.4. Uso del sistema . . . . .	104
9.4.1. Grafo de Exploración . . . . .	104
9.4.2. Full-Text . . . . .	106
9.4.3. Menú de Navegación . . . . .	109
9.4.4. Lista de Resultados . . . . .	110
9.4.5. Selección actual . . . . .	112
<b>10. Conclusiones</b>	<b>114</b>
10.1. Objetivos alcanzados . . . . .	114
10.1.1. Visualización del Conocimiento . . . . .	114
10.1.2. Adaptación a nuevas Fuentes de Datos . . . . .	114
10.2. Mejoras Transversales de KnowledgeFinder II . . . . .	115
10.2.1. Mejora del Código . . . . .	115
10.2.2. Mejora del Rendimiento . . . . .	115
10.2.3. Desarrollo más flexible . . . . .	115
10.3. Lecciones aprendidas . . . . .	116
10.4. Trabajo futuro . . . . .	116
<b>Bibliografía</b>	<b>119</b>
<b>GNU Free Documentation License</b>	<b>120</b>
1. APPLICABILITY AND DEFINITIONS . . . . .	120
2. VERBATIM COPYING . . . . .	121
3. COPYING IN QUANTITY . . . . .	122
4. MODIFICATIONS . . . . .	122
5. COMBINING DOCUMENTS . . . . .	124
6. COLLECTIONS OF DOCUMENTS . . . . .	124
7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	124
8. TRANSLATION . . . . .	125
9. TERMINATION . . . . .	125
10. FUTURE REVISIONS OF THIS LICENSE . . . . .	125
11. RELICENSING . . . . .	126
ADDENDUM: How to use this License for your documents . . . . .	126

# Índice de figuras

1.1.	Logotipo MONITOR-Portal . . . . .	3
1.2.	Logotipo Clearingstellen Verkehr . . . . .	3
1.3.	Aproximación del Proyecto STRADA@DLR [fTR]	4
2.1.	Logotipo Centro Aeroespacial Alemán (DLR) . . . . .	7
2.2.	Los tres “bricks” o prototipos de <i>Extreme Prototyping</i> [Kom06] . . . . .	14
2.3.	Gráfica inicial de Gantt . . . . .	15
2.4.	Gráfica de Gantt de la replanificación . . . . .	15
3.1.	Modelo de proceso de negocio, importación de los datos en KnowledgeFinder . . . . .	22
3.2.	Modelo de proceso de negocio, búsqueda en el portal KnowledgeFinder . . . . .	23
3.3.	Estructura simplificada de Maven en KnowledgeFinder . . . . .	23
3.4.	(Meta-)datos de Clearingstellen Verkehr [fTR13] . . . . .	24
3.5.	(Meta-)datos de MONITOR-Portal [fTR13] . . . . .	25
3.6.	(Meta-)datos de STRADA@DLR [fTR13] . . . . .	25
3.7.	Las instancias del proyecto KnowledgeFinder en Eclipse . . . . .	27
3.8.	Interfaz de usuario para MONITOR-Portal . . . . .	28
3.9.	Modelo de proceso de negocio, importación de los datos en KnowledgeFinder II . . . . .	29
3.10.	Modelo de proceso de negocio, búsqueda en el portal KnowledgeFinder II . . . . .	30
3.11.	Chord Diagram . . . . .	30
3.12.	Bubble Chart . . . . .	31
3.13.	Circle Packing . . . . .	31
3.14.	Grafo - Force Layout . . . . .	32
3.15.	Exploración de una red social de marketing Sigma.js [LAB] . . . . .	38
3.16.	Exploración de la red de blogs Tumblr con Sigma.js [LAB] . . . . .	39
3.17.	Ejemplos visuales uso de Cytoscape [Con] . . . . .	39
3.18.	Exploración del DOM después de la vinculación de datos [euS12] . . . . .	40
3.19.	Ejemplo de integración de Leaflet con D3.js usando mapas de GeoJSON [Bos11] . . . . .	41
3.20.	Ejemplo de funcionamiento de Crossfilter [Sou] . . . . .	41
3.21.	Ejemplo de grafo en D3.js publicado en <a href="http://bl.ocks.org/mbostock/950642">http://bl.ocks.org/mbostock/950642</a>	42
4.1.	Modelo Conceptual de KnowledgeFinder II . . . . .	46
4.2.	Diagrama Casos de Uso: Uso de la interfaz de usuario (anónimo) y usuario registrado	48
4.3.	Diagrama Casos de Uso: Gestión del índice de búsqueda . . . . .	48
4.4.	Modelo de Comportamiento Caso de Uso: Importación de datos . . . . .	52
4.5.	Modelo de Comportamiento aso de Uso: Buscar en Full-Text . . . . .	53
4.6.	Modelo de Comportamiento Caso de Uso: Buscar (Meta-)dato . . . . .	53

4.7. Modelo de Comportamiento Caso de Uso: Detalles de documento . . . . .	54
4.8. Modelo de Comportamiento Caso de Uso: Eliminar filtro . . . . .	54
4.9. Modelo de Comportamiento Caso de Uso: Cambiar lista de resultados . . . . .	55
4.10. Modelo de Comportamiento Caso de Uso: Resaltar (Meta-)dato . . . . .	55
4.11. Modelo de Comportamiento Caso de Uso: Resaltar (Meta-)datos de documento . . . . .	56
4.12. Modelo de Comportamiento Caso de Uso: Mostrar grupo en grafo . . . . .	57
4.13. Mockup - pantalla general KnowledgeFinder II . . . . .	58
4.14. Mockup - pantalla de detalles en KnowledgeFinder II . . . . .	58
 5.1. Diagrama del ámbito y contexto de la arquitectura del sistema . . . . .	59
5.2. Diagrama de despliegue para el proyecto KnowledgeFinder II . . . . .	60
5.3. Nivel 1 - Building View KnowledgeFinder II . . . . .	62
5.4. Nivel 2 - Building View Índice Solr . . . . .	63
5.5. Nivel 2 - Building View Servicio web . . . . .	64
5.6. Nivel 2 - Building View Visualización . . . . .	65
5.7. Diagrama de Secuencia - Iniciación de la Visualización . . . . .	68
5.8. Diagrama de Secuencia - Ejecución de Consulta . . . . .	69
5.9. Diagrama de Secuencia - Modificación del Grafo de Exploración . . . . .	70
5.10. Diagrama de Secuencia - Modificación de la Lista de Resultados . . . . .	71
5.11. Pantalla principal . . . . .	72
5.12. Pantalla detalles de documento . . . . .	74
 9.1. Pantalla principal del STRADA@DLR usando KnowledgeFinder II . . . . .	105
9.2. Grafo de exploración de STRADA@DLR usando KnowledgeFinder II . . . . .	105
9.3. Campo Full-Text de STRADA@DLR usando KnowledgeFinder II . . . . .	106
9.4. Menú de navegación de STRADA@DLR usando KnowledgeFinder II . . . . .	110
9.5. Lista de Resultados . . . . .	111
9.6. Resaltado de Full-Text en la lista de resultados en STRADA@DLR usando KnowledgeFinder II . . . . .	112
9.7. Resaltado de Full-Text en la ventana emergente en STRADA@DLR usando KnowledgeFinder II . . . . .	112
9.8. Selección actual de STRADA@DLR usando KnowledgeFinder II . . . . .	113

# Índice de tablas

2.1. Software usado para la gestión . . . . .	9
2.2. Software usado como entorno de desarrollo . . . . .	10
2.3. Software usado para desarrollar el programa . . . . .	10
2.4. Software usado durante la fase de testing . . . . .	10
2.5. Software usado para el despliegue de la aplicación . . . . .	11
2.6. Software usado para la generación de la documentación . . . . .	11
2.7. Herramientas usadas para la comunicación y coordinación . . . . .	12
2.8. Hardware del equipo usado para el desarrollo . . . . .	12
2.9. Hardware del servidor para la aplicación de pruebas . . . . .	12
2.10. Resumen del gasto de personal . . . . .	16
2.11. Resumen de la estimación de los costes . . . . .	17
2.12. Tabla valoración impacto de riesgos . . . . .	17
2.13. Tabla probabilidad riesgos . . . . .	18
3.1. Tabla resumen de las transformaciones para los rangos de fechas . . . . .	37
5.1. Tabla resumen del esquema de KnowledgeFinder II de Solr para el proyecto STRA- DA@DLR . . . . .	67
6.1. Plugins usados en Eclipse durante el desarrollo. . . . .	76
6.2. Formato de los elementos de configuración para el menú . . . . .	80
6.3. Herramientas de JavaScript usadas . . . . .	82

# Lista de códigos

1.	Ejemplo de importación de datos en KnowledgeFinder . . . . .	26
2.	Ejemplo de vinculación de datos con el DOM usando D3.js [euS12] . . . . .	40
3.	Ejemplo de importación de datos en KnowledgeFinder II para STRADA@DLR . . . . .	77
4.	Ejemplo de esquema del índice Solr en KnowledgeFinder II para STRADA@DLR . . . . .	78
5.	Ejemplo de configuración del servicio web en KnowledgeFinder II para STRADA@DLR . . . . .	79
6.	Ejemplo de configuración del menú en KnowledgeFinder II para STRADA@DLR . . . . .	81
7.	Ejemplo de paleta de colores en KnowledgeFinder II para STRADA@DLR . . . . .	83
8.	Ejemplo de aplicación de la paleta de colores en KnowledgeFinder II para STRADA@DLR . . . . .	84
9.	Ejemplo de ejecución inicial de Solr en Jetty . . . . .	91
10.	Construcción con Maven de DataImport . . . . .	92
11.	Construcción con Maven de índice de búsqueda . . . . .	92
12.	Ejemplo de ejecución de Solr en Jetty con el índice de STRADA@DLR . . . . .	93
13.	Configuración del índice desde copia de trabajo local de SVN . . . . .	93
14.	Configuración del índice desde repositorio remoto de SVN . . . . .	94
15.	Ejemplo de servicio para Solr en una máquina de Debian (etcinit.desolr) . . . . .	95
16.	Ejemplo de ejecución inicial de Liferay en Tomcat . . . . .	97
17.	Construcción con Maven del servicio web . . . . .	97
18.	Extracto del fichero de configuración de Maven para auto-despliegue con Liferay . . . . .	98
19.	Auto-deploying con Maven del servicio web . . . . .	98
20.	Ejemplo de configuración del servicio web para usar el índice de Solr . . . . .	98
21.	Ejemplo de servicio para Liferay en una máquina de Debian . . . . .	99
22.	Construcción con Maven de la visualización . . . . .	100
23.	Extracto del fichero de configuración de Maven para auto-deploying con Liferay . . . . .	101
24.	Auto-deploying con Maven del servicio web . . . . .	101
25.	Configuración para la comunicación portlet de visualización con el servicio web . . . . .	102
26.	Ejemplo - Simple search . . . . .	106
27.	Ejemplo - Matched phrase search . . . . .	106
28.	Ejemplo - Disjunctive search . . . . .	107
29.	Ejemplo - Conjunctive search . . . . .	107
30.	Ejemplo - Exclusive search . . . . .	107
31.	Ejemplo 1 - Wirdcard search . . . . .	107
32.	Ejemplo 2 - Wirdcard search . . . . .	108
33.	Ejemplo 3 - Wirdcard search . . . . .	108
34.	Ejemplo - Fuzzy search . . . . .	108
35.	Ejemplo - Field search . . . . .	108

36. Ejemplo búsqueda combinada 1 . . . . .	109
37. Ejemplo búsqueda combinada 2 . . . . .	109

# Acrónimos

**AJAX** Asynchronous JavaScript and XML. 43, 45, *Glosario:* AJAX

**API** Application Programming Interface. 38, 44, 45, 61–64, 79, 100, 117, *Glosario:* API

**BMWi** Bundesministerium für Wirtschaft und Energie. 7, *Glosario:* BMWi

**DOM** Document Object Model. 40

**DRL** Deutsches Zentrum für Luft- und Raumfahrt e. V.. 3–5, 7–9, 11, 12, 14, 16, 19, 47, 75, 89, 103, 114–116, *Glosario:* DLR

**FK** Institut für Verkehrssystemtechnik. 5, *Glosario:* FK

**FW** Institut für Flughafenwesen und Luftverkehr. 3, 8, 9, 13, 14, 43, 86, 89, 103, 116, *Glosario:* FW

**GWT** Google Web Toolkit. *Glosario:* GWT

**HTML** HyperText Markup Language first. 13, 19, 64, 65, 117

**Http** Hypertext Transfer Protocol. 42

**IDE** Entorno de desarrollo Integrado. 10, 75, 76, 90, *Glosario:* IDE

**Java SDK** Java SE Development Kit. 23, 75, 89, 90, *Glosario:* Java SDK

**JMX** Java Management Extensions. 42

**JSP** JavaServer Pages. 64, 68

**MVC** Modelo Vista Controlador. 43–45, 61, 115

**RDF** Resource Description Framework. 117

**RIA** Rich Internet application. 42, 44, *Glosario:* RIA

**SC-VSS** Verteilte Systeme und Komponentensoftware. 3, 9, 10, 13, 14, 16, 19, 37, 41, 43, 44, 75, 86, 90, 116, *Glosario:* SC-VSS

**SVN** Subversion. 19, 22, 26, 29, 32, 33, 60–63, 67, 75, 76, 86, 90, 92–94, 97, 100, 115, 117,  
*Glosario:* SVN

**TS** Institut für Fahrzeugkonzepte. 5, *Glosario:* TS

**UML** Unified Modeling Language. 6, 11, 22, 23, 29, 46, *Glosario:* UML

**URL** Uniform resource locator. 35, 64, 68, 82, 100, 104, *Glosario:* URL

**VF** Institut für Verkehrsforschung. 3, 8, 9, 13, 14, 43, 86, 89, 103, 116, *Glosario:* VF

**XML** Extensible Markup Language. 18, 35, 42, 44, 45, 96, 117, *Glosario:* XML

# Glosario

**Full-Text** Técnica de búsqueda para encontrar en un documento o en un conjunto de ellos examinando todas las palabras almacenadas e intentando que corresponda con los criterios proporcionados. En situaciones donde el número de documentos es alto, la búsqueda Full-Text debe ser precedido de una indexación. Durante la indexación se crea una lista con los términos de los documentos aplicándose la búsqueda sobre ésta de forma más eficiente. 5, 33–35, 42, 44, 49, 53, 66–68, 73, 103, 104, 106, 111, 112, 114

**Maven** Herramienta de software para la gestión y construcción de proyectos en Java. Su característica clave es la descarga dinámica desde su repositorio de proyectos Open Source en diferentes versiones. 23, 75, 76, 90, 92, 97, 98, 100, 101

<http://maven.apache.org/>

**BMWi** Ministerio de Economía y Energía Alemán. 7

<http://www.bmwi.de/>

**Bootstrap** Framework para frontend creado por Twitter. 82

<http://getbootstrap.com/>

**Bug tracker** Software diseñado para el seguimiento de errores de productos software. De esta forma se ayuda a asegurar la calidad de éstos y asistir en el seguimiento de los defectos del producto a las personas involucradas. 9

**Canvas** Elemento de HTML5 que permite la generación dinámica de gráficos a través de la ejecución de código. Estos gráficos pueden ser estáticos y con animaciones. 38

**Chrome** Navegador web ligero implementado por Google. 102, 104

<http://www.google.com/chrome>

**Chromium** Versión Open Source de Chrome. 102, 104

<http://chromium.org/>

**Compass** Herramienta Open Source de ayuda para trabajar con Sass. 82, 116

<http://compass-style.org/>

**Cron** Administrador de procesos en segundo plano de sistemas UNIX. Permite ejecutar estos procesos en intervalos regulares de tiempo. 26, 47

**Crossfilter** Biblioteca Open Source escrita en JavaScript para explorar conjuntos de datos multivariados de gran tamaño en un navegador web.. 40, 41, 117

<http://square.github.io/crossfilter/>

**CSS** Hoja de estilo en cascada es usado como lenguaje para definir la presentación de un documento estructurado en HTML. 39, 82, 116

**Cytoscape** Plataforma para la visualización y análisis de redes complejas. 38, 39

<http://www.cytoscape.org/>

**D3.js (Data-Driven Documents)** Biblioteca JavaScript para la manipulación de documentos basándose en datos. 39–45, 82, 116

<http://d3js.org/>

**Deploying** Conjunto de actividades realizadas sobre un producto software para hacer este disponible a los usuarios. 10, 98, 101

**Despliegue** . 11, 60, 75, 85, 89, 92, 97, 98, 101, *Ver:*

**DLR** Centro Aeroespacial Alemán. 89, 103

<http://www.dlr.de>

**Extreme Prototyping** Prototipado extremo [Kom06]. 13, 14

**Eclipse** Software Open Source multiplataforma escrita el Java para el desarrollo de aplicaciones. Aunque en sus orígenes sólo aceptaba Java como lenguaje de programación, gracias a sus gran número de plugins, en la actualidad puede considerarse como la interfaz de programación más versátil. 10, 11, 27, 75, 76, 90, 92, 97, 98, 100, 101

**ELIB-Portal** . 25–27, 44, 116

<http://elib.dlr.de/>

**Experiencia de Usuario** Conjunto de factores y elementos relacionados con la interacción del usuario con un sistema obteniendo una percepción positiva o negativa del mismo. Depende de factores relacionados con el diseño y con las emociones percibidas por el usuario. 5, 13, 114, 117

**Firewall** Parte de un sistema o red diseñado para filtrar el acceso. Por lo tanto, permite para bloquear el acceso no autorizado y permitir las comunicaciones autorizadas. 60

**FK** Instituto para las Técnicas de Transporte. 5

<http://www.dlr.de/fk/>

**Flash** Software propietario utilizado tradicionalmente para la generación de animaciones. Estas pueden ser visualizadas en un navegador web o a través de un reproductor de Flash. Actualmente su utilización ha decaído considerablemente por sus fallos de seguridad y la aparición de nuevas tecnologías como HTML5. 37

---

**Framework** Es una estructura conceptual y tecnológica que sirven como base para la organización y desarrollo de software. 3, 5, 8, 10, 13, 22, 24–26, 28, 29, 37, 42, 44, 45, 66, 71, 75, 86, 89, 103

**FW** Instituto para la Investigación del Trasporte Aéreo y el Sistema Aeroportuario. 3, 89, 103  
<http://www.dlr.de/fw/>

**Gantt** Gráfico de barras, creado por Henry Gantt, que ilustra la planificación temporal de un proyecto. 6, 11, 14, 116

**Hardware** El hardware de un sistema informático lo componen todas sus partes físicas. Sus componentes son eléctricos, electrónicos, electromecánicos y mecánicos. 12, 16, 18, 59, 60

**HTML5** Quinta versión del lenguaje de marcado HTML. Sus principales mejoras son el tratamiento de objetos multimedia y elementos de interacción gráfica. 37, 39, 82, 104, 116

**Integración continua** Consiste en la compilación y ejecución de pruebas de forma automática de un proyecto lo más a menudo posible para detectar de esta forma fallos. 9

**Integrated Development Environment (IDE)** El entorno de desarrollo integrado es un software compuesto por un conjuntos de herramientas de programación. Proveen al desarrollador un marco de trabajo amigable para el proceso de desarrollo. 10

**Interfaz de usuario** Es el medio por el que el usuario puede conectarse con una máquina, equipo o comutadora. Se compone de todos los puntos de contacto del usuario y el equipo. En una interfaz gráfica de software, por ejemplo, esta comunicación se produce a través de elementos gráficos. 14, 32–34, 36, 47, 48, 50, 51, 63, 64, 68, 71, 82, 89, 102, 104, 114, 115

**Internet Explorer** Navegador web de Microsoft. 104

<http://windows.microsoft.com/en-us/internet-explorer>

**Java** Lenguaje de programación orientado a objetos publicado por Sun Microsystems en 1995 y adquirido por Oracle en 2009. 10, 11, 19, 37, 44, 62, 75, 76, 78, 115, 116

<https://www.oracle.com/java/>

**Java SE Development Kit** Software que provee herramientas para el desarrollo de aplicaciones Java. 23

**JavaScript** Lenguaje de programación interpretado orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en el lado del cliente para mejorar la interfaz de usuario y para la generación de comportamiento dinámico de las páginas webs. 13, 19, 37–39, 44, 45, 64, 68, 75, 79, 82, 102, 104, 115, 116

**JavaScript Object Notation (JSON)** Formato ligero para el intercambio de datos. Creado como subconjunto de la notación literal de objetos en JavaScript, se ha convertido en una alternativa a XML para AJAX . 10, 33, 42, 45, 63, 80, 86, 96, 117

<http://json.org/>

---

**Jenkins** Software Open Source para la integración continua en el servidor. 19, 75, 76, 85, 90, 91  
<http://jenkins-ci.org/>

**Jetty** Servidor opensource de HTTP implementado en Java y contenedor de servlets. 11, 60, 75, 91–93  
<http://jetty.mortbay.org/jetty/index.html>

**JRebel** Software para el redespliegue automático de cambios en un servidor. 75, 76  
<http://zeroturnaround.com/software/jrebel/>

**KnowledgeFinder II** Segunda versión del framework. 5, 6, 13, 17, 28–30, 32, 33, 37, 41, 43–47, 58–62, 66, 67, 71, 75, 77–79, 81, 83–86, 89, 91, 100, 101, 103, 105, 106, 110, 112–116

**KnowledgeFinder** Primera versión del framework. 3–5, 8, 13, 22–24, 26–28, 41–44, 114, 115

**LATeX** Sistema para la composición de textos usado especialmente para la edición de documentos científicos y técnicos. 11  
<http://latex-project.org/>

**Leaflet** Biblioteca Open Source escrita en JavaScript para la interacción con mapas. 40, 41, 117  
<http://leafletjs.com/>

**Liferay** Portal de gestión de contenidos Open Source escrito en Java. Incluye un gestor de contenido web permitiendo la construcción de portales y páginas web simplemente conjuntando temas, páginas, portlets y una navegación conjunta. 11, 23, 25, 37, 43, 45, 60, 61, 75, 76, 78, 89, 96–102, 115, 117  
<http://www.liferay.com>

**Logging** Grabación secuencial de los acontecimientos que afectan a un producto. Organizados normalmente por orden cronológico, permiten analizar la actividad interna paso a paso y sus interacciones con el medio. 10, 96, 100, 102

**Lucene** API Open Source para la recuperación de información implementada en Java. Usado extensamente en la implementación de motores de búsquedas, indexado de documentos y búsquedas Full-Text. 24, 26, 41, 44

<http://lucene.apache.org/>

**(Meta-)dato** Se consideran datos que ayudan a describir a otros datos. De esta forma, a través de los (Meta-)datos se facilita la labor de clasificar y encontrar los datos que son descritos por ellos. 4, 5, 24, 25, 29, 32–35, 49–51, 53, 55, 56, 64, 65, 72, 89, 103, 104, 106, 109, 111, 114

**MONITOR-Portal** . 8, 24–28, 116, 117  
<http://monitorportal.dlr.de/>

**MantisBT** Bug tracker bajo licencia Open Source desarrollado en PHP. 19, 75, 76, 90  
<https://www.mantisbt.org/>

**Mensajería instantánea** También conocido como chat, es una forma de comunicación en tiempo real entre dos o más personas basada en texto. 12

**Mocking** Objeto simulado que imita el comportamiento de objetos reales de forma controlada. Utilizado durante el proceso de testing, suelen ser usados para pruebas unitarias de software. 10

**Motor de búsqueda** es un sistema de recuperación de información diseñado para encontrar información almacenada en un sistema informático. Los resultados, *hits*, suelen mostrarse en una lista. Los motores de búsqueda ayudan a minimizar el tiempo necesario para encontrar la información deseada. 11, 60, 61, 114

**Mozilla Firefox** Navegador web de la fundación Mozilla. 104

<https://www.mozilla.org/es-ES/firefox>

**NoSQL** Sistema de almacenaje y acceso a datos modelado por cualquier otro sistema que no sea relacional. No utiliza como lenguaje SQL como lenguaje principal de consultas. 37

**Open Source** Término con el que se conoce al software distribuido y desarrollado libremente. Suele referirse al acceso libre del código fuente del software. 38, 39

<http://opensource.org/>

**Paginación** División de una lista o documento en diferentes partes llamadas páginas y la numeración de las mismas. 34, 65, 70, 73, 110

**Plugin** Complemento de una aplicación que añade funcionalidad nueva, generalmente muy específica, a otro producto de software. 10, 11, 40, 42, 43, 75, 76, 90, 97, 98, 101

**Portlet** Componentes de la interfaz de usuario gestionadas y visualizadas en un portal web. 23, 37, 60, 68, 79, 89, 100–102

**Prototipo** En el ciclo del software, ejemplar parcial que intenta simular y muestra algunas propiedades del producto final. 13, 14

**Research Fellow** Miembro investigador de una Universidad o Institución. La posición de “Research Fellow” requiere normalmente un doctorado, o trabajo equivalente por ejemplo en la industria. 9, 16, 17

**Clearingstellen Verkehr** . 24, 117

<http://daten.clearingstelle-verkehr.de/>

**STRADA@DLR** Search TRAsport DAta. 3, 4, 6, 22, 24–27, 44, 66, 67, 76–84, 89, 93, 96, 100, 103–106, 110, 112, 113, 115–117

**Safari** Navegador web desarrollado por Apple. 104

<https://www.apple.com/safari/>

**Sass** Extensión de lenguaje de estilos CSS. 79, 82, 115, 116

<http://sass-lang.com/>

**SC-VSS** Sistemas Distribuidos y Componentes de Software. 3

<http://www.dlr.de/sc/>

**Scroll** Desplazamiento en 2D de los contenidos que conforman elemento de la interfaz de usuario, por ejemplo una lista o una ventana del navegador web. 33, 36, 72, 80, 82, 109

**Servicio web** Tecnología que utiliza un conjunto de protocolos y estándares relacionados con la web para intercambiar datos entre aplicaciones software. 10, 45, 60, 61, 63–65, 68–70, 78, 79, 85, 86, 89, 96–102, 116, 117

**Sigma.js** Biblioteca gráfica de JavaScript para dibujar gráficos. 38, 39

<http://sigmajs.org/>

**Software** El software lo compone el equipamiento lógico de un sistema informático necesarios para la realización de tareas específicas. En Ingeniería de Software se denomina también “producto”. 4, 6, 9–11, 16–19, 26, 38, 40, 43, 44, 60, 61, 75, 82, 85, 92, 97, 100, 116

**Solr** Motor de búsqueda Open Source implementado en Java que se ejecuta sobre algún contenedor de servlets Java. Basado en Lucene, ofrece Application Programming Interface (API)s en XML y JSON, resultado de texto en resultados, búsqueda por facetas, caché y una interfaz de administración. 11, 41, 44, 47, 60–64, 66, 67, 75–79, 85, 86, 89, 91–93, 95, 96, 98, 114–116

<http://lucene.apache.org/solr>

**Solrj** Cliente Java para acceder a Solr. Ofrece una interfaz para añadir, actualizar y consultar índices de Solr. 44

<http://wiki.apache.org/solr/Solrj>

**SVG** Gráficos Vectoriales Redimensionables tanto estáticos como animados en formato XML. 39, 40, 43

<http://www.w3.org/Graphics/SVG/>

**Testing** Actividad perteneciente al proceso de calidad del software. La componen investigaciones empíricas y técnicas de las que se extrae información objetiva e independiente sobre la calidad de un producto software. 10, 19, 85, 116

**Tomcat** Servidor web con soporte de servlets y JSPs escrito en Java. 11, 23, 60, 75, 96, 97, 100, 102

<http://tomcat.apache.org/>

**TS** Instituto de Conceptos de Vehículos. 5

<http://www.dlr.de/ts/>

**Unified Modeling Language (UML)** Lenguaje Unificado de Modelado. Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema (software o no). 6

<http://www.uml.org/>

**Uniform Resource Locator (URL)** Cadena de caracteres con la que se asigna una dirección única de Internet a cada recurso de información disponible. 35

---

**Vaadin** Framework de programación Open Source para aplicaciones RIA programado en Java.

Su funcionamiento se basa en una arquitectura del lado del servidor. Esto significa que la mayoría de la lógica transcurre este servidor. Usando AJAX, se produce la comunicación del navegador web con el servidor asegurando de esta forma la interacción del usuario con los componentes. En la parte superior del cliente se encuentran los componentes de Vaadin los cuales pueden ser ampliados usando Google Web Toolkit (GWT) . 25, 27, 28, 42, 44

<https://vaadin.com/>

**VF** Instituto para la Investigación del Transporte. 3, 89, 103

<http://www.dlr.de/vf/>

**WebGL** API para JavaScript que permite usar la implementación nativa de OpenGL ES 2.0 que en un futuro será incorporada a los navegadores web. Actualmente se utiliza para la visualización de gráficos en 3D en la web. 38

<http://www.khronos.org/webgl/>

**Wiki** Sitio web colaborativo donde los editores pueden trabajar conjuntamente a través de un navegador web. 12, 19

**WireMock** Biblioteca Open Source escrita en Java para simular y mocking servicios web creando un servidor Http propio. 75

<http://wiremock.org/>

# Parte I

## Prolegómeno



# Capítulo 1

## Introducción

El presente Proyecto de Fin de Carrera ha sido desarrollado en el DRL contando con el apoyo del Sistemas Distribuidos y Componentes de Software (SC-VSS), el Instituto para la Investigación del Transporte (VF) y el Instituto para la Investigación del Trasporte Aéreo y el Sistema Aeroportuario (FW) y la supervisión de mi tutor Juan Manuel Dodero durante todo el proceso de desarrollo.

### 1.1. El proyecto KnowledgeFinder

El proyecto KnowledgeFinder nació con el objetivo de ser un framework de búsqueda para portales del conocimiento cuyo finalidad principal es la recuperación de la información y del conocimiento de una forma simple e intuitiva.

El proyecto KnowledgeFinder fue desarrollado dentro del SC-VSS y actualmente se encuentra implantado en varios proyectos utilizándose como apoyo a la gestión del conocimiento.

### 1.2. Motivación

Como parte del programa para la “Integración e interoperabilidad de las bases de datos sobre el transporte en el DRL” el VF así como el FW, trabajando conjuntamente con SC-VSS, han creado un portal unificado que engloba los datos provenientes de “Clearingsstellen Verkehr” y “MONITOR-Portal”. Este nuevo portal, STRADA@DLR es una implementación del KnowledgeFinder descrito en el apartado 1.1. Este portal permite la búsqueda en ambos conjuntos de datos y muestra como el trabajo conjunto de ambos portales de sus respectivos institutos refuerzan la percepción de las actividades relacionadas con la investigación del trasporte en el DRL.



Figura 1.1: Logotipo MONITOR-Portal



Figura 1.2: Logotipo Clearingstelle Verkehr

### 1.2.1. El proyecto STRADA@DLR

Un desafío particular consiste en saber conjugar el contenido como la relación técnica de ambos esquemas de (Meta-)datos. Estos esquemas muestran ciertas coincidencias así como diferencias. Por otra parte, los portales “Clearingstellen Verkehr” y “MONITOR-Portal” usan diferentes sistemas de datos que se integran en la plataforma. Otro punto importante a destacar en la concepción e implementación de STRADA@DLR es la experiencia de usuario, la visualización de los (Meta-)datos y las funcionalidades de búsqueda.

No obstante, durante el desarrollo del portal STRADA@DLR se identificaron posibles mejoras y funcionalidades adicionales para seguir mejorando la experiencia, la usabilidad y aceptación del portal por parte los usuarios. Pero estas mejoras no se pudieron realizar con los recursos en su momento disponibles; temporales y de contenidos.

Gracias al trabajo conjuntamente realizado para la investigación del transporte en el DRL se han obtenido estructuras de datos que sólo se muestran parcialmente en la primera versión del software KnowledgeFinder. Junto a la exploración basada en texto, una representación visual donde se puedan identificar los vínculos entre los datos individuales facilitaría la percepción y accesibilidad de la información.

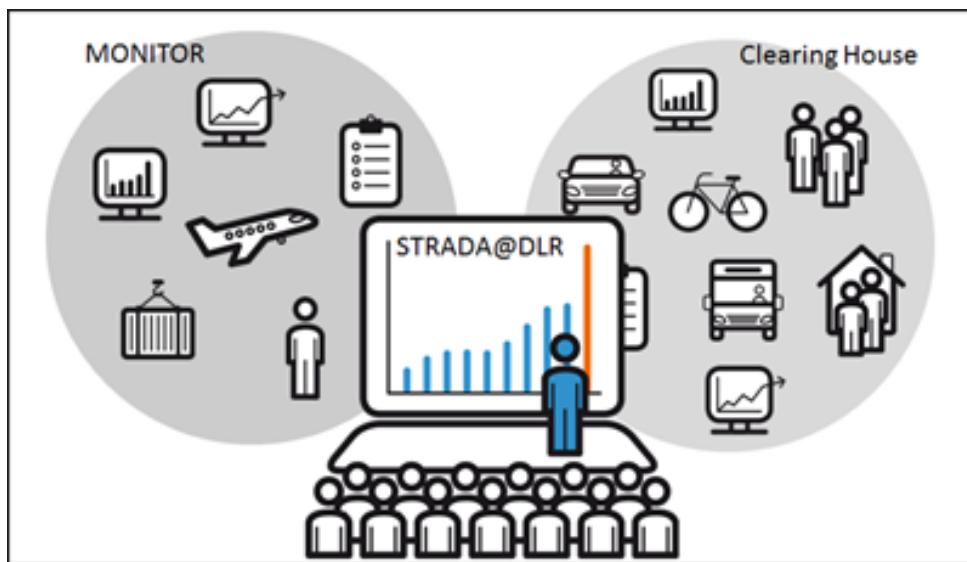


Figura 1.3: Aproximación del Proyecto STRADA@DLR [fTR]

Además de la mejora de la visualización de la información contenida, el portal ha de estar predisposta a reforzarse y ampliarse con participación de otros institutos que aporten sus bases de datos sobre el “Transporte y la Movilidad” para continuar con el desarrollo del conocimiento en el DRL.

### 1.3. Alcance

Este proyecto debe considerarse como un estudio piloto que establece las bases para la posible implementación usando otras fuentes de datos y permitiendo un mejor desarrollo en la gestión del conocimiento en el DRL. Su finalidad es la mejora de la representación externa y la disposición de los datos del DRL contribuyendo al fortalecimiento de la investigación sobre el “Transporte y la Movilidad”.

Para el público general como el especializado en los campos de política e industria se ofrecerán búsquedas avanzadas y optimizadas sobre los (Meta-)datos, que son actualizados continuamente por los institutos involucrados. De igual forma, los usuarios deben ser atraídos por el grado de detalle y de calidad del contenido de la información proporcionada respecto a los (Meta-)datos. Por lo tanto, el DRL jugará un papel importante como proveedor de servicios para la información sobre los datos incluso cuando por cuestiones de licencia estén restringidos para cierto grupos de usuarios.

Por este motivo, el objetivo principal de este proyecto es la creación de una nueva versión del framework de búsqueda KnowledgeFinder, KnowledgeFinder II. Las características más importantes del proyecto a mejorar se pueden desglosar en dos puntos principales; enriquecimiento de la visualización representando estructuras y relaciones complejas de los datos y participación de nuevas fuentes de datos en el DRL relacionadas con el “Transporte y la Movilidad”.

#### 1.3.1. Mejora de la visualización de estructuras y relaciones complejas de los datos

La primera versión del proyecto KnowledgeFinder permite una búsqueda Full-Text sobre los datos de origen. Esta búsqueda está sólo centrada en el texto y, por lo tanto, muy limitada para los esquemas de (Meta-)datos disponibles. Por ello, se deben diseñar e implementar para los usuarios finales unos sistemas de visualización avanzados que representen los esquemas de (Meta-)datos dinámicos.

La finalidad de esta nueva visualización es la de representar las relaciones espaciales, temporales y contextuales entre los conjuntos y fuentes de datos y, por tanto, la generación de nuevos conocimientos y resultados que faciliten el proceso investigador y mejore la experiencia de usuario.

