



Universidad del Bío-Bío

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

Sede Concepción

“Mejora de aplicación para la creación y edición de taxonomías de información contextual, como apoyo a sistemas de recomendación sensibles al contexto”

Memoria para optar al título de Ingeniero Civil en Informática

Autor: Marcelo Andrés Parra Bittner

Profesor Guía: Pedro Campos Soto

2018

Agradecimientos

En primer lugar, quiero agradecer a mi profesor guía el Dr. Pedro Campos Soto por sus enseñanzas, motivación y apoyo a lo largo del semestre y por sobre todo la confianza puesta en mí, el cual fue fundamental para sacar adelante mi proyecto de título.

Agradecer a mis compañeros de carrera, a mis amigos y a mi banda, por la amistad, ánimos y consejos recibidos a lo largo de este periodo.

Un agradecimiento muy especial merece mi mamá, papá, hermano y familia. Por todo el cariño, la confianza y la motivación para seguir siempre adelante.

Por último, agradezco el apoyo del proyecto "Contenido Generado por Usuarios: Información relevante para nuevas aplicaciones de Sistemas de Recomendación", código DIUBB 183515 3/R, financiado por la Dirección de Investigación de la Universidad del Bío-Bío.

Índice General

Capítulo 1. Introducción.....	8
1.1. Descripción del proyecto	9
1.1.1. Herramienta previa desarrollada	9
1.1.2. Problema a resolver	10
1.2. Solución propuesta.....	11
1.2.1. Objetivo general	11
1.2.2. Objetivos específicos.....	11
1.3. Presentación de Capítulos	12
Capítulo 2. Marco Teórico	14
2.1. Introducción a los sistemas de recomendación.....	14
2.1.1. Definición de sistemas de recomendación	14
2.1.2. Técnicas de recomendación.....	15
2.2. Sistemas de recomendación sensibles al contexto	16
2.2.1. Definición de contexto	16
2.2.2. Obtención de la información contextual	17
2.3. Taxonomía.....	17
2.3.1. Definición de taxonomía	17
2.3.2. Taxonomía previamente construida	18
2.4. DBpedia.....	18
2.4.1. Qué es DBpedia.....	18
2.4.2. Extracción de datos	19
2.5. WordNet	19
2.5.1. Qué es WordNet	19
2.5.2. Extracción de datos	20
Capítulo 3. Descripción del sistema	21
3.1. Requerimientos funcionales	21
3.2. Requerimientos no funcionales	24
3.3. Alcances	25
Capítulo 4. Análisis	26
4.1. Diagrama software actual.....	26
4.2. Funcionalidad deseada: Casos de uso.....	30
4.2.1. Actores	30
4.2.2. Diagramas de casos de uso.....	31

Capítulo 5. Diseño e implementación	35
5.1. <i>Interfaz de usuario</i>	36
5.2. <i>Manejo de Archivos.....</i>	38
5.3. <i>Búsqueda de instancias</i>	41
5.4. <i>Búsqueda de sinónimos</i>	45
5.5. <i>Búsqueda de palabras derivadas</i>	48
5.6. <i>Gestión de instancias</i>	51
5.7. <i>Menú pantalla secundaria.....</i>	55
5.8. <i>Diagrama de la aplicación mejorada</i>	59
5.9. <i>Otros elementos de Diseño.....</i>	61
5.9.1. <i>Diseño interfaz y navegación</i>	61
5.9.2. <i>Diseño de arquitectura funcional.....</i>	63
Capítulo 6. Pruebas.....	64
6.1. <i>Elementos de prueba.....</i>	64
6.2. <i>Pruebas de funcionalidad.....</i>	64
6.2.1. <i>Búsqueda de instancias.....</i>	65
6.2.2. <i>Búsqueda de sinónimos</i>	68
6.2.3. <i>Búsqueda de palabras derivadas.....</i>	72
Conclusiones	76
Referencias.....	77
Anexo 1: Especificación de casos de uso.....	79

Índice de Tablas

Tabla 1: Requerimientos funcionales.....	24
Tabla 2: Requerimientos no funcionales.....	24
Tabla 3: Pruebas búsqueda de instancias.	68
Tabla 4: Pruebas búsqueda de sinónimos.....	71
Tabla 5: Pruebas búsqueda de palabras derivadas.	74

Índice de Figuras

Figura 1: Captura de la herramienta desarrollada previamente [1]	10
Figura 2: Matriz de rating de usuario.	14
Figura 3: Pantalla principal.	26
Figura 4: Gestión de taxonomía.	27
Figura 5: Encabezado y menú.	27
Figura 6: Diagrama de clases: Software actual [1].	28
Figura 7: Casos de uso: Gestión de archivos.	31
Figura 8: Casos de uso: Gestión de taxonomía.	32
Figura 9: Casos de uso: Búsqueda de instancias.	33
Figura 10: Casos de uso: Gestión de instancias.	34
Figura 11: Pantalla secundaria.	36
Figura 12: Diagrama de clases: SecondaryFrame.	37
Figura 13: Estructura archivo .xml.	38
Figura 14: Ejemplo taxonomía.	39
Figura 15: Ejemplo HashMap.	40
Figura 16: Diagrama de clases: Manejo de archivos.	41
Figura 17: Ejemplo de taxonomía.	42
Figura 18: Consulta SPARQL.	42
Figura 19: Instancias de Spouses desde DBPedia.	43
Figura 20: Instancias de Spouses desde App.	44
Figura 21: Diagrama de clases: DBPedia.	44
Figura 22: Método getNounSynsetsOf.	45
Figura 23: Sinónimos de “Anger” encontrados en WordNet.	47
Figura 24: Sinónimos extraídos para “Anger”.	48
Figura 25: Método getDerivationallyRelatedForms.	49
Figura 26: Diagrama de clases: WordNet.	50
Figura 27: Diagrama de clases: NodePopupMenuTaxonomy.	52
Figura 28: Panel instancias duplicadas.	53
Figura 29: Código búsqueda instancias duplicadas.	54
Figura 30: Código visualización instancias duplicadas.	54
Figura 31: Diagrama de clases: NodePopupMenuInstance.	55
Figura 32: Diagrama de clases: Software actual.	56
Figura 33: Diagrama de clases: MenuInstaceFrame.	56
Figura 34: Diagrama de clases: Software actual.	56
Figura 35: Código guardado automático de archivos.	57
Figura 36: Diagrama de clases: Pantalla principal.	59
Figura 37: Jerarquía de menú.	61
Figura 38: Esquema de navegación.	62
Figura 39: Árbol de descomposición funcional.	63

Figura 40: Instancias de “Automobiles” encontradas en DBPedia.	65
Figura 41: Sinónimos de “Automobiles” encontrados en WordNet.	69
Figura 42: Palabras derivadas de “Automobiles” encontradas en WordNet.	72

Capítulo 1. Introducción

Actualmente la compra y venta en el comercio electrónico es cada vez mayor y la gran cantidad de productos y servicios ofrecidos en la web puede ser un problema a la hora de realizar una compra entre tantas opciones a elegir, es por este motivo que los sistemas de recomendación son cada vez más usados para su solución.

Un sistema de recomendación utiliza información del usuario, como sus gustos y preferencias, sobre un producto o servicio en particular en el cual está interesado y le recomienda aquellos que se asimilen a sus gustos. Pero además de los gustos y preferencias, un sistema de recomendación sensible al contexto utiliza información sobre las circunstancias en las que un usuario interactúa con artículos en la web, como la hora del día, el estado de ánimo, el clima, el contexto social en el que se encuentra al momento de utilizar un producto o servicio y su ubicación. Mediante estudios se ha demostrado que la información contextual puede tener balances positivos al momento de recomendar productos, además de que los proveedores han informado una mejora en el rendimiento de las ventas cuando la información de contexto se tiene en cuenta.

Como una forma de obtener información de contexto, en [1] se propone el uso de una taxonomía genérica de contexto que está compuesta por clases/categorías de DBpedia, proyecto que genera información semántica a partir de Wikipedia [2]. Cabe señalar que “taxonomía” es la ciencia en la que se clasifican u organizan grupos (de cosas) con características comunes, y se establecen parámetros de diferencia, creando familias, ramas y conjuntos relacionados [3]. En el caso de este proyecto, se busca construir una taxonomía o clasificación de palabras que describen contexto. Por ejemplo, el contexto “Momentos del día” se puede subclasificar en los contextos “Mañana”, “Tarde” y “Noche”. Para la solución de este problema se hará uso de una aplicación previamente construida, la que se mejorará y agregarán nuevas funciones.

DBpedia forma parte de la iniciativa Linked Data. Los datos enlazados, como se le denomina en español, es un método para publicar datos en la web e interconectarlos entre diferentes fuentes de datos [4], esto permite a DBpedia enlazar sus datos con otras fuentes de datos, como WordNet, lo que ayuda a enriquecer la taxonomía con la que se está trabajando.

Haciendo uso de la taxonomía generada, es posible identificar y extraer información de contexto, existentes en reseñas de usuarios. Para lo anterior, se seleccionan palabras que, por sus características, pueden contener información contextual, llamadas palabras candidatas. Luego, se analiza si existe un mapeo entre cada palabra candidata y las categorías de la taxonomía, para devolver una lista de las categorías relacionadas a la palabra. Así, la palabra es enlazada con una o más categorías de la taxonomía, lo que se llamará anotación de contexto.

1.1. Descripción del proyecto

1.1.1. Herramienta previa desarrollada

Para la realización de este proyecto se mejorará y agregarán nuevas funciones a una aplicación desarrollada en un trabajo previo [1]. Esta aplicación tiene por propósito apoyar la creación de taxonomías a partir de información disponible en DBpedia. La herramienta actual permite, mediante el ingreso por teclado de una palabra clave, la búsqueda de la categoría correspondiente en DBpedia. La herramienta permite indicar cuántos niveles de profundidad se desean buscar en DBpedia para una categoría dada, esto es, cuantas relaciones entre categorías recorrer en la búsqueda. Además de restringir la búsqueda a categorías según diferentes criterios del usuario: palabras que empiezan, contienen o terminan con secuencias de letras que el usuario desee. Esto se puede observar en el panel izquierdo de la interfaz de la aplicación existente, mostrada en la Figura 1.

La información sobre categorías recopilada de esta forma, es mostrada luego en forma de árbol, considerando como raíz de dicho árbol la categoría correspondiente a la palabra buscada. Esto se puede observar en el panel del medio mostrado en la Figura 1.

El software permite la creación de una taxonomía nueva mediante la agregación y eliminación de nuevas categorías en la forma de nodos en el árbol del panel derecho de la Figura 1, además de poder agregar cada palabra del árbol generado del panel central al panel derecho. De esta forma, el árbol que se forma en el panel derecho es la taxonomía que se desea construir.

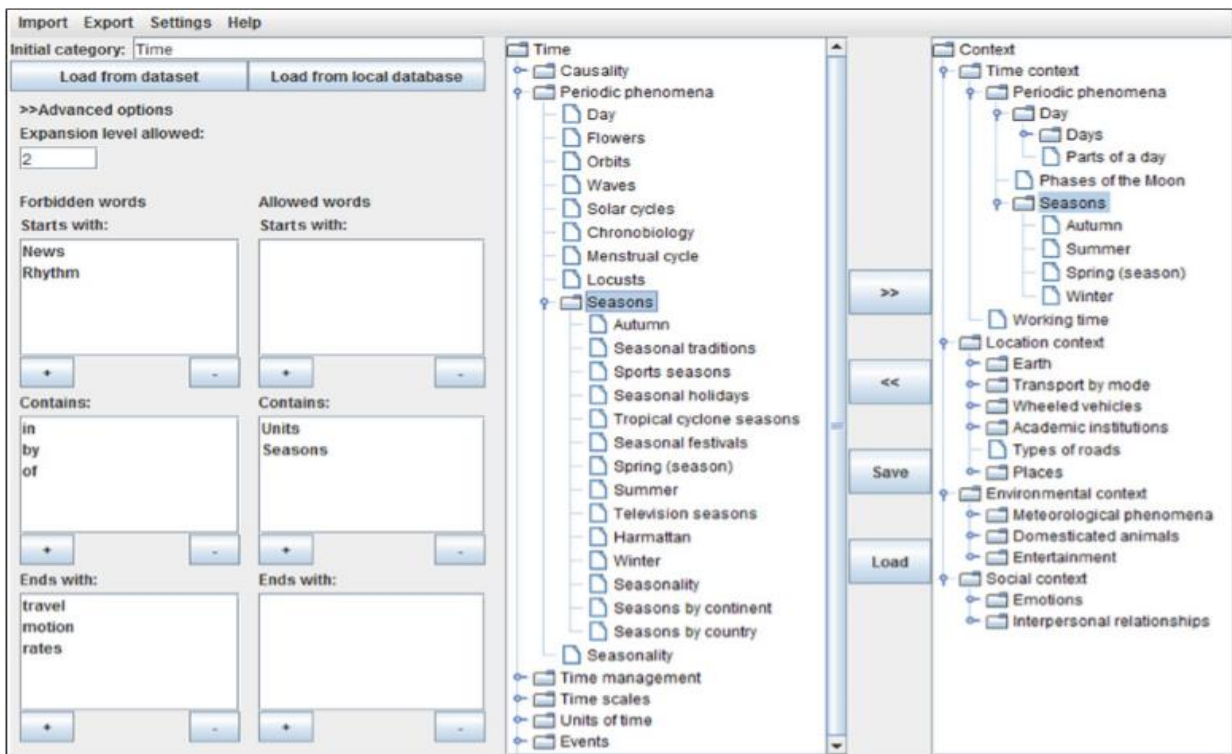


Figura 1: Captura de la herramienta desarrollada previamente [1]

La herramienta también permite la importación y exportación de una taxonomía, de manera local, en archivos con formato xml.

1.1.2. Problema a resolver

La gestión de una taxonomía y la búsqueda de categorías desde DBpedia mediante la herramienta actual no es suficiente para generar una taxonomía de calidad y enriquecida, pues muchos términos que describen información de contexto no son incluidos, lo cual podría dificultar la obtención de información para un sistema de recomendación sensible al contexto. Es por esto que se propone modificar y añadir nuevas funcionalidades en la aplicación, para permitir la búsqueda automatizada de más palabras relacionadas a los diferentes contextos, y gestionar la taxonomía de una manera más completa, por ejemplo identificando y eliminando posibles palabras repetidas en diferentes niveles de la taxonomía, con el fin de generar una taxonomía mejorada y enriquecida de nuevos datos, aprovechando de mejor forma la estructura y las potencialidades de interconexión de fuentes de datos que permite DBpedia.

1.2. Solución propuesta

Para este proyecto se desarrollará una nueva versión de la herramienta construida, mejorando el código existente y agregando nuevas funcionalidades, con el fin de que la herramienta sea capaz de mejorar la taxonomía que se obtiene desde DBpedia mediante la identificación e inserción de instancias que estén relacionadas con alguna categoría de contexto, logrando así un enriquecimiento de la taxonomía. Además, se implementará el acceso y enlace con otras bases de conocimiento como WordNet, desde donde se buscarán sinónimos y palabras derivadas de las categorías de la taxonomía y sus instancias.

De esta forma, al contar con una herramienta con mayores funcionalidades de búsqueda y gestión de la información, se espera facilitar la construcción de taxonomías enriquecidas, lo que además podría permitir mejorar los resultados obtenidos en el trabajo previo [1].

1.2.1. Objetivo general

El principal objetivo de este proyecto es desarrollar una versión mejorada de una herramienta para la edición de taxonomías utilizadas en la extracción de información para recomendación contextual, construidas a partir de DBpedia, base de conocimiento de acceso libre, y que actualmente permite la edición a nivel de categorías; esta nueva versión incorporará la edición de instancias de cada categoría, permitiendo la carga automática desde bases de conocimiento como DBpedia o WordNet, así como modificaciones manuales, enriqueciendo así las taxonomías mediante la incorporación de nuevas instancias y categorías de contexto.

1.2.2. Objetivos específicos

1. Investigar en la literatura sobre lo que son las taxonomías, los sistemas de recomendación sensibles al contexto, Linked Data, DBpedia y WordNet.
2. Analizar la herramienta a mejorar con respecto a su arquitectura, las bibliotecas y sus APIs utilizadas.

3. Implementar las nuevas funcionalidades a desarrollar, integrándolas a la herramienta y modificando el código ya existente.
4. Mejorar una taxonomía de conceptos relacionados con contexto, propuesta previamente, mediante la inclusión de instancias en las categorías de contexto, por medio de búsquedas de palabras contextuales en DBpedia y WordNet, mediante el uso de la herramienta mejorada.
5. Probar el adecuado funcionamiento de la herramienta.

1.3. Presentación de Capítulos

Este Proyecto de Título se presenta en seis capítulos, los cuales se describen a continuación:

En el presente capítulo se da una introducción al proyecto y se explica por qué es necesario el desarrollo de esta aplicación, cuál es el problema actual y cómo solucionarlo. Además se exponen los objetivos generales y específicos del proyecto.

En el capítulo dos se explican los conceptos involucrados en el desarrollo de este proyecto. En el marco teórico se explican conceptos como: Los sistemas de recomendación y algunas técnicas, sistemas de recomendación sensibles al contexto y qué se puede entender como contexto, qué es una taxonomía y definición de DBpedia y WordNet.

Luego, en el capítulo tres, se describe el sistema en desarrollo mediante los requerimientos funcionales y no funcionales, además de los alcances del sistema.

El capítulo cuatro representa el desarrollo de la solución, mediante el análisis de los casos de uso con el fin de conocer las funcionalidades del sistema.

En el capítulo cinco se mostrará el diseño y la implementación de la solución. Primero, se mostrará un diagrama de clases del software actual, luego se explicará detalladamente las nuevas funcionalidades agregadas a la aplicación, mediante la representación de diagramas de clases y

pantallas del sistema. Finalmente, el diseño se representa mediante una jerarquía de menú, esquema de navegación y un diseño de arquitectura funcional.

Las pruebas están implementadas en el capítulo seis, acá se realizaron pruebas de funcionalidad, específicamente para la búsqueda de instancias, sinónimos y palabras derivadas, con el fin de comprobar que el sistema arroje los resultados esperados extraídos desde DBpedia y WordNet.

Finalmente se realizarán las conclusiones finales de la investigación, las referencias y los anexos correspondientes.

Capítulo 2. Marco Teórico

2.1. Introducción a los sistemas de recomendación

2.1.1. Definición de sistemas de recomendación

Un sistema de recomendación se puede definir como el mecanismo utilizado para filtrar distintos tipos de temas o ítems de acuerdo a los requerimientos de un usuario y que puedan ser de utilidad e interés para éste.

Lo que busca un sistema de recomendación es predecir un ranking de preferencia que se le da a ciertos ítems por parte de los usuarios. Para que un algoritmo de recomendación pueda hacer esto, es necesario que los usuarios le den una valoración a algunos de estos ítems, ya sea un libro, una película, música, imágenes etc. La valoración puede ser numérica (de 1 a 5 por ejemplo) o binaria (“me gusta”, “no me gusta”), o puede corresponder a otro tipo de acción, por ejemplo, si ha comprado o consumido un producto.

El conjunto de valoraciones o ratings conocidos por un sistema de recomendación es comúnmente llamado “matriz de ratings”, en la cual cada celda no vacía representa la valoración que un usuario ha dado al ítem correspondiente. En general, se tiene un conjunto U de usuarios y un conjunto I de ítems de cualquier tipo.

		items				
		i_1	...	i_k	...	i_m
USERS	u_1	r_{u_1, i_1}		r_{u_1, i_k}		
	⋮					
	u_j	r_{u_j, i_1}				r_{u_j, i_m}
	⋮					
	u_n	r_{u_n, i_1}				r_{u_n, i_m}

Figura 2: Matriz de rating de usuario.

