

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Grupo Knowledge Reuse (KR)



Grado en Ingeniería Informática
Proyecto Fin de Grado

Sistema de Gestión de KOS: Tesauro

Autor: Álvaro García Martínez
Tutor: Anabel Fraga
Febrero, 2013

Resumen del Proyecto

Nombre del alumno	Álvaro García Martínez
Título	Sistema de Gestión de KOS: Tesaurus
Descripción	El proyecto tiene como objetivo principal el de crear un tesaurus como una herramienta de gestión del conocimiento, dándole las principales características que un tesaurus debe poseer para ser usado en cualquier ámbito, y con el fin de ser integrado en una herramienta más compleja y potente como es el KnowledgeMANAGER .
Summary	The project's main objective is to create a thesaurus as a knowledge management tool, giving the main features that a thesaurus must possess to be used in any field, and in order to be integrated in a more complex and powerful tool as is KnowledgeMANAGER .

Tabla 1 - Resumen del proyecto

Agradecimientos

En este documento se plasman todos los esfuerzos hechos durante estos 4 años de carrera y no querría que ninguna de las personas que me ha apoyado y aguantado a lo largo de este tiempo se quede fuera de mi recuerdo.

En primer lugar, a todos mis compañeros de carrera que son los que junto conmigo saben todo el esfuerzo que supone llegar a este punto.

En segundo lugar, agradecer a mis amigos el hecho de compartir aspectos de mi carrera que no son de su especial agrado pero que han sabido aguantármelos.

En tercer lugar, querría darle las gracias a mi familia por empeñarse cada día de mi vida que debía entrar en la universidad y estudiar una carrera para tener un futuro más provechoso. Les doy las gracias por ello.

En cuarto y último lugar, agradecer el apoyo y ayuda de mi grupo de departamento y en especial a mi tutor, el cual me ha sabido aconsejar y guiar en todo el proceso de desarrollo de mi proyecto fin de grado.

Gracias a todos ellos.

Índice de Contenidos

Resumen del Proyecto	3
Agradecimientos.....	5
Índice de Contenidos.....	7
Índice de Tablas	10
Índice de Ilustraciones	16
Capítulo 1. Introducción	19
1.1. Visión General.....	20
1.2. Objetivos	20
1.3. Contenido de la memoria	20
Capítulo 2. Estado de la cuestión	22
2.1. Ontologías.....	23
2.1.1. Historia	23
2.1.2. La Web Semántica.....	23
2.1.3. Componentes y tipos	24
2.1.4. Gestión de Ontologías	25
2.1.4.1 ¿Qué es un KOS?	25
2.1.4.2. Ventajas de los KOS.....	26
2.1.4.3. Limitaciones de los KOS.....	26
2.2. La herramienta: KnowledgeMANAGER	26
2.2.1. Componentes de la herramienta	27
2.2.2. Gestores de ontologías en el Mercado. Comparativas	27
2.3. Tecnologías necesarias para el desarrollo.....	31
2.3.1. Sistemas operativos.....	32
2.3.2. Lenguaje de programación	32
2.3.3. Entorno de programación.....	33
2.3.4. Sistema gestor de base de datos	34
2.3.5. Gestión del proyecto	35
2.3.6. Gestión de versiones	36
2.3.7. Gestión de pruebas.....	36
2.3.8. Metodologías de desarrollo.....	37
2.3.9. Resumen	39
Capítulo 3. Análisis.....	40
3.1. Definición del sistema.....	41

3.1.1. Determinación del alcance del sistema	41
3.1.2. Identificación del entorno tecnológico	41
3.1.3. Especificación de estándares y normas	42
3.1.4. Identificación de los usuarios participantes y finales	43
3.1.5. Seguridad requerida en el análisis del sistema de información	43
3.2. Establecimiento de requisitos	44
3.2.1. Obtención de requisitos	45
3.2.2. Especificación de casos de uso	74
3.2.2.1. Términos	74
3.2.2.2. Relaciones	79
3.2.2.3. Sinónimos	85
3.2.2.4. Otras funcionalidades.....	89
3.2.3. Descripción de la Interacción de Objetos	93
3.3. Identificación de subsistemas de análisis	95
3.4. Análisis de clases	95
3.4.1. Identificación y especificación de clases	95
3.4.1.1. Clases de Entidad	96
3.4.1.2. Clases de apoyo a la interfaz	96
3.4.1.3. Clases de Interfaz.....	97
3.4.1.4. Clases de control.....	102
3.4.2. Identificación de asociaciones y agregaciones.....	104
3.5. Elaboración del modelo de datos	107
3.5.1. Especificación necesidades migración de datos y carga inicial	108
3.6. Definición de interfaces de usuario.....	109
3.6.1 Especificación de Principios Generales de la Interfaz	109
3.6.2 Identificación de Perfiles y Diálogos.....	109
3.6.3 Especificación Formatos Individuales de la Interfaz de Pantalla	110
3.6.4 Especificación del Comportamiento Dinámico de la Interfaz	110
3.6.5 Especificación de Formatos de Impresión.....	111
3.7. Análisis de consistencia y especificación de requisitos.....	111
3.8. Especificación del plan de pruebas	111
3.8.1. Definición del alcance de pruebas	111
3.8.2. Definición de requisitos para el entorno de pruebas.....	112
3.8.2. Definición de las pruebas de aceptación del sistema.....	112
3.8.2.1. Términos	112

3.8.2.2. Relaciones	119
3.8.2.3. Sinónimos	122
3.8.2.4. Otras funcionalidades	124
Capítulo 4. Diseño	128
4.1. Definición de la arquitectura del sistema	129
4.1.1. Definición de los niveles de arquitectura	129
4.1.2. Especificación de excepciones	130
4.1.3. Especificación de estándares y normas de diseño	130
4.2. Diseño de la arquitectura	130
4.2.1. Diseño de subsistemas	130
4.3. Revisión de la interfaz de usuario	132
4.4. Diseño de clases	133
4.4.1. Identificación y especificación de clases	133
4.4.1.1 Clases de apoyo a la interfaz	133
4.4.1.2. Clases de Interfaz	135
4.4.1.3. Clases de control	147
4.5. Diseño físico de datos	158
4.5.1 Diseño del modelo físico de datos	159
4.6. Diseño de migración y carga inicial de datos	160
4.6.1. Especificación del entorno	160
4.6.2. Diseño del procedimiento de carga inicial	160
4.7. Especificación técnica del plan de pruebas	162
4.7.1. Especificación del entorno de pruebas	162
4.7.2. Especificación técnica de niveles de prueba	163
4.7.2.1. Relaciones	164
4.7.2.2. Elementos de las relaciones	166
4.7.2.3. Sinónimos	168
4.7.2.4. Términos	170
4.7.2.5. Términos raíces	177
4.7.2.6. Hijos y padres	179
4.7.2.7. Otras funcionalidades	180
Capítulo 5. Planificación y Presupuesto	184
5.1. Planificación	185
5.2. Presupuesto	189
Capítulo 6. Conclusiones y Trabajos Futuros	192

6.1. Conclusiones.....	193
6.1.1. Resultados obtenidos en la planificación	193
6.1.2. Evaluación del trabajo realizado	194
6.2. Trabajos Futuros	195
Capítulo 7. Anexos.....	196
7.1. Glosario.....	197
7.2. Bibliografía	199
7.3. Manual de usuario de la aplicación.....	201
7.3.1. Formulario Inicial.....	201
7.3.2. Creación de elementos de un tesouro.....	202
7.3.3. Modificación de un término tesouro.....	204
7.3.4. Búsqueda de un término el tesouro.....	205
7.3.5. Copiar y mover términos	206
7.3.6. Borrado de elementos de un tesouro	207
7.3.7. Importación y exportación de un tesouro.....	209

Índice de Tablas

Tabla 1 - Resumen del proyecto	3
Tabla 2 - Comparativa de las herramientas de gestión analizadas.....	31
Tabla 3 - Valoración de los posibles sistemas operativos	32
Tabla 4 - Valoración de los posibles lenguajes de programación.....	33
Tabla 5 - Valoración de los posibles entornos de programación	34
Tabla 6 - Valoración de los posibles gestores de base de datos	34
Tabla 7 - Resumen de las tecnologías para el desarrollo del proyecto.....	39
Tabla 8 - Plantilla para rellenar los requisitos	44
Tabla 9 - RSF - 001: Carga de datos inicial.....	45
Tabla 10 - RSF - 002: Visualización del tesouro.....	45
Tabla 11 - RSF - 003: Acciones sobre términos.....	46
Tabla 12 - RSF - 004: Añadir un término raíz	47
Tabla 13 - RSF - 005: Añadir una nueva relación.....	47
Tabla 14 - RSF - 006: Añadir un hijo	48
Tabla 15 - RSF - 007: Añadir un sinónimo.....	48
Tabla 16 - RSF - 008: Añadir una asociación	49
Tabla 17 - RSF - 009: Añadir otra relación	49

Tabla 18 - RSF - 010: Editar/ Mostrar información del término	50
Tabla 19 - RSF - 011: Eliminar un término.....	50
Tabla 20 - RSF - 012: Eliminar el término y hacer sus hijos raíces.....	51
Tabla 21 - RSF - 013: Eliminar el término y sus hijos	51
Tabla 22 - RSF - 014: Mover sus hijos a otro término raíz.....	52
Tabla 23 - RSF - 015: Mover sus hijos a otro término padre.....	52
Tabla 24 - RSF - 016: Sustituir por un sinónimo	53
Tabla 25 - RSF - 017: Sustituir por un hijo.....	53
Tabla 26 - RSF - 018: Mover el término a la raíz	54
Tabla 27 - RSF - 019: Mezclar un término con otro término	54
Tabla 28 - RSF - 020: Expandir todo	55
Tabla 29 - RSF - 021: Búsqueda web	55
Tabla 30 - RSF - 022: Añadir buscadores.....	56
Tabla 31 - RSF - 023: Eliminar buscadores.....	56
Tabla 32 - RSF - 024: Opciones de búsqueda	57
Tabla 33 - RSF - 025: Buscadores	57
Tabla 34 - RSF - 026: Refrescar	58
Tabla 35 - RSF - 027: Búsqueda simple de términos	58
Tabla 36 - RSF - 028: Visualización de términos en vista filtrada	59
Tabla 37 - RSF - 029: Volver a la vista jerárquica manteniendo estado.....	59
Tabla 38 - RSF - 030: Volver a la vista jerárquica sin mantener el estado	60
Tabla 39 - RSF - 031: Búsqueda avanzada de términos en el tesauro.....	60
Tabla 40 - RSF - 032: Tooltip para términos en el tesauro.....	61
Tabla 41 - RSF - 033: Información del término	62
Tabla 42 - RSF - 034: Término	62
Tabla 43 - RSF - 035: Miga de pan del término	63
Tabla 44 - RSF - 036: Relaciones	63
Tabla 45 - RSF - 037: Añadir nueva relación.....	64
Tabla 46 - RSF - 038: Eliminar relación.....	64
Tabla 47 - RSF - 039: Mostrar otro participante de la relación.....	65
Tabla 48 - RSF - 040: Sinónimos	65
Tabla 49 - RSF - 041: Añadir sinónimo.....	66
Tabla 50 - RSF - 042: Eliminar sinónimo.....	66
Tabla 51 - RSF - 043: Modificar sinónimo	67
Tabla 52 - RSF - 044: Usar por el término.....	67

Tabla 53 - RSF - 045: Importación	68
Tabla 54 - RSF - 046: Exportación	68
Tabla 55 - RSF - 047: Estadísticas.....	69
Tabla 56 - RSF - 048: Pestaña avanzada del término	69
Tabla 57 - RSF - 049: Modificar información avanzada del término.....	70
Tabla 58 - RSF - 050: Refresco visual.....	71
Tabla 59- RSF - 051: Drag & Drop	71
Tabla 60 - RSIn - 052: Visualización de tesauro y detalles	72
Tabla 61 - RSCa - 053: Ordenación de los listados.....	72
Tabla 62 - RSR - 054: Carga del tesauro	73
Tabla 63 - RSS - 055: Login.....	73
Tabla 64 - CU-001: Añadir un nuevo término raíz.....	75
Tabla 65 - CU-002: Modificar un término.....	75
Tabla 66 - CU-003: Eliminar un término	76
Tabla 67 - CU-004: Copiar un término.....	76
Tabla 68 - CU-005: Mezclar un término en otro.....	77
Tabla 69 - CU-006: Expandir un término	78
Tabla 70 - CU-007: Buscar un término	78
Tabla 71 - CU-008: Mover un término a la raíz.....	79
Tabla 72 - CU-009: Añadir una nueva relación	80
Tabla 73 - CU-010: Añadir un nuevo hijo.....	81
Tabla 74 - CU-011: Añadir una nueva asociación.....	82
Tabla 75 - CU-012: Añadir otra relación	83
Tabla 76 - CU-013: Eliminar una nueva relación	84
Tabla 77 - CU-014: Mostrar el otro participante de la relación.....	85
Tabla 78- CU-015: Añadir un sinónimo.....	86
Tabla 79 - CU-016: Modificar un sinónimo	87
Tabla 80 - CU-017: Eliminar un sinónimo.....	88
Tabla 81 - CU-018: Intercambiar sinónimo por término	89
Tabla 82 - CU-019: Refrescar	90
Tabla 83 - CU-020: Importar.....	90
Tabla 84 - CU-021: Exportar	91
Tabla 85 - CU-022: Añadir un buscador	91
Tabla 86 - CU-023: Eliminar un buscador.....	92
Tabla 87 - CU-024: Buscador web	92

Tabla 88 - Análisis de VocabularyDataSet.....	96
Tabla 89 - Análisis de GrammaticalDataSet	97
Tabla 90 - Análisis de RulesFamiliesDataSet	97
Tabla 91 - Análisis de ArtifactDataSet	97
Tabla 92 - Análisis de Thesaurus	98
Tabla 93 - Análisis de TermTreeView	98
Tabla 94 – Análisis de TermCompositeObservable.....	99
Tabla 95 - Análisis de TermThesaurusDetails.....	99
Tabla 96 - Análisis de General	100
Tabla 97 - Análisis de Relationships.....	100
Tabla 98 - Analisis de Synonyms	101
Tabla 99 - Análisis de AdvancedTerm.....	101
Tabla 100 - Análisis de Export.....	101
Tabla 101 - Análisis de Import	102
Tabla 102 - Análisis de VocabularyRepository.....	102
Tabla 103 - Análisis de GrammaticalRepository	103
Tabla 104 - Análisis de RulesFamiliesRepository	103
Tabla 105 - Análisis de ArtifactRepository	103
Tabla 106 - Análisis de ThesaurusRepository.....	104
Tabla 107 - Asociación Relación – Artefacto	105
Tabla 108 - Asociación Relación - Semántica	105
Tabla 109 - Asociación KnowledgeElement – Artefacto.....	105
Tabla 110 - Asociación KnowledgeElement – Vocabulario.....	106
Tabla 111 - Asociación Vocabulario – Etiqueta sintáctica.....	106
Tabla 112 - Asociación Vocabulario – Semántica	106
Tabla 113 - Agregación KnowledgeElement – Relación	107
Tabla 114 - Iconos característicos de la interfaz.....	110
Tabla 115 - PA-001: Añadir un término raíz	112
Tabla 116 - PA-002: Ver detalles de un término.....	113
Tabla 117 - PA-003: Modificar un término.....	113
Tabla 118 - PA-004: Comprobar campos obligatorios de un término	114
Tabla 119 - PA-005: Buscar un término mediante filtros de búsqueda simple.....	115
Tabla 120 - PA-006: Buscar un término mediante filtros de búsqueda avanzado	116
Tabla 121 - PA-007: Eliminar término.....	117
Tabla 122 - PA-008: Mover un término a la raíz	117

Tabla 123 - PA-009: Drag & Drop de un término	118
Tabla 124 - PA-010: Mezclar un término con otro término	118
Tabla 125 - PA-011: Expandir término	119
Tabla 126- PA-012: Añadir una nueva relación.....	119
Tabla 127 - PA-013: Añadir una nueva asociación	120
Tabla 128 - PA-014: Añadir un nuevo hijo	120
Tabla 129 - PA-015: Eliminar una relación	121
Tabla 130 - PA-016: Comprobar campos obligatorios de una relación	121
Tabla 131 - PA-017: Añadir un sinónimo	122
Tabla 132 - PA-018: Modificar un sinónimo	122
Tabla 133 - PA-019: Intercambiar un sinónimo por un término	123
Tabla 134 - PA-020: Comprobar campos obligatorios de un sinónimo	123
Tabla 135 - PA-021: Eliminar un sinónimo.....	124
Tabla 136 - PA-022: Añadir buscador	124
Tabla 137 - PA-023: Buscador web	125
Tabla 138 - PA-024: Eliminar buscador	125
Tabla 139 - PA-025: Refrescar	126
Tabla 140 - PA-026: Importar	126
Tabla 141 - PA-027: Exportar	127
Tabla 142 - Diseño de VocabularyDataSet.....	134
Tabla 143 - Diseño de GrammaticalDataSet	134
Tabla 144 - Diseño de RulesFamiliesDataSet	135
Tabla 145 - Diseño de ArtifactDataSet	135
Tabla 146- Diseño de Thesaurus	138
Tabla 147 - Diseño de TermTreeView.....	139
Tabla 148 - Diseño de TermCompositeObservable	140
Tabla 149 - Diseño de TermThesaurusDetails.....	141
Tabla 150 - Diseño de General	141
Tabla 151 – Diseño de Relationships	142
Tabla 152 - Diseño de Relationship.....	142
Tabla 153 – Diseño de AddAssociation	143
Tabla 154 – Diseño de SearchTerm	144
Tabla 155 – Diseño de Synonyms.....	144
Tabla 156 - Diseño de Synonym	144
Tabla 157 - Diseño de AdvancedTerm.....	145

Tabla 158 - Diseño de NewTerm	145
Tabla 159 - Diseño de DeleteTerm	146
Tabla 160 - Diseño de MergeTerm	146
Tabla 161 - Diseño de Export	147
Tabla 162 - Diseño de Import	147
Tabla 163 - Diseño de VocabularyRepository	149
Tabla 164 - Diseño de GrammaticalRepository	151
Tabla 165 - Diseño de RulesFamiliesRepository	153
Tabla 166 - Diseño de ArtifactRepository	153
Tabla 167 - Diseño de ThesaurusRepository	158
Tabla 168 - PU-001: Obtención relaciones del tesouro	164
Tabla 169 - PU-002: Relación transitiva	164
Tabla 170 - PU-003: Relación cíclica	165
Tabla 171 - PU-004: Añadir relación	165
Tabla 172 - PU-005: Eliminar relación	166
Tabla 173 - PU-006: Añadir elemento de la relación	166
Tabla 174 - PU-007: Modificar elemento de la relación	167
Tabla 175 - PU-008: Eliminar elemento de la relación	167
Tabla 176 - PU-009: Obtener sinónimos del tesouro	168
Tabla 177 - PU-010: Término es sinónimo	168
Tabla 178 - PU-011: Añadir sinónimo	169
Tabla 179 - PU-012: Eliminar sinónimo	169
Tabla 180 - PU-013: Obtener relación de sinonimia de un término	170
Tabla 181 - PU-014: Añadir un término	170
Tabla 182 - PU-015: Eliminar un término	171
Tabla 183 - PU-016: Modificar un término	172
Tabla 184 - PU-017: Mezclar un término con otro	173
Tabla 185 - PU-018: Drag and Drop de un término	174
Tabla 186 - PU-019: Mover un término a la raíz	174
Tabla 187 - PU-020: Intercambiar sinónimo por término	175
Tabla 188 - PU-021: Existe el término	175
Tabla 189 - PU-022: Obtener término	176
Tabla 190 - PU-023: Obtener todos los términos	176
Tabla 191 - PU-024: Obtener miga de pan	177
Tabla 192 - PU-025: Obtener términos raíces	177

Tabla 193 - PU-026: Añadir término raíz	178
Tabla 194 - PU-027: Eliminar término raíz.....	178
Tabla 195 - PU-028: Término es raíz	179
Tabla 196 - PU-029: Obtener hijos.....	179
Tabla 197 - PU-030: Obtener padres	180
Tabla 198 - PU-031: Importación	180
Tabla 199 - PU-032: Exportación	181
Tabla 200 - PU-033: Añadir buscador	181
Tabla 201 - PU-034: Eliminar buscador.....	182
Tabla 202 - PU-035: Cargar información Tooltips	182
Tabla 203 - PU-036: Guardar configuración Tooltips	183
Tabla 204 - Estimación de horas inicial	185
Tabla 205 - Reparto de horas por mes.....	186
Tabla 206 - Planificación de Septiembre	186
Tabla 207 - Planificación de Octubre.....	187
Tabla 208 - Planificación de Noviembre	187
Tabla 209 - Planificación de Diciembre	188
Tabla 210 - Salario bruto por rol.....	189
Tabla 211 - Coste de personal	190
Tabla 212 - Costes de material, viajes, dietas y fungible	190
Tabla 213 - Resumen de los costes y precio final	191

Índice de Ilustraciones

Ilustración 1 - Gestión de un tesouro con la herramienta DOME.....	28
Ilustración 2 - Gestión de un tesouro con la herramienta SWOOP.....	29
Ilustración 3 - Interfaz web para la gestión de un tesouro con TemaTres	29
Ilustración 4 - Jerarquía de clases en Protégé	30
Ilustración 5 - Modelo en cascada en ingeniería del software	37
Ilustración 6 - Dedicación de esfuerzos en metodologías ágiles	38
Ilustración 7 - Casos de uso Términos	74
Ilustración 8 - Casos de uso Relaciones	79
Ilustración 9 - Casos de uso Sinónimos	85
Ilustración 10 - Casos de uso Otras funcionalidades	89
Ilustración 11- Diagrama de secuencia Añadir un nuevo término raíz	93

Ilustración 12 - Diagrama de secuencia Intercambiar un sinónimo por un término.....	93
Ilustración 13 - Diagrama de secuencia Búsqueda de un término.....	94
Ilustración 14 - Diagrama de secuencia Drag & Drop de términos	94
Ilustración 15 - Subsistemas de diseño.....	95
Ilustración 16 - Modelo de datos	108
Ilustración 17 - Solicitud de confirmación	109
Ilustración 18 - Mensaje de error	109
Ilustración 19 - Selector de etiqueta sintáctica	110
Ilustración 20 - Selector de semántica	111
Ilustración 21 - Modelo arquitectónico en tres capas	129
Ilustración 22 - Subsistemas de diseño.....	131
Ilustración 23 - Diseño de componentes del sistema	131
Ilustración 24 - Diseño de la interfaz de usuario.....	132
Ilustración 25 - Diagrama de clases	133
Ilustración 26 - Diseño físico del modelo de datos	159
Ilustración 27 - Conexión con el servidor de SQL Server	160
Ilustración 28 - Restaurar base de datos Paso 1.....	161
Ilustración 29 - Restaurar base de datos Paso 2.....	161
Ilustración 30 - Restauración de base de datos completada	162
Ilustración 31 – Interfaz NUnit después de pasar las pruebas.....	163
Ilustración 32 - Diagrama de Gantt según planificación	188
Ilustración 33 - Comparación Planificado-Real.....	193
Ilustración 34 - Formulario principal	202
Ilustración 35 – Creación de un elemento del tesauro Paso 1.....	202
Ilustración 36 - Creación de un elemento del tesauro Paso 2	203
Ilustración 37 - Creación de un elemento del tesauro Paso 2	203
Ilustración 38 - Creación de un elemento del tesauro Paso 3	203
Ilustración 39 - Modificación de un término Paso 2	204
Ilustración 40 - Modificación de un término Paso 3.....	204
Ilustración 41 - Búsqueda de un término del tesauro Paso 1	205
Ilustración 42 - Búsqueda de un término del tesauro Paso 2	205
Ilustración 43 - Copiar y mover términos Paso 2.....	206
Ilustración 44 - Copiar y mover términos Paso 3.....	206
Ilustración 45 - Eliminación de una relación Paso 1	207
Ilustración 46 - Eliminación de un sinónimo Paso 1	207

Ilustración 47 - Borrado de un sinónimo o relación Paso 2	208
Ilustración 48 - Borrado de un término Paso 2	208
Ilustración 49 – Borrado de un elemento del tesouro Paso 3	208
Ilustración 50 - Exportación del tesouro Paso 1	209
Ilustración 51 - Exportación del tesouro Paso 2	209
Ilustración 52 - Importación del tesouro Paso 1	210
Ilustración 53 - Importación del tesouro Paso 2	210
Ilustración 54 - Importación de un tesouro Paso 3	211
Ilustración 55 - Importación de un tesouro Paso 4	211

Capítulo 1. Introducción

Este capítulo introductorio pretende ser el punto de entrada al resto del documento, ofreciendo una visión general sobre el tema que se va a realizar y pudiendo exponer los objetivos y motivaciones que han llevado a realizar el presente proyecto además de una breve descripción del resto de contenidos.

1.1. Visión General

La ingeniería del conocimiento es una disciplina moderna que forma parte de la inteligencia artificial (IA) y que se apoya en las tecnologías de la información. Su objetivo es extraer y representar el conocimiento de un experto en un determinado dominio para que sea procesado por un sistema.

La ingeniería del conocimiento, y en particular el procesamiento de ontologías, son uno de los problemas que más llaman la atención en la informática en la actualidad. La razón principal es la emergencia de un nuevo conjunto de aplicaciones donde las ontologías juegan un papel fundamental, la mayoría de las cuales se encuentran en dos campos:

- La necesidad de integración de aplicaciones empresariales cada vez más distribuidas.
- La última generación de sistemas para la Web (Web 2.0, Web Semántica).

1.2. Objetivos

El objetivo de este trabajo fin de grado es el de generar un sistema que permita gestionar un KOS *Knowledge Organization System* (KOS) de manera gráfica y sencilla mediante un tesaurus, importarlo y exportarlo a diversos formatos y obtener datos estadísticos del tesaurus.

Este proyecto es un módulo que permitirá a la herramienta [knowledgeMANAGER](#) la posibilidad de integrar un tesaurus que facilite la creación de un vocabulario controlado utilizado por un dominio concreto de cualquier rama del conocimiento con el objetivo de facilitar la comprensión de dicho conocimiento.

La integración de este módulo dentro de una aplicación mayor implica el conocimiento de la arquitectura, componentes y metodologías que se utiliza en dicha herramienta. Todo este proceso de aprendizaje se expondrá en los apartados de análisis y diseño del documento.

1.3. Contenido de la memoria

Este apartado está destinado a ofrecer una breve explicación de lo que se puede encontrar en cada uno de los capítulos y anexos que componen este documento.

En el primer capítulo, "Introducción", se tratará de dar una visión global del proyecto, que objetivos se han establecido y cuál será el contenido del resto del documento.

En el segundo capítulo, "Estado de la cuestión", se pasa a hablar sobre el ámbito actual del tema a desarrollar en este proyecto. Se tratarán temas como qué son las ontologías, los tipos de sistemas de gestión del conocimiento, las características principales de los tesaurus y muchas más cosas referentes al procesamiento del conocimiento, dando una visión y conocimiento necesarios para el desarrollo del proyecto.

Durante el tercer capítulo, “Análisis”, se extraerán los requisitos de usuarios y software necesarios para cumplir los objetivos establecidos y se analizarán posibles soluciones a los problemas que se pueden encontrar.

En el cuarto capítulo, “Diseño”, se sacará a partir de los requisitos una arquitectura del sistema y se diseñará un prototipo de la herramienta. Además se desarrollarán todas las pruebas necesarias para cumplir con los objetivos.

Localizado en el quinto capítulo, “Planificación y Presupuesto”, se encuentra el presupuesto general de los costes del proyecto en el que se muestran las distintas tareas realizadas y el tiempo invertido en cada una de ellas. Por otro lado, en este apartado es donde se ubica la planificación del proyecto, lo que permite seguir de un modo gráfico el tiempo invertido en cada una de las tareas.

En el capítulo sexto, “Conclusiones y trabajos futuros”, se dan a conocer las conclusiones finales alcanzadas así como los resultados obtenidos. También se exponen las posibles líneas de trabajo que se desprenden del desarrollo de esta aplicación y su utilidad.

Por último, la parte final del documento contiene varios anexos que complementan el resto de esta memoria añadiendo información que mejore la comprensión del conjunto. Entre estos anexos se puede encontrar un glosario de términos y acrónimos, referencias, bibliografía y documentación utilizada además de un breve manual de usuario referente a la aplicación.

Capítulo 2. Estado de la cuestión

En este capítulo se explican todas las líneas de investigación realizadas para el buen entendimiento y desarrollo del proyecto. Los apartados a tratar son: la ingeniería del conocimiento en lo referente a las ontologías (historia, tipos, gestión de ontologías, web semántica...), la herramienta dónde se incorporara el modulo a desarrollar en el que también se comparan diferentes herramientas de gestión de ontologías y las tecnologías utilizadas para su desarrollo.

2.1. Ontologías

En este apartado, se explica el tema de la ingeniería del conocimiento y más concretamente en las ontologías que son parte fundamental de este proyecto. Se comenta la historia y surgimiento de las ontologías, lenguajes y gestión de las ontologías, donde se trata el tema principal del proyecto que es la gestión de un KOS.

2.1.1. Historia

El concepto de ontología es un término muy antiguo utilizado desde la época de Aristóteles.

En filosofía, una ontología es una teoría sobre la naturaleza de la existencia, de qué tipo de cosas existen. Parte de la metafísica que trata del ser en general y de sus propiedades trascendentales [1].

En el siglo XVII el término ontología se usa como sinónimo de metafísica, concretamente como la rama de la metafísica que trata con la naturaleza del ser.

A finales de la década de los años 80, el término ontología es adoptado por la ingeniería del conocimiento, más concretamente por el campo de la inteligencia artificial con el significado de compartir y reutilizar conocimiento.

En la segunda mitad de los años 90, se incorpora este término a la ingeniería de la web para más adelante ser un concepto clave en el surgimiento de la web semántica donde las ontologías serán usadas como artefactos para la representación del conocimiento.

Por lo tanto, en informática, las ontologías surgieron por la necesidad de representar el conocimiento de la Web Semántica; de una forma legible para los ordenadores y que sea consensuado y reutilizable.

2.1.2. La Web Semántica

Como se ha comentado en la sección anterior, las ontologías surgen de la necesidad de tener una web semántica que hasta ahora no existía, es decir, una web dotada de mayor significado que defina mejor la información que en ella se encuentra; con el fin de que cualquier usuario de Internet pueda encontrar respuestas a sus preguntas de una forma más rápida, sencilla y con unos mejores resultados que antes. [17]

La base de la web semántica es utilizar una infraestructura común mediante la cual es posible compartir, transferir y procesar información. Para crear dicha estructura se tiene el apoyo de lenguajes universales que resuelven los problemas ocasionados por una web carente de significado.

En la nueva web, los datos se deberán definir convenientemente según reglas y operaciones también bien definidas que puedan dar la habilidad a una máquina de procesar adecuadamente las palabras de los usuarios.

Es en este punto, en la obtención adecuada de la definición de los datos, donde la web semántica. Los más conocidos y utilizados en la actualidad son: RDF y OWL. Se puede decir que estos son los mecanismos necesarios para construir esa infraestructura común y global.

- RDF es un lenguaje con el objetivo de proporcionar información descriptiva conceptual sobre los recursos que se encuentran en la Web. Es como un [metadato](#) del contenido de internet. SPARQL es el lenguaje de consulta sobre RDF, que permite hacer búsquedas sobre los recursos de la Web Semántica utilizando distintas fuentes datos.
- OWL es un lenguaje de marcado para desarrollar temas o vocabularios específicos en los que asociar esos recursos proporcionados por RDF. Lo que hace OWL es proporcionar un lenguaje que define ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir, en un campo determinado como puede ser el de la banca, ocio, deporte, etc. Las ontologías se encargan de definir los términos utilizados para describir y representar un área de conocimiento, además de incluir definiciones de conceptos básicos y la relación entre ellos.

Para ayudar a estos lenguajes se utiliza también el lenguaje de XML que permite definir la gramática de lenguajes específicos. [2]

2.1.3. Componentes y tipos

Se define una **ontología** como una “especificación formal y explícita de una conceptualización compartida”. De una manera más informal, se puede decir que una ontología es un modelo o esquema, utilizado por la gestión de KOS, de conocimiento consensuado en un determinado dominio y que es reutilizable en diferentes aplicaciones.

Los **componentes** utilizados normalmente para modelar o representar las ontologías son:

- **Conceptos:** Tienen un sentido muy amplio. Pueden ser cualquier cosa de la cual se dice algo, incluyendo también descripciones.
- **Relaciones:** Representan un tipo de asociación entre conceptos del dominio. Generalmente, las ontologías contienen relaciones binarias.
- **Funciones:** Son un caso especial de relaciones, donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- **Axiomas:** Modelan oraciones o sentencias que siempre son verdaderas.
- **Instancias:** Representan valores concretos de los conceptos.

En cuanto a los **tipos de ontologías** que existen, se pueden establecer atendiendo a diversos aspectos:

Según el ámbito de conocimiento:

- **Ontologías generales:** son las ontologías de nivel más alto ya que describen conceptos generales (tiempo, espacio, materia, entre otras)
- **Ontologías de dominio:** describen el vocabulario de un dominio concreto del conocimiento.
- **Ontologías específicas:** son ontologías especializadas que describen los conceptos para un campo o sector limitado del conocimiento o una aplicación determinada.

Según el tipo de agente:

- **Ontologías lingüísticas:** se asocian a aspectos lingüísticos, es decir, a aspectos gramáticos, semánticos y sintácticos destinados a ser utilizados por los seres humanos.
- **Ontologías no lingüísticas:** destinadas a ser utilizadas por agentes inteligentes y robots.
- **Ontologías mixtas:** combinan características de las anteriores.

Según el grado o nivel de abstracción y razonamiento lógico:

- **Ontologías descriptivas:** incluyen descripciones, taxonomías de conceptos, relaciones entre conceptos y propiedades, pero no permiten inferencias lógicas.
- **Ontologías lógicas:** permiten inferencias lógicas mediante la utilización de una serie de componentes llamados axiomas, descritos con anterioridad.

2.1.4. Gestión de Ontologías

2.1.4.1 ¿Qué es un KOS?

Un **KOS**, *Knowledge Organization System* o *Sistema de Organización del Conocimiento*, se define como un sistema que engloba cualquier tipo de esquema utilizado para la organización del conocimiento tales como taxonomías, clasificadores, tesauros y ontologías entre otras.

Estos sistemas de gestión se caracterizan por integrar una serie de tecnologías que ayudan a crear, gestionar y visualizar, de una manera sencilla y simplificada, modelos que muestran conceptos de un determinado dominio y su estructura semántica subyacente.

El objetivo principal de estos sistemas es ayudar a la comprensión, gestión y recuperación de los conceptos contenidos en cada dominio. Dado que en cualquier campo o disciplina del conocimiento es fundamental obtener este objetivo; los sistemas de gestión de KOS se encuentran en diversos estudios desde la Pedagogía a la Inteligencia Artificial, pasando por la Documentación y la **Ingeniería del Software** entre otras.

En cada una de estas disciplinas, los sistemas de KOS desarrollan diferentes esquemas organizativos. Estos esquemas se diferencian por la semántica que introducen dependiendo de los objetivos buscados, pero siempre prevalece, como elemento común, un **vocabulario controlado** para designar los conceptos establecidos en el dominio, y las relaciones semánticas entre estos conceptos. [14]

En los últimos años disciplinas como la Ingeniería del Software, ha definido las etapas comunes que existen en la creación de estos modelos y se ha llegado a la conclusión de que para automatizar en el mayor grado de completitud estas fases se requiere un gran esfuerzo.

Una de las principales fases es la **adquisición del conocimiento**. En esta etapa, se identifican, mediante expertos en el dominio y diferentes fuentes documentales, los conceptos que lo integran y sus posibles equivalencias (sinónimos). Posteriormente, se estructuran de manera jerárquica los conceptos, es decir, se clasifican según estructuras de generalización-especificación (padre-hijo). Otra etapa que frecuentemente aparece, es determinar que otros tipos de relaciones y de restricciones deben existir ente los conceptos. [15][16]

2.1.4.2. Ventajas de los KOS

Según lo definido en el apartado anterior, algunas de las ventajas de los sistemas de KOS son:

- Mejorar y perfeccionar el flujo de información dentro de un dominio de una empresa, mejorando de esta manera la comunicación entre los integrantes de la misma.
- Administrar de manera más sencilla el conocimiento de la empresa con el fin de obtener un modelo de gestión que permita destacar sobre la competencia.
- Reutilizar estos sistemas para el uso en otros esquemas realizados anteriormente con el fin de reducir esfuerzos.

2.1.4.3. Limitaciones de los KOS

En cuanto a las limitaciones de los sistemas de KOS, se pueden citar:

- La construcción o creación de un sistema de KOS ya que se requiere de un alto grado de consenso entre los participantes y un elevado esfuerzo intelectual individual, lo cual se traduce en un proceso lento y laborioso.
- Todo lo referente a los empleados que utilicen la gestión del conocimiento. Si bien el flujo de información es una de las ventajas de un sistema de KOS, se debe asegurar el compromiso de los empleados, que son los poseedores del factor de producción, el conocimiento, y deben estar dispuestos y aprender a compartirlo.
- Creación de buenas herramientas de formación para lograr el propósito anterior.

2.2. La herramienta: KnowledgeMANAGER

En este apartado, se detalla principalmente lo que es la aplicación en la cual se incorporara nuestro módulo, el [knowledgeMANAGER](#). Se explicaran los diferentes módulos de los que consta pero sin entrar al detalle de todos ellos ya que sería desviarnos de nuestro tema principal. Este apartado quiere servir para dar una visión general de la herramienta donde funcionara mi módulo. Además se analizan otras herramientas de gestión de ontologías que existan en el mercado.

2.2.1. Componentes de la herramienta

La herramienta se divide en cinco módulos bien diferenciados:

El primero corresponde a la gestión del dominio, en la cual se encuentra todos los aspectos relacionados con éste. Estos aspectos se subdividen en módulos que los gestionan: Terminología, Tokenización, Normalización, Etiquetado, Ontología ligera y Reglas de inferencia o patrones.

El segundo se llama subdominios y en él se encuentran aspectos como la gestión de subdominios, grupos de reglas de inferencia y el tesouro del dominio que es otro modulo tratado en mi proyecto.

El tercero se corresponde con la indización y motor de búsqueda. La funcionalidad del primero es la de probar diferentes temas relacionados en la indexación y comprobar que los resultados obtenidos son los esperados. La funcionalidad del segundo aún no está ni en fase de análisis pero lo que se pretende básicamente es tener un buscador propio de la ontología.

El cuarto se llama documentos y en ella se gestionan los [asserts](#) de la ontología, es decir, los diferentes artefactos que pueden ser tratados en mi ontología.

El quinto y último se denomina configuración y en él se encuentran muchos aspectos de configuración de la herramienta, además de pruebas de integridad y de indización.

Como se puede ver es una herramienta muy compleja y con mucha funcionalidad de ahí que mi proyecto solo se base en ciertos sub módulos de los anteriores comentados, pero que a pesar de ello requieren de un gran conocimiento de la tecnología y de la forma de trabajo para su desarrollo. Estos aspectos de tecnologías y métodos se comentaran en un posterior apartado dentro de este mismo capítulo.

2.2.2. Gestores de ontologías en el Mercado. Comparativas

Hoy en día en el mercado existen multitud de gestores de ontologías. La mayoría de estos están basados en OWL, lenguaje que anteriormente se comentó en que se basaba, pero por resumir, básicamente es un lenguaje utilizado para compartir datos a través de ontologías en la web.

OWL principalmente está basado para aplicaciones en la web, pero nuestro módulo a desarrollar es una aplicación de escritorio, por lo tanto, se descarta la opción de utilizar este lenguaje para el desarrollo del proyecto.

A continuación se analizaran algunas de las aplicaciones encontradas para la gestión de ontologías. La primera conclusión a la que se llega es que la mayoría de las aplicaciones encontradas están enfocadas a la gestión de tesauros; que es justo el tema central del proyecto. Esto es bueno ya que se pueden sacar buenas conclusiones de que puntos a favor o en contra tienen respecto a mi aplicación y viceversa.

- **Dome:** [3] Proporciona un conjunto de aplicaciones que se basan en su simplicidad, completitud y reutilización. Contiene características interesantes pero está un poco descentralizada en cuanto a la necesidad de tener distintos módulos dentro de la herramienta para gobernar la gestión del tesouro. Además no parece que existan actualmente líneas de progreso en esta herramienta.

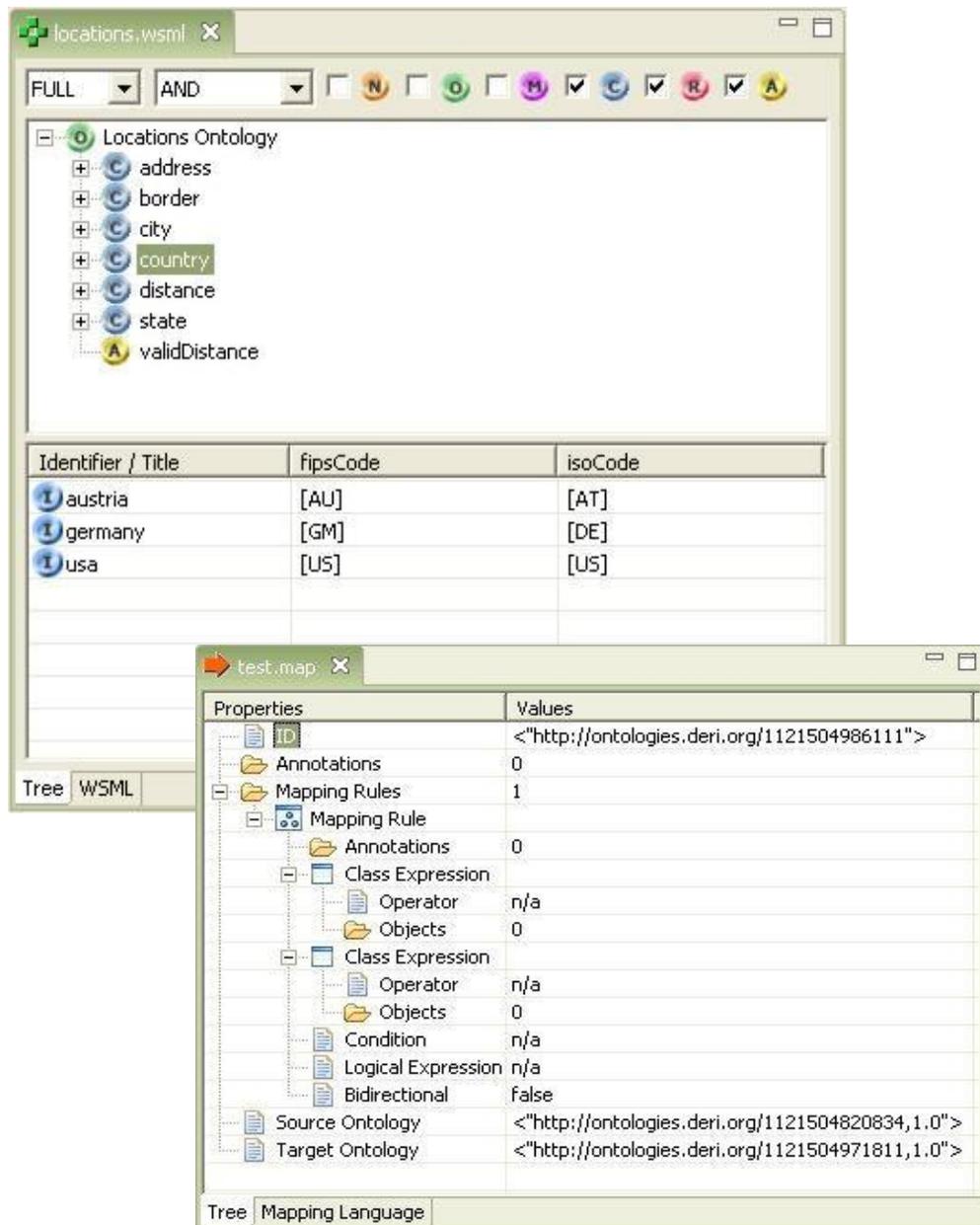


Ilustración 1 - Gestión de un tesouro con la herramienta DOME

Como se puede observar es necesario tener diferentes ventanas para gestionar el tesouro.

