

eman ta zabal zazu



**ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA
INDUSTRIAL DE BILBAO**



**GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y
SISTEMAS DE INFORMACIÓN**

TRABAJO FIN DE GRADO

2017 / 2018

TITULO

**EVALUACIÓN AUTOMÁTICA DE ADIMEN-SUMO
USANDO EL BANCO DE PRUEBAS BLESS**

DATOS DE LA ALUMNA O DEL ALUMNO

NOMBRE: UNAI

APELLIDOS: GARCÍA VALLECILLO

DNI: 79051033x

FDO.:

FECHA:

DATOS DEL DIRECTOR O DE LA DIRECTORA

NOMBRE: JAVIER

APELLIDOS: ÁLVEZ GIMÉNEZ

DEPARTAMENTO: LSI

FDO.:

FECHA:

Índice de contenido

1	Introducción	1
1.1	Planteamiento del problema	1
1.2	Justificación y propósitos	2
1.3	Definiciones, Acrónimos y Abreviaturas	2
2	Documento de objetivos del proyecto.....	3
2.1	Objetivos	3
2.2	Alcance	3
2.2.1	Recursos materiales	3
2.2.2	Recursos humanos	4
2.2.3	Método de trabajo	4
2.3	Arquitectura	5
2.3.1	Herramientas.....	5
2.3.2	Licencia.....	6
2.4	Descripción de las tareas.....	7
2.4.1	Estructura de descomposición del trabajo (EDT)	7
2.4.2	Descripción de los procesos	8
2.5	Planificación temporal.....	13
2.5.1	Resumen de esfuerzo	13
2.5.2	Diagrama de Gantt	14
2.6	Evaluación de riesgos	15
2.6.1	Descripción y análisis según categoría	15
2.7	Evaluación económica	17
2.7.1	Salario.....	17
2.7.2	Amortización del equipo	17
2.7.3	Alquiler del lugar de trabajo.....	17
2.7.4	Software	17
2.7.5	Servidor	18
2.7.6	Gastos comunes	18
2.7.7	Ingresos	18
2.7.8	Total.....	19
3	Antecedentes	21
3.1	BLESS data (Baroni-Lenci Evaluation of Semantic Similarity).....	21
3.2	UKB: Graph Based Word Sense Disambiguation and Similarity.....	21
3.3	WordNet.....	22

3.4	SUMO y Adimen-SUMO	22
3.5	Mapping de WordNet a Adimen-SUMO.....	23
3.6	Relaciones Adimen-SUMO	23
3.7	Razonador automático.....	24
3.8	Metodologías de evaluación	24
4	Captura de requisitos	27
4.1	Casos de uso.....	27
4.2	Modelo de dominio.....	28
5	Análisis y diseño	31
5.1	Diagrama de clases.....	31
6	Desarrollo	35
6.1	Estructuras de datos.....	35
6.1.1	Listas.....	35
6.1.2	Diccionarios	35
6.2	Patrones de preguntas	36
6.3	Generación de preguntas.....	37
7	Análisis experimental de resultados	41
7.1	A nivel de pregunta	41
7.2	A nivel de pares	42
7.3	A nivel de palabra.....	42
7.4	A nivel de dominio.....	43
7.5	Análisis de resultados: conclusiones.....	45
8	Pruebas unitarias.....	47
8.1	Generación de preguntas.....	47
9	Conclusiones.....	49
9.1	Seguimiento	49
9.2	Reflexión personal.....	51
9.3	Líneas futuras	52
10	Bibliografía	53
11	Casos de uso extendidos	55
11.1	Desambiguación de Bless	55
11.2	Generación de preguntas.....	56
11.3	Análisis de resultados.....	57
12	Diagramas de secuencia.....	59
12.1	getBLESSsinRandom.py	59
12.2	bless2ukb.sh.....	60

12.3	getPreguntas.py	61
12.3.1	Carga de <i>Bless</i> y sus términos desambiguados	61
12.3.2	Carga de los <i>mappings</i> de <i>WordNet</i> y la ontología <i>Adimen-SUMO</i>	62
12.3.3	Carga de los diccionarios de <i>mappings</i> y ontología	63
12.3.4	Creación de preguntas y carga en el diccionario.....	64

Índice de ilustraciones

Ilustración 1 Arquitectura	5
Ilustración 2 EDT.....	7
Ilustración 3 Diagrama de GANTT	14
Ilustración 4 Casos de uso	27
Ilustración 5 Modelo de dominio	28
Ilustración 6 Diagrama de clases.....	31
Ilustración 7 Secuencia obtener una pregunta	33
Ilustración 8 Caso de uso Desambiguación de Bless.....	55
Ilustración 9 Caso de uso Generación de preguntas.....	56
Ilustración 10 Caso de uso Análisis de resultados.....	57
Ilustración 11 Secuencia getBLESSsinRandom.py	59
Ilustración 12 Secuencia bless2ukb.....	60
Ilustración 13 Secuencia carga de Bless y desambiguado	61
Ilustración 14 Secuencia carga mapping y ontología	62
Ilustración 15 Secuencia carga de diccionarios.....	63
Ilustración 16 Secuencia creación y carga de preguntas	64