Usando estas valoraciones, el sistema puede identificar preferencias de los usuarios hacia ciertos tipos de ítems, y así determinar o predecir la valoración que daría a otros ítems que no ha valorado. De esta forma, por ejemplo, se pueden recomendar aquellos ítems que tengan la predicción de valoración más alta. Para que un sistema de recomendación pueda identificar preferencias de los usuarios, y por ende generar recomendaciones, se necesita que los usuarios del conjunto U realicen acciones sobre ítems del conjunto I .

2.1.2. Técnicas de recomendación

Para generar recomendaciones existen diversas técnicas, pero acá se abarcarán 3 técnicas principales ya que son las más usadas por los sistemas de recomendación, estas son: filtrado colaborativo, filtrado basado en contenido y enfoques híbridos.

2.1.2.1. Filtrado colaborativo

Este enfoque es el más utilizado para realizar recomendaciones, es usado por grandes sitios de comercio electrónico de diferentes dominios (libros, películas, música, etc.) y tiene muchas variaciones en sus algoritmos.

El filtrado colaborativo construye una base de datos de preferencias para los ítems por parte de los usuarios [5], generando recomendaciones según las preferencias de otros usuarios que tienen gustos parecidos al que se le dará una recomendación. Este filtrado se basa en que si un usuario A califica de manera similar los mismos artículos que un usuario B, A tiene mayor probabilidad de que tenga una preferencia parecida a B por un artículo desconocido que una persona elegida al azar. Por ejemplo, dado un usuario activo Eric, se hace una comparación de los gustos de Eric con las preferencias de los usuarios ingresados en la base de datos, el objetivo de esto es encontrar un conjunto de vecinos que históricamente tienen gustos parecidos a Eric. Los ítems que son del gusto de los vecinos encontrados, tienen una gran probabilidad de que a Eric también le gusten.

En otras palabras, el sistema de recomendación tiene guardado el perfil de un usuario al que se le quiere dar una recomendación, el cual contiene las diferentes valoraciones que el usuario ha dado a ciertos ítems. El sistema compara este perfil con los perfiles de otros usuarios para determinar el grado de similitud que se tienen entre ellos. Cuando el sistema devuelve los perfiles con mayor similitud, se recomiendan los artículos mejor calificados por estos perfiles similares y a los que el usuario no le haya dado un rating.

2.1.2.2. Filtrado basado en contenido

El filtrado basado en contenido toma en cuenta las preferencias históricas de un usuario u y las características de un artículo i . Lo que hace este enfoque es hacer recomendaciones para un usuario basado únicamente en su perfil, mediante el análisis del contenido de los elementos que el usuario ha calificado en el pasado [6].

Explicado de otra manera, lo que hace este enfoque es recomendar artículos basándose en los gustos históricos del usuario u , y que hayan sido calificados de manera positiva por este usuario. De esta manera, el sistema buscará elementos que sean similares a estos artículos, basándose en las características y descripciones que tienen los elementos calificados.

Este método emplea técnicas de Recuperación de Información, por ejemplo, un documento de texto es recomendado a un usuario basándose en una comparación entre su contenido y el perfil del usuario [7].

2.1.2.3. Enfoque híbrido

Este enfoque utiliza métodos de recomendación basado en una mezcla de los dos filtrados analizados anteriormente, con el fin de solucionar los problemas y maximizar los beneficios de estas dos técnicas.

2.2. Sistemas de recomendación sensibles al contexto

2.2.1. Definición de contexto

Diversas áreas dentro de la informática y de otras disciplinas han intentado dar una definición al concepto de “contexto”. Este concepto ha sido estudiado extensivamente y darle una definición específica es muy complicado y abstracto, por lo que no existe una definición unificada.

Los sistemas de recomendación sensibles al contexto utilizan información sobre las circunstancias en las que un usuario está interactuando con algún artículo, como la hora del día, su ubicación, el estado de ánimo, las condiciones climáticas y la compañía social con la que se encuentra, estos datos se pueden clasificar como palabras de contexto [1].

Los estudios han podido demostrar que las condiciones contextuales pueden tener un efecto positivo al momento de generar una recomendación a un usuario [8], por lo tanto, el rendimiento de los sistemas de recomendación es mejor cuando se tiene en cuenta la información contextual.

2.2.2. Obtención de la información contextual

La información contextual puede ser obtenida de manera implícita o explícita. La forma explícita permite obtener la información contextual de manera más directa, pero no siempre es posible obtenerla. Una forma de hacerlo es pidiendo directamente a un usuario que esté interactuando con los ítems, mediante formularios o preguntas concretas.

El usuario no siempre va a estar dispuesto a proporcionar esta información, ya sea por temas de seguridad, discreción, privacidad o simplemente por el tiempo y el esfuerzo que esto genera [1].

Para la obtención de la información contextual de manera implícita no se requiere de mayor tiempo ni esfuerzo por parte del usuario, pero en general no se puede obtener de manera directa, sino que se debe interpretar a partir del comportamiento normal del usuario. La información extraída puede ser significativa para los sistemas de recomendación. Esta información puede ser obtenida mediante sensores físicos utilizados, por ejemplo, en dispositivos móviles, como la ubicación del usuario mediante el GPS o la obtención de medidas de temperaturas (en Grados, Fahrenheit, etc.).

Otra técnica utilizada consiste en identificar y extraer información de contexto escrita por los usuarios en reseñas de artículos, como por ejemplo, en sitios de comercio electrónico.

2.3. Taxonomía

2.3.1. Definición de taxonomía

Taxonomía es la ciencia que estudia la clasificación de grupos de cosas. Una de sus aplicaciones más común es en la clasificación de organismos, estableciendo diferencias entre si y agrupándolos en familias y ramas. Los organismos se agrupan en ramas que a su vez están agrupados en ramas de rango más alto, por lo que se va generando una jerarquía entre los diferentes niveles del árbol creado por estas ramas.

Más en general, una taxonomía es un conjunto de palabras organizadas que sirve para controlar y agrupar términos utilizados en un campo temático de un determinado “vocabulario”, con el fin de facilitar el almacenamiento y la recuperación de elementos de un repositorio [9]. Un elemento permanecerá a cierta rama dependiendo de sus características y su comportamiento.

2.3.2. Taxonomía previamente construida

Para la realización de este proyecto se utilizó una taxonomía de conceptos que describen contexto, creada en [1], la cual se usó como apoyo para la construcción de la herramienta, además de posteriores pruebas.

Primero, se escogieron 4 dimensiones de contexto mediante una encuesta de trabajo previo al desarrollo de la herramienta. Las dimensiones de contexto consideradas, también llamadas categorías o variables de contexto, tienen una estructura jerárquica, por lo que pueden ser modeladas mediante una taxonomía. Estas 4 dimensiones principales son: Localización, Tiempo, Ambiente y Contexto Social. Éstas serán las categorías raíces de la taxonomía.

Dado que construir una taxonomía desde cero es muy costoso, se utilizaron repositorios disponibles en la web, en particular, repositorios basados en Web semántica publicados en Linked Data, y más específicamente, DBpedia, desde donde se obtuvieron categorías e instancias recorriendo cada subcategoría de sus repositorios.

Para enriquecer aún más la taxonomía se utilizó la base de datos léxica inglesa WordNet, para la obtención de sinónimos de cada categoría y cada instancia de las categorías.

2.4. DBpedia

2.4.1. Qué es DBpedia

Actualmente, Wikipedia es la 5ta página web más visitada del mundo [10]. Esta enciclopedia online está compuesta por contenido creado de manera colaborativa. Existen 299 ediciones en diferentes idiomas de Wikipedia, con 5.533.604 artículos en la edición en inglés y más de 40 millones de artículos en diferentes idiomas [11]. Wikipedia es la fuente de trabajo más grande y popular de internet.

DBpedia crea una base de conocimiento multilingüística a gran escala y mediante la extracción de datos estructurados en 111 diferentes idiomas de Wikipedia. Es utilizado ampliamente para pruebas en proyectos de investigación, algoritmos y herramientas de software.

DBpedia es parte de la iniciativa Linked Data (datos enlazados) en la web. Los datos enlazados se crean siguiendo un método para la publicación de datos estructurados que pueden ser interconectados entre sí, utilizando tecnologías web como HTTP, RDF y URI; estas tecnologías

interconectan la información publicada para que pueda ser leída por los computadores, permitiendo la conexión y consultas de datos de diferentes fuentes.

La estructura de la base de conocimiento de DBpedia es mantenida por los usuarios de la comunidad, estos crean mapeos de la información de Wikipedia para representar estructuras en la ontología de DBpedia.

2.4.2. Extracción de datos

La aplicación desarrollada anteriormente [1] permite la búsqueda de instancias para una categoría ingresada por el usuario. El software se conecta automáticamente con la base de datos de DBpedia y a través de consultas mediante un lenguaje estandarizado (SPARQL) se hace la búsqueda de instancias, con la necesidad de tener acceso a internet.

DBpedia permite la extracción de información desde su base de datos a través de la tecnología SPARQL, este es un lenguaje estandarizado para la consulta de grafos RDF; Resource Description Framework (RDF) es un modelo de datos para representar información sobre los recursos de la World Wide Web [12]. En resumen, SPARQL permite a los usuarios tener acceso a información disponible en internet a través de diferentes plataformas, como DBpedia.

2.5. WordNet

2.5.1. Qué es WordNet

WordNet es una base de datos léxica inglesa, compuesta por sustantivos, verbos, adjetivos y adverbios, donde cada palabra expresa un concepto diferente, independiente de si la palabra está ingresada en más de una categoría. WordNet agrupa las palabras según la forma de éstas (cadenas de letras) y por su sentido específico.

Esta base de datos contiene más de 118.000 palabras de diferentes formas, 90.000 sentidos de palabras diferentes y más de 166.000 pares de palabras .[13]

En WordNet, una palabra puede ser interpretada como un sustantivo en algunos contextos lingüísticos, como verbos en otros contextos, y como adjetivos y adverbios en otros contextos, en estos casos se respeta la diferencia de cada categoría y se ingresa esta palabra por separado, por lo que una palabra se puede repetir más de una vez.

2.5.2. Extracción de datos

WordNet permite la obtención de sinónimos y formas derivadas relacionadas. Es posible buscar sinónimos y palabras derivadas para una palabra cualquiera que se entregue como parámetro en una consulta a la base de datos de WordNet.

Las formas derivadas relacionadas son palabras de diferentes categorías sintácticas que tienen la misma forma raíz y están semánticamente relacionadas, por ejemplo, el verbo “saltar” tiene la misma raíz que el sustantivo “salto”, son diferentes categorías sintácticas (verbo y sustantivo) pero están relacionadas gramáticamente.

Capítulo 3. Descripción del sistema

3.1. Requerimientos funcionales

A continuación se describen los requerimientos funcionales de la aplicación en desarrollo:

RF	Requisito Funcional	Especificación
RF1	Construcción de un árbol (taxonomía) manualmente.	<p>RF1.1 Se podrá agregar un nodo haciendo click derecho en un nodo del panel derecho al que se le quiera agregar un hijo, eligiendo la opción correspondiente.</p> <p>RF1.2 Se podrá eliminar un nodo haciendo click derecho en la categoría que se quiera eliminar y eligiendo la opción correspondiente.</p> <p>RF1.3 El árbol creado o modificado se podrá guardar en un archivo .xml mediante el botón “Save”.</p>
RF2	Cargar archivo .xml	<p>RF2.1 Mediante el botón “Load” se podrá navegar en el dispositivo para seleccionar y cargar un archivo .xml con un árbol almacenado.</p> <p>RF2.1.1 En el panel derecho de la ventana principal se visualizará el árbol cargado y se podrá navegar en sus niveles.</p>
RF3	Cargar directorio	<p>RF3.1 Mediante el botón “Choose Directory” se podrá navegar en el dispositivo donde el usuario podrá elegir un directorio y poder acceder a la ventana secundaria.</p> <p>RF3.1.1 Una vez elegido el directorio, al momento de pasar al panel secundario se crearán automáticamente tres archivos .tsv para el almacenamiento de las instancias editadas posteriormente.</p>
RF4		RF4.1 Mediante el botón “DBpedia Instances Search”

	<p>Buscador de instancias, sinónimos y palabras derivadas.</p>	<p>el sistema buscará instancias en DBpedia para cada una de las categorías del árbol cargado y las guardará en un archivo generado automáticamente.</p> <p>RF4.2 Mediante el botón “WordNet Synonyms Search” el sistema buscará sinónimos desde una base de datos de WordNet para cada una de las categorías del árbol cargado y las guardará en un archivo generado automáticamente.</p> <p>RF4.3 Mediante el botón “WordNet Derivationally Search” el sistema buscará palabras derivadas desde una base de datos de WordNet para cada una de las categorías del árbol cargado y las guardará en un archivo generado automáticamente.</p>
<p>RF5</p>	<p>Visualización de instancias por categoría.</p>	<p>RF5.1 En el momento en el que el usuario selecciona un nodo del árbol generado en el panel izquierdo de la ventana secundaria, en el panel paralelo se podrán visualizar automáticamente las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente del nodo seleccionado.</p> <p>RF5.1.1 Al seleccionar el botón “Show Instances Found”, el usuario podrá visualizar las instancias encontradas en DBpedia y WordNet y que no han sido editadas.</p> <p>RF5.1.2 Al seleccionar el botón “Show Edited Instances,” el usuario podrá visualizar las instancias editadas, es decir, si las instancias originales encontradas en DBpedia y/o WordNet fueron editadas, se podrán visualizar al seleccionar este botón.</p>
<p>RF6</p>		<p>RF6.1 El usuario podrá agregar nuevas instancias haciendo click derecho en cualquier nodo del panel derecho y seleccionando la opción para agregar, esta</p>

	<p>El sistema debe permitir la eliminación y agregación de instancias.</p>	<p>nueva instancia se visualizará en la carpeta “Manual” del panel y se guardará en un archivo .tsv en su respectiva categoría.</p> <p>RF6.2 El usuario podrá eliminar una instancia del panel derecho haciendo click derecho en el nodo que quiera eliminar y seleccionando la opción correspondiente, la instancia se borrará del panel y se actualizará el archivo .tsv respectivo.</p>
<p>RF7</p>	<p>El sistema debe permitir el guardado automático de archivos al editar instancias.</p>	<p>RF7.1 En la pestaña “Settings” de la ventana secundaria, existe la opción para el guardado automático de archivos “Save edited files automatically”, donde el usuario podrá marcar o desmarcar la casilla.</p> <p>RF7.1.1 Si la opción está marcada, el sistema guardará automáticamente todas las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente en su archivo .tsv correspondiente al momento en el que el usuario agregue o elimine una instancia de una de las categorías.</p> <p>RF7.1.2 Si la opción está desmarcada, el sistema no guardará nada en el momento en el que el usuario agregue o elimine una instancia.</p> <p>RF7.1.2.1 El usuario tiene la opción de guardar todas las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente mediante la opción en la pestaña “Export” -> “Instance Tree” -> “Save All instances and synonyms to edited files”.</p>
<p>RF8</p>	<p>Visualización de instancias</p>	<p>RF8.1 Mediante el click derecho en una instancia del panel derecho, se podrá elegir la opción para visualizar si la instancia seleccionada está repetida en otra categoría.</p> <p>RF8.1.1 Se abrirá una ventana externa en donde se</p>

	repetidas.	<p>podrán visualizar las categorías en las cuales la instancia está contenida.</p> <p>RF8.1.1.1 Se podrá seleccionar una instancia de la lista y mediante el botón “See route” se podrá ver la ruta de la taxonomía en donde se encuentra la categoría con la instancia repetida.</p> <p>RF8.1.1.2 Se podrá seleccionar una instancia de la lista y mediante el botón “Remove” se eliminará la instancia de la categoría seleccionada.</p>
--	------------	--

Tabla 1: Requerimientos funcionales.

3.2. Requerimientos no funcionales

RNF	Requisitos no funcionales	Especificación
RNF1	Usabilidad	<p>RNF1.1 Se agregarán botones para agilizar la navegación en la aplicación y para la búsqueda de archivos en el computador desde la interfaz gráfica.</p> <p>RNF1.2 Se priorizará el uso del botón secundario del mouse (botón derecho) para distintas funcionalidades en los paneles de la aplicación.</p>
RNF2	Rendimiento	RNF2.1 El sistema debe tener tiempos de respuesta menores a 3 segundos en sus funcionalidades.
RNF3	Multiplataforma	RNF3.1 El sistema deberá funcionar en los sistemas operativos compatibles con la plataforma Java.
RNF4	Seguridad	RNF4.1 El sistema deberá proteger la información utilizada mediante una buena codificación de la aplicación y así no generar pérdidas de datos.
RNF5	Soporte	RNF6.1 Se facilitará la documentación necesaria en el código de la aplicación mediante javadoc para el entendimiento del código por parte del usuario.

Tabla 2: Requerimientos no funcionales.

3.3. Alcances

El presente proyecto tiene como alcance la construcción y modificación de una taxonomía a través de una aplicación en desarrollo, permitiendo la búsqueda de instancias, sinónimos y palabras derivadas desde DBpedia y WordNet. También permite la gestión de instancias, pudiendo agregar y eliminar instancias de cada una de las categorías de contexto encontradas, generando así una taxonomía completa y consistente.

Dicha taxonomía servirá como apoyo para la experimentación con sistemas de recomendación sensibles al contexto. Sin embargo, este Proyecto de Título se limita a mejorar la herramienta de generación de taxonomías, y la ejecución de experimentos de validación con un sistema de recomendación sensible al contexto ya existente.

Capítulo 4. Análisis

En este capítulo se presenta un breve análisis de la funcionalidad implementada en el software actual, y a partir de esto, se plantea la funcionalidad que deberá tener la herramienta mejorada en este proyecto.

4.1. Diagrama software actual

La Figura 3 representa la interfaz gráfica de la aplicación desarrollada en [1]. El **Panel 1** visualiza las categorías encontradas en DBpedia de una palabra ingresada por teclado, en este caso se ingresó la palabra “Time”. El **Panel 2** muestra una taxonomía cargada desde el computador o creada por el usuario desde el nivel cero, en este panel es posible agregar, eliminar y cambiar el nombre a las categorías, además de exportar el árbol a un archivo .xml.