#### 1.3.2. Participación de nuevas fuentes de datos en el DLR relacionadas con el “Transporte y la Movilidad”

Con el fin de aumentar el atractivo y valor del proyecto KnowledgeFinder II tanto dentro como fuera del DRL, el sistema debe poder incorporar con relativa simpleza nuevas bases de datos adicionales relacionadas con el “Transporte y la Movilidad” de institutos del DRL.

Un claro ejemplo de estas posibilidades de mejora del conocimiento son el Instituto de Conceptos de Vehículos (TS) y el Instituto para las Técnicas de Transporte (FK). Ambos poseen una base del conocimiento relacionada con el “Transporte y la Movilidad” y un declarado interés por contribuir a una fuente unificada de conocimiento conjunta.

---

## **1.4. Organización del documento**

Este documento explica el proceso de desarrollo de la aplicación KnowledgeFinder II y la aplicación para el portal STRADA@DLR.

Empezando por la descripción de los objetivos iniciales y requisitos esenciales que deberá reunir dicho proyecto, se describe la planificación temporal del proyecto en un diagrama de Gantt, los costes estimados, estudio de los riesgos y la calidad del sistema.

A medida que se avance por esta documentación, se irá explicando las fases del desarrollo del software; estudio del sistema actual y los requisitos y selección de la solución, análisis del sistema incluyendo gráficos UML para los de casos de uso, diseño del sistema a implantar desglosado en módulos, explicación de la implementación y las pruebas realizadas.

Para finalizar, se incluyen los manuales necesarios para la implantación y explotación del software y el manual para el usuario final. También se comentan las posibles mejoras o aplicaciones que pudieran considerarse en futuras versiones de este software.

# Capítulo 2

## Planificación

### 2.1. El Centro Aeroespacial Alemán (DLR)

El DRL es una de las instituciones públicas más importantes dedicadas a la investigación en la República Federal Alemana.

Sus oficinas centrales se encuentran en la ciudad de Colonia, más de 15 delegaciones, con 32 institutos, repartidas por el territorio nacional alemán y 5 situadas a lo largo del mundo [DLR14]. Actualmente cuenta con más de 7.700 trabajadores y un presupuesto anual dependiente del Ministerio de Economía y Energía Alemán (BMWi)- de 157 millones de euros para el año 2014 [BMW14].

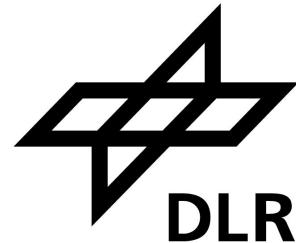


Figura 2.1: Logotipo DRL

En instituciones orientadas a la investigación, como es el DRL, la gestión del conocimiento tiene un significado muy importante. Dado el cambio de personal constante involucrado en las distintas fuentes de conocimiento, reside el peligro de pérdida del mismo.

La importancia de la gestión del conocimiento es debido a que éste concreta las actividades de una entidad. Estas actividades “tienen como meta una mejora de la gestión específica de la organización tanto para el conocimiento interno como externo” [Cis10]. Por lo tanto, el objetivo no debería de ser sólo el almacenamiento sino también la recuperación y reutilización del conocimiento obtenido en situaciones anteriores [Den12].

#### 2.1.1. “Transporte y la Movilidad” en el DLR

Junto con materias relacionadas con la aeronáutica, el espacio, la energía y la seguridad, el transporte es uno de sus temas de estudio, destacando la evolución de éste como unos de los puntos de trabajo más importantes. Tanto es así que 26 institutos del DRL contribuyen en su competencia específica a la mejora del ámbito del transporte. La mayoría de ellos generan gran cantidad de conjuntos de datos estadísticos y bases de datos por lo que la gestión del conocimiento juega un papel clave en puntos como el almacenaje, descripción y su (re)utilización. En el campo

de investigación sobre el transporte y sus datos existen actualmente dos portales en el DRL; el “MONITOR-Portal” del FW, implementación del framework KnowledgeFinder, que aporta datos sobre el transporte aéreo y el portal “Clearingstellen Verkehr” del VF contribuye con los datos sobre la investigación del transporte.

### **2.1.1.1. Instituto para la Investigación del Transporte Aéreo y el Sistema Aeroportuario**

El foco de la investigación de este instituto es examinar cómo los dos sistemas, el tráfico aéreo y el sistema de aeropuertos, evolucionan con paso del tiempo bajo ciertas condiciones y cómo conseguir un estado deseado para cada uno. Para ello, el instituto realiza las siguientes tareas:

- Analizar la situación y el desarrollo actual.
- Examinar la posible evolución futura de ambos sistemas, por ejemplo, a través de estudios de simulaciones.
- La construcción de herramientas de software para ayudar a evaluar las condiciones actuales y evitar situaciones indeseables.
- Desarrollar métodos para gestionar los aeropuertos de forma eficiente.
- Observar los efectos de las medidas aplicadas.

Por lo tanto, el objetivo de la investigación del transporte aéreo es el desarrollo de estrategias y medidas destinadas a introducir cambios en la infraestructura o las nuevas regulaciones para el sistema de transporte aéreo en su conjunto a largo plazo.

Tiene como objetivo la investigación del sistema aeroportuario en el desarrollo de medidas para gestionar los aeropuertos de manera eficiente. A través del estudio del movimiento de pasajeros en los procesos de la parte pública y los procesos de parte aeronáutica se intenta fomentar un transporte intermodal eficiente [FW14].

### **2.1.1.2. Instituto para la Investigación del Transporte**

Este instituto es el principal proveedor oficial para los hogares alemanes de encuestas y estadísticas relacionadas con el transporte. Aporta datos de sobre el transporte público y encuestas de movilidad.

Sus investigaciones se centran en los avances y perspectivas del transporte de pasajeros y comercial para conseguir en el futuro un sistema de transporte moderno, eficiente y sostenible para las personas y el medio ambiente.

Los campos de investigación del instituto son:

- Estudios de patrones de movilidad sobre viajes domésticos y de negocios de personas.
  - Modelos para representar y prever la demanda de transporte regional de pasajeros y comercial.
  - Evaluación de las tecnologías y medidas en cuanto su posible efectividad.
-

- La aceptación y uso efectivo de la red eléctrica para el transporte de pasajeros y comercial.
- Las interacciones entre la información y la comunicación (TIC) con la movilidad.

Este instituto está conectado en red a través de la cooperación en investigación y proyectos a nivel nacional e internacional. Colaborando estrechamente con la docencia e investigación universitaria y la enseñanza en centros de educación superiores y de investigación [VF14].

Los institutos VF y FW asignaron como personas responsables de velar por las necesidades de sus institutos, y por lo tanto de la visión proporcionada de su conocimiento por este proyecto, a varios Research Fellows. Éstos, coordinados por el SC-VSS, colaboraron muy activamente en la ejecución del proyecto, proporcionando la visión de los institutos y el personal investigador que usará en un futuro la aplicación.

### 2.1.2. Equipamiento informático

Las peculiares características del DRL ha condicionado a lo largo del proyecto fuertemente la elección de las herramientas informáticas a usar para este proyecto.

#### 2.1.2.1. Software

Partiendo de una lista con el software permitido en el DRL, se han utilizado los siguientes programas:

##### Herramientas para la gestión

Para controlar el proceso de software y la evolución del proyecto se han utilizado principalmente los programas mostrados en la tabla 2.1.

Software	Tipo	Versión	Comentario
Subversion	Software para el control de versiones	1.7	Utilizado por defecto en el DRL para el desarrollo de software propio.
MantisBT	Bug tracker	1.2	Utilizado por defecto en el DRL para el desarrollo de software propio.
Jenkins	Integración continua	1.573	Utilizado por defecto en el DRL para el desarrollo de software propio.

Tabla 2.1: Software usado para la gestión

##### Herramientas del entorno de desarrollo

El entorno de programación donde ha implementado el proyecto se ha sustentando de la ayuda de los programas indicados en la tabla 2.2.

Software	Tipo	Versión	Comentario
Eclipse	Entorno de desarrollo integrado (IDE)	Kepler SR2	Con los plugins para Subversion, MantisBT y Jenkins
Maven	Gestión y construcción de proyectos	3.2.1	Conjuntamente con el plugin para Eclipse
JRebel	auto-Deploying	5.6.1	Conjuntamente con el plugin para Eclipse

Tabla 2.2: Software usado como entorno de desarrollo

### Herramientas usadas para el desarrollo

El producto resultado de este proyecto requiere o ha necesitado en alguna de sus etapas de desarrollo el software listado en la tabla 2.3.

Software	Tipo	Versión	Comentario
Java	Lenguaje de programación	1.7	Lenguaje conocido ampliamente por SC-VSS
Wiremock	Framework para servicio web	1.18	Mocking web services
Spring	Framework para la web	4.0.3	Conjuntamente con el plugin para Eclipse
Jackson	Librería Java	1.9.13	Librería para tratar documentos JSON
SLF4J	Librería Java	1.9.13	Sistema de logging

Tabla 2.3: Software usado para desarrollar el programa

### Herramientas usadas para la fase de testing

Para comprobar el correcto funcionamiento y asegurar la calidad del producto final se han utilizado las herramientas de software indicadas en la tabla 2.4.

Software	Tipo	Versión	Comentario
Powermock	Librería Java	1.4	Usado para simular comportamientos en los test
EasyMock	Librería Java	3.2	Usado para simular comportamientos en los test
JUnit	Librería Java	4.12	Herramienta de testing de Java

Tabla 2.4: Software usado durante la fase de testing

### Herramientas usadas para el despliegue

El software de la tabla 2.5 fue utilizado tanto durante el desarrollo como en el servidor de pruebas donde reside el proyecto para comprobar en todo su conjunto el correcto funcionamiento y como demostración.

Software	Tipo	Versión	Comentario
Liferay	Portal de gestión de contenidos	6.2.1 CE GA2	Utilizado junto sus plugins para Eclipse durante el desarrollo
Tomcat	Servidor web	7	Usado conjuntamente con gls-liferay
Solr	Motor de búsqueda	4.8.1	Motor de búsqueda desarrollado en Java
Jetty	Servidor web	7	Usado conjuntamente con Solr

Tabla 2.5: Software usado para el despliegue de la aplicación

### Herramientas usadas para documentación

Para generar la presente documentación se han utilizado las herramientas expuestas en la tabla 2.6.

Software	Tipo	Versión	Comentario
writeLATEX	Editor online		Editor web para LATEX
GanttProject	Editor de gráficos	2.5	Herramienta para la generación de gráficos de Gantt
Gimp	Editor de imágenes	2.8.10	Edición de imágenes muy completo
ConceptDraw	Editor de gráficos	10	Edición de gráficos UML

Tabla 2.6: Software usado para la generación de la documentación

### Herramientas usadas para la comunicación y coordinación

Dado el carácter distribuido y descentralizado de los departamentos en el DRL y la distancia física que separan cada uno de los agentes implicados en el proyecto, las herramientas de comunicación usadas son un factor clave durante el desarrollo del proyecto, tabla 2.7.

Software	Tipo	Versión	Comentario
Lync	Servicio de mensajería instantánea	4.0.7	Usado en DRL como sistema oficial de comunicación entre los trabajadores
Outlook	Organizador personal	14.0.7	Usado en DRL como sistema de correo y calendario
Confluence	Wiki	5.5	Wiki para alojar la documentación de los proyectos desarrollados en DRL

Tabla 2.7: Herramientas usadas para la comunicación y coordinación

### 2.1.2.2. Hardware

En lo referente al hardware, se distinguen dos máquinas únicamente; la dedicada a la implementación del producto y un servidor usado como máquina de pruebas.

#### Hardware para el desarrollo

A continuación se muestra en el cuadro 2.8 la información del hardware utilizado para el desarrollo del proyecto.

S.O.	Windows 7
Procesador	Intel Core 2 Duo P8600 - 2.4GHz
Memoria RAM	4 GB
Pantalla	15.4- 22"
Disco Duro	500GB
Tarjeta Gráfica	Intel Graphics Media Accelerator (GMA) 4500MHD

Tabla 2.8: Hardware del equipo usado para el desarrollo

#### Hardware del servidor

Las propiedades del servidor gestionado por DRL donde reside la aplicación de prueba se muestran en la tabla 2.9.

S.O.	Ubuntu 12.04.04 TLS
Procesador	Intel® Xeon® Processor E5-2660
Memoria RAM	4 GB
Disco Duro	500GB

Tabla 2.9: Hardware del servidor para la aplicación de pruebas

## 2.2. Metodología de desarrollo

Las fases iniciales del proyecto pueden dividir en los siguientes puntos:

- **Estudio del proyecto KnowledgeFinder**

La primera fase del proyecto consistió en un estudio exhaustivo de la documentación y la implementación del proyecto predecesor. También se produjeron varias reuniones con los miembros del SC-VSS para perfilar los aspectos menos nítidos y confusos del funcionamiento e implementación de KnowledgeFinder.

- **Estudio de las nuevas necesidades de visualización**

Una vez estudiado y comprendido la situación de la que se partía, se estudiaron las distintas posibilidades de visualización y de interacciones de los usuarios sobre los datos. Todo esto teniendo siempre presente la experiencia de usuario como punto clave del estudio.

- **Estudio de las tecnologías disponibles para la visualización**

Para comprobar qué tecnologías son las más apropiadas para obtener una visualización acorde a las necesidades del proyecto y del usuario, se estudiaron un amplio conjunto de tecnologías y opciones de aplicación.

- **Estudio de la capa de servicios**

Durante los estudios previos explicados anteriormente, se detectaron grandes dificultades para cubrir los objetivos del proyecto (ver capítulo 3). Esto condicionó que una parte de la aplicación, la implementación capa de servicios, tuviera que replantearse en conformidad con la gestión de proyectos en el SC-VSS y las necesidades de los institutos FW y VF.

Una vez obtenida toda la información necesaria para el planteamiento del proyecto, se optó por un proceso de desarrollo basado en prototipos, en concreto “*Extreme Prototyping*” [Kom06][Joe12] dada su adaptabilidad a las posibles diferentes decisiones a tomar.

### 2.2.0.3. Aplicando “*Extreme Prototyping*”

El “*Extreme Prototyping*” es un proceso para desarrollar aplicaciones, especialmente aplicaciones web, basada en el diseño de prototipos funcionales cada vez más complejos. Este proceso se divide en cuatro fases:

- **Static Prototype phase**

La primera fase consiste en un prototipo estático, implementado normalmente HTML. En el caso de este proyecto se optó por bocetos en soporte físico con ejemplos reales implementados en HTML y JavaScript para ilustrar a su vez las posibilidades de interacción de los usuarios y comportamiento del producto futuro.

- **Extended Static Prototype phase**

En siguiente fase de *Extreme Prototyping* se define el modelo lógico de los datos. Los datos provienen de la capa de servicio son definidos. Para el proyecto KnowledgeFinder II se definió los valores para el índice de búsqueda.

- **Extreme Prototype phase**

La tercera fase se apoya la implementación de un prototipo dinámico usando el framework

de programación deseado. Este prototipo es totalmente funcional sustentando su funcionamiento en una capa de servicios simulada. Al concluir esta fase, la interfaz de usuario quedó definida y su futuro funcionamiento demostrable.

#### ■ Service Implementation Phase

La última fase del *Extreme Prototyping* consiste en la implementación de los servicios anteriormente simulados. Junto con el prototipo de la fase anterior y algunos cambios de adaptación se consiguió la integración de todos los componentes implementados anteriormente.

En la imagen 2.2 se expone los fundamentos de esta metodología.

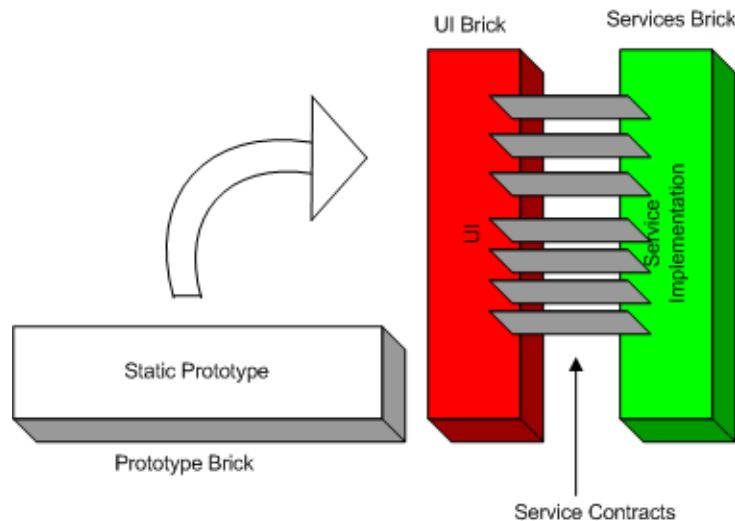


Figura 2.2: Los tres “bricks” o prototipos de *Extreme Prototyping* [Kom06]

Como fase final del presente proyecto se realizaron algunos cambios simples a petición de los institutos VF y FW. Estos cambios para sirvieron para precisar el producto obtenido en las anteriores fases. De esta forma, el producto obtenido final a través del *Extreme Prototyping* cumplió satisfactoriamente las necesidades planteadas.

### 2.3. Planificación del proyecto

La planificación inicial del proyecto se puede observar en el diagrama de Gantt 2.3. En ella se pueden diferenciar la planificación inicial para las distintas fases del desarrollo.

No obstante, durante la decisión de la visualización se acordó junto con el SC-VSS una replanificación del proyecto alargando su fases de desarrollo y la duración del conjunto del proyecto. Estas modificaciones se observan en la gráfica de Gantt 2.4.

### 2.4. Costes

Los costes de este proyecto son principalmente coste de personal. Esto abarca tanto el coste proporcional al salario de cada trabajador del DRL que interviene durante el proceso (VF, FW,

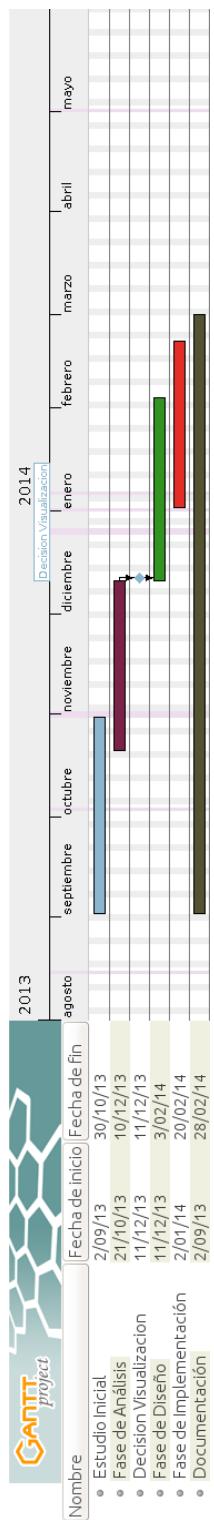


Figura 2.3: Gráfica inicial de Gantt

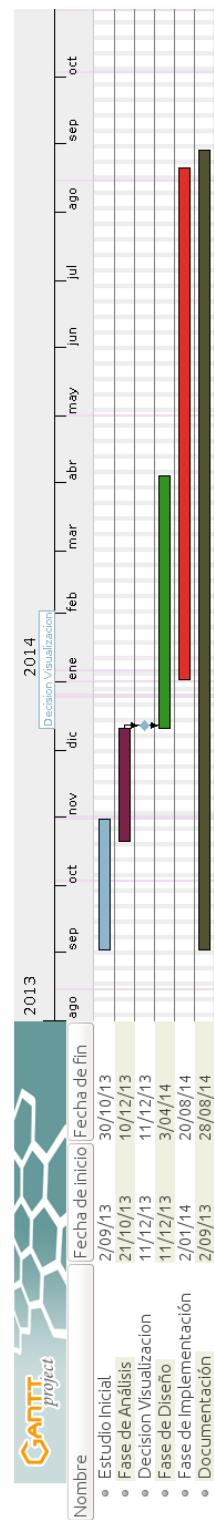


Figura 2.4: Gráfica de Gantt de la replanificación

SC-VSS, personal administrativo del DRL, ...) como los gastos derivados de las reuniones realizadas para su realización (dietas, viajes, ...).

Además habría que incluir el uso de las infraestructuras y recursos tecnológicos (hardware y software) necesitados.

Lamentablemente, dada la naturaleza del DRL, esta información no puede ser publicada al igual que el presupuesto de partida para este proyecto. En cambio podremos realizar una estimación del coste del proyecto teniendo en cuenta los anteriores factores.

#### 2.4.1. Estimación

Si suponemos que el equipo utilizado tiene un coste de 1512€ para un periodo de amortización de tres años se tiene que el coste por mes usado es de 42€. El mobiliario utilizado y el servidor junto con la infraestructura pertenecen al DRL pero implica también tenga un coste relacionado al proyecto, suponemos que asciende a 300€ al mes junto con otros gastos relacionados con el mobiliario (luz, agua, ...).

El proyecto tiene una estimación temporal de once meses, suponiendo que el desarrollador dedica unas 20 horas semanales, hasta la finalización del proyecto, hace un total de 880 horas (11 meses a 4 semanas por mes). Para las reuniones se estima que se necesitan 40 horas para todos Research Fellows y 20 horas de personal administrativo (tabla 2.10).

Cargo	Coste (persona/mes)	Descripción
Desarrollador	5,5 persona/mes	Desarrollador sin experiencia (realiza el producto)
Research Fellow	0,25 persona/mes	Colabora activamente
Personal administrativo	0,125 persona/mes	Gestión de personal y proyectos

Tabla 2.10: Resumen del gasto de personal

Los salarios del personal del DRL es público al ser una institución dependiente del Estado. Por lo tanto, partiendo de la tabla de salarios públicos [fA] obtenemos los datos relevantes para la estimación resumidos en la tabla 2.11. También se destina una partida de 2000€ para reuniones y gastos variados.

De forma complementaria, en la tabla 2.11 también se muestran los cálculos con los salarios medios obtenidos de la herramienta on-line de Infojobs [Inf14].

Calculando la estimación de todos las variables anteriormente expuestas se obtiene una estimación de 23647,88€ (13963,94€ en España ).

Recurso	Número de unidades	Precio/unidad Alemania(€)	Precio/unidad España(€)	Total Alemania(€)	Total España(€)
Desarrollador	5,5	2.965	1351,75	16307,5	7434,625
Research Fellow	0.25	4.558	2.254,08	1139,5	563,5
Admon.	0,125	3.519	1.630,5	439,875	203,8125
Equipo	11	42€	42€	462	462
Infraestructura	11	300	300	3300	3300
Variados	1	2000	2000	2000	2000
TOTAL				23647,88	13963,94

Tabla 2.11: Resumen de la estimación de los costes

## 2.5. Riesgos

En todo proyecto software se pueden producir situaciones que dificultarían el proceso de desarrollo, el cumplimiento de los plazos establecidos o desajustarían el presupuesto asignado. Estos riesgos pueden ser causados por numeroso factores de distinta naturaleza. Por ello es necesario el estudio previo de éstos.

Utilizando como base MAGERIT versión 3 [dPTyAP], se van a describir un conjunto de posibles riesgos genéricos y específicos para el desarrollo del proyecto KnowledgeFinder II.

### 2.5.1. Descripción de Riesgos

En los siguientes puntos se describen los riesgos junto con su valoración del impacto y probabilidad de ocurrencia usando las tablas 2.12 y 2.13.

Valoración	Descripción
MB	Impacto muy bajo
B	Impacto bajo
M	Impacto medio
A	Impacto alto
MA	Impacto muy alto

Tabla 2.12: Tabla valoración impacto de riesgos

Ocurrencia	Descripción
Muy Frecuente(MF)	A diario
Frecuente(F)	Una vez al mes
Frecuencia Normal(FN)	Una vez al año
Poco Frecuente(PF)	Cada varios Años

Tabla 2.13: Tabla probabilidad riesgos

### 2.5.1.1. Riesgos genéricos

Estos tipos de riesgos son comunes para el desarrollo de todo proyecto software:

#### G1 Riesgos de servicios (B-FN)

Representan la problemática posible causada por la prestación de servicios internos.

#### G2 Pérdida de información y datos (MA-PC)

Corrupción o pérdida total de los datos almacenados para el proyecto.

#### G3 Fallos del software(A-FN)

Problemas causados por errores o fallos del software.

#### G4 Problemas con el hardware (MA-PC)

Problemas con los elementos físicos usados (ordenadores, impresoras, ...).

#### G5 Problemas de personal (M-F)

Indisposición inesperada de alguno de los recursos humanos disponibles relacionados con el proyecto.

#### G6 Problemas de red (FN-M)

Fallo de la conexión de los sistemas de comunicación (teléfono, internet, ...).

### 2.5.1.2. Riesgos específicos

Los riesgos particulares del presente proyecto son:

#### E1 Cambio del origen de datos (A-FN)

Si es necesario obtener los datos desde otro sistema de almacenamiento (base de datos, ficheros XML, ...) habría que modificar el sistema de importación para una correcta obtención de éstos.

#### E2 Cambio en el formato del origen de datos (A-FN)

Si se modificara la especificación del formato de los datos obtenidos, habría que modificar el sistema de extracción de los mismos.

#### E3 Variación del sistema de información (A-MF)

A lo largo de la vida del proyecto, las necesidades del sistema de información pueden variar.

#### E4 Modificación del funcionamiento de la visualización (A-MF)

El sistema de visualización puede cambiar obligando al uso o modificación de nuevos Componentes visuales.

### 2.5.2. Subsanación de los Riesgos

El departamento SC-VSS dispone de un servidor con SVN. Usando este sistema a de control de versiones se subsanarán los riesgos que provocan una pérdida de datos (G2-4). Este servidor reside en las instalaciones de T-Systems con copias de seguridad diariamente programadas.

Los posibles problemas causados por los riesgos de personal (G5) implican, sobre todo, el replanteamiento de las reuniones con las personas involucradas del DRL. Usando una planificación de reuniones a largo plazo con posibles alternativas se evitan, en su mayoría, la repercusión en el proyecto. Respecto al equipo de desarrollo, éste se compone solamente de una persona por lo que este riesgo queda considerablemente acotado en sus efectos y sus posibles opciones para resarcirlo.

Para desagaviar los riesgos específicos de la aplicación (E1-4), durante el proceso de desarrollo se ha de estar en contacto directo con el entorno interesado en el proyecto (*Stackholders*).

## 2.6. Aseguramiento de calidad

Para asegurar la calidad del proceso de desarrollo y llevar un control sobre el mismo se debe utilizar el software MantisBT en conexión con los *commits* realizados en SVN. Para ello se obliga a que cada *commit* ejecutado esté relacionado con un ticket o tarea previamente definida en MantisBT.

La documentación del proyecto generada debe accesible y verificable por el resto de miembros del SC-VSS a través de una Wiki interna.

La calidad del código fuente adquiere una relevancia muy alta para asegurar el mantenimiento del software. Para ello, hay que prestar especial atención durante el desarrollo en los siguientes aspectos:

- Código claro y estructurado.
- Utilización de patrones de diseño.
- Utilización de estándares de codificación para Java [Ora99], JavaScript [Incb] y HTML [Inca].
- Código moderadamente comentado.
- Utilización de ficheros de configuración.

A través de testing se debe asegurar el correcto funcionamiento de la aplicación en sus aspectos más importantes. Junto con Jenkins, se debe realizar la integración e inspección continua.

---

## Parte II

# Desarrollo



# Capítulo 3

## Requisitos del Sistema

### 3.1. Situación actual

Como se ha explicado anteriormente en el capítulo 1, el presente proyecto nació con la necesidad de mejoras del framework de búsqueda KnowledgeFinder aplicándose como modelo piloto al proyecto STRADA@DLR.

#### 3.1.1. Procesos de Negocio

A continuación se detallan los procesos de negocio principales del proyecto KnowledgeFinder que actualmente están implantados.

##### 3.1.1.1. Importación de los datos

En este proceso de negocio se produce la importación de los datos provenientes de un único repositorio de SVN. En el proceso implantado es necesario tanto permisos para poder obtener los datos del repositorio remoto como una copia local, previamente descargada. En la imagen 3.1 se representa el modelo proceso de este negocio en UML.

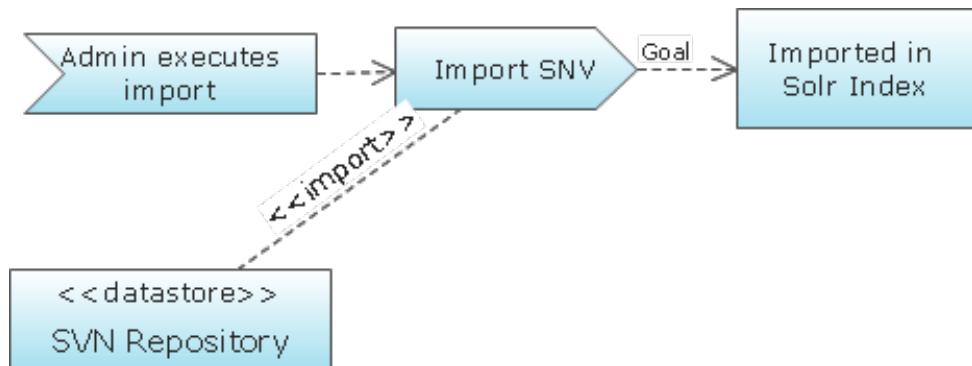


Figura 3.1: Modelo de proceso de negocio, importación de los datos en KnowledgeFinder

##### 3.1.1.2. Búsqueda en el portal

Este proceso de negocio se produce cuando el usuario filtra el conjunto de documentos usando para ello la interfaz web. Como resultado de la acción, el usuario obtiene una lista acorde con el

filtro aplicado. En la imagen 3.2 se representa el modelo proceso de este negocio en UML.

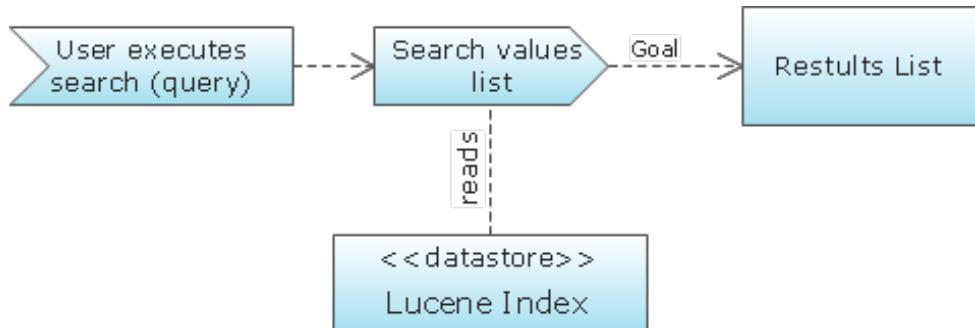


Figura 3.2: Modelo de proceso de negocio, búsqueda en el portal KnowledgeFinder

### 3.1.2. Entorno Tecnológico

## Construcción

La versión de la que se parte de KnowledgeFinder está implementada usando Java SDK 6 Update 45 y Maven para la construcción del producto unificando todas sus partes, en la figura 3.3 se puede observar un esquema de la organización de los archivos de configuración de Maven para esta versión del proyecto.

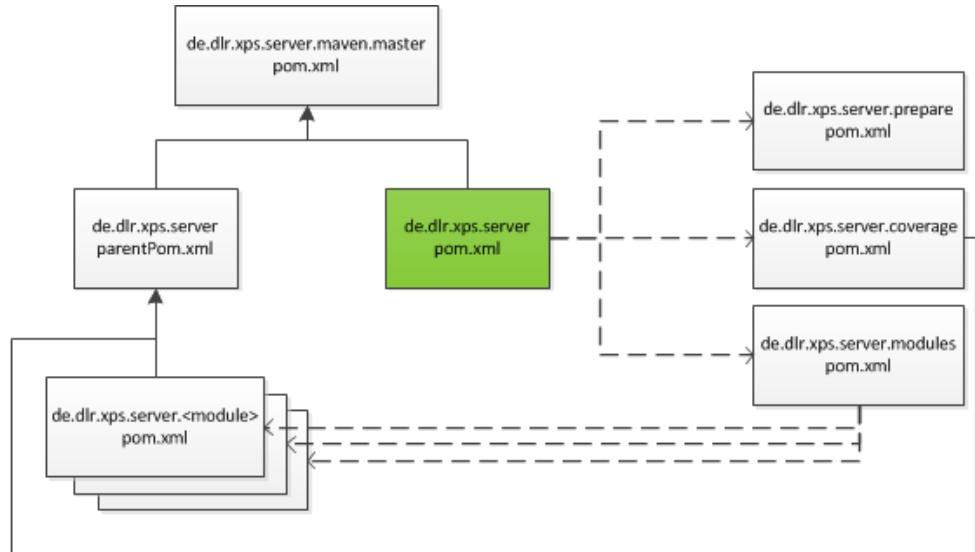


Figura 3.3: Estructura simplificada de Maven en KnowledgeFinder

Dado el portal donde se aloja la información de los institutos a los que está destinado KnowledgeFinder es Liferay, para esta versión del proyecto se implementó como contenedor un portlet para Liferay 6.1 CE GA2 (6.1.1) sobre un servidor Tomcat 6.

## Importación

A través de la lectura del repositorio remoto una copia local se extraen los datos de los documentos y se genera un índice de búsqueda en la fase de importación. Éste es índice es una instancia

de Lucene en su versión 3.0.2.

### Índice de búsqueda

Partiendo de los datos del portal Clearingstellen Verkehr (imagen 3.4) y del portal MONITOR-Portal (imagen 3.5) se crea durante la importación de los datos un índice de búsqueda con las propiedades expuestas en la imagen 3.6 para la instancia STRADA@DLR del framework KnowledgeFinder usada como referencia [fTR13].

Item	Type	Item	Type
<b>General information on data set</b>			
Title	Text	Keywords	Text
Title (abbreviation)	Text	Website for data set	Text
Summary, purpose of data collection	Text	Possible links to other data sets	Text
<b>Temporal and spatial information</b>			
Time period of data collection	Date	City, municipality	Text
Country	Selection	Geo Reference available	Selection
Federal state	Text	Additional information	Text
<b>Investigation and data collection methodology</b>			
Data set variables	Text	Population	Text
Investigation design	Selection	Analysis unit	Selection
Frequency of data collection	Selection	Sample size	Text
Data collection method	Selection	Representativeness	Selection
Data collection tool	Selection		
<b>Access to data set</b>			
Collected by	Text	Data format(s)	Selection
Client	Text	Size of data set	Selection
Contact for data set	Text	General comments and addenda to data set	Text

Figura 3.4: (Meta-)datos de Clearingstellen Verkehr [fTR13]

Item	Type	Item	Type
<b>A. Category Information</b>			
Resource type	Selection	Content category	Selection
<b>B. Basic Information</b>			
Title	Text	Subject/keywords Online link Description	Text
Publisher/supplier	Text		Text
Author	Person		Text
<b>C. Technical Information</b>			
Spatial coverage	Selection	Temporal scope of publication Data status Data format Size of data file Language Resource identifier	Text
Temporal coverage	Text		Selection
Time horizon	Selection		Selection
Temporal resolution	Selection		Text
Variables	Text		Selection
Frequency of publication	Selection		Text
Date of release	Text		
<b>D. Information for Using the Data (Usefulness, Possibilities and Limitations)</b>			
Availability	Text	Target group Users of the data Relation Alternative data sources Possibilities and limitations of the data	Text
Method of data collection/processing	Text		Text
Collection reason	Text		Text
Collection financer	Text		Text
Field(s) of application	Text		Text

Figura 3.5: (Meta-)datos de MONITOR-Portal [fTR13]

Item	Type	Search result teaser	Search filter	Presentation in the detailed metadata information sheet
Title	Text	x		x
Publisher	Text			x
Author	Person			x
Subject/keywords	Selection	x		x
Description	Text	x		x
Spatial coverage	Selection		x	x
Frequency of publication	Text			x
Temporal coverage	Text		x	x
Frequency of data collection	Text			x
Time period of data collection	Text			x
Link	Text			x
Data category	Selection		x	x

Figura 3.6: (Meta-)datos de STRADA@DLR [fTR13]

### Interfaz de usuario

Para la realización de la página web se utilizó el framework de programación web Vaadin 6. Únicamente usando sus componentes predefinidos se implementó la interfaz de usuario y su funcionalidad.

Junto con la configuración de un tema de Liferay por cada instancia del framework (ELIB-Portal, MONITOR-Portal y STRADA@DLR) se terminó de perfilar la apariencia y el comportamiento de la interfaz.