Índice de tablas

Tabla 1 Herramientas	6
Tabla 2 Inicio y aprendizaje	8
Tabla 3 Aprendizaje Bless y desambiguación	8
Tabla 4 Instalación de herramientas y conexión al servidor	8
Tabla 5 Redacción del DOP	8
Tabla 6 Identificación de casos de uso	9
Tabla 7 Diseño de casos de uso	9
Tabla 8 Diseño de modelo de dominio	9
Tabla 9 Diseño de diagramas de secuencia	9
Tabla 10 Lectura de la ontología Bless	10
Tabla 11 Lectura de las relaciones de Wordnet	10
Tabla 12 Lectura de la ontología Adimen-SUMO	10
Tabla 13 Lectura de Mappings	10
Tabla 14 Definición de patrones de preguntas	11
Tabla 15 Generación de tests	11
Tabla 16 Análisis de soluciones	11
Tabla 17 Análisis de los Tests	11
Tabla 18 Pruebas unitarias	12
Tabla 19 Memoria	12
Tabla 20 Validación y entrega	12
Tabla 21 Exposición del proyecto	12
Tabla 22 Resumen esfuerzos planificados	13
Tabla 23 Riesgo - Baja médica	15
Tabla 24 Riesgo - Corte eléctrico	15
Tabla 25 Riesgo - Rotura del equipo	16
Tabla 26 Riesgo - Virus o problema con las herramientas	16
Tabla 27 Riesgo - Acceso a servidor LoRea + TFG	16
Tabla 28 Riesgo - Requisitos ocultos	16
Tabla 29 Coste total estimado del proyecto	19
Tabla 30 Relaciones 1 mapping WordNet	36
Tabla 31 Relaciones 2 mapping WordNet	36
Tabla 32 Ejemplo generación de pregunta	37
Tabla 33 Resultados a nivel de pregunta	41
Tabla 34 Resultados a nivel de pares	42
Tabla 35 Resultados a nivel de palabra criterio 1	43
Tabla 36 Resultados a nivel de palabra criterio 2	43
Tabla 37 Resultados a nivel de dominio	44
Tabla 38 Pruebas unitarias	48
Tabla 39 Resultado seguimiento	49
Tabla 40 Total recalculado coste del proyecto	50
Tabla 41 Caso de uso extendido Desambiguación de Bless	55
Tabla 42 Caso de uso extendido Generación de preguntas	56
Tabla 43 Caso de uso extendido Análisis de resultados	57

1 Introducción

Para comprender mejor el objetivo y el desarrollo de este proyecto es necesario conocer el fundamento del mismo. Comprender el significado de ontología en un ámbito informático.

Una ontología es un sistema de representación del conocimiento que es fruto de seleccionar un dominio o ámbito del conocimiento, y aplicar sobre él un método con el fin de obtener una representación formal de los conceptos que contiene y de las relaciones que existen entre dichos conceptos. Las ontologías introducen un mayor nivel de profundización semántica y proporcionan una descripción lógica y formal que puede ser interpretada tanto por las personas, como por las máquinas.

El grupo de investigación *LoRea* de la *EHU*, partiendo de los axiomas originales de la ontología *SUMO* ha desarrollado la ontología denominada *Adimen-SUMO*. El objetivo principal de dicha ontología es permitir la explotación del conocimiento de *SUMO* usando razonadores automáticos. Esta capacidad tiene múltiples aplicaciones en Sistemas Inteligentes, en Procesamiento del Lenguaje Natural, en Ingeniería del Conocimiento y en Web Semántica, entre otros.

Continuando con el proceso de mejora de *Adimen-SUMO*, Javier Álvez (miembro del grupo de investigación *LoRea*), propuso al alumno el desarrollo de un sistema de evaluación utilizando el banco de pruebas *Bless*.

1.1 Planteamiento del problema

Como parte de la evolución constante de la ontología *Adimen-SUMO* surge la necesidad de evaluar dicha ontología con diferentes datos. En este caso, se utilizará el banco de pruebas *Bless*, que incluye 200 nombres (sustantivos) concretos distintos como conceptos de objetivos que, a su vez se pueden agrupar en 17 clases semánticas más amplias.

Por ejemplo, un sustantivo concreto podría ser '*alligator*' que a su vez se puede considerar en el grupo '*amphibian_reptile*'.

Se desarrollará una aplicación que a partir de las palabras y sus relaciones del banco de pruebas *Bless* genere las preguntas que aplicará el razonador. Para ello, será necesario desambiguar dichas palabras y obtener la información necesaria de la ontología.

Posteriormente, con los resultados obtenidos se realizará un análisis para comprobar como de efectiva es la ontología para este conjunto de datos.

1.2 Justificación y propósitos

El desarrollo de esta aplicación permitirá aplicar diferentes esquemas de preguntas al razonador automático a partir de un banco de pruebas. De esta manera, podremos evaluar con mayor facilidad la ontología con diferentes datos.

1.3 Definiciones, Acrónimos y Abreviaturas

LoRea: Logic and Reasoning Group

SUMO: Suggested Upper Merged Ontology

BLESS: Baroni-Lenci Evaluation of Semantic Similarity

ATP: Automated Theorem Provider

PDF: Portable Document Format

EDT: Estructura de Descomposición del Trabajo.

Desambiguación: Consiste en obtener el significado concreto de una palabra dentro de un contexto. Es decir, identificar con qué sentido se usa una palabra en los términos de una oración.

2 Documento de objetivos del proyecto

2.1 Objetivos

Este proyecto tiene como objetivo evaluar la ontología *Adimen-SUMO* (1) y *WordNet* (2) utilizando el banco de pruebas *Bless* (3) para comprobar su relación en términos de la ontología.