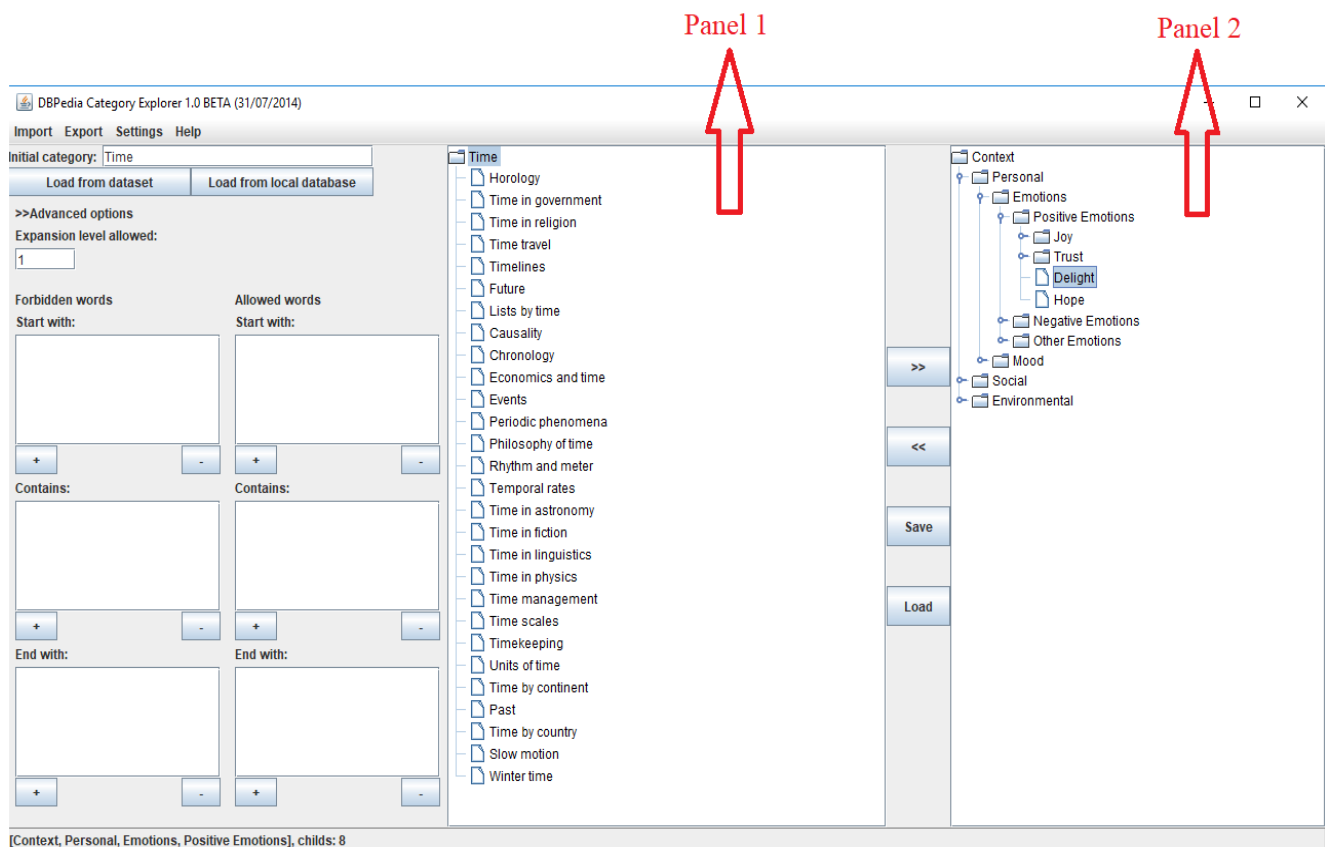


Figura 3: Pantalla principal.

En la Figura 4 se observan las opciones para la gestión de la taxonomía desde el panel derecho (Panel 2) de la pantalla principal del sistema.

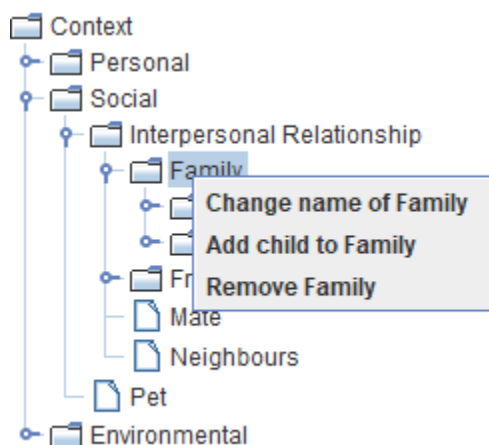


Figura 4: Gestión de taxonomía.

La Figura 5 representa el encabezado de la aplicación desarrollada en [1] y el menú de la pantalla principal del sistema. Al seleccionar alguna de las opciones del menú se desplegarán las opciones asociadas.

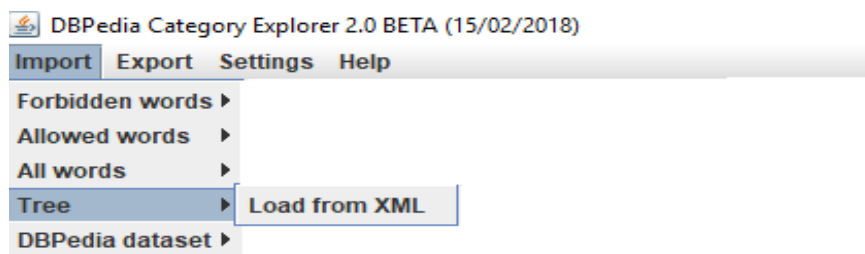


Figura 5: Encabezado y menú.

La Figura 6 describe la estructura del sistema actual. Este diagrama de clases permite visualizar cada una de las clases que conforman la aplicación, las relaciones entre ellas y los atributos y métodos utilizados para el funcionamiento del software.

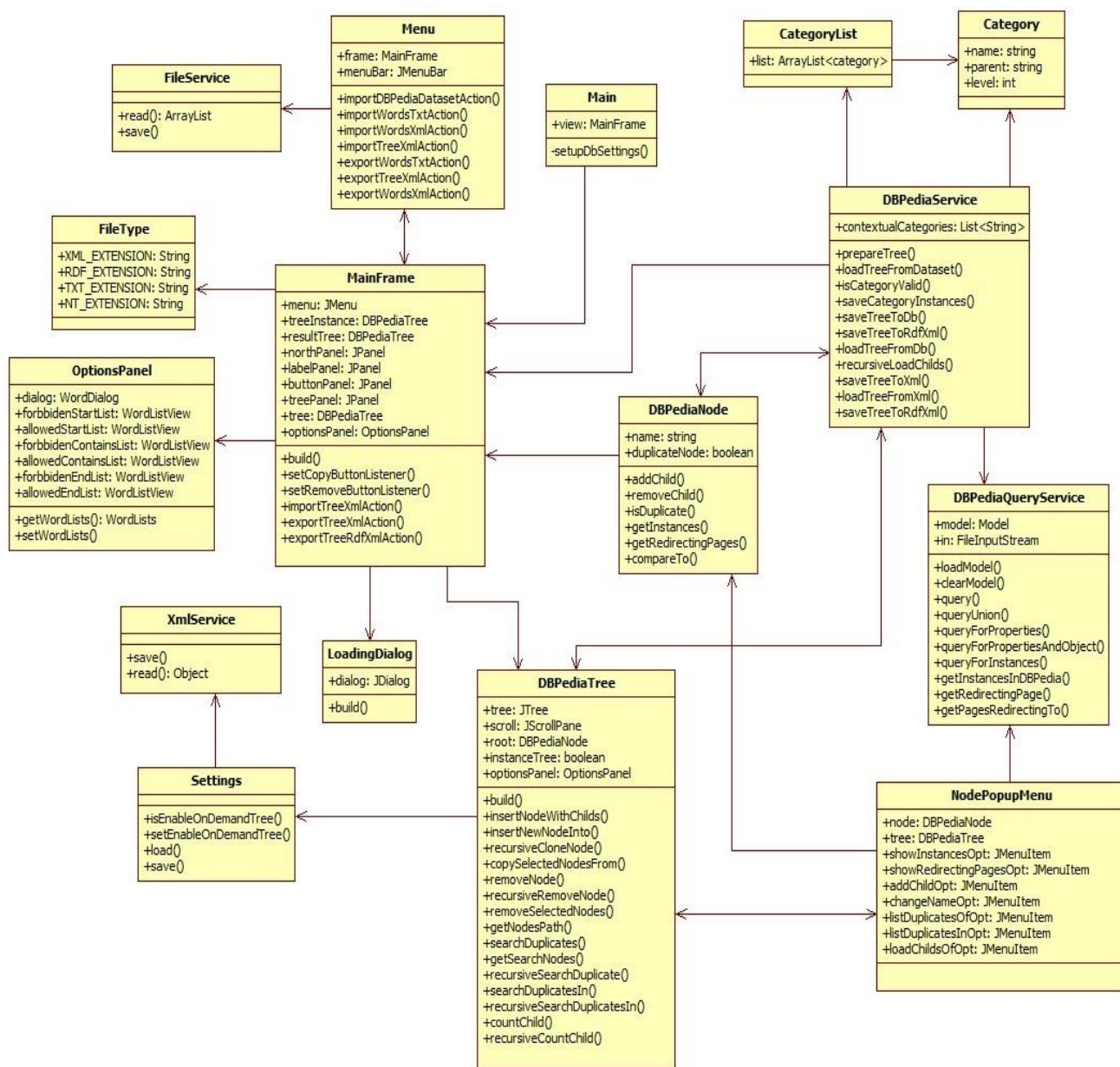


Figura 6: Diagrama de clases: Software actual [1].

A continuación se dará una breve descripción de las principales clases que conforman el diagrama de clases anterior:

Main: Es la clase principal del sistema, desde donde se crea un objeto MainFrame para la visualización de la interfaz y el funcionamiento de la aplicación.

MainFrame: Permite visualizar la interfaz gráfica de la aplicación, los paneles, botones y funciones como la importación y exportación de archivos .xml para la carga y almacenamiento de una taxonomía.

Menu: Esta clase contiene la pestaña visualizada en la parte superior de la aplicación, junto con las funcionalidades de cada menú de la pestaña. Aquí se permite la importación y exportación de palabras de contexto y del árbol generado.

DBpediaTree: La clase DBpediaTree permite crear una taxonomía o árbol de nodos y visualizarlo en el panel de la aplicación, además de aceptar el clickeo principal o secundario del mouse en un nodo del árbol para diferentes funcionalidades.

DBpediaNode: Representa un nodo con un nombre respectivo. Permite agregar hijos al árbol.

FileService: Permite leer y guardar datos a un archivo .tsv.

XmlService: Permite leer y guardar datos a un archivo .xml.

Category: Crea un objeto que representará un nodo de la taxonomía, con un nombre, padre y nivel correspondiente.

OptionsPanel: Esta clase crea el panel izquierdo de la pantalla principal, desde donde el usuario puede filtrar palabras para una búsqueda más específica de categorías de contexto.

DBpediaService: Permite importar y exportar un árbol desde un archivo .xml y generar el árbol mediante la búsqueda de instancias desde DBpedia haciendo un llamado a la clase DBpediaQueryService.

DBpediaQueryService: Permite la búsqueda de instancias para cada categoría de la taxonomía mediante consultas a la base de datos de DBpedia.

NodePopupMenu: Proporciona realizar diferentes funcionalidades al momento de hacer click derecho en un nodo del árbol. Estas funcionalidades pueden ser: agregar un nodo hijo,

ver las instancias del nodo seleccionado, cambiar el nombre de éste y ver una lista de las instancias duplicadas de este nodo.

4.2. Funcionalidad deseada: Casos de uso

En esta sección se modelarán los casos de uso de la aplicación en desarrollo para conocer los nuevos procesos desarrollados en las funcionalidades del sistema. Se utiliza esta notación, parte del lenguaje unificado de modelado UML (Unified Modeling Language), pues es el lenguaje estándar utilizado hoy en día para la descripción de funcionalidades software.

4.2.1. Actores

Para la utilización de la aplicación en desarrollo no es necesario registrarse ni autenticarse. El usuario podrá utilizar todas las funcionalidades presentes en el software sin tener privilegios otorgados por un administrador o desarrollador, por lo que el **usuario** será el único actor presente en los casos de uso.

4.2.2. Diagramas de casos de uso

Casos de uso: Gestión de archivos

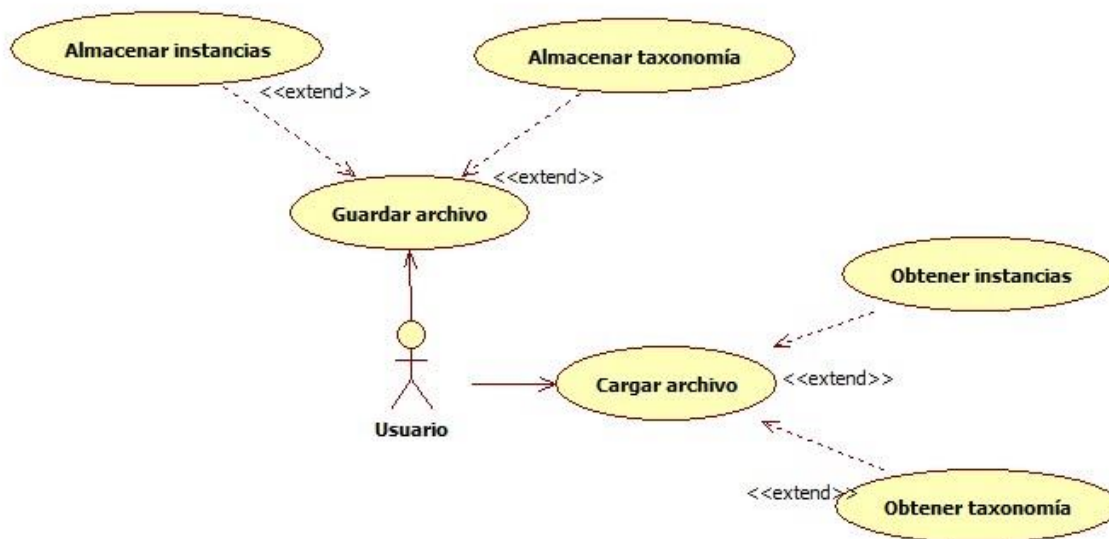


Figura 7: Casos de uso: Gestión de archivos.

En la Figura 7 se muestran los casos de uso correspondientes a la gestión de archivos. En el diagrama, se puede observar que el usuario tiene la posibilidad de cargar o guardar un archivo .xml para obtener o almacenar una taxonomía con sus categorías de contexto y sus nombres, nivel del árbol y nombre del nodo padre respectivo.

También podrá cargar o guardar un archivo .tsv en el dispositivo desde donde se está ejecutando la aplicación para obtener o almacenar las instancias, sinónimos, palabras derivadas y palabras agregadas manualmente de cada categoría de contexto.

Casos de uso: Gestión de taxonomía

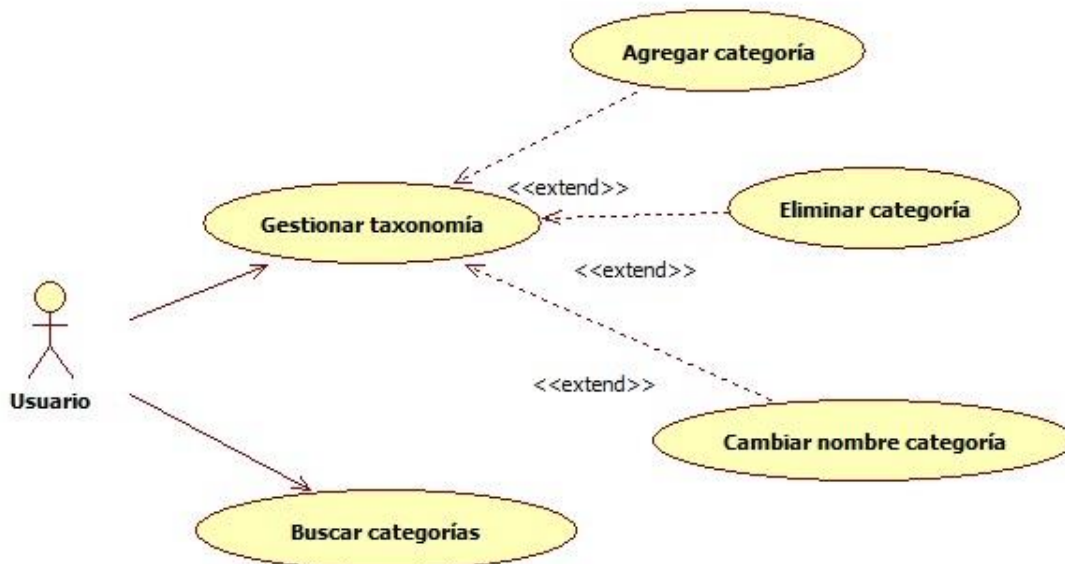


Figura 8: Casos de uso: Gestión de taxonomía.

La Figura 8 muestra los casos de uso correspondientes a la gestión de una taxonomía. Primero, mediante el ingreso de una palabra por teclado, es posible buscar instancias desde DBpedia de esta palabra y visualizarlas en un panel de la pantalla principal.

La aplicación también permite agregar, eliminar o cambiar el nombre a una categoría de la taxonomía visualizada en el panel derecho de la pantalla principal.

Casos de uso: Búsqueda de instancias

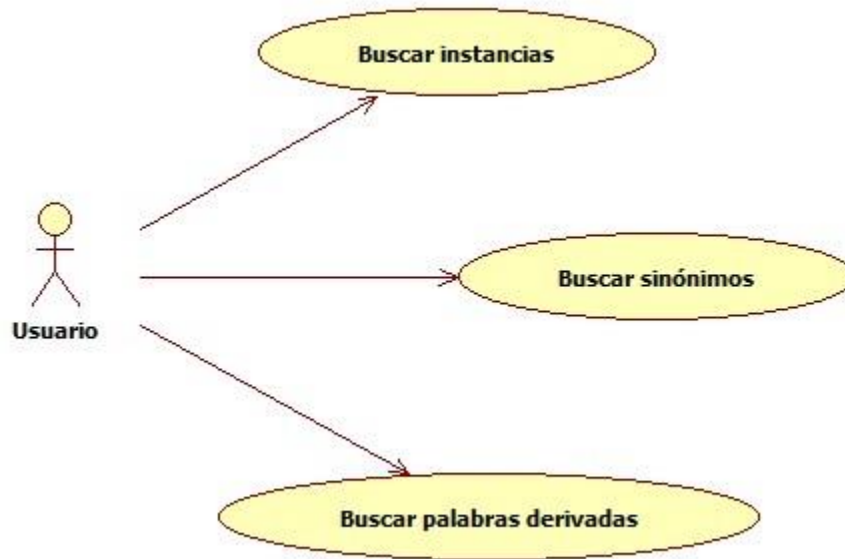


Figura 9: Casos de uso: Búsqueda de instancias.

En la Figura 9 se pueden observar los casos de uso relacionados con la búsqueda de datos desde dos bases de datos: DBpedia y WordNet. El usuario tiene la posibilidad de buscar instancias para cada categoría de contexto desde DBpedia o sinónimos y palabras derivadas para cada categoría desde WordNet. Estas palabras encontradas serán visualizadas en un panel de la aplicación.

Casos de uso: Gestión de instancias

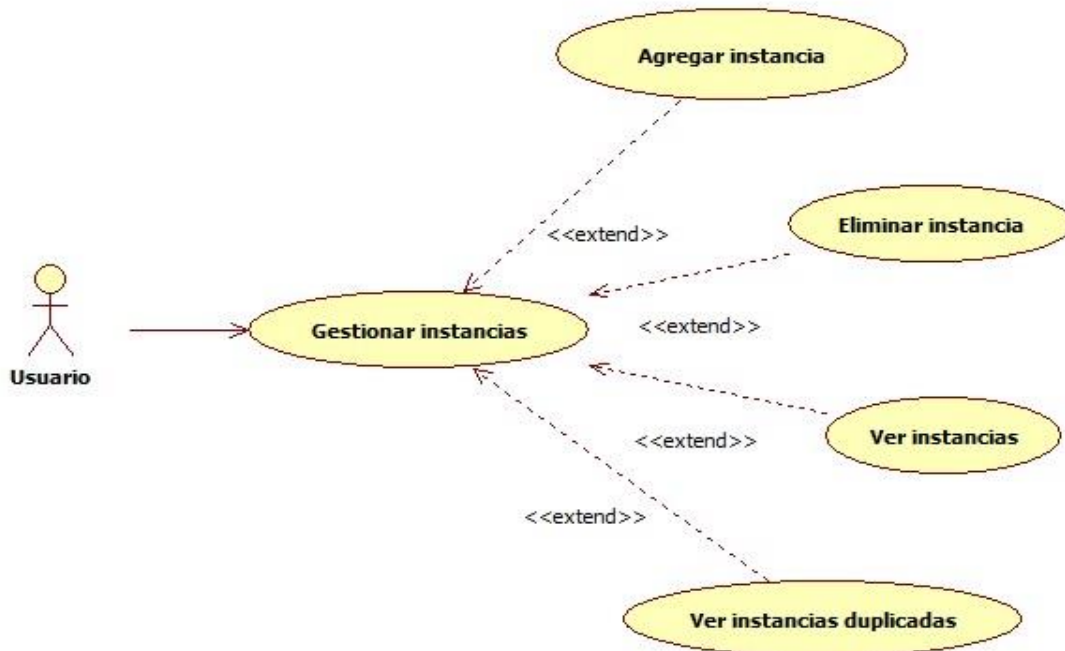


Figura 10: Casos de uso: Gestión de instancias.