## Configuración

Exceptuando la importación de los datos que se configura a nivel de parámetros en la ejecución, la configuración y adaptación del KnowledgeFinder para los distintos escenarios se realiza a nivel de código, es decir, que para cada instancia que se implemente hay que editar el framework para adaptarlo a las necesidades específicas requeridas.

### 3.1.3. Fortalezas y Debilidades

La implementación del framework KnowledgeFinder presenta unos aspectos positivos y negativos que se pueden agrupar en los siguientes puntos:

- **Importación de los datos y el índice de búsqueda**

Como anteriormente se ha comentado, para la importación de los datos en el índice de Lucene es necesario contar con una copia local del repositorio SVN y acceso al repositorio remoto. Esto representa un gran inconveniente durante la fase de desarrollo ya que el repositorio remoto es no debería de tener que ser modificado con datos de prueba durante el desarrollo.

Al obtener los datos también remotamente, el proceso de importación de los datos es dependiente tener acceso a este repositorio. Esto conlleva que el tiempo necesario para finalizar esta tarea se incremente notablemente y que el desarrollador siempre tenga que tener acceso a la red. En el código 1 se muestra un ejemplo de ejecución para importar los datos usando la linea de comandos. Por otra parte, esta simpleza hace que la importación se pueda ejecutar desde la linea de comandos o, por ejemplo, desde algún sistema automático como Cron.

```
1 java -Xmx512m -jar
2 de.dlr.xps.server.datafinder.indexer.fw-jar-with-dependencies.jar
3 --idx-path D:\Projects\xps\index\fw-test-data
4 --svn-username user
5 --svn-password *****
6 --svn-url https://svn.sistec.dlr.de/svn/xps/fw-test-data/
7 --svn-working-copy-path "D:\Repositories\fw-test-data\data\dat"
8 --parse-content
```

Código 1: Ejemplo de importación de datos en KnowledgeFinder.

- **Configuración de las nuevas instancias**

Cada instancia del framework KnowledgeFinder tiene distintos datos. Para configurar estos datos es necesario reimplementar el proyecto ya que la tratamiento de los datos está incluido en el código fuente. Esto implica, a pesar de un proceso de refactorización aplicado sobre el software, una duplicación de código para cada instancia del framework. En la actualidad hay tres instancias en producción o desarrollo (MONITOR-Portal, STRADA@DLR y un último proyecto en fase de estudio ELIB-Portal) lo que causa que exista una gran replicación de código dificultando considerablemente su desarrollo y mantenimiento.

A modo indicativo en la imagen 3.7 vemos la estructura del programa importado en Eclipse. Los módulos o paquetes acabados en “fw” corresponden al proyecto MONITOR-Portal, los acabados en “cs” al STRADA@DLR y “ly” al ELIB-Portal.

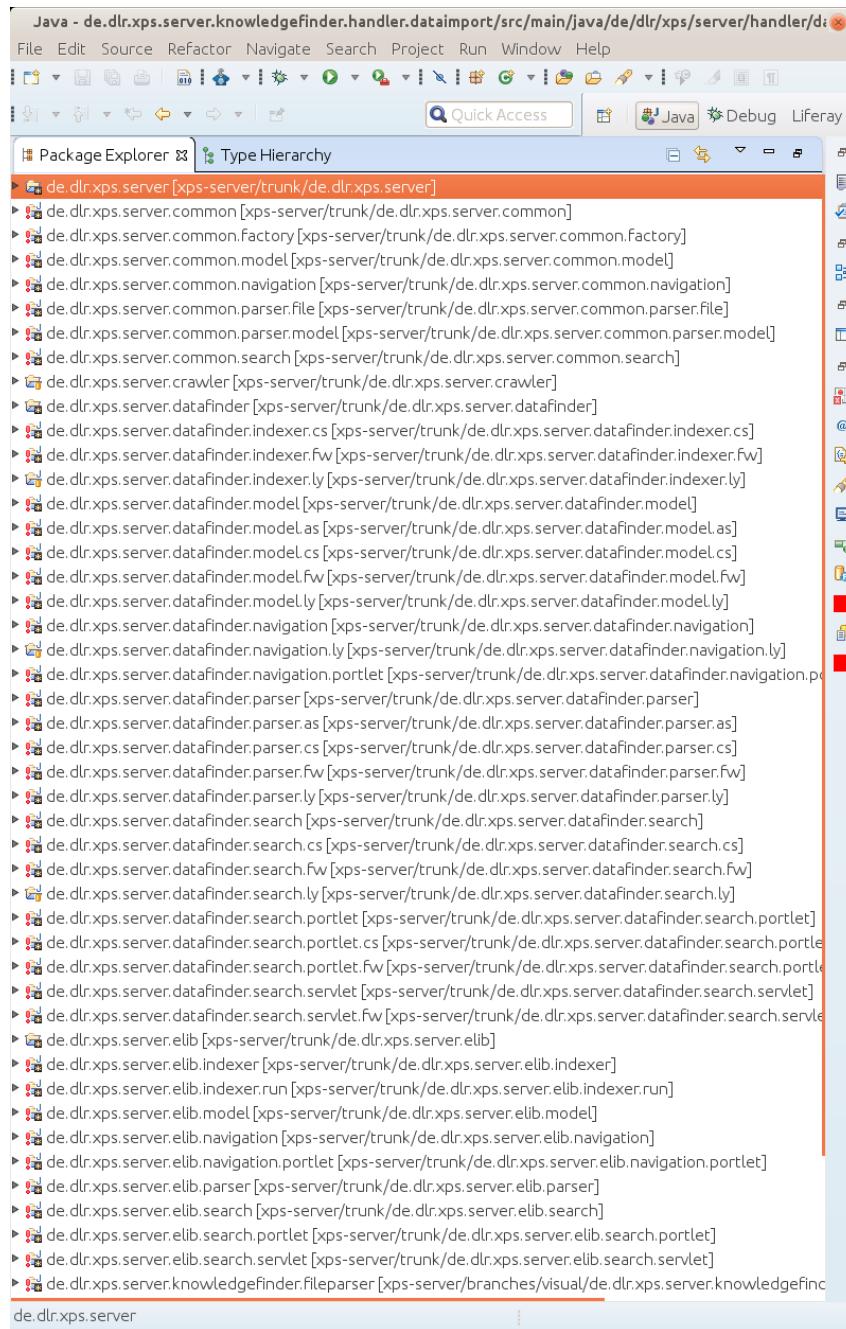


Figura 3.7: Las instancias del proyecto KnowledgeFinder en Eclipse

#### ■ La interfaz de usuario y la comunicación con el servidor

Como se ha comentado en el apartado anterior, la interfaz de usuario está implementada usando una versión no actual de Vaadin, sin soporte técnico desde mayo de 2014. En el código del KnowledgeFinder es confuso, complejo y no se encuentra bien estructurado el

sistema de tal forma que dentro del mismo proyecto no se puede diferenciar claramente entre los módulos, por ejemplo, de la interfaz de usuario y la importación de los datos.

Esta complejidad hace también que el mantenimiento y adaptación de la interfaz de usuario sea más tediosa y laboriosa.

En lo que respecta al tipo de tecnología, el framework Vaadin necesita una constante comunicación con el servidor provocando que por cada interacción con el sistema el usuario tiene que esperar una respuesta del servidor.

El sistema de búsqueda de KnowledgeFinder está centrado y limitado al texto, tanto el sistema de búsqueda como la representación de los datos. A pesar de que los relaciones presentan una más compleja, estas relaciones no se muestran de forma directa. En la imagen 3.8 se puede apreciar los distintos componentes de la interfaz para MONITOR-Portal.

The screenshot shows a web-based application interface for the MONITOR-Portal. At the top, there is a header with the DLR Institute of Air Transport and Airport Research logo and the MONITOR logo. Below the header, a sidebar on the left contains a navigation menu with links to Home, Database (selected), Methodology, Indicators, Future Studies, Background, Contact, and Imprint. The main content area displays a search results page for a 'Meta-Database'. The search bar at the top right has the text 'Relevance' and a 'Search' button. Below the search bar, it says 'Showing 1-10 of 62 entries.' and includes a 'Sort by' dropdown and a 'Short View' checkbox. There are page navigation buttons (1, 2, 3, ..., 5, 6, 7, Next...). The results list three items:

- [+] ACI Airport Economics Survey**  
Publisher/Author: ACI  
Subject/keywords: Passenger traffic, cargo traffic, economic and financial performance of airports, airport ownership, airport projects, airport retail, airport employment, airport company profiles  
Description: The Airport Economics Survey is an annual publication of ACI. It contains financial data as well as detailed analyses which are linked to airport operations of the year before the particular publication date. For this purpose more than 600 airports worldwide are asked regularly about their traffic...  
[More Information](#)
- [+] Annual World Airport Traffic Report**  
Publisher/Author: ACI  
Subject/keywords: Airports, airport traffic volume, passengers per airport, cargo (freight/mail) per airport, movements per airport  
Description: The "Annual World Airport Traffic Report" (WATR), published by ACI, is an annual publication which provides an overview on traffic trends of more than 1000 airports worldwide. The data includes the amount of traffic on a single airport level with regard to the number of movements and passengers as...  
[More Information](#)
- [+] Global Traffic Forecast**  
Publisher/Author: ACI  
Subject/keywords: Passenger demand development forecast, freight and movements forecast, traffic forecast by world region, traffic forecast by country, air transport's long-term development, growth rates  
Description: The Global Traffic Forecast is a regular publication by ACI, which is the world's trade association for airports and DKMA as one leading aviation market consulting company. The forecast is published every year and provides an overview of the expected air traffic development for the next 20 years...  
[More Information](#)

On the right side of the results list, there is a sidebar titled 'Repository: Meta-Database' with a section for 'Available filters' containing various categories with counts:

- [+] Publisher (Top 5) (1)
- EUROCONTROL (10)
- ICAO (10)
- ICAO, Air Transport Intelligence (ATI) (8)
- IATA (5)
- ACI (3)
- [+] Show all filters (32)
- [+] Spatial Coverage (Top 5) (1)
- World (44)
- Europe (26)
- North America (14)
- Africa (13)
- Asia (13)
- [+] Show all filters (27)
- [+] Time Horizon (Top 5) (1)
- Short range historical data (24)
- Long range historical data (22)
- Long-term forecast (over 15 years) (14)
- Medium-term forecast (5 to 15 years) (2)
- Short-term forecast (until 5 years) (2)
- [+] Show all filters (11)
- [+] Resource Type (Top 5) (1)
- Inventory (18)
- Statistic Compendium (17)
- Statistical Database or

Figura 3.8: Interfaz de usuario para MONITOR-Portal

### 3.2. Necesidades de Negocio

A continuación se exponen los objetivos de negocio de clientes y usuarios, junto a los modelos de proceso de negocio, del proyecto KnowledgeFinder II

### 3.2.1. Objetivos de Negocio

Los objetivos de negocio de KnowledgeFinder II son:

- **Simplificar adaptación del framework para las distintas instancias**

El nuevo KnowledgeFinder II tiene que ser un sistema que se adapte fácilmente a los distintos sistemas de conocimiento. Tanto el sistema de importación como la representación para el usuario debe ser fácilmente configurable y extensible.

- **Visualización gráfica del conocimiento**

Para esta nueva versión del framework se debe obtener una visualización gráfica del conocimiento que represente las relaciones y las estructuras complejas que existen los distintos (Meta-)datos. A través de esta visualización el usuario deberá obtener una visión conjunta estas estructuras facilitando de esta forma la búsqueda de documentos.

### 3.2.2. Procesos de Negocio

En esta sección se exponen los procesos de negocio del framework KnowledgeFinder II.

#### 3.2.2.1. Importación de los datos

En este proceso de negocio se produce la importación de los datos provenientes de repositorios de SVN configurados. En la imagen 3.9 se representa el modelo proceso de este negocio en UML.

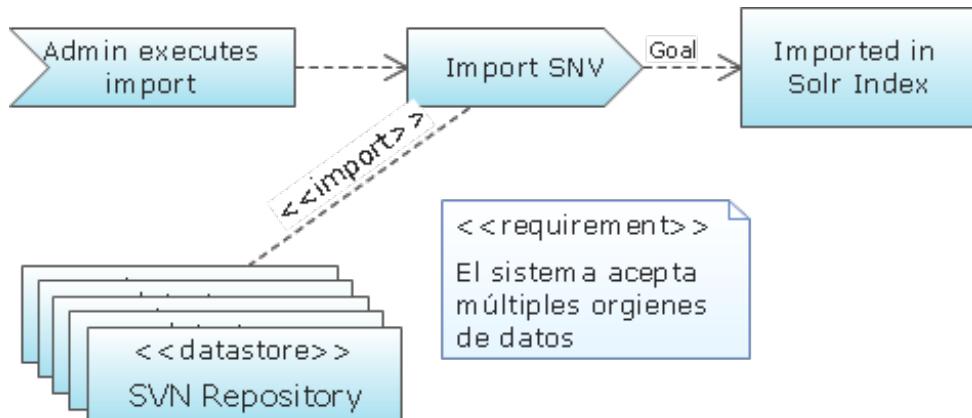


Figura 3.9: Modelo de proceso de negocio, importación de los datos en KnowledgeFinder II

#### 3.2.2.2. Búsqueda en el portal

Este proceso de negocio se produce cuando el usuario filtra el conjunto de documentos usando para ello la interfaz web. Como resultado de esta acción, el usuario obtiene una lista acorde con el filtro aplicado y una visualización de los documentos. En la imagen 3.10 se representa el modelo proceso de este negocio en UML.

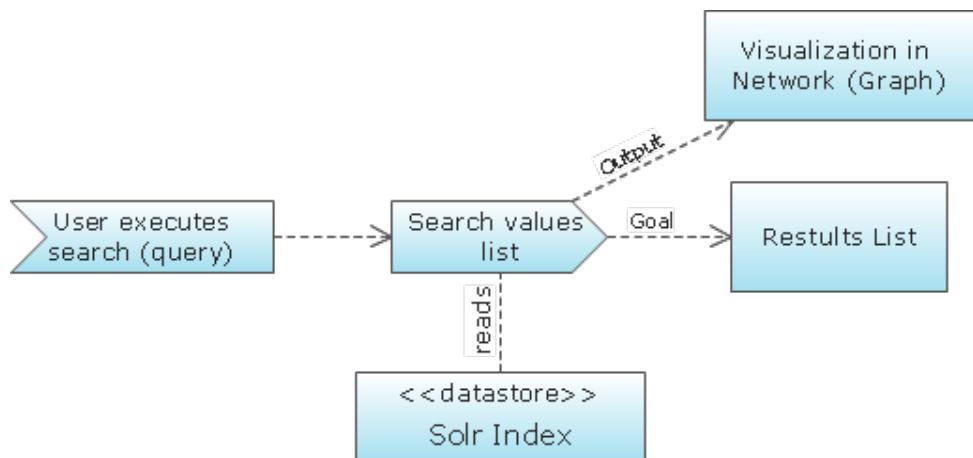


Figura 3.10: Modelo de proceso de negocio, búsqueda en el portal KnowledgeFinder II

### 3.3. Obtención de los Objetivos del Sistema

Una vez extraídas las nuevas necesidades de negocio, era necesario refinarlos para obtener los objetivos reales del sistema. El punto más dificultoso fue la elección del elemento de visualización a diseñar. Para obtener más detalles sobre qué aspectos eran más importantes, se realizó un sondeo entre los institutos involucrados.

Las visualizaciones que más se acercaban a las necesidades concretas fueron:

- #### ■ Chord Diagram (Imagen 3.11)

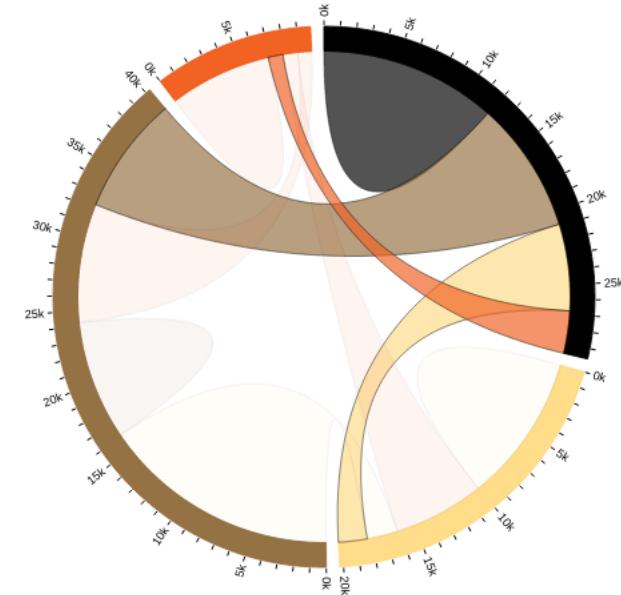


Figura 3.11: Chord Diagram

- #### ■ Bubble Chart (Imagen 3.12)

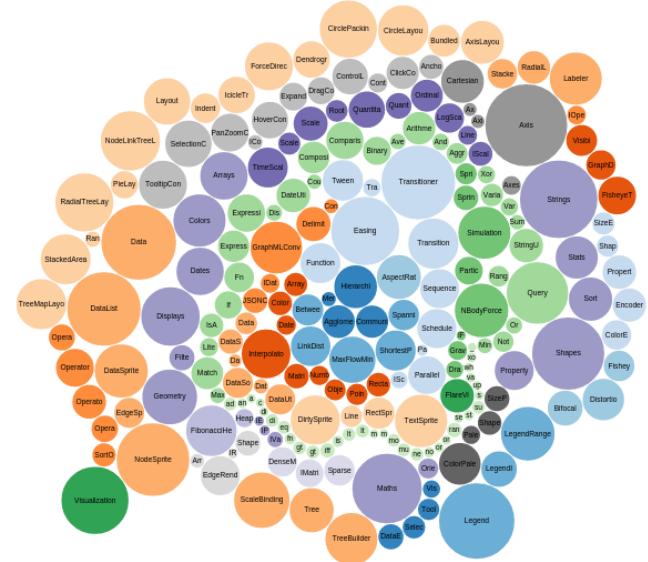


Figura 3.12: Bubble Chart

#### ■ Circle Packing (Imagen 3.13)

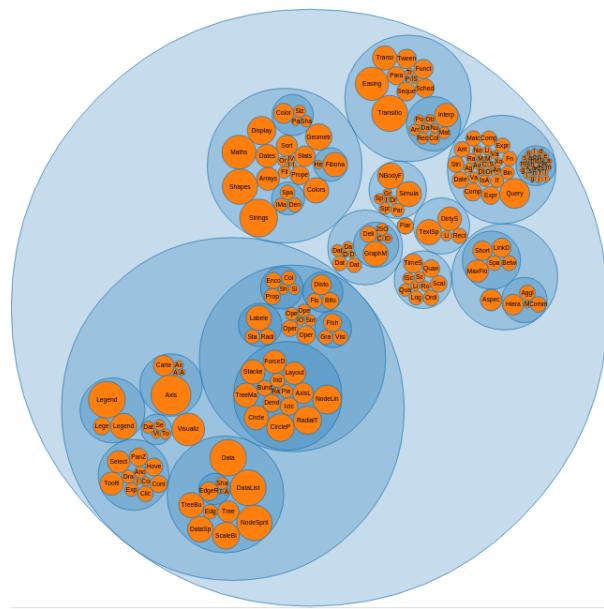


Figura 3.13: Circle Packing

#### ■ Grafo - Force Layout (Imagen 3.14)

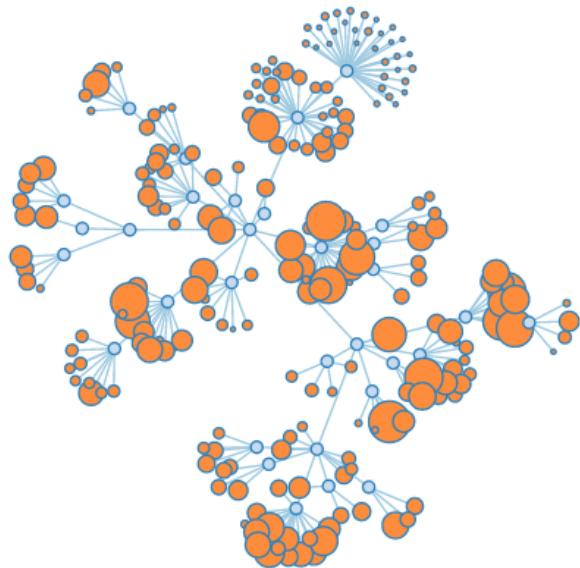


Figura 3.14: Grafo - Force Layout

Tras varias conversaciones para saber qué propiedades eran las más interesantes de cada uno, se optó por el grafo de (Meta-)datos como elemento central de la visualización para el proyecto ya que sus relaciones representan las estructuras complejas con más fidelidad y es más intuitivo para el usuario.

## 3.4. Objetivos del Sistema

### 3.4.1. Importación y adaptación configurable

El sistema de importación e indexación debe aceptar nuevos repositorios SVN. A su vez, el sistema encargado de la interfaz de usuario debe ser igualmente configurable y extensible a través de ficheros de configuración para adaptarse las peculiaridades de cada portal.

### 3.4.2. Representación de los (Meta-)datos en un grafo

Los (Meta-)datos extraídos durante el proceso de indexación se deben mostrar en un grafo interactivo junto con sus relaciones. A través de este grafo, el usuario podrá definir sus criterios de búsquedas.

## 3.5. Catálogo de Requisitos

A continuación, en este apartado se describen los requisitos del proyecto KnowledgeFinder II.

### 3.5.1. Requisitos de interfaces externas

A continuación se describirán los requisitos de conexión con otros sistemas de software con los que el sistema va a interactuar para la importación de los (Meta-)datos así como la interfaz de usuario.

### 3.5.1.1. Importación de los datos externos

La importación de la información del portal se realizará partiendo de uno o varios repositorios de SVN. Estos repositorios podrán ser remotos, necesitando en su caso autenticación, o una copia de trabajo local. La información de los valores a importar se encontrará en las propiedades de SVN en formato JSON.

### 3.5.1.2. Interfaz de usuario

La interfaz de usuario se usará por los usuarios para consultar con los datos indexados en el momento de la importación a través de un navegador web.

Los requisitos de la interfaz de usuario del KnowledgeFinder II la podemos descomponer en los siguientes elementos; menú, grafo de exploración, selección actual, campo Full-Text y la lista de resultados.

#### Menú

Para una navegación clásica, el usuario dispondrá de un menú interactivo de navegación para gestionar los filtros que se desean aplicar a la búsqueda. Para este menú se requiere las siguientes características:

- **Multinivel**

El menú constará como máximo de dos subniveles de navegación y como mínimo uno.

- **Scroll en las opciones de menú**

Para las listas largas de opciones en el menú, éste debe de disponer de un scroll.

- **Contador de documentos por filtro en el menú**

En menú se dispondrán de un contador por cada filtro aplicable con el número de documentos que actualmente contienen el correspondiente filtro.

- **(Re)ordenación del menú**

Las opciones del menú obtenidas deben mantener el siguiente orden; primero las opciones seleccionadas y luego el resto de opciones ordenadas por el número que indique su contador de documentos correspondiente.

- **Grupos en acordeón**

La lista de elementos de un submenú deberán mostrarse dentro de un sistema de acordeón.

### 3.5.1.3. Grafo de exploración

Los (Meta-)datos del conjunto de elementos importados deberán representarse en un grafo interactivo que muestre las relaciones entre ellos y la relevancia de éstos dentro del conocimiento en su totalidad.

### 3.5.1.4. Selección actual

En la interfaz de usuario se deben mostrar en una lista con los filtros actualmente han sido activados.

---

### 3.5.1.5. Campo Full-Text

La interfaz hará posible que el usuario pueda introducir texto libremente para que sea usado en el sistema de búsqueda.

### 3.5.1.6. Lista de resultados

Los documentos resultado de la búsqueda deberán de ser mostrados en una lista. En esta lista se deben poder diferenciar claramente los distintos elementos que la componen. También deberá cumplir las siguientes propiedades:

- **Lista de resultados en forma de acordeón**

La lista de resultados debe mostrar el título de cada uno de los documentos y la posibilidad de una pequeña visualización de resumen de algunos campos seleccionados usándose un sistema de acordeón.

- **Paginación en los resultados**

Para un gran número de resultados, la lista debe ser mostrada usando una paginación.

- **Ordenación en los resultados**

La lista de resultados deberá ser mostrada en un orden seleccionable por el usuario.

- **Detalles de los documentos**

En sistema debe permitir seleccionar un documento de la lista de resultados y mostrar todos sus campos disponibles en una ventana emergente.

- **Contador de elementos en lista**

Como parte de la interfaz, el usuario dispondrá de manera clara de un contador con el número total de documentos para estado actual de la búsqueda.

## 3.5.2. Requisitos funcionales

### Pliegue y despliegue de la lista de resultados

La interfaz de usuario debe permitir el pliegue y despliegue de cada elemento de la lista de resultados o todos con una simple acción.

### Pliegue y despliegue del menú

La interfaz de usuario debe permitir el pliegue y despliegue de los elementos de menú con subgrupos. El usuario podrá desplegar o contraer los subelementos por cada grupo del menú.

### Aplicar filtro de (Meta-)dato

Cuando el usuario pulse sobre un (Meta-)dato (en grafo de exploración o en el menú) se debe filtrar los resultados acorde con el (Meta-)dato seleccionado. Si pulsa una arista que comunica dos (Meta-)datos, se aplicara el filtro de ambos.

### Desactivar filtro de (Meta-)dato

Cuando el usuario pulse sobre un (Meta-)dato en el menú que ya esté aplicado o sobre la lista de la actual selección, éste se desactivará.

---

**Texto informativo de (Meta-)dato**

Cuando el usuario se encuentre sobre un (Meta-)dato (en gráfica de visualización o en el menú) se debe mostrar un texto con información adicional sobre éste.

**Resaltado de (Meta-)dato**

Cuando el usuario se encuentre sobre un (Meta-)dato (en gráfica de visualización o en el menú) se deben resaltar los elementos relacionados en el grafo, el correspondiente en el menú y los documentos que contienen ese (Meta-)dato.

**Resaltado de los (Meta-)datos de documentos**

Cuando el usuario se sitúe sobre un documento de la lista de resultados se deben resaltar los (Meta-)datos y las relaciones entre ellos en el grafo y en el menú.

**Búsqueda en el Full-Text**

El usuario a través del campo de búsqueda Full-Text podrá realizar búsquedas aproximadas, combinadas con operadores disyuntivos o conjuntivos, búsqueda exacta de texto y expresiones regulares simples (!, ?, \*, ...).

**Detalles de los documentos**

En sistema debe permitir seleccionar un documento de la lista de resultados y mostrar todos sus campos disponibles, dependiendo de los roles del usuario, en una ventana emergente.

**Interacción del menú con el gráfico visual**

Desde el menú, el usuario podrá activar o desactivar los grupos de (Meta-)datos se mostrarán en el gráfico.

### 3.5.3. Requisitos no funcionales

**Adaptación de los campos sistema de importación**

El sistema de importación de los datos debe ser fácilmente configurable para añadir o suprimir campos en el índice de búsqueda.

**Adaptación del sistema de importación**

La incorporación de nuevos tipos de fuentes de datos (por ejemplo desde ficheros XML) tiene que ser relativamente simple.

**Adaptable a los roles del usuario**

El sistema debe permitir la configuración y combinación de roles para personalizar el resultado de las consultas y los campos a mostrar.

**Compartición del estado del portal usando URL**

Partiendo del supuesto de que tienen los mismos permisos, los usuarios podrán compartir el estado de la búsqueda y de la visualización a través de la dirección URL generada.

---

**Comportamiento dinámico del elemento visual**

El elemento visual debe comportarse de forma dinámica, es decir, que por cada cambio que deba producirse en él no se deba redibujar todo el gráfico, sólo las partes implicadas.

**Interacción de los componentes**

Los componentes de la interfaz de usuario deben interaccionar entre ellos en la medida de lo posible tal que el usuario no perciba un descenso del rendimiento de la aplicación.

**Tiempo de respuesta de la interfaz**

Tanto la visualización como la búsqueda tendrá tiempo de respuestas aceptables a las peticiones de un usuario con un sistema y conexión estándar. El usuario debe poder interaccionar con el sistema sin percibir la comunicación con el servidor.

**Ampliable con futuros elementos gráficos**

En un futuro debe ser posible la fácil incorporación de nuevos elementos gráficos a la interfaz de usuario. La inclusión de mapas y líneas de tiempo que interactúen con el resto de componentes debe ser simple y eficiente.

**Grafo debe ser fácilmente configurable**

Aspectos como la complejidad o el estilo del gráfico interactivo deben ser fácilmente configurable.

**Lista de resultados debe ser fácilmente configurable**

Aspectos como la paginación o las opciones de ordenación deben ser fácilmente configurables.

**El menú debe ser fácilmente configurable**

El menú con las opciones de filtrado debe ser configurable hasta dos niveles (orden, estilos, textos, subelementos, scroll, desplegado, . . . ).

**Actualización automática del menú**

En su último nivel, las opciones son obtenidas de forma automática en el momento de la primera carga de la página.

**El estado inicial del gráfico fácilmente configurable**

El gráfico de visualización por defecto mostrado en la visita inicial de la aplicación debe ser configurable con relativa simpleza.

**Carga en paralelo de los componentes**

Los componentes que forman el interfaz de usuario deben de generarse y actualizarse en paralelo dentro de las posibilidades de las tecnologías de visualización y del navegador que el usuario esté usando.

**Estilos configurables**

Los estilos de los componentes que forman la interfaz de usuario deben de ser fácilmente configurables conjuntamente.

---

### 3.5.4. Reglas de negocio

#### Java, el lenguaje de programación

Dado el carácter piloto de KnowledgeFinder II, el proyecto debe ser desarrollado en la medida de lo posible en un lenguaje dominado por el personal del SC-VSS. En este caso Java debe de ser usado como lenguaje de programación para el sistema de indexación

### 3.5.5. Requisitos de información

El sistema de información de KnowledgeFinder II está basado en un índice de búsqueda NoSQL. Se mantendrá el mismo esquema expuesto en la imagen 3.6 exceptuando los campos que representan intervalos de tiempo, por ejemplo “Temporal Coverage”. Estos campos del índice proporcionarán la fecha máxima y mínima para los valores posibles dados para simplificar posibles consultas usando fechas. En la tabla 3.1 se representa algunas transformaciones posibles.

Valor	Fecha mín.	Fecha máx.
2003 - 2010	2003-01-01	2010-12-31
- 2010	NULL	2010-12-31
2003 -	2003-01-01	NULL
2003-03 - 2010-11	2003-03-01	2010-11-31
2003-03-10 - 2010-10-31	2003-03-10	2010-10-31

Tabla 3.1: Tabla resumen de las transformaciones para los rangos de fechas

### 3.5.6. Restricciones técnicas del sistema

#### Portlet de Liferay

La página web usada como interfaz de usuario debe ser un portlet compatible con una versión actual de Liferay.

#### Compatibilidad con HTML5

La aplicación debe ser compatible con HTML5 es decir con la amplia mayoría de los navegadores actuales. Por supuesto, el uso de la tecnología Flash no está permitido.

## 3.6. Alternativas de Solución

A continuación se expone el estudio de la solución propuesta para el framework KnowledgeFinder II.

### 3.6.1. Elección de la tecnología de visualización

Para la visualización del grafo descrito anteriormente, es necesario usar tecnología JavaScript. Entre las posibles soluciones destacan:

### 3.6.1.1. Sigma.js

Sigma es una biblioteca Open Source de JavaScript pura dedicada exclusivamente al dibujo de grafos interactivos (dinámicos y estáticos) en aplicaciones web.

Con una configuración por defecto, esta biblioteca permite la representación ajustado al contenido del grafo en WebGL o, si el navegador web lo permite, en Canvas.

Gracias a su API pública y adaptando la configuración, se puede conseguir fácilmente una personalización de cómo dibujar y de cómo interactúa el grafo. También permite que el desarrollador incluya su propia funcionalidad para la representación de los nodos y conexiones.

Entre las propiedades a destacar es su rendimiento para representar gráficos con un altísimo número de nodos, en la imagen 3.15 se puede observar la exploración de un gran grafo de una red social usando este software. Por otra parte, en la imagen 3.16 se representa una captura de pantalla de un ejemplo más simple de Sigma.js.

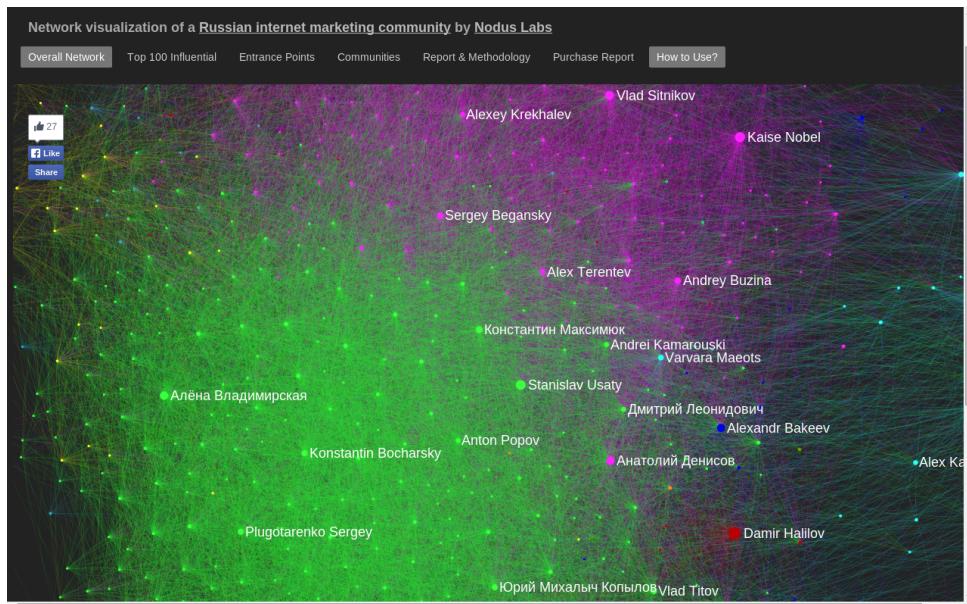


Figura 3.15: Exploración de una red social de marketing Sigma.js [LAB]

### 3.6.1.2. Cytoscape

Cytoscape es una biblioteca Open Source escrita en JavaScript para teoría de grafos, su visualización y análisis.

Permite la manipulación y representación de grafos interactivos facilitando al usuario crear sus propios eventos y personalizar la interacción con los grafos. Se integra con facilidad en las aplicaciones web y es compatible con los actuales navegadores.

En su biblioteca de funciones hay implementadas un conjunto de herramientas enfocadas a la teoría de grafos.

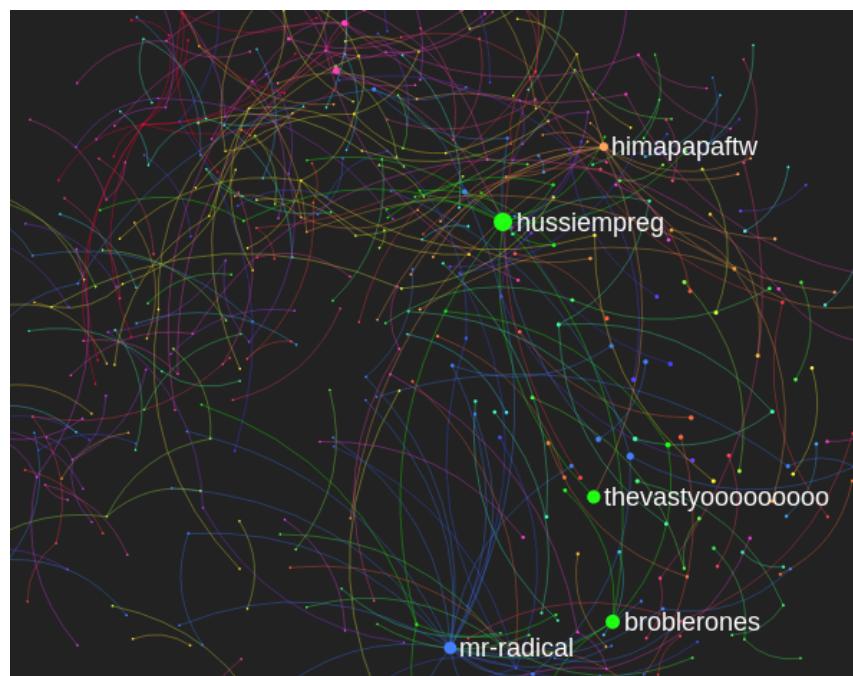


Figura 3.16: Exploración de la red de blogs Tumblr con Sigma.js [LAB]

En la imagen 3.17 podemos observar algunos ejemplos gráficos extraídos de su repositorio [Con].