- **Swoop:** [4] Esta herramienta permite crear, editar y testear ontologías OWL. Su posterior uso sería en la web, por lo que perdería fuerza en nuestro objetivo. Fue construido en la Universidad de Maryland, pero actualmente es un proyecto open-source con una gran comunidad detrás.

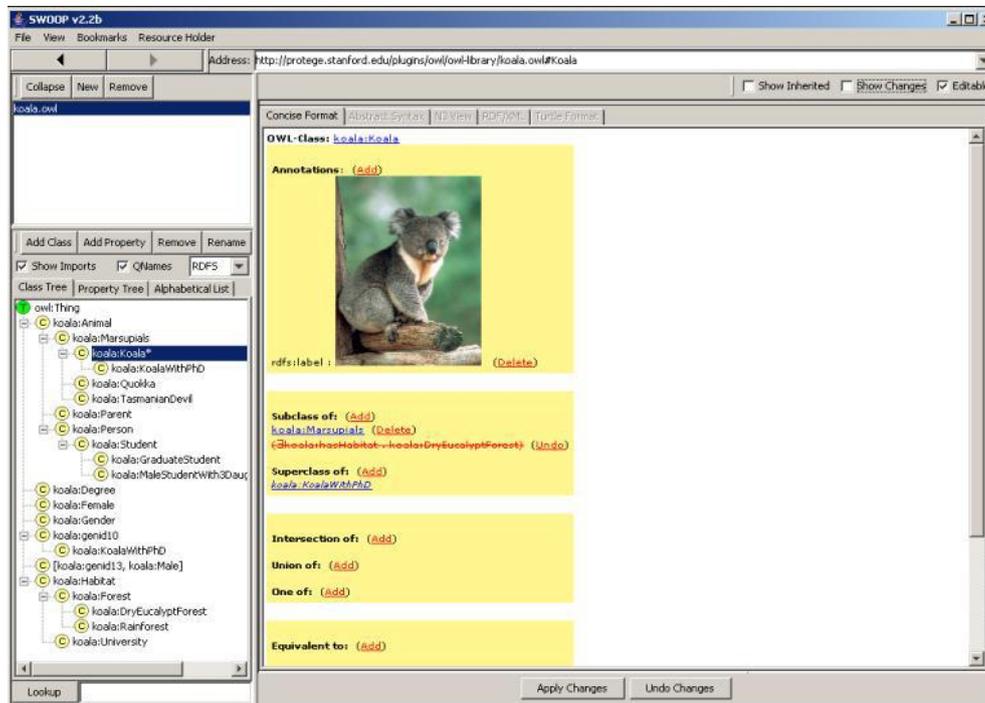


Ilustración 2 - Gestión de un tesouro con la herramienta SWOOP

- **TemaTres:** es un gestor de ontologías basado en web. Su principal uso es la gestión de lenguajes documentales, y está orientado especialmente al desarrollo de tesouros jerárquicos. Contiene más de un millón de términos y presenta un API para ser utilizado a través de webservices. Permite exportar en todos los formatos.

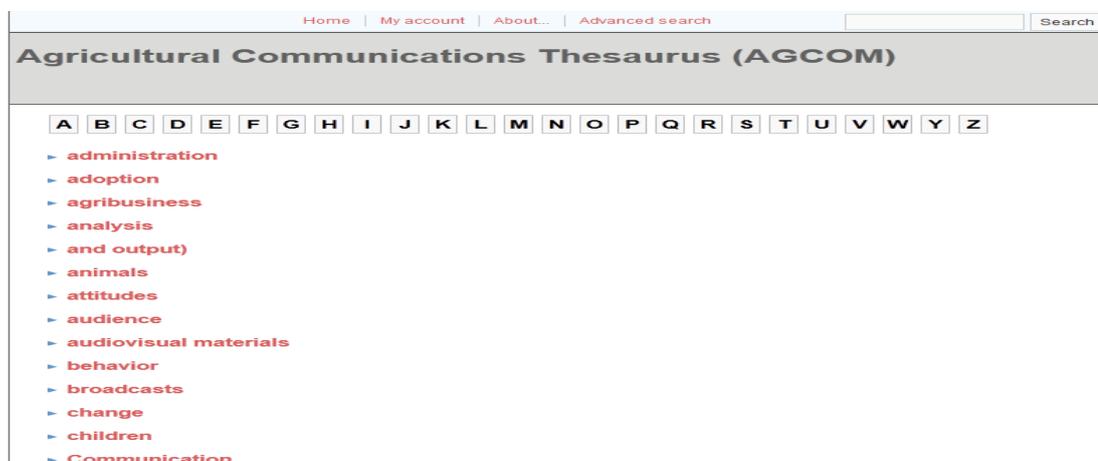


Ilustración 3 - Interfaz web para la gestión de un tesouro con TemaTres

Este gestor tiene muchas características incluidas en mi proyecto pero al estar basado en web pierde potencia con respecto a mi objetivo.

- **Protégé:** [5] es un gestor y editor de ontologías de código abierto. Está basado en lenguaje Java, y proporciona un entorno fácil de usar que lo hace muy flexible para construir prototipos rápidos y desarrollo de aplicaciones. Su plataforma soporta dos maneras distintas de modelar ontologías. Por un lado, con su propio frame denominado Protégé-Frame, y, por el otro lado, a través de editores OWL. Además pueden exportar sus ontologías en distintos formatos. Hoy en día está siendo utilizado en distintas áreas como defensa, biomedicina o modelado corporativo, debido a que tiene una gran comunidad de desarrolladores y académicos tras él.

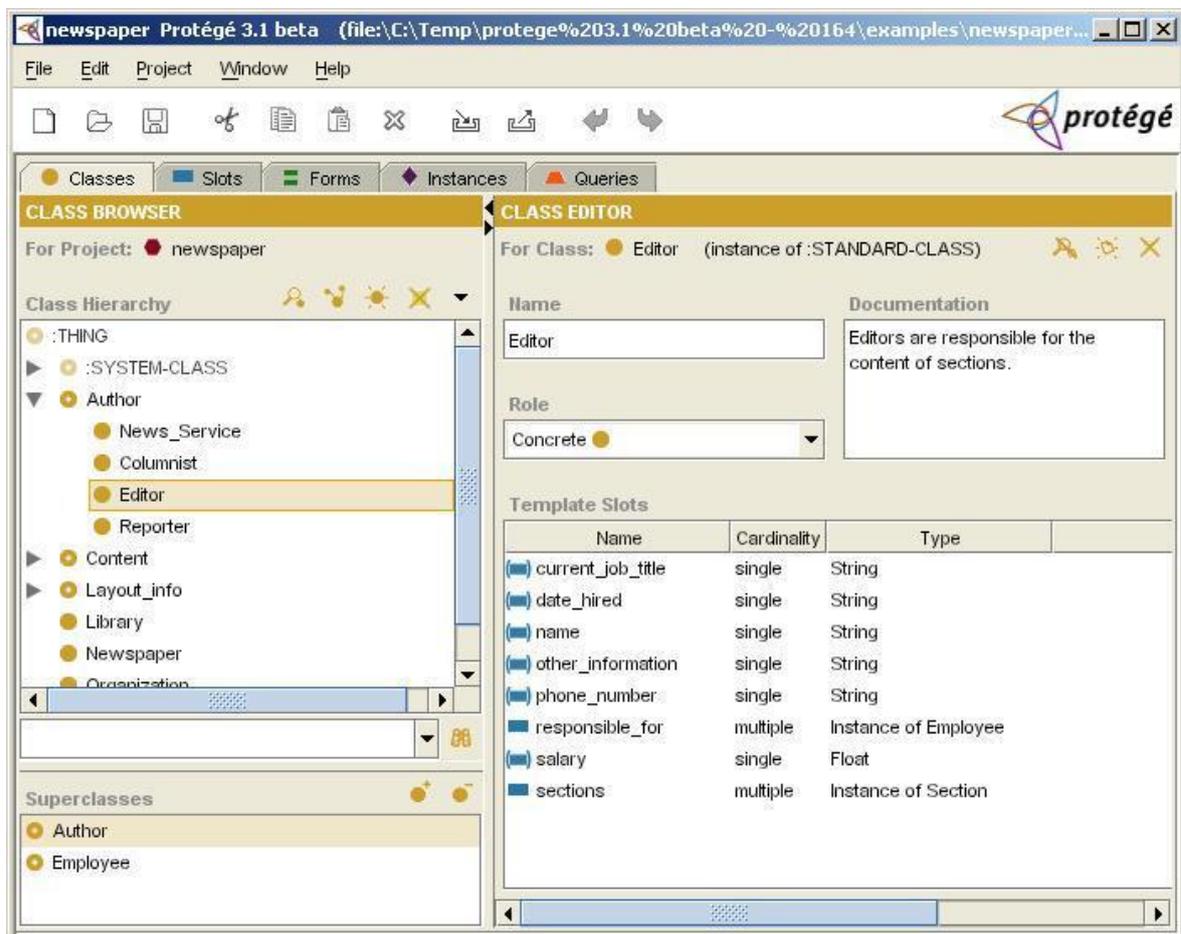


Ilustración 4 - Jerarquía de clases en Protégé

Es la herramienta con más potencial de las que he analizado. El problema es que no puede ser utilizada en mi proyecto ya que la necesidad de mi proyecto es gestionar una ontología a través de un tesauro utilizando el modelo de negocio actual, que es el CAKE, con una base de datos ya en uso. Sobre esto se hablará en mayor detalle en el capítulo de Análisis.

A continuación se realiza una comparativa de las aplicaciones analizadas, en función de la información que cada herramienta proporciona al público y lo que mi proyecto intenta solucionar.

	Dome	Swoop	TemaTres	Protégé
Aplicación de escritorio	SI	SI	NO	SI
Gestión de tesauros	SI	SI	SI	SI
Exportación a distintos formatos	NO	SI	SI	SI
Gestión de vocabulario controlado	SI	SI	SI	SI
Gestión de <u>light-ontology</u>	NO	NO	NO	SI

Tabla 2 - Comparativa de las herramientas de gestión analizadas

Como se puede observar en la tabla, todas las aplicaciones cumplen con la necesidad de tener un vocabulario controlado con el cual elaborar el tesauro y también la mayoría cumple con la exportación en distintos formatos.

La gestión de [light-ontology](#) esta solamente soportado por la herramienta protégé, de ahí que anteriormente dijera que podría haber utilizado su tecnología para mi proyecto pero la necesidad de ser compatible con el modelo actual de negocio, hace que se tenga que desarrollar el presente proyecto y que ninguna de las herramientas cumpla las expectativas que se desean.

2.3. Tecnologías necesarias para el desarrollo

En este apartado, se analizan las diferentes tecnologías utilizadas para el desarrollo del proyecto, haciendo una comparación de las diferentes opciones que se tenían y porque se ha elegido una y desechado otras.

En las elecciones de las diferentes tecnologías se utilizaran los siguientes criterios:

- **Licencia:** si es libre o propietaria.
- **Coste:** ligado con el criterio anterior, el cual mostrara el coste de la tecnología.
- **Rendimiento:** medida en la cual se cree que se sacaran mejores resultados.
- **Experiencia:** mide el tiempo trabajado con la tecnología.

Cada uno de estos criterios tiene tres posibles valores: alto (3), medio (2) y bajo (1).

2.3.1. Sistemas operativos

El sistema operativo hace de capa intermedia entre las aplicaciones de desarrollo y el desarrollador. El objetivo de la elección de un sistema operativo es la de optimizar los recursos para lograr un desarrollo eficiente. Los sistemas operativos posibles son:

- **Mac OS Lion 10.7:** última versión del sistema operativo creado por Apple.
- **GNU/Linux RedHat:** distribución Linux orientada al mundo empresarial.
- **Microsoft Windows 7 Professional:** última versión del sistema operativo creado por Microsoft. Actualmente en versión beta Windows 8.

A continuación, se muestra la tabla de valoraciones de los sistemas operativos:

Sistema operativo	Licencia	Coste	Rendimiento	Experiencia
Mac OS	Propietaria	35 €	Alto	Baja
GNU/ Linux	Libre	0 €	Medio	Baja
Windows 7 Professional	Propietaria	229 €	Alto	Alta

Tabla 3 - Valoración de los posibles sistemas operativos

Como se puede observar en la tabla, en cuestiones de coste el sistema operativo de Linux es el más favorable pero la experiencia en este entorno es muy baja por lo que es un principal inconveniente. Otro punto a destacar es el rendimiento alto de los otros dos sistemas operativos que unido con la experiencia que tengo en ellos se puede averiguar que el sistema operativo que se va a utilizar en el desarrollo del proyecto es Windows 7.

El inconveniente que puede tener a priori el sistema operativo que he escogido es el precio pero gracias al convenio existente entre Microsoft y la Universidad Carlos III de Madrid, en el cual, los alumnos de la universidad pueden descargarse gratuitamente y con licencia diferentes aplicaciones y sistemas operativos de Microsoft. Este programa se denomina [MSDN Academic Alliance de Microsoft](#).

Por lo tanto, el coste de la elección sería de 0 €.

2.3.2. Lenguaje de programación

Los posibles lenguajes de programación para el desarrollo del proyecto son:

- **Java**
- **C**
- **C++**
- **.NET Framework**

Estos son los diferentes lenguajes que se han usado a lo largo de la carrera y además son los más utilizados hoy en día [6].

A continuación, se muestra la tabla de valoraciones de los lenguajes de programación:

Lenguajes de programación	Licencia	Coste	Rendimiento	Experiencia
Java	Libre	0 €	Alto	Media
C	Libre	0 €	Medio	Media
C++	Libre	0 €	Medio	Baja
.Net Framework	Propietaria	0 €	Alto	Alta

Tabla 4 - Valoración de los posibles lenguajes de programación

Como se puede observar en este caso el coste de todos estos productos es cero, por lo tanto la elección se llevara a cabo por las columnas de experiencia y el rendimiento que yo mismo pueda sacar a dichos lenguajes para realizar mi proyecto.

El lenguaje C++ es descartado por la falta de experiencia en este lenguaje a la hora de crear aplicaciones. Elegirlo supondría un riesgo muy grande en el proyecto.

El lenguaje C es descartado ya que se piensa que no tiene potencial suficiente para resolver los problemas del sistema de información que el cliente desea que se construya. Como se puede ver en la tabla en el rendimiento obtiene una puntuación de medio.

Entre los dos lenguajes restantes, cabe mencionar que en .NET se tienen conocimientos más elevados de programación que en java. Por este simple hecho creo que para minimizar riesgos futuros en la programación de la aplicación la elección más conveniente es .NET.

2.3.3. Entorno de programación

El entorno de programación no es más que una plataforma o [framework](#) que ayuda a programar en un determinado lenguaje. Los posibles entornos a utilizar son:

- **Microsoft Visual Studio 2010 Professional:** entorno de desarrollo de Microsoft.
- **Eclipse:** entorno de desarrollo [open-source](#).
- **Netbeans:** entorno de desarrollo [open-source](#). Creado por [Sun Microsystems](#) y ya incorpora Java.

A continuación, se muestra la tabla de valoraciones de los entornos de programación:

Entornos de programación	Licencia	Coste	Rendimiento	Experiencia
Microsoft Visual Studio 2010	Propietaria	415 €	Alto	Alta
Eclipse	Libre	0 €	Alto	Media
Netbeans	Libre	0 €	Alto	Baja

Tabla 5 - Valoración de los posibles entornos de programación

Como se puede observar en la tabla el único que tiene coste es el Visual Studio por lo que a priori se podría descartar, pero también es el que más experiencia se tiene y los riesgos serían menores que con los otros dos. Además gracias al programa comentado anteriormente el [MSDN Academic Alliance de Microsoft](#), el precio del software es 0. Por lo tanto a igualdad de condiciones en el precio la elección es Visual Studio ya que se conseguirá mayores resultados.

2.3.4. Sistema gestor de base de datos

Los gestores de base de datos constituyen la capa de control, administración y gestión de la base de datos que es accedida desde la aplicación. Los diferentes gestores propuestos son:

- **MySQL:** es de software libre y permite la gestión de bases de datos relacionales.
- **Access:** sistema de Microsoft que permite una simple gestión de bases de datos relacionales.
- **Microsoft SQL Server 2008:** sistema de Microsoft que permite la gestión completa y exhaustiva de bases de datos relacionales.

A continuación, se muestra la tabla de valoraciones de los gestores de base de datos:

Gestores de base de datos	Licencia	Coste	Rendimiento	Experiencia
MySQL	Libre	0 €	Alto	Media
Access	Propietaria	0 €*	Bajo	Baja
Microsoft SQL Server	Propietaria	525 €	Alto	Alta

Tabla 6 - Valoración de los posibles gestores de base de datos

* El coste es el derivado del paquete de office de Microsoft.

Para los sistemas gestores de base de datos ocurre algo parecido al anterior punto, el de los entornos de programación, la mayor experiencia y rendimiento se la puedo sacar a Microsoft SQL Server 2008 y gracias a la alianza entre Microsoft y la universidad también comentada en apartados anteriores el coste del producto es 0. Por lo tanto la herramienta gestora de base de datos a utilizar va a ser esa.

2.3.5. Gestión del proyecto

En este punto se analizará las herramientas utilizadas para la gestión de proyecto, es decir, herramientas de documentación, de gestión de incidencias o relativas a los tiempos de trabajo o diagramas de clases.

Documentación

En cuanto a la documentación, se tienen muchas opciones pero la más potente en mi humilde opinión es la suite de Office de Microsoft. Otras herramientas consideradas han sido OpenOffice de código abierto o iWork de Apple pero el uso continuado a lo largo de mis estudios con la herramienta de Microsoft me hace más útil y rápido utilizar dicha herramienta para la documentación. Las herramientas a utilizar de la suite son:

- **Microsoft Word:** como procesador de textos.
- **Microsoft Excel:** como plantilla para cálculos numéricos.
- **Microsoft PowerPoint:** como gestor de contenido multimedia. Ejemplo: diapositivas o algunas de las figuras utilizadas a lo largo de esta memoria.

Planificación de Tareas y Tiempos

En cuanto al registro de tiempos y programación de las tareas se va a utilizar la herramienta Gantt-project, que es de código libre y es muy fácil de usar. Se tuvo la opción de utilizar Microsoft Project, pero es una herramienta muy potente y con demasiada funcionalidad para lo que realmente iba a usar, por lo tanto descarte el uso de dicha herramienta.

Registro de incidencias

Cabe mencionar el uso de una herramienta muy potente en la gestión de proyectos. Esta herramienta es JIRA. En el departamento en el cual he estado desarrollando mi proyecto lo utilizan para incluir requisitos, mejoras, bugs, registro de tiempos, etc que se hayan encontrado en la aplicación. Con esta herramienta tienes un control de lo que llevas hecho y lo que te queda. Además se pueden añadir versiones para distintas fechas con lo cual puedes marcar hitos de proyecto. En resumen, es una herramienta muy completa y útil para la gestión de proyectos.

En este proyecto, simplemente se ha usado para registrar posibles fallos que iba encontrando en la aplicación o mejoras que podría incluir. ¿Por qué usar esta herramienta sólo para el registro de incidencias? La respuesta es la falta de experiencia que tengo en el uso de dicha herramienta. Por ello uso una herramienta para la planificación y otra para el registro de incidencias. Es para mí una experiencia nueva y satisfactoria el poder integrar a mi conocimiento una herramienta profesional como es el JIRA para la gestión de proyectos.

Diagramas UML

Para la gestión de los diagramas de UML utilizados en el desarrollo del proyecto se ha utilizado StartUML que es una herramienta de [open-source](#) muy potente y fácil de usar. Todos los diagramas UML de esta memoria fueron diseñados con este programa, como por ejemplo, la arquitectura del sistema, diagramas de clases o los casos de uso, entre otros.

Se pensó en utilizar el programa Visual Paradigm pero es una aplicación que viene con periodo de prueba muy corto y se descartó la opción de utilizarlo. Además el potencial de ambas herramientas es muy similar por lo que no se perderá calidad en la generación de los diagramas.

2.3.6. Gestión de versiones

La gestión de versiones es especialmente útil para proyectos en los que trabaja mucha gente al mismo tiempo y sobre los mismos documentos o ficheros. Aunque el presente proyecto, siendo un TFG de la universidad, ha sido desarrollado por una sola persona, se ha decidido utilizar un gestor de control de versiones ya que de esta forma es mucho más fácil el poder desarrollar el código en varias máquinas sin necesidad de transportar ficheros en unidades externas de almacenamiento.

Para el desarrollo del proyecto se ha utilizado la herramienta de control de versiones Subversión ya que es la herramienta que utilizan en mi departamento y me han enseñado a utilizarlo. Se ha creado un repositorio propio en la nube, en el cual se incluye el código y la documentación del proyecto, localizado en una maquina servidora encendida todos los días las 24 horas del día. Esta siempre accesible para su uso excepto en caso de caída del servidor. En estos casos se tienen copias de seguridad locales para no perder días de trabajo.

Además los sistemas basados en Subversión, se caracterizan por versionar las diferentes subidas que se hacen durante el ciclo de vida. De esta manera, se puede recuperar fácilmente una versión estable anterior en caso de haber detectado fallos en la funcionalidad por cualquier motivo.

Otro motivo de porque usar subversión son la existencia de componentes que permiten integrar al entorno de desarrollo el sistema de control de versiones. En mi caso he utilizado el programa [Ankh SVN](#), que se integra en Visual Studio para controlar las versiones directamente desde el.

2.3.7. Gestión de pruebas

Cualquier sistema de información, tiene como una parte fundamental el desarrollo de pruebas para dar fiabilidad al sistema. Un sistema que carezca de una batería de pruebas adecuada que al menos pruebe que sus distintas funcionalidades funcionan correctamente, no puede ser un sistema fiable que se ponga en producción y pueda ser utilizado por el cliente. Aunque es prácticamente imposible certificar que un

sistema funciona al 100% a pesar de tener una batería de pruebas robusta, la fase de pruebas la considero fundamental dentro de un proyecto.

Para las pruebas unitarias de mi proyecto, se va a utilizar el framework [NUnit](#), que presenta una interfaz de usuario intuitiva y muy sencilla para ejecutar las pruebas. Además es un programa que he utilizado más veces a lo largo de la carrera y una vez más es posible integrarlo en el Visual Studio lo que acelera el proceso de desarrollo de dichas pruebas.

2.3.8. Metodologías de desarrollo

En cuanto a metodología de desarrollo, se puede pensar que la elección más lógica es un proceso normal en cascada en el cual vamos planteando y resolviendo los diferentes fases una detrás de la otra, incluso teniendo la posibilidad de volver a una fase anterior para modificar algo como ocurre en el modelo en cascada modificado que se muestra en la siguiente imagen:

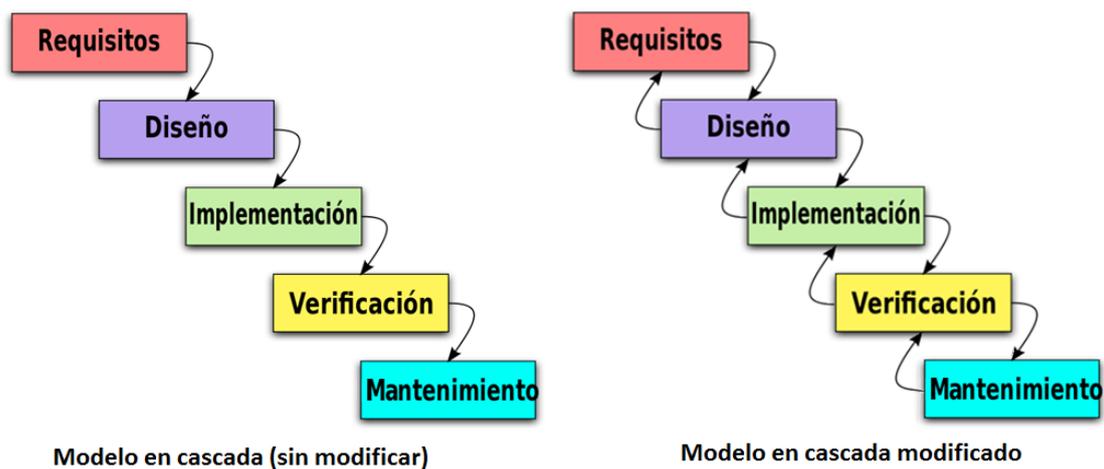


Ilustración 5 - Modelo en cascada en ingeniería del software

Pero este modelo en mi opinión es muy restringido igualmente ya que a pesar de dejarte volver a una etapa anterior, las fases deben estar bien cerradas para seguir con la siguiente fase.

En mi opinión, el software en todo su ciclo de vida es cambiante por naturaleza y los ingenieros no deben de negarlo sin afrontarlo y trabajar en consecuencia. Por ello se han creado las metodologías ágiles de desarrollo, las cuales tienen como premisa que el software puede cambiar de un día para otro y apuestan por trabajar en paralelo en las fases de codificación y pruebas con el fin de crear procesos iterativos en los que una necesidad de codificación pueda suponer un cambio en el diseño de un componente sin ningún problema. Con estas metodologías también se puede llegar a introducir más requisitos no vistos anteriormente y no supondrían un quebradero de cabeza el poder abordarlos.

Por todo ello, para el desarrollo de mi proyecto voy a utilizar metodologías ágiles. A continuación, se muestra un ejemplo de dedicación de esfuerzos en las distintas fases de proyecto utilizando una metodología ágil:



Ilustración 6 - Dedicación de esfuerzos en metodologías ágiles

Como se puede observar en la figura, en el proceso de iniciación y elaboración las fases de requisitos, análisis y diseño y modelo de negocio son las más predominantes pero también se observa que se comienza a implementar y hacer pruebas desde el principio, lo que ayuda a saber desde un principio las alternativas de diseño y mejoras en la arquitectura que se pueden llevar a cabo.

En la fase de construcción también se ve que se pueden añadir requisitos y por lo tanto modificar el análisis y el diseño establecido, aunque lo más predominante son la implementación y pruebas ya que en esta fase se tienen ya que gracias a las fases anteriores de implementación, se tienen muy claros los componentes de los cuales va a constar el sistema a desarrollar. Para finalizar en la fase de transición se termina de implementar y se hace un proceso más exhaustivo de pruebas y se procede al despliegue, es decir, poner el software en producción.

Hay muchas metodologías ágiles para seguir, como por ejemplo:

- XP – Extreme Programming.
- [Scrum](#).
- Crystal Clear.

Para este proyecto no voy a seguir al pie de la letra ninguna de ellas sino que lo que voy a realizar es una dedicación de esfuerzo parecida a la de la ilustración 6, ya que repito como ya he dicho en este mismo apartado creo que el software es cambiante y las diferentes fases de desarrollo deben hacerse como se establece en una metodología ágil siempre que el proyecto lo permita. Hay casos extremos de proyectos con mucha envergadura como pueden ser la creación de aviones que quizás las metodologías ágiles no sean aun un futuro cercano en su desarrollo.

2.3.9. Resumen

En este apartado se pretende resumir en pocas palabras todas las tecnologías que se van a utilizar y usar en este proyecto. Para un mejor entendimiento se resume todo en la siguiente tabla:

Sistema operativo	Windows 7 Professional	Gestión de proyecto	Microsoft Office Gantt-Proyect JIRA StartUML
Lenguaje de programación	.Net Framework	Gestión de versiones	Subversión
Entorno de programación	Visual Studio 2010	Gestión de pruebas	NUnit
Gestión de base de datos	Microsoft SQL Server	Metodología de desarrollo	Ágil

Tabla 7 - Resumen de las tecnologías para el desarrollo del proyecto

Capítulo 3. Análisis

Este capítulo tiene como objetivo la obtención de una especificación a alto nivel del sistema que se quiere desarrollar, es decir, recoger las necesidades que se deben resolver y modelar. Aclarar que no se explica cómo solucionar el problema planteado por el cliente.

3.1. Definición del sistema

3.1.1. Determinación del alcance del sistema

El presente proyecto es una pequeña parte de un proyecto más grande llamado [knowledgeMANAGER](#), que da nombre a la herramienta. El objetivo que se busca es ofrecer una herramienta que permita a un usuario final la gestión y control integral de la organización del conocimiento de una empresa a través de la creación y mantenimiento de dominios concretos (*KOS, Knowledge Organization System*).

Más concretamente, el objetivo del proyecto que se abarca con este documento es añadir a la herramienta mencionada con anterioridad, un módulo que sea capaz de crear y mantener un tesouro de términos y relaciones y otro pequeño modulo que se encargue de la gestión de la normalización dentro de la herramienta de gestión del conocimiento.

Mediante la creación de tesauros, se puede mantener de una forma clara y concisa una lista de palabras o términos que se emplean para representar conceptos dentro de un determinado dominio; así como la capacidad sencilla de crear relaciones sobre esos conceptos. Estas relaciones pueden ser de diversa índole dependiendo el tipo de semántica que se le quiera dar a los términos, yendo desde las relaciones típicas de padre-hijo, hasta relaciones de sinonimia y asociación e incluso poder soportar relaciones poli jerárquicas.

Por lo tanto el objetivo es crear un módulo que permita la creación de un tesouro de términos y relaciones y el mantenimiento y modificación de este; además de la inclusión de un apartado de la gestión del conocimiento que es la normalización de los términos del vocabulario. Durante este proceso, se deberá permitir la reutilización de aspectos ya utilizados en otra ontología anteriormente desarrollada llamada swReuser (CAKE Studio) o la mejora de muchos aspectos de esta.

Este módulo debe servir al experto del dominio para gestionar y mantener las relaciones entre los términos de un vocabulario controlado. Además de poder utilizar aspectos de la normalización de términos utilizados en la indización de palabras documentos o cualquier tipo de [asserts](#).

3.1.2. Identificación del entorno tecnológico

El sistema a desarrollar consiste en un módulo de gestión de tesauros y normalización del [knowledgeMANAGER](#), una aplicación de mayor tamaño. Por tanto, el entorno tecnológico mínimo en que debe funcionar en una máquina cliente es el mismo que dicha aplicación:

- Microsoft .NET Framework 4.0 instalado.
- Procesador a 1.5GHz mínimo.
- 1024 MB de memoria RAM.
- 128 GB de disco duro.
- Sistema operativo: Windows XP, Windows Vista, Windows 7 (o cualquier revisión de todos ellos).

3.1.3. Especificación de estándares y normas

En este apartado se especifican las restricciones generales que se tengan en la realización del proyecto, posibles supuestos o dependencias y se expone el entorno operacional necesario.

Restricciones Generales

En cuanto a las restricciones generales se encuentran las siguientes:

- Todas las operaciones que tengan acceso a datos deben estar abstraídos mediante una capa DAL. Esta capa está definida mediante un objeto denominado “engine” que es del tipo Cake.Engine.CAKEEngine y que está contenido en la librería Cake.Engine.dll. Por lo tanto todos los objetos que pidan acceso a datos deberán crearse este objeto.
- Todas las operaciones de indexación y normalización deben estar abstraídos mediante un componente externo. Este componente se define mediante un objeto denominado “indexer” que es del tipo Cake.Indexer.Indexer y que está contenido en la librería Cake.Indexer.dll. Por lo tanto todos los objetos que pidan este tipo de operaciones deberán crearse este objeto.
- Todas las clases de la capa de negocio (BLL) deben heredar de una clase común de tipo RepositoryBase.
- Todos los controles de usuario de la capa de presentación (PL) deben heredar de una clase común llamada KnowledgeManagerBaseUserControl.
- Todos los formularios de la capa de presentación (PL) deben heredar de una clase común llamada KnowledgeManagerBaseForm.
- Se deben escribir comentarios apropiados para describir las clases y métodos correctamente con el fin de generar documentación XML.
 - **Para las clases:** es suficiente con indicar el propósito de dicha clase.
 - **Para los métodos:** es necesario añadir un resumen de responsabilidades del método, la fecha y el autor. Además si se considera necesario, se puede añadir información sobre los parámetros de entrada y salida.

Dependencias y Supuestos

Aunque el modulo a desarrollar no supone la creación del [knowledgeMANAGER](#) completo, se va a seguir el estilo grafico definido para la capa de presentación de los distintos módulos, con el fin de mantener la homogeneidad de todos sus componentes de cara a la versión final de la aplicación.

Entorno operacional

En cuanto a los equipos utilizados para la elaboración del código de la aplicación, decir que todos tienen la misma configuración. Entre esas configuraciones destaco las siguientes:

- Sistema operativo Windows 7 Ultimate 64-bit
- Visual Studio 2010 Professional
- .Net Framework 4.0
- Microsoft SQL Server 2008
- Microsoft Office 2010
- Tortoise SVN 1.7.7 64-bit

3.1.4. Identificación de los usuarios participantes y finales

En este apartado se pueden distinguir dos tipos de usuario. Por un lado los stakeholders, que son aquellos usuarios interesados en la realización de la aplicación. Por otro lado los usuarios finales, que son aquellas personas o entidades que van a utilizar realmente el programa.

Stakeholders

- **Universidad Carlos III de Madrid:** entidad donde se realiza el proyecto TFG mediante una beca y donde se va a presentar.
- **The Reuse Company [10]:** interesado en el desarrollo del proyecto para incluir el módulo en su software [knowledgeMANAGER](#).

Usuarios finales

Como se especifica en las tesis doctorales de Anabel Fraga [19] y Sonia Sánchez-Cuadrado [18] cuando se habla de gestión de dominios se definen los siguientes roles y usuarios finales con el fin de optimizar los procesos y obtener los mejores resultados posibles:

- **Jefe de proyecto:** persona que toma decisiones sobre el uso del módulo dentro de la compañía.
- **Arquitectos y analistas software:** personas que utilizan otras herramientas cuyas entradas son las salidas de este módulo, y viceversa.
- **Responsable de calidad:** persona que tome decisiones en herramientas externas que pueden afectar al funcionamiento del módulo.
- **Experto del dominio:** persona encargada de gestionar la ontología de un determinado dominio en una compañía de desarrollo software.

3.1.5. Seguridad requerida en el análisis del sistema de información

Cabe mencionar que aunque el desarrollo de este módulo del [knowledgeMANAGER](#) se desarrolla como TFG para la Universidad Carlos III de Madrid, es condición indispensable y obligatoria que tanto el código desarrollado, como las librerías de [CAKE](#) utilizadas y la documentación generada son confidenciales y propiedad de la empresa The Reuse Company.

3.2. Establecimiento de requisitos

En este apartado se plasman todos los requisitos encontrados para el proyecto. Para dichos requisitos se sigue un formato de tabla común para todos como el siguiente:

RSX - YYY: Nombre del requisito			
Autor:	Autor del requisito	Tipo:	Tipo del requisito
Fecha:	dd/mm/aaaa	Fuente:	Fuente de la que procede
Prioridad:	A/M/B	Necesidad:	A/M/B
Complejidad:	A/M/B	Estado:	Propuesto
Objetivo:	Objetivo principal del requisito		
Precondiciones:	Precondiciones, si las hay, de un requisito		
Descripción:	Descripción clara y concisa del requisito		
Trazabilidad	Requisitos trazables entre sí, es decir, relacionados entre sí.		

Tabla 8 - Plantilla para rellenar los requisitos

La nomenclatura del nombre de cada requisito se hará de la siguiente manera: RSX – YYY, donde YYY representaran tres dígitos auto numéricos que identificarán a cada requisito y X se corresponde con uno o dos caracteres que especificarán el tipo de requisito. Los tipos pueden ser:

- **F:** Requisito Funcional
- **R:** Requisito de rendimiento
- **S:** Requisito de seguridad
- **Ca:** Requisito de calidad
- **In:** Requisito de interfaz

Los campos de prioridad, necesidad y complejidad tendrán tres posibles valores:

- Alto
- Medio
- Bajo

3.2.1. Obtención de requisitos

A continuación, se exponen todos los requisitos del proyecto. Para ello, se sigue la plantilla definida con anterioridad.

RSF - 001: Carga de datos inicial			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Sentar las bases para poder ejecutar la aplicación por primera vez.		
Precondiciones:	Debe existir la base de datos con la que se va a hacer la conexión.		
Descripción:	La base de datos debe contener en todo momento un artefacto Domain*, etiqueta sintáctica NOUN y VERB.		
Trazabilidad	-		

Tabla 9 - RSF - 001: Carga de datos inicial

NOTA: Artefacto Domain es donde se guarda el tesouro de relaciones y términos que se utilizará en la aplicación.

RSF - 002: Visualización del tesouro			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	El usuario debe visualizar en pantalla el tesouro de la ontología.		
Precondiciones:	-		
Descripción:	El sistema debe ser capaz de mostrar los términos y las relaciones del tesouro existentes en la ontología. En esta visualización se deben mostrar las relaciones jerárquicas entre los términos, es decir, relaciones padre-hijo.		
Trazabilidad	RSF – 002, RSF – 027, RSF – 028, RSF – 029, RSF – 030, RSF – 033, RSF – 048, RSF – 049, RSF – 050, RSF – 051, RSF – 052		

Tabla 10 - RSF - 002: Visualización del tesouro

RSF - 003: Acciones sobre términos			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Ofrecer al usuario distintas opciones sobre la vista de términos.		
Precondiciones:	-		
Descripción:	<p>En la vista del tesaurus, el sistema debe ser capaz de mostrar al usuario distintas opciones disponibles sobre los términos, mediante un menú contextual que se abre con el botón derecho. Opciones:</p> <ul style="list-style-type: none"> • Añadir un término raíz • Añadir una nueva relación • Añadir un hijo • Añadir un sinónimo • Añadir una asociación • Añadir otra relación • Editar / Mostrar información de un término • Eliminar un término • Mover el término a la raíz • Mezclar un término con otro término • Expandir todo • Búsqueda web • Refrescar 		
Trazabilidad	RSF – 002, RSF – 004, RSF – 005, RSF – 006, RSF – 007, RSF – 008, RSF – 009, RSF – 010, RSF – 011, RSF – 019, RSF – 020. RSF – 021, RSF – 026.		