La Figura 10 representa los casos de uso para la gestión de instancias. Desde el panel derecho de la pantalla secundaria es posible agregar, eliminar y ver las instancias que están duplicadas en otras categorías de contexto para una instancia seleccionada por el usuario.

Capítulo 5. *Diseño e implementación*

Para la mejora e implementación de las nuevas funcionalidades al software actual, se modificaron varias clases Java del código anterior, agregando nuevos métodos. Además, se agregó una nueva ventana de interfaz de usuario para incorporar estas nuevas funcionalidades facilitando la usabilidad de la aplicación.

En las siguientes secciones se detallará cada funcionalidad agregada al software nuevo. Primero se dará una breve descripción de cada uno de estos temas:

- **Interfaz de usuario:** Se detallará la nueva interfaz de usuario agregada al software actual, esto abarca sólo la parte visual desarrollada, en donde el usuario podrá visualizar una ventana con diferentes botones y paneles.
- **Manejo de archivos:** El usuario podrá ser capaz de importar y exportar archivos independientes para la carga y almacenamiento de la taxonomía, instancias, sinónimos, palabras derivadas e instancias agregadas manualmente. Aquí se explicará cómo es la estructura de los archivos .xml y .tsv (*tab separated values*) generados y cómo el sistema será capaz de realizar la gestión de archivos.
- **Búsqueda de instancias:** El usuario podrá buscar instancias desde DBpedia para cada categoría de contexto de la taxonomía. Se detallará cómo se realiza la búsqueda de instancias.
- **Búsqueda de sinónimos:** El usuario podrá buscar sinónimos desde WordNet para cada categoría de contexto e instancias de la taxonomía. Se explicará cómo se realiza la búsqueda de sinónimos.
- **Búsqueda de palabras derivadas:** El usuario podrá buscar palabras derivadas desde WordNet para cada categoría de contexto, instancias y sinónimos de la taxonomía. Se explicará cómo se realiza la búsqueda de palabras derivadas.
- **Gestión de instancias:** El usuario podrá gestionar las instancias desde el panel derecho de la pantalla secundaria de la aplicación, donde podrá agregar y eliminar instancias a una categoría de contexto seleccionada. En este ítem se explicarán las nuevas funcionalidades para la gestión de instancias.
- **Menú pantalla secundaria:** En este ítem se explicarán las funcionalidades agregadas en las pestañas del menú superior de la ventana desarrollada.

5.1. Interfaz de usuario

La Figura 11 muestra la nueva ventana creada en el software en desarrollo. En esta pantalla se implementaron diversos botones para la búsqueda y visualización de instancias, además de dos paneles: El **Panel 1** desde donde se visualiza la taxonomía cargada o creada desde la pantalla principal del sistema, y el **Panel 2**, el cual permite la visualización de instancias, sinónimos, palabras derivadas, palabras agregadas manualmente y la cantidad instancias de una categoría seleccionada desde la taxonomía del primer panel.

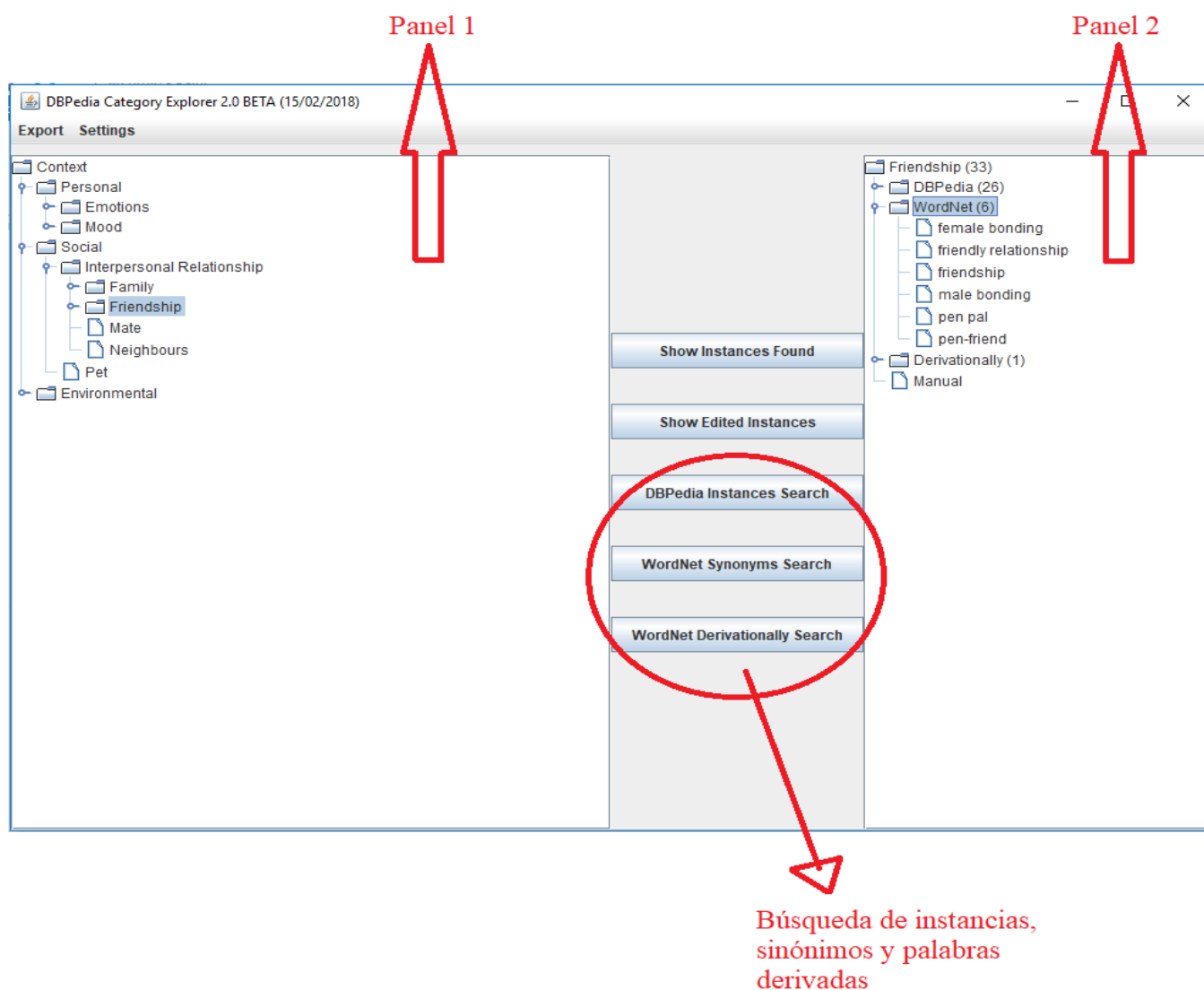


Figura 11: Pantalla secundaria.

Para la creación de esta nueva ventana se desarrolló una clase llamada **SecondaryFrame**, modelada en el diagrama de clases de la Figura 12. Esta clase es desplegada al momento de hacer click en el botón “Instance Frame” de la pantalla principal del sistema.

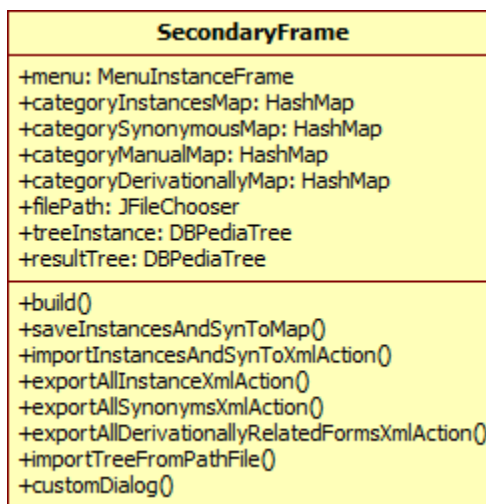


Figura 12: Diagrama de clases: SecondaryFrame

Para la creación de esta interfaz gráfica se utilizó la biblioteca gráfica Swing de Java, la cual incluye distintos widgets utilizados para el desarrollo de GUIs (interfaz gráfica de usuario). Algunos componentes utilizados son:

JFrame: Utilizado para la creación de ventanas (ventana principal y ventana secundaria) donde se añaden distintos objetos (botones, etiquetas, textos, etc.).

JButton: Utilizado para la creación de botones.

JLabel: Utilizado para la visualización de texto.

JPanel: Utilizado para la creación de los paneles donde se visualiza la taxonomía y las instancias.

JDialog: Utilizado para la creación de ventanas secundarias con distintos objetos.

JFileChooser: Utilizado para la búsqueda de ficheros en el computador desde una ventana secundaria.

Las funcionalidades presentes en los botones y paneles de la interfaz de usuario serán explicadas en las siguientes páginas. A continuación se explicará cómo la aplicación es capaz de realizar la importación y exportación de datos hacia archivos almacenados en el computador.

5.2. Manejo de Archivos

El manejo de archivos toma en cuenta la importación y exportación de archivos .xml y .tsv. Los archivos .tsv almacenarán cada categoría de la taxonomía y sus respectivas instancias. Un archivo .xml almacenará una taxonomía con los siguientes datos de una categoría: El nivel del árbol de la categoría, su nombre y el nombre de su nodo padre. El siguiente ejemplo muestra la estructura de un archivo .xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<category-list>
  <category>
    <level>2</level>
    <name>Family</name>
    <parent>Interpersonal_Relationship</parent>
  </category>
  <category>
    <level>2</level>
    <name>Neighbours</name>
    <parent>Interpersonal_Relationship</parent>
  </category>
  <category>
    <level>1</level>
    <name>Interpersonal_Relationship</name>
    <parent>Social</parent>
  </category>
  <category>
    <level>1</level>
    <name>Pet</name>
    <parent>Social</parent>
  </category>
  <category>
    <level>0</level>
    <name>Social</name>
    <parent>Context</parent>
  </category>
</category-list>
```

Figura 13: Estructura archivo .xml.

Donde Social será el nodo raíz del árbol ya que tiene nivel cero, Interpersonal Relationship y Pet, con nivel uno, serán hijos de Social, y Neighbours y Family serán hijos de Interpersonal Relationship. La Figura 14 muestra la taxonomía construida del archivo .xml.



Figura 14: Ejemplo taxonomía.

Para el manejo de archivos se modificaron dos clases creadas en el software actual, a éstas se le agregaron los métodos para la importación y exportación de archivos ubicados en la clase DBpediaService del software anterior. Dichas clases son explicadas a continuación:

XmlService: Permite importar las categorías contextuales de la taxonomía desde un archivo .xml para guardarlas en una lista de string. También es posible cargar y guardar un árbol (taxonomía) desde un archivo .xml.

Para el manejo de archivos .xml se utilizó la biblioteca **JAXBContext** de Java, la cual provee de dos clases:

Marshalling: La clase Marshaller proporciona a la aplicación la capacidad de convertir un árbol de contenido, en este caso la taxonomía, en datos .XML.

Unmarshalling: La clase Unmarshaller proporciona a la aplicación la capacidad de convertir datos contenidos en un archivo .XML en un árbol de objetos de contenido Java.

FileService: Esta clase permite la carga de las categorías y sus instancias desde un archivo .tsv para almacenarlas en un HashMap. También, mediante otro método, es posible la exportación a un archivo .tsv de los datos guardados en un HashMap, como por ejemplo, las categorías con sus instancias o sinónimos.

¿Qué es un HashMap?

Un **HashMap** es utilizado para almacenar colecciones de objetos. Corresponde a una estructura de tipo Diccionario o Mapa, en que almacena pares $\{clave, valor\}$, y en que *clave* y *valor* pueden ser a su vez objetos complejos. Internamente, un HashMap es implementado mediante una tabla Hash. En este caso se utilizarán cuatro objetos HashMap para almacenar los siguientes datos: Las categorías con sus instancias, las categorías con sus sinónimos, las categorías con sus palabras derivadas y las categorías con las palabras ingresadas manualmente. La siguiente línea de código declara un objeto HashMap, en que las claves son de tipo String, y los valores corresponden a listas de String:

```
public static Map<String, List<String>> categoryInstancesMap = new HashMap();
```

El HashMap “**categoryInstanceMap**” almacena las instancias de cada categoría de la taxonomía. Como claves se almacenan cada una de las categorías y como valores se almacenan listas de palabras correspondientes a instancias de DBpedia, para su categoría correspondiente. En la Figura 15 se muestra un ejemplo de un HashMap con cada categoría asociada a una lista de instancias:

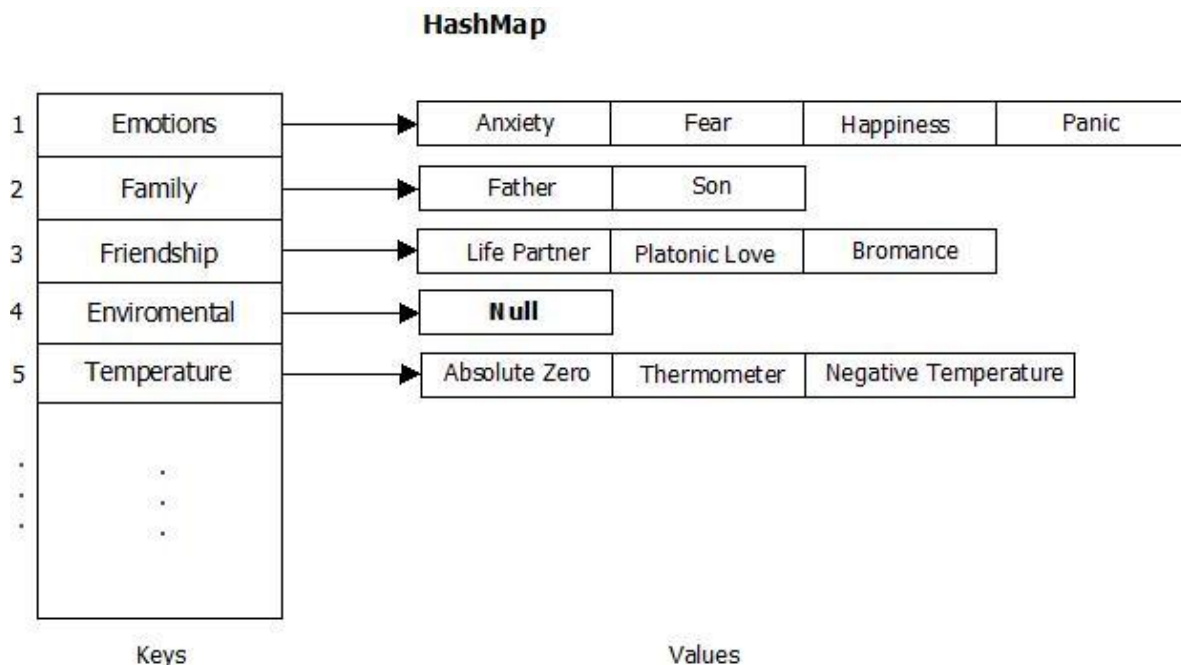


Figura 15: Ejemplo HashMap.

Cada clave (mostrada en el sector **Keys** en el lado izquierdo de la Figura 15) tendrá almacenada una categoría de la taxonomía. En esta estructura estarán almacenadas todas las categorías de la taxonomía. Cada valor correspondiente a la clave categoría tendrá asociada una lista de strings, (mostrada en el sector **Values** en el lado derecho de la Figura 15) la cual almacenará todas las instancias de su categoría correspondiente. Una categoría puede tener una lista vacía, como es el caso de “Enviromental”, el cual tiene un valor **Null** en su lista de valores.

La Figura 16 visualiza el modelado para el manejo de archivos:

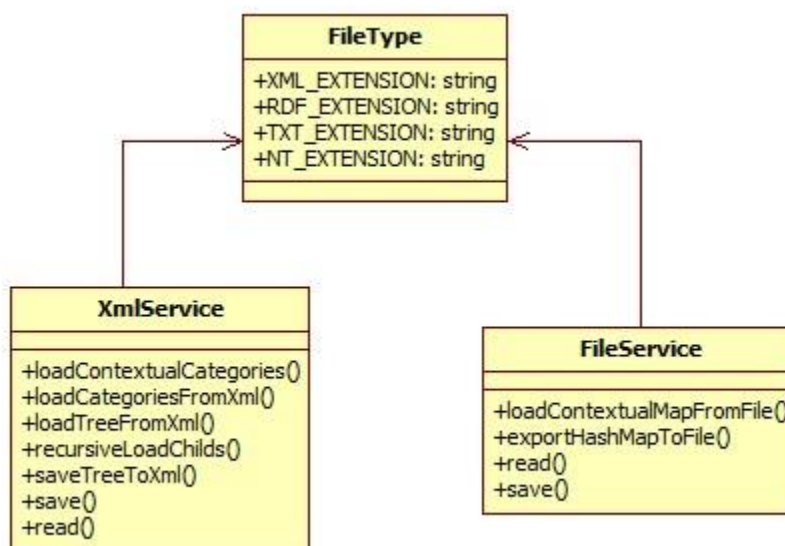


Figura 16: Diagrama de clases: Manejo de archivos.

5.3. Búsqueda de instancias

Para la búsqueda de instancias desde DBpedia se utiliza una consulta implementada en el método **getInstancesInDBpedia()** almacenada en la clase **DBpediaQueryService** de la aplicación, este método es llamado desde la clase **DBpediaService**, el cual, mediante un ciclo

que recorre todas las categorías de la taxonomía, le entrega como parámetro una categoría por cada iteración, con el fin de buscar instancias para cada categoría.

Cada nodo del árbol utilizado representa una categoría, las cuales están jerarquizadas por niveles, desde el nodo raíz hasta el nodo más profundo de la taxonomía. La Figura 17 representa una taxonomía, donde para la búsqueda de instancias se consultará por cada nodo del árbol.

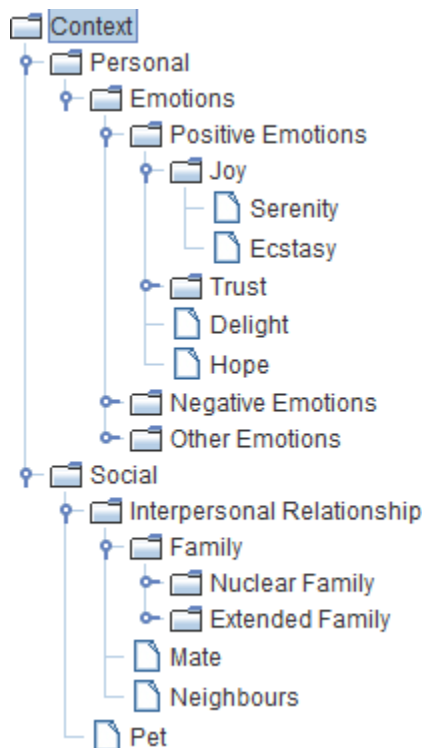


Figura 17: Ejemplo de taxonomía.

Para la búsqueda de instancias se emplea una consulta desarrollada en el lenguaje SPARQL, mostrada a continuación:

```

486 | String query =
487 |     "SELECT ?instance WHERE { \n"
488 |     + "{?instance dcterms:subject " + "<" + categoryURI + "> .} UNION"
489 |     + "{?instance rdf:type " + "<" + categoryURI + "> .} UNION \n"
490 |     + "{?instance dbpprop:wordnet_type " + "<" + categoryURI + "> .} \n"
491 |     + "} \n";
    
```

Figura 18: Consulta SPARQL.

El parámetro **categoryURI** de la consulta, implementada en la línea 488, contiene la siguiente ruta entregada anteriormente en el código de la aplicación:

`http://dbpedia.org/resource/Category:categoryURI`

Donde **categoryURI** irá tomando el nombre de cada categoría de la taxonomía para buscar sus instancias respectivas. Luego, la variable **query**, de la línea 486, irá almacenando cada instancia encontrada en DBpedia y la irá guardando en una lista, la cual será retornada cuando no queden instancias por consultar.

Tomando la categoría “Spouses” como ejemplo, la línea 488 del código implica que se buscarán las instancias de la categoría **dcterms:subject** de la ruta “`http://dbpedia.org/resource/Category:Spouses`”.

La Figura 19 muestra las instancias almacenadas en el repositorio oficial de DBpedia para la categoría Spouses de la ruta de la página web anterior.



Figura 19: Instancias de Spouses desde DBPedia.

La Figura 20 muestra el panel de instancias de la aplicación en desarrollo con las instancias encontradas desde DBpedia para la categoría Spouses.

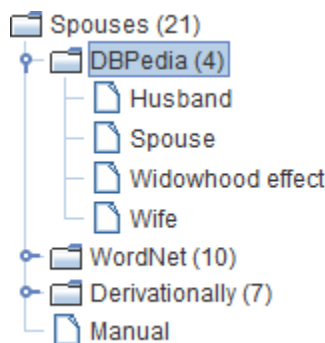


Figura 20: Instancias de Spouses desde App.

Como se puede ver, las instancias encontradas para Spouses fueron las mismas cuatro instancias visualizadas desde la página web de DBpedia, por lo que los resultados encontrados fueron los esperados.

Con respecto a la herramienta original, se refactorizó la clase **DBpediaService**, sacando los métodos utilizados para la exportación e importación de archivos, las cuales fueron trasladados a las clases específicas para gestión de archivos (XMLService y FileService).

El diagrama de clases de la Figura 21 representa las clases con los métodos para la búsqueda de instancias desde DBpedia.

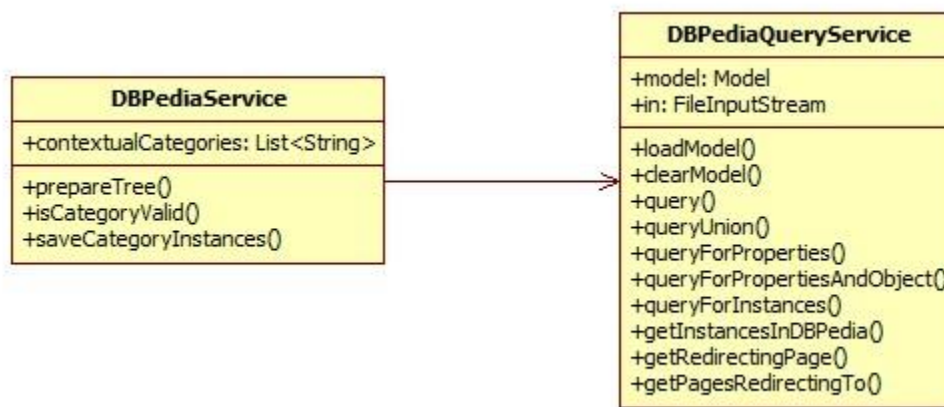


Figura 21: Diagrama de clases: DBpedia.

5.4. Búsqueda de sinónimos

Para la búsqueda de sinónimos de cada categoría e instancia de la taxonomía se utiliza el método `searchInstancesSynonyms()` de la clase `WordNetService`, este método recorre todas las categorías de la taxonomía y las instancias de cada categoría, luego, llama a los métodos `getSynsetsOf` de la clase `WordNet` para la extracción de sinónimos desde la base de datos de WordNet, para luego almacenarlas. Para realizar esta búsqueda se utiliza el siguiente código:

```

Start searchWordSynonyms(List<String> keywords)

    WordNet wordnet; //Se crea un objeto de la clase WordNet

    for each keyword in keywords do      //Para cada palabra en la lista
        synsetsNounAux .addAll(wordnet.getNounSynsetsOf(keyword));
        synsetsNounAux.addAll(wordnet.getAdjectiveSynsetsOf(keyword));
        synsetsNounAux.addAll(wordnet.getVerbSynsetsOf(keyword));
        synsetsNounAux.addAll(wordnet.getAdverbSynsetsOf(keyword));

    End_for

End

```

El ciclo *for* recorre cada palabra de un listado de palabras (keywords), correspondientes a una categoría y sus instancias en la taxonomía. Luego, mediante un llamado a los métodos `getSynsetsOf` de la clase `WordNet`, se buscan los sinónimos para cada palabra, buscando sinónimos correspondientes a sustantivos, adjetivos, verbos y adverbios. Posteriormente, se almacenan en una lista de sinónimos llamada `synsetsNoun`, la cual será retornada.

A continuación se visualiza el código de un método `getSynsetsOf` para la extracción de sinónimos desde la base de datos de WordNet:

```

public List<Synset> getNounSynsetsOf(String term) {
    Synset[] synsets = database.getSynsets(term, SynsetType.NOUN);
    return Arrays.asList(synsets);
}

```

Figura 22: Método `getNounSynsetsOf`.

Cabe destacar que, en el código de la Figura 22, el objeto **database** es un objeto de tipo WordNet, el cual contiene una copia de la base de datos de WordNet, y provee métodos que permiten acceder a los conjuntos de palabras con significados similares (Synsets) identificados en ella.

El método **getNounSynsetsOf()** tiene como parámetro la variable llamada **term**, la cual será una categoría o una instancia entregada desde la clase **WordNetService**.

Luego, se declara un arreglo de tipo **Synset** que almacenará, en este caso, los sinónimos de los sustantivos extraídos desde la base de datos de WordNet. El método **getSynsets()** para la extracción de sinónimos viene incluida en la biblioteca Java de WordNet. Para extraer los adjetivos, verbos y adverbios, la sintaxis de los métodos es similar pero se cambia el tipo NOUN por ADJECTIVE, VERB o ADVERB. Finalmente, la lista de sinónimos es retornada.

¿Para qué implementar estos métodos?

Los métodos **getSynsets()** para la extracción de sustantivos, adjetivos, verbos, adverbios fueron implementados para extraer la mayor cantidad de información contextual, en este caso de sinónimos, desde la base de datos de WordNet, con el fin de enriquecer la taxonomía de una manera más completa y con información que pueda ser útil para los sistemas de recomendación sensibles al contexto.

A continuación se muestra un ejemplo de la extracción de sinónimos desde la base de datos de WordNet. La Figura 23 muestra el repositorio de los sinónimos encontrados para la palabra “Anger” ingresada por teclado desde la página oficial de WordNet. La base de datos arrojó seis sinónimos identificadas con un círculo rojo, cinco de estas palabras son sustantivos y una palabra es un verbo. El objetivo de los métodos de WordNet de la aplicación en desarrollo es extraer estas seis palabras para almacenarlas y visualizarlas al usuario.

The screenshot shows the WordNet Search interface. At the top, there is a header "WordNet Search - 3.1" with links to "WordNet home page", "Glossary", and "Help". Below this is a search form with the text "Word to search for:" followed by an input field containing "Anger" and a "Search WordNet" button. Underneath, there are "Display Options:" with a dropdown menu set to "(Select option to change)" and a "Change" button. A key explains that "S:" shows synsets and "W:" shows word relations. Below the key, it says "Display options for sense: {offset} (gloss) 'an example sentence'".

The results are categorized into "Noun" and "Verb". Under "Noun", there are three entries, each with a synset ID, a link to the synset, and a list of synonyms circled in red:

- {07531593} [S: \(n\) anger](#), [choler](#), [ire](#) (a strong emotion; a feeling that is oriented toward some real or supposed grievance)
- {14060021} [S: \(n\) anger](#), [angriness](#) (the state of being angry)
- {00760325} [S: \(n\) wrath](#), [anger](#), [ire](#), [ira](#) (belligerence aroused by a real or supposed wrong (personified as one of the deadly sins))

Under "Verb", there are two entries, each with a synset ID, a link to the synset, and a list of synonyms circled in red:

- {01789790} [S: \(v\) anger](#) (make angry) "The news angered him"
- {01790925} [S: \(v\) anger](#), [see red](#) (become angry) "He angers easily"

Figura 23: Sinónimos de “Anger” encontrados en WordNet.

La Figura 24 muestra el panel de la aplicación en donde se pueden visualizar las mismas seis palabras, además de la palabra “anger”, encontradas en WordNet, por lo que el resultado arrojado por la aplicación fue el esperado.



Figura 24: Sinónimos extraídos para “Anger”.

5.5. Búsqueda de palabras derivadas

Las palabras derivadas o formas derivadas relacionadas son palabras de diferentes categorías sintácticas pero que están relacionadas gramáticamente, por ejemplo el sustantivo “desayuno” y el verbo “desayunar”. Para realizar la búsqueda de estas palabras se realiza una acción similar a la búsqueda de sinónimos. En este caso se buscarán palabras derivadas para todas las categorías, instancias y sinónimos de la taxonomía, mediante el siguiente método de la clase **WordNetService** :

Start searchDerivationally (List<String> keywords)

```

WordNet wordnet ;           //Se crea un objeto de la clase WordNet
Set<String> dervListAux
for each keyword in keywords do           //Para cada palabra en la lista
    synsetsNoun.addAll(wordnet.getDerivationallyRelatedForms(keyword,
    SynsetType.NOUN));
    synsetsNoun.addAll(wordnet.getDerivationallyRelatedForms(keyword,
    SynsetType.VERB));
    synsetsNoun.addAll(wordnet.getDerivationallyRelatedForms(keyword,
    SynsetType.ADJECTIVE));
  
```



```

synsetsNoun.addAll(wordnet.getDerivationallyRelatedForms(keyword,
SynsetType.ADVERB));

Set<Synset> synsetsNoun.addAll(synsetsNounAux);

End_for

End

```

El ciclo *for* recorre cada palabra en de un listado de palabras (correspondiente a una categoría, instancias y sinónimos en la taxonomía). Luego, mediante un llamado al método **getDerivationallyRelatedForms** de la clase **WordNet**, se buscan las palabras derivadas de cada instancia dependiendo si es un sustantivo, adjetivo, verbo o adverbio. Posteriormente, se almacenan en una lista de sinónimos llamada **synsetsNoun** la cual será retornada.

La siguiente imagen muestra el método **getDerivationallyRelatedForms** de la clase **WordNet**, la cual busca las palabras derivadas desde la base de datos de WordNet:

```

63 public Set<String> getDerivationallyRelatedForms(String string, SynsetType type) {
64     System.setProperty("wordnet.database.dir", wordnetPath);
65     Set<String> derivationallyForm = new HashSet<String>();
66     Synset[] synsets = database.getSynsets(string, type);
67     for (Synset synset : synsets) {
68         WordSense[] senses = synset.getDerivationallyRelatedForms(string);
69         for (WordSense sense : senses) {
70             derivationallyForm.add(sense.getWordForm());
71         }
72     }
73     return derivationallyForm;
74 }

```

Figura 25: Método getDerivationallyRelatedForms.

La variable **wordnetPath**, de la línea 64 del código, contiene la ruta de almacenamiento de la base de datos de WordNet. La lista de sinónimos **synsets** de la línea 66 obtiene los sinónimos para una categoría o instancia dada, el método **getSynsets** viene incluido en una biblioteca de WordNet. Luego, mediante un ciclo *for*, se buscan las palabras derivadas para cada uno de los

sinónimos encontrados, estas palabras son almacenadas en una lista llamada **derivationallyForm**, la cual es retornada.

La Figura 26 representa el modelado de las funcionalidades para la búsqueda de sinónimos y palabras derivadas desde la base de datos de WordNet.

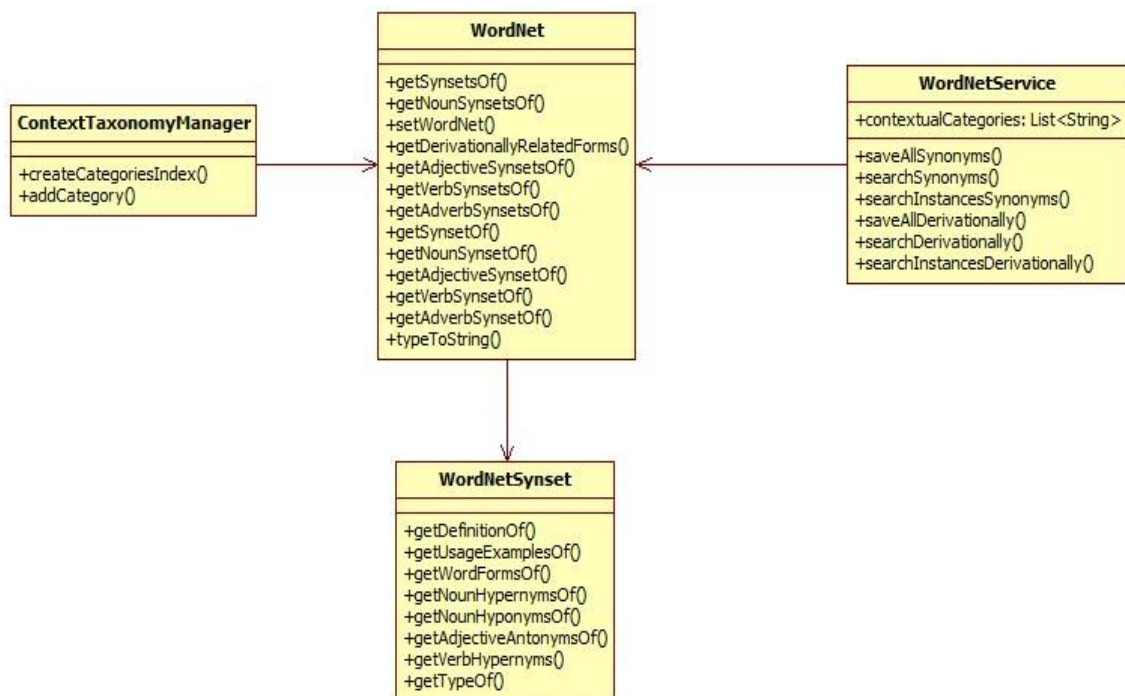


Figura 26: Diagrama de clases: WordNet.

A continuación se dará una breve descripción de cada una de las clases:

WordNet: La clase WordNet contiene los métodos para la búsqueda de sinónimos y palabras derivadas desde la base de datos.

WordNetService: Contiene los métodos para la búsqueda de sinónimos y palabras derivadas de cada categoría y cada instancia mediante la llamada a la clase WordNet, para luego almacenarlos en un objeto HashMap.

WordNetSynset: Contiene otros métodos para la búsqueda de sinónimos desde WordNet.

ContextTaxonomyManager: Contiene un método para la búsqueda de sinónimos y retornarlos en una lista de string.

5.6. Gestión de instancias

En el panel de la taxonomía o panel izquierdo de la nueva ventada desarrollada, es posible, mediante el click derecho del mouse en una categoría cualquiera, realizar diferentes acciones para la gestión de instancias. A continuación se especificará cada una de las acciones permitidas:

showIntancesOpt: Permite la búsqueda de instancias desde DBpedia de la categoría seleccionada y visualizarlas en una ventana desplegada, estas instancias sólo podrán ser visualizadas.

showSynOpt: Permite la búsqueda de sinónimos de la categoría seleccionada. A través del método `addCategory()` de la clase `ContextTaxonomyManager()`, se retorna una lista de sinónimos mediante una búsqueda en la base de datos de WordNet de una categoría específica y las visualiza en una nueva ventana desplegada.

loadInstanceAndSynToLeftPanel: Permite visualizar las instancias del nodo seleccionado en el panel izquierdo.

Click(): Al momento de seleccionar con el botón izquierdo del mouse una categoría de la taxonomía, automáticamente se visualizarán las instancias, sinónimos, palabras derivadas y

palabras ingresadas manualmente en el panel derecho de la ventana. También permite visualizar la cantidad de instancias que tiene la categoría seleccionada.

La Figura 27 representa el modelado de la clase mencionada.

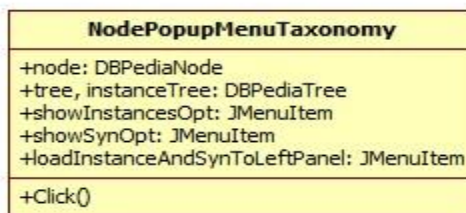


Figura 27: Diagrama de clases: NodePopupMenuTaxonomy.

Desde el panel de instancias o panel derecho de la pantalla secundaria es posible realizar diferentes opciones haciendo click con el botón derecho del mouse en alguna instancia del panel. Las opciones son:

deleteInstanceOpt: Permite eliminar la instancia seleccionada.

addInstanceOpt: Permite agregar una nueva instancia a la categoría actual.

ShowSynOpt: Permite buscar sinónimos desde WordNet de la instancia seleccionada y visualizarlos en una nueva ventana desplegada.

showDuplicatesInstances: Si en otras categorías existen instancias con el mismo nombre que la instancia seleccionada, éstas serán visualizadas en una ventana secundaria. Las instancias duplicadas se podrán gestionar mediante la clase **NodePopupMenuDuplicates**, la cual permite seleccionar las siguientes opciones:

deleteInstanceOpt: Permite eliminar la instancia duplicada de la categoría seleccionada desde la ventana desplegada.

showRoute: Permite ver la ruta de la taxonomía para encontrar la categoría en donde se encuentra la instancia repetida.

La Figura 28 muestra la ventana en donde se mostrarán las instancias duplicadas:

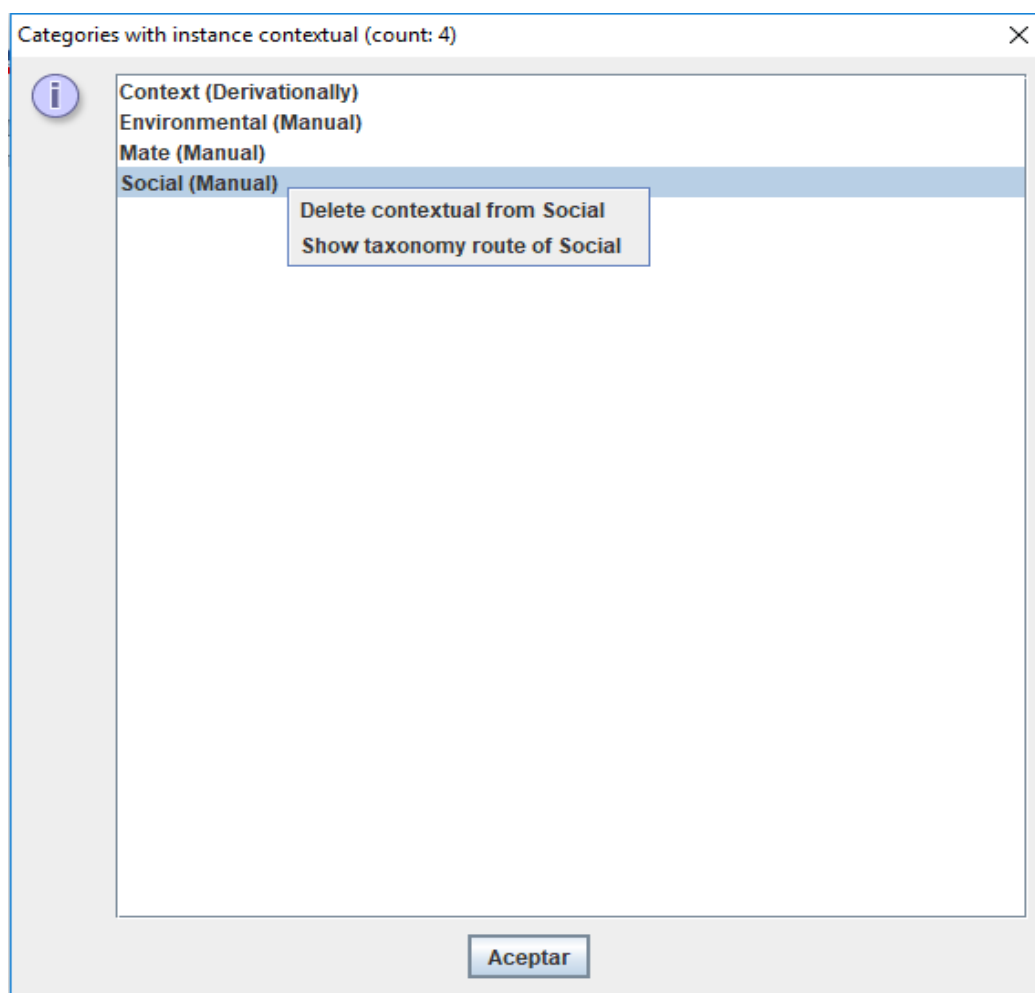


Figura 28: Panel instancias duplicadas.

El siguiente código muestra cómo se realiza la búsqueda de instancias duplicadas para luego visualizarlas:

```

252     Set<Map.Entry<String, List<String>>> set;
      set = secondaryFrame.categoryInstancesMap.entrySet();
254     for (Map.Entry<String, List<String>> me : set) {
255         List<String> strings = me.getValue();
          for (String string : strings) {
257             if (string.equals(node.getName())) {
                repeatedCategoryList.add(me.getKey().toString() + " (DBPedia)");
259             }
260         }
261     }

```

Figura 29: Código búsqueda instancias duplicadas.

La variable **set** de la línea 253 almacena los datos del HashMap **categoryInstancesMap**, con el fin de recorrer todas las instancias almacenadas en el mapa. El ciclo *for* de la línea 254 recorre el mapa guardado en **set**, la variable **me** va guardando cada lista de instancias asociadas a cada categoría de la taxonomía, hasta que no queden listas por consultar. Luego, en el ciclo *for* de la línea 256, para cada instancia de la lista se pregunta si tiene el mismo nombre que la instancia seleccionada por el usuario para consultar si tiene instancias duplicadas. Si la instancia está duplicada, en la lista **repeatedCategoryList** se guarda la categoría en la que esta instancia está almacenada.

El siguiente código permite visualizar las instancias duplicadas en un panel secundario:

```

263     JList<String> list = new JList(repeatedCategoryList.toArray());
264     JPanel panel = new JPanel();
265     JScrollPane scrollPane = new JScrollPane(list);
266     panel.add(scrollPane);

```

Figura 30: Código visualización instancias duplicadas.

En la línea 263 del código se crea una lista que guardará las categorías en donde se encuentran las instancias duplicadas encontradas. Por ejemplo, si se consulta por la instancia “Love”, todas las categorías que contengan la palabra “Love” estarán almacenadas en la lista **repeatedCategoryList**. Luego, en la línea 265 se crea un scrollPane, el cual permite crear una vista desplazable con las categorías de la lista. Finalmente, esta vista es añadida a un nuevo panel secundario que permitirá visualizar las categorías en donde se encuentran las instancias duplicadas.

La Figura 31 representa el modelado de las clases mencionadas.

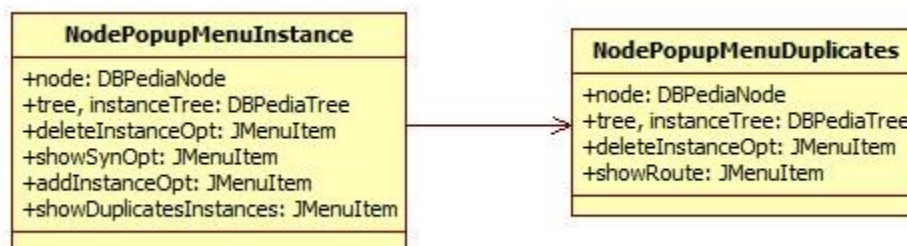


Figura 31: Diagrama de clases: NodePopupMenuInstance.

5.7. Menú pantalla secundaria

La Figura 32 muestra las pestañas del menú superior de la ventana secundaria de la aplicación, donde es posible realizar las siguientes acciones:

exportTreeXmlAction: Permite exportar todas las instancias, sinónimos, palabras derivadas y palabras agregadas manualmente de una categoría en particular visualizadas en el panel derecho de la ventana a un archivo .xml.

exportEditedTreeAction: Permite exportar todas las instancias, sinónimos, palabras derivadas y palabras agregadas manualmente a sus archivos correspondientes.

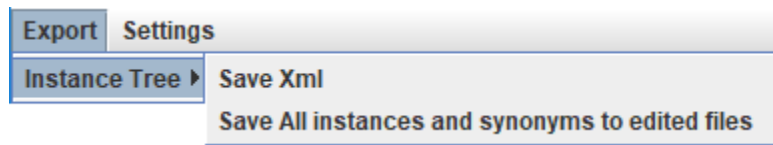


Figura 32: Diagrama de clases: Software actual.

La Figura 33 representa el modelado del menú mencionado.

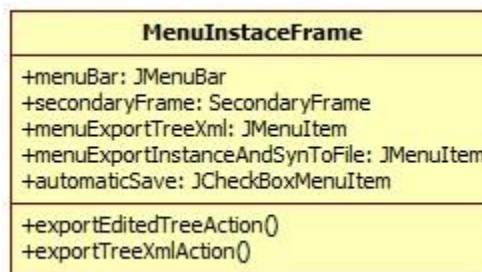


Figura 33: Diagrama de clases: MenuInstaceFrame.

Si el **JCheckBoxMenuItem** “automaticSave”, o como muestra la Figura 34, el casillero “Save edited files automatically”, está activado, se guardarán automáticamente todas las instancias, sinónimos y palabras derivadas en los archivos respectivos al momento de editar una instancia del panel derecho, ya sea eliminando o agregando una nueva instancia.

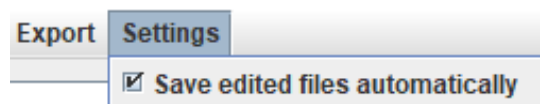


Figura 34: Diagrama de clases: Software actual.

El siguiente código muestra la implementación de este checkBox:

```

55      automaticSave.setSelected(true);
56      automaticSave.addActionListener(new ActionListener() {
57          @Override
58          public void actionPerformed(ActionEvent arg0) {
59              if (!automaticSave.getState()) {
60                  automaticSave.setSelected(false);
61                  autoSave = automaticSave.isSelected();
62              } else {
63                  automaticSave.setSelected(true);
64                  autoSave = automaticSave.isSelected();
65              }
66          }
67      });

```

Figura 35: Código guardado automático de archivos.

El código anterior contiene dos variables:

automaticSave de tipo JCheckBoxMenuItem: Si esta variable, de tipo checkBox, está activado, se exportarán las instancias automáticamente en sus archivos al momento de ser editados, si está desactivado no se guardará nada.

autoSave de tipo Boolean: Esta variable guardará el estado de automaticSave, si este último está activado, autoSave tendrá un valor **True** y si está desactivado tomará valor **False**. Para conocer si los archivos se guardarán automáticamente, se llama a esta variable desde cualquier clase para conocer el valor del boolean, por ejemplo:

```

import static es.uam.eps.irg.view.menu.MenuInstanceFrame.autoSave;

if (autoSave) {

    FileService fileService = new FileService();

    fileService.exportHashMapToFile(EditedContextInstancesPathAndFile,
    secondaryFrame.categoryInstancesMap);