Para realizar dicha tarea se utilizarán los siguientes recursos:

- Banco de pruebas *Bless* ¹
- Ontología *Adimen-SUMO* ²
- Mapping de *WordNet* a *Adimen-SUMO* (4) ²
- Relaciones *WordNet* ³
- Herramientas desarrolladas por el grupo *LoRea* y razonadores (ATPs) ²
- Servidor de *TFG* ⁴ + *LoRea*

Mediante la información obtenida de estos recursos, se definirán los patrones de preguntas dependiendo de cómo se hayan interpretado cada una de las relaciones de *Bless* en términos de la ontología.

Finalmente, gracias a los patrones de preguntas diseñados, se generará un conjunto de preguntas que serán utilizadas para la evaluación.

Una vez generadas dichas preguntas se llevará a cabo en el servidor del grupo *LoRea* la experimentación con los razonadores proporcionados para obtener los resultados y realizar el análisis a nivel de *Bless*.

2.2 Alcance

2.2.1 Recursos materiales

Además de materiales de oficina, tanto alumno como director emplearán ordenadores con un determinado software, una conexión a Internet para la comunicación, y la gestión del proyecto en un servidor web en la nube.

Todo el software que utilizaremos será de carácter gratuito o gratuito temporalmente para alumnos de la universidad.

1 - <https://sites.google.com/site/geometricalmodels/shared-evaluation>

2 - <http://adimen.si.ehu.es/web/AdimenSUMO>

3 - <https://wordnet.princeton.edu/>

4 - Servidor U012216

2.2.2 Recursos humanos

Las partes involucradas en la elaboración del siguiente proyecto son el alumno y el director. El primero se encargará de la investigación, adquisición de conocimientos necesarios y desarrollo del proyecto. Por otra parte, la función del segundo será la de orientar y aconsejar al alumno durante el mismo.

El tiempo total estimado para la realización de este Trabajo Fin de Grado es de 291 horas.

2.2.3 Método de trabajo

La comunicación entre alumno y director se llevará a cabo de forma telemática y presencial usando los recursos online disponibles considerados óptimos para cada situación.

Se realizarán reuniones periódicas presenciales, en la Escuela de Ingeniería de Bilbao (EIB), entre el director y el alumno para comentar el progreso del proyecto.

Las reuniones tendrán lugar cada 15 días, a pesar de que el número de ellas estará sujeto a modificaciones según necesidades del alumno.

Para la entrega de documentos oficiales, serán elaborados con la aplicación *Microsoft Office Word 2013*, empleando el formato *docx*, y para su entrega el formato PDF.

Ante posibles pérdidas de información, como medida de seguridad, se efectuarán copias de seguridad del trabajo realizado semanalmente o quincenalmente en un disco duro externo, además de en un servidor de alojamiento de archivos en la nube, concretamente *Dropbox*.

2.3 Arquitectura

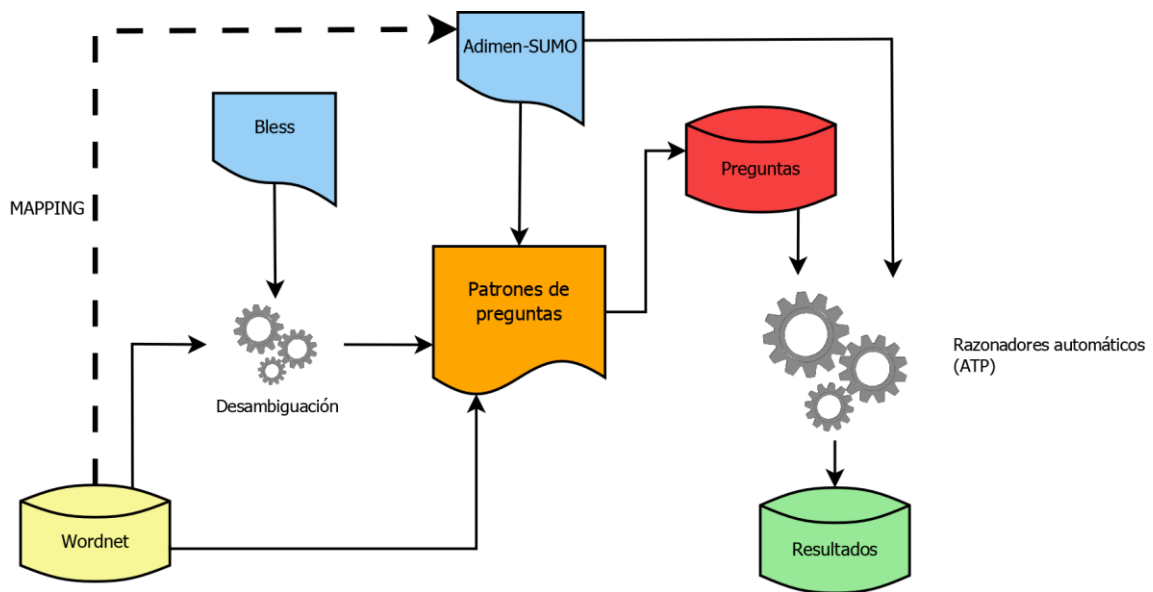


Ilustración 1 Arquitectura

2.3.1 Herramientas

A continuación, se detallarán las herramientas utilizadas para el desarrollo de este proyecto:

Herramienta	Descripción
<i>FileZilla</i> ¹	Cliente FTP gratuito para la visualización y transferencia de archivos con un servidor. Se utilizará para conectarse al servidor <i>LoRea + TFG</i> y así permitir el alojamiento y descarga de archivos del mismo.
<i>Notepad++</i> ²	Editor de texto gratuito orientado principalmente a programadores, ya que permite resaltar con colores la sintaxis de numerosos lenguajes de programación para facilitar su lectura. Se utilizará para la lectura de ficheros que contienen los conjuntos de hechos, <i>mappings</i> , esquemas de preguntas y resultados.
<i>Word 2013</i> ³	Procesador de texto para el desarrollo de documentos. Se utilizará para la realización de la documentación del proyecto.