Tabla 11 - RSF - 003: Acciones sobre términos

RSF - 004: Añadir un término raíz			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Crear un nuevo término raíz en el tesouro de la ontología.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de crear un término raíz a partir de los datos de carga iniciales. Los datos a rellenar por el usuario son: Nombre (obligatorio), código de clasificación (opcional) y nota de alcance (opcional).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 12 - RSF - 004: Añadir un término raíz

RSF - 005: Añadir una nueva relación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Crear una nueva relación en el tesouro de la ontología.		
Precondiciones:	Deben existir al menos dos términos en el tesouro para crear la relación.		
Descripción:	El usuario debe ser capaz de crear una relación entre términos. Los datos a rellenar por el usuario son: Término 1 de la relación (obligatorio), tipo de relación (obligatorio) y término 1 de la relación (obligatorio).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 13 - RSF - 005: Añadir una nueva relación

RSF - 006: Añadir un hijo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Crear una nueva relación padre-hijo en el tesouro de la ontología.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de crear un término hijo a partir de los datos de carga iniciales. Los datos a rellenar por el usuario son: Nombre (obligatorio), código de clasificación (opcional) y nota de alcance (opcional).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 14 - RSF - 006: Añadir un hijo

RSF - 007: Añadir un sinónimo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Crear una nueva relación de sinonimia entre términos del tesouro de la ontología.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de crear una relación de sinonimia entre términos. Los datos a rellenar por el usuario son: Término sinónimo de la relación (obligatorio).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 15 - RSF - 007: Añadir un sinónimo

RSF - 008: Añadir una asociación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Crear una nueva relación de asociación entre términos del tesouro de la ontología.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de crear una relación de asociación entre términos. Los datos a rellenar por el usuario son: Término asociado de la relación (obligatorio).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 16 - RSF - 008: Añadir una asociación

RSF - 009: Añadir otra relación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Crear una nueva relación de cualquier tipo entre términos del tesouro de la ontología.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de crear una relación entre términos. Los datos a rellenar por el usuario son: Tipo de relación (obligatorio) y término 1 de la relación (obligatorio).		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 17 - RSF - 009: Añadir otra relación

RSF - 010: Editar/ Mostrar información del término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	08/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Mostrar un formulario con la información del término con la posibilidad de modificar la información.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de modificar la información de un término del tesouro. Los datos modificables por el usuario son: Nombre, código de clasificación, nota histórica y nota de alcance.		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 18 - RSF - 010: Editar/ Mostrar información del término

RSF - 011: Eliminar un término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Eliminar un término del tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	<p>El usuario debe ser capaz de eliminar un término del tesouro. Opciones de borrado:</p> <ul style="list-style-type: none"> • Eliminar el término y hacer sus hijos raíces • Eliminar el término y sus hijos • Mover sus hijos a otro término raíz • Mover sus hijos a otro padre • Sustituir por un sinónimo • Sustituir por un hijo 		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 19 - RSF - 011: Eliminar un término

RSF - 012: Eliminar el término y hacer sus hijos raíces			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Eliminar un término del tesoro y convertir sus hijos (si tiene) en términos raíces del tesoro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de eliminar un término del tesoro y mover sus hijos a términos raíces. El proceso es transparente al usuario.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 20 - RSF - 012: Eliminar el término y hacer sus hijos raíces

RSF - 013: Eliminar el término y sus hijos			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Eliminar un término y sus hijos (si tiene) del tesoro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de eliminar un término y sus hijos del tesoro. El proceso es transparente al usuario.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 21 - RSF - 013: Eliminar el término y sus hijos

RSF - 014: Mover sus hijos a otro término raíz			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Eliminar un término y mover sus hijos a otro término raíz del tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro. Deben existir al menos dos términos raíces en el tesouro.		
Descripción:	El usuario debe ser capaz de eliminar un término y mover sus hijos a otro término raíz del tesouro. El usuario debe elegir el término raíz al que quiere mover los hijos.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 22 - RSF - 014: Mover sus hijos a otro término raíz

RSF - 015: Mover sus hijos a otro término padre			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Eliminar un término y mover sus hijos a otro término padre del tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro. Deben existir al menos dos términos padres en la rama del tesouro.		
Descripción:	El usuario debe ser capaz de eliminar un término y mover sus hijos a otro término padre del tesouro. El usuario debe elegir el término padre al que quiere mover los hijos.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 23 - RSF - 015: Mover sus hijos a otro término padre

RSF - 016: Sustituir por un sinónimo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	El sinónimo del término pasa a ser el término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesauro. El término seleccionado debe tener al menos una relación de sinonimia.		
Descripción:	El usuario debe ser capaz de eliminar un término y sustituirlo por un sinónimo. El usuario debe elegir el término sinónimo que va a reemplazar al término a borrar.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 24 - RSF - 016: Sustituir por un sinónimo

RSF - 017: Sustituir por un hijo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	El hijo del término pasa a ser el término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesauro. El término seleccionado debe tener al menos un hijo.		
Descripción:	El usuario debe ser capaz de eliminar un término y sustituirlo por un hijo. El usuario debe elegir el término hijo que va a reemplazar al término a borrar.		
Trazabilidad	RSF – 003, RSF – 011		

Tabla 25 - RSF - 017: Sustituir por un hijo

RSF - 018: Mover el término a la raíz			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	El término pasa a ser un término raíz del tesouro.		
Precondiciones:	Debe existir al menos un término que no sea raíz seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de mover un término a la raíz. El proceso es transparente para el usuario.		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 26 - RSF - 018: Mover el término a la raíz

RSF - 019: Mezclar un término con otro término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Media
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Fusionar los dos términos en uno.		
Precondiciones:	Deben existir al menos dos términos en el tesouro.		
Descripción:	El usuario debe ser capaz de fusionar dos términos en uno. Para ello debe elegir el segundo término de la fusión y cuál es el término que se va a quedar del tesouro. Este proceso elimina, modifica y añade nuevas relaciones en el tesouro.		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 27 - RSF - 019: Mezclar un término con otro término

RSF - 020: Expandir todo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	09/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Expandir todos los hijos de una rama del tesouro.		
Precondiciones:	Debe existir al menos un término con hijos en el tesouro.		
Descripción:	El usuario debe ser capaz de expandir una rama del tesouro. Este proceso es transparente al usuario y puede tardar mucho dependiendo de lo grande que sea la rama del tesouro.		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 28 - RSF - 020: Expandir todo

RSF - 021: Búsqueda web			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Baja
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Buscar en motores de búsqueda los términos del tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	<p>El usuario debe ser capaz de buscar términos del tesouro en la web. Para este proceso:</p> <ul style="list-style-type: none"> • Añadir buscador • Eliminar buscadores • Opciones de búsqueda • Buscadores 		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 29 - RSF - 021: Búsqueda web

RSF - 022: Añadir buscadores			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Baja
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Añadir un motor de búsqueda a los buscadores del tesoro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de añadir un motor de búsqueda al tesoro. Los datos a añadir por el usuario son: Nombre, URL del buscador habiendo buscado en él la palabra TEST.		
Trazabilidad	RSF – 003, RSF – 021		

Tabla 30 - RSF - 022: Añadir buscadores

RSF - 023: Eliminar buscadores			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Baja
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Eliminar un motor de búsqueda de los buscadores del tesoro.		
Precondiciones:	Debe existir al menos un buscador en los buscadores del tesoro.		
Descripción:	El usuario debe ser capaz de eliminar un motor de búsqueda del tesoro. El usuario debe seleccionar los motores de búsqueda que quiere borrar.		
Trazabilidad	RSF – 003, RSF – 021		

Tabla 31 - RSF - 023: Eliminar buscadores

RSF - 024: Opciones de búsqueda			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Baja
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Facilitar opciones de búsqueda al usuario en los buscadores.		
Precondiciones:	-		
Descripción:	<p>El usuario debe ser capaz de elegir las opciones de búsqueda que más le convengan. Opciones:</p> <ul style="list-style-type: none"> • Incluir Padres • Incluir Hijos • Incluir Sinónimos <p>Esto significa que en la búsqueda incluirá los padres, hijos o sinónimos del término del tesouro.</p>		
Trazabilidad	RSF – 003, RSF – 021		

Tabla 32 - RSF - 024: Opciones de búsqueda

RSF - 025: Buscadores			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Baja	Necesidad:	Baja
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Listar los diferentes buscadores y buscar en ellos el término seleccionado		
Precondiciones:	Debe existir un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de buscar el término en un motor de búsqueda añadido a los buscadores del tesouro.		
Trazabilidad	RSF – 003, RSF – 021		

Tabla 33 - RSF - 025: Buscadores

RSF - 026: Refrescar			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Refrescar el tesouro de la ontología.		
Precondiciones:	Debe existir al menos un término en el tesouro.		
Descripción:	El usuario debe ser capaz de refrescar el tesouro de la ontología por si quiere volver al estado inicial.		
Trazabilidad	RSF – 003, RSF – 048		

Tabla 34 - RSF - 026: Refrescar

RSF - 027: Búsqueda simple de términos			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Búsqueda simple por nombre de términos o sinónimos del tesouro.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de buscar de una manera rápida en el tesouro de la ontología. En los resultados no se muestran jerarquías padre-hijo.		
Trazabilidad	RSF – 002, RSF – 048		

Tabla 35 - RSF - 027: Búsqueda simple de términos

RSF - 028: Visualización de términos en vista filtrada			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Visualización de los términos después de una búsqueda simple.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de visualizar los resultados de los términos tras una búsqueda simple.		
Trazabilidad	RSF – 002, RSF – 029, RSF – 030, RSF – 048		

Tabla 36 - RSF - 028: Visualización de términos en vista filtrada

RSF - 029: Volver a la vista jerárquica manteniendo estado			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Visualización de la vista jerárquica del tesauro manteniendo el estado de éste.		
Precondiciones:	Estar en la vista filtrada del tesauro.		
Descripción:	El usuario debe ser capaz de volver a visualizar la jerarquía del tesauro con el mismo estado en el que se encontraba.		
Trazabilidad	RSF – 002, RSF – 028, RSF – 048		

Tabla 37 - RSF - 029: Volver a la vista jerárquica manteniendo estado

RSF - 030: Volver a la vista jerárquica sin mantener el estado			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	10/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Visualización de la vista jerárquica del tesouro sin mantener el estado de éste.		
Precondiciones:	Estar en la vista filtrada del tesouro.		
Descripción:	El usuario debe ser capaz de volver a visualizar la jerarquía del tesouro en su estado inicial sin mantener el estado en el que se encontraba.		
Trazabilidad	RSF – 002, RSF – 028, RSF – 048		

Tabla 38 - RSF - 030: Volver a la vista jerárquica sin mantener el estado

RSF - 031: Búsqueda avanzada de términos en el tesouro			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Buscar términos incluidos en el tesouro utilizando diferentes criterios u opciones más específicas de búsqueda.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de buscar un término perteneciente al tesouro utilizando distintos filtros de búsqueda, mostrándose los resultados en la misma vista de términos. Se debe poder filtrar por Nombre, etiqueta sintáctica, semántica, rama del tesouro, tipo de comparación y búsqueda en. En el caso de existir más de un criterio, la búsqueda debe ser multicampo.		
Trazabilidad	-		

Tabla 39 - RSF - 031: Búsqueda avanzada de términos en el tesouro

RSF - 032: Tooltip para términos en el tesaurus			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Mostrar información relevante al término al dejar el ratón encima de un término durante un periodo de tiempo.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de mostrar información de un término con el uso de tooltips, pudiendo elegir si se muestra o no el tooltip y si se muestra además que tipo de información mostrar. Información: <ul style="list-style-type: none">• Padres• Términos raíces• Sinónimos• Clasificador• Nota de alcance• Nota histórica		
Trazabilidad	RSF – 048		

Tabla 40 - RSF - 032: Tooltip para términos en el tesaurus

RSF - 033: Información del término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Mostrar información de un término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	<p>El usuario debe ser capaz de visualizar en cualquier momento toda la información de un término en el tesouro. Esta información es:</p> <ul style="list-style-type: none"> • Término: información básica del término • Relaciones: todas las relaciones del término • Sinónimos: todos los sinónimos del término • Avanzado: información específica del término 		
Trazabilidad	RSF – 002, RSF – 034, RSF – 036, RSF – 040		

Tabla 41 - RSF - 033: Información del término

RSF - 034: Término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Mostrar información básica de un término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	<p>El usuario debe ser capaz de visualizar en cualquier momento toda la información básica de un término en el tesouro. La información es: Nombre, nota de alcance, clasificador, nota historia y miga de pan.</p>		
Trazabilidad	RSF – 033		

Tabla 42 - RSF - 034: Término

RSF - 035: Miga de pan del término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Mostrar camino del término en la jerarquía del tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	El usuario debe ser capaz de visualizar en cualquier momento la miga de pan del término además de poder navegar en el tesouro pulsando en los diferentes términos mostrados en esa miga de pan. Esto debe funcionar en ambas vistas: jerárquica y filtrada.		
Trazabilidad	RSF – 033, RSF – 035, RSF – 048		

Tabla 43 - RSF - 035: Miga de pan del término

RSF - 036: Relaciones			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Mostrar todas las relaciones que tiene un término en el tesouro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro.		
Descripción:	<p>El usuario debe ser capaz de visualizar todas las relaciones de un término en el tesouro y su nombre. Además en esta ventana se debe visualizar de diferentes maneras las relaciones:</p> <ul style="list-style-type: none"> • Todas las relaciones: muestra todas las relaciones • Por categoría: se muestran por tipo • Gráficas: se muestra un diagrama 		
Trazabilidad	RSF – 033		

Tabla 44 - RSF - 036: Relaciones

RSF - 037: Añadir nueva relación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Añadir una relación nueva desde la pestaña de todas las relaciones del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de añadir una relación del término seleccionado y otro término del tesoro desde esta ventana.		
Trazabilidad	RSF – 033, RSF – 036, RSF – 048		

Tabla 45 - RSF - 037: Añadir nueva relación

RSF - 038: Eliminar relación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Eliminar una relación desde la pestaña de todas las relaciones del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de eliminar una relación del término seleccionado del tesoro desde esta ventana.		
Trazabilidad	RSF – 033, RSF – 036, RSF – 048		

Tabla 46 - RSF - 038: Eliminar relación

RSF - 039: Mostrar otro participante de la relación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Mostrar información del otro término de la relación en un formulario.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesaurus.		
Descripción:	El usuario debe ser capaz de visualizar la información del otro participante de la relación en un formulario nuevo de detalles.		
Trazabilidad	RSF – 033, RSF – 036		

Tabla 47 - RSF - 039: Mostrar otro participante de la relación

RSF - 040: Sinónimos			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	11/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Mostrar todos los sinónimos de un término del tesaurus.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesaurus.		
Descripción:	El usuario debe ser capaz de visualizar todos los sinónimos de un término del tesaurus.		
Trazabilidad	RSF – 033		

Tabla 48 - RSF - 040: Sinónimos

RSF - 041: Añadir sinónimo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Añadir una nueva relación de sinonimia desde la pestaña de sinónimos del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de añadir una nueva relación de sinonimia desde la ventana de sinónimos del tesoro.		
Trazabilidad	RSF – 033, RSF – 040, RSF – 048		

Tabla 49 - RSF - 041: Añadir sinónimo

RSF - 042: Eliminar sinónimo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Eliminar una relación de sinonimia desde la pestaña de sinónimos del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de eliminar una relación de sinonimia desde la ventana de sinónimos del tesoro.		
Trazabilidad	RSF – 033, RSF – 040, RSF – 048		

Tabla 50 - RSF - 042: Eliminar sinónimo

RSF - 043: Modificar sinónimo			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Modificar el nombre del término sinónimo de la relación desde la pestaña de sinónimos del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesaurus.		
Descripción:	El usuario debe ser capaz de editar el nombre del sinónimo de un término desde la ventana de sinónimos del tesaurus.		
Trazabilidad	RSF – 033, RSF – 040, RSF – 048		

Tabla 51 - RSF - 043: Modificar sinónimo

RSF - 044: Usar por el término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Intercambiar la relación de sinonimia, siendo el sinónimo el término y el término el sinónimo de la relación.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesaurus.		
Descripción:	El usuario debe ser capaz de intercambiar el orden de la relación de sinonimia dentro de la venta de sinónimos del tesaurus.		
Trazabilidad	RSF – 033, RSF – 040, RSF – 048		

Tabla 52 - RSF - 044: Usar por el término

RSF - 045: Importación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Importar un tesouro para facilitar la creación de uno.		
Precondiciones:	Los ficheros de importación deben ser de la extensión *.jcd y *.the.		
Descripción:	El usuario debe ser capaz de importar un tesouro en diversos formatos.		
Trazabilidad	-		

Tabla 53 - RSF - 045: Importación

RSF - 046: Exportación			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Exportar el tesouro de la ontología en diferentes formatos para facilitar su estudio e importación posterior.		
Precondiciones:	Tener un tesouro de dominio creado.		
Descripción:	El usuario debe ser capaz de exportar un tesouro en diversos formatos. Puede elegir si quiere exportar todo el tesouro o partes de él.		
Trazabilidad	-		

Tabla 54 - RSF - 046: Exportación

RSF - 047: Estadísticas			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Mostrar las estadísticas del tesoro de una forma que se puedan sacar fáciles conclusiones.		
Precondiciones:	Tener un tesoro de dominio creado.		
Descripción:	El usuario debe ser capaz de visualizar las estadísticas de un tesoro. Los datos a visualizar son: número de hijos, número de términos raíces, número total de términos, media de relaciones por término, media de sinónimos por término. Otras menos importantes en nuestro modulo: número total de sugerencias de los términos, número total de sugerencias de las relaciones.		
Trazabilidad	-		

Tabla 55 - RSF - 047: Estadísticas

RSF - 048: Pestaña avanzada del término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Mostrar la información avanzada de un término en el tesoro.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de visualizar la información avanzada de un término del tesoro.		
Trazabilidad	RSF – 033		

Tabla 56 - RSF - 048: Pestaña avanzada del término

RSF - 049: Modificar información avanzada del término			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Se debe poder modificar la información avanzada del término.		
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro.		
Descripción:	El usuario debe ser capaz de modificar la información avanzada de un término del tesoro. Esta información es: <ul style="list-style-type: none">• Etiqueta sintáctica• Semántica• Lenguaje• Otras datos		
Trazabilidad	RSF – 033		

Tabla 57 - RSF - 049: Modificar información avanzada del término

RSF - 050: Refresco visual			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	12/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Refrescar todo lo que visualmente se tenga en pantalla y se esté modificando con alguna funcionalidad.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de visualizar los cambios que sean producidos por alguna funcionalidad en todas las ventanas que tenga abiertas e impliquen algún cambio.		
Trazabilidad	RSF - 002, RSF - 004, RSF - 005, RSF - 006, RSF - 007, RSF - 008, RSF - 009, RSF - 010, RSF - 011, RSF - 019, RSF - 020, RSF - 021, RSF - 026, RSF - 027, RSF - 028, RSF - 029, RSF - 030, RSF - 032, RSF - 035, RSF - 037, RSF - 038, RSF - 041, RSF - 042, RSF - 043, RSF - 044, RSF - 049		

Tabla 58 - RSF - 050: Refresco visual

RSF - 051: Drag & Drop			
Autor:	Álvaro García	Tipo:	Funcional
Fecha:	15/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Alta
Complejidad:	Alta	Estado:	Propuesto
Objetivo:	Arrastrar y soltar elementos del tesouro con el movimiento del ratón.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de poder arrastrar y soltar términos del tesouro y con ello añadir, modificar o eliminar relaciones entre términos.		
Trazabilidad	RSF - 002, RSF - 048		

Tabla 59- RSF - 051: Drag & Drop

RSIn - 052: Visualización de tesouro y detalles			
Autor:	Álvaro García	Tipo:	Interfaz
Fecha:	15/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Facilitar la visualización del tesouro y el detalle de cualquier término seleccionado.		
Precondiciones:	-		
Descripción:	El usuario debe ser capaz de visualizar al mismo tiempo la jerarquía del tesouro y los detalles del término seleccionado. Además debe poder expandir o contraer cualquiera de las vistas para mejorar la visualización.		
Trazabilidad	RSF – 002, RSF – 033		

Tabla 60 - RSIn - 052: Visualización de tesouro y detalles

RSCa - 053: Ordenación de los listados			
Autor:	Álvaro García	Tipo:	Calidad
Fecha:	15/10/2012	Fuente:	The Reuse Company
Prioridad:	Media	Necesidad:	Media
Complejidad:	Baja	Estado:	Propuesto
Objetivo:	Mejorar la navegación y usabilidad de los listados		
Precondiciones:	-		
Descripción:	El sistema debe permitir la ordenación de los elementos de los listados haciendo click en la cabecera de la columna por la que se quiere ordenar.		
Trazabilidad	RSF – 002, RSF – 028, RSF – 031, RSF – 036, RSF – 040, RSF – 048		

Tabla 61 - RSCa - 053: Ordenación de los listados

RSR - 054: Carga del tesouro			
Autor:	Álvaro García	Tipo:	Rendimiento
Fecha:	15/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Mejorar los tiempos de carga del tesouro de jerarquías.		
Precondiciones:	-		
Descripción:	El sistema debe cargar las ramas necesarias en cada momento para mejorar el rendimiento de visualización y usabilidad del tesouro en tesouros más complejos y grandes.		
Trazabilidad	RSF – 002, RSF – 050		

Tabla 62 - RSR - 054: Carga del tesouro

RSS - 055: Login			
Autor:	Álvaro García	Tipo:	Seguridad
Fecha:	15/10/2012	Fuente:	The Reuse Company
Prioridad:	Alta	Necesidad:	Alta
Complejidad:	Media	Estado:	Propuesto
Objetivo:	Loguearse a la aplicación con usuarios específicos y controlados.		
Precondiciones:	-		
Descripción:	El sistema debe pedir loguearse con usuarios específicos de la aplicación y de la base de datos que se va a utilizar.		
Trazabilidad	-		

Tabla 63 - RSS - 055: Login

3.2.2. Especificación de casos de uso

En este apartado se analizan los casos de uso, introduciendo ilustraciones para mostrar visualmente dichos casos y tablas para completar la descripción de los casos y algunos de los escenarios donde se utilizarán.

Para una mejor comprensión se han dividido los casos de uso por funcionalidad: términos, relaciones, sinónimos y otras funcionalidades.

Además en cada caso de uso de las principales funciones, es decir, si se repite alguna o es similar no se duplicara se integraran los diagramas de secuencia para ver la interacción entre los objetos.

3.2.2.1. Términos

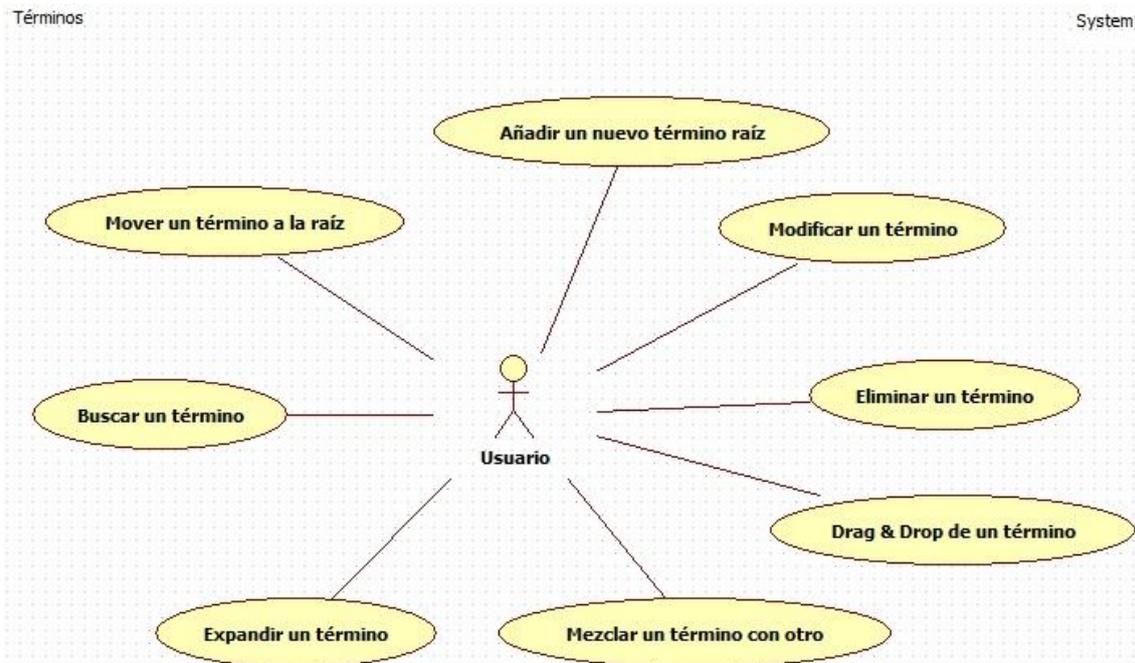


Ilustración 7 - Casos de uso Términos

Caso de uso: Añadir un nuevo término raíz	
Identificador:	CU-001
Descripción:	El usuario tiene la posibilidad de añadir un nuevo término raíz que será introducido en el tesouro de la ontología.
Precondiciones:	-
Postcondiciones	Un nuevo término raíz ha sido introducido en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none"> 1. Entra en el apartado de Tesouro. 2. En el menú contextual, elegir la opción de añadir un nuevo término raíz. 3. Rellenar en un nuevo formulario los atributos que sean obligatorios para crear el término.
Condiciones de fallo:	No está permitido crear un término raíz con nombre vacío o repetido.

Tabla 64 - CU-001: Añadir un nuevo término raíz

Caso de uso: Modificar un término	
Identificador:	CU-002
Descripción:	El usuario tiene la posibilidad de modificar un término que ha sido creado previamente en el tesouro de la ontología.
Precondiciones:	Debe existir al menos un término en el tesouro de la ontología y debe haber al menos un término seleccionado.
Postcondiciones	Las modificaciones sobre el término son almacenadas en el tesouro de la ontología y visibles para el usuario.
Escenario:	<ol style="list-style-type: none"> 1. Entra en el apartado de Tesouro. 2. Selecciona un término y elegir la opción de editar un término en el menú contextual. 3. Modificar cualquier campo en el formulario emergente. 3. Pulsar sobre el botón aceptar.
Condiciones de fallo:	Se repiten las mismas condiciones de fallo que en CU-001.

Tabla 65 - CU-002: Modificar un término

Caso de uso: Eliminar un término	
Identificador:	CU-003
Descripción:	El usuario tiene la posibilidad de eliminar un término que ya existe en el tesouro de la ontología.
Precondiciones:	Debe existir al menos un término en el tesouro de la ontología y debe haber al menos un término seleccionado.
Postcondiciones	Un término existente ha sido eliminado del tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Selecciona un término y elegir la opción de eliminar un término en el menú contextual.3. Elegir la opción de borrado deseada.3. Confirmar el borrado.
Condiciones de fallo:	-

Tabla 66 - CU-003: Eliminar un término

Caso de uso: Drag & Drop de un término	
Identificador:	CU-004
Descripción:	El usuario tiene la posibilidad de copiar o mover un término en otra rama del tesouro de la ontología.
Precondiciones:	Debe existir al menos dos términos en el tesouro de la ontología y debe haber al menos un término seleccionado.
Postcondiciones	Las modificaciones sobre el término son almacenadas en el tesouro de la ontología y visibles para el usuario.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Selecciona un término y se arrastra el término pulsando control en el teclado para copiar o simplemente arrastrar para mover.3. Soltar encima del término donde se quiere copiar el término.4. Confirmar la copia del término.
Condiciones de fallo:	No se puede copiar o mover si ya existe una relación entre los términos implicados.

Tabla 67 - CU-004: Copiar un término

Caso de uso: Mezclar un término en otro	
Identificador:	CU-005
Descripción:	El usuario tiene la posibilidad de mezclar un término con otro que ha sido creado previamente en el tesauro de la ontología.
Precondiciones:	Debe existir al menos dos términos en el tesauro de la ontología y debe haber al menos un término seleccionado.
Postcondiciones	Las modificaciones sobre el término son almacenadas en el tesauro de la ontología y visibles para el usuario.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesauro.2. Selecciona un término y elegir la opción de mezclar un término con otro del menú contextual.3. Rellenar el campo del nuevo formulario con el término a fusionar. Para ello:<ol style="list-style-type: none">3.1. Pinchar en buscar y se abre un nuevo formulario.3.2. Escribir el término que se quiere fusionar.3.3. Selecciona el término y pulsa en aceptar.4. Elegir el término que se quedara en el tesauro de la ontología.5. Pulsar sobre el botón aceptar.
Condiciones de fallo:	-

Tabla 68 - CU-005: Mezclar un término en otro

Caso de uso: Expandir un término	
Identificador:	CU-006
Descripción:	El usuario tiene la posibilidad de expandir un término del tesouro de la ontología.
Precondiciones:	Debe existir al menos un término con hijos en el tesouro de la ontología y debe haber al menos un término seleccionado.
Postcondiciones	Los cambios sobre el término son visualizados en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Selecciona un término y elegir opción expandir del menú contextual del tesouro.3. Esperar a que el proceso acabe.
Condiciones de fallo:	-

Tabla 69 - CU-006: Expandir un término

Caso de uso: Buscar un término	
Identificador:	CU-007
Descripción:	El usuario tiene la posibilidad de buscar un término del tesouro de la ontología.
Precondiciones:	Debe existir al menos un término en el tesouro de la ontología.
Postcondiciones	Los resultados de búsqueda sobre el término son visualizados en la vista filtrada del tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Escribir el nombre del término en el buscador simple.3. Pulsar en el botón buscar.
Condiciones de fallo:	-

Tabla 70 - CU-007: Buscar un término

Caso de uso: Mover un término a la raíz	
Identificador:	CU-008
Descripción:	El usuario tiene la posibilidad de mover un término a la raíz del tesoro de la ontología.
Precondiciones:	Debe existir al menos un término seleccionado que no sea raíz.
Postcondiciones	Las modificaciones sobre el término son almacenadas en el tesoro de la ontología y visibles para el usuario.
Escenario:	1. Entra en el apartado de Tesoro. 2. Selecciona un término y elegir opción del menú contextual mover término a la raíz.
Condiciones de fallo:	No se puede copiar o mover si ya existe una relación entre los términos implicados.

Tabla 71 - CU-008: Mover un término a la raíz

3.2.2.2. Relaciones

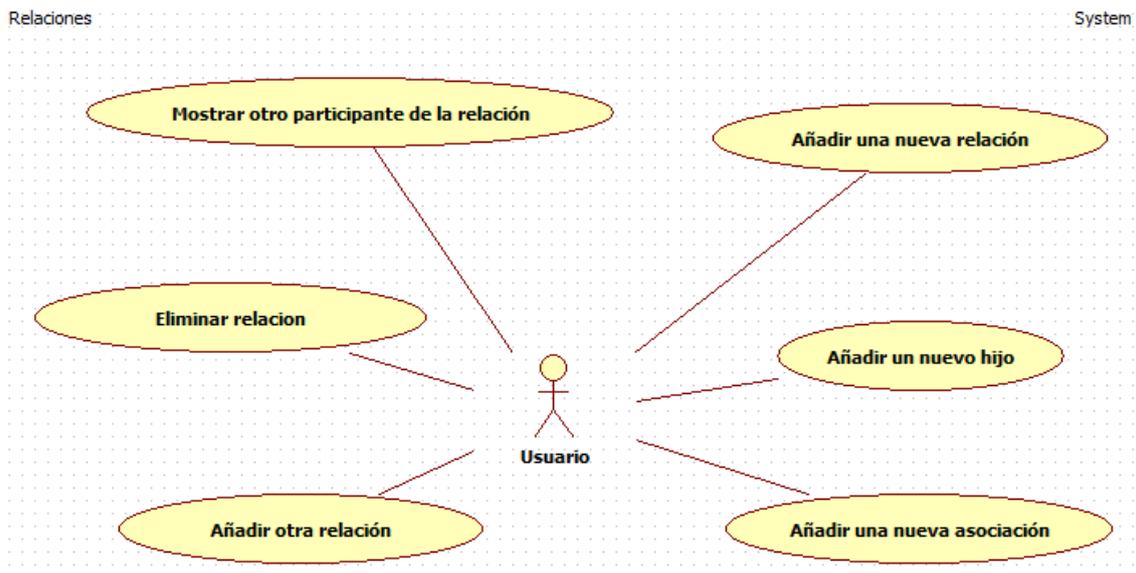


Ilustración 8 - Casos de uso Relaciones

Caso de uso: Añadir una nueva relación	
Identificador:	CU-009
Descripción:	El usuario tiene la posibilidad de añadir una nueva relación que será introducido en el tesouro de la ontología.
Precondiciones:	Deben existir al menos dos términos en el tesouro de la ontología.
Postcondiciones	Una nueva relación ha sido introducida en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. En el menú contextual, elegir la opción de añadir una nueva relación.3. Rellenar en un nuevo formulario los términos que forman parte de la relación. Para ello:<ol style="list-style-type: none">3.1. Pinchar en buscar y se abre un nuevo formulario.3.2. Escribir el término que se quiere relacionar.3.3. Selecciona el término y pulsa en aceptar.3.4. Repetir el proceso con el tipo de relación (en este caso se elige el tipo de relación) y el segundo término de la relación.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	No está permitido crear una relación entre términos que ya están relacionados de alguna manera.

Tabla 72 - CU-009: Añadir una nueva relación

Caso de uso: Añadir un nuevo hijo	
Identificador:	CU-010
Descripción:	El usuario tiene la posibilidad de añadir un nuevo hijo que será introducido en un término del tesouro de la ontología.
Precondiciones:	Deben existir al menos un término en el tesouro de la ontología.
Postcondiciones	Una nueva relación de jerarquía ha sido introducida en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. En el menú contextual, elegir la opción de añadir nuevo hijo.3. Rellenar en un nuevo formulario los atributos que sean obligatorios para crear el término.
Condiciones de fallo:	No está permitido crear un término hijo con nombre vacío o repetido.

Tabla 73 - CU-010: Añadir un nuevo hijo

Caso de uso: Añadir una nueva asociación	
Identificador:	CU-011
Descripción:	El usuario tiene la posibilidad de añadir una nueva relación de asociación que será introducido del tesouro de la ontología.
Precondiciones:	Deben existir al menos dos términos en el tesouro de la ontología.
Postcondiciones	Una nueva relación de asociación ha sido introducida en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. En el menú contextual, elegir la opción de añadir nueva asociación.3. Rellenar en un nuevo formulario el término asociado que forman parte de la relación. Para ello:<ol style="list-style-type: none">3.1. Pinchar en buscar y se abre un nuevo formulario.3.2. Escribir el término que se quiere asociar.3.3. Selecciona el término y pulsa en aceptar.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	No está permitido crear una relación entre términos que ya están relacionados de alguna manera.

Tabla 74 - CU-011: Añadir una nueva asociación

Caso de uso: Añadir otra relación	
Identificador:	CU-012
Descripción:	El usuario tiene la posibilidad de añadir otra relación que será introducido en el tesouro de la ontología.
Precondiciones:	Deben existir al menos dos términos en el tesouro de la ontología.
Postcondiciones	Una nueva relación ha sido introducida en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. En el menú contextual, elegir la opción de añadir una nueva relación.3. Rellenar en un nuevo formulario el término que forman parte de la relación y el tipo de relación deseada. Para ello:<ol style="list-style-type: none">3.1. Pinchar en buscar y se abre un nuevo formulario.3.2. Escribir el término que se quiere relacionar.3.3. Selecciona el término y pulsa en aceptar.3.4. Repetir el proceso con el tipo de relación (en este caso se elige el tipo de relación) y el segundo término de la relación.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	No está permitido crear una relación entre términos que ya están relacionados de alguna manera.

Tabla 75 - CU-012: Añadir otra relación

Caso de uso: Eliminar una relación	
Identificador:	CU-013
Descripción:	El usuario tiene la posibilidad de eliminar una relación entre términos ya existentes en el tesouro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesouro de la ontología. El término seleccionado debe tener alguna relación en el tesouro de la ontología.
Postcondiciones	Una relación ha sido eliminada en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Seleccionar un término y entrar en la pestaña de relaciones del término.3. Seleccionar la relación.4. Pinchar botón derecho menú contextual y elegir opción de borrar relación.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	-

Tabla 76 - CU-013: Eliminar una nueva relación

Caso de uso: Mostrar el otro participante de la relación	
Identificador:	CU-014
Descripción:	El usuario tiene la posibilidad de visualizar el otro participante de la relación existente en el tesouro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesouro de la ontología. El término seleccionado debe tener alguna relación en el tesouro de la ontología.
Postcondiciones	Se visualiza la información del otro término de la relación del tesouro.
Escenario:	<ol style="list-style-type: none"> 1. Entra en el apartado de Tesouro. 2. Seleccionar un término y entrar en la pestaña de relaciones del término. 3. Seleccionar la relación. 4. Pinchar botón derecho menú contextual y elegir opción de mostrar otro participante de la relación.
Condiciones de fallo:	-

Tabla 77 - CU-014: Mostrar el otro participante de la relación

3.2.2.3. Sinónimos

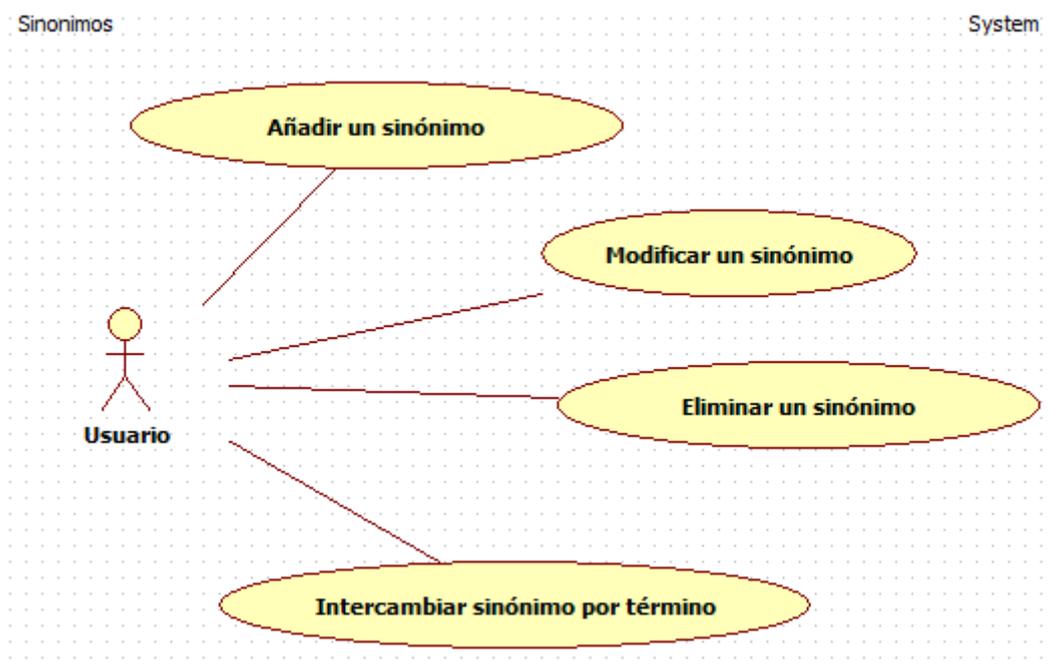


Ilustración 9 - Casos de uso Sinónimos

Caso de uso: Añadir un sinónimo	
Identificador:	CU-015
Descripción:	El usuario tiene la posibilidad de añadir una nueva relación de sinonimia que será introducido del tesauro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesauro de la ontología.
Postcondiciones	Una nueva relación de sinonimia ha sido introducida en el tesauro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesauro.2. En el menú contextual, elegir la opción de añadir nuevo sinónimo.3. Rellenar en un nuevo formulario el término sinónimo que forman parte de la relación.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	No está permitido crear un término sinónimo con nombre vacío o repetido.

Tabla 78- CU-015: Añadir un sinónimo

Caso de uso: Modificar un sinónimo	
Identificador:	CU-016
Descripción:	El usuario tiene la posibilidad de modificar un sinónimo de un término existente en el tesouro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesouro de la ontología. El término seleccionado debe tener algún sinónimo en el tesouro de la ontología.
Postcondiciones	Las modificaciones sobre el sinónimo del término son almacenadas en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Seleccionar un término y entrar en la pestaña de sinónimos del término.3. Seleccionar el sinónimo.4. Pinchar botón derecho menú contextual y elegir opción de editar sinónimo.5. Modifica la información en el formulario emergente.6. Pulse botón aceptar para confirmar.
Condiciones de fallo:	Las mismas que el caso de uso CU-015.

Tabla 79 - CU-016: Modificar un sinónimo

Caso de uso: Eliminar un sinónimo	
Identificador:	CU-017
Descripción:	El usuario tiene la posibilidad de eliminar un sinónimo de un término existente en el tesouro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesouro de la ontología. El término seleccionado debe tener algún sinónimo en el tesouro de la ontología.
Postcondiciones	Un sinónimo de un término ha sido eliminado del tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Seleccionar un término y entrar en la pestaña de sinónimos del término.3. Seleccionar el sinónimo.4. Pinchar botón derecho menú contextual y elegir opción de eliminar sinónimo.5. Pulse botón de confirmación.
Condiciones de fallo:	-

Tabla 80 - CU-017: Eliminar un sinónimo

Caso de uso: Intercambiar sinónimo por término	
Identificador:	CU-018
Descripción:	El usuario tiene la posibilidad de intercambiar el sinónimo por el término cambiando así el sentido de la relación de sinonimia existente en el tesauro de la ontología.
Precondiciones:	Deben existir al menos un término seleccionado en el tesauro de la ontología. El término seleccionado debe tener algún sinónimo en el tesauro de la ontología.
Postcondiciones	Un sinónimo de un término ha sido intercambiado por el término del tesauro de la ontología.
Escenario:	<ol style="list-style-type: none"> 1. Entra en el apartado de Tesauro. 2. Seleccionar un término y entrar en la pestaña de sinónimos del término. 3. Seleccionar el sinónimo. 4. Pinchar botón derecho menú contextual y elegir opción de intercambiar sinónimo por término.
Condiciones de fallo:	-

Tabla 81 - CU-018: Intercambiar sinónimo por término

3.2.2.4. Otras funcionalidades

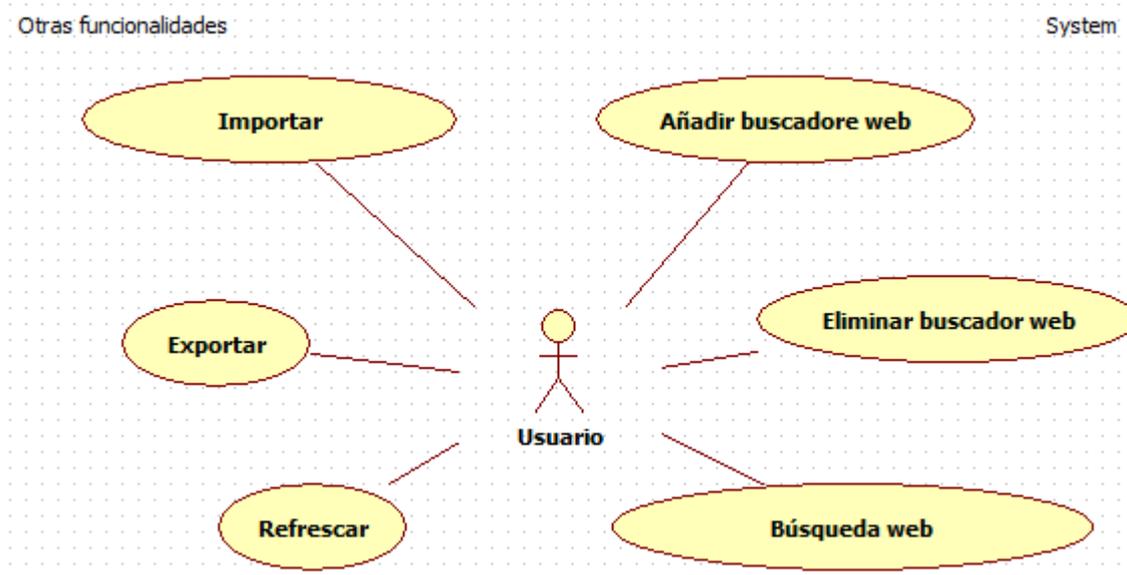


Ilustración 10 - Casos de uso Otras funcionalidades

Caso de uso: Refrescar	
Identificador:	CU-019
Descripción:	El usuario tiene la posibilidad de refrescar el tesouro de la ontología.
Precondiciones:	Debe existir al menos un término en el tesouro de la ontología.
Postcondiciones	Los resultados de refresco son visualizados en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. Pulsar icono de home en el tesouro o elegir refrescar en el menú contextual del tesouro.
Condiciones de fallo:	-

Tabla 82 - CU-019: Refrescar

Caso de uso: Importar	
Identificador:	CU-020
Descripción:	El usuario tiene la posibilidad de importar un tesouro para la ontología.
Precondiciones:	-
Postcondiciones	El tesouro importado se visualiza en el apartado de tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Importación.2. Pulsar botón importar.3. Elegir el fichero a importar. Pueden ser *.jcd o *.the.3. Esperar que el proceso acabe.4. Confirmar el proceso.
Condiciones de fallo:	Fallos en la importación de algunos términos por la duplicidad de ellos ya en el tesouro.

Tabla 83 - CU-020: Importar

Caso de uso: Exportar	
Identificador:	CU-021
Descripción:	El usuario tiene la posibilidad de exportar el tesouro existente de la ontología.
Precondiciones:	Debe existir al menos un tesouro de la ontología.
Postcondiciones	Se visualiza en un fichero el tesouro exportado de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Exportación.2. Elegir la exportación deseada y pulsar en el botón exportar.3. Esperar que el proceso acabe.4. Confirmar el proceso.
Condiciones de fallo:	-

Tabla 84 - CU-021: Exportar

Caso de uso: Añadir un buscador	
Identificador:	CU-022
Descripción:	El usuario tiene la posibilidad de añadir un buscador web al tesouro de la ontología.
Precondiciones:	Debe existir al menos un término seleccionado en el tesouro de la ontología.
Postcondiciones	Una nuevo buscador web en el tesouro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesouro.2. En el menú contextual, elegir la opción de añadir nuevo buscador web.3. Rellenar en un nuevo formulario los parámetros: URL (siguiendo instrucciones mostradas) y nombre del buscador.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	Mal introducida la URL o nombre repetido del buscador

Tabla 85 - CU-022: Añadir un buscador

Caso de uso: Eliminar un buscador	
Identificador:	CU-023
Descripción:	El usuario tiene la posibilidad de eliminar un buscador web el tesoro de la ontología.
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro de la ontología. Debe existir al menos un buscador web en el tesoro de la ontología.
Postcondiciones	Un buscador web es eliminado del tesoro de la ontología.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesoro.2. En el menú contextual, elegir la opción de eliminar buscador web.3. Selecciona los buscadores a borrar.4. Pinchar en aceptar para confirmar.
Condiciones de fallo:	-

Tabla 86 - CU-023: Eliminar un buscador

Caso de uso: Buscador web	
Identificador:	CU-024
Descripción:	El usuario tiene la posibilidad de buscar un término del tesoro en la web en cualquier buscador web añadido al tesoro de la ontología.
Precondiciones:	Debe existir al menos un término seleccionado en el tesoro de la ontología. Debe existir al menos un buscador web en el tesoro de la ontología.
Postcondiciones	Visualización de resultados sobre ese término en el buscador web.
Escenario:	<ol style="list-style-type: none">1. Entra en el apartado de Tesoro.2. En el menú contextual, elegir la opción de buscadores y elegir el buscador.
Condiciones de fallo:	Problemas en el buscador web ajeno a la aplicación.

Tabla 87 - CU-024: Buscador web

3.2.3. Descripción de la Interacción de Objetos

Para describir los casos de interacción entre objetos se utilizan diagramas de secuencia. Ya que muchas funcionalidades son muy parecidas en proceso solo se expondrán los diagramas más significativos.

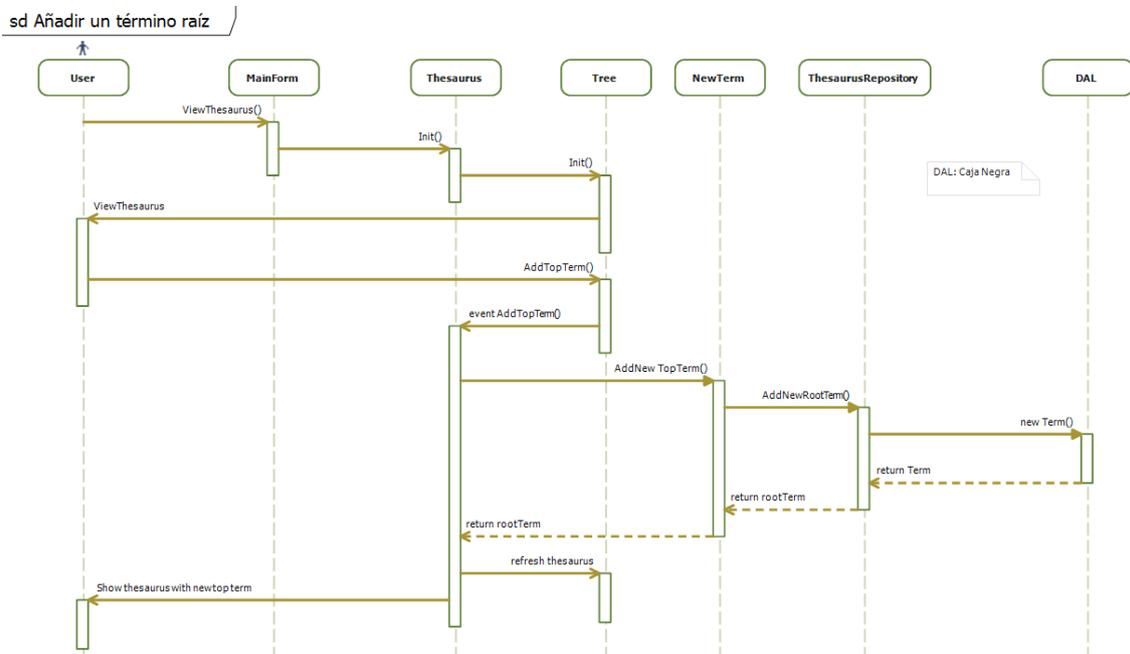


Ilustración 11- Diagrama de secuencia Añadir un nuevo término raíz

Todas las funcionalidades del menú contextual del tesoro son muy parecidas en interacción, simplemente cambian los formularios y funciones a los que se llama.