### Demos

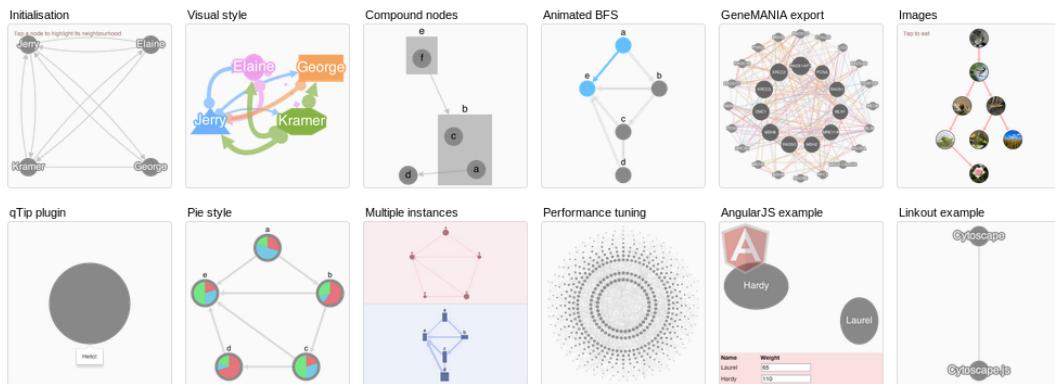


Figura 3.17: Ejemplos visuales uso de Cytoscape [Con]

#### 3.6.1.3. D3.js

D3.js es una biblioteca Open Source de JavaScript para la manipulación de documentos basados en datos. Apoyándose en HTML5, SVG y CSS ayuda a mostrar a través de elementos visuales y manipular los datos de forma dinámica en aplicaciones web.

Una de sus bases es mantener los estándares facilitando de esta forma que el software desarrollado con él sea compatible con los navegadores actuales.

La funcionalidad de D3.js se basa en la conexión de la información con el DOM y la aplicación de transformaciones sobre el documento basado en unos en datos proporcionados. En el código 2 podemos ver cómo de sencillo y simple se relacionan los datos con los componentes del DOM. En la imagen 3.18 el resultado interno de esta vinculación.

```
1 >>> my_data = [20, 7, 32]
2 [20, 7, 32]
3 >>> d3.selectAll('.box').data(my_data).text( function(d) { return d } )
4 [[div.box, div.box, div.box]]
```

Código 2: Ejemplo de vinculación de datos con el DOM usando D3.js [euS12]

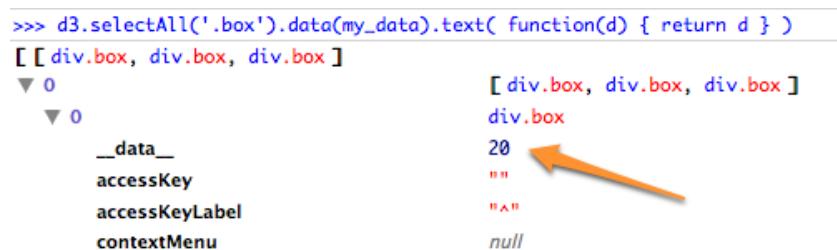


Figura 3.18: Exploración del DOM después de la vinculación de datos [euS12]

Esta librería soluciona de forma eficiente la manipulación de documentos basados en datos y gracias a ello, permite que comportamientos dinámicos y de animación en los elementos visuales.

Todo esto junto con la claridad y calidad del código y la documentación de D3.js facilita enormemente creación y reutilización de componentes visuales. Dentro de la biblioteca se encuentran ya definidos y extensamente documentados numerosos plugins visuales que ayudan para desarrollar nuestros propios elementos, adaptándose fácilmente a las necesidades de cada situación.

Uno de los puntos más fuertes de esta librería es su interacción entre los componentes y la facilidad de adaptación para usarse con otros sistemas. Por ejemplo, hacer funcionar D3.js con la biblioteca para el uso de mapas Leaflet (imagen 3.19) o con las gráficas de Crossfilter (imagen 3.20) es relativamente trivial.

Por otra parte, el gran número de ejemplos públicos desarrollado por terceros publicados bajo un mismo patrón bajo la dirección <http://bl.ocks.org/> hace que éstos puedan visualizarse directamente en la web y junto respaldado por una muy activa comunidad hacen que la labor del desarrollador sea productiva y eficaz. En la figura 3.21 podemos observar uno de estos ejemplos aplicado a grafos.

D3.js trabaja sobre SVG por defecto para sus librerías gráficas. Esto tiene el inconveniente de que elementos visuales muy complejos tiene un bajo rendimiento.

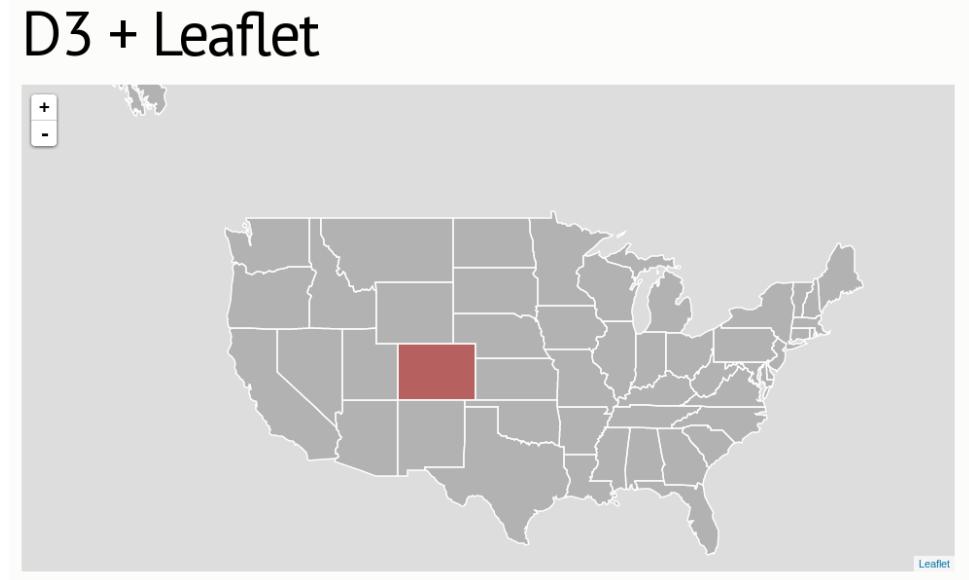


Figura 3.19: Ejemplo de integración de Leaflet con D3.js usando mapas de GeoJSON [Bos11]

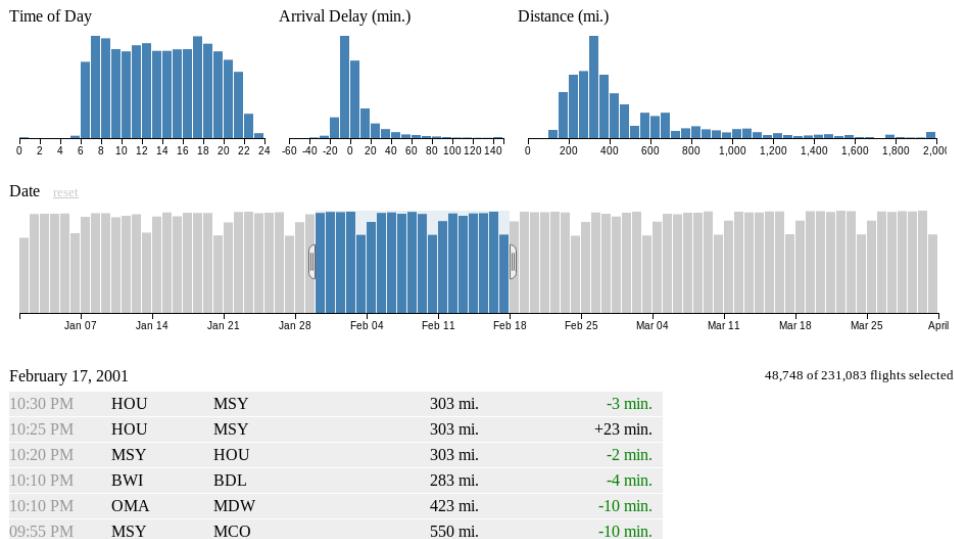


Figura 3.20: Ejemplo de funcionamiento de Crossfilter [Sou]

### 3.6.2. Elección del índice de búsqueda

Para el proyecto KnowledgeFinder utilizó un índice de búsqueda Lucene en su versión 3.0.2. Dadas las restricciones provenientes de SC-VSS el índice de búsqueda del proyecto KnowledgeFinder II deberá basarse en éste.

Por este motivo, la única posible elección es la utilización, sobre una instancia de Lucene, de la plataforma de búsqueda Solr o directamente el propio Lucene.

Solr amplia el potencial de Lucene mejorando principalmente las siguientes funcionalidades:

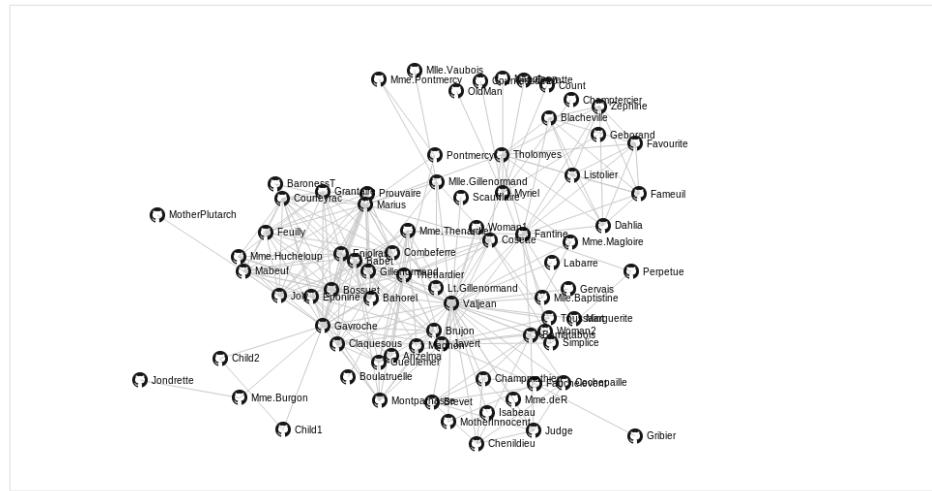


Figura 3.21: Ejemplo de grafo en D3.js publicado en <http://bl.ocks.org/mbostock/950642>

- Opciones más potentes para Full-Text.
- Optimizado para gran volumen de tráfico web.
- Interfaces estandarizadas de XML, JSON y Http.
- Panel intuitivo de administración.
- Estadísticas del servidor para monitorización en JMX.
- Escalable linealmente, replicación automática, tolerancia a fallos y restauración automática [Wikib].
- Cercano a la indexación en tiempo real.
- Flexible y fácilmente adaptable usando configuración en formato XML.
- Arquitectura ampliable de plugins.

### 3.6.3. Elección del tipo de aplicación

Las dos opciones planteadas como posibles tipos de aplicación dependiendo el tipo de arquitectura a implantar.

#### 3.6.3.1. Aplicación Enriquecida de Internet (RIA)

Como se explicó en la sección 3.1.2 el framework KnowledgeFinder utiliza para su desarrollo una arquitectura RIA usando los componentes de Vaadin 6.8.14 .

Para poder representar los elementos gráficos anteriormente descritos, es necesario la actualización a la versión actual de Vaadin y el desarrollo de los nuevos componentes y de esta forma adaptarlos al modelo arquitectónico definido por RIA.

En este modelo en el cliente, el navegador web, no se realiza cálculos. Estos son diferidos a través de AJAX al servidor que es el encargado de toda la lógica de la aplicación. Esto conlleva a que la mayoría de las interacciones del usuario con la aplicación necesite una consulta al servidor web.

### 3.6.3.2. Modelo Vista Controlador (MVC)

Este planteamiento de arquitectura web dista del planteado en el proyecto predecesor KnowledgeFinder. Se basa en separar en tres partes la aplicación:

- **El Modelo**

Representa la información con la que el sistema trabaja. En el caso de KnowledgeFinder II sería principalmente el índice de búsqueda.

- **El Controlador**

Responde a los eventos invocados desde la vista por, normalmente, el usuario e invoca las peticiones al modelo. En nuestro caso sería un servicio web que serviría de intermediario entre la la vista y el índice de búsqueda. Este servicio web estaría integrado en el propio Liferay para definir su comportamiento dependiendo del usuario actual de la plataforma.

- **La Vista**

Es la representación del modelo en un formato para interactuar, por ejemplo en una interfaz de usuario. Para el proyecto KnowledgeFinder II la vista representaría la página web donde se aloja la visualización y todos sus componentes.

## 3.7. Solución Propuesta

La elección del software a emplear para la implementación de KnowledgeFinder II se trabajó muy estrechamente con FW y VF. Su feed-back fue decisivo para la toma de decisiones a nivel de usuario.

Respecto al desarrollo y el tipo de tecnología a desarrollar, el SC-VSS como futuro heredero del proyecto estimó que las soluciones planteadas fuesen compatibles con la motivación y erudición de los componentes del departamento.

### 3.7.1. Elección de la tecnología de visualización

El proceso de estudio comenzó con pruebas y demostraciones de las tecnologías disponibles para la visualización del conocimiento, el punto más influyente para VF y FW. Cómo resultado de este estudio, realizado en la fase de “Extended Static Prototype” (apartado 2.2.0.3), se optó por la tecnología de D3.js. Los puntos decisivos para esta decisión fueron:

- Facilidad de personalizar el comportamiento de los componentes gracias a SVG.
  - Calidad de la documentación y ejemplos ilustrativos de su uso.
  - Compatibilidad entre los plugins actualmente definidos.
  - Calidad del software. item Facilidad de adaptación a las actuales y futuras necesidades específicas de KnowledgeFinder II.
-

### 3.7.2. Elección del índice de búsqueda

Tras un desacuerdo y una fuerte oposición inicial por parte del SC-VSS en cambiar el núcleo del sistema de indexación por motivos temporales y de costes adicionales, se optó finalmente por la redefinición del sistema de importación e índice de búsqueda usando Solr. Como puntos claves de esta decisión fueron las siguientes propiedades de Solr:

- Unificación de los proyectos Lucene y Solr en marzo de 2010.
- Sistema de configuración de Solr basado en ficheros XML.
- Opciones más potentes para Full-Text.
- Optimizado para gran volumen de tráfico web.
- Múltiples clientes para el acceso a Solr con diferentes lenguajes de programación. Por ejemplo, Solrj proporciona una API para Java que permite añadir, actualizar y consultar el índice de Solr.
- Capacidad de crear clusters de servidores para una mejor tolerancia a fallos.
- Herramientas de monitorización.

### 3.7.3. Elección del tipo de aplicación

Durante la fase de “Extreme Prototype” (apartado 2.2.0.3) se demostró que con una API de un servicio web relativamente simple se conseguía una visualización muy satisfactoria para el usuario.

Nuevamente, se encontró con un primer rechazo por parte del SC-VSS de cambiar el tipo de aplicación totalmente. Esto conllevaba a descartar totalmente Vaadin como herramienta, replanteamiento del framework KnowledgeFinder II para los proyectos de próxima implantación (ELIB-Portal, STRADA@DLR, ...) y la implementación partiendo desde cero del software.

A pesar de ser un gran cambio respecto al pensamiento inicial de SC-VSS de cómo se desarrollaría KnowledgeFinder II, se optó por la implantación del Modelo Vista Controlador con un servicio web en base a los siguientes argumentos:

#### ▪ Costosa y necesaria actualización de Vaadin

La versión de Vaadin usada para el proyecto KnowledgeFinder se encuentra desactualizada y sin mantenimiento. Para un correcto y eficiente funcionamiento con los sistemas complejos de visualización la nueva API de JavaScript de Vaadin 7 sería necesaria.

#### ▪ Difícil incorporación de los elementos de D3.js en Vaadin

Incluso la nueva versión del framework Vaadin no proporciona ningún componente gráfico útil para la visualización de KnowledgeFinder II. Aún existiendo algunos proyectos inacabados como gwt-d3 [gd], la integración de los componentes de D3.js sería demasiado costosa.

#### ▪ Incierto rendimiento adecuado de la aplicación usando RIA

Con D3.js se producen su alto número de peticiones al servidor. Con una arquitectura RIA esto se traduciría en una posible saturación de la parte del servidor y una lenta reacción de la interfaz de usuario.

- **Flexibilidad de MVC**

El patrón MVC se basa en las ideas de reutilización de código y la separación de la aplicación (modelo, vista, controlador) para un fácil desarrollo y mantenimiento. En el caso de KnowledgeFinder II, esta arquitectura hace además que la adaptación y ampliación del framework sea más sencilla y elegante.

- **D3.js y los recursos externos**

La API de D3.js facilita el uso de éste con servicios web en múltiples formatos (JSON, XML, fichero de texto, ...) a través de llamadas por AJAX.

- **Liferay y los servicios web**

El impuesto portal Liferay proporciona una interfaz de comunicación por servicios web. Esto hace que la implementación nuevos servicios para KnowledgeFinder II sea relativamente sencilla [Lif14]. Para una comunicación de los componentes JavaScript a través de AJAX es una herramienta muy útil.

# Capítulo 4

## Análisis del Sistema

Esta sección cubre el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.

### 4.1. Modelo Conceptual

En la imagen 4.1 se muestra el diagrama conceptual para KnowledgeFinder II.

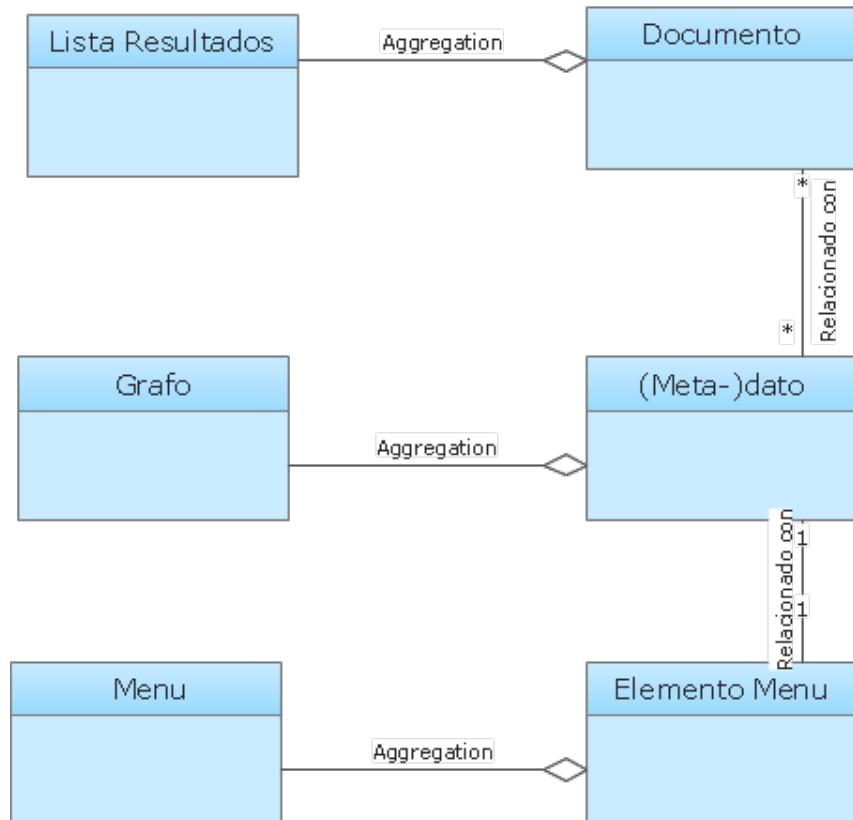


Figura 4.1: Modelo Conceptual de KnowledgeFinder II

## 4.2. Modelo de Casos de Uso

Partiendo de los requisitos funcionales descritos en el apartado 3.5.2 se emplean los casos de uso para describir las interacciones de los actores y el KnowledgeFinder II.

### 4.2.1. Diagramas de Casos de Uso

En este apartado se va a describir algunos de los diagramas de caso de uso del proyecto.

### 4.2.2. Actores

A continuación se exponen los diferentes roles que juegan los usuarios que interactúan con el sistema. Los actores pueden ser roles de personas físicas, sistemas externos o incluso el tiempo (eventos temporales).

#### 4.2.2.1. Administrador del índice de búsqueda

Este actor será el encargado de controlar las importaciones de los datos en Solr. Puede ser una persona física que, a través de la interfaz de Solr, ejecute la importación o una tarea programada, por ejemplo un Cron.

#### 4.2.2.2. Usuario visitante

Este actor utiliza la interfaz de usuario sin estar registrado. No tiene autorización para iniciar sesión en el sistema.

#### 4.2.2.3. Usuario registrado

Este actor tiene permisos especiales concedidos por el DRL para poder entrar en el sistema. Dependiendo de los permisos asignados, este actor dispondrá de un conjunto de documentos y campos extras. El sistema de acceso es externo al sistema.

#### 4.2.2.4. Diagrama Casos de Uso: Uso de la interfaz de usuario (anónimo y registrado)

En la imagen 4.2 se representa los casos de uso para un usuario no registrado en el sistema (usuario visitante) y para uno registrado.

#### 4.2.2.5. Diagrama Casos de Uso: Gestión del índice de búsqueda

En la imagen 4.3 se representa los casos de uso de un administrador del índice de búsqueda.

### 4.2.3. Especificación de los Casos de Uso

#### 4.2.3.1. Caso de Uso: Importación de datos

- **Descripción** El usuario activa la importación de los datos.
  - **Actor/es** Usuario administrador del índice.
  - **Escenario Principal**
-

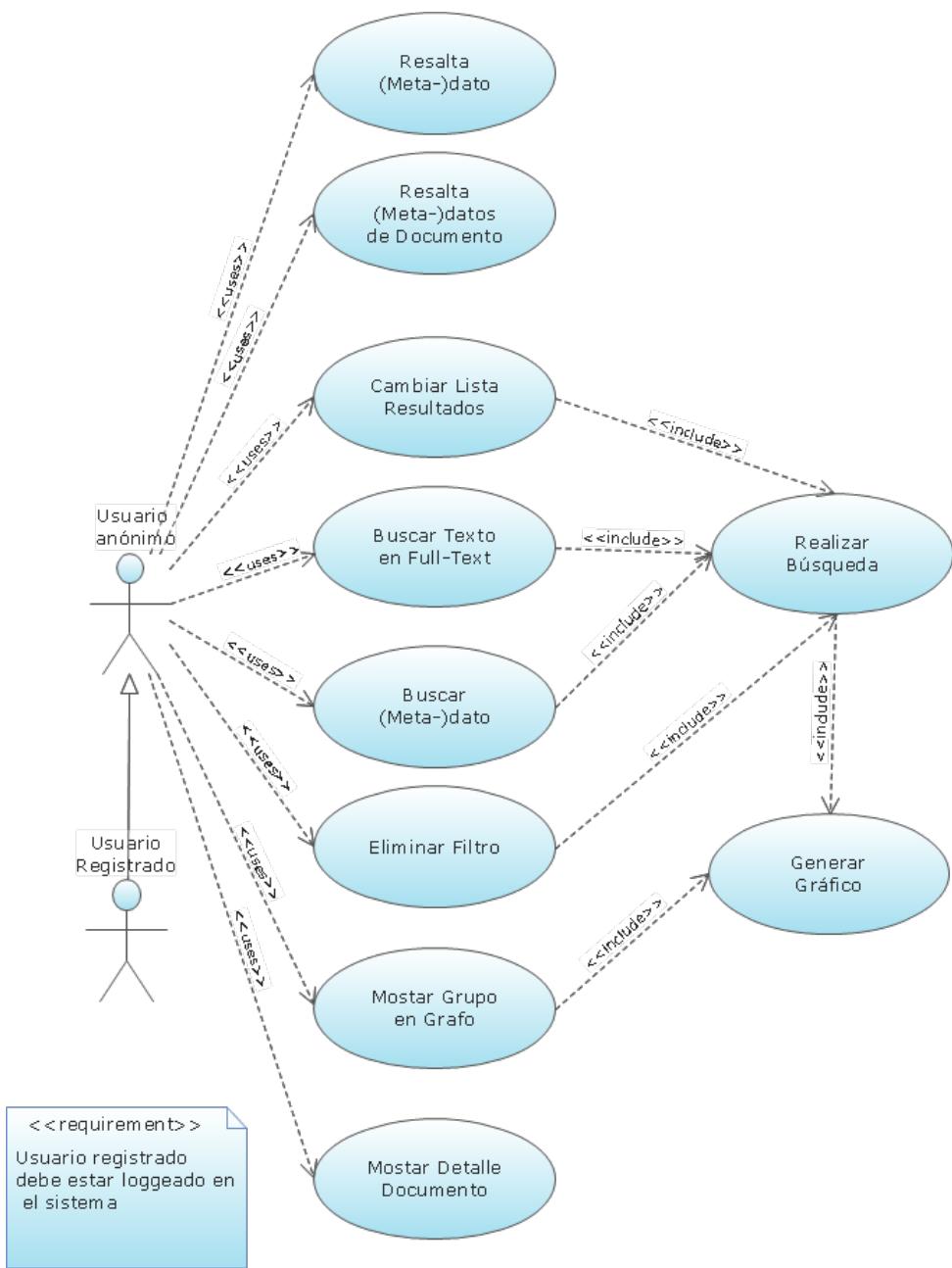


Figura 4.2: Diagrama Casos de Uso: Uso de la interfaz de usuario (anónimo) y usuario registrado



Figura 4.3: Diagrama Casos de Uso: Gestión del índice de búsqueda

1. El usuario selecciona una configuración a ejecutar.
2. El usuario ejecuta importación con la configuración seleccionada.
3. El sistema importa los documentos en el índice de búsqueda.
4. El sistema muestra el resultado de la importación incluyendo los posibles errores.

#### 4.2.3.2. Caso de Uso: Buscar en Full-Text

- **Descripción** El usuario introduce un texto o expresión regular en el campo Full-Text.
- **Actor/es** Usuarios con o sin sesión iniciada.
- **Escenario Principal**
  1. Usuario entra el texto a buscar en el campo Full-Text.
  2. El sistema actualiza la lista de documentos únicamente con los elementos que contienen el texto introducido.
  3. El sistema actualiza el grafo de visualización con los nuevos valores de los (Meta-)datos de los documentos encontrados (actualiza el tamaño de los nodos y de las relaciones entre ellos).
  4. El sistema actualiza los valores de los contadores en el menú.
  5. El sistema añade el valor introducido a la lista de filtros activos.
  6. El sistema sitúa la paginación en su primera página.

#### 4.2.3.3. Caso de Uso: Buscar (Meta-)dato

- **Descripción** El usuario filtra los documentos por un (Meta-)dato a través del grafo de visualización o del menú.
- **Actor/es** Usuarios con o sin sesión iniciada.
- **Escenario Principal**
  1. Usuario pulsa sobre un nodo el nodo del grafo o sobre el elemento de la tabla que desea filtrar.
  2. El sistema actualiza la lista de documentos suprimiendo aquellos que no contienen (Meta-)dato seleccionado.
  3. El sistema actualiza el grafo de visualización con los nuevos valores de los (Meta-)datos de los documentos encontrados (actualiza el tamaño de los nodos y de las relaciones entre ellos).
  4. El sistema actualiza los valores de los contadores en el menú.
  5. El sistema añade el (Meta-)dato a la lista de filtros activos.
  6. El sistema sitúa la paginación en su primera página.
- **Escenario Alternativo**
  - 2.a El (Meta-)dato seleccionado se muestra en el grafo.
    - 1 El sistema elimina el elemento del grafo..
    - 2 Continuar con el paso 3.

#### 4.2.3.4. Caso de Uso: Detalles de documento

- **Descripción** El usuario de interfaz de usuario obtiene los detalles de un documento en una ventana emergente.

- **Actor/es** Usuarios con o sin sesión iniciada.

- **Escenario Principal**

1. Usuario selecciona un elemento de la lista de resultados.
2. El sistema despliega el acordeón del elemento.
3. El usuario pulsa el botón para ver los detalles del documento.
4. El sistema abre una ventana emergente donde muestra los campos permitidos para el usuario.

#### 4.2.3.5. Caso de Uso: Eliminar filtro

- **Descripción** El usuario de interfaz de usuario elimina un filtro de los actuales activos en la lista filtros activos.

- **Actor/es** Usuarios con o sin sesión iniciada.

- **Escenario Principal**

1. Usuario selecciona un elemento de la lista formada por los filtros que con anterioridad fueron activados.
2. El sistema actualiza la lista de documentos aplicando el resto de filtros que todavía siguen activos.
3. El sistema actualiza el grafo de visualización con los nuevos valores de los (Meta-)datos de los documentos encontrados (actualiza el tamaño de los nodos y de las relaciones entre ellos).
4. El sistema actualiza los valores de los contadores en el menú.
5. El sistema suprime el filtro la lista de filtros activos.
6. El sistema sitúa la paginación en su primera página.

#### 4.2.3.6. Caso de Uso: Cambiar lista de resultados

- **Descripción** El usuario de interfaz de usuario modifica la lista de los resultados cambiando la paginación o el criterio de ordenación.

- **Actor/es** Usuarios con o sin sesión iniciada.

- **Escenario Principal**

1. Usuario selecciona una página de la lista de paginación.
2. El sistema actualiza la lista de documentos mostrando únicamente los elementos de esa página.

- **Escenario Alternativo**
-

- 2.a El usuario selecciona otro criterio de ordenación para la lista de resultados.
  - 1 El sistema actualiza la lista de documentos mostrándolos en el orden seleccionado, sin cambiar la página donde se encontraba anteriormente.

#### 4.2.3.7. Caso de Uso: Resaltar (Meta-)dato

- **Descripción** El usuario de interfaz de usuario resalta las relaciones de un (Meta-)dato.
- **Actor/es** Usuarios con o sin sesión iniciada.
- **Escenario Principal**
  1. Usuario se sitúa sobre un (Meta-)dato en el grafo o en el menú.
  2. El sistema muestra menos nítido los elementos del grafo que no tienen relación con este (Meta-)dato.
  3. El sistema muestra resaltado los documentos de la actual página de resultados que contienen este (Meta-)dato.
  4. El sistema muestra una pequeña ventana con información adicional sobre este (Meta-)dato.

- **Escenario Alternativo**

- 2.a El (Meta-)dato seleccionado tiene subelementos en el menú.
  - 1 El sistema muestra menos nítido los elementos del grafo que no tienen relación con ningún subelemento.
  - 2 El sistema muestra resaltado los documentos de la actual página de resultados que contienen algún subelemento.

#### 4.2.3.8. Caso de Uso: Resaltar (Meta-)datos de documento

- **Descripción** El usuario de interfaz de usuario resalta las relaciones de un documento.
- **Actor/es** Usuarios con o sin sesión iniciada.
- **Escenario Principal**
  1. Usuario se sitúa sobre un documento de la lista de resultados.
  2. El sistema muestra menos nítido los elementos del grafo que no tienen relación con este documento.
  3. El sistema muestra resaltado en el menú los elementos los (Meta-)datos del documento.

#### 4.2.3.9. Caso de Uso: Mostrar grupo en grafo

- **Descripción** El usuario de interfaz de usuario selecciona un grupo para mostrar en el grafo.
  - **Actor/es** Usuarios con o sin sesión iniciada.
  - **Escenario Principal**
-

1. Usuario selecciona un grupo del menú.
2. El sistema añade los subelementos del grupo en el grafo.

- **Escenario Alternativo**

- 2.a El grupo seleccionado ya se está mostrando en el menú.
  - 1 El sistema elimina del grafo los subelementos del grupo.

### 4.3. Modelo de Comportamiento

Partiendo de los casos de uso anteriores, se detallan ahora los modelos de comportamiento asociados.