}

```

Primero, se importa el boolean **autoSave** de la clase **MenuInstanceFrame**, luego, en la condición *if*, si el **checkBox** está activado y por lo tanto **autoSave** tiene un valor **True**, se ejecutarán las acciones dentro de la condición, en este caso se exportarán las instancias de las categorías a su archivo correspondiente.

5.8. Diagrama de la aplicación mejorada

Pantalla Principal

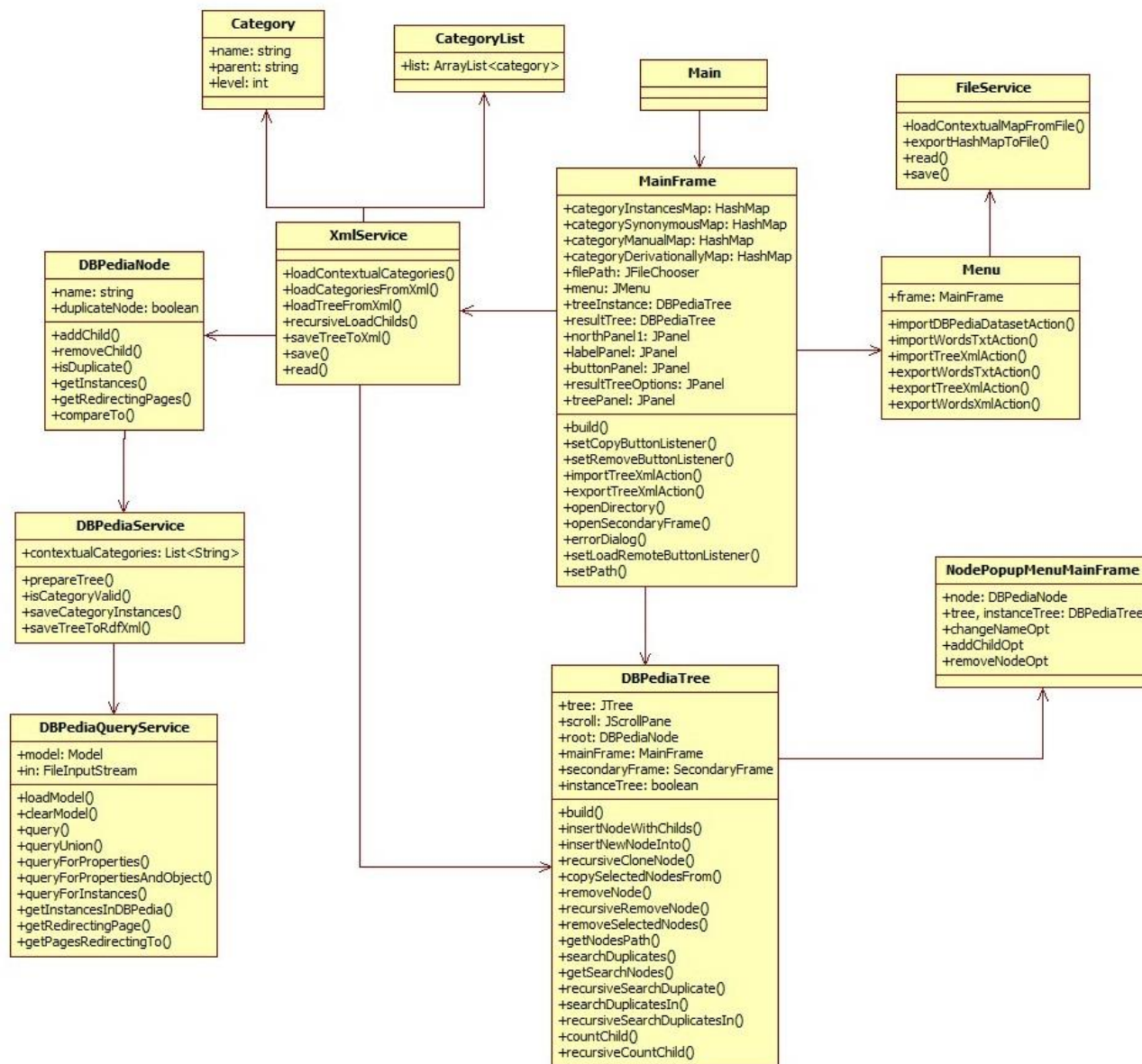


Figura 36: Diagrama de clases: Pantalla principal.

La Figura 36 muestra las clases de la pantalla principal de la aplicación. La pantalla principal abarca todo lo desarrollado en la investigación previa [1] más los nuevos botones y funcionalidades agregadas al software nuevo.

La clase **MainFrame** ahora permite seleccionar un directorio del dispositivo para guardar los diferentes archivos que se crearán y, mediante el botón “Instance Frame”, abrir la pantalla secundaria desarrollada en el nuevo software.

5.9. Otros elementos de Diseño

En la sección de “Diseño interfaz y navegación” se podrán visualizar dos diagramas: La Jerarquía de Menú y Esquema de Navegación. Estos diagramas sirven para conocer las opciones que tiene el usuario para acceder a las distintas funcionalidades del sistema. La Jerarquía de Menú representa un agrupamiento de las opciones de menú de la aplicación y el Esquema de Navegación representa las opciones que tiene el usuario para navegar dentro de las distintas funcionalidades.

Luego, se muestra un Árbol de descomposición funcional, el cual permite especificar las interrelaciones entre todos los módulos del software y la dependencia de ellos.

5.9.1. Diseño interfaz y navegación

Jerarquía de Menú

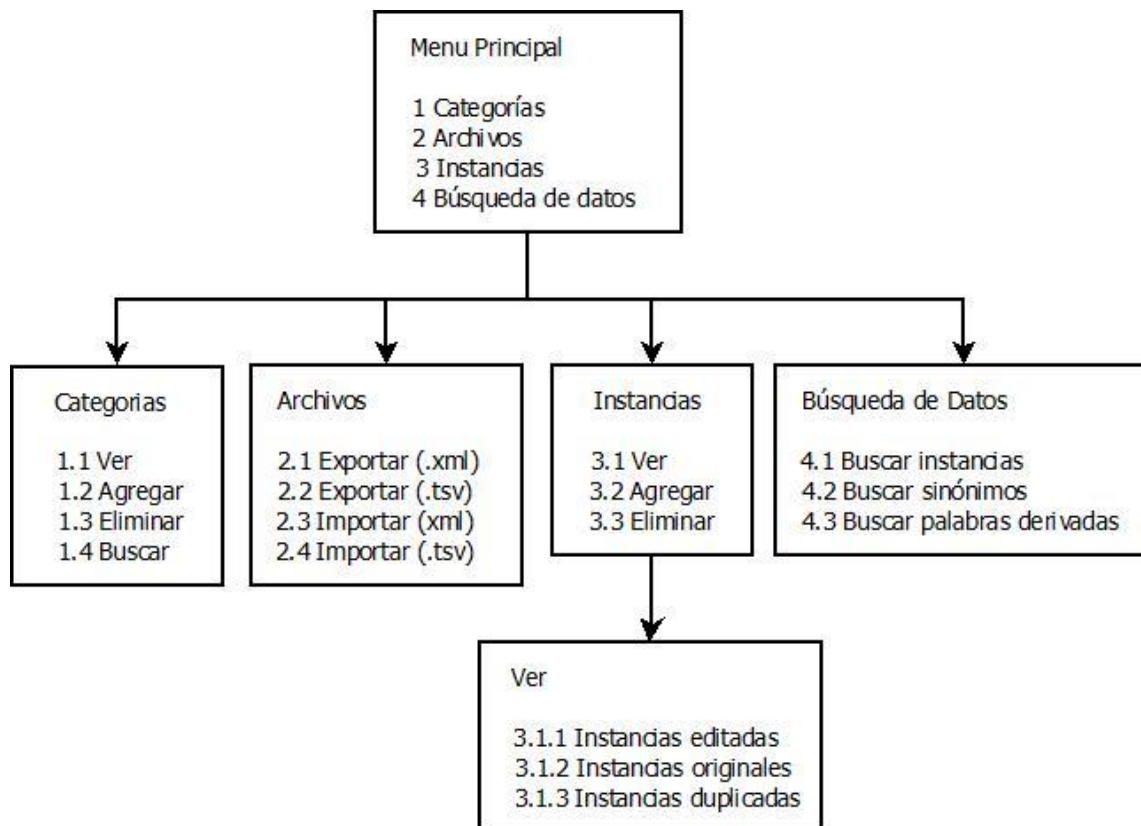


Figura 37: Jerarquía de menú.

Esquema de Navegación

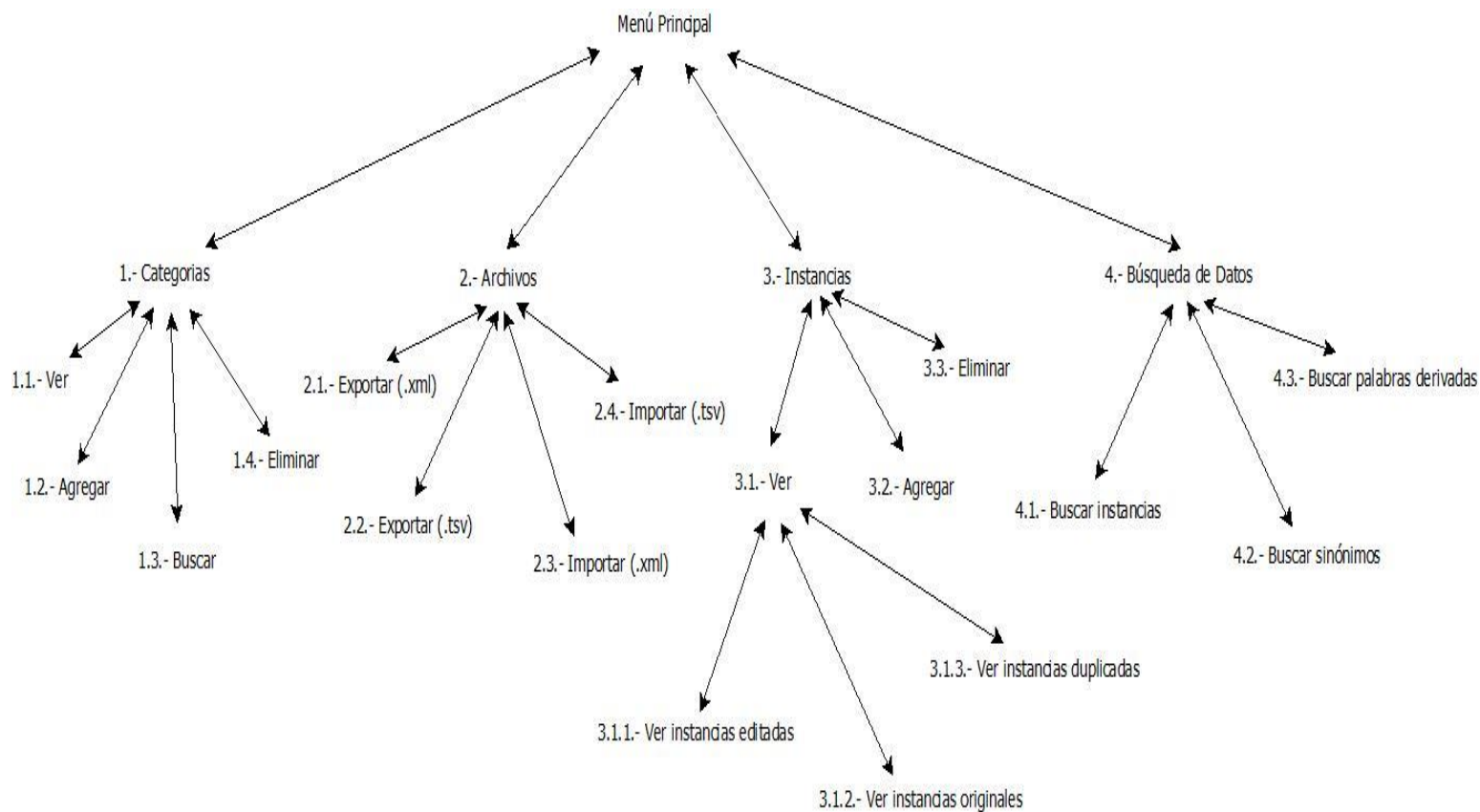


Figura 38: Esquema de navegación.

5.9.2. Diseño de arquitectura funcional

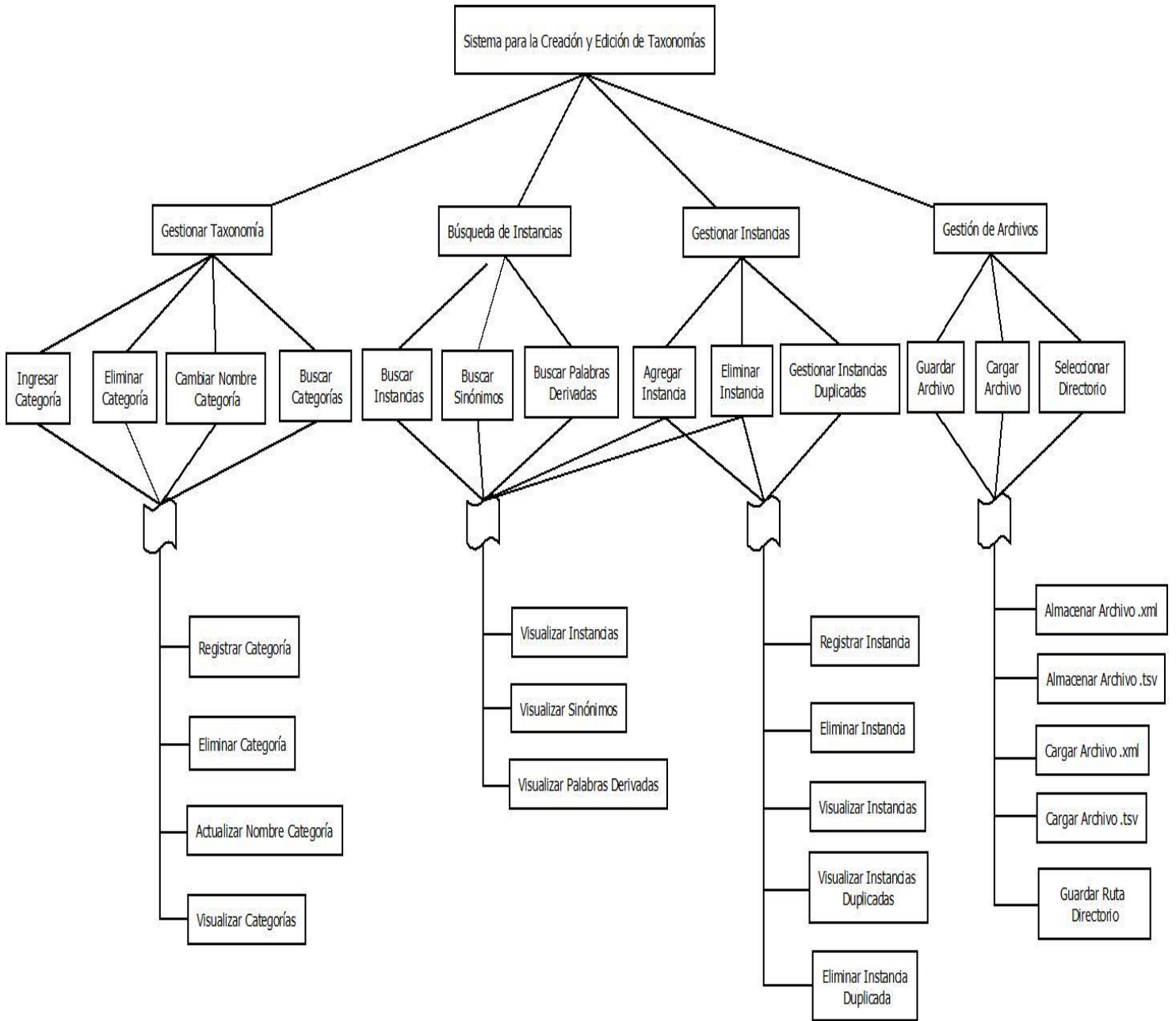


Figura 39: Árbol de descomposición funcional.

Capítulo 6. Pruebas

En el presente capítulo se presentarán las pruebas realizadas en la aplicación desarrollada. La importancia de realizar estas pruebas está estrechamente relacionada con la calidad de la taxonomía que se podrá crear en la aplicación.

Las pruebas se realizarán con el fin de comprobar que las nuevas funcionalidades relacionadas con la búsqueda de instancias, sinónimos y palabras derivadas extraídas de DBpedia y WordNet respectivamente sean las esperadas.

Para verificar que las salidas esperadas para una categoría de contexto sean las correctas, se buscarán las instancias en la página oficial de DBpedia mediante el ingreso de una categoría de contexto, desde donde se podrán visualizar las instancias encontradas en sus bases de datos. De manera similar para WordNet, se buscarán los sinónimos y palabras derivadas mediante el ingreso de una categoría de contexto en su página oficial, con el fin de visualizar las palabras encontradas en las bases de datos de WordNet.

6.1. Elementos de prueba

Los elementos de prueba corresponden a los casos de uso identificados en el Capítulo 4. Se realizarán pruebas para los casos de uso de búsqueda de instancias desde DBpedia, y búsqueda de sinónimos y palabras derivadas desde WordNet.

6.2. Pruebas de funcionalidad

En este ítem se realizarán las pruebas para comprobar las funcionalidades de búsqueda de información contextual desde DBpedia y WordNet de la aplicación. Las pruebas se realizarán mediante la búsqueda de instancias para cinco categorías de contexto elegidas al azar: Automobiles, cold, frienship, melancholia y shopping malls.

6.2.1. Búsqueda de instancias

Para la realización de las pruebas para la búsqueda de instancias desde DBpedia se creó la Tabla 3, donde se muestra cada categoría de contexto elegida para realizar las pruebas, las instancias encontradas en las bases de datos de la página web oficial de DBpedia, las cuales serán las salidas esperadas y las salidas obtenidas, que serán visualizadas mediante un pantallazo al panel de la aplicación con las instancias encontradas por ésta.


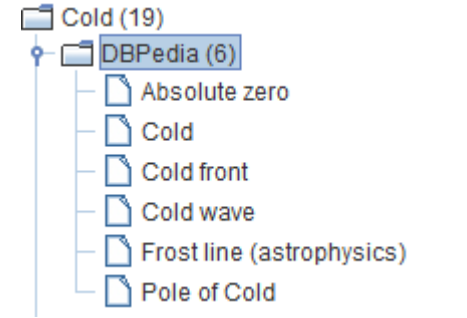
La Figura 40 da un ejemplo de cómo se visualizan las instancias desde la página web de DBpedia para una categoría ingresada, en este caso la categoría “Automobiles”:


is dct:subject of

- dbr:Indonesia_International_Motor_Show
- dbr:Vehicle_fire
- dbr:Car
- dbr:New_car_smell
- dbr:Amphibious_automobile
- dbr:Willow_Sports_Car
- dbr:Malaise_era
- dbr:Stance_(vehicle)

Figura 40: Instancias de “Automobiles” encontradas en DBPedia.

Estas instancias serán mostradas en la columna Salida Esperada de la Tabla 3.

Datos de Entrada	Salida Esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
Automobiles	<ol style="list-style-type: none"> 1. Indonesia International Motor Show 2. Vehicle fire 3. Car 4. New car smell 5. Amphibious automobile 6. Willow Sports Car 7. Malaise era 8. Stance (vehicle) 		Exitoso	Las ocho instancias de la Salida Esperada están contenidas en la Salida Obtenida.
Cold	<ol style="list-style-type: none"> 1. Cold front 2. Pole of Cold 3. Absolute zero 4. Frost line (astrophysics) 5. Cold 6. Cold wave 		Exitoso	Las seis instancias de la Salida Esperada están contenidas en la Salida Obtenida.

<p>Friendship</p>	<ol style="list-style-type: none"> 1. Internet relationship 2. Showmance 3. World Friendship Crusade 4. Visiting friends and relatives 5. Interspecies friendship 6. McKenzie friend 7. Cross-sex friendship 8. Pen pal 9. Platonic love 10. Friendship 11. Frenemy 12. Romantic friendship 13. Bromance 14. Life partner 15. Critical friend 16. Friendship bracelet 17. Friendship bench 18. Female bonding 19. Male bonding 20. Wilson the Volleyball 21. Womance 22. Same-sex intimacy 23. Best friends forever 24. Owen and Mzee 25. Man date 26. Princesses combinées 		<p>Exitoso</p>	<p>Las 26 instancias de la Salida Esperada están contenidas en la Salida Obtenida.</p>
-------------------	---	---	----------------	--

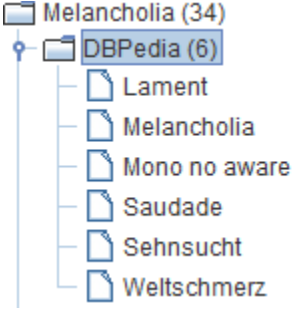
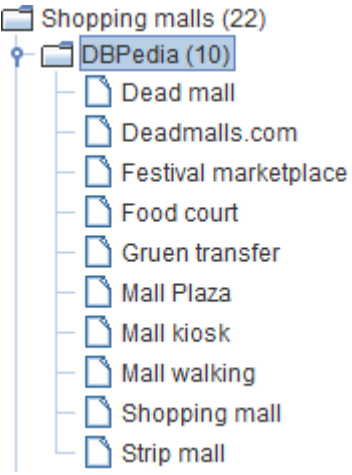
<p>Melancholia</p>	<ol style="list-style-type: none"> 1. Sehnsucht 2. Melancholia 3. Lament 4. Weltschmerz 5. Saudade 6. Mono no aware 		<p>Exitoso</p>	<p>Las seis instancias de la Salida Esperada están contenidas en la Salida Obtenida.</p>
<p>Shopping malls</p>	<ol style="list-style-type: none"> 1. Mall walking 2. Food court 3. Festival marketplace 4. Shopping mall 5. Strip mall 6. Dead mall 7. Mall Plaza 8. Gruen transfer 9. Deadmalls.com 10. Mall kiosk 		<p>Exitoso</p>	<p>Las 10 instancias de la Salida Esperada están contenidas en la Salida Obtenida.</p>

Tabla 3: Pruebas búsqueda de instancias.

6.2.2. Búsqueda de sinónimos

Para la realización de las pruebas para la búsqueda de sinónimos desde WordNet se creó la Tabla 4. Los datos de la columna Salida Esperada serán extraídos desde la página web oficial de WordNet, donde se visualizan los sinónimos para cada categoría de contexto de prueba ingresada en la página.

La aplicación desarrollada busca los sinónimos para cada categoría de la taxonomía y para cada instancia de cada categoría, por esta razón, para facilitar el proceso de pruebas se hará una comparación sólo con los sinónimos de las categorías de contexto encontradas, sin tener en cuenta los sinónimos de las instancias. Por tanto, los datos de la Salida Obtenida, además de contener los datos de la Salida Esperada, va a contener los sinónimos de las instancias de la categoría de contexto ingresada, las cuales no serán objetos de prueba.

La Figura 41 da un ejemplo de cómo se visualizan los sinónimos desde la página web de WordNet para una categoría ingresada por teclado. Los sinónimos estarán marcados con un círculo rojo. En este caso se ingresó la categoría “Automobiles”:

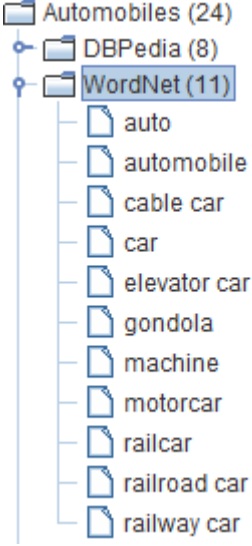
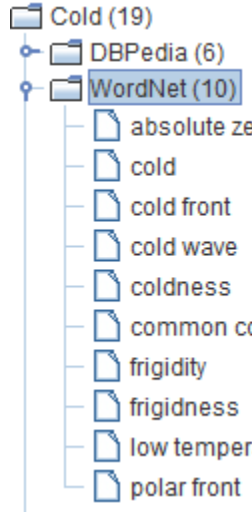
The image shows a screenshot of the WordNet Search interface. At the top, there is a header with the text "WordNet Search - 3.1" and navigation links: "- [WordNet home page](#) - [Glossary](#) - [Help](#)". Below the header, there is a search form with a text input field containing "Automobiles" and a "Search WordNet" button. Underneath, there are "Display Options" with a dropdown menu set to "(Select option to change)" and a "Change" button. A key is provided: "Key: 'S:' = Show Synset (semantic) relations, 'W:' = Show Word (lexical) relations". Below this, the display options for the sense are listed: "{offset} (gloss) 'an example sentence'".

The results are categorized by part of speech:

- Noun**
 - {02961779} [S:](#) (n) [car](#), [auto](#), [automobile](#), [machine](#), [motorcar](#) (a motor vehicle with four wheels; usually propelled by an internal combustion engine) "he needs a car to get to work"
- Verb**
 - {01934709} [S:](#) (v) [automobile](#) (travel in an automobile)

In the original image, the words "car", "auto", "automobile", "machine", and "motorcar" in the Noun result, and the word "automobile" in the Verb result, are circled in red.

Figura 41: Sinónimos de “Automobiles” encontrados en WordNet.

Datos de Entrada	Salida Esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
Automobiles	<ol style="list-style-type: none"> 1. Car 2. Auto 3. Automobile 4. Machine 5. Motorcar 		Exitoso	Los cinco sinónimos de la Salida Esperada están contenidos en la Salida Obtenida.
Cold	<ol style="list-style-type: none"> 1. Common gold 2. Coldness 3. Low temperature 4. Frigidity 5. Frigidness 6. Coldness 7. Sale 8. Dusty 9. Moth-eaten 10. Frigid 11. Cold_blooded 12. Inhuman 13. Insensate 		Fracaso	Los sinónimos del 7 al 13 de la Salida Espera no se encuentran en la Salida Obtenida. Estas seis palabras son adjetivos , por lo que se revisará la búsqueda de adjetivos en la aplicación.

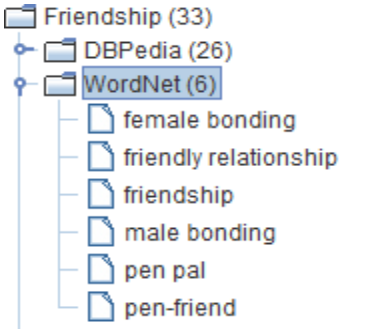
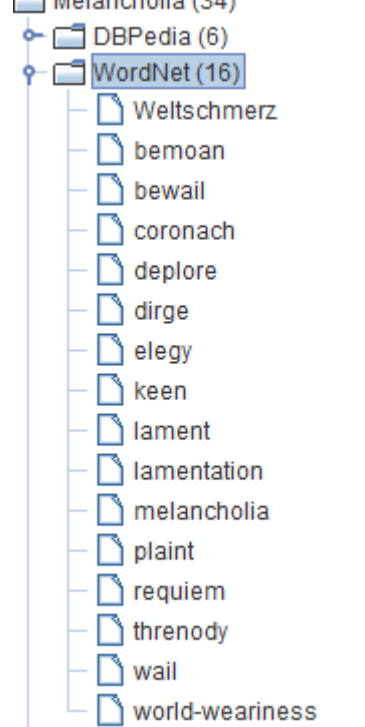

<p>Friendship</p>	<ol style="list-style-type: none"> 1. Friendship 2. Friendly relationship 		<p>Exitoso</p>	<p>Los dos sinónimos de la Salida Esperada están contenidos en la Salida Obtenida.</p>
<p>Melancholia</p>	<ol style="list-style-type: none"> 1. Melancholia 2. Lament 3. Lamentation 4. Plaint 5. Wail 6. Dirge 7. Coronach 8. Requiem 9. Threnody 10. Elegy 11. Keen 12. Deplore 13. Bewail 14. Bemoan 		<p>Exitoso</p>	<p>Los 14 sinónimos de la Salida Esperada están contenidos en la Salida Obtenida.</p>
<p>Shopping malls</p>	<ol style="list-style-type: none"> 1. Plaza 2. Mall 3. Center 4. Shopping mall 5. Shopping center 6. Shopping centre 		<p>Exitoso</p>	<p>Los seis sinónimos de la Salida Esperada están contenidos en la Salida Obtenida.</p>

Tabla 4: Pruebas búsqueda de sinónimos.

6.2.3. Búsqueda de palabras derivadas

Para la realización de las pruebas para la búsqueda de palabras derivadas desde WordNet se creó la Tabla 5. Los datos de la columna Salida Esperada serán extraídos desde la página web oficial de WordNet, donde se visualizan las palabras derivadas para cada categoría de contexto de prueba ingresada en la página.

La aplicación desarrollada busca las palabras derivadas para todas las categorías y para cada instancia y sinónimo de cada categoría. Por lo que para las pruebas de estas palabras se usarán sólo dos categorías, donde se buscarán las palabras derivadas para cada instancia y sinónimo de esta categoría en la página web de WordNet, con el objetivo comparar las Salidas Esperadas con las Salidas Obtenidas.

La Figura 42 da un ejemplo de cómo se visualizan las palabras derivadas desde la página web de WordNet para una categoría ingresada por teclado. Las palabras derivadas estarán marcadas con un círculo rojo. En este caso se ingresó la categoría “Automobiles”:

WordNet Search - 3.1
- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: {offset} (gloss) "an example sentence"

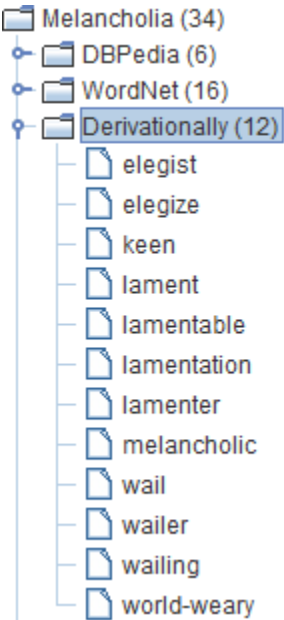
Noun

- {02961779} **S:** (n) [car](#), [auto](#), [automobile](#), [machine](#), [motorcar](#) (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
 - [direct hyponym](#) / [full hyponym](#)
 - [part meronym](#)
 - [domain term category](#)
 - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
 - **derivationally related form**
 - {10353814} **W:** (n) [automobilist](#) [Related to: [automobile](#)] (someone who drives (or travels in) an automobile)
 - {01934709} **W:** (v) [automobile](#) [Related to: [automobile](#)] (travel in an automobile)
 - {10298715} **W:** (n) [machinist](#) [Related to: [machine](#)] (a craftsman skilled in operating machine tools)

Verb

- {01934709} **S:** (v) [automobile](#) (travel in an automobile)

Figura 42: Palabras derivadas de “Automobiles” encontradas en WordNet.

Datos de Entrada	Salida Esperada	Salida Obtenida	Éxito/ Fracaso	Observaciones
Melancholia	<ol style="list-style-type: none"> 1. Melancholic 2. Lament (Lament) 3. Wail (Wail) 4. Elegist (Elegy) 5. Elegize (Elegy) 6. Lamentation (Lament) 7. Lamentor (Lament) 8. Lamentable (Lament) 9. Keen (Keen) 10. Wailer (Wail) 11. Wailing (Wail) 12. World-Weary (World-Weariness) 		Exitoso	<p>Las doce palabras derivadas de la Salida Esperada están contenidas en la Salida Obtenida.</p> <p>Las palabras entre paréntesis de la Salida Esperada son sinónimos de Melancholia, de los cuales se encontró la palabra derivada correspondiente.</p> <p>Las instancias de Melancholia no contenían palabras derivadas.</p>

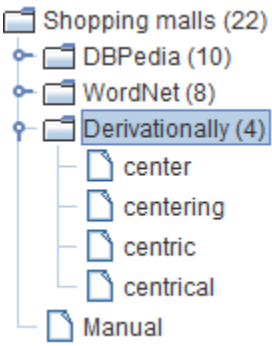
<p>Shopping malls</p>	<ol style="list-style-type: none"> 1. Central (Center) 2. Centric (Center) 3. Central (Center) 4. Centering (Center) 		<p>Exitoso</p>	<p>Las cuatro palabras derivadas de la Salida Esperada están contenidas en la Salida Obtenida.</p> <p>Las palabras entre paréntesis de la Salida Esperada son sinónimos de Shopping malls, de los cuales se encontró la palabra derivada correspondiente.</p>
-----------------------	--	--	----------------	---

Tabla 5: Pruebas búsqueda de palabras derivadas.

6.3 Conclusiones de pruebas

Las pruebas realizadas permitieron observar que las funcionalidades para la búsqueda de información contextual desde DBpedia y WordNet resultaron mayormente exitosas. A excepción de la búsqueda de sinónimos de tipo adjetivos para las categorías de contexto. Esto se puede deber a una diferencia entre la versión de la Base de Datos de WordNet disponible para descargar (2.1), y la versión disponible en línea (3.1), la cual es alimentada continuamente con nuevas palabras.

De los resultados expuestos anteriormente, se concluye que se ha logrado cumplir con los requerimientos funcionales para la búsqueda de instancias, sinónimos y palabras derivadas establecidas al inicio de este proyecto.

Conclusiones

De acuerdo a los objetivos específicos propuesto inicialmente, se puede comprobar que estos fueron cumplidos a cabalidad, dando como resultado una aplicación mejorada y con nuevas funcionalidades con respecto a la aplicación ya existente.

La investigación en la literatura sobre los diferentes temas tratados en este proyecto sirvió para conocer en mayor medida todo lo relacionado con los sistemas de recomendación y así crear una herramienta acorde a lo que se está investigando. Además de investigar sobre las bibliotecas y APIs utilizadas en el lenguaje Java utilizadas en el desarrollo del software, lo que aportó para la programación de una herramienta capaz de cumplir correctamente las mejoras y nuevas funcionalidades agregadas.

La aplicación permite crear una taxonomía de conceptos relacionadas con contexto mediante la búsqueda e inclusión de palabras contextuales desde DBpedia, además de sinónimos y palabras derivadas extraídas desde WordNet, generando así una taxonomía completa y enriquecida.

El software podrá ser de gran apoyo para los sistemas de recomendación sensibles al contexto y para las múltiples investigaciones que se están realizando y se realizarán en esta área de la investigación. Esto mediante la creación y utilización de la taxonomía generada por la aplicación.

Las herramientas y tecnologías utilizadas para la implementación del nuevo software significaron poner en práctica habilidades de investigación y desarrollo aprendidas a lo largo de la carrera. La investigación aportó significativamente para el aprendizaje de nuevas tecnologías utilizadas fuertemente en el área de la informática y para la construcción de las nuevas funcionalidades de la aplicación desarrollada.

Referencias

- [1] P. Campos, I. Cantador y N. Rodríguez. Extracting Context data from user reviews for recommendation: A Linked Data approach. In Proc. Workshop on Complex Recommendation (ComplexRec), Como, Italy, 2017.