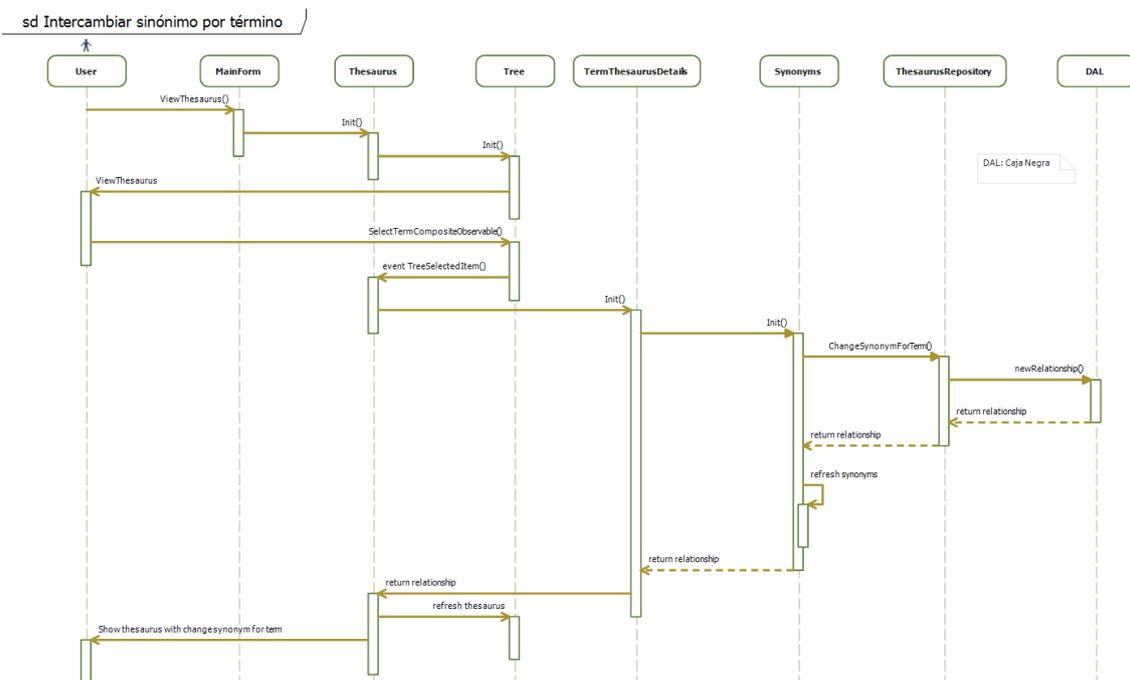


Ilustración 12 - Diagrama de secuencia Intercambiar un sinónimo por un término

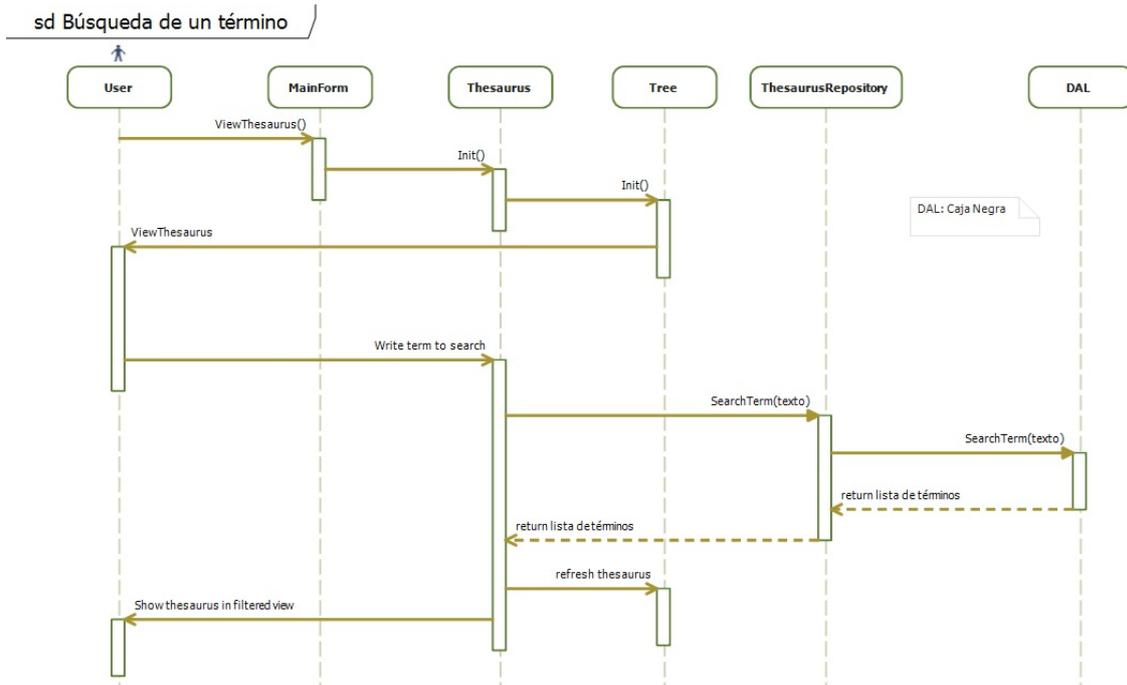


Ilustración 13 - Diagrama de secuencia Búsqueda de un término

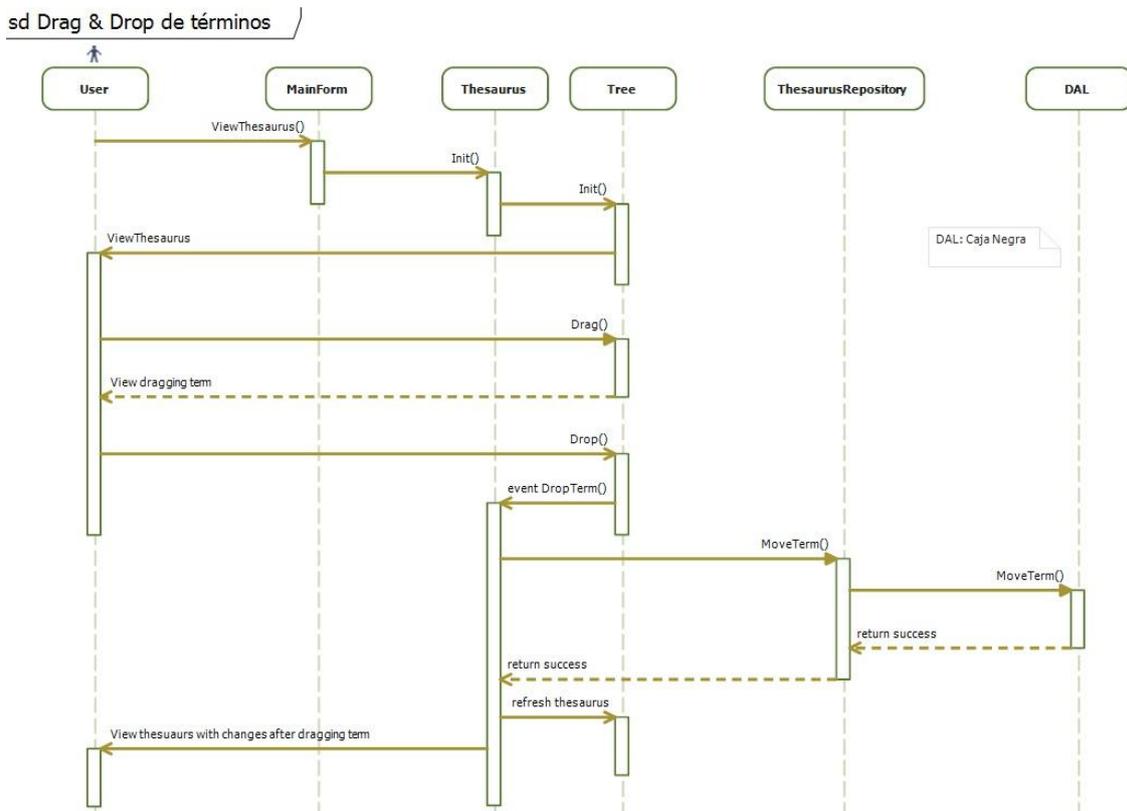


Ilustración 14 - Diagrama de secuencia Drag & Drop de términos

3.3. Identificación de subsistemas de análisis

Dada la complejidad del módulo, es conveniente no utilizar un único subsistema que se encargue de todas las funcionalidades. Por ello, se decide dividir el sistema en varios subsistemas. Los subsistemas de análisis propuestos son:

- **Formulario principal:** se encarga de mostrar la interfaz de usuario común a todos los subsistemas.
- **Tesauro:** se encarga de representar la funcionalidad de la parte de la aplicación relacionada con el tesauro de términos y relaciones. Este subsistema es muy grande por lo cual se decide dividir en subsistemas más pequeños:
 - **Términos:** se encarga de toda la gestión de términos en el tesauro.
 - **Relaciones:** se encarga de toda la gestión de relaciones en el tesauro
 - **Sinónimos:** se encarga de toda la gestión de sinónimos en el tesauro
 - **Otras funcionalidades:** se encarga de otras funcionalidades del tesauro, como es la importación, exportación entre otras.
- **Normalización:** se encarga de representar la funcionalidad de la parte de la aplicación relacionada con la normalización de términos.

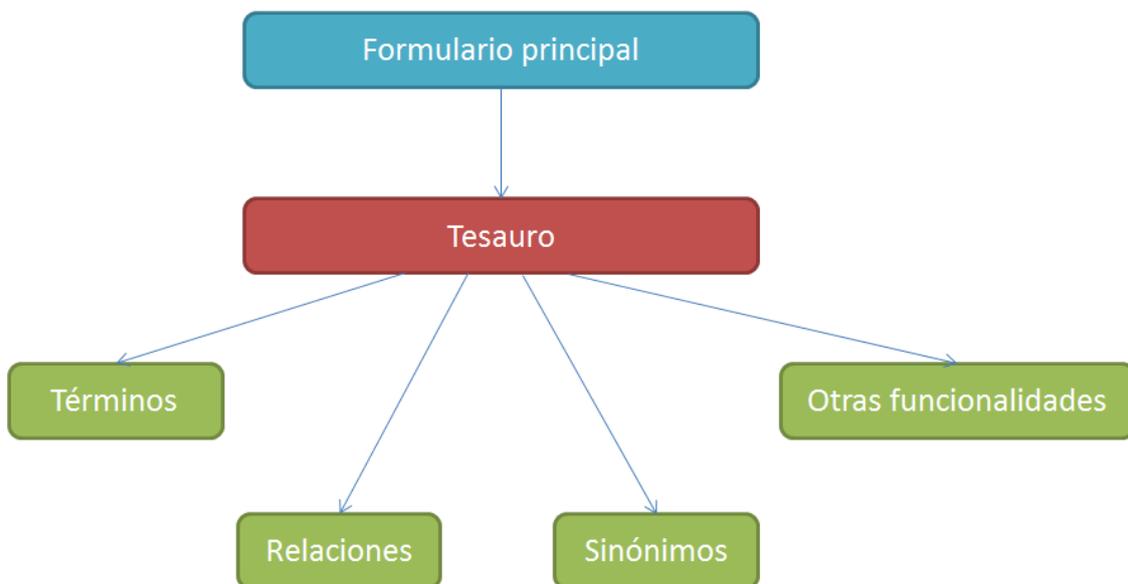


Ilustración 15 - Subsistemas de diseño

3.4. Análisis de clases

3.4.1. Identificación y especificación de clases

Una vez se han definido los casos de uso que van a formar parte de la aplicación, se identifican las clases u objetos necesarios para llevar a cabo la funcionalidad especificada en cada caso de uso. No se realiza un análisis y diseño detallado de cada clase, ya que se profundizará en detalles durante la fase de diseño.

Para alcanzar este objetivo, se utilizan tres tipos de clases:

- **Clases de entidad:** representan la información contenida en la base de datos.

- **Clases de apoyo a la interfaz:** representan la información rellena por la lógica de negocio y que sirve a la interfaz para representar la información.
- **Clases de interfaz:** describen la interacción entre el sistema y los actores.
- **Clases de control:** representan la secuencia de transacciones y la lógica de negocio.

En cada apartado se identifican las clases y posteriormente se especifican algunas de las propiedades que va a tener cada una de ellas. En esta sección se definen los atributos y funciones que debe tener cada clase, de forma que se satisfagan los requisitos especificados. Dichas funciones se detallan más durante la fase de diseño.

3.4.1.1. Clases de Entidad

Como clase de entidad se tiene una librería externa llamada CakeEngine, la cual me provee de la información necesaria. Las clases utilizadas de esta librería son:

- **Rshp:** contiene información acerca de las relaciones de la ontología.
- **Ke:** contiene información acerca de los elementos de las relaciones.
- **Artifact:** contiene información acerca de los artefactos de la ontología.
- **Term:** contiene información acerca de los términos de la ontología
- **Semantic:** contiene información acerca de las semánticas de la ontología.
- **TermTag:** contiene información acerca de las etiquetas sintácticas de la ontología.

Estas clases son caja negra para mí y se salen del ámbito de mi proyecto. Simplemente las comento aquí para saber en todo momento cual es mi fuente de datos.

3.4.1.2. Clases de apoyo a la interfaz

- **VocabularyDataSet:** contiene los atributos que forman un término.
- **GrammaticalDataSet:** contiene los atributos que forman una semántica.
- **RulesFamiliesDataSet:** contiene los atributos que forman una etiqueta sintáctica.
- **ArtifactDataSet:** contiene los atributos que forman un artefacto.

A continuación, se especifican las características de las clases:

VocabularyDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	Tabla de datos que contiene los atributos referidos a un término.
Operaciones:	-

Tabla 88 - Análisis de VocabularyDataSet

GrammaticalDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	Tabla de datos que contiene los atributos referidos a un buscador del tesouro.
Operaciones:	-

Tabla 89 - Análisis de GrammaticalDataSet

RulesFamiliesDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	Tabla de datos que contiene los atributos referidos a una etiqueta sintáctica.
Operaciones:	-

Tabla 90 - Análisis de RulesFamiliesDataSet

ArtifactDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	Tabla de datos que contiene los atributos referidos a un artefacto.
Operaciones:	-

Tabla 91 - Análisis de ArtifactDataSet

3.4.1.3. Clases de Interfaz

- **MainForm:** formulario de apertura, común para toda la aplicación.
- **Thesaurus:** contiene el árbol del tesouro y la información del término.
- **TermTreeView:** contiene el árbol del tesouro.
- **TermCompositeObservable:** contiene la información de los términos del tesouro.
- **TermThesaurusDetails:** contiene la información del término.
- **General:** muestra la información básica de un término en el tesouro.
- **Relationships:** muestra todas las relaciones de un término en el tesouro.
- **Synonyms:** muestra todos los sinónimos de un término.

- **AdvancedTerm:** Muestra detalles específicos de un término.
- **Import:** muestra la importación del tesouro.
- **Export:** muestra la exportación del tesouro.

A continuación, se especifican las características de las clases:

Thesaurus	
Responsabilidades:	Interfaz que contiene toda la funcionalidad que viene desde el árbol y los detalles del término. Funciona como un controlador.
Atributos:	Objeto que contiene los términos raíces del árbol. Objeto que contiene el término seleccionado del árbol. Diccionario que contiene la lista de términos del tesouro. Diccionario que controla los formularios de detalle de términos abiertos.
Operaciones:	Funciones que pertenecen al menú contextual del tesouro. Eventos que capturan la funcionalidad perteneciente al menú contextual del tesouro. Funciones que pertenecen a la ventana de detalle de los términos. Eventos que capturan la funcionalidad perteneciente a la ventana de detalle de los términos. Funciones de refresco del tesouro.

Tabla 92 - Análisis de Thesaurus

TermTreeView	
Responsabilidades:	Interfaz para mostrar el tesouro de la ontología.
Atributos:	Objeto que contiene el término seleccionado del árbol. Diccionario que contiene la lista de términos del tesouro.
Operaciones:	Carga del menú contextual de los términos. Captura los eventos pertenecientes al menú contextual del tesouro. Propaga los eventos a la clase controladora. Se encarga de los eventos del drag and drop.

Tabla 93 - Análisis de TermTreeView

TermCompositeObservable	
Responsabilidades:	Clase que representa los términos del tesouro.
Atributos:	Objeto que contiene el término. Objeto que contiene el padre visual. Diccionario que contiene la lista de términos del tesouro. Colección de hijos de un término.
Operaciones:	-

Tabla 94 – Análisis de TermCompositeObservable

TermThesaurusDetails	
Responsabilidades:	Interfaz que contiene las pestañas de los detalles del término.
Atributos:	Objeto con la información básica del término. Objeto con de información de relaciones del término. Objeto con la información de sinónimos del término. Objeto con la información avanzada del término.
Operaciones:	Muestra la información de un término en el tesouro. Muestra la información de las relaciones de un término en el tesouro. Muestra la información de los sinónimos de un término en el tesouro. Acepta los cambios realizados en las diferentes pestañas y propaga eventos a la clase controladora. Validar que los campos introducidos son válidos.

Tabla 95 - Análisis de TermThesaurusDetails

General	
Responsabilidades:	Interfaz que contiene la información básica del término.
Atributos:	Objeto que contiene la tabla con la miga de pan del término.
Operaciones:	Muestra la información de un término en el tesoro. Acepta los cambios realizados en las diferentes pestañas y propaga eventos a la clase controladora. Validar que los campos introducidos son válidos.

Tabla 96 - Análisis de General

Relationships	
Responsabilidades:	Interfaz que contiene la información de las relaciones de un término.
Atributos:	Objeto que contiene la tabla con la relaciones del término.
Operaciones:	Muestra la información de las relaciones de un término en el tesoro. Añadir una relación al tesoro. Eliminar una relación del tesoro. Mostrar el otro participante de la relación. Propagar los eventos necesarios en la clase controladora para refrescar.

Tabla 97 - Análisis de Relationships

Synonyms	
Responsabilidades:	Interfaz que contiene la información de los sinónimos de un término.
Atributos:	Objeto que contiene la tabla con los sinónimos del término.
Operaciones:	Muestra la información de los sinónimos de un término en el tesouro. Añadir un sinónimo al tesouro. Eliminar un sinónimo del tesouro. Intercambiar un sinónimo por un término del tesouro. Propagar los eventos necesarios en la clase controladora para refrescar.

Tabla 98 - Analisis de Synonyms

AdvancedTerm	
Responsabilidades:	Interfaz que contiene la información avanzada del término del tesouro.
Atributos:	-
Operaciones:	Muestra la información detallada de un término en el tesouro. Modificar etiqueta sintáctica. Modificar semántica. Modificar lenguaje. Modificar otros atributos del término.

Tabla 99 - Análisis de AdvancedTerm

Export	
Responsabilidades:	Interfaz que contiene la información para exportar un tesouro.
Atributos:	Objeto de la BLL que permite acceder a la lógica de negocio del tesouro.
Operaciones:	Exportación del tesouro.

Tabla 100 - Análisis de Export

Import	
Responsabilidades:	Interfaz que contiene la información para importar un tesouro.
Atributos:	Objeto de la BLL que permite acceder a la lógica de negocio del tesouro.
Operaciones:	Importación del tesouro.

Tabla 101 - Análisis de Import

3.4.1.4. Clases de control

- **VocabularyRepository:** contiene la lógica de negocio de los términos.
- **GrammaticalRepository:** contiene la lógica de negocio de las semánticas.
- **RulesFamiliesRepository:** contiene la lógica de negocio de las etiquetas sintácticas.
- **ArtifactRepository:** contiene la lógica de negocio de los artefactos.
- **ThesaurusRepository:** contiene la lógica de negocio de todas las operaciones del tesouro.

A continuación, se especifican las características de las clases:

VocabularyRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de términos.
Atributos:	-
Operaciones:	Función que devuelve una tabla de datos con todos los términos del tesouro de la ontología. Función que devuelve una tabla de datos con todos los términos del tesouro que coincidan con los criterios introducidos por el usuario en el filtro de búsqueda.

Tabla 102 - Análisis de VocabularyRepository

GrammaticalRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de semánticas.
Atributos:	-
Operaciones:	Función que devuelve una tabla de datos con todas las semánticas del tesauro de la ontología. Función que devuelve una tabla de datos con todas las semánticas del tesauro que coincidan con los criterios introducidos por el usuario en el filtro de búsqueda. Función que inserta una semántica en la ontología.

Tabla 103 - Análisis de GrammaticalRepository

RulesFamiliesRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de etiquetas sintácticas.
Atributos:	-
Operaciones:	Función que devuelve una tabla de datos con todas las etiquetas sintácticas del tesauro de la ontología. Función que devuelve una tabla de datos con todas las etiquetas sintácticas del tesauro que coincidan con los criterios introducidos por el usuario en el filtro de búsqueda. Función que inserta una etiqueta sintáctica en la ontología.

Tabla 104 - Análisis de RulesFamiliesRepository

ArtifactRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de artefactos.
Atributos:	-
Operaciones:	Función que devuelve una tabla de datos con todos los artefactos de la ontología. Función que devuelve el artefacto correspondiente al tesauro de la ontología.

Tabla 105 - Análisis de ArtifactRepository

ThesaurusRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio de todas las operaciones del tesouro.
Atributos:	Lista de términos del tesouro. Lista de sinónimos del tesouro. Lista de términos raíces del tesouro. Artefacto Semántica
Operaciones:	Funciones relacionadas con las relaciones. Funciones relacionadas con los elementos de las relaciones. Funciones relacionadas con los sinónimos. Funciones relacionadas con los términos del tesouro. Funciones relacionadas con los términos raíces del tesouro. Funciones relacionadas con la obtención de hijos del tesouro. Funciones relacionadas con la obtención de padres del tesouro. Funciones relacionadas con la importación/ exportación del tesouro. Funciones relacionadas con la búsqueda web del tesouro. Funciones relacionadas con el uso de tooltips en el tesouro.

Tabla 106 - Análisis de ThesaurusRepository

3.4.2. Identificación de asociaciones y agregaciones

En esta sección se definen las asociaciones y agregaciones más representativas para cada clase.

Por un lado, una **agregación** es un tipo de asociación en la que un objeto forma parte del otro, es decir, el todo se relaciona con sus partes. También se denomina como relación “parte de”.

Por otro lado, por **asociación** se entiende una relación entre instancias de clases. Por lo tanto los objetos de la clase de un extremo de la asociación deben conocer los objetos del otro extremo de la relación.

Relación $\leftarrow \rightarrow$ Artefacto	
Tipo:	Asociación
Clases (cardinalidad):	Relación (*) Artefacto (1)
Descripción:	Una relación está asociada a un solo artefacto, y cada artefacto puede estar asociado a 0 o más relaciones.

Tabla 107 - Asociación Relación – Artefacto

Relación $\leftarrow \rightarrow$ Semántica	
Tipo:	Asociación
Clases (cardinalidad):	Relación (*) Semántica (1)
Descripción:	Una relación está asociada a una sola semántica, y cada semántica puede estar asociado a 0 o más relaciones.

Tabla 108 - Asociación Relación - Semántica

KnowledgeElement $\leftarrow \rightarrow$ Artefacto	
Tipo:	Asociación
Clases (cardinalidad):	KnowledgeElement (*) Artefacto (1)
Descripción:	Un KnowledgeElement está asociada a una solo artefacto, y cada artefacto puede estar asociado a 0 o más KnowledgeElement.

Tabla 109 - Asociación KnowledgeElement – Artefacto

KnowledgeElement $\leftarrow \rightarrow$ Vocabulario	
Tipo:	Asociación
Clases (cardinalidad):	KnowledgeElement (1) Término (1)
Descripción:	Una KnowledgeElement está asociado a una solo término, y cada término puede estar asociado a 1 KnowledgeElement.

Tabla 110 - Asociación KnowledgeElement – Vocabulario

Vocabulario $\leftarrow \rightarrow$ Etiqueta sintáctica	
Tipo:	Asociación
Clases (cardinalidad):	Término (*) Etiqueta sintáctica (1)
Descripción:	Un término está asociado a una sola etiqueta sintáctica, y cada etiqueta sintáctica puede estar asociada a 0 o más términos.

Tabla 111 - Asociación Vocabulario – Etiqueta sintáctica

Vocabulario $\leftarrow \rightarrow$ Semántica	
Tipo:	Asociación
Clases (cardinalidad):	Término (*) Semántica (0...1)
Descripción:	Una término está asociado a ninguna o una sola semántica, y cada semántica puede estar asociada a 0 o más términos.

Tabla 112 - Asociación Vocabulario – Semántica

KnowledgeElement ← → Relación	
Tipo:	Agregación
Clases (cardinalidad):	KnowledgeElement (1...2) Relación (0...*)
Descripción:	Un KnowledgeElement aparece en ninguna o muchas relaciones, y cada relación está formada por 1 o 2 KnowledgeElement.

Tabla 113 - Agregación KnowledgeElement – Relación

3.5. Elaboración del modelo de datos

La definición del modelo de datos puede entenderse como una actividad más de diseño que de análisis. Pero con el fin de facilitar la comprensión del problema en esta fase, se definen las **entidades** que van a formar parte del sistema de información.

Se parte de un modelo de datos ya existente, elaborado previamente para construir un modelo basado en relaciones, compuesto por toda clase de elementos (textuales, modelos de diseño, código, bases de datos, etc) destinado a generalizar su gestión en aplicaciones de ordenador. [12] [13].

No se utiliza el modelo de datos completo, ya que es enorme y hay gran cantidad de tablas que no se utilizan durante este módulo. De esta forma, se han definido las siguientes entidades:

Las tres primeras entidades son fundamentales en el desarrollo de mi proyecto, las otras son necesarias i utilizadas para algunas funcionalidades del módulo a desarrollar.

- **Artifacts:** Representa la información de salida de nuestra ontología, es decir, nuestros asserts. En mi modulo se usa para representar el tesouro.
- **Knowledge_Elements:** Representa a los términos dentro de un artefacto (en nuestro caso el tesouro) o relación del tesouro.
 - El atributo *Cod_Artifact* hace referencia al artefacto de la ontología.
 - El atributo *Cod_RSHP* hace referencia a la relación de la ontología.
- **RSHP:** Representa la relaciones del tesouro.
 - El atributo *Cod_Artifact* hace referencia al artefacto de la ontología.
 - El atributo *Type* hace referencia a una semántica de la ontología.
- **Vocabulary:** Representa la información de un término de la ontología. Es la entidad fundamental de la aplicación [KnowledgeMANAGER](#). Contiene una lista muy amplia de atributos debido a su uso en toda la aplicación.
 - El atributo *Type* hace referencia a una etiqueta sintáctica de la ontología.

- El atributo *Grramtical* hace referencia a una semántica de la ontología.

- **Grammatical:** Representa la información de una semántica de la ontología.
- **Rules_Families:** Representa la información de una etiqueta sintáctica de la ontología.

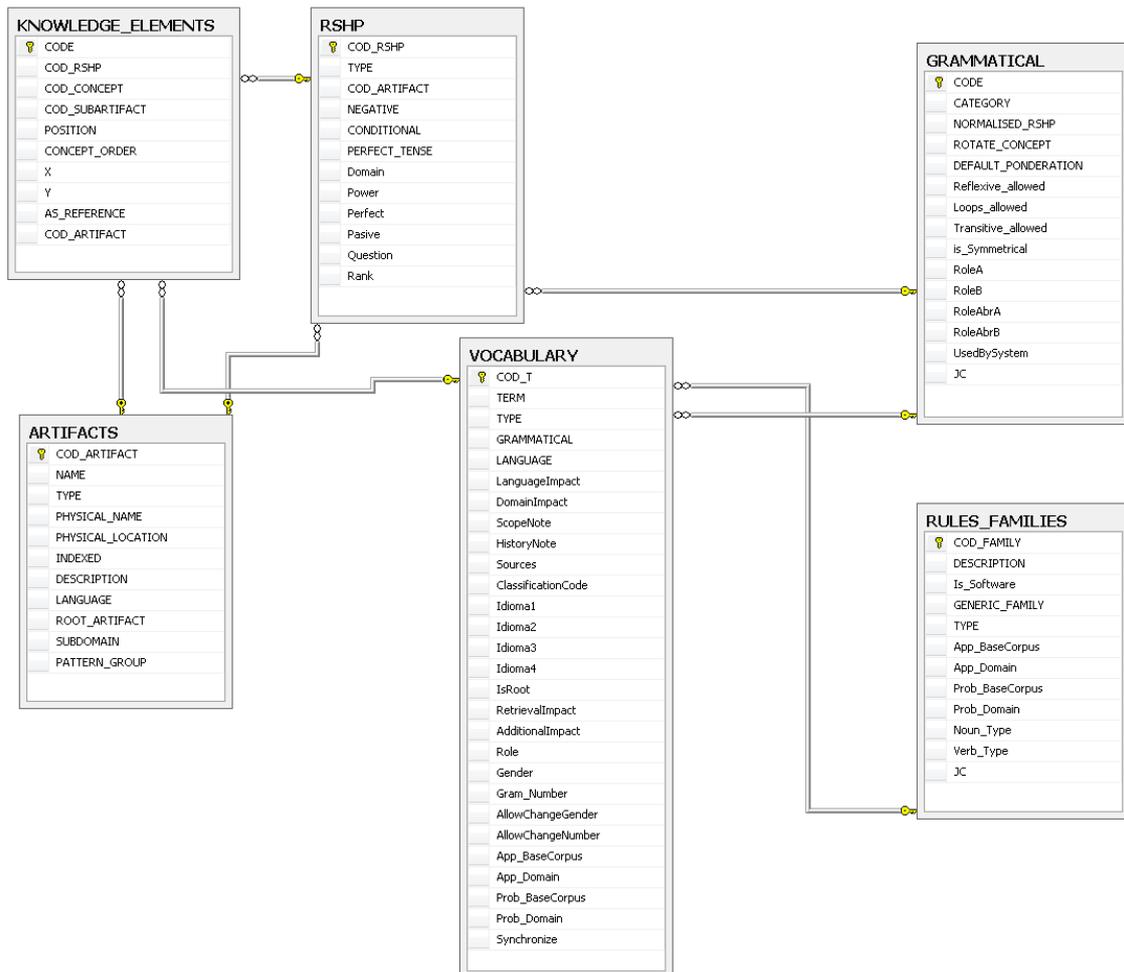


Ilustración 16 - Modelo de datos

3.5.1. Especificación necesidades migración de datos y carga inicial

En cuanto a la carga inicial de datos, hay que tener en cuenta una serie de requisitos mínimos obligatorios para que la aplicación pueda iniciarse satisfactoriamente. Aunque ya ha sido definido en la fase de especificación de requisitos, se recuerda en este capítulo.

- La base de datos debe contener la etiqueta sintáctica NOUN y VERB.
- La base de datos debe contener el artefacto DOMAIN.

A la hora de realizar altas, bajas, modificaciones o consultas de datos desde la interfaz de la aplicación, no se altera la estructura de ninguna tabla de la base de datos.

No supone ningún problema para los miembros del equipo de desarrollo ni a la hora de realizar el despliegue de la aplicación, ya que junto con el código fuente entregable se incluye un fichero de restauración de un ejemplo de base de datos válida.

3.6. Definición de interfaces de usuario

En este apartado se describen las principales características de la interfaz de usuario. Se sigue la misma interfaz que en el resto del [KnowledgeMANAGER](#), por lo que no se especifican en profundidad los detalles de la interfaz.

3.6.1 Especificación de Principios Generales de la Interfaz

Una de las principales características de la capa de presentación de la aplicación [KnowledgeMANAGER](#) es la reutilización de los *Windows Forms* y los *User Control*.

Existe uno genérico de cada tipo, con las características comunes, y de ellos heredan el resto de clases de interfaz.

3.6.2 Identificación de Perfiles y Diálogos

En este apartado se adjunta gráficamente cómo debe ser cada diálogo que se muestra al usuario, según del tipo que sea.

Cuando el sistema necesita confirmación por parte del usuario, se lanza un diálogo de solicitud de confirmación.

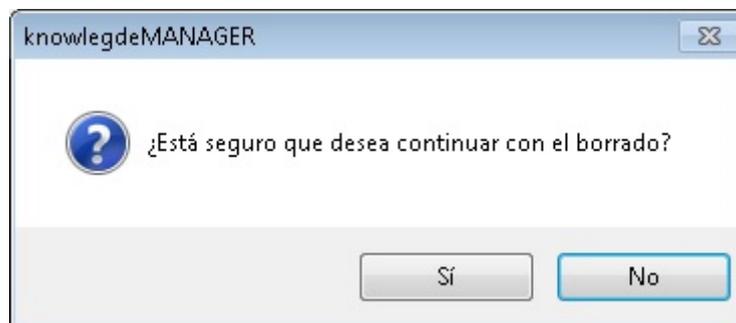


Ilustración 17 - Solicitud de confirmación

Cuando el sistema ha intentado llevar a cabo una operación, pero se produjo un error durante el proceso, se lanza un mensaje de error descriptivo.

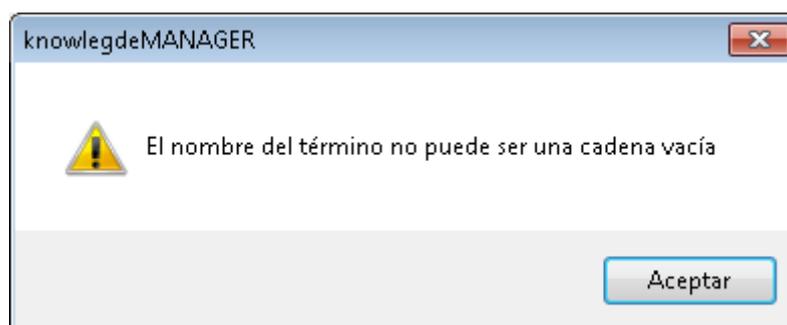


Ilustración 18 - Mensaje de error

3.6.3 Especificación Formatos Individuales de la Interfaz de Pantalla

En este apartado se indicaran los elementos visuales característicos dentro de la interfaz de usuario.

 Tesoro	 Términos	 Relaciones
 Sinónimos	 Importación	 Exportación

Tabla 114 - Iconos característicos de la interfaz

3.6.4 Especificación del Comportamiento Dinámico de la Interfaz

Existen algunas situaciones en que estando en un determinado punto del tesoro es necesario acceder a un objeto de otro punto de la aplicación distinto. Por ejemplo, si contemplamos la opción de buscador avanzado de un término del tesoro, se tiene la posibilidad de elegir una etiqueta sintáctica o semántica para ese término.

Es por ello que se han utilizado **formularios selectores creados** que emulan la funcionalidad de un *ComboBox* (elegir una opción) pero incluyen un filtro de búsqueda, para que encontrar lo que se busca sea más fácil. Existe un selector de términos, de etiquetas sintácticas y de semánticas, en mi módulo se pueden utilizar los dos últimos.

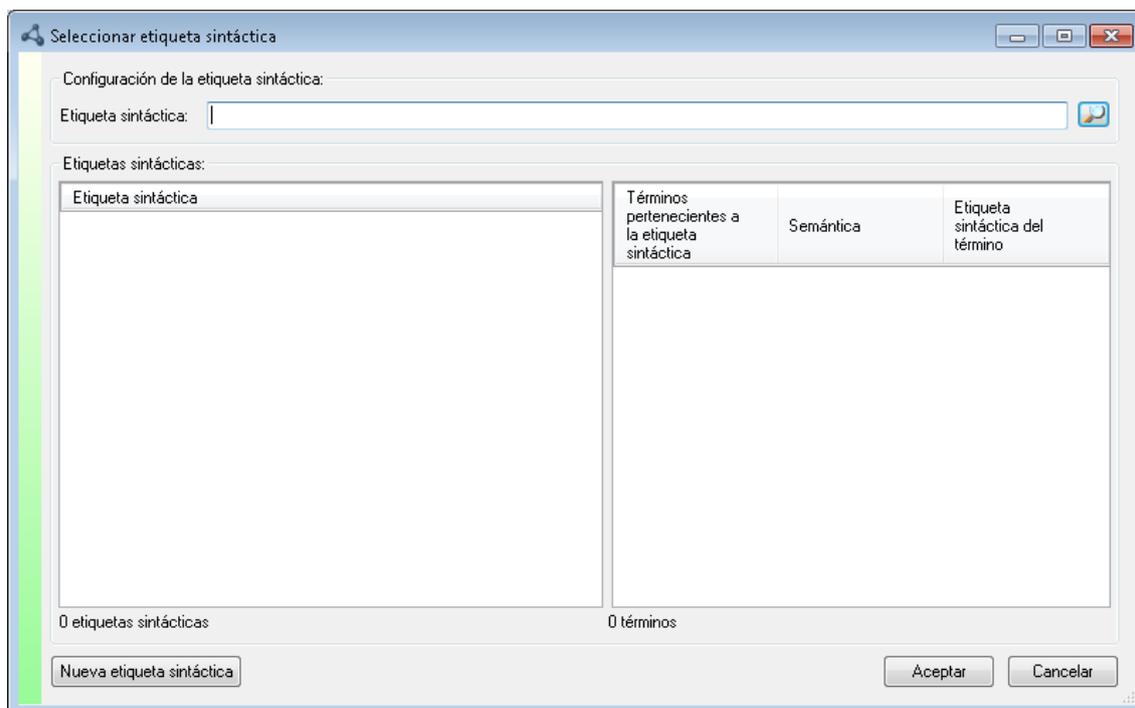


Ilustración 19 - Selector de etiqueta sintáctica

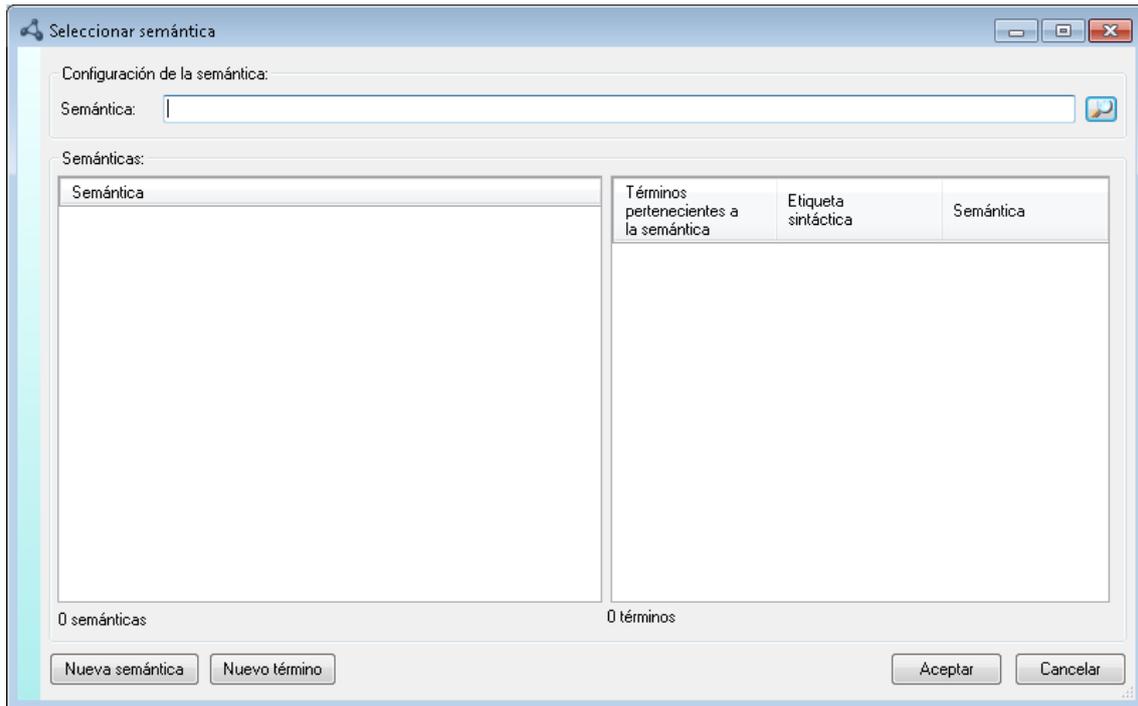


Ilustración 20 - Selector de semántica

3.6.5 Especificación de Formatos de Impresión

No aplica. N/A

3.7. Análisis de consistencia y especificación de requisitos

Una parte de todo documento de análisis consiste en realizar un propio análisis sobre él. Los requisitos de un sistema deben ser **trazables**, es decir, se pueden identificar todos los elementos del producto relacionados con ese requisito. Así se mantiene la consistencia entre los distintos elementos de un proyecto.

Una matriz de trazabilidad es una buena opción para controlar las dependencias de cada requisito. Pero por temas de espacio y dimensiones de la tabla, se considera que la mejor opción es añadir un campo más, **Trazabilidad**, en cada tabla de requisitos, de forma que se indican los identificadores de los requisitos con que traza cada uno de los requisitos.

3.8. Especificación del plan de pruebas

3.8.1. Definición del alcance de pruebas

El sistema de información cuenta con un plan de pruebas que garantiza que el sistema cumple con las necesidades requeridas por el usuario, además de tener la calidad necesaria.

En este nivel de análisis de la aplicación se desarrollan las pruebas de aceptación, que certifican a alto nivel que la aplicación es capaz de validar los requisitos especificados por el cliente.

Sin embargo, estas pruebas de aceptación no son lo suficiente detalladas como para decir que el sistema sea fiable 100%. Es por ello que durante la fase de diseño se añadirán las pruebas unitarias realizadas, con el fin de obtener un producto funcional y lo más fiable posible.

3.8.2. Definición de requisitos para el entorno de pruebas

El entorno de pruebas mínimo para el desarrollo es equivalente al utilizado para el desarrollo del proyecto, ya que es el entorno mínimo donde se ejecutará la aplicación una vez acabe la fase de desarrollo. Dichos requisitos son:

Requisitos software:

- Microsoft Framework 4.0
- Visual Studio 2010 Professional
- SQL Server 2008

Requisitos hardware:

- Procesador 1.5GHz mínimo.
- 1024 MB de memoria RAM
- 128 GB totales de disco duro

3.8.2. Definición de las pruebas de aceptación del sistema

Para una mejor comprensión de las pruebas se dividen según los subsistemas planteados en este mismo apartado de análisis.