#### 4.3.0.10. Caso de Uso: Importación de datos

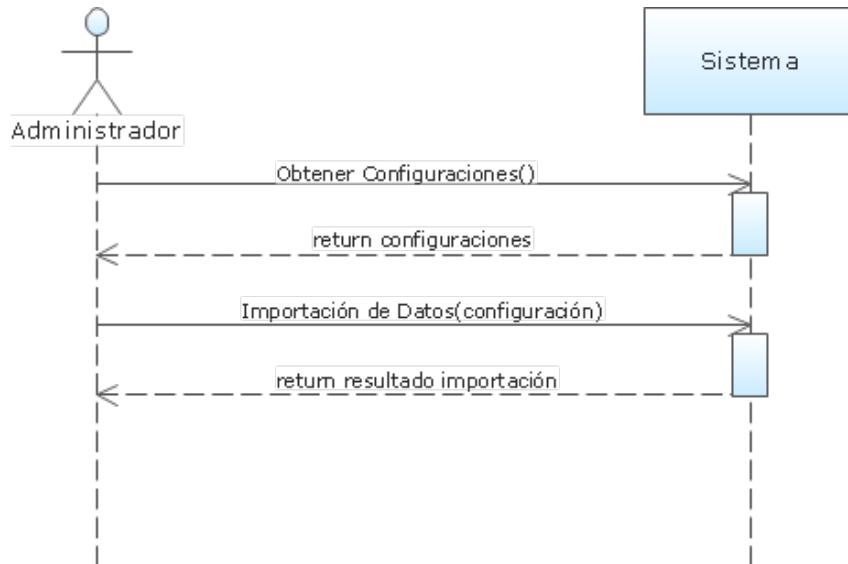


Figura 4.4: Modelo de Comportamiento Caso de Uso: Importación de datos

#### 4.3.0.11. Caso de Uso: Buscar en Full-Text

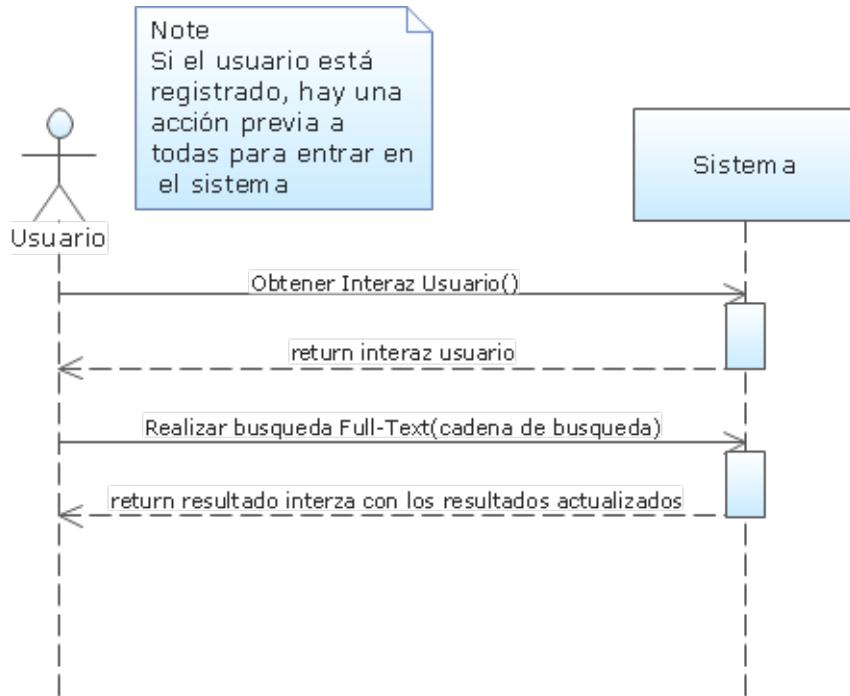


Figura 4.5: Modelo de Comportamiento Caso de Uso: Buscar en Full-Text

#### 4.3.0.12. Caso de Uso: Buscar (Meta-)dato

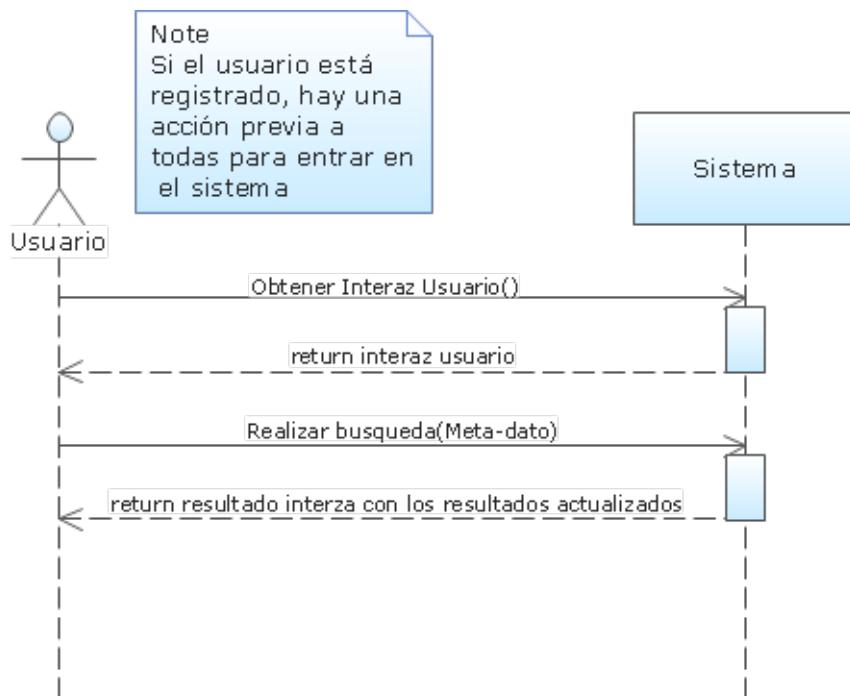


Figura 4.6: Modelo de Comportamiento Caso de Uso: Buscar (Meta-)dato

#### 4.3.0.13. Caso de Uso: Detalles de documento

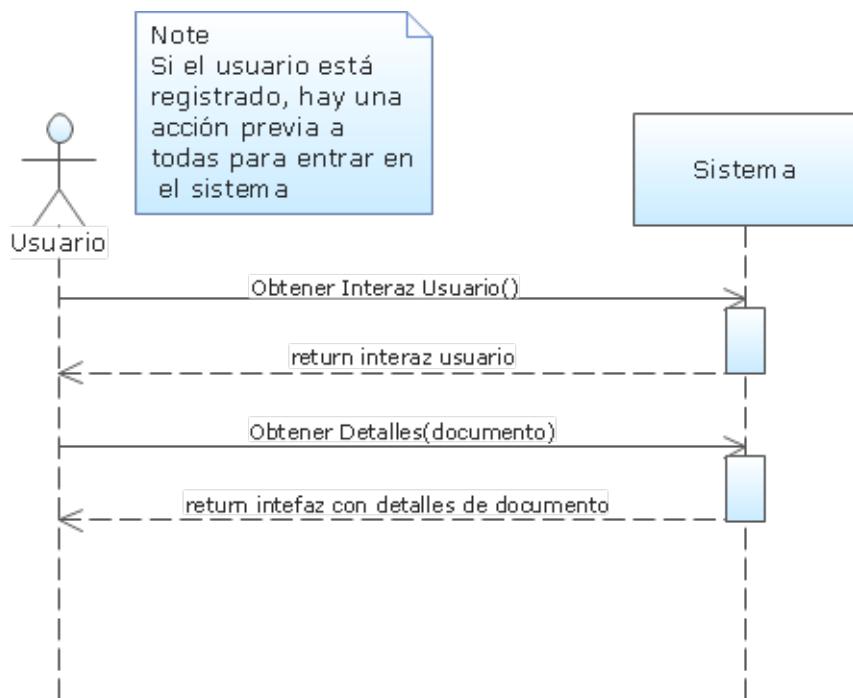


Figura 4.7: Modelo de Comportamiento Caso de Uso: Detalles de documento

#### 4.3.0.14. Caso de Uso: Eliminar filtro

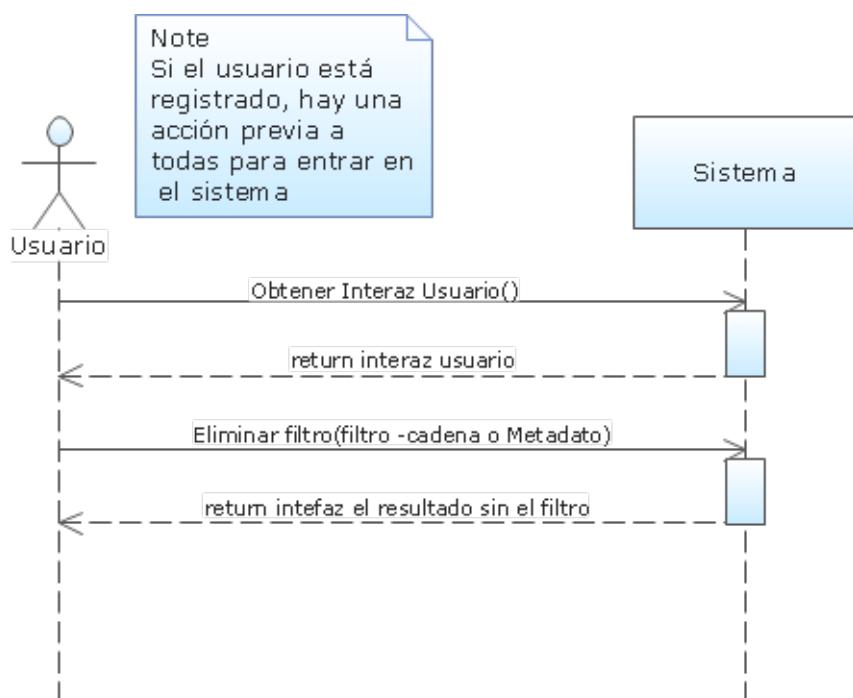


Figura 4.8: Modelo de Comportamiento Caso de Uso: Eliminar filtro

#### 4.3.0.15. Caso de Uso: Cambiar lista de resultados

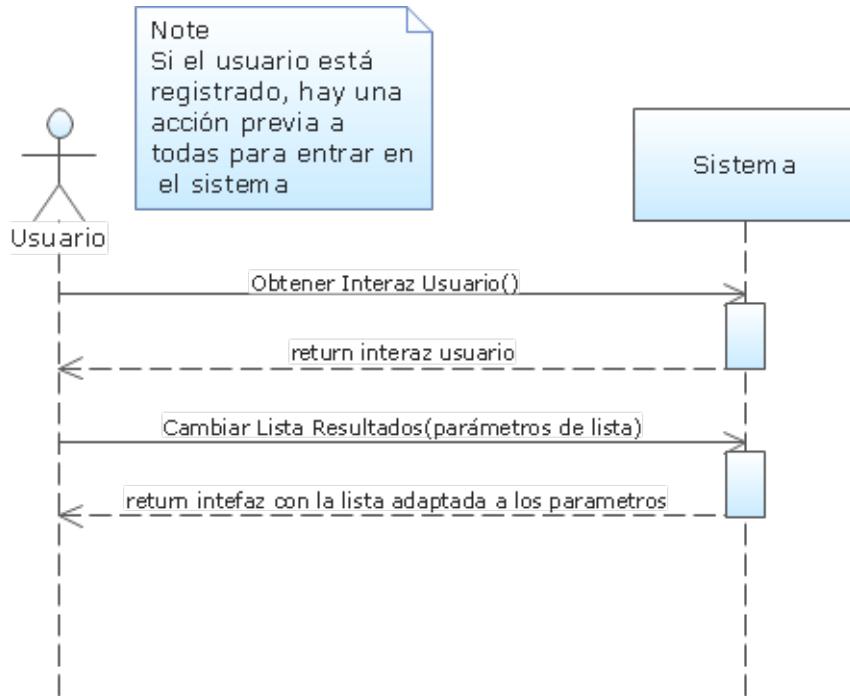


Figura 4.9: Modelo de Comportamiento Caso de Uso: Cambiar lista de resultados

#### 4.3.0.16. Caso de Uso: Resaltar (Meta-)dato

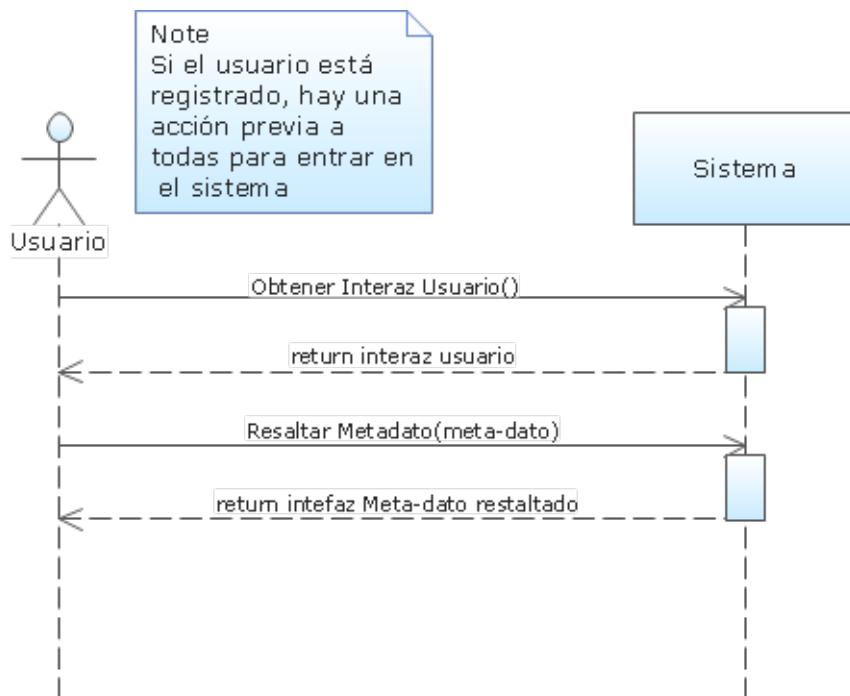


Figura 4.10: Modelo de Comportamiento Caso de Uso: Resaltar (Meta-)dato

#### 4.3.0.17. Caso de Uso: Resaltar (Meta-)datos de documento

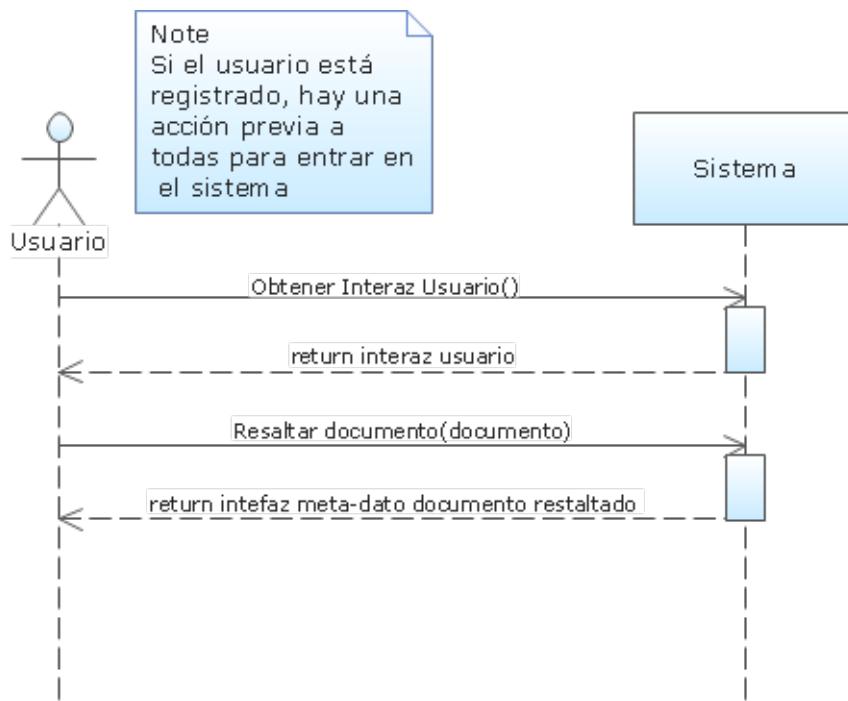


Figura 4.11: Modelo de Comportamiento Caso de Uso: Resaltar (Meta-)datos de documento

#### 4.3.0.18. Caso de Uso: Mostrar grupo en grafo

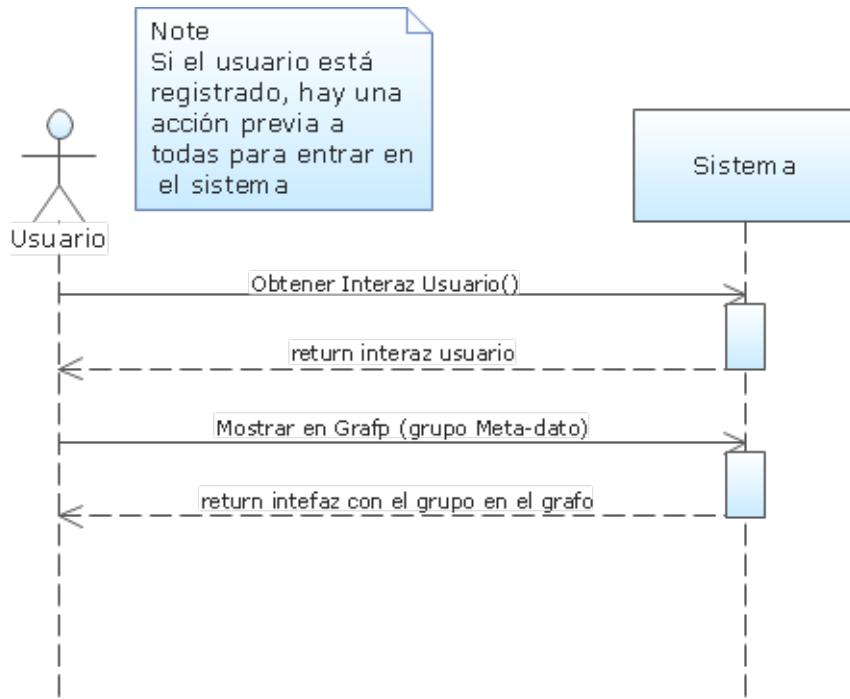


Figura 4.12: Modelo de Comportamiento Caso de Uso: Mostrar grupo en grafo

## 4.4. Modelo de Interfaz de Usuario

A continuación mostramos unos prototipos de baja fidelidad de la interfaz de usuario. En la imagen 4.13 podemos ver la estructura de la pantalla general de la aplicación.

Cuando un usuario visualiza los detalles de un documento la pantalla tiene una semejanza a la expuesta en la imagen 4.14.

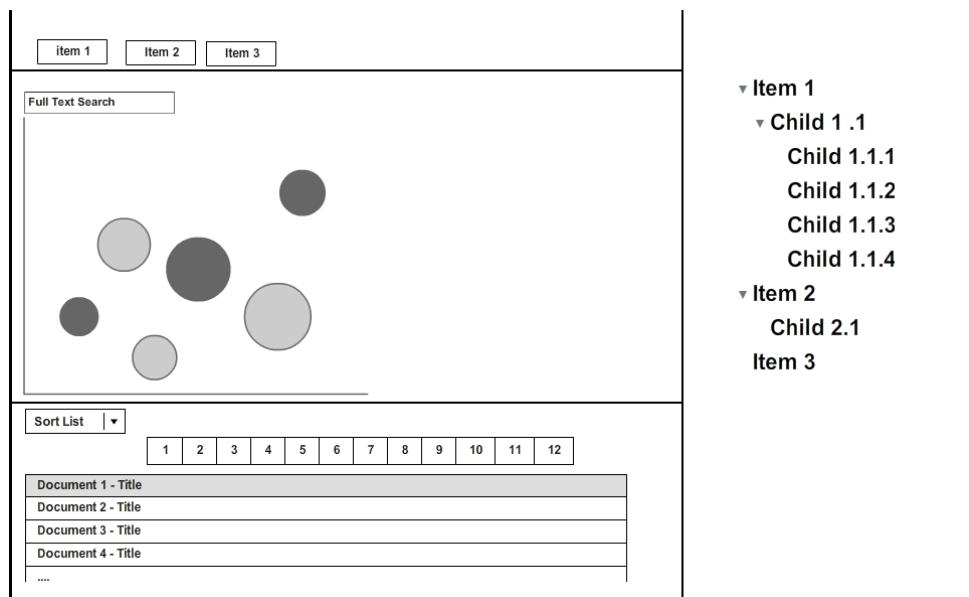


Figura 4.13: Mockup - pantalla general KnowledgeFinder II

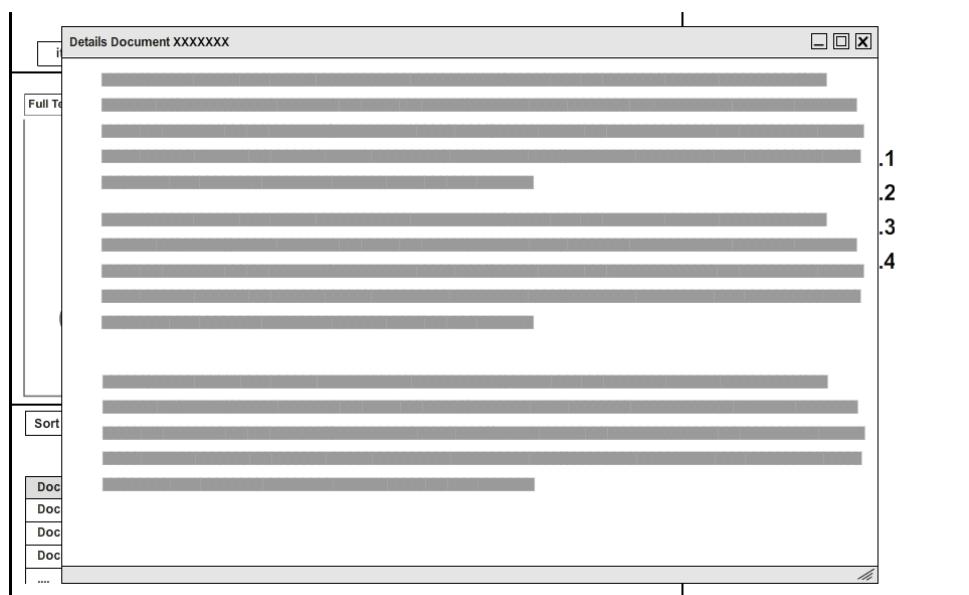


Figura 4.14: Mockup - pantalla de detalles en KnowledgeFinder II

# Capítulo 5

## Diseño del Sistema

### 5.1. Arquitectura del Sistema

En este apartado de la documentación se expone la arquitectura general del proyecto KnowledgeFinder II, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

#### 5.1.1. Ámbito y Contexto del Sistema

En la figura 5.1 se representa la arquitectura de KnowledgeFinder II en su contexto. En ella se aprecia cómo esta aplicación interacciona con sus entidades externas.

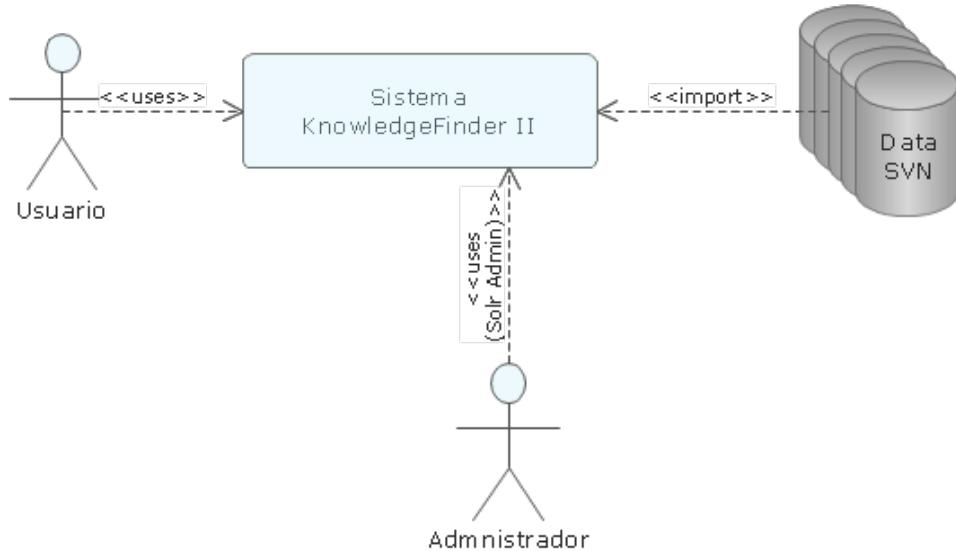


Figura 5.1: Diagrama del ámbito y contexto de la arquitectura del sistema

#### 5.1.2. Arquitectura Física

A continuación se va a describir el sistema de hardware necesario que conforman la arquitectura física de nuestro sistema. En la figura 5.2 podemos diferenciar los siguientes componentes:

- **Cliente**

Representa el hardware que utiliza el usuario para acceder a los servicios ofrecidos por KnowledgeFinder II. Éstos están disponibles para ser usados por un navegador web o por un software que permita el acceso a servicios web.

- **Servidor para Liferay (Tomcat)**

Para alojar la instancia de Liferay que contendrá el servicio web y el portlet para la web de KnowledgeFinder II es necesario el hardware apropiado para contener un servidor de Tomcat.

- **Servidor para Solr**

El motor de búsqueda Solr puede estar instalado en varios tipos de servidores (Tomcat, Jetty, ...). Para ello se requiere hardware que pueda proporcionar alguno de estos. Solr puede residir, por ejemplo, en la misma instancia de Tomcat donde se encuentra Liferay en ejecución.

Este servidor debe tener acceso a los repositorios remotos o a una copia de trabajo local de SVN para el proceso de importación de los datos.

- **Control de acceso (nota)**

Como medida de seguridad, se recomienda proteger el acceso a la instancia de Solr [Wikc]. Para ello se pueden disponer de varias medidas de seguridad, por ejemplo un firewall.

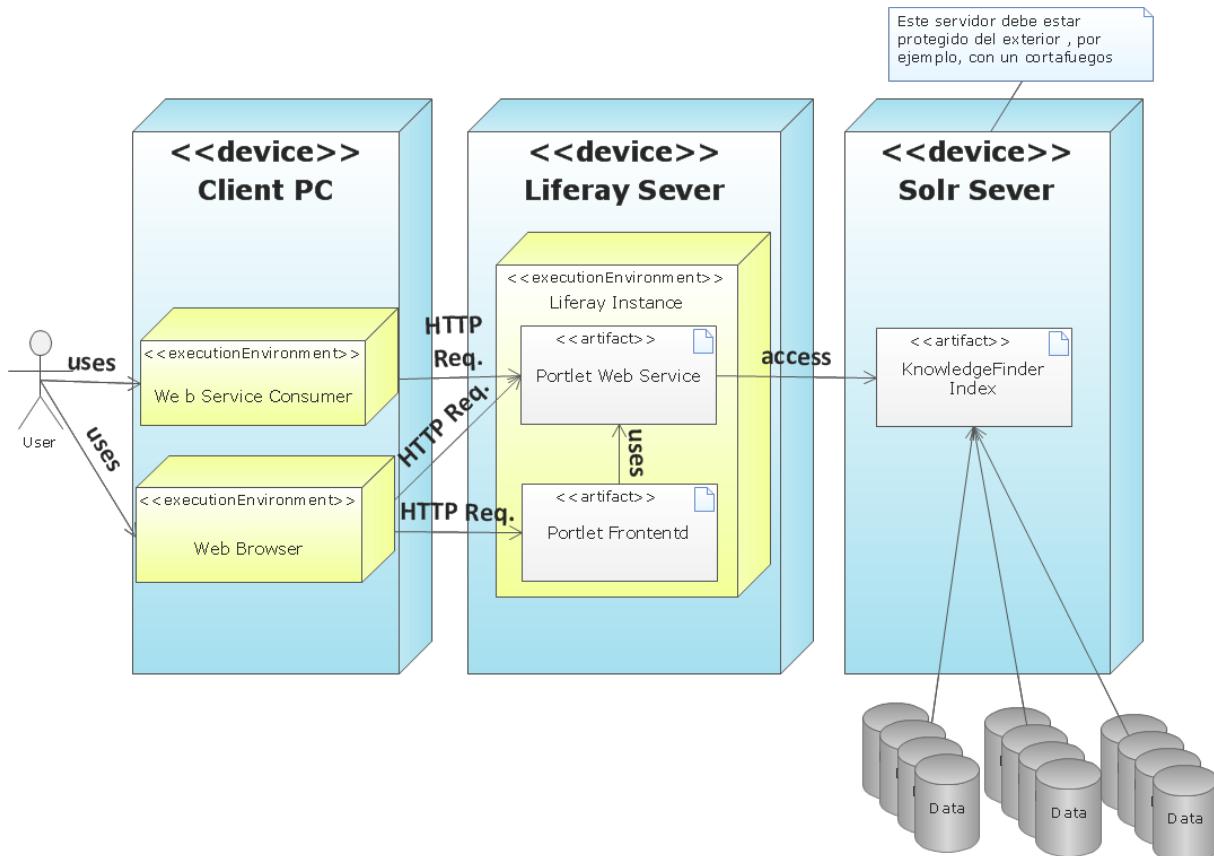


Figura 5.2: Diagrama de despliegue para el proyecto KnowledgeFinder II

### 5.1.3. Arquitectura Lógica

La arquitectura de diseño especifica la forma en que los artefactos software interactúan entre sí para lograr el comportamiento deseado en el sistema. En esta sección se muestra la comunicación entre el software base seleccionado, los componentes reutilizados y los componentes desarrollados para cumplir los requisitos de la aplicación. También se recogen los servicios de sistemas externos con los que interactúa nuestro sistema.

### 5.1.4. Modelo Vista Controlador en KnowledgeFinder II

Como se expuso en apartado 3.6.3.2, el proyecto KnowledgeFinder II sigue una arquitectura de software basada en el MVC. A continuación se detalla cómo es el diseño aplicando esta arquitectura.

- **El Modelo**

En el caso de KnowledgeFinder II sería principalmente el índice de búsqueda. La información está almacenada en éste y es accesible gracias a la API que proporciona el motor de búsqueda.

- **El Controlador**

El intermediario entre la vista y el índice de búsqueda es un servicio web alojado en Liferay. De esta forma, el controlador tiene acceso a los roles del usuario definidos en Liferay que actualmente está usando la aplicación para definir su comportamiento. La respuesta del servicio web variará para distintas combinaciones de roles de usuarios.

- **La Vista**

Para el proyecto KnowledgeFinder II la vista representaría la página web donde se aloja la visualización y todos sus componentes. Es el interfaz principal del usuario para interactuar con el sistema.

### 5.1.5. Desglose de Componentes

Basándonos en el esquema expuesto en [DGS] vamos a describir la arquitectura del software de KnowledgeFinder II por vistas de construcción. Comenzando por un nivel de abstracción donde se muestran los artefactos que conforman el sistema, se desglosará cada uno de ellos en sus componentes internas.

#### 5.1.5.1. Nivel 1 - Building View

En la imagen 5.3 se puede ver como interactúa las distintos componentes del sistema KnowledgeFinder II. En ella también se aprecian las interfaces externas y los tres subsistemas principales que serán descritos en el siguiente nivel.

- **Panel de administración de Solr**

Interfaz para activar la importación de los datos.

- **Repositorios SVN**

Conjunto de repositorios SVN desde donde se obtienen los datos.

---

- **Navegador de cliente**

El usuario final interacciona directamente con el sistema usando un navegador web.

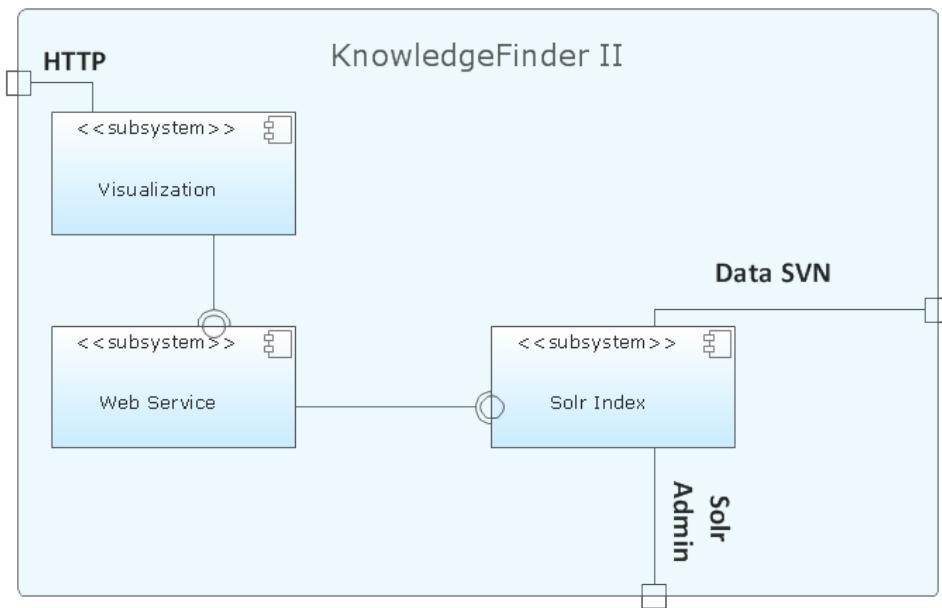


Figura 5.3: Nivel 1 - Building View KnowledgeFinder II

#### 5.1.5.2. Nivel 2 - Building View

##### Índice de búsqueda en Solr

Este subsistema es responsable de la importación de la información para el índice de búsqueda de Solr desde los repositorios SVN. También se encarga de proporcionar estos datos importados a través de una API.

Este subsistema se descompone en los siguientes componentes representados en la imagen 5.4:

- **SVN Crawler**

Componente encargado de extraer los ficheros de SVN junto con todas sus propiedades. Recorre todo el repositorio (local o remoto) y empaqueta todos los elementos del repositorio.

- **SVN DataPicker**

Componente encargado de transformar las extraer las propiedades de los elementos proporcionados por el crawler y empaquetarlos en objetos iterables de Java.

- **SVN Parser**

Extrae las propiedades del SVN en formato JSON y las prepara para la importación.

- **SVN DataImport**

Se encarga de configurar y coordinar el proceso de importación.

- **Solr Transformers**

Transforma los datos durante la importación para adaptarlo a los requisitos específicos del índice.

- **Solr**

Instancia de Solr donde reside el índice importado. Proporciona la interfaz para importar y consultar los datos a través de una API y un interfaz de usuario web para su administración.

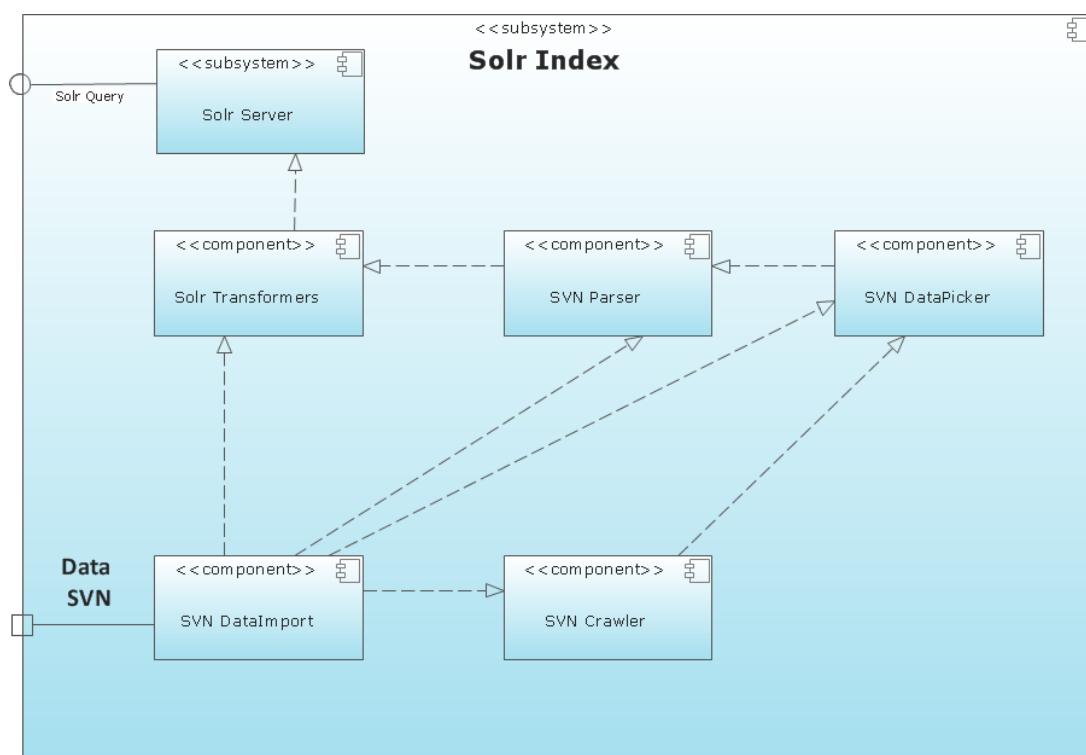


Figura 5.4: Nivel 2 - Building View Índice Solr

### Servicio web

Este subsistema provee un servicio web que permite acceder a los datos proporcionados por el índice de búsqueda Solr. Como muestra la imagen 5.5, se descompone en los siguientes elementos:

- **KnowledgeFinder Service**

Encargado de recibir las peticiones del servicio web (get-document, get-nodes) y adaptar sus parámetros a la API de Solr. Como resultado de las peticiones, crea una respuesta en formato JSON con los valores pertinentes.

- **User-Query Factory**

Elemento encargado de transformar las peticiones a la API de Solr en función de los roles del usuario actual.

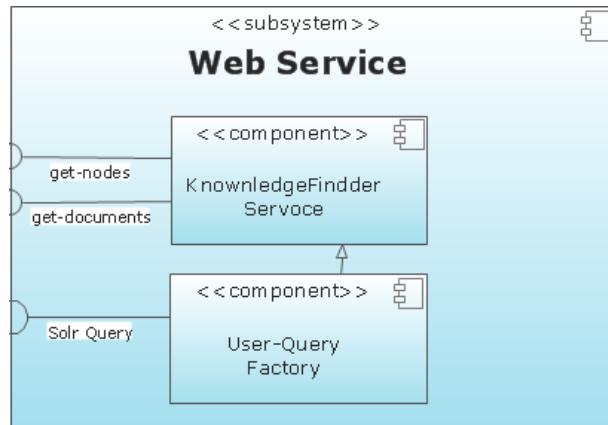


Figura 5.5: Nivel 2 - Building View Servicio web

### Visualización

El subsistema de visualización se encarga de proporcionar la interfaz de usuario a través de elementos gráficos. A grandes rasgos, muestra la información obtenida a través del servicio web de forma intuitiva y atractiva para el usuario. En la imagen 5.6 se encuentran reflejados sus componentes y las relaciones entre ellos:

- **TemplateController**

Controlador que genera una página de JSP para una URL definida por el administrador.

- **Filters Loader**

Este componente crea los filtros necesarios a partir de los (Meta-)datos obtenido en peticiones al servicio web.

- **Página HTML**

La página JSP obtiene la información del TemplateController genera una página HTML. Con los Filters Loader crea variables que son necesarias por todos los componentes JavaScript.

- **Main.js**

Encargado de cargar todos los componentes de JavaScript. También coordina los evento del usuario y el comportamiento de los componentes entre ellos.

- **d3.knowledgefinder.graph.js**

Componente encargado de crear un grafo que representa los (Meta-)datos y las relaciones entre ellos.

- **d3.knowledgefinder.modal.js**

Componente que genera una ventana emergente con los detalles de un documento previamente seleccionado de la lista de resultados.

- **d3.knowledgefinder.result.js**

Lista los documentos obtenidos durante la consulta y los organiza usando un sistema de paginación.

- **d3.knowledgefinder.selection.js**

Componente encargado de mostrar una lista de botones con los actuales parámetros de búsqueda. A través de esta lista el usuario puede consultar y desactivar los elementos que componen la búsqueda.