- [2] Lehmann, Jens, et al. "DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia." *Semantic Web 6.2* (2015): 167-195.
- [3] Schuster, Daniel, et al. "Pervasive social context: Taxonomy and survey." *ACM Transactions on Intelligent Systems and Technology (TIST)* 4.3 (2013): 46.
- [4] <http://wiki.dbpedia.org/services-resources/interlinking>
- [5] Sarwar, Badrul, et al. "Item-Based Collaborative Filtering Recommendation Algorithms".
- [6] Balabanovic, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. *Communications of the ACM* **40**(3), 66–72 (1997)
- [7] Fernández, M. (1998). *Sistemas de recomendación*.
<http://ict.udlap.mx/people/lulu/documento/capitulo4.html>.
- [8] Panniello, U., Gorgoglione, M., Tuzhilin, A. 2016. In CARSs we trust: How context-aware recommendations affect customers' trust and other business
- [9] M. Hlava. What is a Taxonomy? 2012
- [10] <https://www.alexacom/topsites>
- [11] <https://en.wikipedia.org/wiki/Wikipedia>

performance measures of recommender systems. *Information Systems Research* 27(1), 182–196.

[12] J. Pérez, M. Arenas, C. Gutierrez. *Semantics and Complexity of SPARQL*.

[13] Miller, George A. *WordNet: A Lexical Database for English*. 1995.

[14] Cockburn, A. *Writing effective use cases, The crystal collection for software professionals*. Addison-Wesley Professional Reading, 2000.

[15] D. Berardi, D. Calvanese y G. De Giacomo. (2005) Reasoning on UML class diagrams. *Artificial intelligence*. 168(1-2), 70-118.

Anexo 1: Especificación de casos de uso

Casos de uso: Gestión de archivos

Caso de uso: Guardar archivo

Actor: Usuario

Descripción: El sistema debe permitir al usuario guardar un archivo en formato .xml o .tsv para almacenar una taxonomía o las instancias de cada categoría.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Elegir Directorio”.

Post-Condiciones: El sistema almacena los datos en los archivos respectivos.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona si desea guardar una taxonomía (archivo .xml) o las instancias (archivo .tsv).	
	2. El sistema valida la opción elegida por el usuario y deriva a la acción seleccionada.

Cursos alternos: No aplica.

Caso de uso: Almacenar taxonomía

Actor: Usuario

Descripción: El sistema debe permitir al usuario guardar un archivo en formato .xml para almacenar la taxonomía visible en el panel de la pantalla principal de la aplicación.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Elegir Directorio”.

Post-Condiciones: El sistema almacena los datos de la taxonomía en el archivo .xml.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona el botón “Save Tree” de la pantalla principal.	
	2. El sistema despliega una ventana para navegar por el dispositivo mostrando sólo los archivos xml.
3. El usuario elige el directorio donde quiere guardar el archivo, escribe el nombre del archivo que se creará o un archivo .xml ya existente para reemplazarlo y luego aprieta el botón “Guardar”.	
	4. El sistema guarda los datos de la taxonomía en el archivo creado.

Cursos alternos: No aplica.

Caso de uso: Almacenar instancias

Actor: Usuario

Descripción: El sistema debe permitir al usuario guardar archivos en formato .tsv para almacenar todas las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente de cada categoría de contexto.

Pre-Condiciones: ---

Post-Condiciones: El sistema almacena las instancias en los archivos correspondientes.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción "Save All instances and synonyms to edited files" en la pestaña "Export" de la ventana secundaria.	
	2. El sistema almacena las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente de cada categoría de contexto en los archivos correspondientes.

Cursos alternos:

Ítem 2: Si uno de los archivos .tsv no existe, el sistema creará el archivo automáticamente.

Caso de uso: Cargar archivo

Actor: Usuario

Descripción: El sistema debe permitir al usuario cargar un archivo en formato .xml o .tsv para obtener una taxonomía o las instancias de cada categoría respectivamente..

Pre-Condiciones: Se debe haber cumplido el flujo del CU "Elegir Directorio".

Post-Condiciones: El sistema obtiene la taxonomía o las instancias según la opción elegida por el usuario.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
------------------	--------------------

1. El caso de uso comienza cuando el usuario selecciona si desea cargar una taxonomía (archivo .xml) o las instancias (archivo .tsv).	
	2. El sistema valida la opción elegida por el usuario y deriva a la acción seleccionada.

Cursos alternos: No aplica.

Caso de uso: Obtener taxonomía

Actor: Usuario

Descripción: El sistema debe permitir al usuario cargar un archivo en formato .xml para obtener y visualizar una taxonomía en el panel de la pantalla principal de la aplicación.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Elegir Directorio”.

Post-Condiciones: El sistema carga y visualiza los datos de la taxonomía almacenados en el archivo .xml.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona el botón “Load” de la pantalla principal.	
	2. El sistema despliega una ventana donde se puede navegar por los directorios del dispositivo mostrando sólo los archivos .xml.
3. Busca y selecciona un archivo .xml con una taxonomía almacenada.	
	4. Carga los datos del archivo .xml y

	permite visualizar la taxonomía en el panel de la pantalla principal.
--	---

Cursos alternos:

Ítem 3: El usuario no selecciona ningún archivo y cancela. El sistema cierra la ventana desplegada.

Ítem 3: El usuario selecciona un archivo .xml el cual no contiene una taxonomía o faltan datos de ésta. Se indica un error y se cierra la ventana actual.

Caso de uso: Obtener instancias

Actor: Usuario

Descripción: El sistema debe permitir al usuario cargar las instancias de cada categoría de contexto almacenadas en los archivos .tsv.

Pre-Condiciones: Los archivos deben estar almacenados en el directorio seleccionado por el usuario.

Post-Condiciones: El sistema carga las carpetas respectivas para visualizar las instancias por cada categoría de la taxonomía.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario aprieta el botón “Show Instances Found” para visualizar las instancias originales encontradas en las bases de datos o el botón “Show Edited Instances” para visualizar las instancias editadas por el usuario.	
	2. Carga las instancias, sinónimos, palabras derivadas e instancias

	agregadas manualmente de los archivos .tsv correspondientes.
3. Podrá seleccionar una categoría del panel de la taxonomía para visualizar las instancias en el panel derecho de la pantalla secundaria.	

Cursos alternos:

Item 1: Si los archivos a cargar no existen en la carpeta seleccionada anteriormente por el usuario, el sistema muestra un error y no genera la carga de los archivos.

Casos de uso: Gestión de taxonomía

Caso de uso: Buscar categorías

Actor: Usuario

Descripción: El sistema debe permitir al usuario buscar instancias para una palabra ingresada por teclado.

Pre-Condiciones: ---

Post-Condiciones: El sistema busca y visualiza en pantalla las instancias encontradas.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. Ingresa una palabra en el cuadro “Initial Category” de la pantalla principal, además puede filtrar por cantidad de niveles, palabras que empiezan, contienen o terminan con una letra o palabra determinada.	

	2. Busca en DBpedia las instancias para la palabra ingresada por el usuario y las visualiza en un panel de la pantalla principal.
--	---

Cursos alternos:

Item 1: Si el usuario ingresa una palabra la cual no contiene instancias, el sistema sólo visualiza en el panel el nombre de la palabra ingresada.

Caso de uso: Gestionar taxonomía

Actor: Usuario

Descripción: El sistema debe permitir al usuario gestionar una taxonomía, en donde podrá:

- Agregar categoría
- Eliminar categoría
- Cambiar nombre categoría

Pre-Condiciones: ---

Post-Condiciones: El sistema actualiza la taxonomía según la acción del usuario.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario elige la opción entre agregar una categoría, eliminar categoría o cambiar el nombre a una categoría.	
	2. El sistema deriva a la acción seleccionada.

Cursos alternos: No aplica.

Caso de uso: Agregar categoría

Actor: Usuario

Descripción: El sistema debe permitir al usuario agregar un nodo al árbol de la taxonomía.

Pre-Condiciones: ---

Post-Condiciones: El sistema agrega el nodo o categoría a la taxonomía.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción “Add child to” haciendo click derecho en una categoría de la taxonomía.	
	2. El sistema muestra una ventana con un cuadro para ingresar el nombre de la categoría a ingresar.
3. Ingresa el nombre que desea para la nueva categoría.	
	4. Guarda la nueva categoría como hijo de la categoría seleccionada por el usuario en el punto 1.

Cursos alternos:

Item 3: Si el usuario no ingresa ninguna palabra, el sistema cierra el cuadro desplegado y no muestra nada.

Caso de uso: Eliminar categoría

Actor: Usuario

Descripción: El sistema debe permitir al usuario eliminar un nodo del árbol de la taxonomía.

Pre-Condiciones: Debe haber una taxonomía cargada o creada en la aplicación.

Post-Condiciones: El sistema elimina el nodo o categoría de la taxonomía.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción “Remove” haciendo click derecho en una categoría de la taxonomía.	
	2. Elimina la categoría seleccionada de la taxonomía.

Cursos alternos:

Item 2: Si la categoría seleccionada es el nodo raíz de la taxonomía, se indica error y no se elimina nada.

Caso de uso: Cambiar nombre categoría

Actor: Usuario

Descripción: El sistema debe permitir al usuario cambiar el nombre de una categoría del árbol de la taxonomía.

Pre-Condiciones: Debe haber una taxonomía cargada o creada en la aplicación.

Post-Condiciones: El sistema cambia el nombre de la categoría actualizada.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción	

“Change name of” haciendo click derecho en una categoría de la taxonomía.	
	2. El sistema muestra una ventana con un cuadro para ingresar el nuevo nombre de la categoría.
3. Ingresa el nuevo nombre que desea para la categoría seleccionada.	
	4. Actualiza el nombre de la categoría.

Cursos alternos:

Ítem 3: Si el usuario no ingresa ningún nombre, el sistema no cambia el nombre de la categoría seleccionada.

Casos de uso: Búsqueda de instancias

Caso de uso: Búsqueda de instancias, sinónimos y palabras derivadas

Actor: Usuario

Descripción: El sistema debe permitir al usuario buscar instancias, sinónimos y palabras derivadas para cada categoría de contexto de la taxonomía.

Pre-Condiciones: ---

Post-Condiciones: El sistema obtiene las instancias de cada categoría de contexto y las almacena en archivos .tsv.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
------------------	--------------------

<p>1. El caso de uso comienza cuando el usuario aprieta el botón “DBpedia Instances Search” para la búsqueda de instancias, “WordNet Synonyms Search” para la búsqueda de sinónimos o “WordNet Derivationally Search” para la búsqueda de palabras derivadas.</p>	
	<p>2. Busca los datos respectivos según la opción elegida por el usuario desde DBpedia o WordNet.</p>
	<p>3. Si la opción “Save edited files automatically” está activada, las instancias encontradas serán almacenadas en un archivo .tsv correspondiente al tipo de dato buscado.</p>

Cursos alternos:

Ítem 3: Si los archivos para almacenar las instancias no existen, el sistema los crea y guarda los datos correspondientes.

Casos de uso: Gestión de instancias

Caso de uso: Gestionar instancias

Actor: Usuario

Descripción: El sistema debe permitir al usuario gestionar las instancias, en donde podrá:

- Agregar instancia
- Eliminar instancia

- Ver instancias
- Ver instancias duplicadas

Pre-Condiciones: ---

Post-Condiciones: El sistema actualiza o visualiza las instancias según la acción del usuario.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
3. El caso de uso comienza cuando el usuario elige una opción para la gestión de una instancia.	
	4. El sistema valida la opción elegida por el usuario y deriva a la acción seleccionada.

Cursos alternos: No aplica.

Caso de uso: Agregar instancia

Actor: Usuario

Descripción: El sistema debe permitir al usuario agregar una instancia para una categoría de la taxonomía.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Obtener instancias”.

Post-Condiciones: El sistema agrega la instancia a la categoría correspondiente.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción “Add instance to” haciendo click derecho en algún nodo del panel de instancias	

de la pantalla secundaria.	
	2. Despliega una ventana con un cuadro para ingresar el nombre de la nueva instancia.
3. Ingresar el nombre que desea para la nueva instancia y selecciona el botón "Aceptar".	
	4. Guarda la instancia en la aplicación y la visualiza en la carpeta "Manual" del panel de instancias en la categoría correspondiente.
	5. Si la opción "Save edited files automatically" está activada, las instancias agregadas manualmente serán almacenadas en un archivo .tsv correspondiente a las instancias manuales.

Cursos alternos:

Ítem 3: Si el usuario no ingresa ningún nombre, el sistema cierra la ventana desplegada y no agrega ninguna instancia.

Ítem 5: Si los archivos para almacenar las instancias agregadas manualmente no existen, el sistema lo crea y guarda los datos correspondientes.

Caso de uso: Eliminar instancia

Actor: Usuario

Descripción: El sistema debe permitir al usuario eliminar una instancia de una categoría de la taxonomía.

Pre-Condiciones: Se debe haber cumplido el flujo del CU "Obtener instancias".

Post-Condiciones: El sistema elimina la instancia de la categoría correspondiente.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción “Delete” haciendo click derecho en alguna instancia del panel de instancias de la pantalla secundaria.	
	2. Elimina la instancia seleccionada de la aplicación y la borra del panel desde donde se está visualizando.
	3. Si la opción “Save edited files automatically” está activada, se almacenan las instancias, sinónimos, palabras derivadas o instancias agregadas manualmente en un archivo .tsv dependiendo desde qué carpeta se eliminó la instancia.

Cursos alternos:

Item 2: Si el usuario selecciona para eliminar una de las siguientes carpetas: nombre del nodo raíz, DBpedia, WordNet, Derivationally o Manual. Se indicará un error y no se borrará nada.

Caso de uso: Ver instancias

Actor: Usuario

Descripción: El sistema debe permitir al usuario ver todas las instancias seleccionando una categoría de contexto de la taxonomía.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Obtener instancias”.

Post-Condiciones: El sistema permite visualizar las instancias de la categoría seleccionada.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona una categoría del panel de la taxonomía con el botón izquierdo del mouse.	
	2. Visualiza automáticamente las instancias, sinónimos, palabras derivadas e instancias agregadas manualmente en el panel de instancias.

Cursos alternos: No aplica.

Caso de uso: Ver instancias duplicadas

Actor: Usuario

Descripción: El sistema debe permitir visualizar si una instancia seleccionada por el usuario está repetida en más de una categoría de contexto.

Pre-Condiciones: Se debe haber cumplido el flujo del CU “Obtener instancias”.

Post-Condiciones: El sistema muestra las categorías en donde se encuentra una instancia seleccionada por el usuario.

Flujo de eventos básicos:

Acción del Actor	Acción del Sistema
1. El caso de uso comienza cuando el usuario selecciona la opción “Show	

<p>duplicates of” haciendo click derecho en alguna instancia del panel de instancias de la pantalla secundaria.</p>	
	<p>2. Despliega una nueva ventana con el nombre de cada categoría en donde se encuentra la instancia seleccionada.</p>
<p>3. El usuario puede seleccionar una categoría de la ventana y apretar el botón “Remove” o seleccionar “See Route”.</p>	
	<p>4. Si se seleccionó la opción “Remove” se elimina la instancia de la categoría elegida por el usuario y se actualiza en el archivo .tsv. Si se seleccionó “See Route” se mostrará la ruta de la taxonomía para llegar a la categoría respectiva.</p>

Cursos alternos:

Item 3: Si el usuario no elige ninguna opción y cierra la ventana, el sistema vuelve a la pantalla anterior.