3.8.2.1. Términos

PA-001: Añadir un término raíz	
Descripción:	Se comprueba que se puede añadir un término raíz al tesouro de la ontología. <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Hacer click con el botón derecho en el tesouro y elegir la opción de 'Añadir un nuevo término raíz'.3. Introducir los siguientes datos en el nuevo formulario:<ul style="list-style-type: none">• Nombre: TerminoRaiz1• Clasificador: -• Nota de alcance: -4. Hacer click en el botón Aceptar.
Resultado:	Tras hacer click en el botón Aceptar, en el tesouro de los términos aparece el nuevo término raíz 'TerminoRaiz1' que se acaba de crear.
Requisitos relacionados:	RSF - 002, RSF - 003, RSF - 004

Tabla 115 - PA-001: Añadir un término raíz

PA-002: Ver detalles de un término	
Descripción:	Se comprueba que se puede ver los detalles de un término del tesouro de la ontología. 1. Acceder al apartado de Tesouro. 2. Hacer click con el botón izquierdo sobre el término previamente creado.
Resultado:	Tras hacer click en el término, se visualiza en la pantalla de detalles de la derecha la información del término seleccionado.
Requisitos relacionados:	RSF - 002, RSF - 033, RSF - 0034

Tabla 116 - PA-002: Ver detalles de un término

PA-003: Modificar un término	
Descripción:	Se comprueba que se puede modificar un término del tesouro de la ontología. 1. Acceder al apartado de Tesouro. 2. Hacer click con el botón derecho en el tesouro sobre el término creado previamente y elegir la opción de 'Editar / Mostrar información del término'. 3. Introducir los siguientes datos en el nuevo formulario: <ul style="list-style-type: none">• Nombre: TerminoRaiz1_Modificado• Clasificador: -• Nota de alcance: -• Nota histórica: termino modificado
Resultado:	Tras hacer click en el botón Aceptar, en el tesouro de los términos aparece el término raíz con el nombre modificado a 'TerminoRaiz1_Modificado' que se acaba de editar.
Requisitos relacionados:	RSF - 002, RSF - 003, RSF - 010, RSF - 033, RSF - 034

Tabla 117 - PA-003: Modificar un término

PA-004: Comprobar campos obligatorios de un término	
Descripción:	<p>Se comprueba que los campos obligatorios a la hora de rellenar la información de un término son correctos.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesaurus.2. Hacer click con el botón derecho en el tesaurus y elegir la opción de 'Añadir nuevo término raíz'.3. Introducir los siguientes datos en el nuevo formulario:<ul style="list-style-type: none">• Nombre: -• Clasificador: -• Nota de alcance: -• Nota histórica: termino modificado4. Hacer click en el botón Aceptar.
Resultado:	<p>Tras hacer click en el botón Aceptar, debe aparecer un mensaje de error sobre la interfaz avisando que el campo 'Nombre' no puede dejarse en blanco.</p>
Requisitos relacionados:	<p>RSF - 002, RSF - 003, RSF - 004, RSF - 033, RSF - 034</p>

Tabla 118 - PA-004: Comprobar campos obligatorios de un término

PA-005: Buscar un término mediante filtros de búsqueda simple	
Descripción:	<p>Se comprueba que el filtro de búsqueda simple funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Escribir en el campo de búsqueda que está encima del tesouro el nombre:<ul style="list-style-type: none">• Nombre: TerminoQueNoExiste3. Hacer click en el botón de la lupa que corresponde con buscar: Resultado: no se debe mostrar ningún resultado ya que es un término que no existe en el tesouro.4. Borrar el campo de búsqueda introducido con anterioridad pulsando esta vez al aspa roja.5. Escribir en el campo de búsqueda que está encima del tesouro el nombre:<ul style="list-style-type: none">• Nombre: TerminoRaiz1_Modificado6. Hacer click en el botón de la lupa que corresponde con buscar: Resultado: se debe mostrar un único resultado cuyo nombre es 'TerminoRaiz1_Modificado '.7. Para finalizar pulsa en el botón de la casa para volver a la vista del tesouro inicial.
Resultado:	Los resultados se han comentado en la descripción de esta prueba.
Requisitos relacionados:	RSF - 027, RSF - 028, RSF - 029, RSF - 030

Tabla 119 - PA-005: Buscar un término mediante filtros de búsqueda simple

PA-006: Buscar un término mediante filtros de búsqueda avanzado	
Descripción:	<p>Se comprueba que el filtro de búsqueda avanzado funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar sobre la lupa que se encuentra en la parte superior de la ventana.2. Escribir en el campo de búsqueda del formulario emergente el nombre:<ul style="list-style-type: none">• Nombre: TerminoQueNoExiste3. Hacer click en el botón de buscar: Resultado: no se debe mostrar ningún resultado ya que es un término que no existe en el tesouro.4. Borrar el campo de búsqueda introducido con anterioridad.5. Escribir en el campo de búsqueda que está encima del tesouro el nombre:<ul style="list-style-type: none">• Nombre: TerminoRaiz1_Modificado6. Pinchar en el campo de etiqueta sintactica y en el formulario emergente escribir en el campo:<ul style="list-style-type: none">• Etiqueta sintáctica: Nombre7. Hacer click en Aceptar para volver al formulario de búsqueda avanzada de término.8. Hacer click en el botón de buscar: Resultado: se debe mostrar un único resultado cuyo nombre es 'TerminoRaiz1_Modificado'.
Resultado:	Los resultados se han comentado en la descripción de esta prueba.
Requisitos relacionados:	RSF - 031

Tabla 120 - PA-006: Buscar un término mediante filtros de búsqueda avanzado

PA-007: Eliminar término	
Descripción:	Se comprueba que eliminar un término funciona correctamente. 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'TérminoRaiz1_Modificado' y elegir la opción de 'Eliminar término'. 3. En el formulario emergente saldrán distintas opciones de borrado. En este caso al no tener hijos solo saldrá marcada la opción de 'Borrar el término raíz'. 4. Pinchar en el botón Aceptar. 5. Pinchar el botón Aceptar del mensaje de confirmación.
Resultado:	Tras confirmar el borrado, no debe existir ningún término en el tesouro de términos.
Requisitos relacionados:	RSF - 003, RSF - 011, RSF - 013

Tabla 121 - PA-007: Eliminar término

Para las siguientes pruebas se tendrá el siguiente tesouro creado:

.Raiz1
..Hijo11
..Hijo12
.Raiz2
..Hijo21
...Hijo211
.Raiz3

NOTA: Los puntos significan el nivel en el que se encuentra el término

PA-008: Mover un término a la raíz	
Descripción:	Se comprueba que mover un término a la raíz funciona correctamente. 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'Hijo211' y elegir la opción de 'Mover un término a la raíz'.
Resultado:	Se debe visualizar el término 'Hijo211' como un término raíz solo.
Requisitos relacionados:	RSF - 003, RSF - 018

Tabla 122 - PA-008: Mover un término a la raíz

PA-009: Drag & Drop de un término	
Descripción:	<p>Se comprueba que drag & drop de un término funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesauro.2. Pinchar con el botón izquierdo sobre el término 'Hijo11' y arrastrar con la tecla control pulsada hasta el término 'Raiz3'. Resultado: Se debe visualizar el término 'Hijo11' como hijo del término 'Raiz1' y 'Raiz3'.3. Pinchar con el botón izquierdo sobre el término 'Hijo12' y arrastrar hasta el término 'Raiz3'. Resultado: Se debe visualizar el término 'Hijo12' como hijo del término 'Raiz3' únicamente.
Resultado:	Los resultados se muestran en la propia descripción de la prueba.
Requisitos relacionados:	RSF - 003, RSF - 049

Tabla 123 - PA-009: Drag & Drop de un término

PA-010: Mezclar un término con otro término	
Descripción:	<p>Se comprueba que mezclar un término con otro término funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesauro.2. Pinchar con el botón derecho sobre el término 'Raiz2' y elegir la opción de 'Mezclar el término con otro término'.3. En el formulario emergente, pinchar en el campo del segundo término y en el formulario emergente escribir el nombre:<ul style="list-style-type: none">• Nombre: 'Raiz2'4. Pulsar en buscar, seleccionar el término y pulsar en Aceptar para volver al formulario anterior.5. Elegir como término único el término 'Raiz1' y pulsar en aceptar.6. Confirmar el proceso pulsando Aceptar.
Resultado:	El resultado visual debe ser que 'Raiz1' tiene como hijos 'Hijo11', 'Hijo 12' e 'Hijo21' que a su vez tiene como hijo a 'Hijo211'.
Requisitos relacionados:	RSF - 003, RSF - 019

Tabla 124 - PA-010: Mezclar un término con otro término

PA-011: Expandir término	
Descripción:	<p>Se comprueba que expandir un término funciona correctamente.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'Raiz2' y elegir la opción de 'Expandir todo'.
Resultado:	El resultado visual debe ser que se vean los hijos de 'Raiz2' y el hijo de 'Hijo21'.
Requisitos relacionados:	RSF - 003, RSF - 020

Tabla 125 - PA-011: Expandir término

3.8.2.2. Relaciones

PA-012: Añadir una nueva relación	
Descripción:	<p>Se comprueba que añadir una nueva relación funciona correctamente.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho y elegir la opción de 'Añadir una nueva relación'. 3. Rellenar en un nuevo formulario los términos que forman parte de la relación y el tipo de relación. Para ello: <ol style="list-style-type: none"> 3.1. Pinchar en buscar y se abre un nuevo formulario. 3.2. Escribir el término 'Raiz3'. 3.3. Selecciona el término y pulsa en aceptar. 3.4. Repetir el proceso con el tipo de relación (en este caso se elige 'BroaderTerm', significa padre de) y el segundo término de la relación 'Hijo11'. 4. Pinchar en aceptar para confirmar.
Resultado:	El resultado visual debe ser que 'Raiz3' tiene un nuevo hijo que es 'Hijo11'.
Requisitos relacionados:	RSF - 002, RSF - 003, RSF - 005, RSF - 036, RSF - 037

Tabla 126- PA-012: Añadir una nueva relación

PA-013: Añadir una nueva asociación	
Descripción:	<p>Se comprueba que añadir una nueva relación de asociación funciona correctamente.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'Raiz3' y elegir la opción de 'Añadir una asociación'. 3. Rellenar en un nuevo formulario el término asociado a la relación. Para ello: <ol style="list-style-type: none"> 3.1. Pinchar en buscar y se abre un nuevo formulario. 3.2. Escribir el término 'Hijo21'. 3.3. Selecciona el término y pulsa en aceptar. 4. Pinchar en aceptar para confirmar.
Resultado:	El resultado visual debe ser que en relaciones se ha añadido una fila con la nueva relación de asociación entre los términos.
Requisitos relacionados:	RSF - 003, RSF – 008, RSF - 036

Tabla 127 - PA-013: Añadir una nueva asociación

PA-014: Añadir un nuevo hijo	
Descripción:	<p>Se comprueba que añadir un nuevo hijo funciona correctamente.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'Raiz3' y elegir la opción de 'Añadir un hijo'. 3. Rellenar en un nuevo formulario los datos del nuevo término: <ul style="list-style-type: none"> • Nombre: Raiz3_NuevoHijo • Clasificador: - • Nota de alcance: - 4. Pinchar en aceptar para confirmar.
Resultado:	<p>En relaciones se ha añadido una fila con la nueva relación de jerarquía entre los términos.</p> <p>En el árbol se visualiza que el término 'Raiz3' tiene un hijo llamado 'Raiz3_NuevoHijo'.</p>
Requisitos relacionados:	RSF - 002, RSF – 003, RSF – 006, RSF – 036

Tabla 128 - PA-014: Añadir un nuevo hijo

PA-015: Eliminar una relación	
Descripción:	Se comprueba que eliminar una relación funciona correctamente. <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Seleccionar el término 'Raiz3' del tesouro y pinchar en 'Relaciones'.3. Pinchar con el botón derecho en la relación de jerarquía que tiene con el término 'Raiz3_NuevoHijo' y elegir 'Eliminar relación'.4. Confirmar el borrado.
Resultado:	El resultado visual debe ser que en relaciones se ha eliminado una fila con la relación entre los términos. En el árbol 'Raiz3' ya no tiene hijos.
Requisitos relacionados:	RSF - 002, RSF - 003, RSF - 036, RSF - 038

Tabla 129 - PA-015: Eliminar una relación

PA-016: Comprobar campos obligatorios de una relación	
Descripción:	Se comprueba que los campos obligatorios a la hora de rellenar la información de una relación son correctos. <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón derecho sobre el término 'Raiz3' y elegir la opción de 'Añadir otra relación'.3. Rellenar en un nuevo formulario los datos del nuevo término:<ul style="list-style-type: none">• Tipo de relación: -• Término2: 'Raiz2'4. Pinchar en aceptar y se visualiza un mensaje de error en el cual nos dice que el tipo de relación debe estar relleno.
Resultado:	El resultado se describe en la descripción de la prueba.
Requisitos relacionados:	RSF - 003, RSF - 009

Tabla 130 - PA-016: Comprobar campos obligatorios de una relación

3.8.2.3. Sinónimos

PA-017: Añadir un sinónimo	
Descripción:	<p>Se comprueba que añadir un nuevo sinónimo funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón derecho sobre el término 'Raiz3' y elegir la opción de 'Añadir un sinónimo'.3. Rellenar en un nuevo formulario el término sinónimo. Para ello:<ol style="list-style-type: none">3.1. Escribir en el campo de texto 'Sinónimo de Raiz3'.4. Pinchar en aceptar para confirmar.
Resultado:	<p>El resultado visual debe ser que en Sinónimos se ha añadido una fila con el sinónimo creado.</p> <p>En Relaciones también se ha añadido una fila con la nueva relación de sinonimia.</p>
Requisitos relacionados:	RSF - 003, RSF – 007, RSF – 036, RSF – 040, RSF - 041

Tabla 131 - PA-017: Añadir un sinónimo

PA-018: Modificar un sinónimo	
Descripción:	<p>Se comprueba que modificar un sinónimo funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón izquierdo sobre el término 'Raiz3' y pinchar en la pestaña Sinónimos de la información del término.3. Pinchar con el botón derecho en el sinónimo y elegir la opción de 'Editar sinónimo':4. En el nuevo formulario cambiar el nombre por: 'Sinónimo de Raiz3_Modificado'.4. Pinchar en aceptar para confirmar.
Resultado:	<p>El resultado visual debe ser que en Sinónimos de se ha modificado la fila con el nuevo nombre del sinónimo.</p> <p>En Relaciones también se ha modificado el nombre en la fila de la relación.</p>
Requisitos relacionados:	RSF – 036, RSF – 040, RSF - 043

Tabla 132 - PA-018: Modificar un sinónimo

PA-019: Intercambiar un sinónimo por un término	
Descripción:	<p>Se comprueba que intercambiar un sinónimo por un término funciona correctamente.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón izquierdo sobre el término 'Raíz3' y pinchar en la pestaña Sinónimos de la información del término. 3. Pinchar con el botón derecho en el sinónimo 'Sinónimo de Raiz3_Modificado' y elegir la opción de 'Usar por': 4. Pulsar en aceptar para confirmar.
Resultado:	<p>En Sinónimos debe existir una fila con el nombre 'Raíz3'.</p> <p>En Relaciones se ha modificado el sentido de la relación de sinonimia del término ahora 'Sinónimo de Raiz3_Modificado'.</p> <p>En el tesouro se ha modificado el nombre del término ahora es 'Sinónimo de Raiz3_Modificado'.</p>
Requisitos relacionados:	RSF – 002, RSF – 036, RSF – 040, RSF - 044

Tabla 133 - PA-019: Intercambiar un sinónimo por un término

PA-020: Comprobar campos obligatorios de un sinónimo	
Descripción:	<p>Se comprueba que los campos obligatorios a la hora de rellenar la información de un sinónimo son correctos.</p> <ol style="list-style-type: none"> 1. Acceder al apartado de Tesouro. 2. Pinchar con el botón derecho sobre el término 'Sinónimo de Raiz3_Modificado' y elegir la pestaña de Sinónimos de la información del término. 2. Pinchar con el botón derecho sobre el sinónimo 'Raíz3' y elegir la opción de 'Añadir un sinónimo'. 3. Dejar el campo de texto vacío en el nuevo formulario. 4. Pinchar en aceptar para confirmar. El resultado es un mensaje de error que avisa de que no se puede crear un sinónimo con el nombre vacío o igual a algún término ya creado.
Resultado:	El resultado se comenta ya en la descripción de la prueba.
Requisitos relacionados:	RSF – 003, RSF – 005, RSF – 036, RSF - 037

Tabla 134 - PA-020: Comprobar campos obligatorios de un sinónimo

PA-021: Eliminar un sinónimo	
Descripción:	<p>Se comprueba que eliminar un sinónimo funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón izquierdo sobre el término 'Sinónimo de Raiz3_Modificado' y pinchar en la pestaña Sinónimos de la información del término.3. Pinchar con el botón derecho en el sinónimo y elegir la opción de 'Eliminar sinónimo':4. Pulsar en aceptar para confirmar borrado.
Resultado:	<p>En Sinónimos no debe existir ningún sinónimo.</p> <p>En Relaciones se ha eliminado la relación de sinonimia del término 'Sinónimo de Raiz3_Modificado'.</p>
Requisitos relacionados:	RSF – 036, RSF – 040, RSF - 042

Tabla 135 - PA-021: Eliminar un sinónimo

3.8.2.4. Otras funcionalidades

PA-022: Añadir buscador	
Descripción:	<p>Se comprueba que añadir un buscador web funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón izquierdo sobre el término 'Raiz3' y elegir la opción 'Añadir buscador'.3. En el formulario emergente rellenar los campos: URL: https://www.google.es/search?q=TEST&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:es-ES:official&client=firefox-a&source=hp&channel=np Esta url corresponde a buscar en el buscador deseado la palabra TEST. Nombre: Google4. Pulsar en aceptar para confirmar.
Resultado:	En el apartado de buscadores aparece Google como nuevo buscador.
Requisitos relacionados:	RSF – 003, RSF – 021, RSF - 022

Tabla 136 - PA-022: Añadir buscador

PA-023: Buscador web	
Descripción:	Se comprueba que buscar un término en la web funciona correctamente. <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón izquierdo sobre el término 'Raiz3' y elegir la opción 'Google' que es el buscador creado anteriormente.
Resultado:	Se visualiza los resultados de buscar en google el término 'Raiz3'.
Requisitos relacionados:	RSF – 003, RSF – 021, RSF - 025

Tabla 137 - PA-023: Buscador web

PA-024: Eliminar buscador	
Descripción:	Se comprueba que eliminar buscador web funciona correctamente. <ol style="list-style-type: none">1. Acceder al apartado de Tesouro.2. Pinchar con el botón izquierdo sobre el término 'Raiz3' y elegir la opción 'Eliminar buscador'.3. En el formulario emergente seleccionar el buscador Google creado previamente.4. Pulsar en aceptar para confirmar el borrado.
Resultado:	Ya no hay buscadores web agregados al tesouro.
Requisitos relacionados:	RSF – 003, RSF – 021, RSF - 023

Tabla 138 - PA-024: Eliminar buscador

PA-025: Refrescar	
Descripción:	Se comprueba que refrescar el tesoro funciona correctamente. <ol style="list-style-type: none">1. Acceder al apartado de Tesoro.2. Abrir las ramas posibles del tesoro. Estas son: 'Raiz1', 'Raiz2', 'Hijo21'2. Pinchar con el botón izquierdo sobre el término 'Raiz3' y elegir la opción 'Refrescar'.
Resultado:	El resultado es que el tesoro vuelve a estar en el estado inicial.
Requisitos relacionados:	RSF – 002, RSF – 003, RSF - 026

Tabla 139 - PA-025: Refrescar

PA-026: Importar	
Descripción:	Se comprueba que importar el tesoro funciona correctamente. <ol style="list-style-type: none">1. Acceder al apartado de Subdominios y después pinchar en Importar.2. Se debe tener en un fichero *.jcd la siguiente estructura: .Raiz1 ..Hijo11 ..Hijo12 .Raiz2 ..Hijo21 ...Hijo211 .Raiz33. Pinchar en el botón importar y esperar.4. Confirmar el resultado del proceso de importación.
Resultado:	El resultado es que el ahora en nuestro tesoro aparecen esos términos con sus respectivas relaciones.
Requisitos relacionados:	RSF – 002, RSF - 045

Tabla 140 - PA-026: Importar

PA-027: Exportar	
Descripción:	<p>Se comprueba que exportar el tesouro funciona correctamente.</p> <ol style="list-style-type: none">1. Acceder al apartado de Subdominios y después pinchar en Exportar.2. Elegir el tipo fichero con extensión *.jcd.3. Pinchar en el botón exportar y esperar.4. Confirmar el resultado del proceso de exportación. <p>4. Resultado: Abrir el fichero y comprobar que la estructura es:</p> <ul style="list-style-type: none">.Raiz1..Hijo11..Hijo12.Raiz2..Hijo21...Hijo211.Raiz3
Resultado:	El resultado de la prueba se comenta en la propia descripción.
Requisitos relacionados:	RSF - 046

Tabla 141 - PA-027: Exportar

Capítulo 4. Diseño

El objetivo de este capítulo es elaborar el diseño de la solución con un nivel de profundidad mayor del obtenido en el capítulo de análisis, con el fin de satisfacer los requisitos planteados durante la fase de análisis. En resumen, el objetivo es crear de forma satisfactoria un módulo que gestione correctamente los subsistemas estudiados, y que sea tanto verificable como validado por parte del cliente.

4.1. Definición de la arquitectura del sistema

En este apartado se definen aspectos de la arquitectura del sistema y se lleva a cabo un estudio a nivel global en cuanto a la interacción del módulo desarrollado con el resto del sistema se refiere. Por otro lado, también se establece la normativo de código que se aplica en el desarrollo y se definen las excepciones que se puedan tener dentro del sistema.

4.1.1. Definición de los niveles de arquitectura

El subsistema a desarrollar así como toda la aplicación del [knowledgeMANAGER](#) sigue una arquitectura basada en tres capas.

- **Capa de presentación**, denominada *PL (Presentation Layer)*. Separa la responsabilidad de mostrar la interfaz de usuario y la interacción con el mismo. Haciendo uso de esta capa, se utiliza el patrón de diseño *facade*.
- **Capa de negocio**, denominada *BLL (Business Logic Layer)*. Es responsable de almacenar todas las funciones que determinan la lógica de negocio de la aplicación.
- **Capa de datos**, denominada *DAL (Data Access Logic)*. Su responsabilidad es interactuar con la base de datos, para realizar una gestión óptima de las operaciones sobre esta.

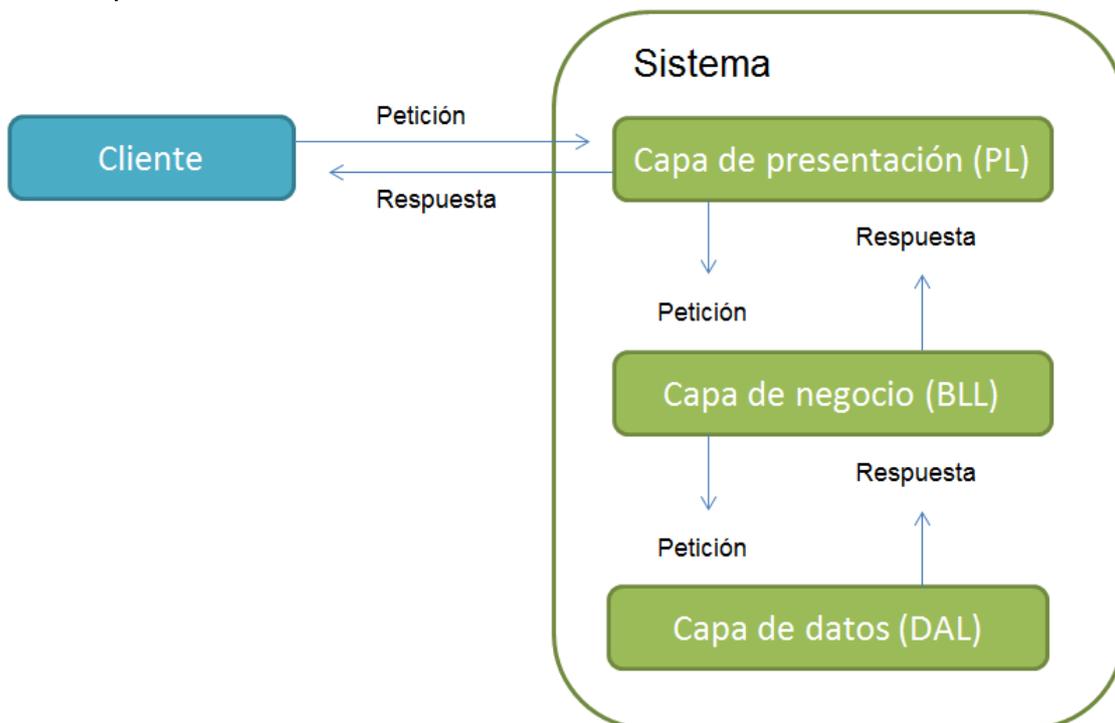


Ilustración 21 - Modelo arquitectónico en tres capas

Como se puede observar el modelo utilizado se divide en tres capas. Se caracteriza por dividir la funcionalidad y las responsabilidades de la aplicación, llevando a cabo peticiones y respuestas entre las capas cuando se solicita algún tipo de información por parte del usuario de la aplicación.

4.1.2. Especificación de excepciones

Se ha diseñado el módulo del tesouro del [knowledgeMANAGER](#) para evitar el lanzamiento de excepciones en la medida de lo posible.

Por un lado, las excepciones que se capturan en la capa de presentación (PL) suelen ser debidas a fallos en la interacción entre el sistema y el usuario. Cuando son capturadas dichas excepciones se muestra un mensaje al usuario tipo *MessageBox*, además de ser registradas en un fichero de *log de errores*.

Por otro lado, las excepciones que se capturan en la capa de negocio (BLL) son registradas en el fichero de *log de errores* anteriormente nombrado y además devuelven valores nulos a la capa de presentación (PL) para saber que algo ha ido mal durante el proceso.

4.1.3. Especificación de estándares y normas de diseño

Para la codificación del sistema es necesario seguir una serie de normas para mantener la homogeneidad de código en todo el proyecto:

- **Atributos:** se escriben en minúscula. Si el nombre del atributo abarca más de una palabra, se escribe la inicial de cada palabra en mayúscula, exceptuando la primera que es íntegra en minúscula.
- **Constantes:** se escriben en mayúscula. Si el nombre de la constante abarca más de una palabra, se separan mediante un guion bajo.
- **Funciones:** se escriben en minúscula, excepto la primera letra de cada palabra, que se escribe en mayúscula.
- **Clases:** se escriben en minúscula, excepto la primera letra de cada palabra, que se escribe en mayúscula.

No es totalmente estricto, pero deben existir comentarios para todos los métodos y clases generadas así como algunos atributos y constantes que se vea necesario para su posterior comprensión. Cualquier norma que no esté especificada en este estándar, es decisión del desarrollador.

4.2. Diseño de la arquitectura

Uno de los principales objetivos de un buen diseño debe ser la separación de responsabilidades, con el fin de facilitar la escalabilidad del producto y su fácil mantenimiento ante posibles cambios que se produzcan en los requisitos a lo largo o después del proyecto. Además se reduce la complejidad del sistema a desarrollar.

4.2.1. Diseño de subsistemas

Con el fin de comenzar a dividir las responsabilidades del sistema, en nuestro caso el subsistema principal es el tesouro pero como ya se ha mencionado en el análisis se ha dividido el subsistema del tesouro en módulos más pequeños según los tipos de funcionalidad que abarcan. Se mantienen los subsistemas diseñados previamente y se especifican formalmente:

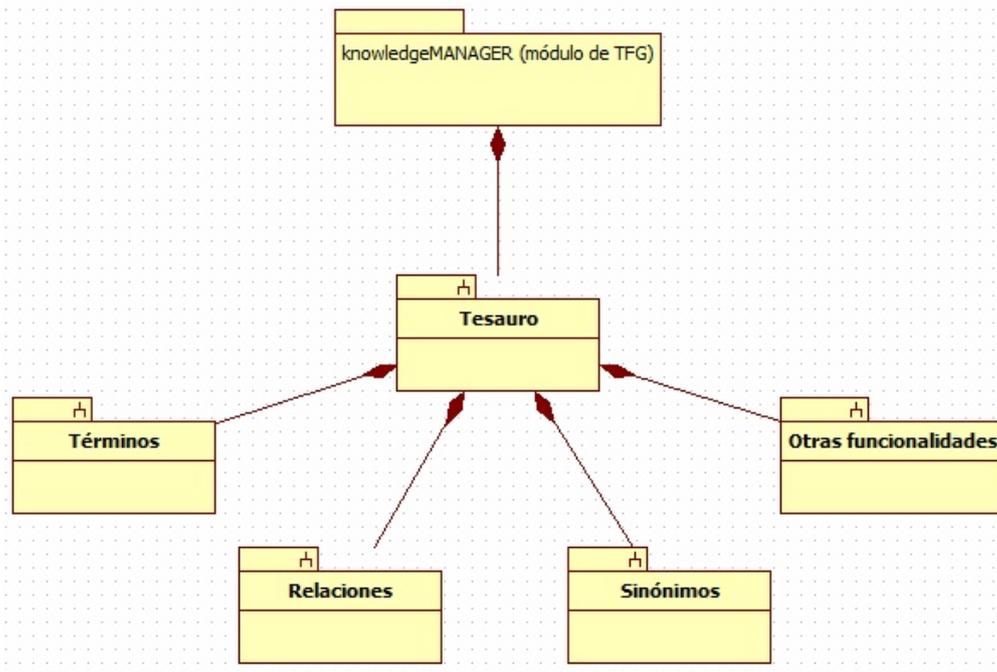


Ilustración 22 - Subsistemas de diseño

Como se puede observar en la imagen y se ha comentado varias veces a lo largo del presente proyecto, el módulo desarrollado se acopla dentro del proyecto [knowledgeMANAGER](#). Esto hace necesario que el módulo interactúe con otros componentes, lo que supone el uso de librerías externas ya desarrolladas.

A continuación se muestra el diagrama de interacciones en las que interviene este módulo con el resto del sistema:

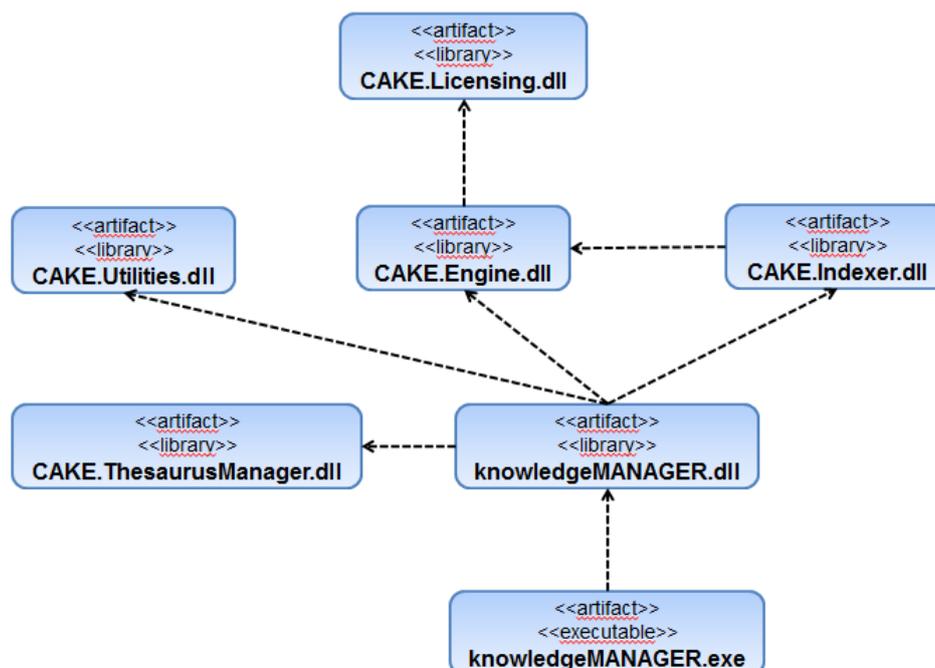


Ilustración 23 - Diseño de componentes del sistema

Completando al diagrama anterior, existen cinco componentes fundamentales en los que se basa el módulo a desarrollar:

- **Cake.Engine:** lleva a cabo toda la comunicación con la base de datos. Es la capa DAL de toda la aplicación, y se inicializa una sola vez al iniciar la aplicación.
- **Cake.Indexer:** se encarga de todas las operaciones de *indexing* y *retrieval*, e interactúa con la base de datos a través del *Cake.Engine*.
- **Cake.Licensing:** lleva a cabo todo lo relacionado con el licenciamiento de la aplicación.
- **Cake.Utilities:** gestiona el formulario de conexión a la base de datos y la ventana de *splash* de espera.
- **Cake.ThesaurusManager:** se encarga de manejar el diseño gráfico de las relaciones del tesouro.

Todos ellos se tratan de *sistemas caja negra*. Por ello no se profundiza en el análisis y diseño de cada uno de ellos.

Además de todo esto, se debe mencionar la particularidad del tesouro ya que se ha utilizado para su desarrollo el uso de [Windows Presentation Foundation \(WPF\)](#) y la utilización de la clase [ObservableCollection](#) que da al tesouro la visualización y rendimiento perfectos para el uso futuro en tesouros más complejos y grandes.

4.3. Revisión de la interfaz de usuario

Se mantiene la interfaz de usuario definida en el análisis. Dichas interfaces se desarrollan mediante controles de usuario que son integrados en un formulario principal, común a toda la aplicación. Así, dependiendo la opción del menú que se utilice, se mantiene el mismo formulario exterior pero cambia el control de usuario que se utilice en cada momento.

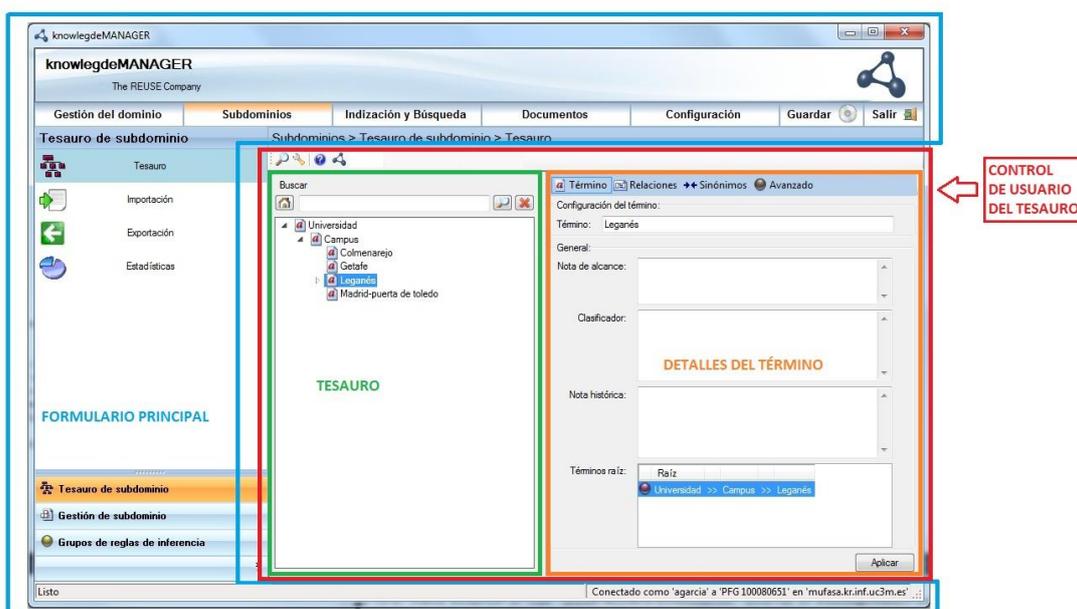


Ilustración 24 - Diseño de la interfaz de usuario

4.4. Diseño de clases

A continuación se profundiza en el diseño de clases llevado a cabo durante el capítulo de análisis. Se parte de la identificación de clases que se hizo durante dicha fase, para completar todas las clases que se consideren necesarias para el desarrollo del sistema. Además se estudian en detalle cuáles son los atributos y funciones necesarias en cada clase. A continuación se muestra el diagrama de clases del proyecto:

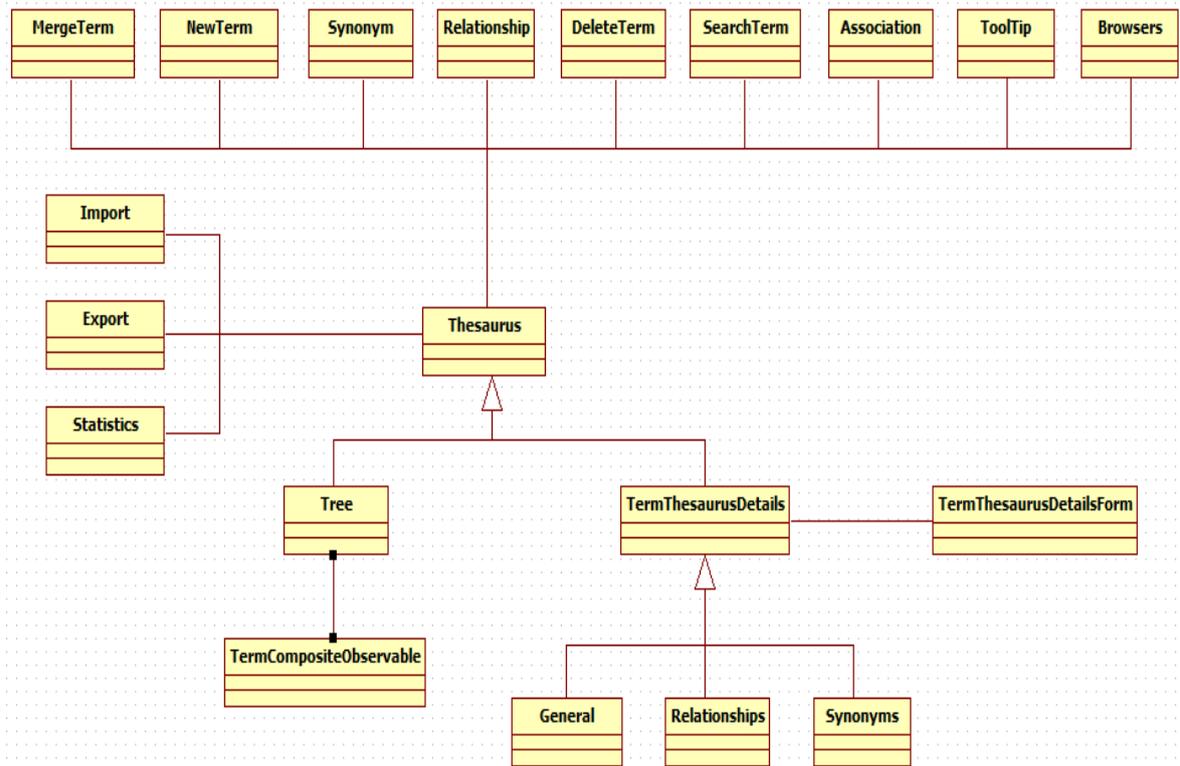


Ilustración 25 - Diagrama de clases

4.4.1. Identificación y especificación de clases

Como en el capítulo de análisis se tienen cuatro tipos de clases, de las cuales se obvia las clases de entidad ya que como se dijo son usadas como caja negra.

En esta ocasión, la especificación de clases es mucho más detallada que la realizada en el apartado de análisis ya que en este punto se tiene mayor conocimiento de la aplicación. Se agrupa siguiendo el formato del apartado anterior por tipo de clase.

4.4.1.1 Clases de apoyo a la interfaz

- **VocabularyDataSet:** contiene los atributos que forman un término.
- **GrammaticalDataSet:** contiene los atributos que forman una semántica.
- **RulesFamiliesDataSet:** contiene los atributos que forman una etiqueta sintáctica.
- **ArtifactDataSet:** contiene los atributos que forman un artefacto.

A continuación, se especifican las características de las clases:

VocabularyDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	<p><code>private VocabularyDataTable vocabularyDataTable</code> Tabla de datos que contiene los atributos referidos a un término.</p> <p><code>private TermForComboBoxDataTable vocabularyDataTableForComboBox</code> Tabla de datos que contiene los atributos necesarios para mostrar un término en un control <i>ComboBox</i> desplegable.</p>
Operaciones:	-

Tabla 142 - Diseño de VocabularyDataSet

GrammaticalDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	<p><code>private GrammaticalDataTable semanticDataTable</code> Tabla de datos que contiene los atributos referidos a una semántica.</p> <p><code>private GrammaticalForComboBoxDataTable semanticDataTableForComboBox</code> Tabla de datos que contiene los atributos necesarios para mostrar una semántica en un control <i>ComboBox</i> desplegable.</p>
Operaciones:	-

Tabla 143 - Diseño de GrammaticalDataSet

RulesFamiliesDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	<p><code>private TermTagsDataTable termTagsDataTable</code> Tabla de datos que contiene los atributos referidos a una etiqueta sintáctica.</p> <p><code>private TermTagForComboBoxDataTable termTagsDataTableForComboBox</code> Tabla de datos que contiene los atributos necesarios para mostrar una etiqueta sintáctica en un control <i>ComboBox</i> desplegable.</p>
Operaciones:	-

Tabla 144 - Diseño de RulesFamiliesDataSet

ArtifactDataSet	
Responsabilidades:	Lleva a cabo la interacción con la capa de datos para obtener la información necesaria de la base de datos.
Atributos:	<p><code>private ArtifactsDataTable artifactsDataTable</code> Tabla de datos que contiene los atributos referidos a un artefacto.</p> <p><code>private ArtifactsDataTableForComboBox artifactsDataTableForComboBox</code> Tabla de datos que contiene los atributos necesarios para mostrar un artefacto en un control <i>ComboBox</i> desplegable.</p>
Operaciones:	-

Tabla 145 - Diseño de ArtifactDataSet

4.4.1.2. Clases de Interfaz

- **Thesaurus:** contiene el árbol del tesouro y la información del término.
- **TermTreeView:** contiene el árbol del tesouro.
- **TermCompositeObservable:** contiene la información de los términos del tesouro.

- **TermThesaurusDetails:** contiene la información del término.
- **General:** muestra la información básica de un término en el tesouro.
- **Relationships:** muestra todas las relaciones de un término en el tesouro.
- **Synonyms:** muestra todos los sinónimos de un término.
- **AdvancedTerm:** Muestra detalles específicos de un término.
- **NewTerm:** formulario para crear un nuevo término en el tesouro.
- **Relationship:** formulario para crear una nueva relación en el tesouro
- **Synonym:** formulario para crear un nuevo sinónimo en el tesouro.
- **Association:** formulario para crear una nueva relación de asociación en el tesouro.
- **SearchTerm:** formulario para realizar búsquedas avanzadas de un término en el tesouro.
- **DeleteTerm:** formulario para eliminar un término del tesouro.
- **TermThesaurusDetailsForm:** formulario para modificar un término del tesouro. Tiene la misma funcionalidad que TermThesaurusDetails.
- **MergeTerm:** formulario para mezclar términos del tesouro.
- **Import:** muestra la importación del tesouro.
- **Export:** muestra la exportación del tesouro.
- **Statistics:** muestra las estadísticas del tesouro.