- **d3.knowledgefinder.table.js**

La tabla generada por el componente de Página HTML es manipulada a través de este componente de acuerdo a los resultados y búsquedas.

- **d3.knowledgefinder.urlquery.js**

Componente auxiliar para manipular los parámetros y facilitar al resto de componentes de la visualización la comunicación con el servicio web.

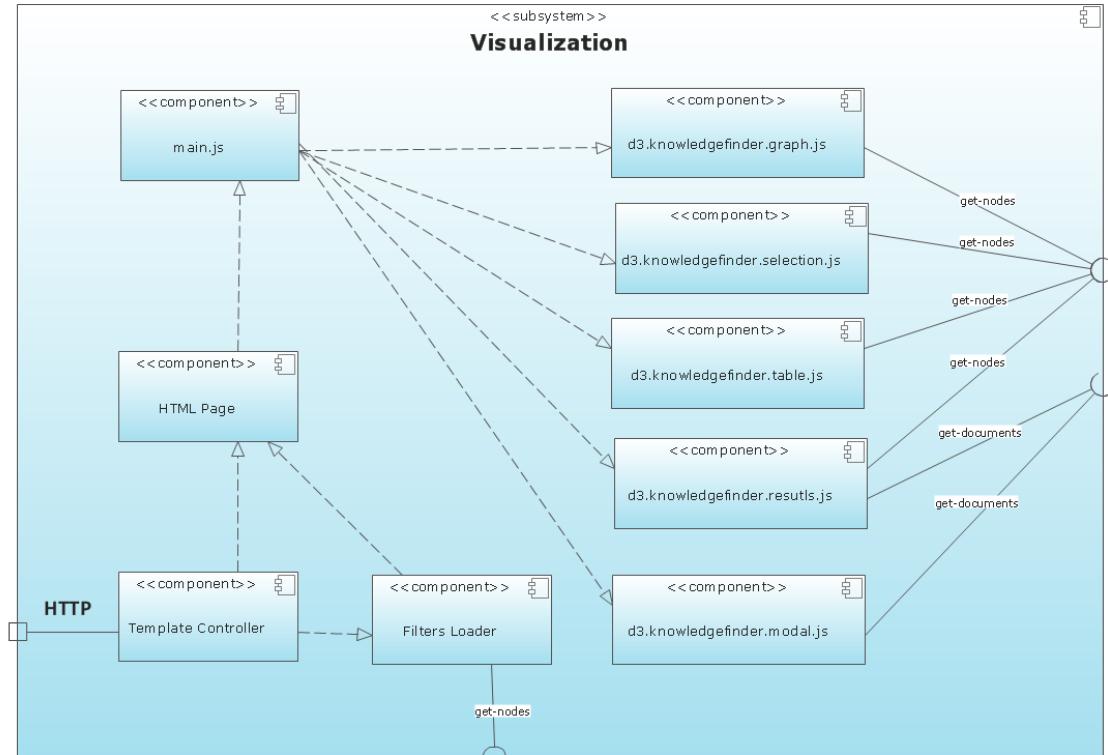


Figura 5.6: Nivel 2 - Building View Visualización

## 5.2. Diseño Físico de Datos

La estructura física de los datos que utiliza el sistema es el índice de búsqueda de Solr. Su definición se describe en el esquema. Estos esquemas deben definirse dependiendo de las necesidades de cada instancia del framework KnowledgeFinder II.

A continuación se detalla en la tabla 5.1 en detalle la estructura del índice de búsqueda una vez que la importación ha finalizado (instancia del proyecto STRADA@DLR de KnowledgeFinder II). En ella se puede observar, entre otros datos, qué campos se utilizan para la búsqueda Full-Text, los campos que son creados en el momento de la importación (dinámicos), qué campos son usados para la búsqueda en el índice, etcétera .

Campo	Obligatorio	Tipo	Multiple	Index	Stored	Notas
id	X	string		X	X	Unique Key
title	X	text		X	X	Full-Text
description		text			X	Full-Text
keywords		string	X	X	X	Full-Text
categories		string	X		X	
filePath		string			X	
institute		string			X	Generado durante la importación
timePeriodOfDataCollection		string			X	
startTimePeriodOfDataCollection		date		X	X	
endTimePeriodOfDataCollection		date		X	X	
startDateDate		date		X	X	
temporalCoverage		string			X	
startTemporalCoverage		date		X	X	
endTemporalCoverage		date		X	X	
spatialCoverage		string	X		X	
world		string	X	X		Generado desde spatialCoverage
regionsWorld		string	X	X		Generado desde spatialCoverage
countries		string	X	X		Generado desde spatialCoverage
regions		string	X	X		Generado desde spatialCoverage
size_author		int			X	Generado desde authors
contentcreationdatetime		int		X		Propiedad interna de SVN
contentmodificationdatetime		int		X		Propiedad interna de SVN
author_uid_*		string			X	Generado desde authors (campo dinámico), Full-Text
author(firstName_*		string			X	Generado desde authors (campo dinámico), Full-Text
author(lastName_*		string			X	Generado desde authors (campo dinámico), Full-Text
author(mail_*		string			X	Generado desde authors (campo dinámico), Full-Text
author(organization_*		string			X	Generado desde authors (campo dinámico), Full-Text
category_*		string	X	X	X	Generado desde categories (campo dinámico), Full-Text

Tabla 5.1: Tabla resumen del esquema de KnowledgeFinder II de Solr para el proyecto STRADA@DLR

### 5.3. Vista de Procesos

Siguiendo nuevamente el esquema expuesto en [DGS], en esta sección se describe el comportamiento de los componentes del sistema descritos en la sección 5.1.5 como elementos de ejecución.

A continuación se describen los escenarios más interesantes y los casos de uso más importantes del sistema destacando qué componentes de los subsistemas y qué subsistemas tienen mayor relevancia con mayor transcendencia para el lector de esta documentación.

#### 5.3.1. Iniciación de la Visualización

La página web que contiene la visualización debe ser inicializada en el servidor cuando un usuario realiza una consulta sobre la URL correcta.

El primer paso lo realiza el controlador del portlet. Éste adquiere los filtros y establece los elementos del menú a través de llamadas al servicio web. Una vez que obtiene todos los datos, genera la página JSP. Cuando la página ha sido creada se inicializan todos los componentes de JavaScript usando nuevamente llamadas al servicio web. Este proceso se describe con más detalles en la imagen 5.7.

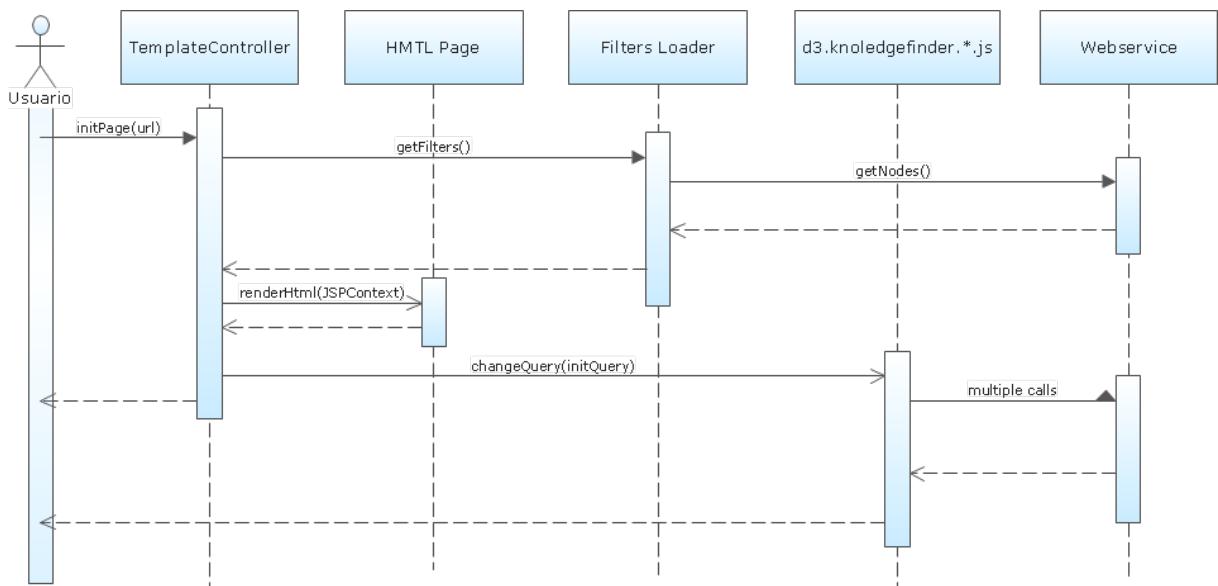


Figura 5.7: Diagrama de Secuencia - Iniciación de la Visualización

#### 5.3.2. Ejecución de Consulta

Cuando el usuario ejecuta una consulta añadiendo o eliminando algún filtro, por ejemplo, seleccionando un elemento del menú o introduciendo un valor en el campo Full-Text el sistema debe obtener el nuevo estado para todos los componentes de la interfaz de usuario.

Concretamente, la lista de resultados debe mostrar el nuevo conjunto de documentos fruto de la nueva consulta. El grafo debe calcular los nuevos valores para sus componentes y redibujarlos en función de éstos. El menú debe adecuarse igualmente al nuevo estado de la aplicación. Para todo ello, se realizan diversas consultas al servicio web de forma asíncrona adaptadas a cada componente. El gráfico representado en la imagen 5.8 esquematiza este comportamiento.

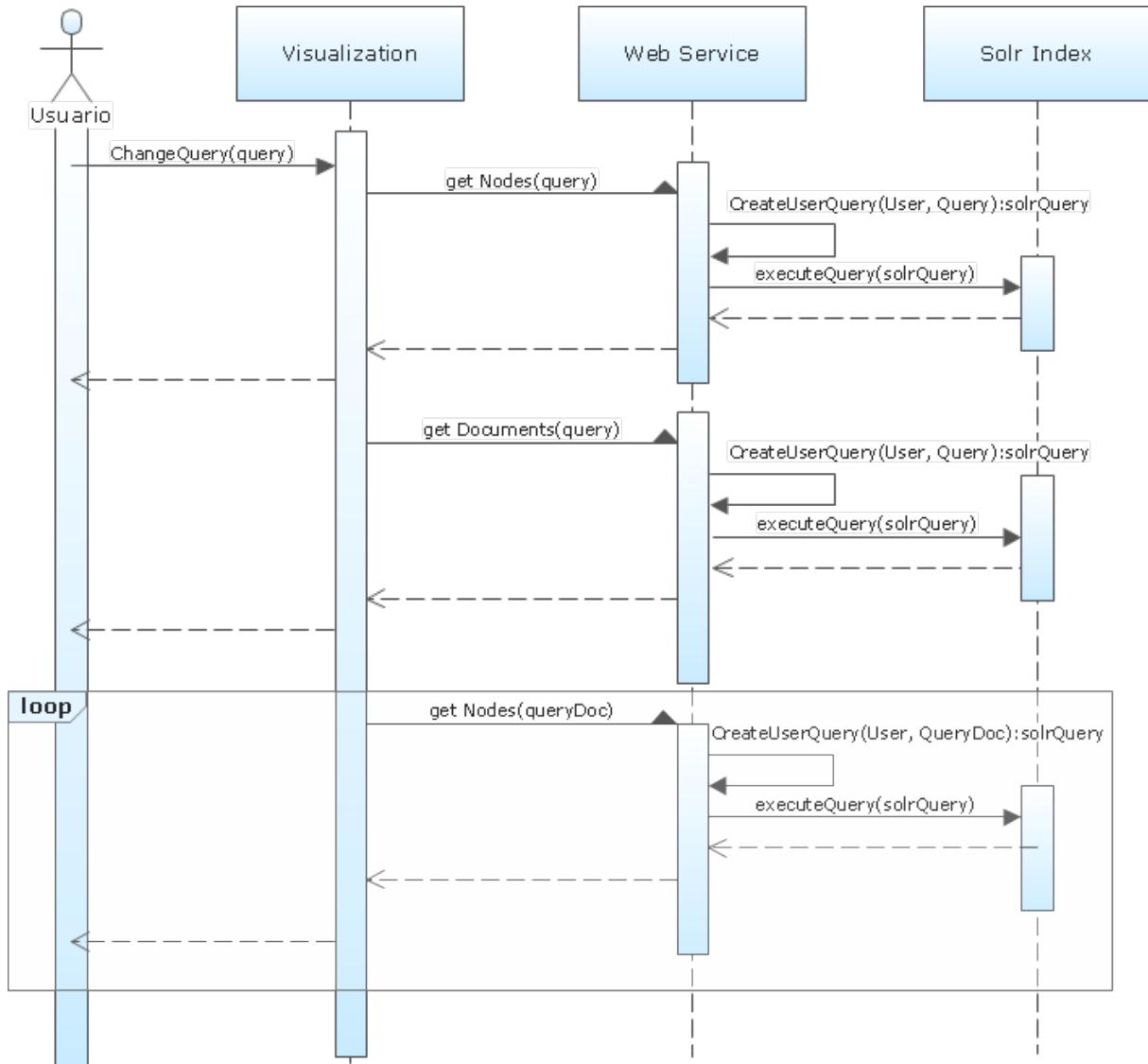


Figura 5.8: Diagrama de Secuencia - Ejecución de Consulta

### 5.3.3. Modificación del Grafo de Exploración

Sin modificar la consulta, el usuario puede modificar los grupos que se muestran en el grafo a través de las opciones del menú. El grafo obtiene los valores para este nuevo estado a través de consultas al servicio web. Con el diagrama de secuencia de la imagen 5.9 se representa este proceso.

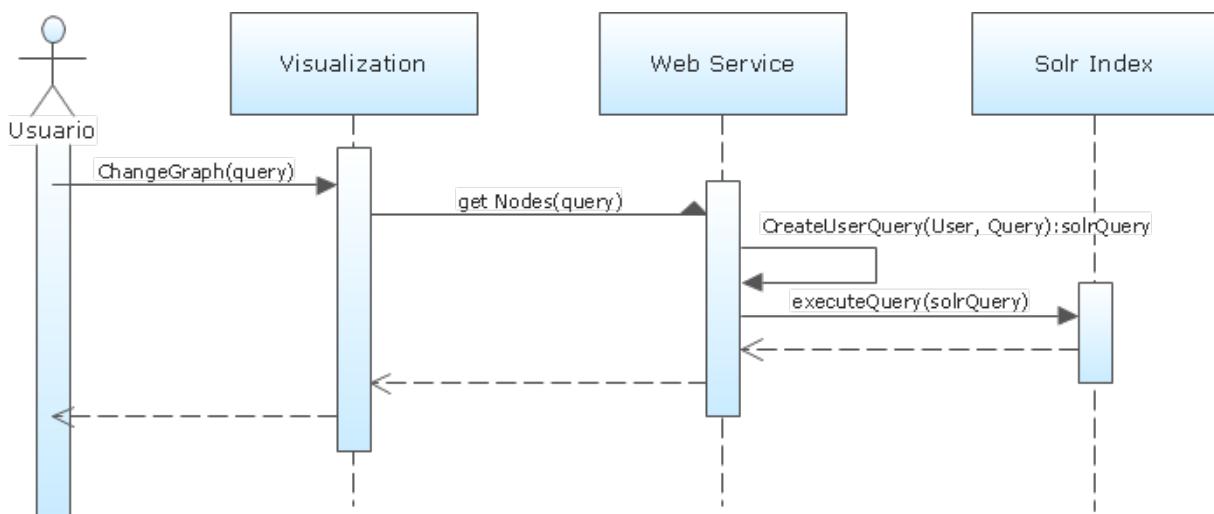


Figura 5.9: Diagrama de Secuencia - Modificación del Grafo de Exploración

### 5.3.4. Modificación de la Lista de Resultados

La lista de resultados está organizada en páginas y ordenada por según la elección del usuario dentro de unas posibles opciones. Tanto si el usuario cambia de página usando los elementos de la paginación o selecciona otro criterio de ordenación el sistema debe actualizar los elementos de esta lista a través de consultas al servicio web. En la imagen 5.10 se expone el diagrama de secuencia para estas acciones.

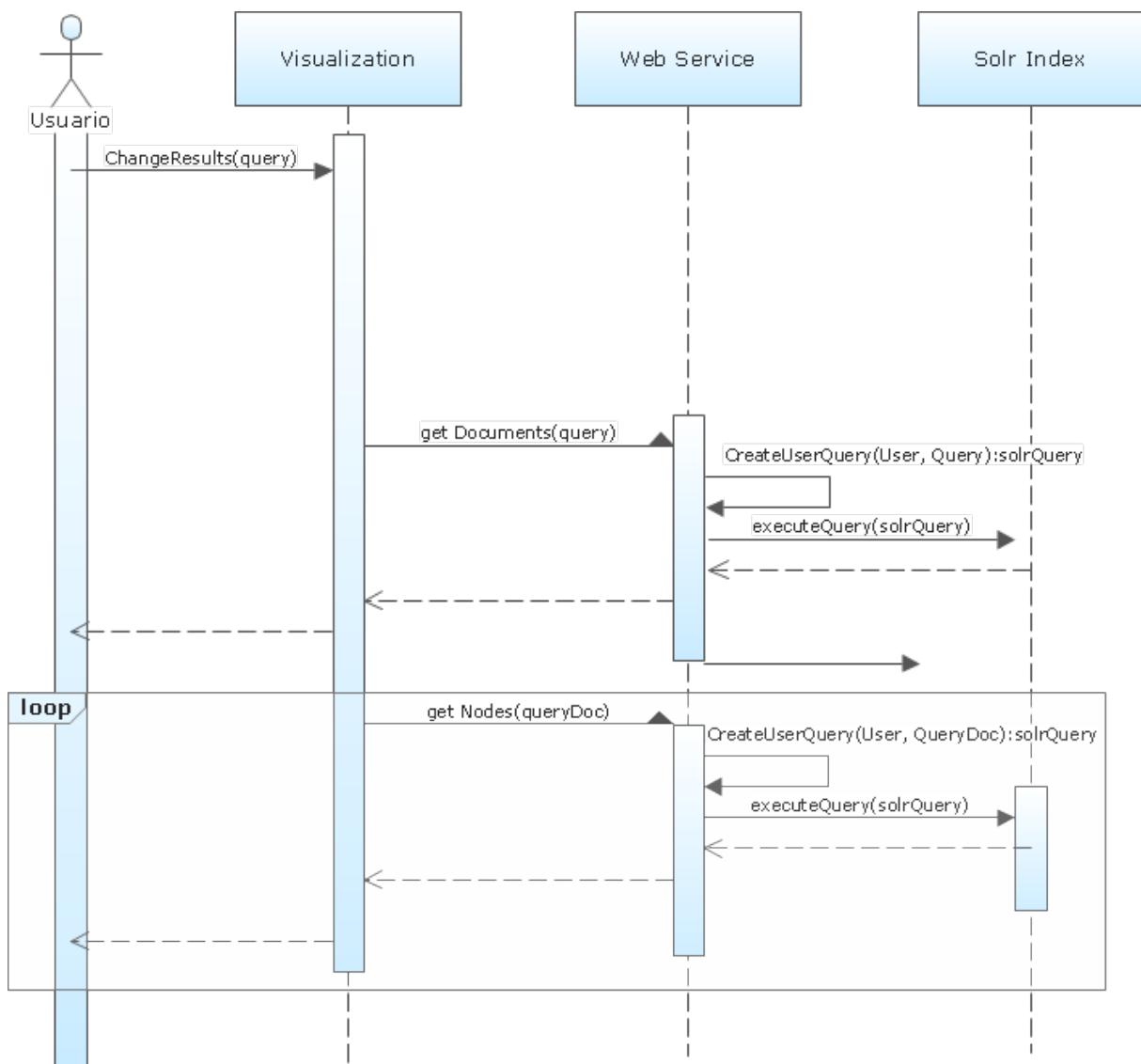


Figura 5.10: Diagrama de Secuencia - Modificación de la Lista de Resultados

## 5.4. Diseño detallado de la Interfaz de Usuario

En esta sección se va a describir con detalle el diseño y el comportamiento de la interfaz de usuario del framework KnowledgeFinder II. Para esta versión se definen dos pantallas; la pantalla principal y la pantalla de detalles de documento.

### 5.4.1. Pantalla Principal

La pantalla principal de la aplicación se puede observar en la imagen 5.11. Como la figura indica, se diferencian cinco componentes que a continuación se exponen.

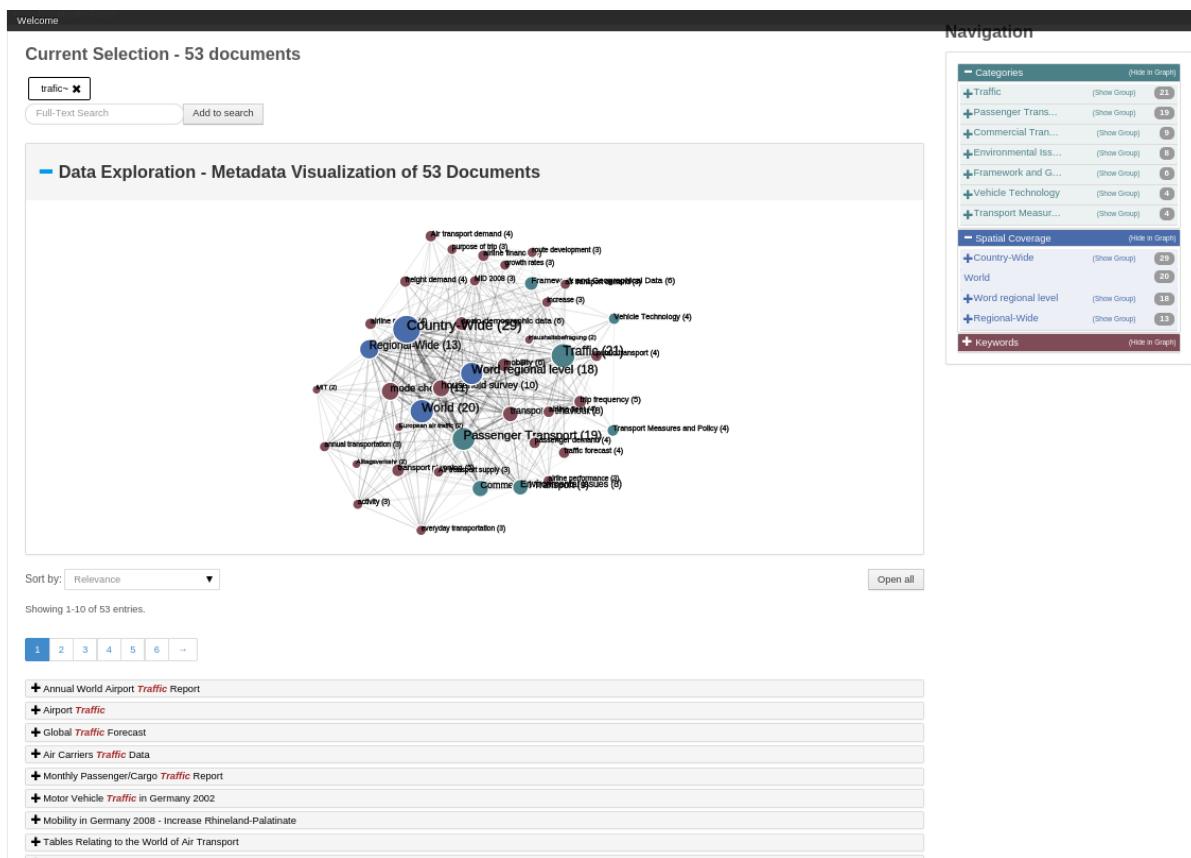


Figura 5.11: Pantalla principal

#### 5.4.1.1. Grapó de Exploración

Representa el elemento más importante de la visualización. En el grafo se representa los filtros aplicables a los documentos que concuerdan con la búsqueda aplicada. Cada filtro representa un nodo y las aristas las relación entre nodos, el número de documentos con ambos filtros. El tamaño de los nodos y el grosor de las aristas está directamente relacionado con el número de documentos para cada elemento.

Cuando el usuario pulsa sobre algún nodo o alguna arista del grafo se aplican los correspondientes filtros.

#### 5.4.1.2. Menú

Ofrece todos los posibles filtros de (Meta-)datos disponibles. Están ordenados por números de documentos afectados por el filtro y se reordenan cuando estos valores cambian.

Se agrupan en, como máximo, dos subniveles siendo el elemento principal de cada grupo. Estos grupos se representan en modo acordeón con un posible scroll sobre las listas de filtros.

A través de un botón en los encabezados de los grupos, el usuario podrá seleccionar qué grupos serán mostrados en el gráfico y pulsando sobre los elementos del menú, qué filtros se han de

aplicar para búsqueda.

#### 5.4.1.3. Lista de Resultados

Los resultado de la búsqueda realizada en cada momento es mostrada en una lista con los títulos de los documentos. Esta lista está dividida usando un sistema de paginación.

Cuando el usuario pulsa sobre algún elemento de la lista, se despliega una sección en forma de acordeón con la descripción del documento seleccionado y un botón para desplegar sus detalles en una ventana emergente (ver sección 5.4.2). También se puede desplegar/plegar este acordeón para todos los elementos de la lista pulsando un botón y puede ser ordenada por varios criterios seleccionables a través de un menú *Drop-down*.

#### 5.4.1.4. Filtros Aplicados

En la parte superior de la pantalla principal se muestran los filtros actuales y cadenas de búsquedas aplicadas. Pulsando sobre ellos se eliminan de los criterios de búsqueda.

#### 5.4.1.5. Comportamientos Transversales

Los componentes anteriormente descritos se relacionan entre sí a través de los siguientes cuando el usuario se posiciona en un elemento del grafo o sobre uno del menú:

- El grafo muestra resaltado respecto al resto de elementos el subgrafo compuesto por las componentes conexas del filtro en cuestión.
- En el menú de elementos se resalta el filtro.
- En la lista de resultados se resaltan los que tienen relación ese filtro.

#### 5.4.1.6. Campo Full-Text

Entre el grafo de exploración y los filtros aplicados se encuentra un simple formulario para la introducción de cadenas de búsqueda. En ella, el usuario puede de definir su propias cadenas de búsqueda.

### 5.4.2. Pantalla de Detalles de Documento

Como se ha explicado antes, los detalles de los documentos se exponen en una ventana emergente. En la imagen 5.12 se observa la simpleza de ésta. Se muestran los campos configurados para cada usuario y la opción de cerrar la ventana para volver a la pantalla anterior.

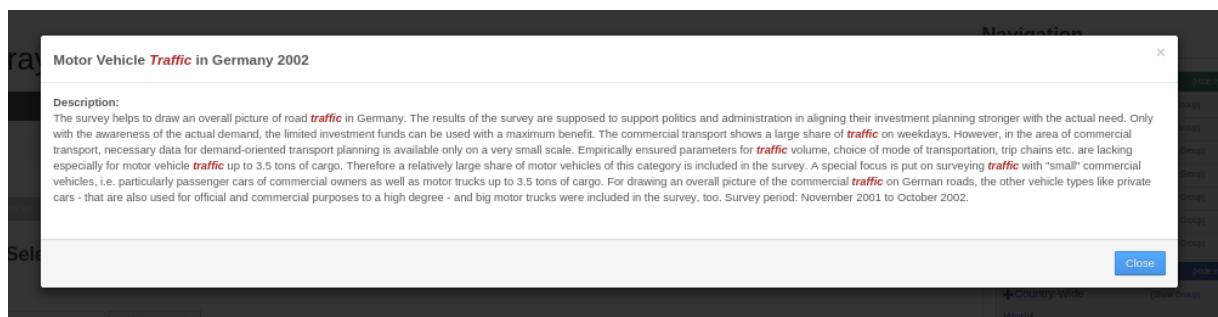


Figura 5.12: Pantalla detalles de documento

# Capítulo 6

## Construcción del Sistema

En este capítulo se detallarán los elementos relacionados con la implementación del framework KnowledgeFinder II y el entorno tecnológico donde se ha desarrollado.

### 6.1. Entorno de Construcción

A continuación vamos a describir las herramientas utilizadas durante el desarrollo resumidas en la sección 2.1.

En entorno de desarrollo empleado fue Eclipse Kepler SR2 utilizando Java 1.7 con Java SDK y JavaScript como lenguajes de programación principales para el desarrollo del producto.

Para controlar las versiones del producto se utilizó un repositorio SVN instalado en el DRL para los proyectos internos el SC-VSS. Este repositorio se utilizó conjuntamente con el gestor de proyectos MantisBT y Jenkins para la integración continua. Ambas herramientas empleadas por el SC-VSS para todos sus proyectos.

En el SC-VSS se dispone de un propio repositorio de Maven para la gestión y construcción de proyectos. Por ello, esta herramienta fue esencial para la construcción y distribución del proyecto desarrollado.

Como se comentó anteriormente, KnowledgeFinder II utiliza Liferay como portal y Solr para el índice de búsqueda. Por este motivo se utilizó una instancia de este portal (versión 6.2.1 CE GA2) funcionando sobre un servidor Tomcat 7 y de Solr 4.8.1 sobre Jetty en su configuración de ejemplo.

Para hacer más cómodo el desarrollo, se empleó el software JRebel 5.6.1 que ayuda a la aplicación automática de la aplicación de las modificaciones en el código desplegado. También se utilizó el framework WireMock 1.18 para simular los servicios durante la fase *Extended Static Prototype* (ver apartado 2.2.0.3).

En la tabla 6.1 se listan los plugins usados conjuntamente con Eclipse para integrar las herramientas anteriormente descritas en el entorno de desarrollo y completar la funcionalidad de este IDE.

Plugin	Versión	Comentario
Subclipse	1.10.5	Plugin para SVN
Mylyn Mantis Connector	3.11.0	Plugin para MantisBT
Hudson/Jenkins Mylyn Builds Connector	1.4.0	Plugin para Jenkins
JRebel for Eclipse	5.6.2	Plugin para JRebel
Liferay IDE	2.1.1 GA2	Plugin para Liferay
m2e-liferay	2.1.1 GA2	Plugin para uso conjunto de Maven y Liferay

Tabla 6.1: Plugins usados en Eclipse durante el desarrollo.

## 6.2. Código Fuente

Como se ha descrito en apartados anteriores, la aplicación está descompuesta en tres subsistemas. A continuación se va a describir el código de cada uno de ellos y los ficheros claves para la configuración para el proyecto usado como ejemplo STRADA@DLR.

### 6.2.1. Índice Solr

La importación de la información contenida en los repositorios SVN esta implementada principalmente en los siguientes paquetes de Java:

- **de.dlr.xps.server.handler.dataimport.crawler** En este paquete se implementa la funcionalidad del subsistema SVN Crawler (ver 5.1.5.2)
- **de.dlr.xps.server.handler.dataimport.datapicker** En este paquete se implementa la funcionalidad del subsistema SVN DataPicker (ver 5.1.5.2)
- **de.dlr.xps.server.handler.dataimport.parser** En este paquete se implementa la funcionalidad del subsistema SVN Parser (ver 5.1.5.2)
- **de.dlr.xps.server.handler.dataimport.transformer** En este paquete se implementa la funcionalidad del subsistema Solr Transformers (ver 5.1.5.2)
- **de.dlr.xps.server.handler.dataimport** En este paquete se implementa la funcionalidad del subsistema SVN DataImport (ver 5.1.5.2)

Los archivos de configuración para el proyecto STRADA@DLR indican cómo debe de realizarse la importación y el esquema del índice de búsqueda. En el fragmento de código 3 es observa el fichero *data-conf.xml* resumido para un origen de datos.

```

1 <dataConfig>
2   <dataSource name="StradaSVN" type="SVNDataSource" />
3   <document>
4     <entity name="strada"
5       processor="SVNEntityProcessor"
6       transformer=""
7         RegexTransformer,
8         TemplateTransformer,
9         de.dlr.xps.server.handler.dataimport.transformer.GenerateIdTransformer,
10        de.dlr.xps.server.handler.dataimport.transformer.LStripTransformer,
11        de.dlr.xps.server.handler.dataimport.transformer.ExcludeValuesTransformer,
12        de.dlr.xps.server.handler.dataimport.transformer.CategoriesIdTransformer,
13        de.dlr.xps.server.handler.dataimport.transformer.CategoriesSeparatedTransformer,
14        de.dlr.xps.server.handler.dataimport.transformer.DateIncompleteFormatTransformer,
15        de.dlr.xps.server.handler.dataimport.transformer.SpatialCoverageTransformer,
16        de.dlr.xps.server.handler.dataimport.transformer.DictionaryListTransformer"
17
18   query="/home/efrain/Escritorio/dlr_software svn/local/CS"
19   <!-- RegexTransformer -->
20   <field column="timePeriodOfDataCollection"
21     regex="(.*)\s-\s(.*)"
22     groupNames="startCollection,endCollection"/>
23   <!-- RegexTransformer -->
24   <field column="temporalCoverage"
25     regex="(.*)\s-\s(.*)"
26     groupNames="startCoverage,endCoverage"/>
27   <!-- DateIncompleteFormatTransformer -->
28   <field column="startTemporalCoverage"
29     sourceColName="startCoverage"
30     parseToDate="start"
31     regex="^(?<year>\d{4})(<month>\d{2})?(<day>\d{2})$"/>
32   <field column="endTemporalCoverage"
33     sourceColName="endCoverage"
34     parseToDate="end"
35     regex="^(?<year>\d{4})(<month>\d{2})?(<day>\d{2})$"/>
36   <!-- ExcludeValuesTransformer, CategoriesIdTransformer, CategoriesSeparatedTransformer -->
37   <field column="contentCategories"
38     name="categories"
39     exclude="import/cs-exclude-categories.txt"
40     categories="import/cs-categories.json"
41     categories_id_name="categories_id"
42     categories_split_prefix="category_"/>
43
44   <!-- LStripTransformer, ExcludeValuesTransformer, SpatialCoverageTransformer -->
45   <field column="spatialCoverage"
46     lstrip="import/strip-spatial.txt"
47     exclude="import/exclude-spatial.txt"
48     spatialColumns="world,regionsWorld,countries,regions"
49     worldFile="import/world.txt"
50     worldRegionsFile="import/worldRegions.txt"/>
51   <!-- DictionaryListTransformer -->
52   <field column="authors"
53     dictionaryId="author" />
54 </entity>
55 ...
56 </document>
57 </dataConfig>
```

Código 3: Ejemplo de importación de datos en KnowledgeFinder II para STRADA@DLR

Para definir el esquema del índice Solr se utiliza el fichero *schema.xml*. En el código 4 se observa como se ha definido éste para STRADA@DLR.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <schema name="knowledgefinder" version="1.5">
3   <fields>
4     <field name="_version_" type="long" indexed="true" stored="true" />
5     <field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
6     <field name="title" type="text_general" indexed="true" stored="true" multiValued="false" required="true" />
7     <field name="description" type="text_general" indexed="false" stored="true" multiValued="false"/>
8     <field name="keywords" type="string" indexed="true" stored="true" multiValued="true" />
9     <field name="categories" type="string" indexed="false" stored="true" multiValued="true" />
10    <field name="filePath" type="string" indexed="false" stored="true" multiValued="false" />
11    <field name="institute" type="string" indexed="true" stored="true" multiValued="false" />
12    <field name="timePeriodOfDataCollection" type="string" indexed="false" stored="true" multiValued="false" />
13    <field name="startTimePeriodOfDataCollection" type="tdate" indexed="true" stored="true" multiValued="false" />
14    <field name="endTimePeriodOfDataCollection" type="tdate" indexed="true" stored="true" multiValued="false" />
15    <field name="startDateDate" type="tdate" indexed="true" stored="true" multiValued="false" />
16    <field name="temporalCoverage" type="string" indexed="false" stored="true" multiValued="false" />
17    <field name="startTemporalCoverage" type="tdate" indexed="true" stored="true" multiValued="false" />
18    <field name="endTemporalCoverage" type="tdate" indexed="true" stored="true" multiValued="false" />
19    <field name="spatialCoverage" type="string" indexed="false" stored="true" multiValued="true" />
20    <field name="world" type="string" indexed="true" stored="false" multiValued="true" />
21    <field name="regionsWorld" type="string" indexed="true" stored="false" multiValued="true" />
22    <field name="countries" type="string" indexed="true" stored="false" multiValued="true" />
23    <field name="regions" type="string" indexed="true" stored="false" multiValued="true" />
24    <field name="size_author" type="int" indexed="false" stored="true" multiValued="false" />
25    <field name="___contentcreationdatetime___" type="int" indexed="true" stored="false" multiValued="false" />
26    <field name="___contentmodificationdatetime___" type="int" indexed="true" stored="false" multiValued="false" />
27    <dynamicField name="author_uid_*" type="string" indexed="false" stored="true" multiValued="false" />
28    <dynamicField name="author(firstName_*)" type="string" indexed="false" stored="true" multiValued="false" />
29    <dynamicField name="author(lastName_*)" type="string" indexed="false" stored="true" multiValued="false" />
30    <dynamicField name="author(mail_*)" type="string" indexed="false" stored="true" multiValued="false" />
31    <dynamicField name="author(organization_*)" type="string" indexed="false" stored="true" multiValued="false" />
32    <field name="full_text_search" type="text_general" indexed="true" stored="false" multiValued="true" />
33    <field name="title_sort" type="alphaOnlySort" indexed="true" stored="true" multiValued="false" />
34
35    <copyField source="title" dest="full_text_search" />
36    <copyField source="title" dest="title_sort" />
37    <copyField source="description" dest="full_text_search" />
38    <copyField source="keywords" dest="full_text_search" />
39    <copyField source="categories" dest="full_text_search" />
40    <copyField source="world" dest="full_text_search" />
41    <copyField source="regionsWorld" dest="full_text_search" />
42    <copyField source="countries" dest="full_text_search" />
43    <copyField source="regions" dest="full_text_search" />
44    <copyField source="author_uid_*" dest="full_text_search" />
45    <copyField source="author(firstName_*)" dest="full_text_search" />
46    <copyField source="author(lastName_*)" dest="full_text_search" />
47    <copyField source="author(mail_*)" dest="full_text_search" />
48    <copyField source="author(organization_*)" dest="full_text_search" />
49    <!-- From Transformer Categories -->
50    <dynamicField name="category_*" type="string" indexed="true" stored="true" multiValued="true" />
51    <copyField source="category_*" dest="full_text_search" />
52    <!-- Not logger required -->
53    <field name="categories_id" type="ignored" indexed="false" stored="false" multiValued="true" />
54    <dynamicField name="*" type="ignored" indexed="true" stored="true" multiValued="true" />
55  </fields>
56  <uniqueKey>id</uniqueKey>
57  ...

```

Código 4: Ejemplo de esquema del índice Solr en KnowledgeFinder II para STRADA@DLR

### 6.2.2. Servicio Web

El servicio web que corre sobre Liferay está organizado en tres paquetes de Java:

- **de.dlr.xps.server.knowledgefinder.webservice.solr**

En este paquete se encuentra el código para obtener la configuración del servicio web.

- **de.dlr.xps.server.knowledgefinder.webservice.solr.query**

Para personalizar y configurar las consultas para cada tipo de rol se han definido en este paquete unas clases en modo de ejemplo para los supuestos roles de usuario.

- **de.dlr.xps.server.knowledgefinder.webservice.service**

En este paquete se encuentran las clases que definen la API del servicio web y su implementación.

Para configurar el servicio web y definir el punto de conexión con el servidor Solr se tiene el fichero *webservice.properties*. En el código 5 se puede observar un ejemplo de configuración para el proyecto STRADA@DLR.