<i>DIA</i> ⁴	Editor de diagramas. Se utilizará para la generación de diagramas y gráficos.
<i>Dropbox</i> ⁵	Servidor de alojamiento de archivos en la nube. Se utilizará para mantener copias de seguridad del trabajo realizado.
Servidor <i>LoRea</i> + Servidor <i>TFG</i>	Servidores en los cuales se alojará toda la información relacionada con el proyecto. En ellos se realizarán las experimentaciones necesarias para la obtención de los resultados a partir del conjunto de preguntas, mediante los razonadores.
<i>Microsoft Project</i> ⁶	Software para la planificación de proyectos. Se utilizará para realizar la planificación temporal mediante un diagrama de Gantt.
Razonador Automático Vampire ⁷	Demostador Automático de Teoremas (ATP) en lógica de primer orden.
<i>Bash</i> de <i>Ubuntu</i>	Desde <i>Windows 10</i> se utilizará esta herramienta para la conexión al servidor de trabajo.
<i>Anaconda Spyder</i> ⁸	Entorno de desarrollo para el lenguaje <i>Python</i> . Se utilizará para el desarrollo del proyecto.

Tabla 1 Herramientas

2.3.2 Licencia

La documentación de este proyecto y su código estarán protegidos bajo licencia de *Creative Commons Attribution-ShareAlike 3.0*. ⁹



1 - https://filezilla-project.org/client_features.php

2 - <https://notepad-plus-plus.org/>

3 - <https://products.office.com/es-es/microsoft-word-2013>

4 - <http://dia-installer.de/index.html.en>

5 - <https://www.dropbox.com/>

6 - <https://products.office.com/es-es/project/project-and-portfolio-management-software?tab=tabs-1>

7 - <https://vprover.github.io/>

8 - <https://anaconda.org/anaconda/spyder>

9 - <https://creativecommons.org/licenses/by-sa/3.0/es/>

2.4 Descripción de las tareas

A continuación, se detallarán en profundidad las tareas que componen el proyecto.