A continuación, se especifican las características de las clases:

Thesaurus	
Responsabilidades:	Interfaz que contiene toda la funcionalidad que viene desde el árbol y los detalles del término. Funciona como un controlador.
Atributos:	<pre>private ObservableCollection<TermCompositeObservable> observableRoots</pre> <p>Objeto que contiene los términos raíces del árbol.</p> <pre>private TermCompositeObservable selectedTermCompositeObservable</pre> <p>Objeto que contiene el término seleccionado del árbol.</p> <pre>private Dictionary<Term, Dictionary<int, List<TermCompositeObservable>>> termsCompositeObservableInTreeFromTerm;</pre> <p>Diccionario que contiene la lista de términos del tesouro.</p> <pre>private Dictionary<int, TermThesaurusDetailsForm> openedForms</pre> <p>Diccionario que controla los formularios de detalle de términos abiertos.</p>
Operaciones:	<p>Funciones que pertenecen al menú contextual del tesouro:</p> <pre>public TermCompositeObservable AddNewTopTermInTree()</pre> <p>Función que llama a la BLL para crear un nuevo término raíz</p>

	<p>del tesouro.</p> <pre>public TermCompositeObservable AddNewChildInTree()</pre> <p>Función que llama a la BLL para crear un nuevo término hijo a un término del tesouro.</p> <pre>public bool AddRelationship ()</pre> <p>Función que llama a la BLL para crear una relación entre términos del tesouro.</p> <pre>private bool AddSynonymInTree()</pre> <p>Función que llama a la BLL para crear un nuevo sinónimo al tesouro.</p> <pre>public bool AddAssociation()</pre> <p>Función que llama a la BLL para crear una nueva relación de asociación entre términos del tesouro.</p> <pre>public bool DeleteTerm(Term term)</pre> <p>Función que llama a la BLL para eliminar un término del tesouro.</p> <pre>public bool MakeARootTerm()</pre> <p>Función que llama a la BLL para mover un término del tesouro a la raíz.</p> <pre>public bool DropTerm(TermCompositeObservable sourceItem, TermCompositeObservable destinationParentItem, TermCompositeObservable parentSourceItem, System.Windows.DragDropEffects finalDropEffect)</pre> <p>Función que llama a la BLL para utilizar el drag and drop entre los términos pasados por parámetros.</p> <pre>public bool MergeTerm(TermCompositeObservable selectedTermCompositeObservable, TermCompositeObservable termToMerge)</pre> <p>Función que llama a la BLL para utilizar la fusión de términos entre los términos pasados por parámetros.</p> <pre>public bool EditTerm(TermCompositeObservable selectedTermComposite)</pre> <p>Función que abre un formulario de edición del término del tesouro.</p> <pre>public bool RefreshTree()</pre> <p>Función que refresca el tesouro poniéndolo en su estado inicial.</p> <pre>public bool ExpandAllTree(TermCompositeObservable selectedTermComposite)</pre> <p>Función que expande toda la rama del término seleccionado por parámetros.</p>
--	---

	<p>Funciones que pertenecen a la ventana de detalle de los términos:</p> <pre>public bool ChangeSelectedTerm(TermCompositeObservable selectedTermComposite, bool changeToSynonymTab)</pre> <p>Función que modifica la información de la pestaña de detalles del término.</p> <p>Funciones propias de la interfaz:</p> <pre>public bool Search()</pre> <p>Función que llama a la BLL para buscar un término del tesouro.</p> <pre>public bool ShowAdvancedSearch()</pre> <p>Función que llama al formulario de búsqueda avanzada de términos del tesouro.</p> <pre>public ObservableCollection<TermCompositeObservable> LoadObservableRoots()</pre> <p>Función que llama a la BLL para cargar los términos iniciales raíces del tesouro.</p> <pre>public bool ManageOpenedForms(TermCompositeObservable selectedTermComposite)</pre> <p>Controla los formularios abiertos de términos para su posible actualización y mejora de la usabilidad del usuario.</p> <p>Funciones de refresco del tesouro:</p> <pre>public bool RefreshTreeAfterAddRelationship(Term source, Term destination, SemanticItem selectedSemantic)</pre> <p>Función que refresca el árbol después de añadir una relación en el tesouro.</p> <pre>private void RefreshTreeAfterDeleteRelationship(Term source, Term destination, SemanticItem selectedSemantic)</pre> <p>Función que refresca el árbol después de eliminar una relación en el tesouro.</p> <pre>public bool RefreshSynonymsInThesaurus(Term synonym)</pre> <p>Función que refresca el árbol después de añadir un sinónimo en el tesouro.</p> <pre>public bool RefreshTreeAfterDeleteTerm(Term newParent, Term synonym, OperationTypeOfDeleteTerm action)</pre> <p>Función que refresca el árbol después de eliminar un término en el tesouro.</p> <pre>public bool UpdateAllOpened()</pre> <p>Función que refresca las pestañas de los formularios que estén abiertos y se encuentren implicados.</p>
--	--

Tabla 146- Diseño de Thesaurus

TermTreeView	
Responsabilidades:	Interfaz para mostrar el tesauro de la ontología.
Atributos:	<pre>private TermCompositeObservable selectedTermCompositeObservable</pre> <p>Objeto que contiene el término seleccionado del árbol.</p> <pre>private Dictionary<Term, Dictionary<int, List<TermCompositeObservable>>> termsCompositeObservableInTreeFromTerm;</pre> <p>Diccionario que contiene la lista de términos del tesauro.</p>
Operaciones:	<pre>public bool LoadOptionsMenuContextual()</pre> <p>Carga del menú contextual de los términos.</p> <pre>public bool MouseMove(TermCompositeObservable selectedTermCompositeObservable)</pre> <p>Se encarga de obtener el término a mover</p> <pre>private void DragOver(TermCompositeObservable selectedTermCompositeObservable)</pre> <p>Se encarga de obtener el término donde se va a mover.</p> <pre>private void Drop(TermCompositeObservable selectedTermCompositeObservable)</pre> <p>Se encarga de la operación de soltar el término a mover.</p>

Tabla 147 - Diseño de TermTreeView

TermCompositeObservable	
Responsabilidades:	Clase que representa los términos del tesoro.
Atributos:	<pre>private Dictionary<Term, Dictionary<int, List<TermCompositeObservable>>> termsCompositeObservableInTreeFromTerm;</pre> Diccionario que contiene la lista de términos del tesoro. <pre>private Term theTerm</pre> Objeto que contiene el término. <pre>private TermCompositeObservable observableParent</pre> Objeto que contiene el padre visual. <pre>private ObservableCollection<TermCompositeObservable> observableChildren</pre> Colección de hijos de un término.
Operaciones:	Esta clase contiene propiedades bindeadas y modificables con el árbol para modificarlas en ambos sentidos con el fin de mejorar el refresco.

Tabla 148 - Diseño de TermCompositeObservable

TermThesaurusDetails	
Responsabilidades:	Interfaz que contiene las pestañas de los detalles del término.
Atributos:	<p><code>private General</code> general Objeto con la información básica del término.</p> <p><code>private Relationships</code> relationships Objeto con de información de relaciones del término.</p> <p><code>private Synonyms</code> synonyms Objeto con la información de sinónimos del término.</p> <p><code>private TermEditionUserControl</code> termEditionUserControl Objeto con la información avanzada del término.</p>
Operaciones:	<p><code>public void</code> TermGeneral () Muestra la información de un término en el tesouro.</p> <p><code>public void</code> TermRelationships() Muestra la información de las relaciones de un término en el tesouro.</p> <p><code>public void</code> TermSynonyms() Muestra la información de los sinónimos de un término en el tesouro.</p> <p><code>public bool</code> Accept() Acepta los cambios realizados en las diferentes pestañas y propaga eventos a la clase controladora.</p> <p><code>public bool</code> ValidateInformationTerm () Validar que los campos introducidos son válidos.</p>

Tabla 149 - Diseño de TermThesaurusDetails

General	
Responsabilidades:	Interfaz que contiene la información básica del término.
Atributos:	<p><code>private DataView</code> breadCrumbDataView Objeto que contiene la tabla con la miga de pan del término.</p>
Operaciones:	<p><code>public void</code> UpdateUI() Muestra la información de un término en el tesouro.</p>

Tabla 150 - Diseño de General

Relationships	
Responsabilidades:	Interfaz que contiene la información de relaciones del término.
Atributos:	<code>private DataView relationshipsDataView</code> Objeto que contiene la tabla las relaciones del término.
Operaciones:	<code>public void UpdateUI()</code> Muestra la información de las relaciones de término en el tesouro. <code>public RSHP AddRelationship()</code> Añade una relación en la tabla de relaciones del término. <code>public bool DeleteRelationship()</code> Elimina una relación de la tabla de relaciones del término. <code>public bool ShowAnotherParticipantInRelationship()</code> Muestra el formulario de edición del término el otro participante de la relación.

Tabla 151 – Diseño de Relationships

Relationship	
Responsabilidades:	Interfaz para añadir una relación en el tesouro.
Atributos:	<code>private Term term</code> Objeto que contiene el término del tesouro.
Operaciones:	<code>public void SelectTerm()</code> Muestra el formulario de búsqueda avanzada del tesouro para elegir un término. <code>public void SelectSemantic()</code> Muestra un nuevo formulario para cambiar la semántica del término. <code>public bool ValidateRelationship()</code> Valida si la relación a crear es una relación posible según las restricciones de la capa de negocio. <code>public bool Accept()</code> Envía los datos de la relación a crear a la BLL para que haga las operaciones oportunas.

Tabla 152 - Diseño de Relationship

AddAssociation	
Responsabilidades:	Interfaz para añadir una relación de asociación en el tesoro.
Atributos:	<code>private Term term</code> Objeto que contiene el término del tesoro.
Operaciones:	<code>public bool ValidateAssociation()</code> Valida si la relación de asociación a crear es una relación posible según las restricciones de la capa de negocio. <code>public bool Accept()</code> Envía los datos de la relación a crear a la BLL para que haga las operaciones oportunas.

Tabla 153 – Diseño de AddAssociation

SearchTerm	
Responsabilidades:	Interfaz para la búsqueda avanzada de términos en el tesoro.
Atributos:	<code>private DataView termsDataView</code> Objeto que contiene la tabla los términos que coincidan con los criterios de búsqueda..
Operaciones:	<code>public bool SelectTermTag()</code> Muestra un nuevo formulario para cambiar la etiqueta sintáctica del término. <code>public void SelectSemantic()</code> Muestra un nuevo formulario para cambiar la semántica del término. <code>public void SelectLanguage()</code> Muestra un comboBox para cambiar el lenguaje del término. <code>public void SelectLanguage()</code> Muestra un nuevo formulario para cambiar el lenguaje del término. <code>public void SelectRootTerm()</code> Muestra un comboBox para elegir la rama del tesoro en la que buscar el término. <code>public bool Accept()</code> Envía los datos del término seleccionado al formulario o control de usuario que lo llamo.

Tabla 154 – Diseño de SearchTerm

Synonyms	
Responsabilidades:	Interfaz que contiene la información de sinónimos del término.
Atributos:	<code>private DataView synonymsDataView</code> Objeto que contiene la tabla con los sinónimos del término.
Operaciones:	<code>public void UpdateUI()</code> Muestra la información de los sinónimos de un término en el tesouro. <code>public Term AddSynonym()</code> Añade un sinónimo en la tabla de sinónimos del término. <code>public bool DeleteSynonym()</code> Elimina un sinónimo de la tabla de sinónimos del término. <code>public bool EditSynonym()</code> Modifica el sinónimo de la tabla de sinónimos del término. <code>public bool InterchangeSynonymForTerm()</code> Intercambiar un sinónimo por un término del tesouro.

Tabla 155 – Diseño de Synonyms

Synonym	
Responsabilidades:	Interfaz para añadir o editar un sinónimo en el tesouro.
Atributos:	<code>private Term term</code> Objeto que contiene el término del tesouro.
Operaciones:	<code>public bool ValidateSynonym()</code> Valida si el sinónimo a crear es una sinónimo posible según las restricciones de la capa de negocio. <code>public bool Accept()</code> Envía los datos del sinónimo a crear a la BLL para que haga las operaciones oportunas.

Tabla 156 - Diseño de Synonym

AdvancedTerm	
Responsabilidades:	Interfaz que contiene la información avanzada del término del tesouro.
Atributos:	-
Operaciones:	<pre>public void UpdateUI()</pre> Muestra la información avanzada de un término en el tesouro. <pre>public bool SelectTermTag()</pre> Muestra un nuevo formulario para cambiar la etiqueta sintáctica del término. <pre>public void SelectSemantic()</pre> Muestra un nuevo formulario para cambiar la semántica del término. <pre>public void SelectLanguage()</pre> Muestra un comboBox para cambiar el lenguaje del término.

Tabla 157 - Diseño de AdvancedTerm

NewTerm	
Responsabilidades:	Interfaz que contiene la información avanzada del término del tesouro.
Atributos:	<pre>private Term term</pre> Objeto que contiene el término del tesouro.
Operaciones:	<pre>public bool ValidateThesaurusTerm()</pre> Valida si el término a crear es una término posible según las restricciones de la capa de negocio. <pre>public bool Accept()</pre> Envía los datos del término a crear a la BLL para que haga las operaciones oportunas.

Tabla 158 - Diseño de NewTerm

DeleteTerm	
Responsabilidades:	Interfaz para eliminar un término del tesouro.
Atributos:	<code>private Term term</code> Objeto que contiene el término del tesouro.
Operaciones:	<code>public bool SelectDeleteOperation()</code> Elegir la operación de borrado. <code>public bool Accept()</code> Envía los datos del término a eliminar y la operación de borrado escogida a la BLL para que haga las operaciones oportunas.

Tabla 159 - Diseño de DeleteTerm

MergeTerm	
Responsabilidades:	Interfaz para fusionar dos términos del tesouro.
Atributos:	<code>private Term term</code> Objeto que contiene el término del tesouro. <code>private Term termToMerge</code> Objeto que contiene el término a mezclar del tesouro.
Operaciones:	<code>public bool Accept()</code> Envía los datos del término a eliminar y la operación de borrado escogida a la BLL para que haga las operaciones oportunas.

Tabla 160 - Diseño de MergeTerm

Export	
Responsabilidades:	Interfaz que contiene la información para exportar un tesouro.
Atributos:	<code>private ThesaurusRepository myThesuarusRepository</code> Objeto de la BLL que permite acceder a la lógica de negocio del tesouro.
Operaciones:	<code>public bool Export()</code> Envía los datos necesarios para la exportación a la BLL para que haga las operaciones oportunas.

Tabla 161 - Diseño de Export

Import	
Responsabilidades:	Interfaz que contiene la información para importar un tesouro.
Atributos:	<code>private ThesaurusRepository myThesuarusRepository</code> Objeto de la BLL que permite acceder a la lógica de negocio del tesouro.
Operaciones:	<code>public bool Import()</code> Envía los datos necesarios para la importación a la BLL para que haga las operaciones oportunas.

Tabla 162 - Diseño de Import

4.4.1.3. Clases de control

- **VocabularyRepository:** contiene la lógica de negocio de los términos.
- **GrammaticalRepository:** contiene la lógica de negocio de las semánticas.
- **RulesFamiliesRepository:** contiene la lógica de negocio de las etiquetas sintácticas.
- **ArtifactRepository:** contiene la lógica de negocio de los artefactos.
- **ThesaurusRepository:** contiene la lógica de negocio de todas las operaciones del tesouro.

A continuación, se especifican las características de las clases:

VocabularyRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de términos.
Atributos:	<pre>private ThesaurusRepository myThesaurusRepository</pre> <p>Objeto que gestiona la lógica de negocio para las funciones que requieran información del tesauro.</p>
Operaciones:	<pre>public VocabularyDataSet.VocabularyDataTable GetTermsForDataGridView()</pre> <p>Devuelve una tabla de datos con todos los términos de la ontología.</p> <pre>public VocabularyDataSet.VocabularyDataTable GetTermsForDataGridView(Term_Tag termTag)</pre> <p>Devuelve una tabla de datos con todos los términos de la ontología cuya etiqueta sintáctica sea la recibida por parámetro.</p> <pre>public VocabularyDataSet.VocabularyDataTable GetTermsForDataGridView(SemanticItem semantic)</pre> <p>Devuelve una tabla de datos con todos los términos de la ontología semántica sea la recibida por parámetro.</p> <pre>public VocabularyDataSet.VocabularyDataTable GetTermsForDataGridView(String termName, Term_Tag tag, Language language, SemanticItem semantic, bool? belongsToDomain)</pre> <p>Devuelve una tabla de datos con todos los términos de la ontología cuyos campos coincidan con los pasados por parámetro.</p> <pre>private void MakeAListOfTermsOnlyThesaurus(bool? belongsToDomain, List<Term> finalTerms, List<Term> filteredTerms)</pre> <p>Devuelve una tabla de datos con todos los términos que contiene la lista pasada por parámetro.</p> <pre>public Engine.Term GetTermByExactName(string termName)</pre> <p>Devuelve el primer término de la ontología cuyo nombre coincida exactamente con la cadena pasada por parámetro.</p> <pre>public bool ValidateTerm(Cake.Engine.Term editingTerm, Cake.Engine.Term_Tag termTag, Cake.Engine.SemanticItem grammatical, string name, bool? domain, Cake.Engine.Language language, int domainImpact, int languageImpact)</pre> <p>Comprueba si el término y sus atributos (recibido todo como parámetro) es válido para ser almacenado en la ontología. Devuelve false en caso de no ser válido.</p> <pre>public bool UpdateTerm(Cake.Engine.Term editingTerm, Cake.Engine.Term_Tag termTag, Cake.Engine.SemanticItem grammatical, string name, bool synchronize,</pre>

	<p><code>Cake.Engine.Language</code> language, <code>int</code> domainImpact, <code>int</code> languageImpact, <code>Cake.Engine.Term</code> scopeNoteTerm)</p> <p>Tras invocar a la función <code>ValidateTerm</code>, envía a la capa de datos la petición para modificar la información de un término en la ontología.</p> <pre>public bool InsertTerm(Cake.Engine.Term_Tag termTag, Cake.Engine.SemanticItem grammatical, string name, bool? domain, Cake.Engine.Language language, int domainImpact, int languageImpact, string scopeNoteTerm, ref Cake.Engine.Term newTerm)</pre> <p>Tras invocar a la función <code>ValidateTerm</code>, envía a la capa de datos la petición para insertar la información de un nuevo término en la ontología.</p> <pre>public bool DuplicateTerm(Cake.Engine.Term term, ref Cake.Engine.Term newTerm)</pre> <p>Envía a la capa de datos la petición para crear un nuevo término a partir de la información de uno ya existente.</p> <pre>public bool ValidateDeleting(Cake.Engine.Term deletingTerm, ref List<Cake.Engine.Special_Sentence> specialSentences, ref List<Cake.Engine.Contraction> contractions, ref List<Cake.Engine.S_Exception> substituteException, ref List<Cake.Engine.Artifact> artifactsWithKes, ref List<Cake.Engine.ClarifierRule> clarifierRule, ref List<Cake.Engine.Pattern_SLOT> boilerplates, ref List<Cake.Engine.Artifact> artifactsWithMetas, ref List<Cake.Engine.Qualification> qualification, ref List<Cake.Engine.Locution> locution, ref List<Cake.Engine.Suggestion> suggestions)</pre> <p>Comprueba si existe algún element que impida el borrado de un término, debido a las restricciones de integridad existentes a nivel de base de datos. Devuelve false en caso de no ser posible el borrado del término.</p> <pre>public bool DeleteTerm(Cake.Engine.Term term)</pre> <p>Envía a la capa de datos la petición de borrado del término.</p>
--	--

Tabla 163 - Diseño de VocabularyRepository

GrammaticalRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de semánticas.
Atributos:	-
Operaciones:	<pre>public GrammaticalDataSet.GRAMMATICALDataTable GetSemantics()</pre> <p>Devuelve una tabla de datos con todas las semánticas de la ontología.</p> <pre>public GrammaticalDataSet.GrammaticalForComboBoxDataTable</pre>

<p>GetSemanticsForComboBox() Devuelve una tabla de datos con menos campos con todas las semánticas de la ontología.</p> <pre>public GrammaticalDataSet.GrammaticalForComboBoxDataTable GetSemantics(string semantic, bool? usedByThesaurus, bool onlyRoots)</pre> <p>Devuelve una tabla de datos con las semánticas de la ontología que coincidan con los criterios de búsqueda recibidos por parámetro.</p> <pre>public GrammaticalDataSet.GrammaticalForComboBoxDataTable GetSemanticsWithoutFirstRowForComboBox()</pre> <p>Devuelve una tabla de datos con todas las semánticas de la ontología.</p> <pre>public GrammaticalDataSet.GrammaticalForComboBoxDataTable GetSemanticsWithoutFirstRowForComboBox(SemanticItem semantic, bool includeAllChildren)</pre> <p>Devuelve una tabla de datos con las semánticas de la ontología cuya semántica coincida con la recibida por parámetro.</p> <pre>public GrammaticalDataSet.GrammaticalForComboBoxDataTable GetSemanticsWithoutFirstRowForComboBox(String grammaticalName)</pre> <p>Devuelve una tabla de datos con las semánticas de la ontología cuyo nombre contenga la cadena recibida por parámetro.</p> <pre>public bool ValidateSemantic(SemanticItem editingSemantic, string name, SemanticItem nomalizedSemanticItem, byte defaultPonderation, string roleA, string roleB, string roleAbreviationA, string roleAbreviationB)</pre> <p>Comprueba si la semántica y sus atributos (recibido todo como parámetro) es válida para ser almacenada en la ontología. Devuelve false en caso de no ser válida.</p> <pre>public bool UpdateSemantic(SemanticItem editingSemantic, string name, SemanticItem nomalizedSemanticItem, bool rotateConcept, byte defaultPonderation, bool reflexiveAllowed, bool loopsAllowed, bool transitiveAllowed, bool isSymmetrical, bool usedByThesaurus, string roleA, string roleB, string roleAbreviationA, string roleAbreviationB)</pre> <p>Tras invocar a la función ValidateSemantic, envía a la capa de datos la petición para modificar la información de una semántica en la ontología.</p> <pre>public bool InsertSemantic(string name, SemanticItem nomalizedSemanticItem, bool rotateConcept, byte defaultPonderation, bool reflexiveAllowed, bool loopsAllowed, bool transitiveAllowed, bool isSymmetrical, bool usedByThesaurus, string roleA, string roleB, string roleAbreviationA, string roleAbreviationB, ref SemanticItem newSemantic)</pre> <p>Tras invocar a la función ValidateSemantic, envía a la capa de</p>

	<p>datos la petición para insertar la información de una nueva semántica en la ontología.</p> <pre>public bool DuplicateSemantic(Cake.Engine.SemanticItem semantic, ref Cake.Engine.SemanticItem newSemantic)</pre> <p>Envía a la capa de datos la petición para crear una nueva semántica a partir de la información de una ya existente.</p> <pre>public bool ValidateDeleting(Cake.Engine.SemanticItem editingSemantic, ref List<Cake.Engine.Term> terms, ref List<Cake.Engine.Role> roles, ref List<Cake.Engine.ClarifierRule_BIT> clarifierRulesBit, ref List<Cake.Engine.Pattern_SLOT> patternSlots, ref List<Cake.Engine.SemanticItem> otherSubtypes)</pre> <p>Comprueba si existe algún elemento que impida el borrado de una semántica, debido a las restricciones de integridad existentes a nivel de base de datos. Devuelve false en caso de no ser posible el borrado de la semántica.</p> <pre>public bool DeleteSemantic(Cake.Engine.SemanticItem semantic)</pre> <p>Envía a la capa de datos la petición de borrado de la semántica.</p>
--	--

Tabla 164 - Diseño de GrammaticalRepository

RulesFamiliesRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de etiquetas sintácticas.
Atributos:	-
Operaciones:	<pre>public Rules_FamiliesDataSet.TermTagForComboBoxDataTable GetTermTagsForComboBox()</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología, adaptada para controles ComboBox despleables.</p> <pre>public Rules_FamiliesDataSet.TermTagsDataTable GetTermTagsForDataGridview()</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología.</p> <pre>public Rules_FamiliesDataSet.TermTagsDataTable GetTermTagsOnlyRootForDataGridview()</pre> <p>Devuelve una tabla de datos con las etiquetas sintácticas de la ontología que no tengan ningún padre (roots).</p> <pre>public Rules_FamiliesDataSet.TermTagsDataTable GetTermTagsForDataGridview(Term_Tag termTag)</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología que coincidan con la etiqueta sintáctica pasada por parámetro.</p>

<pre>public Rules_FamiliesDataSet.TermTagsDataTable GetTermTagsForDataGridview(String termTagName)</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología cuyos nombre contenga la cadena pasada por parámetro.</p> <pre>public Rules_FamiliesDataSet.TermTagsDataTable GetTermTags(string termTag, bool onlyRoots, bool nounType, bool verbType)</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología cuyos atributos coincidan con los indicados por parámetro.</p> <pre>public bool ValidateTermTag(Term_Tag editingTermTag, string name, Term_Tag parent, double domainStats, int appDomain, double corpusStats, int appCorpus)</pre> <p>Comprueba si la etiqueta sintáctica y sus atributos (recibido todo como parámetro) es válida para ser almacenada en la ontología. Devuelve false en caso de no ser válida.</p> <pre>public bool UpdateTermTag(Term_Tag editingTermTag, string name, Term_Tag parent, double domainStats, int appDomain, double corpusStats, int appCorpus, bool nounType, bool verbType)</pre> <p>Tras invocar a la función ValidateTermTag, envía a la capa de datos la petición para modificar la información de una etiqueta sintáctica en la ontología.</p> <pre>public bool InsertTermTag(string name, Term_Tag parent, double domainStats, int appDomain, double corpusStats, int appCorpus, bool nounType, bool verbType, ref Term_Tag newTermTag)</pre> <p>Tras invocar a la función ValidateTermTag, envía a la capa de datos la petición para insertar la información de una nueva etiqueta sintáctica en la ontología.</p> <pre>public bool DuplicateTermTag(Cake.Engine.Term_Tag termTag, ref Cake.Engine.Term_Tag newTermTag)</pre> <p>Tras invocar a la función ValidateTermTag, envía a la capa de datos la petición para duplicar la información de una etiqueta sintáctica de la ontología.</p> <pre>public bool ValidateDeleting(Cake.Engine.Term_Tag editingTermTag, ref List<Cake.Engine.Term> terms, ref List<Cake.Engine.Role> roles, ref List<Cake.Engine.Rule> rules, ref List<Cake.Engine.BigramStatistics> bigrams, ref List<Cake.Engine.Qualification> qualifications, ref List<Cake.Engine.ClarifierRule> oldInferenceRules, ref List<Cake.Engine.ClarifierRule> taggingRules, ref List<Cake.Engine.Pattern> inferenceRules, ref List<Cake.Engine.Locution> locutions, ref List<Cake.Engine.Term_Tag> termTags)</pre> <p>Comprueba si existe algún elemento que impida el borrado de una etiqueta sintáctica, debido a las restricciones de integridad existentes a nivel de base de datos. Devuelve false en caso de no ser posible el borrado de la etiqueta sintáctica.</p>

	<pre>public bool DeleteTermTag(Cake.Engine.Term_Tag termTag)</pre> <p>Envía a la capa de datos la petición de borrado de la etiqueta sintáctica.</p>
--	--

Tabla 165 - Diseño de RulesFamiliesRepository

ArtifactRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio del módulo de artefactos.
Atributos:	-
Operaciones:	<pre>public ArtifactDataSet.ArtifactsDataTable GetArtifactsForDataGridview(Engine.ArtifactType artifactType, Engine.SubDomain subDomain)</pre> <p>Devuelve una tabla de datos con todas las etiquetas sintácticas de la ontología que coincidan con la etiqueta sintáctica pasada por parámetro.</p> <pre>public bool DeleteArtifact(Artifact artifact)</pre> <p>Envía a la capa de datos la petición de borrado del artefacto.</p> <pre>public Artifact CreateIndexArtifact(CAKEEngine engine)</pre> <p>Envía a la capa de datos la petición de creación del artefacto.</p>

Tabla 166 - Diseño de ArtifactRepository

ThesaurusRepository	
Responsabilidades:	Clase de la BLL que contiene la lógica de negocio de todas las operaciones del tesouro.
Atributos:	<pre>private List<Term> rootsTerms</pre> <p>Contiene los términos raíces del tesouro</p> <pre>private SemanticItem semanticRootsTerms</pre> <p>Contiene la semántica de jerarquía asociada al tesouro.</p> <pre>private CakeIntegerDictionary<Term> allTerms</pre> <p>Contiene todos los términos del tesouro</p> <pre>private CakeIntegerDictionary<Term> synonymsTerms</pre> <p>Contiene todos los términos sinónimos del tesouro.</p> <pre>private Artifact artifact</pre> <p>Contiene la información del artefacto tesouro.</p>
Operaciones:	Funciones sobre relaciones:

<pre>public CakeIntegerDictionary<RSHP> GetRSHPs(Term sourceTerm, Term destinationTerm) Devuelve un diccionario con las relaciones que coincidan con los términos pasados por parámetros. public RSHP GetRshpByParticipants(Term sourceTerm, Term destinationTerm) Devuelve la primera relación que coincida con los términos pasados por parámetros. public bool IsTransitive(Term NewParent, Term NewChild, Term OldParent, ref Term problematicTerm) Comprueba si hay una relación transitiva entre los términos pasados por parámetros. Si la hay el término problemática en esa relación se devuelve. public bool IsCiclic(Term sourceTerm, Term destinationTerm) Comprueba si hay una relación cíclica entre los términos pasados por parámetros. public RSHP RSHP_Add(SemanticItem Semantic, Term sourceTerm, Term destinationTerm) Envía a la capa de datos la petición para añadir la información de una nueva relación en el tesouro de la ontología. public bool RSHP_Delete(RSHP rshp) Envía a la capa de datos la petición para eliminar la relación pasada por parámetros del tesouro de la ontología. public bool RSHP_Delete(Term sourceTerm, Term destinationTerm, ref string errorMessage) Envía a la capa de datos la petición para eliminar la relación del tesouro de la ontología pasando los términos por parámetros. Si hay un error se especifica por el parámetro pasado por referencia. Funciones sobre elementos de las relaciones: public KE AddKE(CAKEElement Owner, Term oTerm, byte Order, int Position) Envía a la capa de datos la petición para añadir la información de un elemento de la relación en el tesouro de la ontología. public KE ModifyKE(KE originalKE, Term oTerm) Envía a la capa de datos la petición para modificar la información de un elemento de la relación en el tesouro de la ontología. public bool DeleteKE(KE oKE) Envía a la capa de datos la petición para eliminar la información de un elemento de la relación en el tesouro de la ontología.</pre>

<p>Funciones sobre los sinónimos:</p> <pre>public bool LoadSynonymsTerms()</pre> <p>Carga la lista de sinónimos del tesouro.</p> <pre>public List<Term> SynonymsTerms ()</pre> <p>Devuelve la lista de todos los sinónimos del tesouro.</p> <pre>public bool IsSynonym(Term Term)</pre> <p>Comprueba si el término introducido por parámetros es un sinónimo del tesouro de la ontología.</p> <pre>public Term AddSynonymTerm(Term synonym)</pre> <p>Función que añade a la lista de sinónimos el término introducido por parámetros.</p> <pre>public bool RemoveSynonymTerm(Term synonym)</pre> <p>Función que elimina de la lista de sinónimos el término introducido por parámetros.</p> <pre>public CakeIntegerDictionary<RSHP> GetSynonymyRelationships(Term thesaurusTerm)</pre> <p>Función que devuelve un diccionario con las relaciones del tipo sinonimia que tenga el término pasado por parámetros.</p> <p>Funciones sobre los términos del tesouro:</p> <pre>public Term AddTerm(string TermName, Term_Tag TermType, Language TermLang = null, SemanticItem TermSemantic = null, bool TermSincronize = false, double TermLangImpact = 0.0, double TermDomImpact = 0.0)</pre> <p>Envía a la capa de datos la petición para añadir la información de un término en el tesouro de la ontología.</p> <pre>public bool Term_Remove(Term term)</pre> <p>Envía a la capa de datos la petición para eliminar la información de un término en el tesouro de la ontología.</p> <pre>public Term Term_Modify(Term thesaurusTerm, string TermName, string ClassificationCode, string ScopeNote, string HistoryNote, string Sources, GenderType Gender, NumberType Number, bool AllowChangeGender, bool AllowChangeNumber, Role Role, string language1 = "", string language2 = "", string language3 = "", string language4 = "", Term_Tag Kind = null)</pre> <p>Envía a la capa de datos la petición para modificar la información de un término en el tesouro de la ontología.</p> <pre>public bool Term_Merge(Term sourceTerm, Term parentSourceTerm, Term destinationTerm, ref List<Term> parentsAddedToTheDestination, ref List<Term> childrenAddedToTheDestination, ref List<Term> childrenAddedToTheSourceParent, ref List<Term> synonymsAddedToTheDestination, ref List<Term> relatedAddedToTheDestination, ref string errorMessage)</pre>
--

	<p>Función que fusiona los términos pasados por parámetros y devuelve los cambios producidos en el tesouro para su posterior refresco.</p> <pre>public bool MoveTerm(Term oldParent, Term child, Term newParent)</pre> <p>Función que implementa la funcionalidad drag and drop del tesouro con los términos pasados por parámetros.</p> <pre>public bool MakeRootTerm(Term term)</pre> <p>Función que convierte el término pasado por parámetros en un término raíz del tesouro.</p> <pre>public bool Term_UseFor(Term term, Term synonym)</pre> <p>Función que intercambia el término por su sinónimo, ambos pasados por parámetros, dentro del tesouro de la ontología.</p> <pre>public bool ExistsName(string TermName)</pre> <p>Función que comprueba si el nombre pasado por parámetros ya existe como nombre dentro del tesouro de la ontología.</p> <pre>public Term GetTerm(string termName)</pre> <p>Función que devuelve el término del tesouro de la ontología que coincida con la cadena de texto pasada por parámetros.</p> <pre>public bool LoadAllTerms()</pre> <p>Carga la lista de todos los términos del tesouro.</p> <pre>public List<Term> AllTerms()</pre> <p>Devuelve la lista de todos los términos del tesouro.</p> <pre>private Term AddTermToAllTerms(Term term)</pre> <p>Función que añade a la lista de todos los términos el término introducido por parámetros.</p> <pre>private bool RemoveTermFromAllTerms(Term term)</pre> <p>Función que elimina de la lista de todos los términos el término introducido por parámetros.</p> <pre>public List<List<Term>> GetBreadCrumb(Term term)</pre> <p>Función que devuelve en una lista de listas la miga de pan, es decir, los posibles caminos en los cuales se encuentra el término introducido por parámetros.</p> <pre>public DataTable GetBreadCrumbAsDataTable(Term term)</pre> <p>Función que devuelve en una tabla la miga de pan, es decir, los posibles caminos en los cuales se encuentra el término introducido por parámetros.</p> <p>Funciones sobre los términos raíces del tesouro:</p> <pre>public List<Term> RootsTerms()</pre>
--	--

	<p>Devuelve la lista de todos los términos raíces del tesouro.</p> <pre>public Term AddRootTerm(Term thesaurusTerm)</pre> <p>Función que añade a la lista de los términos raíces el término introducido por parámetros.</p> <pre>public bool RemoveRootTerm(Term thesaurusTerm)</pre> <p>Función que elimina de la lista de los términos raíces el término introducido por parámetros.</p> <pre>public bool IsRoot(Term rootCandidate)</pre> <p>Función que comprueba si el término introducido por parámetros es un término raíz del tesouro de la ontología.</p> <p>Funciones sobre la obtención de hijos del tesouro:</p> <pre>public CakeIntegerDictionary<Term> GetChildren(Term parent)</pre> <p>Función que devuelve en un diccionario todos los hijos del término introducido por parámetros.</p> <p>Funciones sobre la obtención de padres del tesouro:</p> <pre>public CakeIntegerDictionary<Term> GetParents(Term child)</pre> <p>Función que devuelve en un diccionario todos los padres del término introducido por parámetros.</p> <p>Funciones sobre la importación/ exportación del tesouro:</p> <pre>public ThesaurusRepository ImportProcess(string path, ref int numAddedTerms, ref int numAddedRelationships, string pathFileLog)</pre> <p>Función que importa un tesouro en la ontología pasándole por parámetros la ruta del fichero donde se encuentra el tesouro.</p> <pre>public bool ExportThesaurus(ThesaurusRepository myThesaurus)</pre> <p>Función que exporta el tesouro de la ontología pasándole por parámetros el tesouro.</p> <p>Funciones sobre la búsqueda web del tesouro:</p> <pre>public bool AddBrowser(string browserName, string url)</pre> <p>Función que añade un buscador web al tesouro de la ontología pasándole el nombre y la url del buscador.</p> <pre>public bool DeleteBrowser(string browserName)</pre>
--	--

	<p>Función que elimina un buscador web del tesouro de la ontología.</p> <pre>public bool searchTermInBrowser(Term thesaurusTerm, string browserName)</pre> <p>Función que busca en el buscador web del tesouro de la ontología el término introducido por parámetros.</p> <p>Funciones sobre el uso de tooltips en el tesouro:</p> <pre>public bool LoadToolTips()</pre> <p>Función que carga los tooltips configurados en el tesouro de la ontología.</p> <pre>public bool SaveToolTips()</pre> <p>Función que guarda una nueva configuración de los tooltips para el tesouro de la ontología.</p>
--	---

Tabla 167 - Diseño de ThesaurusRepository

4.5. Diseño físico de datos

En este capítulo se concreta la estructura física de datos que se utilizará en el sistema. Aunque es bastante similar al modelo de datos definido durante la fase de análisis, en este punto del proyecto se conocen con mayor exactitud los requisitos específicos del sistema, las particularidades del entorno tecnológico, y los distintos aspectos arquitectónicos.

4.5.1 Diseño del modelo físico de datos

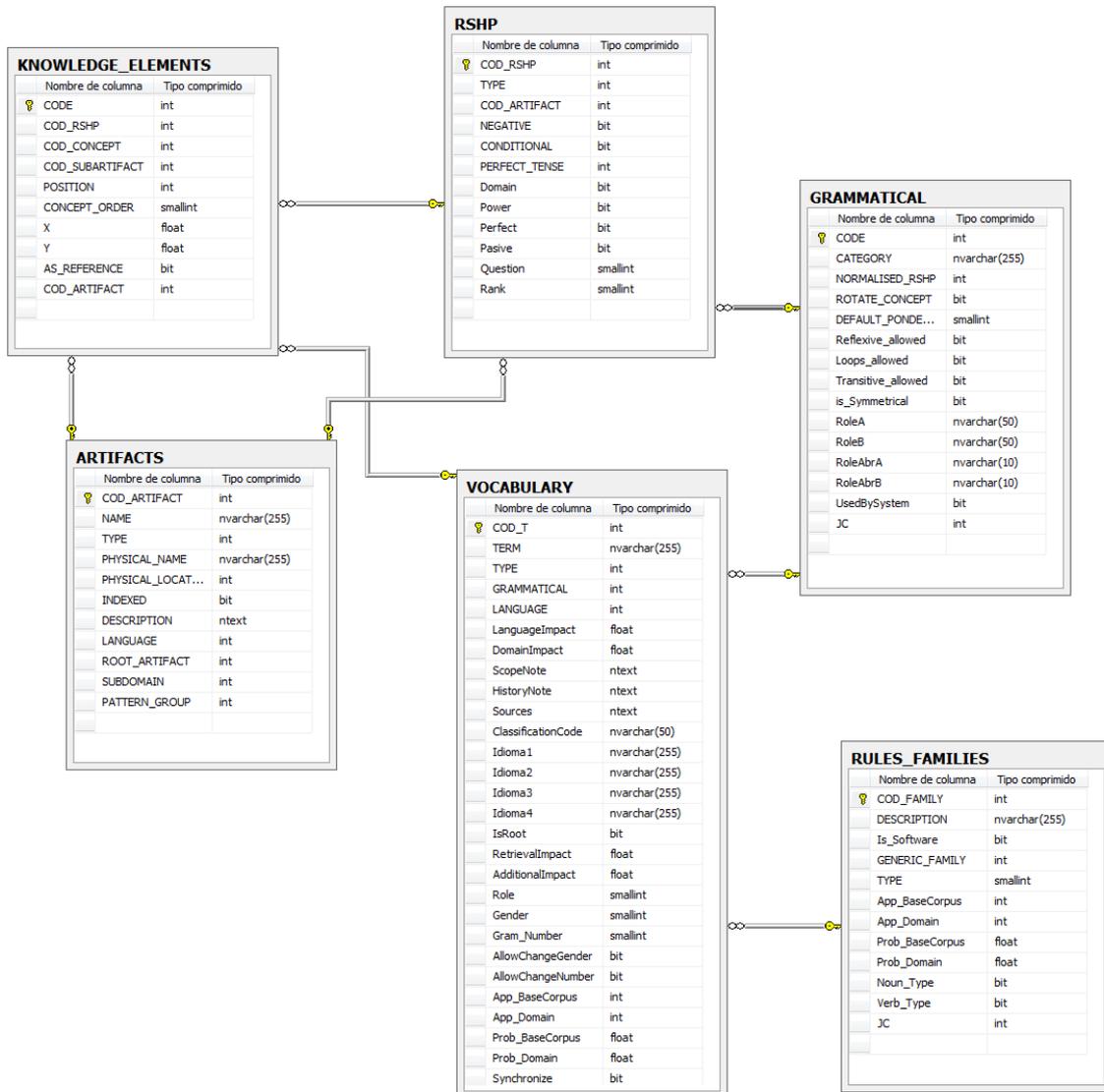


Ilustración 26 - Diseño físico del modelo de datos

En la ilustración se pueden observar las claves primarias de las tablas mediante el símbolo de una llave. Además se pueden observar las claves ajenas mediante las relaciones que existen entre las tablas, con lo que se puede apreciar fácilmente la estrecha relación entre las tablas Knowledge_Element y RSHP por ejemplo.

Recordar que los nombres de las tablas se corresponde con:

- Vocabulary a Términos
- Grammatical a Semánticas.
- Rules_Families a Etiquetas sintácticas.
- RSHP a relaciones.
- Knowledge_Elements a Términos de un artefacto o relación.
- Artifacts a artefactos.

Además se incluye el tipo de dato de cada atributo, por si fuera necesario en algún momento durante el desarrollo del sistema.

4.6. Diseño de migración y carga inicial de datos

Recordar que este proyecto es un módulo dentro de una aplicación más grande que necesita aunque en menor medida unos datos iniciales para ser ejecutada. Estos datos fueron especificados en el apartado de requisitos del sistema.

4.6.1. Especificación del entorno

Para disponer de los datos iniciales necesarios, el sistema debe tener instalado un gestor de bases de datos SQL Server 2008. En este caso, se va a usar SQL Server Management Studio como se indica en el capítulo 2 cuando se especifica la tecnología necesaria para el desarrollo.

4.6.2. Diseño del procedimiento de carga inicial

Para poder ejecutar correctamente la aplicación, es necesario disponer previamente de una base de datos correctamente preparada, con el fin de no tener que modificar las tablas de la base de datos de forma manual cada vez que se quiera hacer un despliegue de la aplicación.