```
1 ##  
2 ## Solr connection  
3 ##  
4     solr.scheme=http  
5     solr.host=localhost  
6     solr.port=8983  
7     solr.username=username  
8     solr.password =password  
9     solr.core=solr/strada
```

Código 5: Ejemplo de configuración del servicio web en KnowledgeFinder II para STRADA@DLR

### 6.2.3. Visualización

El subsistema de visualización se puede dividir en grupos de código con propósitos claramente diferenciados:

- **de.dlr.xps.server.knowledgefinder.frontend.controller**

En este paquete se encuentra el código encargado de inicializar el portlet con la para la página web.

- **de.dlr.xps.server.knowledgefinder.frontend.menu**

A través de un fichero de configuración, genera el el menú para incorporarlo a la visualización.

- **Directorio styles-sass**

En este directorio se encuentra el código en Sass para definir los estilos de la aplicación.

- **Directorio js**

Todo el código JavaScript creado para la visualización se encuentra alojado en este directorio junto con las librerías empleadas.

### 6.2.3.1. Menú

Para configurar el menú se utiliza el fichero en formato JSON *menuConfiguration.json*. La tabla 6.2 muestra el formato de los elementos de configuración que lo componen. En el código 6 se muestra la configuración para STRADA@DLR.

Propiedad	Comentario
id	Valor identificativo del elemento
name	Nombre mostrado en el menú
query	Búsqueda aplicada para el elemento principal
cssClass	Clase de CSS para el grupo
subItemsFacet	Consulta facet para obtener los subelementos
collapsed	Indica si se debe mostrar el grupo plegado
scrollable	Indica se utiliza un scroll para el grupo
subItems	Lista de subelementos de configuración
children_filter	Indica si los subelementos deben ignorarse

Tabla 6.2: Formato de los elementos de configuración para el menú

```

1  {
2      "menuItems": [
3          {
4              "id": "CAT",
5              "name": "Categories",
6              "cssClass": "CAT-filter",
7              "collapsed": false,
8              "scrollable": false,
9              "subItems": [
10                  {
11                      "id": "PT",
12                      "name": "Passenger Transport",
13                      "subItemsFacet": "category_Passenger Transport",
14                      "cssClass": "CAT-PT-filter",
15                      "collapsed": true,
16                      "scrollable": true,
17                      "children_filter": true
18                  },
19                  ...
20              ]
21          },
22          {
23              "id": "SC",
24              "name": "Spatial Coverage",
25              "cssClass": "SC-filter",
26              "collapsed": false,
27              "scrollable": false,
28              "subItems": [
29                  {
30                      "id": "SCW",
31                      "name": "World",
32                      "subItemsFacet": "world",
33                      "cssClass": "SC-SCW-filter",
34                      "collapsed": true,
35                      "scrollable": true,
36                      "children_filter": false
37                  },
38                  {
39                      "id": "SCWR",
40                      "name": "Word regional level",
41                      "subItemsFacet": "regionsWorld",
42                      "cssClass": "SC-SCWR-filter",
43                      "collapsed": true,
44                      "scrollable": true,
45                      "children_filter": true
46                  },
47                  ...
48              ]
49          },
50          {
51              "id": "KW",
52              "name": "Keywords",
53              "cssClass": "red",
54              "query": "keywords:[\\\" TO *]",
55              "cssClass": "KW-filter",
56              "subItemsFacet": "keywords",
57              "collapsed": true,
58              "scrollable": true,
59              "subItems": []
60          }
61      ]
62  }

```

Código 6: Ejemplo de configuración del menú en KnowledgeFinder II para STRADA@DLR

### 6.2.3.2. JavaScript

En la tabla 6.3 se muestra las herramientas externas usadas para este proyecto de JavaScript.

Herramienta	Comentario
D3.js	Herramienta principal de la visualización
bootstrap	Código para componentes visuales
jQuery	Herramienta requerida por otras
Queue.js	Ejecución en paralelo de código JavaScript
URI.js	Herramienta para el manejo de URLs
History.js	Manipulación del historial para HTML5
mCustomScrollbar.js	Scroll en menú

Tabla 6.3: Herramientas de JavaScript usadas

### 6.2.3.3. Estilos CSS

Todos los estilos de la aplicación se han personalizado basándose en la librería bootstrap y se han escrito usando Sass. Para la compilación y aplicación de todos los ficheros fuentes de Sass se ha usado el software Compass.

Para configurar los colores de los elementos gráficos de visualización de la interfaz de usuario se ha definido en un único fichero de Sass, *\_filter\_colors.scss*, la paleta de colores a usar. En los códigos 7 y 8 se muestran cómo es para STRADA@DLR.

```
1  /* Germany colors
2  $CAT-color: lighten(#000000, 50%);  

3  $SC-color: #CD0000;  

4  $KW-color: #FFCC00  

5  */  

6  

7  /* DLR colors */  

8  $CAT-color: rgb(75, 128, 136);  

9  $SC-color: rgb(73, 107, 169);  

10 $KW-color: rgb(126, 75, 86);  

11  

12 $filterColors: (  

13     CAT-filter: $CAT-color,  

14     CAT-PT-filter: $CAT-color,  

15     CAT-T-filter: $CAT-color,  

16     CAT-CT-filter: $CAT-color,  

17     CAT-EI-filter: $CAT-color,  

18     CAT-VT-filter: $CAT-color,  

19     CAT-IN-filter: $CAT-color,  

20     CAT-FDG-filter: $CAT-color,  

21     CAT-TMP-filter: $CAT-color,  

22  

23     SC-filter: $SC-color,  

24     SC-SCW-filter: $SC-color,  

25     SC-SCWR-filter: $SC-color,  

26     SC-SCC-filter: $SC-color,  

27     SC-SCR-filter: $SC-color,  

28  

29     KW-filter: $KW-color,  

30 );
```

Código 7: Ejemplo de paleta de colores en KnowledgeFinder II para STRADA@DLR

```

1  @each $filter, $color in $filterColors {
2    .#${$filter} {
3      /* Graph circles */
4      &.gnode {
5        > circle {
6          fill: $color;
7        }
8      }
9      /* Current selection */
10     &.sel-filter {
11       color: $color;
12       border-color: $color;
13     }
14     /* Navigation table, selected */
15     &.selected {
16       .header-filter {
17         background-color: rgba($color, 0.25);
18       }
19     }
20   }
21 }
22 .CAT-filter {
23   .header-group {
24     color: white;
25     background-color: map-get($filterColors, CAT-filter);
26   }
27   .header-filter {
28     background-color: rgba(map-get($filterColors, CAT-filter), .1);
29   }
30   > a {
31     color: map-get($filterColors, CAT-filter);
32   }
33 }
34 .header-subfilter {
35   .name {
36     color: map-get($filterColors, CAT-filter);
37   }
38 }
39 }
40 /* Example, how to change color of a subfilter */
41 /*
42 .CAT-CT-filter {
43   .header-filter {
44     background-color: rgba(map-get($filterColors, CAT-CT-filter), .1);
45   }
46   > a {
47     color: map-get($filterColors, CAT-CT-filter);
48   }
49   .header-subfilter {
50     .name {
51       color: map-get($filterColors, CAT-CT-filter);
52     }
53   }
54 */
55 */
56 ...

```

Código 8: Ejemplo de aplicación de la paleta de colores en KnowledgeFinder II para STRADA@DLR

# Capítulo 7

## Pruebas del Sistema

En este capítulo se explican el plan de pruebas del proyecto KnowledgeFinder II.

### 7.1. Estrategia

Por las características del producto, las pruebas de cada componente se han planteado de forma distinta:

- **Índice de búsqueda Solr** Se han realizado pruebas a bajo nivel para comprobar el correcto funcionamiento de la lectura, transformación e importación de los datos en el servidor Solr.
- **Servicio web** Este componente ha sido probado durante el proceso de desarrollo de la visualización. Por este motivo, junto con el carácter piloto del software y la complejidad de los elementos implicados, no se ha desarrollado explícitamente ningún tipo de pruebas sobre el código.
- **Visualización** Este componente reúne la mayoría de los requisitos funcionales del usuario final. Por ello, se han realizado en él con una alta perseverancia durante todo el proceso de desarrollo las pruebas de sistemas.

### 7.2. Entorno de Pruebas

El conjunto de pruebas se ha realizado a tres niveles:

- **Equipo de desarrollo** Las pruebas se han ejecutado en el equipo del programador durante el desarrollo.
- **Servidor de pruebas** Para ver el producto en funcionamiento en un equipo semejor a donde se pondrá en producción se ha dispuesto de un servidor de pruebas remoto.
- **Jenkins** Esta herramienta ha realizado y registrado el resultado de la fase de testing durante el despliegue de las distintas versiones intermedias del producto.

### 7.3. Roles

En desarrollador se encarga de realizar las pruebas durante todo el desarrollo del producto. Todas las pruebas unitarias y de integración han sido ejecutadas únicamente por el desarrollador y las herramientas automáticas.

El desarrollador también ha comprobado cumplimiento de los requisitos realizando las pruebas de sistema durante el transcurso la creación del producto.

Conjuntamente con el personal de SC-VSS y de los institutos VF y FW asignados al proyecto, el desarrollador se ha encargado de las pruebas de aceptación.

## 7.4. Niveles de Pruebas

### 7.4.1. Pruebas Unitarias

El proceso de lectura de la información contenida en SVN y las transformaciones previas a la importación ha sido probado a través de pruebas unitarias para cada componente funcional.

### 7.4.2. Pruebas de Integración

Gracias al paquete para pruebas implementado en Solr, los elementos encargados de la importación han sido probados en un servidor Solr destinado a tal fin. De esta forma se aseguró que el código encargado de la importación funciona correctamente sobre la versión de Solr deseada.

### 7.4.3. Pruebas de Sistema

En lo que refiere al índice de búsqueda, se ha comprobado que el índice resultado de la importación contiene la información esperada. Para ello se ha comparado la información contenida en el repositorio SVN en formato JSON con los datos del índice.

Por otra parte, para comprobar el correcto funcionamiento de la aplicación y que el sistema cumple con todos los requisitos establecidos se han realizado pruebas manuales a lo largo de la fase de implementación de la visualización y del servicio web.

Con una lista que representa cada caso de uso establecido, se ha comprobado en cada versión de los componentes de visualización y servicio web que los requisitos funcionales concluidos y los requisitos no funcionales han seguido estando satisfechos.

### 7.4.4. Pruebas de Aceptación

Usando el servidor de pruebas anteriormente nombrado, se han realizado las pruebas de aceptación durante la fase cercana a la finalización del esta versión del framework KnowledgeFinder II. Principalmente han consistido en la comprobación de las pruebas de sistema en el servidor.

---

## **Parte III**

### **Epílogo**



## Capítulo 8

# Manual de implantación y explotación

Las instrucciones de uso del framework KnowledgeFinder II aplicado al STRADA@DLR se detallan a continuación.

### 8.1. Introducción

En el presente capítulo se desarrollará el manual de instalación y explotación del índice de búsqueda en Solr, del servicio web desplegado en Liferay y de la interfaz de usuario en un portlet del proyecto STRADA@DLR implementado con el framework KnowledgeFinder II.

Esta aplicación web permite consultar las bases de datos de “Transporte y la Movilidad” de los institutos del Centro Aeroespacial Alemán (DLR) Instituto para la Investigación del Transporte Aéreo y el Sistema Aeroportuario (FW) y Instituto para la Investigación del Transporte (VF) conjuntamente los documentos de ambos portales ofreciendo un sistema de búsqueda avanzada y optimizada sobre los (Meta-)datos de los documentos de los institutos involucrados.

El portal ofrece una visualización del conocimiento interactiva de los (Meta-)datos que representa las relaciones espaciales, temporales y contextuales entre los documentos y fuentes de datos. Esta representación aporta nuevos conocimientos que facilitan las tareas de investigación.

Este manual se divide en tres módulos que se tratarán de exponer de forma independiente:

- Índice Solr
- Servicio web en Liferay
- Portlet de visualización

### 8.2. Requisitos generales previos

A continuación se exponen los requisitos necesarios comunes a todos los módulos que se describirán en este manual.

#### 8.2.1. Java

Basándonos en la documentación de Solr 4.8.1 [Wika], el servidor requiere la versión de Java SDK 7 o superior. Aunque impuesto por Solr, se recomienda usar la misma versión de Java SDK

para los restantes módulos.

Una vez se tenga instalado, hay que configurar la variable del sistema *JAVA\_HOME* para usar el Java SDK que acabamos de instalar.

### 8.2.2. Eclipse

Para el desarrollo de este software se recomienda utilizar el IDE Eclipse. Éste se puede descargar en la dirección oficial <http://www.eclipse.org/downloads/>.

### 8.2.3. Maven 3

La instalación de Maven se puede realizar en algunos sistemas operativos, como Ubuntu, desde los repositorios oficiales. Desde la dirección oficial (<http://maven.apache.org/download.html>) también se pueden descargar e instalar fácilmente.

Si no se ha configurado automáticamente durante el paso anterior, hay que crear una variable de sistema *M2\_HOME* que apunte al directorio donde se ha instalado, por ejemplo: C:\Program Files(x86)\apache-maven-3.1.0.

Si la organización cuenta con un repositorio de Maven propio, éste se puede añadir editando el fichero *settings.xml* de Maven. Por ejemplo, para el SC-VSS, el repositorio apunta a <http://repo.sistec.dlr.de/nexus/content/groups/public/>.

Para usarlo conjuntamente con Eclipse (apartado 8.2.2) se recomienda instalar el plugin Maven2Eclipse desde la dirección <http://download.eclipse.org/technology/m2e/releases>

### 8.2.4. Cliente SVN

Para poder obtener el código y trabajar con el repositorio SVN es necesario instalar un cliente de SVN 1.7 o superior.

También se puede optar por instalar directamente un plugin para Eclipse que incluya este cliente, por ejemplo Subclipse que está disponible bajo la dirección [http://subclipse.tigris.org/update\\_1.10.x](http://subclipse.tigris.org/update_1.10.x).

### 8.2.5. MantisBT y Jenkins

Eclipse se conecta con MantisBT y Jenkins a través de Mylyn. Este plugin está disponible en la dirección <http://download.eclipse.org/mylyn/releases/latest>.

Para configurar en Mylyn MantisBT en Eclipse siga los siguientes pasos:

1. Window >Show View >Task List
  2. Click on Add MyLyn Support
  3. Install connector (Mantis)
  4. Boton derecho en task view, add query
-

5. Add new repository

Type: Mantis

Path: <https://www.sistec.dlr.de/mantis>

User/Pass: Tus credenciales

Validate

6. Crear las consultas (queries) deseadas.

A continuación, la configuración de Mylyn con Jenkins:

1. Window >Show View >Builds

2. Add build

Type: Hudson [Jenkins]

Path: <https://ci.sc.dlr.de/jenkins>

User/Pass: Tus credentials

Validate

3. Elegir los builds de KnowledgeFinder II de la lista.

## 8.3. Índice Solr

En este apartado se describe el manual de desarrollo, instalación y explotación del sistema de importación de datos y su aplicación en un servidor Solr.

### 8.3.1. Requisitos previos

#### 8.3.1.1. Apache Solr 4.8.1

Desde la dirección <http://www.apache.org/dyn/closer.cgi/lucene/solr/4.8.1> nos podemos descargar Solr. Una vez descargado y descomprimido en un directorio, por ejemplo “UnzipDirectory”, ejecutamos el servidor Jetty de ejemplo (código 9).

```
1 java -jar [UnzipDirectory]/solr-4.8.1/example/start.jar
```

Código 9: Ejemplo de ejecución inicial de Solr en Jetty

Si todo ha funcionado correctamente, Solr estaría accesible en la dirección <http://localhost:8983/solr>.

### 8.3.2. Procedimientos de instalación

#### 8.3.2.1. Obtención del software

Checkout desde SVN el índice desde <https://svn.sistec.dlr.de/svn/xps/xps-server/branches/visual/solrIndex/> si no se dispone de una copia de trabajo.

Checkout desde SVN del sistema de importación desde <https://svn.sistec.dlr.de/svn/xps/xps-server/branches/visual/de.dlr.xps.server.knowledgefinder.handler.dataimport/> si no se dispone de una copia de trabajo.

#### 8.3.2.2. Construcción del software

Situándonos en el directorio donde se ha realizado el checkout, ejecutamos las líneas expuestas en el código 10 y 11.

```
1 cd ./de.dlr.xps.server.knowledgefinder.handler.dataimport/  
2 mvn clean install
```

Código 10: Construcción con Maven de DataImport

```
1 cd ../solrIndex/  
2 mvn clean install
```

Código 11: Construcción con Maven de índice de búsqueda

A través de Eclipse se puede realizar también siguiendo estos pasos:

- Añadir un nuevo “Maven Run Configuration” usando como directorio base `#{workspace_loc : de.dlr.xps.server.knowledgefinder.handler.dataimport}` y como “goal” clean install para construir el sistema de importación.
- Añadir un nuevo “Maven Run Configuration” usando como directorio base `#{workspace_loc : solr_index}` y como “goal” clean install para construir el índice.

#### 8.3.2.3. Despliegue en el Solr

Para concluir se debe de configurar Solr para que utilice el índice. Para ello se debe ejecutar el servidor Jetty como se vio en el apartado 8.3.1.1 pero indicando la variable `solr.solr.home` (código 12).

```
1 java -jar -Dsolr.solr.home={workspace_loc}/solr_index  
2 [UnzipDirectory]/solr-4.8.1/example/start.jar
```

Código 12: Ejemplo de ejecución de Solr en Jetty con el índice de STRADA@DLR

#### 8.3.2.4. Configuración del índice

Después de la instalación, hay que configurar el origen de los datos.

Los datos se pueden obtener desde el repositorio SVN (<https://svn-dmz.sistec.dlr.de/svn/xps/fw-test-data/>) y configurar el índice con la ruta de la copia de trabajo local (código 13).

```
1 ...
2 <entity name="strada"
3 processor="SVNEntityProcessor"
4 transformer="..."
5 query="/KFdata/dat/CS"
6 ...
7 >
8 ...
9 <entity name="fw"
10 processor="SVNEntityProcessor"
11 transformer="..."
12 query="/KFdata/dat/FW"
13 ...
14 >
15 ...
```

Código 13: Configuración del índice desde copia de trabajo local de SVN

Otra opción es que la importación se realice directamente leyendo el repositorio de SVN remoto. Para ello hay que configurar también las credenciales para acceder al mismo (código 14)

```
1 ...
2 <entity name="strada"
3 processor="SVNEntityProcessor"
4 transformer="..."
5 query="https://svn-dmz.sistec.dlr.de/svn/xps/fw-test-data/dat/CS"
6 username="usernameSVN"
7 password="passwordSVN"
8 ...
9 >
10 ...
11 <entity name="fw"
12 processor="SVNEntityProcessor"
13 transformer="..."
14 query="https://svn-dmz.sistec.dlr.de/svn/xps/fw-test-data/dat/FW"
15 username="usernameSVN"
16 password="passwordSVN"
17 ...
18 >
19 ...
```

Código 14: Configuración del índice desde repositorio remoto de SVN

### Configuración en servidor

En el código 15 se muestra un ejemplo de configuración para servicio en una máquina de la familia Debian.

```
1 #!/bin/bash
2 #startup script for the Solr server
3 #
4 # description: Starts and stops the Solr with Jetty daemon.
5 # processname: solr (java)
6 # pidfile: /var/run/solr.pid
7 JAVA_7_HOME=/usr/lib/jvm/jdk1.7.0_60/
8 SOLR_HOME=/home/lima_ef/solrIndex/
9 JETTY_HOME=/opt/solr-4.8.1/example/
10 STOP_PORT=8079
11 STOP_KEY=keyStopSolr
12 case $1
13 in
14 start)
15 export JAVA_HOME=$JAVA_7_HOME
16 java -jar -Dsolr.solr.home=$SOLR_HOME -DSTOP.PORT=$STOP_PORT
17 -Djetty.home=$JETTY_HOME
18 /opt/solr-4.8.1/example/start.jar --start
19 ;;
20 stop)
21 export JAVA_HOME=$JAVA_7_HOME
22 java -jar -Dsolr.solr.home=$SOLR_HOME -DSTOP.PORT=$STOP_PORT
23 -Djetty.home=$JETTY_HOME
24 /opt/solr-4.8.1/example/start.jar --stop
25 ;;
26 restart)
27 export JAVA_HOME=$JAVA_7_HOME
28 java -jar -Dsolr.solr.home=$SOLR_HOME -DSTOP.PORT=$STOP_PORT
29 -Djetty.home=$JETTY_HOME
30 /opt/solr-4.8.1/example/start.jar --stop
31 java -jar -Dsolr.solr.home=$SOLR_HOME -DSTOP.PORT=$STOP_PORT
32 -Djetty.home=$JETTY_HOME
33 /opt/solr-4.8.1/example/start.jar --start
34 ;;
35 *)
36 echo "Usage: /etc/init.d/solr start|stop|restart"
37 ;;
38 esac
39 exit 0
```

Código 15: Ejemplo de servicio para Solr en una máquina de Debian (etcinit.desolr)

### 8.3.3. Pruebas de implantación

Para comprobar que todo el proceso se ha realizado correctamente, la importación debe ser exitosa. Para ello:

1. Comprobar que en la dirección `http://localhost:8983/solr/#/strada` se encuentran las propiedades generales del índice.
2. Recargar el índice:
  - Ir a *Core Admin*.
  - Seleccionar el índice para STRADA@DLR.
  - Pulsar *Reload*.
3. Ir a la dirección `http://localhost:8983/solr/#/strada/dataimport//dataimport`
4. Pulsar *Execute*
5. Para ver si los datos se han importado correctamente, pulsar *Refresh Status* y comprobar el número de documentos importados.
6. Mediante una simple consulta, se puede ver cómo han sido importados los datos. Por ejemplo, visitando `http://localhost:8983/solr/strada/select?q=*&wt=json&indent=true` se obtienen todos los datos importados en formato JSON.

#### 8.3.4. Procedimientos de operación y nivel de servicio

Solr dispone de su propio sistema de logging. Durante toda la ejecución de Solr se puede consultar a través de la interfaz de administración o en la dirección `http://localhost:8983/solr/admin/info/logging` (formato XML).

### 8.4. Servicio web en Liferay

En esta sección se describe el manual de desarrollo, instalación y explotación del servicio web que se aloja en Liferay para la utilización del índice Solr anteriormente explicado.

#### 8.4.1. Requisitos previos

##### 8.4.1.1. Índice Solr

Como es de esperar, este servicio web necesita una instancia del índice Solr para el proyecto STRADA@DLR en funcionamiento. En el punto anterior 8.3 se explica cómo realizarlo.

##### 8.4.1.2. Liferay v6.2 CE Server (Tomcat 7)

Desde la dirección `http://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.1%20GA2/liferay-portal-tomcat-6.2-ce-ga2-20140319114139101.zip/download` nos podemos descargar Liferay. Una vez descargado, lo descomprimimos en un directorio, por ejemplo “UnzipDirectory”.

Para ejecutar el servidor Tomcat con Liferay de ejemplo (código 16).

---

```
1 java -jar [UnzipDirectory]/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/
2           bin/catalina.sh start
```

Código 16: Ejemplo de ejecución inicial de Liferay en Tomcat

Si todo ha funcionado correctamente, Liferay estará accesible en la dirección <http://localhost:8080/> donde se mostrará el formulario para la configuración del portal.

Para trabajar más cómodamente en Eclipse con Maven y Liferay, se recomienda instalar los plugins Liferay IDE y m2e-liferay disponibles en <http://releases.liferay.com/tools/ide/latest/milestone/>.

#### 8.4.2. Procedimientos de instalación

##### 8.4.2.1. Obtención del software

Realizamos un checkout del código del servicio web desde SVN <https://svn.sistec.dlr.de/svn/xps/xps-server/branches/visual/webservice/> si no se dispone de una copia de trabajo.

##### 8.4.2.2. Construcción del software

Situándonos en el directorio donde se ha realizado el checkout, ejecutamos las líneas expuestas en el código 17.

```
1 cd ./webservice/
2 mvn clean install
```

Código 17: Construcción con Maven del servicio web

A través de Eclipse se puede realizar también añadiendo un nuevo “Maven Run Configuration” usando como directorio base

`${workspace_loc : webservice}`

y como “goal” clean install para construir el servicio web.

##### 8.4.2.3. Despliegue en Liferay

Gracias al plugin anteriormente indicado, en el archivo de configuración de Maven del servicio web se puede configurar para que Eclipse realice los despliegues en el portal Liferay de forma automática. Para ello hay que editar el fichero *pom.xml* como se indica en el código 18.

```
1 <properties>
2 ...
3 <liferay.version>6.2.1</liferay.version>
4 <liferay.auto.deploy.dir>
5     [UnzipDirectory]/liferay-portal-6.2-ce-ga2/deploy/
6 </liferay.auto.deploy.dir>
7 <liferay.maven.plugin.version>
8     6.2.1
9 </liferay.maven.plugin.version>
10 ...
11 </properties>
12 ...
```

Código 18: Extracto del fichero de configuración de Maven para auto-despliegue con Liferay

y luego ejecutar el plugin desde la consola (código 19) o configurando Eclipse añadiendo un nuevo “Maven Run Configuration” usando como directorio base

```
 ${workspace_loc : webservice}
```

y como “goal” liferay:deploy.

```
1 cd ./webservice/
2 mvn liferay:deploy
```

Código 19: Auto-deploying con Maven del servicio web

#### 8.4.2.4. Configuración del servicio web

También se debe de configurar el servicio web para que utilice el índice Solr. Para ello se debe adaptar el fichero *webservice.properties* (código 20).

```
1 ##
2 ## Solr connection
3 ##
4 solr.scheme=http
5 solr.host=localhost
6 solr.port=8983
7 # solr.username=username
8 # solr.password =password
9 solr.core=solr/strada
```

Código 20: Ejemplo de configuración del servicio web para usar el índice de Solr

## Configuración en servidor

En el código 21 se muestra un ejemplo de configuración para crear un servicio en una máquina de la familia Debian.

```
1 #!/bin/bash
2 #
3 #
4 #
5 #
6 #
7 Startup script for the Tomcat server
8 description: Starts and stops the Tomcat daemon.
9 processname: liferay (java)
10 pidfile: /var/run/liferay.pid
11 # See how we were called.
12 JAVA_7_HOME="/usr/lib/jvm/jdk1.7.0_60/"
13 case $1
14 in
15 start)
16 export JAVA_HOME=$JAVA_7_HOME
17 su - liferay -c 'sh /opt/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/bin/startup.sh'
18 ;;
19 stop)
20 export JAVA_HOME=$JAVA_7_HOME
21 su - liferay -c 'sh /opt/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/bin/shutdown.sh'
22 ;;
23 restart)
24 export JAVA_HOME=$JAVA_7_HOME
25 su - liferay -c 'sh /opt/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/bin/shutdown.sh'
26 su - liferay -c 'sh /opt/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/bin/startup.sh'
27 ;;
28 *)
29 echo "Usage: /etc/init.d/liferay start|stop|restart"
30 ;;
31 esac
32 exit 0
```

Código 21: Ejemplo de servicio para Liferay en una máquina de Debian

### 8.4.3. Pruebas de implantación

Para comprobar que todo el proceso se ha realizado correctamente, simplemente hay que ejecutar algunas de las funciones del servidor. Utilizando la interfaz para servicios web de Liferay se pueden usar los formularios que automáticamente éste genera y ver que las peticiones realizan su cometido:

1. Abrir la dirección <http://localhost:8080/api/jsonws>.
2. Seleccionar la API de KnowledgeFinder II.
3. Utilizar el formulario para ejecutar alguna función de la API.
4. Comprobar el resultado abriendo la URL generada.

#### 8.4.4. Procedimientos de operación y nivel de servicio

Cada petición de Liferay queda registrado en un sistema de logging propio de Tomcat. Durante toda la ejecución del servicio web se puede consultar a través de los ficheros de la carpeta [UnzipDirectory]/liferay-portal-6.2-ce-ga2/tomcat-7.0.42/logs/.

### 8.5. Visualización

En esta sección se describe el manual de desarrollo, instalación y explotación del portlet alojado en Liferay encargado de la visualización.

#### 8.5.1. Requisitos previos

##### 8.5.1.1. Servicio web accesible

La visualización requiere del servicio web explicado en la sección anterior (8.4. Necesita una instancia del índice servicio web para el proyecto STRADA@DLR en funcionamiento).

##### 8.5.1.2. Liferay v6.2 CE Server (Tomcat 7)

Ver apartado 8.4.1.2

#### 8.5.2. Procedimientos de instalación

##### 8.5.2.1. Obtención del software

Obtener el código del componente de visualización desde SVN (<https://svn.sistec.dlr.de/svn/xps/xps-server/branches/visual/de.dlr.xps.server.knowledgefinder.frontend/>) si no se dispone de una copia de trabajo.

##### 8.5.2.2. Construcción del software

Situándonos en el directorio donde se ha realizado el checkout, ejecutamos las líneas expuestas en el código 22.

```
1 cd ./de.dlr.xps.server.knowledgefinder.frontend/  
2 mvn clean install
```

Código 22: Construcción con Maven de la visualización

A través de Eclipse se puede realizar también añadiendo un nuevo “Maven Run Configuration” usando como directorio base

---

`${workspace_loc : de.dlr.xps.server.knowledgefinder.frontend}`  
y como “goal” clean install para construir el portlet para la visualización.

#### 8.5.2.3. Despliegue en Liferay

Gracias al plugin anteriormente indicado, en el archivo de configuración de Maven de la visualización se puede configurar para que Eclipse realice los despliegues en el portal Liferay de forma automática. Para ello hay que editar el fichero *pom.xml* como se indica en el código 23.

```
1 <properties>
2 ...
3 <liferay.version>6.2.1</liferay.version>
4 <liferay.auto.deploy.dir>
5     [UnzipDirectory]/liferay-portal-6.2-ce-ga2/deploy/
6 </liferay.auto.deploy.dir>
7 <liferay.maven.plugin.version>6.2.1</liferay.maven.plugin.version>
8 ...
9 </properties>
10 ...
```

Código 23: Extracto del fichero de configuración de Maven para auto-deploying con Liferay

y luego ejecutar el plugin desde la consola (código 24) o configurando Eclipse añadiendo un nuevo “Maven Run Configuration” usando como directorio base  
 `${workspace_loc : de.dlr.xps.server.knowledgefinder.frontend}`  
y como “goal” liferay:deploy.

```
1 cd ./de.dlr.xps.server.knowledgefinder.frontend/
2 mvn liferay:deploy
```

Código 24: Auto-deploying con Maven del servicio web

Una vez que el portlet ha sido desplegado, hay que añadirlo a una página de Liferay siguiendo los estos pasos:

1. Acceder a Liferay.
2. Registrarse en el sistema.
3. Seleccionar la página donde se quiere añadir.
4. Pulsar en el botón *Add*. Este aparece en la esquina superior izquierda del portal.
5. Seleccionar *Applications* y luego el portlet de KnowledgeFinder II.
6. Arrastrar el elemento de la lista sobre la posición de la página.

#### 8.5.2.4. Configuración de la visualización

También se debe de configurar el portlet encargado de la visualización para que acceda a las funciones del servicio web. Para ello se debe adaptar el fichero *service.properties* (código 25).