2.4.1 Estructura de descomposición del trabajo (EDT)

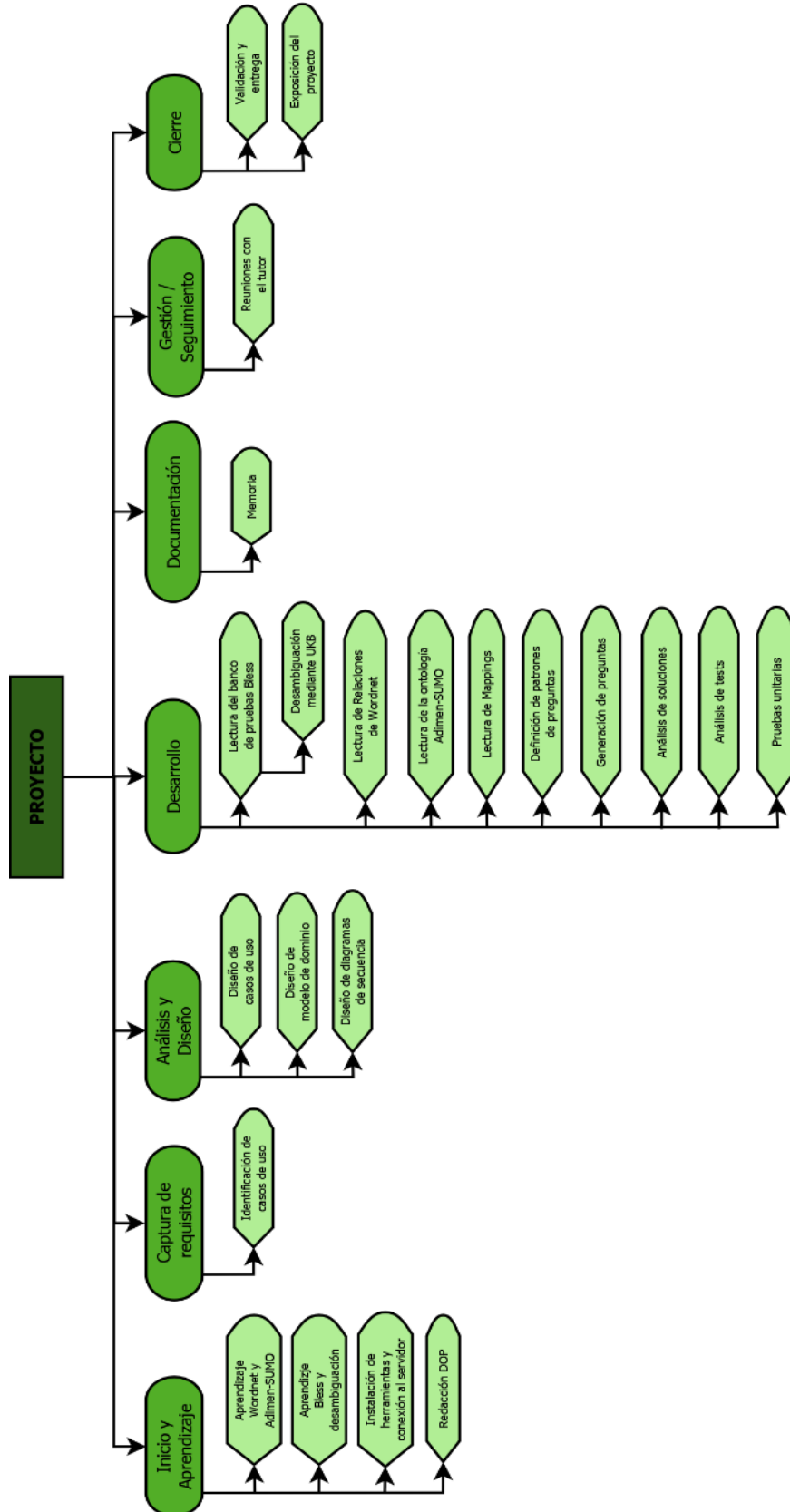


Ilustración 2 EDT

2.4.2 Descripción de los procesos

2.4.2.1 Inicio y aprendizaje

Aprendizaje de Wordnet y Adimen-SUMO	
Descripción	Aprendizaje de los recursos <i>WordNet</i> y <i>Adimen-SUMO</i> , acudiendo a reuniones de coordinación y seguimiento.
Esfuerzo	16 horas/persona.
Entradas	Ninguna.
Salidas/Entregables	Conocimientos necesarios para la realización del proyecto.
Recursos necesarios	Publicaciones y artículos sobre <i>WordNet</i> y <i>Adimen-SUMO</i> .

Tabla 2 Inicio y aprendizaje

Aprendizaje Bless y desambiguación	
Descripción	Aprendizaje sobre el banco de pruebas <i>Bless</i> que será utilizado en el proyecto y como desambiguar los términos.
Esfuerzo	16 horas/persona.
Entradas	Conjunto de datos <i>Bless</i> .
Salidas/Entregables	Conocimientos necesarios para la realización del proyecto.
Recursos necesarios	Publicaciones y artículos sobre <i>Bless</i> .

Tabla 3 Aprendizaje Bless y desambiguación

Instalación de herramientas y conexión al servidor	
Descripción	Instalación de las herramientas que se utilizarán para realizar el proyecto y comprobación de las conexiones con los recursos necesarios (Servidor <i>LoRea</i> + <i>TFG</i>).
Esfuerzo	2 horas/persona.
Entradas	Usuario creado en servidor <i>LoRea</i> , instaladores de herramientas.
Salidas/Entregables	Equipo preparado para el desarrollo.
Recursos necesarios	Herramientas y equipo.

Tabla 4 Instalación de herramientas y conexión al servidor

Redacción del DOP	
Descripción	Redacción del documento de objetivos del proyecto (DOP).
Esfuerzo	16 horas/persona.
Entradas	Aprendizaje de <i>Wordnet</i> y <i>Adimen-SUMO</i> , Aprendizaje <i>Bless</i> y desambiguación, Diagramas.
Salidas/Entregables	Documento de Objetivos del Proyecto (DOP).
Recursos necesarios	Procesador de texto (<i>Word 2013</i>).

Tabla 5 Redacción del DOP

2.4.2.2 *Captura de Requisitos*

Identificación de casos de uso	
Descripción	Identificación de los casos de uso involucrados.
Esfuerzo	8 horas/persona.
Entradas	Captura de requisitos.
Salidas/Entregables	Casos de uso.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), Editor de diagramas (<i>DIA</i>).

Tabla 6 Identificación de casos de uso

2.4.2.3 *Análisis y diseño*

Diseño de casos de uso	
Descripción	Diseño de casos de uso con su correspondiente documentación.
Esfuerzo	6 horas/persona.
Entradas	Captura de requisitos.
Salidas/Entregables	Diagramas de casos de uso.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), Editor de diagramas (<i>DIA</i>).

Tabla 7 Diseño de casos de uso

Diseño de modelo de dominio	
Descripción	Diseño del modelo de dominio con su correspondiente documentación.
Esfuerzo	6 horas/persona.
Entradas	Captura de requisitos.
Salidas/Entregables	Modelo de dominio.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), Editor de diagramas (<i>DIA</i>).

Tabla 8 Diseño de modelo de dominio

Diseño de diagramas de secuencia	
Descripción	Diseño de los diagramas de secuencia con su correspondiente documentación.
Esfuerzo	6 horas/persona.
Entradas	Captura de requisitos.
Salidas/Entregables	Diagramas de secuencia.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), Editor de diagramas (<i>DIA</i>).

Tabla 9 Diseño de diagramas de secuencia

2.4.2.4 Desarrollo

Lectura del banco de pruebas <i>Bless</i> y desambiguación mediante UKBs	
Descripción	Lectura del banco de pruebas <i>Bless</i> , y la desambiguación de los términos mediante UKBs.
Esfuerzo	8 horas/persona.
Entradas	Análisis y diseño.
Salidas/Entregables	Módulo de lectura del origen de datos <i>Bless</i> .
Recursos necesarios	Banco de pruebas <i>Bless</i> .