Para ello se adjunta un fichero de copia de seguridad de base de datos (.bak), que contiene la información mínima para que la aplicación pueda ser ejecutada. Teniendo por un lado, el gestor SQL Server Management Studio y por otro, el fichero de base de datos, es necesario importarlo para crear nuestra base de datos. A continuación, se explica el proceso de importación:

1. Conectar con tu servidor de bases de datos.

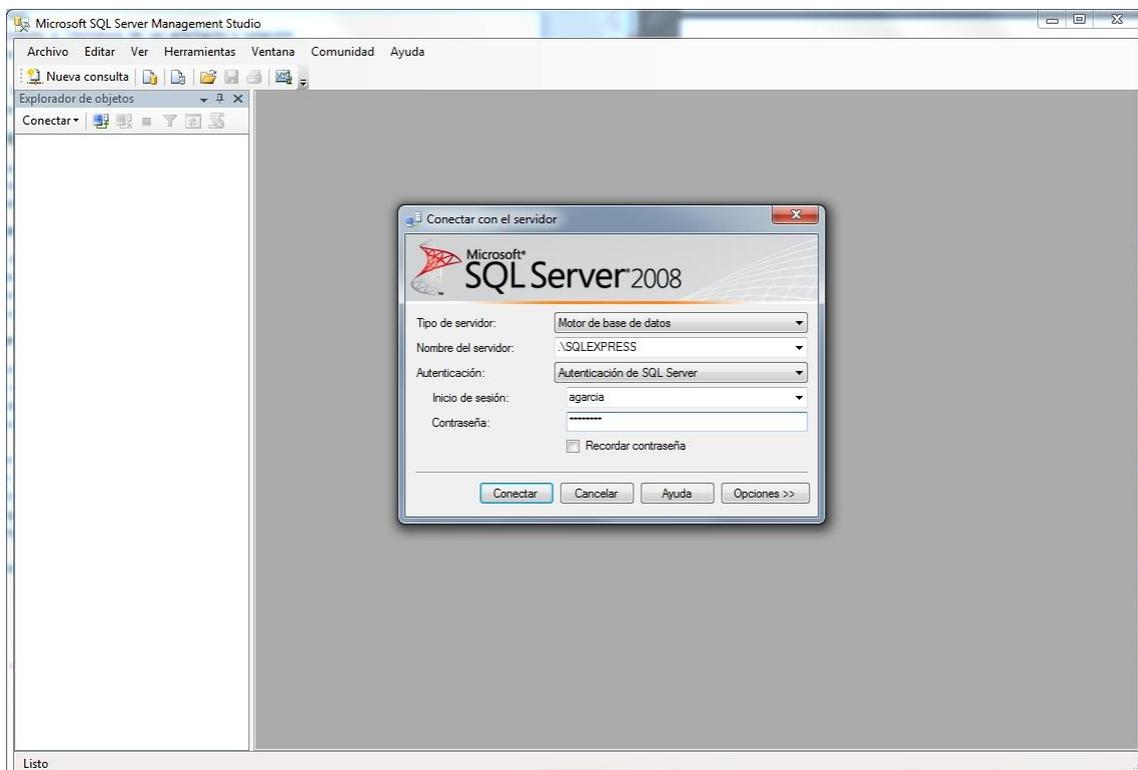


Ilustración 27 - Conexión con el servidor de SQL Server

2. Elegir la opción de 'Restaurar base de datos'

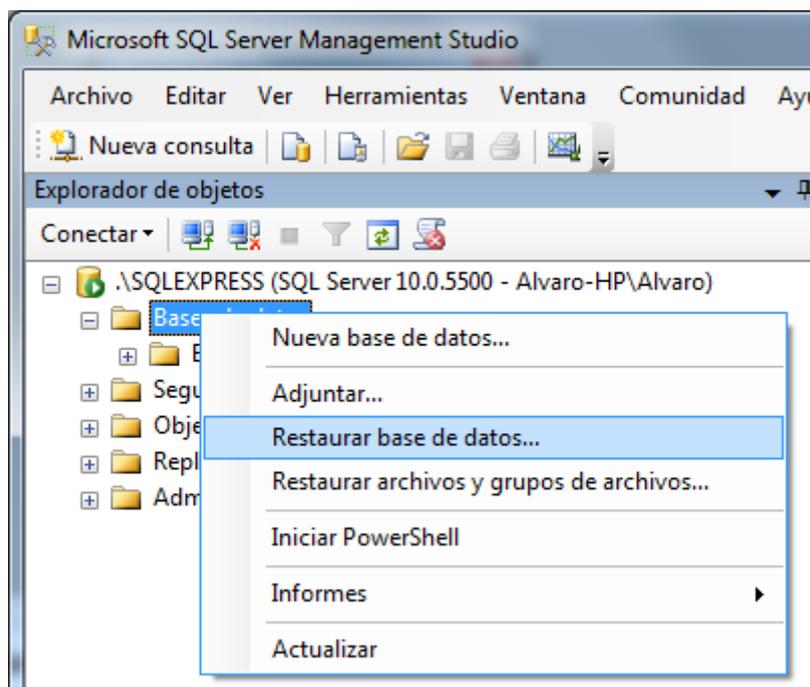


Ilustración 28 - Restaurar base de datos Paso 1

3. Elegir la ruta del fichero con extensión .bak necesario para la restauración.

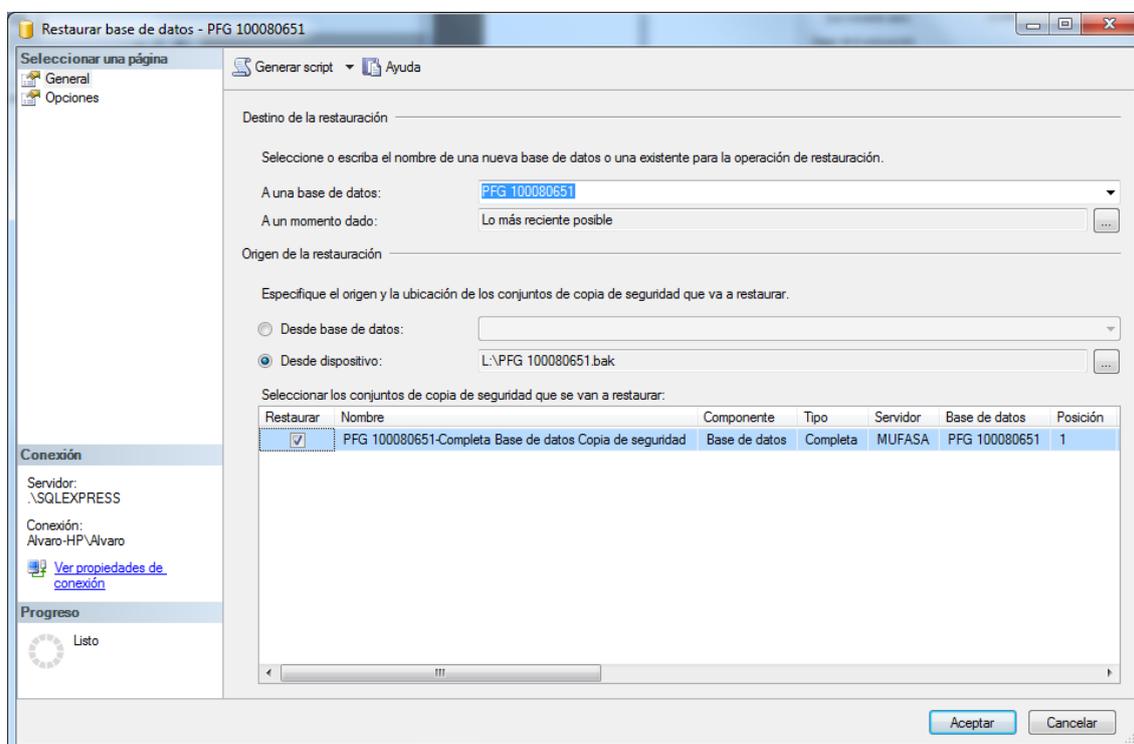


Ilustración 29 - Restaurar base de datos Paso 2

4. Comprobar que la base de datos se ha creado correctamente.

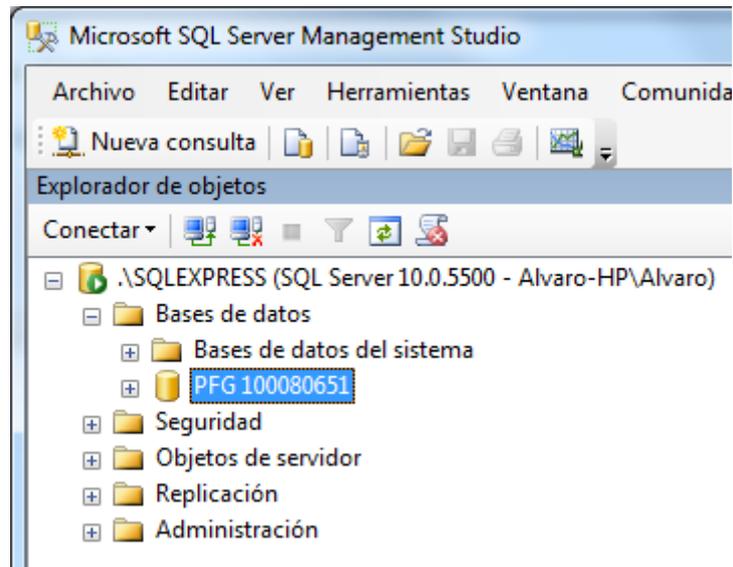


Ilustración 30 - Restauración de base de datos completada

4.7. Especificación técnica del plan de pruebas

La aplicación desarrollada está basada en un modelo de arquitectura de tres capas en la cual, por un lado se separa la capa de presentación y por otro lado la capa de negocio.

En la fase de análisis se definieron las pruebas de aceptación de la aplicación asociadas a la capa de presentación. En este apartado se van a definir las pruebas unitarias de todas las funciones públicas de la capa de negocio con el fin de obtener un producto lo más fiable posible.

4.7.1. Especificación del entorno de pruebas

El entorno hardware y software utilizado para llevar a cabo las pruebas unitarias es el mismo que se ha definido en el capítulo anterior de Análisis. Lo único diferente es que es la necesidad de utilizar el programa [NUnit](#), que es un software que permite llevar a cabo pruebas unitarias desarrolladas en el lenguaje C#, que es el utilizado en dicho proyecto.

Se utiliza la versión 2.6.1 de nunit, que a fecha actual es la versión estable más actualizada para el *framework* .NET 4.0 utilizado.

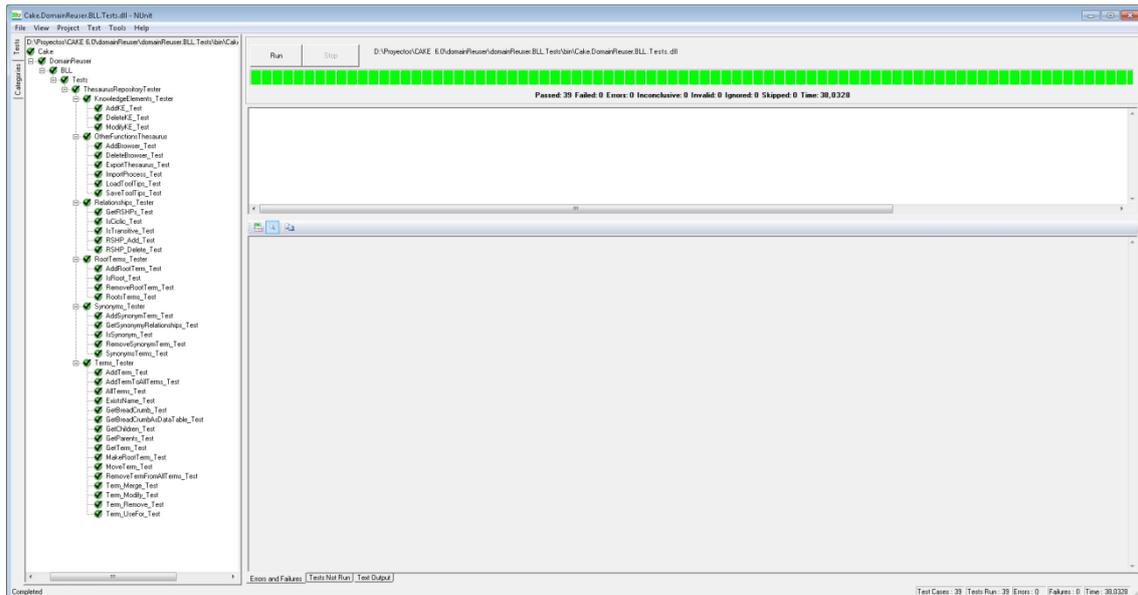


Ilustración 31 – Interfaz NUnit después de pasar las pruebas

4.7.2. Especificación técnica de niveles de prueba

Las pruebas unitarias que se van a desarrollar en este módulo van a ser provenientes de la clase que lleva toda la lógica del negocio del tesauro, es decir, la clase *ThesaurusRepository*, que en el apartado de análisis de clases de este mismo capítulo se especificaron sus funciones. Las demás clases de negocio ya tiene un módulo de pruebas realizado y no se especificaran aquí.

Los métodos de test probados son independientes al resto, es decir, el estado final de la base de datos al finalizar un test no debe ser el estado inicial del siguiente test, sino que todos parten de la misma información.

Para facilitar esta tarea, se utiliza una base de datos de desarrollo de Microsoft Access. Esta contiene el estado inicial de cada test. Además se tiene una función etiquetada como `[SetUp]` que se ejecuta automáticamente antes de cada test y cuya función es la de hacer una copia de la base de datos original y trabajar en el test con ella. Con esto se consigue que los test sean independientes unos de otros.

En cada prueba se necesitan variables que deben crearse y que en cada prueba se describe que se necesita.

4.7.2.1. Relaciones

PU-001: Obtención relaciones del tesouro	
Descripción:	Comprueba la obtención de todas las relaciones existentes entre dos términos del tesouro.
Función:	GetRSHPs(<i>Term</i> sourceTerm, <i>Term</i> destinationTerm)
Pasos:	Invocar llamada a la función GetRSHPs(sourceTerm, destinationTerm)
Especificaciones de entrada:	En ambas variables de entrada hay un término del tesouro.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un diccionario de relaciones que cumplen con los parámetros especificados.
Produce excepción:	No

Tabla 168 - PU-001: Obtención relaciones del tesouro

PU-002: Relación transitiva	
Descripción:	Comprueba que en la creación de una relación no exista transitividad entre términos.
Función:	IsTransitive(<i>Term</i> NewParent, <i>Term</i> NewChild, <i>Term</i> OldParent, <i>ref Term</i> problematicTerm)
Pasos:	Invocar llamada a la función: IsTransitive(NewParent, NewChild, OldParent, <i>ref</i> problematicTerm)
Especificaciones de entrada:	En las variables de entrada se introducen: nuevo padre de la relación, nuevo hijo de la relación, antiguo padre de la relación y un término vacío que se rellenara si la relación es transitiva.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si la relación es transitiva o no.
Produce excepción:	Sí

Tabla 169 - PU-002: Relación transitiva

PU-003: Relación cíclica	
Descripción:	Comprueba que en la creación de una relación no exista transitividad entre términos.
Función:	IsCiclic(<i>Term</i> sourceTerm, <i>Term</i> destinationTerm)
Pasos:	Invocar llamada a la función: IsCiclic(sourceTerm, destinationTerm)
Especificaciones de entrada:	En las variables de entrada se introducen: término origen de la relación, término destino de la relación.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si la relación es cíclica o no.
Produce excepción:	No

Tabla 170 - PU-003: Relación cíclica

PU-004: Añadir relación	
Descripción:	Comprueba la inserción de una relación dentro del tesauro.
Función:	RSHP_Add(<i>SemanticItem</i> Semantic, <i>Term</i> sourceTerm, <i>Term</i> destinationTerm)
Pasos:	Invocar a la función: RSHP_Add(Semantic, sourceTerm, destinationTerm)
Especificaciones de entrada:	En las variables de entrada se introducen: la semántica de la relación y los dos términos que se relacionan, término origen y destino.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera una relación entre los términos indicados por parámetros.
Produce excepción:	No

Tabla 171 - PU-004: Añadir relación

PU-005: Eliminar relación	
Descripción:	Comprueba la eliminación de una relación dentro del tesouro.
Función:	RSHP_Delete(<i>Term</i> sourceTerm, <i>Term</i> destinationTerm, <i>ref string</i> errorMessage)
Pasos:	Invocar a la función: RSHP_Delete(sourceTerm, destinationTerm, <i>ref</i> errorMessage)
Especificaciones de entrada:	En las variables de entrada se introducen: los dos términos que se relacionan, término origen y destino y una variable que nos mostrara un mensaje de error en el caso de no haber podido borrarla correctamente.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si la relación ha sido borrada con éxito o no.
Produce excepción:	Sí

Tabla 172 - PU-005: Eliminar relación

4.7.2.2. Elementos de las relaciones

PU-006: Añadir elemento de la relación	
Descripción:	Comprueba la inserción de un elemento de la relación dentro del tesouro.
Función:	AddKE(<i>CAKEElement</i> Owner, <i>Term</i> oTerm, <i>byte</i> Order, <i>int</i> Position)
Pasos:	Invocar a la función: AddKE(Owner, oTerm, Order, Position)
Especificaciones de entrada:	En las variables de entrada se introducen: el tipo de elemento que es (relación o artefacto), término asociado y dos variables más que nos indican la posición que tiene el elemento dentro de la relación o el artefacto.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un elemento de la relación con los parámetros indicados.
Produce excepción:	No

Tabla 173 - PU-006: Añadir elemento de la relación

PU-007: Modificar elemento de la relación	
Descripción:	Comprueba la modificación de un elemento de la relación dentro del tesouro.
Función:	ModifyKE(KE originalKE, Term oTerm)
Pasos:	Invocar a la función: ModifyKE(originalKE, oTerm)
Especificaciones de entrada:	En las variables de entrada se introducen: el elemento de la relación a modificar y un nuevo término con el cual asociar ese elemento, con ello se modifica los parámetros del elemento.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un elemento de la relación con los parámetros del nuevo término indicado.
Produce excepción:	No

Tabla 174 - PU-007: Modificar elemento de la relación

PU-008: Eliminar elemento de la relación	
Descripción:	Comprueba la modificación de un elemento de la relación dentro del tesouro.
Función:	DeleteKE(KE oKE)
Pasos:	Invocar a la función: DeleteKE(oKE)
Especificaciones de entrada:	En las variables de entrada se introduce: el elemento de la relación a eliminar.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el elemento de la relación ha sido borrado con éxito o no.
Produce excepción:	No

Tabla 175 - PU-008: Eliminar elemento de la relación

4.7.2.3. Sinónimos

PU-009: Obtener sinónimos del tesouro	
Descripción:	Comprueba la obtención de la lista de términos sinónimos del tesouro.
Función:	SynonymsTerms()
Pasos:	Invocar a la función: <code>SynonymsTerms()</code>
Especificaciones de entrada:	-
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera una lista con los sinónimos del tesouro.
Produce excepción:	No

Tabla 176 - PU-009: Obtener sinónimos del tesouro

PU-010: Término es sinónimo	
Descripción:	Comprueba si un término es sinónimo o no dentro del tesouro.
Función:	IsSynonym(<code>Term</code> <code>Term</code>)
Pasos:	Invocar a la función: <code>IsSynonym(Term)</code>
Especificaciones de entrada:	Como variable de entrada se introduce el término que se quiere comprobar si es sinónimo.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término es un sinónimo o no.
Produce excepción:	No

Tabla 177 - PU-010: Término es sinónimo

PU-011: Añadir sinónimo	
Descripción:	Comprueba que se ha añadido un término sinónimo a la lista de términos sinónimos del tesouro.
Función:	AddSynonymTerm(<i>Term</i> synonym)
Pasos:	Invocar a la función: AddSynonymTerm(synonym)
Especificaciones de entrada:	Como variable de entrada el término que se va a introducir en la lista.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un término que se debe comprobar en la lista de sinónimos.
Produce excepción:	No

Tabla 178 - PU-011: Añadir sinónimo

PU-012: Eliminar sinónimo	
Descripción:	Comprueba que se ha eliminado un término sinónimo de la lista de términos sinónimos del tesouro.
Función:	RemoveSynonymTerm(<i>Term</i> synonym)
Pasos:	Invocar a la función: RemoveSynonymTerm(synonym)
Especificaciones de entrada:	Como variable de entrada el término que se va a eliminar de la lista.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el sinónimo ha sido borrado con éxito o no.
Produce excepción:	No

Tabla 179 - PU-012: Eliminar sinónimo

PU-013: Obtener relación de sinonimia de un término	
Descripción:	Comprueba la obtención de una relación de sinonimia dado un término del tesouro.
Función:	GetSynonymyRelationships(Term thesaurusTerm)
Pasos:	Invocar a la función: GetSynonymyRelationships(thesaurusTerm)
Especificaciones de entrada:	Como variable de entrada se introduce un término del tesouro.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un diccionario con las relaciones que cumplieran los requisitos del parámetro de entrada.
Produce excepción:	No

Tabla 180 - PU-013: Obtener relación de sinonimia de un término

4.7.2.4. Términos

PU-014: Añadir un término	
Descripción:	Comprueba que se ha añadido un término al tesouro.
Función:	AddTerm(string TermName, Term_Tag TermType, Language TermLang = null, SemanticItem TermSemantic = null, bool TermSynchronize = false, double TermLangImpact = 0.0, double TermDomImpact = 0.0)
Pasos:	Invocar a la función: AddTerm(TermName, TermType, TermLang, TermSemantic, TermSynchronize, TermLangImpact, TermDomImpact)
Especificaciones de entrada:	Como variables de entrada: el nombre del término a crear, su etiqueta sintáctica y los siguientes son opcionales: su semántica, su lenguaje que se pone por defecto si no se introduce, y otras variables no deterministas en mi modulo.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un término cuyos atributos son los parámetros indicados.
Produce excepción:	No

Tabla 181 - PU-014: Añadir un término

PU-015: Eliminar un término	
Descripción:	Comprueba que se ha eliminado un término del tesauro.
Función:	Term_Remove(Term term)
Pasos:	Invocar a la función: Term_Remove(term)
Especificaciones de entrada:	Como variable de entrada el término que se quiere eliminar del tesauro.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término ha sido borrado con éxito o no.
Produce excepción:	No

Tabla 182 - PU-015: Eliminar un término

PU-016: Modificar un término	
Descripción:	Comprueba que se ha modificado un término del tesouro.
Función:	<code>Term_Modify(Term thesaurusTerm, string TermName, string ClassificationCode, string ScopeNote, string HistoryNote, string Sources, GenderType Gender, NumberType Number, bool AllowChangeGender, bool AllowChangeNumber, Role Role, string language1 = "", string language2 = "", string language3 = "", string language4 = "", Term_Tag Kind = null)</code>
Pasos:	Invocar a la función: <code>Term_Modify(thesaurusTerm, TermName, ClassificationCode, ScopeNote, HistoryNote, Sources, Gender, Number, AllowChangeGender, AllowChangeNumber, Role, language1 = "", language2 = "", language3 = "", language4 = "", Kind = null)</code>
Especificaciones de entrada:	Como variables de entrada: recibe variables pertenecientes a modificar la información del tesouro (término, nombre, código de clasificación, scopeNote, nota histórica) y variables de entrada con la información detalla del término (semántica, etiqueta sintáctica, lenguaje, entre otras).
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un término cuyos atributos son los nuevos introducidos por parámetros.
Produce excepción:	No

Tabla 183 - PU-016: Modificar un término

PU-017: Mezclar un término con otro	
Descripción:	Comprueba que un término del tesouro se ha fusionado con otro.
Función:	<pre>Term_Merge(Term sourceTerm, Term parentSourceTerm, Term destinationTerm, ref List<Term> parentsAddedToTheDestination, ref List<Term> childrenAddedToTheDestination, ref List<Term> childrenAddedToTheSourceParent, ref List<Term> synonymsAddedToTheDestination, ref List<Term> relatedAddedToTheDestination, ref string errorMessage)</pre>
Pasos:	Invocar a la función: <pre>Term_Merge(sourceTerm, parentSourceTerm, destinationTerm, ref parentsAddedToTheDestination, refchildrenAddedToTheDestination, refchildrenAddedToTheSourceParent, refsynonymsAddedToTheDestination, refrelatedAddedToTheDestination, ref errorMessage)</pre>
Especificaciones de entrada:	Como variables de entrada se tienen: término origen, término destino y padre del origen. Como variables auxiliares se tienen listas con los hijos, padres y tipos de relaciones que han sido cambiados en el proceso y que son necesarios para el refresco. Además se tienen una variable con un mensaje que controla los posibles fallos encontrados en el proceso.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término ha sido mezclado con éxito o no.
Produce excepción:	Si

Tabla 184 - PU-017: Mezclar un término con otro

PU-018: Drag and Drop de un término	
Descripción:	Comprueba que se copian o mueven términos del tesouro correctamente.
Función:	MoveTerm(<i>Term</i> oldParent, <i>Term</i> child, <i>Term</i> newParent)
Pasos:	Invocar a la función: MoveTerm(oldParent, child, newParent)
Especificaciones de entrada:	Como variables de entrada: el término a mover, el término destino y el antiguo padre del término origen.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término ha sido movido o copiado al término destino con éxito o no.
Produce excepción:	No

Tabla 185 - PU-018: Drag and Drop de un término

PU-019: Mover un término a la raíz	
Descripción:	Comprueba que un término del tesouro se ha movido a la raíz.
Función:	MakeRootTerm(<i>Term</i> term)
Pasos:	Invocar a la función: MakeRootTerm(term)
Especificaciones de entrada:	Como variable de entrada el término que se quiere mover a la raíz.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término ha sido movido a la raíz con éxito o no.
Produce excepción:	No

Tabla 186 - PU-019: Mover un término a la raíz

PU-020: Intercambiar sinónimo por término	
Descripción:	Comprobar que en una relación de sinonimia se intercambian los papeles.
Función:	Term_UseFor(<i>Term</i> term, <i>Term</i> synonym)
Pasos:	Invocar a la función: Term_UseFor(term, synonym)
Especificaciones de entrada:	Como variable de entrada se introduce el término y su sinónimo para darle la vuelta a la relación.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término ha sido intercambiado por uno de sus sinónimos con éxito o no.
Produce excepción:	No

Tabla 187 - PU-020: Intercambiar sinónimo por término

PU-021: Existe el término	
Descripción:	Comprueba si un término existe en el tesouro de términos de la ontología.
Función:	ExistsName(<i>string</i> TermName)
Pasos:	Invocar a la función: ExistsName(TermName)
Especificaciones de entrada:	Como variable de entrada se tiene una cadena con el nombre que se quiere comprobar si existe en el tesouro.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término con el nombre introducido por parámetros existe o no en el tesouro.
Produce excepción:	No

Tabla 188 - PU-021: Existe el término

PU-022: Obtener término	
Descripción:	Comprueba la obtención de un término del tesouro.
Función:	GetTerm(<i>string</i> termName)
Pasos:	Invocar a la función: GetTerm(termName)
Especificaciones de entrada:	Como variable de entrada una cadena de texto con el nombre del término.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un término que cumpla con el nombre introducido por parámetros.
Produce excepción:	No

Tabla 189 - PU-022: Obtener término

PU-023: Obtener todos los términos	
Descripción:	Comprueba la obtención de la lista de todos los términos del tesouro.
Función:	AllTerms()
Pasos:	Invocar a la función: AllTerms()
Especificaciones de entrada:	-
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera una lista con todos los términos del tesouro.
Produce excepción:	No

Tabla 190 - PU-023: Obtener todos los términos

PU-024: Obtener miga de pan	
Descripción:	Comprueba la obtención de la lista de listas con la miga de pan de un término del tesouro.
Función:	GetBreadCrumb(<i>Term</i> term)
Pasos:	Invocar a la función: GetBreadCrumb(term)
Especificaciones de entrada:	Como variable de entrada el término del que se quiera sacar la miga de pan.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera una lista de listas con todos los posibles caminos del término pasado por parámetros dentro del tesouro.
Produce excepción:	No

Tabla 191 - PU-024: Obtener miga de pan

4.7.2.5. Términos raíces

PU-025: Obtener términos raíces	
Descripción:	Comprueba la obtención de la lista de todos los términos raíces del tesouro.
Función:	RootsTerms()
Pasos:	Invocar a la función: RootsTerms()
Especificaciones de entrada:	-
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera una lista con todos los términos raíces del tesouro.
Produce excepción:	No

Tabla 192 - PU-025: Obtener términos raíces

PU-026: Añadir término raíz	
Descripción:	Comprueba que se ha añadido un término raíz a la lista de términos raíces del tesauro.
Función:	AddRootTerm(<i>Term</i> thesaurusTerm)
Pasos:	Invocar a la función: AddRootTerm(thesaurusTerm)
Especificaciones de entrada:	Como variable de entrada se introduce el término que se quiere añadir de la lista.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un término que se debe comprobar en la lista de términos raíces.
Produce excepción:	No

Tabla 193 - PU-026: Añadir término raíz

PU-027: Eliminar término raíz	
Descripción:	Comprueba que se ha eliminado un término raíz de la lista de términos raíces del tesauro.
Función:	RemoveRootTerm(<i>Term</i> thesaurusTerm)
Pasos:	Invocar a la función: RemoveRootTerm(thesaurusTerm)
Especificaciones de entrada:	Como variable de entrada se introduce el término que se quiere eliminar de la lista.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término raíz ha sido borrado con éxito o no.
Produce excepción:	No

Tabla 194 - PU-027: Eliminar término raíz

PU-028: Término es raíz	
Descripción:	Comprueba si un término es raíz o no dentro del tesauro.
Función:	IsRoot(<i>Term</i> rootCandidate)
Pasos:	Invocar a la función: IsRoot(rootCandidate)
Especificaciones de entrada:	Como variable de entrada se introduce el término del que se quiere saber si es un término raíz.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el término es un término raíz o no.
Produce excepción:	No

Tabla 195 - PU-028: Término es raíz

4.7.2.6. Hijos y padres

PU-029: Obtener hijos	
Descripción:	Comprueba la obtención de una lista con los hijos de un término del tesauro.
Función:	GetChildren(<i>Term</i> parent)
Pasos:	Invocar a la función: GetChildren(parent)
Especificaciones de entrada:	Como variable de entrada se introduce el término del cual se quiere averiguar sus hijos.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un diccionario con los términos hijos del término introducido por parámetros.
Produce excepción:	No

Tabla 196 - PU-029: Obtener hijos

PU-030: Obtener padres	
Descripción:	Comprueba la obtención de una lista con los padres de un término del tesouro.
Función:	GetParents(<i>Term</i> child)
Pasos:	Invocar a la función: GetParents(child)
Especificaciones de entrada:	Como variable de entrada se introduce el término del cual se quiere averiguar sus padres.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un diccionario con los términos padres del término introducido por parámetros.
Produce excepción:	No

Tabla 197 - PU-030: Obtener padres

4.7.2.7. Otras funcionalidades

PU-031: Importación	
Descripción:	Comprueba que se ha importado un tesouro de la ontología.
Función:	ImportProcess(<i>string</i> path, <i>ref int</i> numAddedTerms, <i>ref int</i> numAddedRelationships, <i>string</i> pathFileLog)
Pasos:	Invocar a la función: ImportProcess(path, <i>ref</i> numAddedTerms, <i>ref</i> numAddedRelationships, pathFileLog)
Especificaciones de entrada:	Como variable de entrada se introduce la ruta del fichero donde se encuentra el tesouro a importar, dos variables de control para saber qué cantidad de términos y relaciones se ha importado y la ruta de un log donde se introduce los fallos en la importación.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un objeto tesouro con el contenido del fichero introducido por parámetros.
Produce excepción:	No

Tabla 198 - PU-031: Importación

PU-032: Exportación	
Descripción:	Comprueba que se ha exportado un tesoro de la ontología.
Función:	ExportThesaurus(ThesaurusRepository myThesaurus)
Pasos:	Invocar a la función: ExportThesaurus(myThesaurus)
Especificaciones de entrada:	Como variable de entrada se introduce el tesoro que se quiere exportar.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el proceso de exportación ha tenido éxito o no.
Produce excepción:	No

Tabla 199 - PU-032: Exportación

PU-033: Añadir buscador	
Descripción:	Comprueba que se ha añadido un buscador web al tesoro de la ontología.
Función:	AddBrowser(string browserName, string url)
Pasos:	Invocar a la función: AddBrowser(browserName, url)
Especificaciones de entrada:	Como variables de entrada: el nombre que se le da al buscador y la url del buscador.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el buscador ha sido añadido con éxito o no.
Produce excepción:	No

Tabla 200 - PU-033: Añadir buscador

PU-034: Eliminar buscador	
Descripción:	Comprueba que se ha eliminado un buscador web al tesouro de la ontología.
Función:	DeleteBrowser(string browserName)
Pasos:	Invocar a la función: DeleteBrowser(browserName)
Especificaciones de entrada:	Como variable de entrada el nombre del buscador a eliminar.
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el buscador ha sido eliminado con éxito o no.
Produce excepción:	No

Tabla 201 - PU-034: Eliminar buscador

PU-035: Cargar información Tooltips	
Descripción:	Comprueba la carga de la información guardada en un archivo de configuración para mostrar los tooltips de términos en el tesouro.
Función:	LoadToolTips()
Pasos:	Invocar a la función: LoadToolTips()
Especificaciones de entrada:	-
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si la carga de los tooltips ha tenido éxito o no.
Produce excepción:	No

Tabla 202 - PU-035: Cargar información Tooltips

PU-036: Guardar configuración Tooltips	
Descripción:	Comprueba el salvado de la información guardada en un archivo de configuración para mostrar los tooltips de términos en el tesouro.
Función:	SaveToolTips()
Pasos:	Invocar a la función: SaveToolTips()
Especificaciones de entrada:	-
Especificaciones de salida:	Tras la llamada a la función de este caso de prueba se recupera un valor booleano que nos indica si el guardado de la nueva configuración de tooltips ha tenido éxito o no.
Produce excepción:	No

Tabla 203 - PU-036: Guardar configuración Tooltips

Capítulo 5. Planificación y Presupuesto

El objetivo de este capítulo es la de exponer de una manera detalla la planificación y el presupuesto establecidos para el proyecto con el fin de tener el proyecto controlado de una manera más económica y comercial.

5.1. Planificación

En este apartado se pretende recoger todo lo referente al proceso de planificación que se ha llevado a cabo para el desarrollo del proyecto.

Esta planificación se realiza en función de los 12 créditos [ECTS](#) que corresponden con el trabajo fin de grado impuesto en la normativa de la universidad. Recordar que cada crédito [ECTS](#) equivale a 25 o 30 horas de trabajo. Por lo tanto, el esfuerzo del trabajo fin de grado debe estar en alrededor de 300 a 360 horas de trabajo de alumno.

Esta estimación se lleva a cabo en función de la experiencia obtenida a lo largo de la carrera en las diferentes asignaturas y de la experiencia fuera de ella. Para la realización de la planificación no se usaran herramientas complejas; tan solo se utilizara [Gantt-Proyect](#) como se dijo en el capítulo de estado del arte para mejorar la comprensión de dicha planificación gracias a los diagramas que proporciona dicha herramienta. El por qué no usar herramientas profesionales para ello, se debe a que el proyecto no es lo suficientemente complejo para aplicar herramientas profesionales.

Para simplificar el reparto de las horas, se divide el trabajo a realizar en distintas fases:

Tarea	Fecha de inicio	Fecha de fin	Horas estimadas
Planificación	03/09/2012	05/09/2012	9
Presupuesto	05/09/2012	07/09/2012	7
Estado del arte	10/09/2012	28/09/2012	40
Análisis	01/10/2012	02/11/2012	45
Diseño	08/10/2012	30/11/2012	60
Implementación	15/10/2012	21/12/2012	85
Pruebas	29/10/2012	21/12/2012	35
Seguimiento	02/11/2012	28/12/2012	9
Conclusiones	24/12/2012	28/12/2012	7
TOTAL			297

Tabla 204 - Estimación de horas inicial

Como se puede observar en la tabla, la fecha de inicio del proyecto es el día 3 de Septiembre de 2012 y la fecha de fin estimada es el 28 de Diciembre de 2012. El proyecto se pretende acabar en 4 meses.

Continuación se detalla la jornada laboral que se pretende seguir a lo largo de los 4 meses de proyecto.

Mes	Días laborables	Jornada laboral	Total horas
Septiembre	20	3 horas / día	60
Octubre	23	4 horas / día	92
Noviembre	22	4 horas / día	88
Diciembre	19	3 horas / día	57
TOTAL			297

Tabla 205 - Reparto de horas por mes

Como se puede observar se dividen entre 3 y 4 horas de media las jornadas laborales entre los meses para el desarrollo del proyecto, aprovechando los meses intermedios que tienen más cantidad de días laborables y el proyecto está en su fase más compleja para introducir 1 hora más de trabajo por día.

Para mejorar aún más la planificación del proyecto se van a incluir en las siguientes tablas un desglose de la planificación de las tareas a desarrollar en cada mes y las horas que se dedicaran.

Para la planificación de Septiembre:

Septiembre				
Tarea	Semana 1 (5 días)	Semana 2 (5 días)	Semana 3 (5 días)	Semana 4 (5 días)
Planificación	9	0	0	0
Presupuesto	7	0	0	0
Estado del arte	0	14	13	13

Tabla 206 - Planificación de Septiembre

En este primer mes de trabajo se trata de investigar acerca del problema que se quiere solucionar y evaluar cómo solucionarlo. Además se establece la planificación y el presupuesto para todo el proyecto.

Para la planificación de Octubre:

Octubre					
Tarea	Semana 1 (5 días)	Semana 2 (5 días)	Semana 3 (5 días)	Semana 4 (5 días)	Semana 5 (3 días)
Análisis	9	9	8	8	8
Diseño	0	2	4	5	4
Implementación	0	0	2	5	2
Pruebas	0	0	0	3	1

Tabla 207 - Planificación de Octubre

Para Octubre, se pretende acometer el análisis del problema en su mayor medida, aunque ya se empezaran a introducir aspectos de diseño e incluso programa algunos trozos de código para poder probarlos y ver que el análisis del problema se está llevando acabo de la mejor manera posible.

Para la planificación de Noviembre:

Noviembre					
Tarea	Semana 1 (2 días)	Semana 2 (5 días)	Semana 3 (5 días)	Semana 4 (5 días)	Semana 5 (5 días)
Análisis	3	0	0	0	0
Diseño	10	10	10	10	5
Implementación	8	16	13	12	12
Pruebas	0	2	2	2	2
Seguimiento	1	1	1	1	1

Tabla 208 - Planificación de Noviembre

Para este mes, se cierra el proceso de análisis del proyecto ya que se tienen los requisitos bien definidos y se pone mucho esfuerzo en este mes a plantear y dar soluciones de diseño a la aplicación con la ayuda de la implementación y pruebas de las funcionalidades. Además se hacen seguimientos del proyecto con el tutor con el fin de ver si se está avanzando en la línea adecuada. Este mes es principalmente el mes más difícil ya que se pretende llegar a terminar el diseño de la aplicación y tener gran parte de la codificación al mismo tiempo hecha.

Para la planificación de Diciembre:

Diciembre					
Tarea	Semana 1 (4 días)	Semana 2 (5 días)	Semana 3 (5 días)	Semana 4 (4 días)	Semana 5 (1 días)
Implementación	5	5	5	0	0
Pruebas	8	8	7	0	0
Seguimiento	1	1	1	1	0
Conclusiones	3	3	1	0	0

Tabla 209 - Planificación de Diciembre

Para este último mes de proyecto, se siguen teniendo reuniones de seguimiento cada semana y se finaliza con la implementación de la aplicación. Durante las últimas semanas de implementación se hace un desarrollo más exhaustivo sobre las pruebas para garantizar la fiabilidad y calidad de la aplicación. Además se finaliza el contenido de la memoria del proyecto con la inclusión de los apartados finales de conclusiones y anexos del proyecto.

Para finalizar se incluye un [diagrama de Gantt](#) según la planificación descrita en el apartado:

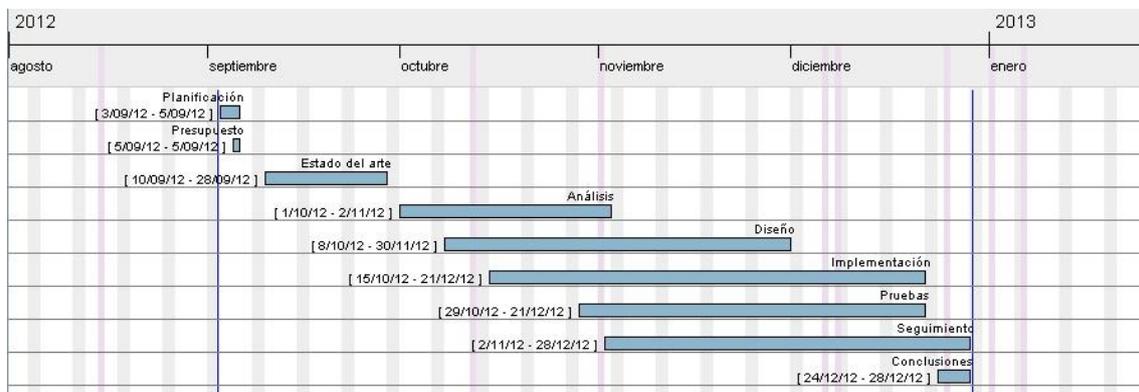


Ilustración 32 - Diagrama de Gantt según planificación

En esta imagen se puede observar claramente la planificación del proyecto viendo la duración de las diferentes tareas y la fecha de inicio y fin de proyecto.

5.2. Presupuesto

En este apartado se pretende resumir todo lo referente a los presupuestos que se han llevado a cabo para el desarrollo del proyecto.

A partir de lo expuesto en el apartado anterior sobre planificación, se van a definir los roles participantes en el proyecto y el sueldo bruto por hora de ellos, teniendo en cuenta que hay 14 pagas al año, los meses tienen una media de 22 días laborables y que las jornadas de trabajo son de 8 horas.

Rol	Sueldo bruto anual (€ / año)	Sueldo bruto por hora (€ / hora)
Jefe de proyecto	100.000	40,58
Analista	70.000	28,40
Diseñador	60.000	24,35
Programador	30.000	12,17
Tester	27.000	10,95

Tabla 210 - Salario bruto por rol

Una vez se tienen los roles se va a calcular el número de horas que trabaja cada uno de ellos con el fin de obtener los costes de personal. El resumen se muestra en la siguiente tabla:

	Jefe de proyecto	Analista	Diseñador	Programador	Tester
Planificación	9	0	0	0	0
Presupuesto	7	0	0	0	0
Estado del arte	10	25	5	0	0
Análisis	5	40	0	0	0
Diseño	10	0	50	0	0
Implementación	0	0	0	85	0
Pruebas	0	0	0	0	35
Seguimiento	4	2	1	1	1
Conclusiones	4	2	1	0	0

Horas / rol	49 horas	69 horas	57 horas	86 horas	36 horas
Coste / rol	1.988,42 €	1.959,6 €	1.387,95 €	1.046,62	394,2 €
				TOTAL	6.776,79 €

Tabla 211 - Coste de personal

Además de los costes del personal del equipo de desarrollo, se incluyen los costes de material, transporte y dietas que supone el proyecto. En cuanto al ordenador portátil utilizado, para calcular el valor se considera que es útil durante un año, es decir 365 días.

Material	Cantidad	Cálculo del coste	Coste
Ordenador portátil	1 (90 días de uso)	$(\text{CostePortatil}/\text{DíasUtiles}) * \text{DíasDeUso} = 999/365 * 90$	246,32 €
Windows 7	1	MSDN Academic Alliance	0 €
Visual Studio 2010	1	MSDN Academic Alliance	0 €
Office 2010	1	MSDN Academic Alliance	0 €
CD-ROM	5	$5 * 0,2$	1 €
Conexión a internet	5 meses	$\text{CosteMes} * \text{NúmeroMeses} = 38,6 * 5$	193 €
Transporte	5 meses	$\text{CosteMes} * \text{NúmeroMeses} = 35,5 * 5$	177,5 €
Dietas	4 meses	$\text{CosteMes} * \text{NúmeroMeses} = 40 * 4$	160 €
TOTAL			777,82 €

Tabla 212 - Costes de material, viajes, dietas y fungible

Por último se hace un resumen con el coste total del proyecto. A partir de los costes de personal más los de material, transporte, dietas y fungible, se añade un porcentaje para costes indirectos del proyecto que puedan no haberse tenido en cuenta (12%), y un margen de riesgo que en este caso será del 9%. Además el beneficio sacado al proyecto será de un 15 % y la cuota del IVA del 21 %.

Resumen del presupuesto del proyecto		
Concepto	Cantidad	Total
Costes directos	6776,79 € + 777,82 €	7.554,61 €
Costes indirectos (12 %)	906,55 €	8.461,16 €
Riesgo (9 %)	761, 5 €	9.222, 66 €
Beneficio (15%)	1.383,39 €	10.606,05 €
IVA (21%)	2.227,27 €	12.833,32 €
Precio final		12.833,32 €

Tabla 213 - Resumen de los costes y precio final

Como se puede observar el precio final del proyecto asciende a **12.833,32 €**.

Capítulo 6. Conclusiones y Trabajos Futuros

El objetivo de este capítulo es el de determinar las conclusiones sacadas y las experiencias obtenidas sobre el desarrollo del proyecto. Además de poder dejar algunas líneas de investigación y desarrollo para el futuro.

6.1. Conclusiones

6.1.1. Resultados obtenidos en la planificación

En este sub apartado se quiere comparar la planificación inicial estimada con la que al final ha sido, es decir, la planificación real que se ha ido haciendo a lo largo del proyecto.

En la planificación inicial se estimó el día de acabado del proyecto a día de 28 de Diciembre de 2012, dando un margen considerable a la fecha de entrega del proyecto que se prevé a mediados o finales de Enero de 2013.

Por un lado se ha producido un pequeño retraso en la fecha de finalización del proyecto, debido a que se tuvo que modificar el diseño de una de las partes del proyecto y por lo tanto la fase de diseño se desfaso una semana con respecto a lo planificado. Además eso conllevó también un retraso de una semana en las siguientes tareas por lo que la fecha real de finalización del proyecto ha sido el 4 de Enero de 2013.

Por otro lado, se han conseguido todos los objetivos fijados durante el análisis y diseño del sistema, y se ha construido al 100% el módulo del [knowledgeMANAGER](#) que se había pensado para el proyecto.

A continuación, se muestra un diagrama que compara los valores planificados y reales de la planificación:

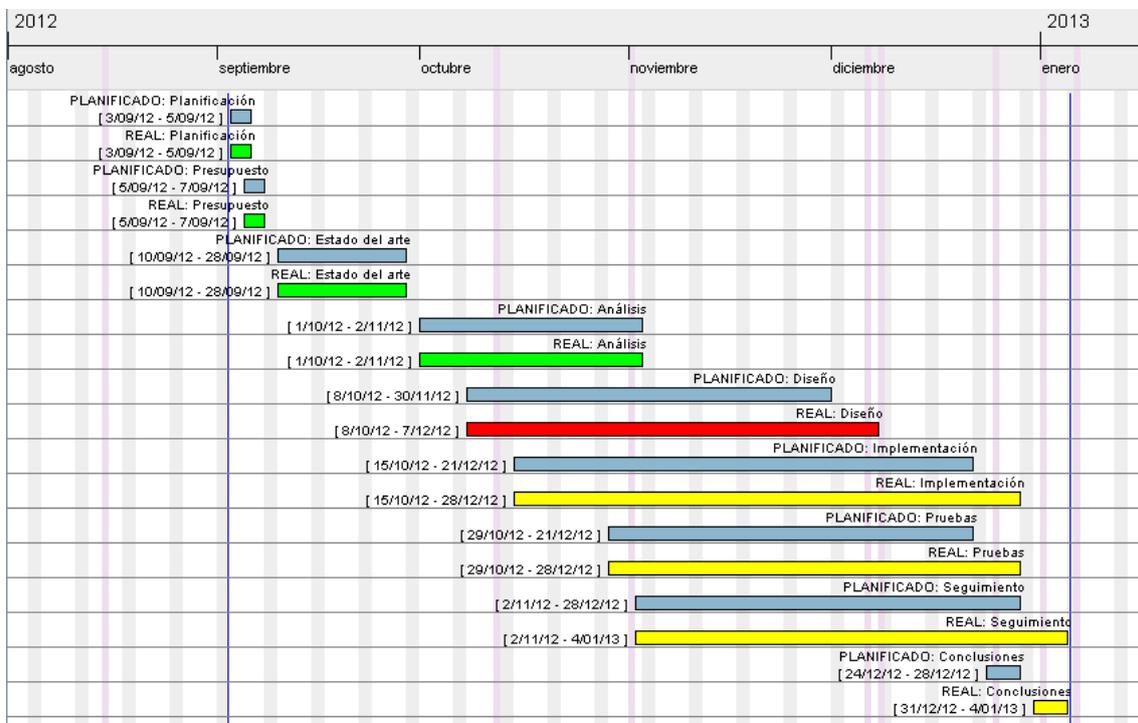


Ilustración 33 - Comparación Planificado-Real

Como se puede ver en el diagrama, las tareas azules representan las tareas planificadas, mientras que las verdes muestran que las reales cumplen con las planificadas (se están cumpliendo plazos); hasta la tarea de diseño que se muestra de color rojo, que es donde se ha comentado con anterioridad que se tuvo un problema con el rediseño de algunas funcionalidades de la aplicación que hicieron que el proyecto se retrasara. A partir de este punto las tareas en amarillo muestran las últimas tareas que han sufrido retraso por el problema del rediseño.