```
1 host=http://localhost:8080
2 url=/api/jsonws/KnowledgeFinderWebservice.knowledgefinder/
3 urlDocuments=get-documents/
4 urlNodes=get-nodes/
```

Código 25: Configuración para la comunicación portlet de visualización con el servicio web

#### Configuración en servidor

En el código 21 se muestra un ejemplo de configuración para crear un servicio en una máquina de la familia Debian.

#### 8.5.3. Pruebas de implantación

Para comprobar que todo el proceso se ha realizado correctamente, simplemente hay que ejecutar algunas de las funciones de la visualización. Si no aparece ningún elemento gráfico en la interfaz de usuario puede significar que ésta no se puede comunicar con el servicio web.

#### 8.5.4. Procedimientos de operación y nivel de servicio

Cada petición de Liferay queda registrado en un sistema de logging propio de Tomcat. Durante toda la ejecución de la visualización se puede consultar a través de los ficheros de la carpeta [UnzipDirectory]liferay-portal-6.2-ce-ga2tomcat-7.0.42logs.

A nivel de interfaz de usuario, algunos navegadores como Chrome o Chromium incorporan una consola donde se muestran los errores en tiempo de ejecución y el logging de JavaScript.

# Capítulo 9

## Manual de usuario

### 9.1. Introducción

En el presente capítulo se desarrollará el manual de usuario, el cual permite obtener la información necesaria para los usuarios finales para poder utilizar el portal STRADA@DLR en la versión implementada usando el framework KnowledgeFinder II.

Esta aplicación web permite consultar las bases de datos de “Transporte y la Movilidad” de los institutos del Centro Aeroespacial Alemán (DLR) Instituto para la Investigación del Transporte Aéreo y el Sistema Aeroportuario (FW) y Instituto para la Investigación del Transporte (VF) conjuntamente los documentos de ambos portales ofreciendo un sistema de búsqueda avanzada y optimizada sobre los (Meta-)datos de los documentos de los institutos involucrados.

El portal ofrece una visualización del conocimiento interactiva de los (Meta-)datos que representa las relaciones espaciales, temporales y contextuales entre los documentos y fuentes de datos. Esta representación aporta nuevos conocimientos que facilitan las tareas de investigación.

Una vez concluida la instalación como se ha explicado en el capítulo , la aplicación se encuentra preparada para ser usada siguiendo este manual.

### 9.2. Características

Las principales funcionalidades de esta versión del STRADA@DLR son las que se describen a continuación:

- **Generación de grafo dinámico de exploración del conocimiento**
- **Complejidad personalizable del grafo de exploración**
- **Búsqueda Full-Text.**
- **Búsqueda a través del grafo de exploración.**
- **Búsqueda a través del menú de opciones.**
- **Lista editable con los valores actuales de búsqueda.**

- Pantalla de detalles por documento.
- URL con el estado de la aplicación compatible.
- Interacción visual de los elemento de la interfaz de usuario.

### 9.3. Requisitos previos

Como se ha comentado anteriormente, es necesario la instalación previa de la aplicación. Los requisitos de ésta y los pasos a seguir se encuentran explicados en el capítulo anterior.

En lo que refiere al usuario del nuevo portal STRADA@DLR, para poder acceder a él con una completa funcionalidad es necesario disponer de un navegador web actual con JavaScript activo y que acepte HTML5. Se recomienda el uso de Chrome o Chromium en sus últimas versiones aunque se pueden utilizar otros navegadores como Internet Explorer 9+, Mozilla Firefox 32+ o Safari 5+.

### 9.4. Uso del sistema

A continuación se describe cómo usar la aplicación y las posibilidades que ofrece. En la imagen 9.1 se observa el emplazamiento de los componentes principales del portal STRADA@DLR; grafo de exploración, menú de navegación, Full-Text, lista de resultados y selección actual. Los estilos de estos componentes están configurados para STRADA@DLR siguiendo un código de colores que facilita la relación visual entre ellos.

Desglosado por estos componentes, se describe las acciones posibles a realizar con cada uno de ellos y las interacciones con el resto de elementos.

#### 9.4.1. Grafo de Exploración

El grafo de exploración es el elemento principal de la visualización. Este está compuesto por nodos y por aristas que los unen. En la imagen 9.2 vemos un grafo de exploración con varios (Meta-)datos interrelacionados.

##### 9.4.1.1. Nodos

Los nodos representan los filtros o (Meta-)datos de los documentos y su tamaño varía dependiendo del número de documentos que los contienen para el actual conjunto de documentos.

Cuando se pulsa sobre uno de ellos, éste es usado para filtrar los documentos con este (Meta-)dato.

Por otra parte, mientras el usuario sitúa el cursor sobre uno de ellos (sin pulsar) todos los nodos del grafo no pertenecientes a su componente conexa (sin relación directa) se atenúan junto con sus relaciones y aparece un cuadro de texto con propiedades del (Meta-)dato. También se resaltan en la lista de resultados los títulos de los documentos con este filtro y en la menú de navegación el filtro.seleccionado. Por ejemplo, en la imagen 9.2 podemos ver con está resaltado en nodo *Traffic* y los nodos con sus relaciones que están relacionados directamente con él.

---

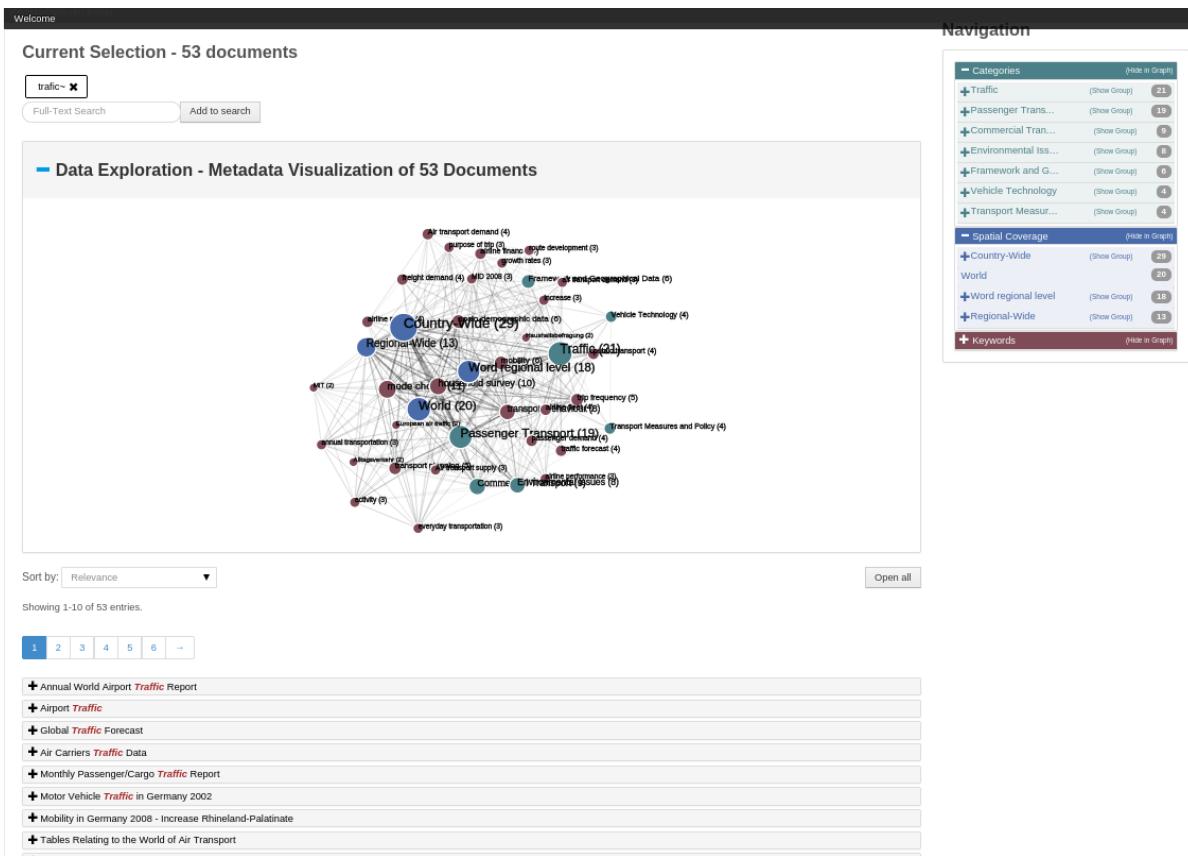


Figura 9.1: Pantalla principal del STRADA@DLR usando KnowledgeFinder II

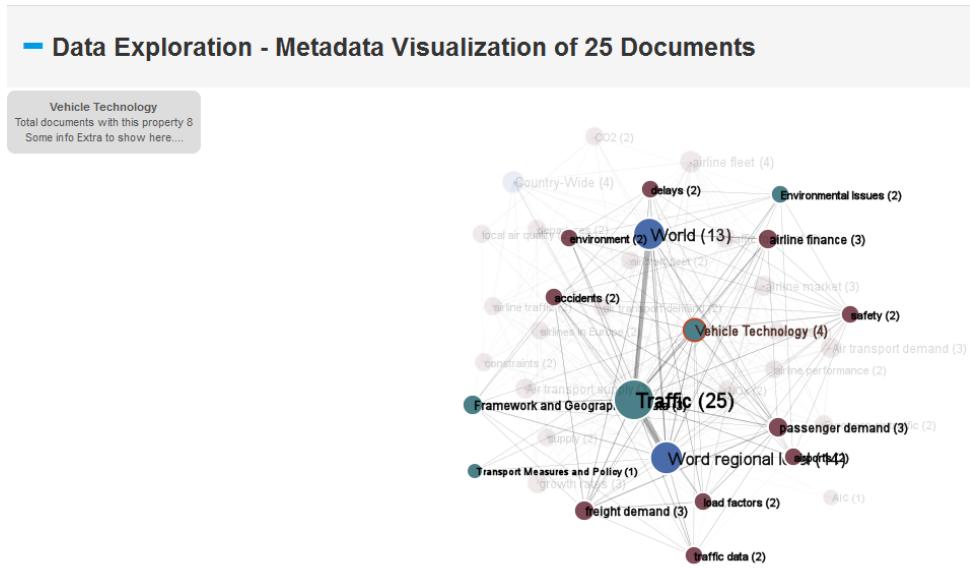


Figura 9.2: Grafo de exploración de STRADA@DLR usando KnowledgeFinder II

#### 9.4.1.2. Aristas

Las aristas representan las relaciones entre (Meta-)datos en función del número de documentos que contienen sus extremos. Mientras su grosor es directamente proporcional al a su valor, la longitud de la arista es inversamente proporcional.

Cuando se pulsa sobre una arista, su ambos (Meta-)datos situados en sus extremos son aplicados para filtrar los documentos.

Si el usuario sitúa el cursor sobre una arista (sin pulsar) todos los nodos que no tienen ninguna relación con los extremos de la arista se atenúan y se muestra un mensaje con las propiedades de la arista. En la lista de resultados y en el menú se resaltan los elementos que contienen los (Meta-)datos de los extremos.

#### 9.4.2. Full-Text

El formulario para la búsqueda Full-Text es el componente que proporciona versatilidad a la búsqueda (Imagen 9.3). El texto introducido es buscado entre los valores campos *title*, *description*, *authors* y *keywords*. A continuación se describen las distintas formas que hay para utilizar los valores para la búsqueda.



Figura 9.3: Campo Full-Text de STRADA@DLR usando KnowledgeFinder II

##### 9.4.2.1. Simple search

Por defecto, se busca los documentos que contiene los alguno valores introducidos. Por ejemplo, si se introduce los valores del código 26 se obtendrán los documentos con *germany* o *2002*.

1 `germany 2002`

Código 26: Ejemplo - Simple search

##### 9.4.2.2. Matched phrase search

Usando las comillas dobles se busca los documentos que contiene los exactamente la frase introducida. Por ejemplo, si se introduce los valores del código 27 se obtendrán los documentos que contiene exactamente *germany 2002*.

1 `"germany 2002"`

Código 27: Ejemplo - Matched phrase search

#### 9.4.2.3. Disjunctive search

Al igual que la búsqueda simple, con el operador OR (en mayúsculas) se busca los documentos que contiene los alguno valores introducidos. Por ejemplo, si se introduce los valores del código 28 se obtendrán los documentos con *germany* o *2002*.

```
1 germany OR 2002
```

Código 28: Ejemplo - Disjunctive search

#### 9.4.2.4. Conjunctive search

Usando el operador AND (en mayúsculas) se busca los documentos que contienen los valores introducidos. Por ejemplo si se introduce la cadena del código 29 se obtendrán los documentos que contienen ambos valores; *germany* y *2002*.

```
1 germany AND 2002
```

Código 29: Ejemplo - Conjunctive search

#### 9.4.2.5. Exclusive search

Con los operadores - y NOT (en mayúsculas) buscar documentos que no contienen ciertos valores. Por ejemplo si se introduce la cadena del código 30 o del código 30 se obtendrán los documentos que contienen *germany* pero no *2002*.

```
1 germany NOT 2002
```

Código 30: Ejemplo - Exclusive search

#### 9.4.2.6. Wirdcard searches

Con el uso de elementos comodines ? (para un carácter) y \* (para múltiples caracteres o ninguno) en la cadena de búsqueda se pueden realizar consultas más complejas. A continuación se muestran algunos ejemplos con comodines:

- Para buscar documentos con valores entre 2000 y 2009 se podría usar la cadena búsqueda en el cuadro de código 31.

```
1 200?
```

Código 31: Ejemplo 1 - Wirdcard search

- Para obtener los documentos con *german* o *germany* se pueden usar los comodines al final de la cadena de búsqueda (código 32).

```
1 german*
```

Código 32: Ejemplo 2 - Wirdcard search

- Los comodines también se pueden combinar entre ellos en una misma cadena de búsqueda para precisar el término a buscar. La búsqueda del código 33 encontraría *germany* pero *german*.

```
1 ge*man?
```

Código 33: Ejemplo 3 - Wirdcard search

#### 9.4.2.7. Fuzzy searches

Las búsquedas *Fuzzy* ayudan a encontrar palabras que se deletreen de forma semejante. Para ello hay que añadir el carácter ~ a la cadena de búsqueda. Por ejemplo, con los valores del código 34 se encontrarán correctamente los términos con *deutschland* aunque no esté bien escrito.

```
1 deutshland~
```

Código 34: Ejemplo - Fuzzy search

#### 9.4.2.8. Field searches

Las búsquedas se pueden concretar para los campos *title* *description* y *keywords*. Concretando los campos afectados se centra la búsqueda sólo en él. en el código 35 se muestra un ejemplo de buscar la cadena *germany* sólo en el título.

```
1 title:germany~
```

Código 35: Ejemplo - Field search

Todas las búsquedas anteriormente mostradas se pueden combinar y agrupar entre ellas. Como ejemplo se muestran los códigos 36 y 37. En el primero se buscan los documentos cuyo título tengan algo parecido a *deutshland* y su descripción no contenga la palabra *germany*. El segundo ejemplo se busca documentos con la palabra clave *transport* o *traffic* y su título *germany*.

```
1 title:deutschland~ AND (NOT description:germany)
```

Código 36: Ejemplo búsqueda combinada 1

```
1 (keywords:transport OR keywords:traffic) AND title:germany
```

Código 37: Ejemplo búsqueda combinada 2

#### 9.4.3. Menú de Navegación

En menú de navegación representa todos los filtros posibles a aplicar sobre el conjunto de datos. Estos filtros se encuentran organizados en dos niveles desplegables y con un scroll para las listas de elementos más largas.

El número de documentos que poseen cada filtro se muestra junto a éste y los elementos siempre están ordenados por este número. En la imagen 9.4 se observa los diferentes estados en que los filtros se pueden encontrar.

- **Filtro activo**

Representa un (Meta-)dato el cual el usuario puede seleccionar pulsando sobre él. Se aplicará el filtro de búsqueda de este (Meta-)dato sobre el conjunto de documentos.

- **Filtro seleccionado**

Filtro que está siendo usado entre los criterios de búsqueda. Pulsando sobre el ícono *X* de éste, el filtro se desactiva y éste vuelve a estar activo.

- **Filtro desactivado**

Representa un filtro que no puede ser usado para el conjunto de documentos actual ya que ningún documento lo tiene entre sus (Meta-)datos.

En la imagen 9.4 también se observan los elementos para configurar la complejidad del grafo de exploración (*Hide in Graph / Show in Graph* en las categorías principales y *Show Group / Hide Group*) en los grupos de filtros de segundo nivel. De esta forma, se pueden seleccionar los elementos que se desea ver representado en el grafo de exploración.

- **Show in Graph**

Pulsando este botón se muestran, si procede, en el grafo de exploración los elementos del subnivel siguiente.

- **Hide in Graph**

Para ocultar en el grafo de exploración los elementos del primer subnivel pertenecientes a este conjunto de filtros, simplemente se debe pulsar este botón.

- **Show Group**

Se muestran en el grafo de exploración los subfiltros de este grupo.

- **Hide Group**

Se ocultan en el grafo de exploración todos los subfiltros de este grupo.

## Navigation

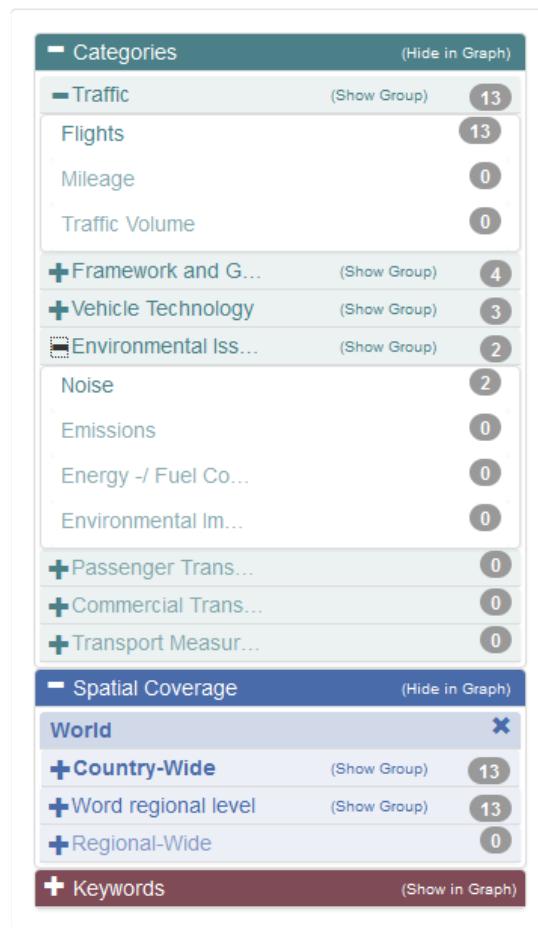


Figura 9.4: Menú de navegación de STRADA@DLR usando KnowledgeFinder II

### Interacción dinámica

El usuario tiene la opción de usar este menú para interaccionar con el reto de componentes sin aplicar ningún filtro ni modificar los elementos mostrados en el grafo. Para ello, simplemente situándose sobre los elementos del menú sobre un filtro éste se marca con un círculo y los elementos que no pertenecen a su componente conexa se difuminan (ver imagen 9.2. Junto al grafo aparece también un mensaje indicando propiedades del filtro implicado. A parte de ésto, en la lista de resultados se resaltan también los que posean este filtro.

#### 9.4.4. Lista de Resultados

La lista de resultados representa el conjunto de documentos ordenados que satisfacen los filtros de búsqueda que se aplican.

Esta lista está organizada a través de un sistema de paginación y puede ser reordenada. El usuario puede elegir entre algunas de las opciones del elemento *dropdown* para personalizar el orden de

la lista. Las opciones son:

- A-Z Title: Alfabéticamente según el título del documento.
- Z-A Title: Alfabéticamente en orden inverso según el título del documento.
- Relevante: Por orden ascendente de puntuación en la búsqueda.
- Actuality: Ordenado por fecha de modificación ascendentemente.

La lista está diseñada con forma de acordeón. Se pueden expandir y contraer todos los elementos a través del botón *Open all/Close all* o pulsando sobre los títulos de los documentos. Cuando un elemento de la lista se encuentra expandido abierto se puede observar la descripción y el botón *More Information* el cual muestra toda la información disponible del documento en una ventana emergente (imagen 9.5).

The screenshot shows a search results page with the following layout:

- Sort by:** Relevance (dropdown menu)
- Open all** button
- Showing 1-10 of 29 entries.**
- Pagination:** 1 2 3 →
- Document List:**
  - + ACI Airport Economics Survey** (expanded)
    - Global Traffic Forecast**
    - Keywords**: Passenger demand development forecast, freight and movements forecast, traffic forecast by world region, traffic forecast by country, air transport's long-term development, growth rates
    - Description**: The Global Traffic Forecast is a regular publication by ACI, which is the world's trade association for airports and DKMA as one leading aviation market consulting company. The forecast is published every year and provides an overview of the expected air traffic development for the next 20 years ...
    - More Information**
  - + STAR (Summary of Traffic and Airline Results)**
  - + World Airline Report**
  - + Global Market Forecast**
  - + Monthly Passenger/Cargo Traffic Report**
  - + Online Fleets**
  - + Current Market Outlook**
  - + World Air Cargo Forecast**
  - + Annual Analyses of the European Air Transport Market**
- Pagination:** 1 2 3 →

Figura 9.5: Lista de Resultados

Tanto en la ventana emergente como los valores de la lista de resultados, si una búsqueda Full-Text se encuentra entre los valores de la consulta, las cadenas que coinciden con estas búsquedas son resaltadas. En las imágenes 9.6 y 9.7 se observa este comportamiento para la búsqueda de la cadena *traffic~* en la lista de resultados y en la ventana emergente de detalles respectivamente.

### Interacción dinámica

Cuando el usuario se mueve sobre alguno de elementos de la lista de resultados, todos los nodos sin relación con este documento y sus relaciones son difuminados y en el menú de navegación son resaltados los (Meta-)datos del documento afectado.

**+ Global *Traffic* Forecast**

**- Monthly Passenger/Cargo *Traffic* Report**

## Keywords

U.S. air traffic, American carriers, passenger and cargo traffic, capacity and demand

## Description

The Passenger and Cargo *Traffic* Reports are a monthly publication of the organisation association for U.S. airlines with a 75-years history and represents the majority of the

[More Information](#)

**+ STAR (Summary of *Traffic* and Airline Results)**

**+ ACI Airport Economics Survey**

**+ Medium-Term Forecast**

**+ Air Transport Statistics**

Figura 9.6: Resaltado de Full-Text en la lista de resultados en STRADA@DLR usando KnowledgeFinder II

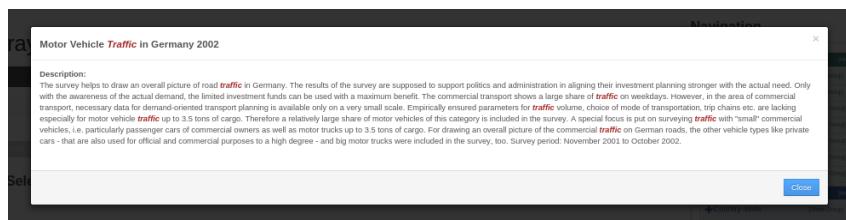


Figura 9.7: Resaltado de Full-Text en la ventana emergente en STRADA@DLR usando KnowledgeFinder II

### 9.4.5. Selección actual

La selección actual (imagen 9.8) es una lista de botones que representan los filtros que actualmente están activos y las cadenas de búsqueda introducidas a través del Full-Text.

Cuando se pulsa sobre alguno de ellos, el filtro se elimina de conjunto de filtros aplicados y este botón desaparece de la lista.

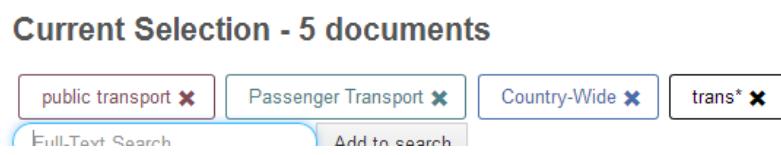


Figura 9.8: Selección actual de STRADA@DLR usando KnowledgeFinder II

# Capítulo 10

## Conclusiones

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

### 10.1. Objetivos alcanzados

Tras la finalización del desarrollo de KnowledgeFinder II puedo afirmar que se han cubierto todos los retos y objetivos planteados a la comienzo de este Proyecto de Fin de Carrera. A continuación se describen los objetivos generales y específicos más representativos alcanzados.

#### 10.1.1. Visualización del Conocimiento

La primera versión de KnowledgeFinder proporciona un sistema de búsqueda insuficiente para la información y las necesidades de los portales al estar centrada únicamente en el texto. En esta nueva versión, la complejidad de las relaciones de los (Meta-)datos y de la información mostrada a través de las herramientas de visualización proporcionan a los usuarios de los portales que utilicen KnowledgeFinder II como base de su desarrollo, una interfaz de usuario intuitiva, dinámica e interactiva donde se representan las estructuras de datos complejas del conocimiento.

Con un esquema organizativo para la interfaz de usuario semejante a la antigua versión, la curva de aprendizaje y adaptación al nuevo sistema no es muy marcada. La visualización aparece como el elemento principal de la búsqueda pero sin menospreciar las posibilidades a través del menú y de la búsqueda Full-Text.

La experiencia de usuario se ha visto claramente beneficiada. Todos los componentes de la interfaz interactúan entre ellos y proporciona al usuario información instantánea de las relaciones y estructuras de los documentos y los (Meta-)datos. El usuario ya no verá el portal como un simple buscador de documentos sino como un portal sobre el conocimiento.

Por todo lo anterior, puedo afirmar que este nuevo sistema genera nuevos conocimientos y resultados que facilitan el proceso investigador del “ Transporte y la Movilidad” en el DRL.

#### 10.1.2. Adaptación a nuevas Fuentes de Datos

Gracias a la redifinición del proceso para la importación de los datos y a la elección de Solr como motor de búsqueda, se ha obtenido un producto que acepta la incorporación de nuevas

fuentes de datos (locales y remotas) al índice de búsqueda simplemente adaptando la configuración. Siendo esta idea uno de sus puntos clave durante la implementación, el resto de la aplicación es igualmente configurable y flexible para la incorporación de nuevos orígenes de datos.

Gracias a estas consideraciones durante el desarrollo, el trabajo conjunto de las diferentes fuentes de información pertenecientes al DRL ha impulsado notablemente el programa para “Integración e interoperabilidad de las bases de datos sobre el transporte en el DRL”.

## 10.2. Mejoras Transversales de KnowledgeFinder II

A parte de los objetivos alcanzados que se plantearon durante la concepción del trabajo presente, se han obtenido importantes mejoras respecto a la versión anterior del KnowledgeFinder II. A continuación se explican algunas de las más significativas.

### 10.2.1. Mejora del Código

A pesar de un proceso realizado sobre KnowledgeFinder de refactorización y mejora del código, la calidad del mismo seguía siendo bastante deficiente (duplicación de código, código no usado, ...). Por otra parte, la configuración de cada instancia para los distintos portales se realizaba a nivel de código lo que empeoraba aun más la situación.

Gracias al replanteamiento de todo el sistema para KnowledgeFinder II se ha conseguido que atajar todos estos problemas usando ficheros de configuración adaptables para cada portal. Por ejemplo, para la instancia del portal STRADA@DLR de KnowledgeFinder se han contado 424 ficheros Java de implementación propia con un total de 67565 líneas de código. Para la misma instancia pero basado en KnowledgeFinder II, 89 ficheros de código Java, JavaScript y Sass de implementación propia con un total de 11856 líneas de código.

### 10.2.2. Mejora del Rendimiento

Usando Solr y la lectura directa del repositorio SVN, el proceso de importación para los datos ha reducido su proceso de creación del índice de búsqueda de 25 minutos a 2 minutos aproximadamente usando KnowledgeFinder II.

Otra gran mejora ha sido el rendimiento de la interfaz de usuario. En los portales donde se usan actualmente la versión inicial la reacción de la interfaz es lenta y torpe, llegando a confundir al usuario. Con la versión, incluso proveyendo de la visualización, esta interfaz reacciona diligentemente a las peticiones del usuario.

### 10.2.3. Desarrollo más flexible

Usando MVC en la nueva versión, el sistema es más adaptable a posibles cambios estructurales de las necesidades de los portales. Por ejemplo, si se decidiera por cambiar el sistema donde se aloja los portales de búsqueda, actualmente Liferay, no implicaría una reimplementación de todo el sistema.

---

### 10.3. Lecciones aprendidas

En este apartado se detallan las buenas prácticas adquiridas, tanto tecnológicas con procedimentales, y el conocimiento conseguido durante la ejecución del presente Proyecto de Fin de Carrera.

Mi experiencia trabajando en el DRL para el SC-VSS y la realización de este proyecto conjuntamente los institutos VF y FW ha sido realmente gratificante y enriquecedora en el ámbito profesional y personal. Por la naturaleza de esta institución, la burocracia necesaria para cada requerimiento para el proyecto a resultado a veces frustrante, siendo un riesgo incluso para el cumplimiento de la planificación establecida. A pesar de ello, esta burocracia, una vez tenida en cuenta con sus plazos, ha facilitado la cooperación y comunicación con el resto de entidades del DRL implicadas.

El replanteamiento desde cero de la implementación de KnowledgeFinder II fue una experiencia profesional muy valiosa. Partiendo de la base que la anterior versión fue trabajo de dos tesis doctorales y de bastante dedicación para su implementación en MONITOR-Portal, STRADA@DLR y ELIB-Portal, fue comprensible el primer rechazo y la negativa por parte de SC-VSS de reimplementar los componentes de software ya concluidos y que se alejaban de la finalidad inicial del proyecto. A pesar de ello, con una extensa justificación basada en la calidad del producto y su aplicación futura, fue posible la implementación del producto completamente y con la satisfacción y agrado final del SC-VSS por el producto obtenido.

A nivel tecnológico, he adquirido una gran experiencia en el desarrollo de herramientas para Solr y la utilización de éste. Gracias a él y al servicio web implementado, he afianzado mis conocimientos en Java y en el desarrollo de pruebas para la fase de testing. Por otra parte, durante el estudio y el desarrollo de la visualización, he sido fascinado por las nuevas posibilidades que ofrece HTML5 junto a herramientas gráficas de JavaScript como la usada D3.js. Partiendo de un desconocimiento casi total de JavaScript, gracias al desarrollo de esta parte de la aplicación, he obtenido un dominio considerable en este lenguaje. Por último, también he conseguido una base estable de programación de estilos CSS usando Sass y ayudado por la herramienta Compass.

Por otra parte, a nivel organizativo, la duración estimada inicial del proyecto (gráfica de Gantt 2.3) era de seis meses. Hasta el momento del replanteamiento de la implementación se cumplieron aproximadamente todos los plazos planificados para cada fase de proyecto excepto la fase de análisis que se alargó dos semanas más de lo estipulado. Cuando se aceptó la replanificación del proyecto (gráfica de Gantt 2.4), se dispuso más tiempo para dedicarle, principalmente, a las fases de análisis e implementación. Aunque este primero se finalizó tres semanas antes, se tuvo que realizar otra fase de análisis del nuevo sistema. Con estas modificaciones he aprendido la importancia para la gestión de proyectos de la estimación temporal y que, a pesar de la necesidad de cumplir los plazos, la flexibilidad de proceso de desarrollo debe estar ligado al tipo de proyecto.

### 10.4. Trabajo futuro

Una vez concluida esta versión del producto, hay mejoras que podrían indicar del trabajo futuro en el que se encaminará la culminación de este proyecto piloto. A continuación se muestra una lista con las posibles mejoras que, por falta de recursos o porque se salen del marco del proyecto, no han sido implementadas:

**■ Importación de nuevas fuentes de datos**

Durante el desarrollo del proyecto sólo se ha tenido la posibilidad de trabajar con los datos de pruebas de MONITOR-Portal y Clearingstellen Verkehr. La incorporación de nuevas fuentes de datos de otros portales e institutos se plantea como trabajo futuro cercano.

**■ Importación de otros formatos**

Actualmente sólo se aceptan elementos en formato JSON proveniente de las propiedades de SVN. Para otras fuentes de datos, sería interesante la importación a través de otros métodos, por ejemplo, ficheros XML.

**■ Adaptación a nuevos formatos**

El formato de los datos de los repositorios SVN se encuentra actualmente en revisión y redefinición. Por ello, como trabajo futuro necesario será la adaptación del proyecto a los mismos.

**■ Temas para Liferay**

Para la implantación en producción del mismo para el portal STRADA@DLR, la creación de unos estilos para Liferay y la adaptación del código HTML.

**■ Nuevos componentes para la visualización**

Para completar la experiencia de usuario se pueden añadir nuevos componentes de visualización como, por ejemplo, Crossfilter para la selección de intervalos de tiempo o Leaflet para la visualización espacial de los documentos.

**■ Aplicación del servicio web**

Añadiendo funcionalidad al servicio web se puede obtener una API más rica para el uso de los datos por otros sistemas externos.

**■ Búsqueda semántica**

Añadir algún modelado de datos, por ejemplo RDF, para utilizar el potencial de la búsqueda y web semántica.

**■ Linked Data**

Añadir el conocimiento a Linked Data (<http://linkeddata.org/>).

# Bibliografía

- [BMW14] Bundesministerium für Wirtschaft und Energie BMWi. Haushalt 2014 - einzelplan 09. <http://www.bmwi.de/DE/Ministerium/haushalt,did=306790.html>, 2014.
- [Bos11] Mike Bostock. D3 + leaflet. <http://bost.ocks.org/mike/leaflet/>, 2011.
- [Cis10] Peter Cissek. Strategische unternehmensplanung in einer data warehouse-umgebung unterstützt durch ein wissensmanagementsystem, 2010.
- [Con] Cytoscape Consortium. <http://cytoscapeweb.cytoscape.org/>.
- [Den12] Andreas Dengel. Semantische technologien: Grundlagen. konzepte. anwendungen. heidelberg : Spektrum akademischer verlag, 2012.
- [DGS] Dr. Peter Hruschka Kontakt Dr. Gernot Starke. Softwarearchitektur arc42. <http://arc42.de/>.
- [DLR14] DLR Standorte DLR. <http://www.dlr.de/dlr/desktopdefault.aspx/tabid-10175/>, 2014.
- [dPTyAP] Ministerio de Política Territorial y Administración Pública. Magerit v3. [http://administracionelectronica.gob.es/?\\_nfpb=true&\\_pageLabel=P800292251293651550991&langPae=es&detalleLista=PAE\\_1276529683497133](http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P800292251293651550991&langPae=es&detalleLista=PAE_1276529683497133). Método formal para investigar los riesgos que soportan los Sistemas de Información y para recomendar las medidas apropiadas que deberían adoptarse para controlar estos riesgos.
- [euS12] The evolving ultrasaurus (Sarah). D3.js experiments in the console. <http://www.ultrasaurus.com/>, 2012.
- [fA] Bundesagentur für Arbeit. Tarifvertrag für den Öffentlichen dienst (tvöd). <http://oeffentlicher-dienst.info/tvoed/>.
- [FK14] Institut für Fahrzeugkonzepte FK. <http://www.dlr.de/fk/>, 2014.
- [fTR] DLR Institute for Transport Research. Utransport statistics at the german aerospace center (dlr) - a first step towards interlinked statistics portals.
- [fTR13] DLR Institute for Transport Research. Using dublin core standard for the metadata description of transport statistics – practical experience from a project dedicated to the set-up of an interlinked statistics portal, 2013.
- [FW14] Institut für Flughafenwesen und Luftverkehr FW. <http://www.dlr.de/fw/>, 2014.

- [gd] gwtd3/gwt d3. Cgwt-d3: the gwt wrapper around d3.js. <https://github.com/gwtd3/gwt-d3>.
- [GEE00] Magnus Penker GHans-Erik Eriksson. Business modeling with uml : Business patterns at work, 2000.
- [Inca] Google Inc. Google html/css style guide. <https://google-styleguide.googlecode.com/svn/trunk/htmcssguide.xml>.
- [Incb] Google Inc. Google javascript style guide. <https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>.
- [Inf14] InfoJobs. Salarios promedios infojobs. <http://plandecarrera.infojobs.net/>, 2014.
- [Joe12] Dr. William J Joel. Course notes: Object-oriented software engineering (cs350), 2012.
- [Kom06] Satya Komatineni. Reshaping it project delivery through extreme prototyping. <http://www.onjava.com/pub/a/onjava/2006/11/15/reshaping-it-project-delivery-through-extreme-prototyping.html>, 2006.
- [LAB] Nodus LAB. <http://noduslabs.com/>.
- [Lif14] Liferay. Liferay portal 6.2 developer's guide - json web services. <https://www.liferay.com/es/documentation/liferay-portal/6.2/development/-/ai/json-web-services-liferay-portal-6-2-dev-guide-05-en>, 2014.
- [Ora99] Oracle. Code conventions for the java programming language. <http://www.oracle.com/technetwork/java/index-135089.htm>, 1999.
- [RS99] Doug Rosenberg and Kendall Scott. Use case driven object modeling with um, 1999.
- [Sou] Square Open Source. Crossfilter - fast multidimensional filtering for coordinated views. <http://square.github.io/crossfilter/>.
- [Spa00] Geoffrey Sparks. The business process model, sparx systems, 2000.
- [TS14] Institut für Verkehrssystemtechnik TS. <http://www.dlr.de/ts/>, 2014.
- [VF14] Institut für Verkehrsorschung VF. <http://www.dlr.de/vf/>, 2014.
- [Wika] Apache Wiki. Installing solr. <https://cwiki.apache.org/confluence/display/solr/Installing+Solr>.
- [Wikb] Apache Wiki. Solr cloud. <https://cwiki.apache.org/confluence/display/solr/SolrCloud>.
- [Wikc] Apache Wiki. Solr security. <https://wiki.apache.org/solr/SolrSecurity>.

# GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within

that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, TeXinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text. The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of

Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the

copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.