Tabla 10 Lectura de la ontología *Bless*

Lectura de las relaciones de <i>Wordnet</i>	
Descripción	Lectura de las relaciones de <i>WordNet</i> proporcionadas.
Esfuerzo	3 horas/persona.
Entradas	Análisis y diseño.
Salidas/Entregables	Módulo de lectura de relaciones de <i>WordNet</i> .
Recursos necesarios	Relaciones de <i>WordNet</i> .

Tabla 11 Lectura de las relaciones de *Wordnet*

Lectura de la ontología <i>Adimen-SUMO</i>	
Descripción	Lectura de la ontología <i>Adimen-SUMO</i> .
Esfuerzo	3 horas/persona.
Entradas	Análisis y diseño.
Salidas/Entregables	Módulo de lectura de la ontología <i>Adimen-SUMO</i> .
Recursos necesarios	Ontología <i>Adimen-SUMO</i> .

Tabla 12 Lectura de la ontología *Adimen-SUMO*

Lectura de Mappings	
Descripción	Lectura de los <i>mappings</i> de <i>WordNet</i> a <i>Adimen-SUMO</i> .
Esfuerzo	3 horas/persona.
Entradas	Análisis y diseño.
Salidas/Entregables	Módulo de lectura de <i>mappings</i> .
Recursos necesarios	<i>Mappings</i> .

Tabla 13 Lectura de *Mappings*

Definición de patrones de preguntas	
Descripción	Definir los patrones de preguntas en base a la información obtenida en las tareas anteriores.
Esfuerzo	30 horas/persona.
Entradas	Relaciones de <i>WordNet</i> , ontología <i>Adimen-SUMO</i> , banco de pruebas <i>Bless</i> , <i>mappings</i> , <i>synsets</i> clasificados.
Salidas/Entregables	Patrones de preguntas definidos.
Recursos necesarios	<i>Notepad++</i> .

Tabla 14 Definición de patrones de preguntas

Generación de preguntas	
Descripción	Generación de las preguntas mediante los patrones definidos.
Esfuerzo	30 horas/persona.
Entradas	Relaciones de <i>WordNet</i> , ontología <i>Adimen-SUMO</i> , banco de pruebas <i>Bless</i> , <i>mappings</i> .
Salidas/Entregables	Preguntas.
Recursos necesarios	<i>Notepad++</i> , <i>Bash de Ubuntu</i> .

Tabla 15 Generación de tests

Análisis de soluciones	
Descripción	Desarrollo del módulo de análisis de las soluciones obtenidas tras la experimentación mediante <i>ATPs</i> .
Esfuerzo	20 horas/persona.
Entradas	Soluciones.
Salidas/Entregables	Informe de soluciones.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), <i>Notepad++</i> .

Tabla 16 Análisis de soluciones

Análisis de los Tests	
Descripción	Desarrollo del módulo de análisis de tests basados en el banco de pruebas <i>Bless</i> .
Esfuerzo	20 horas/persona.
Entradas	Soluciones.
Salidas/Entregables	Informe de tests.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>), <i>Notepad++</i> .

Tabla 17 Análisis de los Tests

Pruebas unitarias	
Descripción	Realización de pruebas unitarias para la comprobación del correcto funcionamiento del código.
Esfuerzo	15 horas/persona.
Entradas	Implementación finalizada.
Salidas/Entregables	Código testeado.
Recursos necesarios	<i>Notepad++, Bash de Ubuntu.</i>

Tabla 18 Pruebas unitarias

2.4.2.5 Documentación

Memoria	
Descripción	Redacción de la memoria con toda la información referente al proyecto.
Esfuerzo	75 horas/persona.
Entradas	Implementación finalizada, DOP.
Salidas/Entregables	Memoria del TFG.
Recursos necesarios	Procesador de texto (<i>Word 2013</i>).

Tabla 19 Memoria

2.4.2.6 Cierre

Validación y entrega	
Descripción	Entregar la memoria final del proyecto y proceder a su validación.
Esfuerzo	2 horas/persona.
Entradas	Toda la documentación hecha hasta el momento y la implementación finalizada del proyecto.
Salidas/Entregables	Resguardo de entrega y asignación día de exposición.
Recursos necesarios	Ninguno.

Tabla 20 Validación y entrega

Exposición del proyecto	
Descripción	Preparación y exposición del proyecto utilizando medios audiovisuales si procede.
Esfuerzo	6 horas/persona.
Entradas	Memoria y proyecto finalizado.
Salidas/Entregables	Ninguna.
Recursos necesarios	Equipo en el aula de exposición, software ofimático.

Tabla 21 Exposición del proyecto

2.5 Planificación temporal

2.5.1 Resumen de esfuerzo

A continuación, se resumen los tiempos planificados de cada tarea.