Para solucionar esto se ha dedicado un esfuerzo extra de horas en el proyecto que suman un total de 20 horas más de las planificadas que fueron 297 horas como se indica en el apartado de planificación. Por lo que el proyecto ha tenido un esfuerzo total de 317 horas, que a pesar de ello entra dentro de lo estipulado en la normativa del trabajo fin de grado en cuanto a los créditos [ECTS](#).

6.1.2. Evaluación del trabajo realizado

Este punto intenta englobar un poco el trabajo y el desarrollo realizado a lo largo de todo el proyecto. Además se quieren comprobar los resultados obtenidos en cuanto a la planificación y el desarrollo del proyecto.

Lo más importante y destacado a nivel global del aprendizaje del proyecto es la importancia que tiene el analizar el producto que se quiere desarrollar antes de ponerse a hacer cualquier otra cosa. Es fundamental dicha premisa para obtener un final feliz en un desarrollo, que no es más que obtener beneficios con el producto creado.

Gracias a la realización del TFG, he reforzado el conocimiento de todos los procesos de ingeniería, que se desarrollan a lo largo del ciclo de vida de un software, aprendidos a lo largo de los cuatro años de la carrera.

A nivel individual, he aprendido y adquirido mayor conocimiento en nuevas tecnologías de desarrollo indispensables para un desarrollo más cómodo y limpio. Además puedo decir que el trabajar dentro de unas normas de un departamento y amoldarse a su forma de trabajo, me han hecho ganar mayor profesionalidad y exigencia a la hora de realizar mis funciones, con lo que es de agradecer para mi futuro profesional. Con vistas a futuros trabajos, he de aprender de mis errores para no cometerlos más adelante.

En cuanto al desarrollo del proyecto, cabe mencionar que el principal problema encontrado es la de dimensionar de una manera correcta el problema desde el principio, ya que a priori se puede pensar que la funcionalidad está clara, pero cuando se empieza a desarrollar te das cuenta de cosas que hay que incluir y el proyecto puede irse de plazos. De todos modos, gracias al uso de una metodología ágil en la medida de lo posible dentro del desarrollo del proyecto, me ha ayudado a poder redimensionar el proyecto y evitar retrasos elevados.

Como satisfacción dentro de lo que he realizado, cabe destacar que ya está incorporado mi módulo en la aplicación [knowledgeMANAGER](#) y va a ser usado por un cliente real en poco tiempo.

6.2. Trabajos Futuros

Este punto intenta enumerar algunos puntos interesantes que se puedan seguir desarrollando en el futuro con respecto a mi trabajo.

La posibilidad de implementar el módulo del tesouro en una plataforma web o móvil en la que se pueda realizar algunas de las funcionalidades que pueden hacerse en la aplicación y con el fin de dar mayor usabilidad y disponibilidad al usuario.

La posibilidad de tener varios tesouros dentro de una misma ontología, con el fin de dar al usuario diferentes sistemas de dominio con un mismo vocabulario controlado.

Capítulo 7. Anexos

El objetivo de este capítulo es el de añadir información adicional y necesaria al proyecto para ampliar el conocimiento del mismo. Hay tres anexos: glosario, bibliografía y manual de usuario de la aplicación.

7.1. Glosario

Ankh SVN: es un cliente libre de Subversion, implementado como una fuente de control plug-in para Microsoft Visual Studio 2005, 2008 y 2010.

Asserts: son objetos o tipos de datos reconocidos por el [knowledgeMANAGER](#). Por ejemplo, documentos de indexación.

CAKE: modelo de arquitectura de datos y negocio utilizado por el departamento para el desarrollo de su producto.

ECTS (European Credit Transfer and Accumulation System (Sistema Europeo de Transferencia y Acumulación de Créditos): es un sistema utilizado por las universidades europeas para convalidar asignaturas y, dentro del denominado proceso de Bolonia cuantificar el trabajo relativo al estudiante que trabaja bajo los grados auspiciados por el Espacio Europeo de Educación Superior (EEES).

Diagrama de Gantt: es un gráfico cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Facade: Patrón de diseño que se utiliza para indicar que se trata de una interfaz.

FrameWork: conjunto de herramientas empleadas para llevar a cabo una acción.

KnowledgeMANAGER: es una aplicación de gestión del conocimiento desarrollada por el departamento y en la cual se integra mi modulo.

Light-Ontology: Este proceso se encarga de gestionar todo lo relacionado con el significado sintáctico y semántico que se le puede asignar a posteriori a un término o a una relación entre términos.

Metadato: un metadato es una unidad de información que describe otra unidad de información o dato.

MSDN Academic Alliance de Microsoft: es un programa dirigido a departamentos de universidades que permite poner a disposición de profesores y de los estudiantes la gama de herramientas de desarrollo, productos de servidor, sistemas operativos y bibliotecas de información de Microsoft con fines únicamente de investigación y docencia. [7]

NUnit: software que permite llevar a cabo pruebas unitarias desarrolladas en el lenguaje C#.

ObservableCollection: Representa una colección de datos dinámicos que proporciona notificaciones cuando se agregan o se quitan elementos o cuando se actualiza toda la lista.

Open – source: código libre que puede transmitirse sin necesidad de utilizar licencias que cuestan dinero.

Oracle Corporation: es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos Oracle hasta sistemas de gestión. [9]

Scrum: es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Sun Microsystems: fue una empresa informática que se dedicaba a vender ordenadores, componentes informáticos, software y servicios informáticos. Fue adquirida en el año 2009 por [Oracle Corporation](#), anteriormente parte de Silicon Valley, fabricante de semiconductores y software. [8]

Windows Presentation Foundation (WPF): proporciona a los programadores un modelo de programación unificado con el que compilar experiencias de cliente inteligentes de Windows, en las que se incorpora la interfaz de usuario, multimedia y documentos. [11]

7.2. Bibliografía

[1] [WEB1] (RAE, 2012)

- Enlace: <http://lema.rae.es/drae/?val=ontologia>
- Última visita: septiembre 2012

[2] [WEB2] Información sobre ontologías:

- Enlace: <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>
- Última visita: septiembre 2012

[3]: [WEB3] Gestor de ontologías DOME.

- Enlace: <http://dome.sourceforge.net/>
- Última visita: septiembre 2012

[4]: [WEB4] Gestor de ontologías SWOOP.

- Enlace: <http://code.google.com/p/swoop/>
- Última visita: septiembre 2012

[5]: [WEB5] Gestor de ontologías PROTEGE.

- Enlace: <http://protege.semanticweb.org/>
- Última visita: septiembre 2012

[6]: [WEB6] Lenguajes de programación más usados, evolución histórica.

- Enlace: <http://www.genbetadev.com/lenguajes-de-programacion/ranking-2011-de-lenguajes-mas-usados-java-se-mantiene-lider-y-objective-c-dobla-sus-resultados>
- Última visita: septiembre 2012

[7]: [WEB7] Laboratorio departamento de informática Universidad Carlos III

- Enlace: <http://www.lab.inf.uc3m.es/servicios/msdnaa>
- Última visita: septiembre 2012

[8]: [WEB8] Sun Microsystems

- Enlace: http://es.wikipedia.org/wiki/Sun_Microsystems
- Última visita: septiembre 2012

[9]: [WEB9] Oracle

- Enlace: http://es.wikipedia.org/wiki/Oracle_Corporation
- Última visita: septiembre 2012

[10]: [WEB10] Página principal de *The Reuse Company*.

- Enlace: <http://www.reusecompany.com/>
- Última visita: octubre 2012

[11]: [WEB11] Página MSDN de Microsoft.

- Enlace: <http://msdn.microsoft.com/es-es/library/ms754130.aspx>
- Última visita: noviembre 2012

[12]: Llorens J., Morato J., Genova G. RSHP: An information representation model based on relationships. In: Ernesto Damiani, Lakhmi C. Jain, Mauro Madravio (Eds.), *Soft Computing in Software Engineering (Studies in Fuzziness and Soft Computing Series, Vol. 159)*, Springer, pp 221-253. 2004.

[13]: Llorens, J., Fuentes J.M., Diaz I. RSHP: A Scheme to Classify Information in a Domain Analysis Environment. IEEE-AAAI- ICEIS 2001. International Conference on Enterprise Information Systems. Actas del congreso, pp. 686-690. Setúbal, Portugal. 2001

[14]: Llorens, J.; Fuentes, J.: *Computer Aided Knowledge Environment CAKE*. The Reuse Company, Winter 2004.

[15]: Llorens, J. *Definición de una metodología y una estructura de repositorio orientadas a la reutilización: el tesaurus de software*, 1996.

[16]: A. Fraga. *A Methodology for reusing any kind of knowledge at low cost: Universal Knowledge Reuse*. Advisors: Juan Llorens and Gonzalo Génova. Carlos III of Madrid University. *Computer Science Engineering*. 2010.

[17]: Morato, Jorge. *Hacia una web semántica social*, 2008.

[18]: Sanchez-Cuadrado, Sonia. *Definición de una metodología para la construcción automatizada de sistemas de organización del conocimiento*. Directores: José Antonio Moreiro González y Jorge Morato Lara. Universidad Carlos III de Madrid, Departamento de Biblioteconomía y Documentación. 2007.

[19]: A. Fraga. *A Methodology for reusing any kind of knowledge at low cost: Universal Knowledge Reuse*. Advisors: Juan Llorens and Gonzalo Génova. Carlos III of Madrid University. *Computer Science Engineering*. 2010.

7.3. Manual de usuario de la aplicación

En este anexo se intentan explicar cómo realizar algunas de las funciones más importantes del módulo del tesouro desarrollado para la aplicación [knowledgeMANAGER](#).

A continuación, se nombran las distintas acciones que se van a explicar en dicho manual de usuario. No se muestran detalles para la instalación y conexión con la base de datos, ya que es necesario adquirir una licencia por parte de *The Reuse Company*:

1. Formulario inicial
2. Creación de elementos de un tesouro
 - Término
 - Relación de jerarquía (hijo)
 - Sinónimo
3. Modificación de un término
4. Búsqueda de un término
5. Copiar y mover términos
6. Borrado de elementos del tesouro
 - Término
 - Relación
7. Importación y exportación de un tesouro

7.3.1. Formulario Inicial

Una vez se introducen correctamente las credenciales para acceder a la aplicación, se accede a un primer formulario en el que encontramos un menú horizontal superior con cuatro partes diferenciadas de la aplicación: Gestión del dominio, Subdominios, Indexación y recuperación, Documentos y configuración como se ha mencionado en el estado del arte de este documento cuando se hablaba de la aplicación. Nuestro modulo pertenece a subdominios donde se encuentra el tesouro de la ontología.

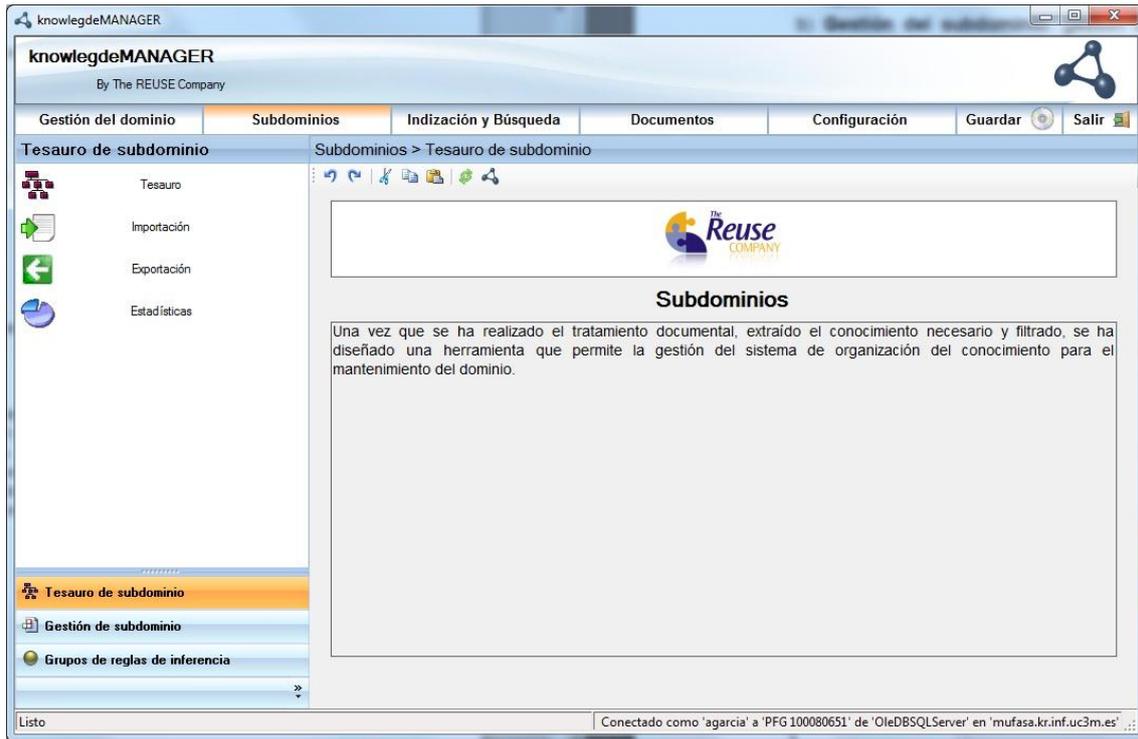


Ilustración 34 - Formulario principal

7.3.2. Creación de elementos de un tesouro

En un tesouro se pueden crear términos y relaciones. En el siguiente apartado se explicara al detalle cómo se crea un término y las relaciones más usadas del tesouro, jerarquía y sinonimia. La creación de otro tipo de relaciones es similar por lo tanto se han elegido estas como modelo de explicación.

El primer paso en todas ellas es abrir el menú contextual del tesouro:

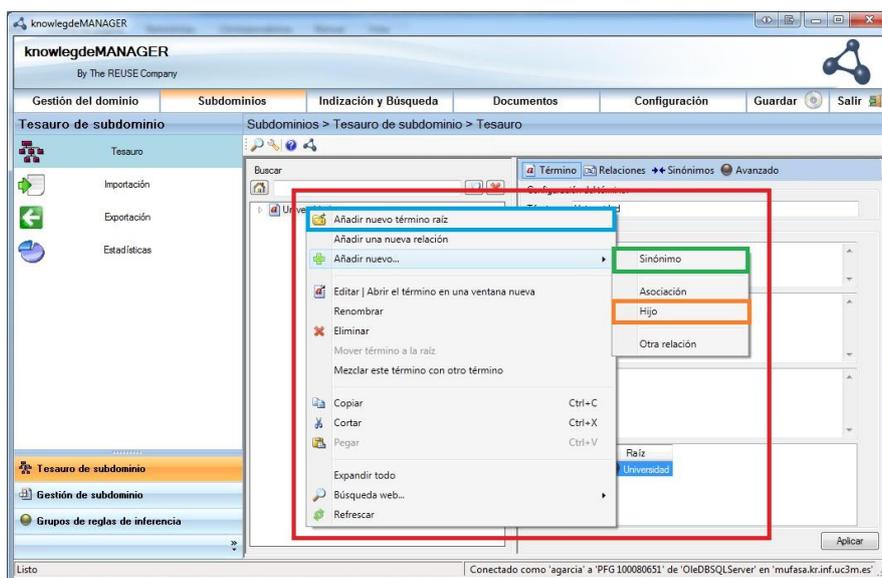


Ilustración 35 – Creación de un elemento del tesouro Paso 1

Como se puede ver en la imagen en color azul, para crear un término, se elige la opción de Añadir un nuevo término raíz del tesouro. De color naranja, para crear una relación de jerarquía, es decir, padre e hijo, se elige la opción de Añadir nuevo hijo. De color verde, para crear una relación de sinonimia, es decir, un sinónimo, se elige la opción de Añadir nuevo sinónimo:

A continuación nos saldrán, formularios en los cuales rellenar la información necesaria para la creación de estos elementos.

Para la creación del término y su hijo el formulario es:

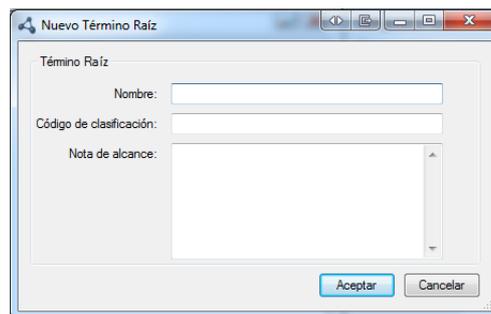


Ilustración 36 - Creación de un elemento del tesouro Paso 2

Para la creación del sinónimo el formulario es:

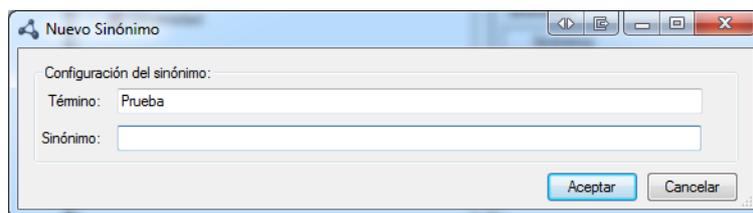


Ilustración 37 - Creación de un elemento del tesouro Paso 2

Para finalizar, se pulsa en aceptar y se ven los cambios en el tesouro.

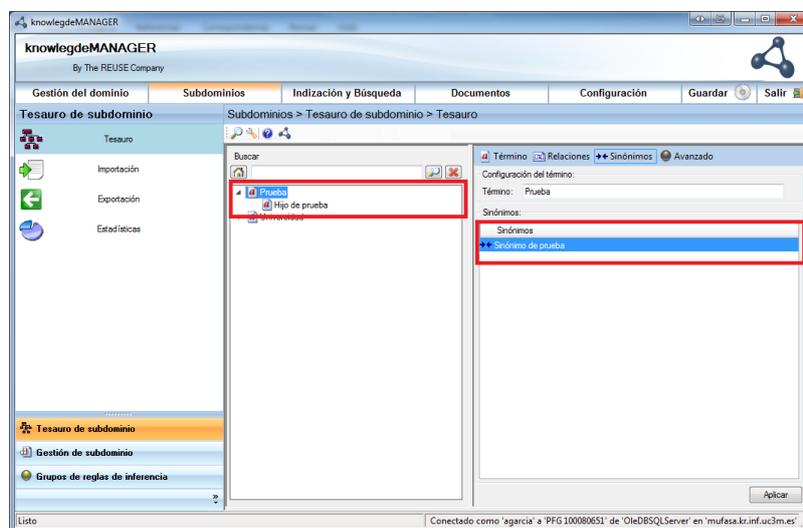


Ilustración 38 - Creación de un elemento del tesouro Paso 3

Como se puede observar en color rojo en la ilustración los cambios del tesouro son refrescados en el formulario inicial.

7.3.3. Modificación de un término tesouro

En este apartado se va a comentar como modificar información acerca de un término del tesouro ya que es el elemento básico de un tesouro.

El primer paso es abrir en el menú contextual del tesouro, y elegir la opción de editar/mostrar información del término. Como ya se mostró antes la imagen del menú contextual y por no agrandar de imágenes este manual, esta y las siguientes veces que se necesite el menú contextual para alguna acción no se mostrara a no ser que se vea necesario.

El segundo paso es modificar el formulario con la nueva información que se quiera del término.

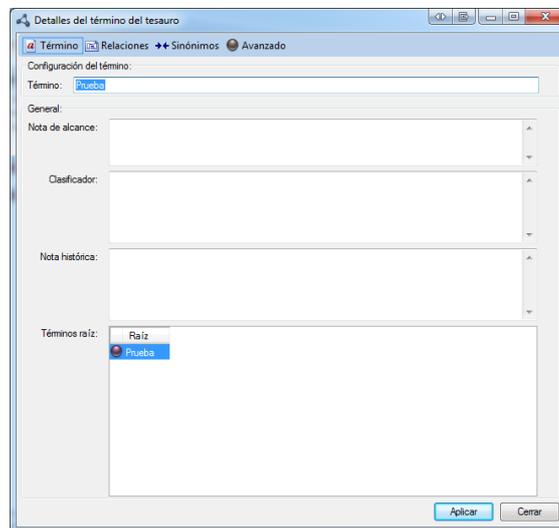


Ilustración 39 - Modificación de un término Paso 2

Tercer paso, visualización de los cambios en el tesouro:

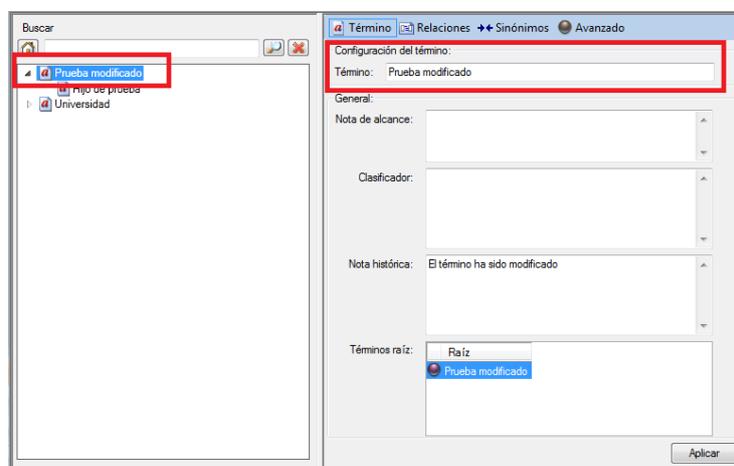


Ilustración 40 - Modificación de un término Paso 3

Como se refleja en la imagen los cambios han sido refrescados en el tesouro en ambas pantallas con éxito.

7.3.4. Búsqueda de un término el tesouro

En este apartado se describe lo fácil e intuitiva que es la búsqueda de un término dentro del tesouro.

En la pantalla inicial del tesouro, en la parte superior de este, se encuentra una caja de texto en la cual se busca por nombre del término dentro del tesouro.

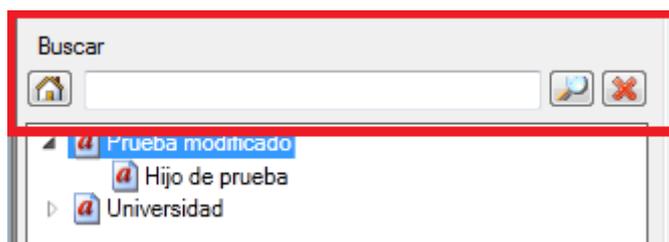


Ilustración 41 - Búsqueda de un término del tesouro Paso 1

Tan solo se debe escribir el nombre del término que se quiera buscar y se mostraran por pantalla los resultados, esta vez sin vista jerárquica, es decir, sin visión de hijos y padres.

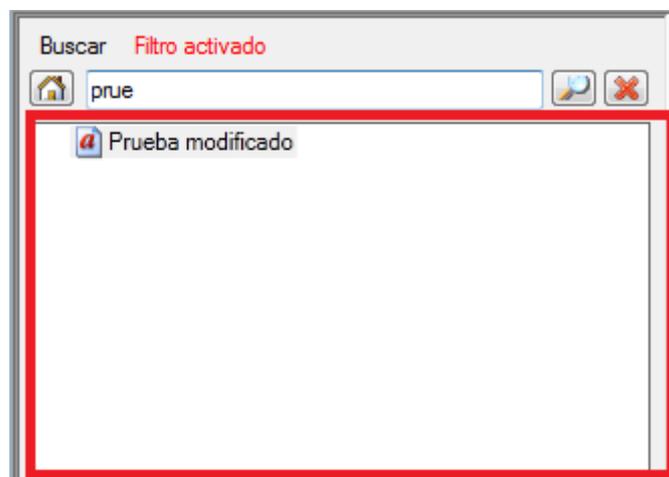


Ilustración 42 - Búsqueda de un término del tesouro Paso 2

Como se puede observar en la imagen se ve que estas en la vista filtrada con los resultados de la búsqueda introducida.

7.3.5. Copiar y mover términos

Esta opción es muy útil en el tesoro ya que da mucha velocidad al usuario a la hora de crear un tesoro. Para mover y copiar términos dentro del tesoro se tienen varias opciones, como por ejemplo mover un término a la raíz, drag and drop de términos o las simples teclas rápidas de copiar, cortar, pegar (ctrl c, ctrl x, ctrl v).

En este apartado se mostrara como hacer estas acciones de una manera sencilla.

Primer paso es elegir el término que se quiere copiar o mover dentro del tesoro.

Como segundo paso se puede utilizar las teclas rápidas de copiar pegar o el drag and drop. El problema de ello es que visualmente no se aprecia por ello no hay pantallazo de esto.

Antes de llevar a cabo la acción de mover o copiar se le pregunta al usuario si realmente desea hacer la operación y se le indica el término que quiere copiar y donde lo quiere copiar.

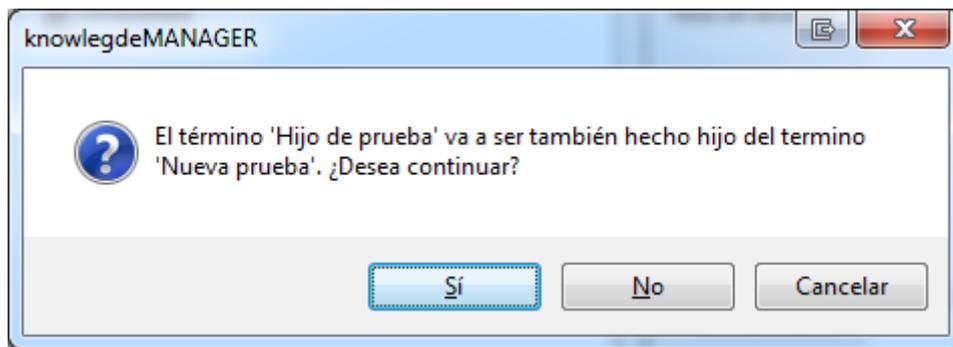


Ilustración 43 - Copiar y mover términos Paso 2

Una vez finalizado, los cambios se ven el tesoro si se escogió la opción de Sí.

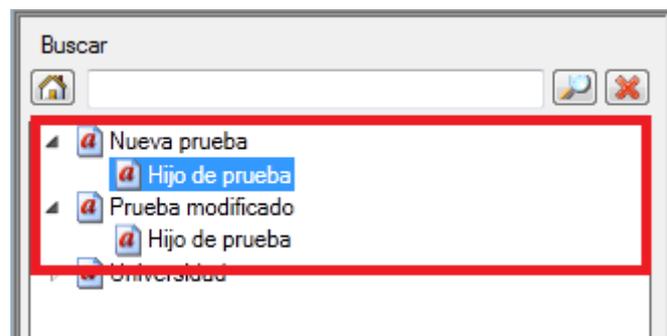


Ilustración 44 - Copiar y mover términos Paso 3

7.3.6. Borrado de elementos de un tesaurus

En este apartado se pretende visualizar el proceso de borrado de los elementos del tesaurus. En nuestro caso se va a eliminar un término, un sinónimo y una relación que son elementos característicos del árbol.

El primer paso es como de costumbre es seleccionar el término que se va a eliminar. Para la eliminación una relación o un sinónimo, se puede hacer de manera sencilla desde la pestaña de sinónimos o relaciones del término.

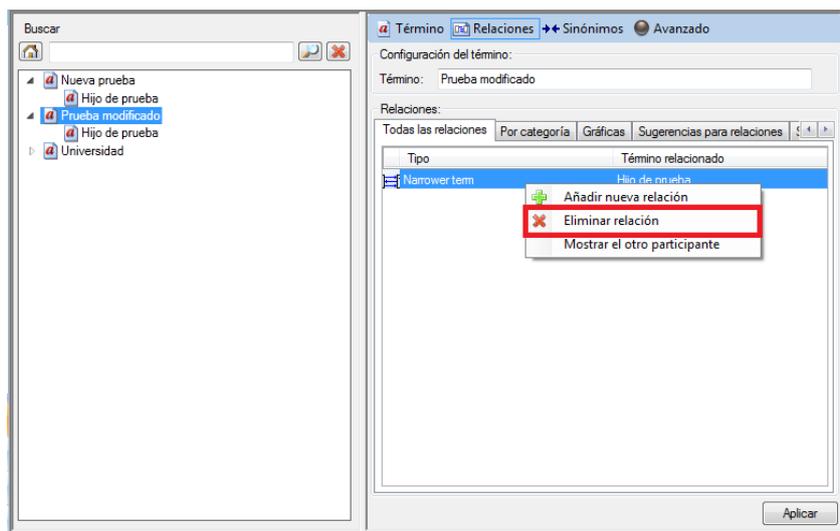


Ilustración 45 - Eliminación de una relación Paso 1

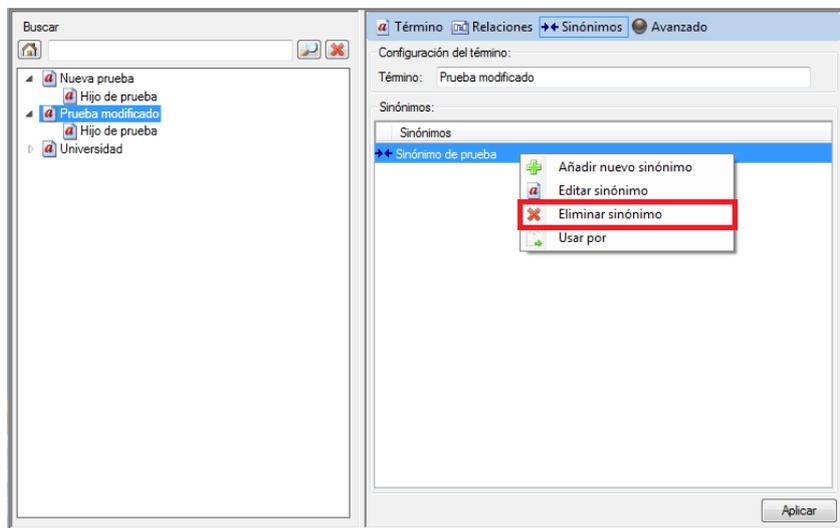


Ilustración 46 - Eliminación de un sinónimo Paso 1

El paso siguiente es confirmar el borrado de la relación o del sinónimo a borrar. Para el ejemplo se muestra el borrado del sinónimo pero son iguales simplemente cambia el mensaje..

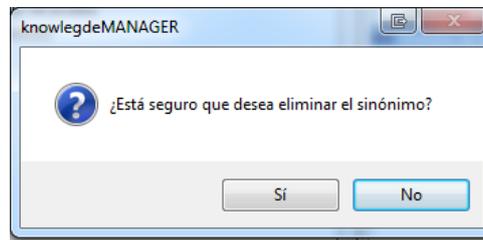


Ilustración 47 - Borrado de un sinónimo o relación Paso 2

Si lo que se está borrando es un término, entonces se debe elegir una de las opciones de borrado que se muestran en la siguiente pantalla. Se dan dichas opciones para dar más posibilidades al usuario de qué es lo que realmente quiere hacer.

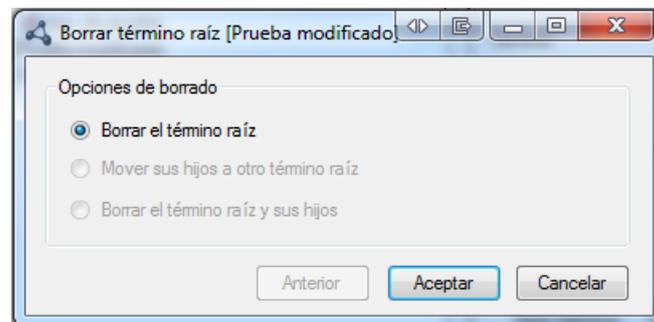


Ilustración 48 - Borrado de un término Paso 2

El último paso es visualizar el árbol con el refresco actualizado de las acciones llevadas a cabo en este apartado.

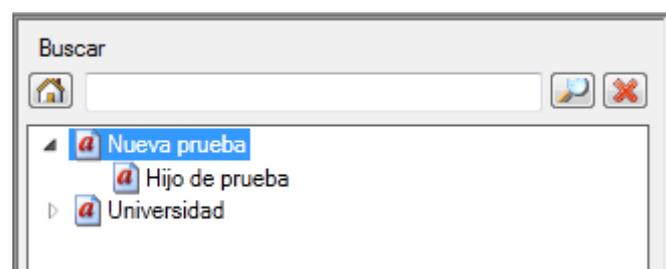


Ilustración 49 – Borrado de un elemento del tesouro Paso 3

Lo que se ha hecho es eliminar el sinónimo del término 'Prueba modificado', después se ha borrado la relación de jerarquía que tenía el término borrando así su hijo de nombre 'Hijo de prueba'. Y para finalizar se ha borrado el término 'Prueba modificado'. El resultado es que ha desaparecido del árbol ese término y sus relaciones. Por consiguiente Hijo de prueba sigue en el tesouro porque en el apartado anterior de copia de términos se había copiado en otro padre.

7.3.7. Importación y exportación de un tesouro

Para la exportación, simplemente se selecciona la opción de exportación del menú de la izquierda; a continuación, se elige la opción de exportación que se quiera, es decir, en que formato se quiere exportar y se pulsa en el botón de Exportar.

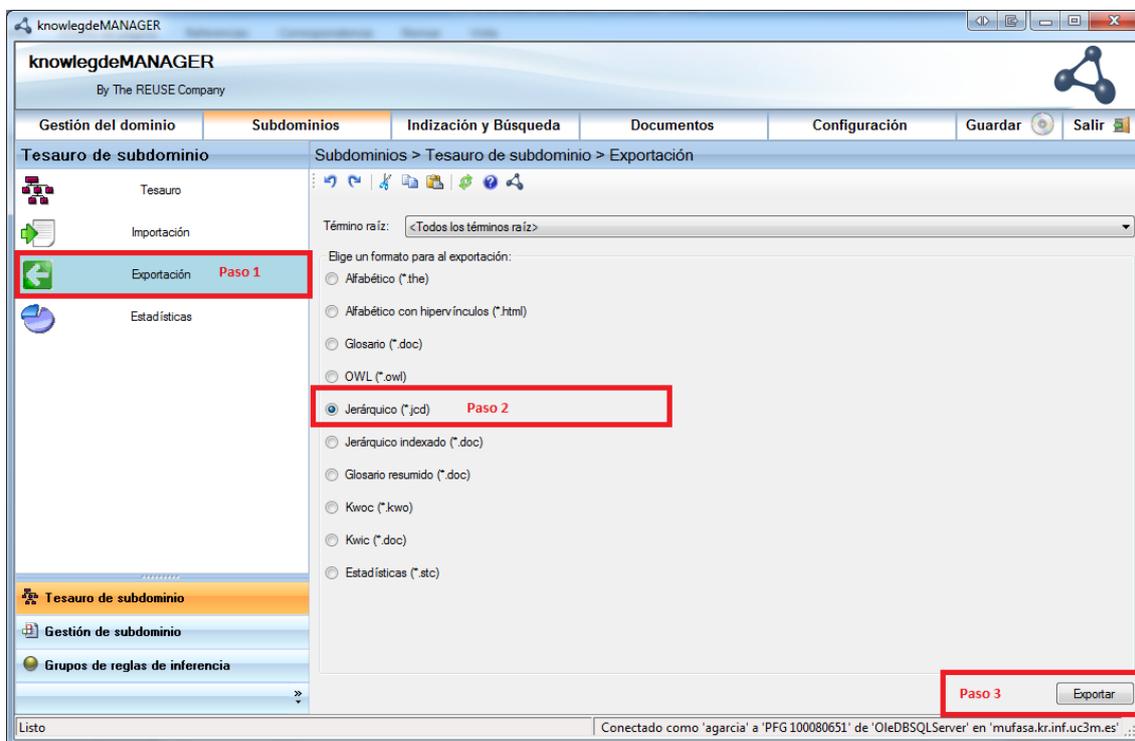


Ilustración 50 - Exportación del tesouro Paso 1

Se elige el nombre del fichero y donde se va a guardar (esto es la ventana de Windows 7 que sale al utilizar la opción de búsqueda de directorio). Se espera a que concluya. Al finalizar, se muestra una dialogo que muestra que el proceso se ha realizado con éxito y si se quiere abrir el fichero generado.

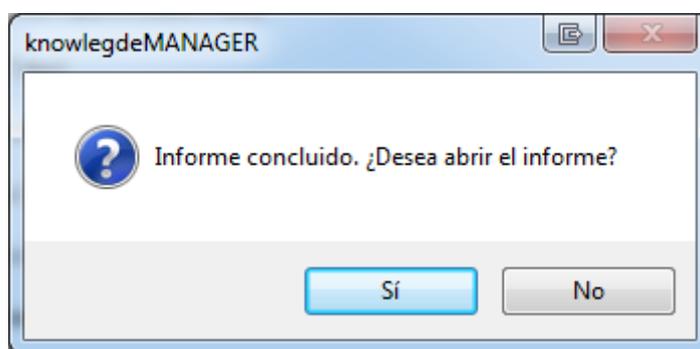


Ilustración 51 - Exportación del tesouro Paso 2

Para la importación, se van a borrar todos los términos del tesouro quedando este de la siguiente manera:



Ilustración 52 - Importación del tesouro Paso 1

Los pasos a seguir son simplemente se selecciona la opción de importación del menú de la izquierda; a continuación, se pulsa en el botón de Importar y se elige un fichero con extensión *.the o *.jcd que son las extensiones que entiende el tesouro y se pulsa en aceptar.

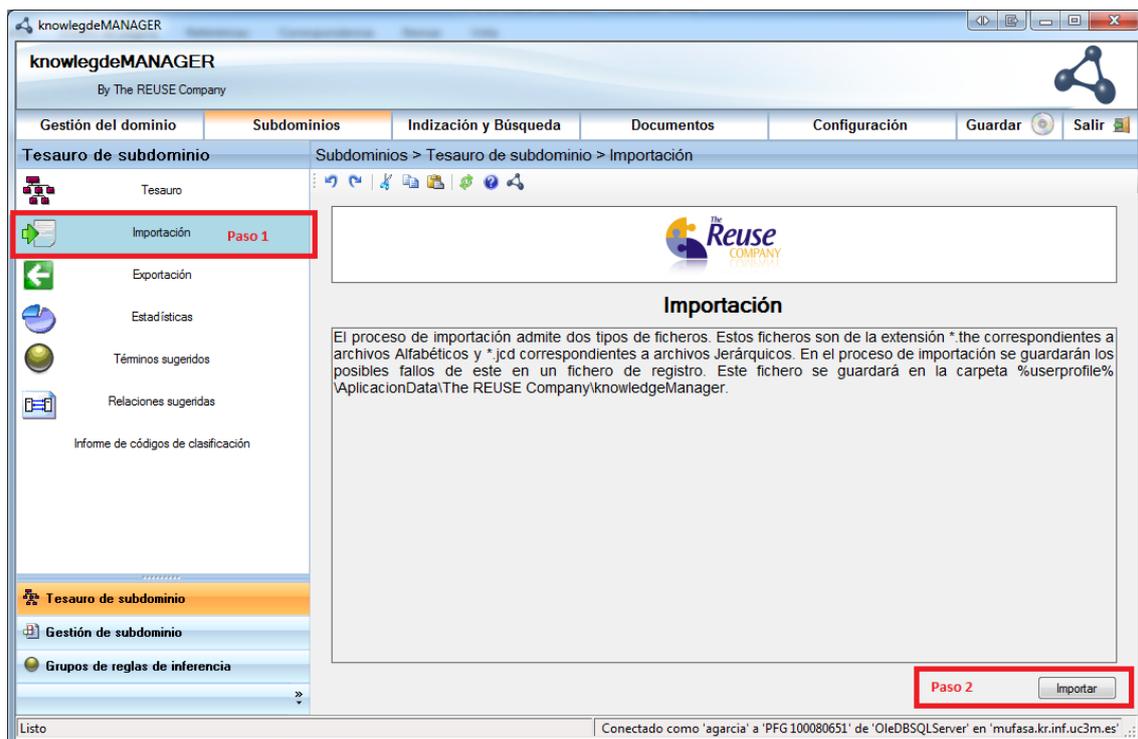


Ilustración 53 - Importación del tesouro Paso 2

Se espera a que concluya. Al finalizar, se muestra una dialogo que muestra que el proceso se ha realizado con éxito y nos muestra un numero con los términos y relaciones que han sido añadidos al tesouro.

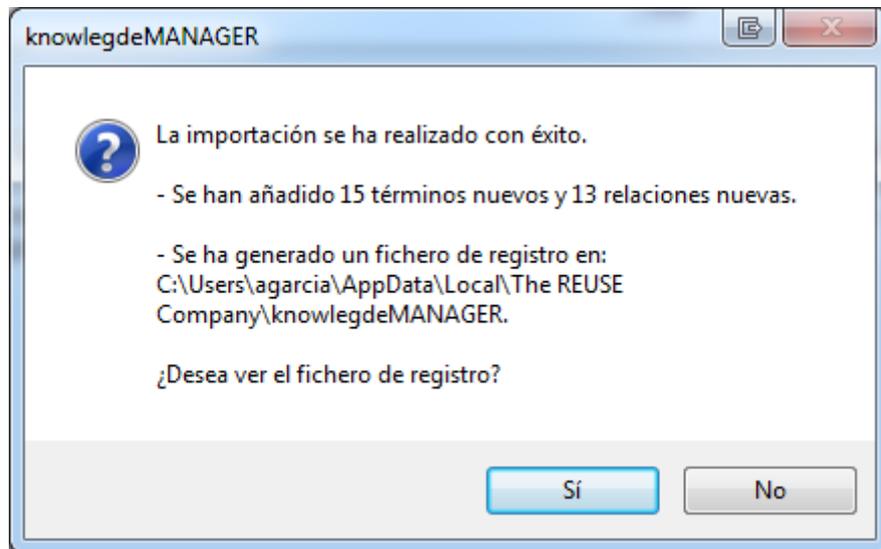


Ilustración 54 - Importación de un tesouro Paso 3

Se da la posibilidad de ver un registro de por qué algunos términos no han podido ser insertados en el tesouro.

Lo último que se debe hacer es comprobar que el tesouro se visualiza correctamente según la importación.

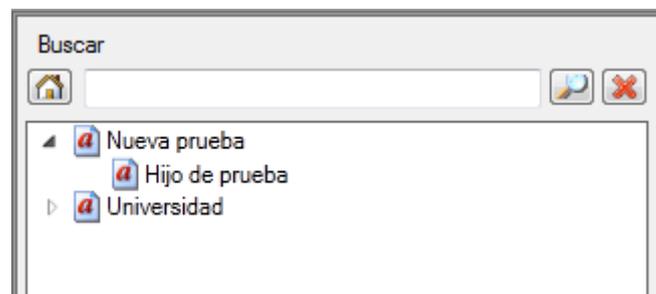


Ilustración 55 - Importación de un tesouro Paso 4

Con esto concluye el manual de usuario con la mayoría de las principales funcionalidades descritas.