Proceso	Duración
Inicio y aprendizaje	
Aprendizaje de <i>Wordnet</i> y <i>Adimen-SUMO</i>	16
Aprendizaje <i>Bless</i> y desambiguación	16
Instalación de herramientas y conexión al servidor	2
Redacción del DOP	16
Captura de requisitos	
Identificación de casos de uso	8
Análisis y diseño	
Diseño de casos de uso	6
Diseño de modelo de dominio	6
Diseño de diagramas de secuencia	6
Desarrollo	
Lectura del banco de pruebas <i>Bless</i> y desambiguación mediante <i>UKBs</i>	8
Lectura de las relaciones de <i>Wordnet</i>	3
Lectura de la ontología <i>Adimen-SUMO</i>	3
Lectura de <i>Mappings</i>	3
Definición de patrones de preguntas	30
Generación de preguntas	30
Análisis de soluciones	20
Análisis de los <i>Tests</i>	20
Pruebas unitarias	15
Documentación	
Memoria	75
Cierre	
Validación y entrega	2
Exposición del proyecto	6
Total	291

Tabla 22 Resumen esfuerzos planificados

2.5.2 Diagrama de Gantt

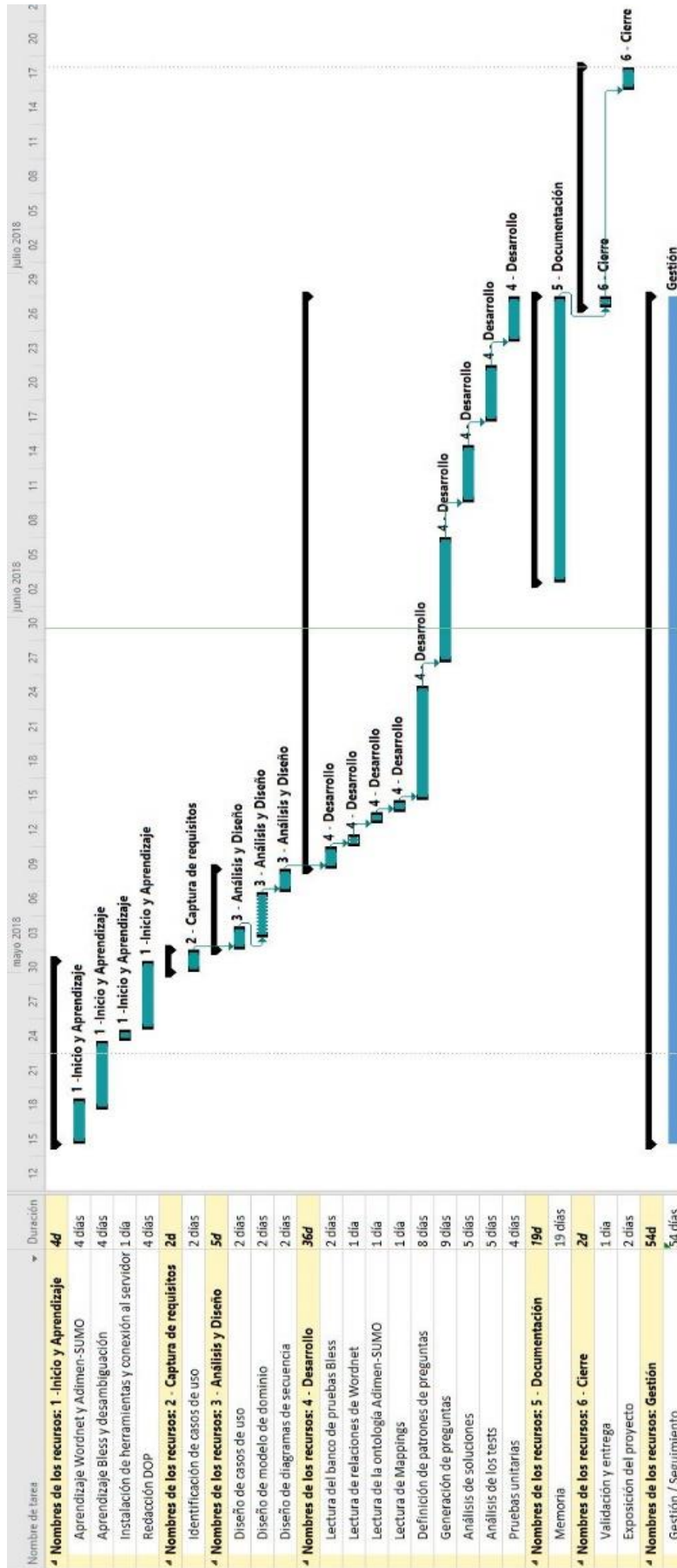


Ilustración 3 Diagrama de GANTT

2.6 Evaluación de riesgos

En esta sección se tratará de identificar los riesgos que puedan aparecer en el transcurso del proyecto, analizar su probabilidad de suceso, crear planes de prevención para intentar evitarlos y planes de contingencia para que una vez ocurrida la incidencia se logre minimizar las posibles consecuencias que se deriven de ese riesgo.

Se consideran los posibles impactos y probabilidades cuantificados en 5 niveles no numéricos: muy baja, baja, media, alta, muy alta.

Se analizan los riesgos de las siguientes categorías: personal, hardware, software y planificación.

2.6.1 Descripción y análisis según categoría

2.6.1.1 Personal

Baja médica:

Descripción	El alumno no puede trabajar en el proyecto por enfermedad.
Probabilidad	Baja.
Impacto	Alto.
Prevención	Intentar cumplir en la mayor medida posible los plazos de entrega planificados, para evitar retrasos.
Plan de contingencia	Cuando se reincorpore el alumno, realizar horas extra a las planificadas y priorizar el proyecto.

Tabla 23 Riesgo - Baja médica

2.6.1.2 Hardware

Corte eléctrico:

Descripción	Se produce un corte del suministro eléctrico.
Probabilidad	Muy baja.
Impacto	Bajo, ya que se tienen copias de seguridad.
Prevención	Utilizar el equipo (PC portátil) conectado al cargador y con la batería.
Plan de contingencia	Continuar en la medida de lo posible el trabajo en otro equipo disponible.

Tabla 24 Riesgo - Corte eléctrico

Rotura del equipo:

Descripción	Problemas con algún componente del ordenador.
Probabilidad	Media.
Impacto	Alto.
Prevención	Utilizar el equipo de manera responsable y mantener las copias de seguridad en un lugar accesible desde otro equipo.
Plan de contingencia	Reparar o sustituir el equipo.

Tabla 25 Riesgo - Rotura del equipo

2.6.1.3 Software

Virus o problema con las herramientas:

Descripción	Ser afectado por un virus o malware, o cualquier incidente con las herramientas utilizadas en el proyecto.
Probabilidad	Baja
Impacto	Alto, causaría retrasos.
Prevención	Mantener la política de copias de seguridad y utilizar un antivirus en el equipo.
Plan de contingencia	Eliminar el virus o malware y restaurar las partes afectadas.

Tabla 26 Riesgo - Virus o problema con las herramientas

2.6.1.4 Externos

Acceso a servidor LoRea + TFG:

Descripción	Problemas para lanzar las ejecuciones en el servidor LoRea.
Probabilidad	Media.
Impacto	Bajo.
Prevención	No aplica.
Plan de contingencia	Continuar con las tareas que no dependen del servidor.

Tabla 27 Riesgo - Acceso a servidor LoRea + TFG

Requisitos ocultos:

Descripción	Riesgos que no se pueden identificar.
Probabilidad	Bajo.
Impacto	Medio.
Prevención	Contar con un margen de tiempo para este tipo de problemas.
Plan de contingencia	Continuar con tareas que se puedan realizar y tratar de solucionar los problemas invirtiendo más tiempo.

Tabla 28 Riesgo - Requisitos ocultos

2.7 Evaluación económica

La finalidad de este Trabajo Fin de Grado no es de carácter comercial, por lo que no se prevén ingresos derivados del mismo. A continuación, se presentan los costes estimados de la realización de este proyecto como dato orientativo.

2.7.1 Salario

Horas totales: 291 h

Coste hora: 32 €/h

Total coste (€) = 291 h * 32 €/h = 9.312 €

2.7.2 Amortización del equipo

Precio: 750 €

Unidades: 1.

Tiempo estimado para su total amortización: 6 años (72 meses).

Horas jornada: 4 h/día de Lunes a Viernes.

Tiempo de uso en el proyecto: $\frac{291 \text{ h}}{4 \text{ h/día}} \times \frac{1}{5 \text{ días/sem}} \times \frac{1}{4 \text{ sem/mes}} = 3,637 \text{ meses.}$

Amortización equipo = (Coste/Duración estimada)*Tiempo de uso*Unidades
 $(750/72) * 3,637 * 1 = 37,89 \text{ €}$

2.7.3 Alquiler del lugar de trabajo

El lugar de trabajo será el domicilio del alumno, por lo tanto, se considera 0€ el coste del alquiler del lugar de trabajo.

2.7.4 Software

Las herramientas utilizadas son de carácter libre o de licencias temporales adquiridas a través de la UPV/EHU.

El proyecto tendrá naturaleza de software libre bajo licencia *Creative Commons Attribution-ShareAlike 3.0*. Por lo tanto, se considera el coste de licencias 0€ y se ha realizado con herramientas gratuitas de distribución libre o de código abierto, exceptuando la herramienta *Microsoft Office 2013*. Cuya licencia ha sido adquirida con el equipo en el cual se desarrollará el proyecto.

2.7.5 Servidor

En cuanto a los costes de ejecución en el servidor, se calculará teniendo en cuenta el número de datos que se utilizan de *Bless*.

En *Bless* existen 26554 pares de palabras, y se estima que el número de preguntas que se obtendrán cumplirá una relación de 1 a 5. Es decir, 1 pregunta por cada 5 pares (o un 20%), 5310 preguntas. Como además también se ejecutará la negación de cada pregunta, el número total de *tests* a evaluar será 10620.

Tiempo de ejecución: $5310 \times 2 \times 5 \text{ min/pregunta} = 53100 \text{ minutos} = 885 \text{ horas}$

Coste de ejecución: $885 \text{ horas} \times 0.27 \text{ €/hora} = 238.95 \text{ €}$

El coste de ejecución se ha calculado teniendo en cuenta el coste por hora de *Microsoft Azure Cloud Services* ¹.

2.7.6 Gastos comunes

Los gastos pertenecientes a este apartado son aquellos originados por las tareas que se deben realizar durante el proyecto. Como material de oficina, internet, electricidad o teléfono. Su coste será el 5% del coste total del proyecto.

2.7.7 Ingresos

En cuanto a los ingresos que se esperan tras la realización del trabajo, cabe destacar que al tratarse de un proyecto de investigación no se obtendría ningún tipo de ingreso.

1 - <https://azure.microsoft.com/es-es/pricing/details/cloud-services/>

2.7.8 Total

Concepto	Importe
Salario	9.312 €
Amortización del equipo	37,89 €
Alquiler del lugar de trabajo	0 €
Software	0 €
Servidor	238,95 €
Subtotal	9.588,84 €
Gastos comunes	526,03 €
TOTAL	10.068,28 €

Tabla 29 Coste total estimado del proyecto

3 Antecedentes

3.1 BLESS data (Baroni-Lenci Evaluation of Semantic Similarity)

Bless (1) se trata de un conjunto de datos adecuado para la evaluación de modelos de distribución. Este conjunto de datos incluye nombres (sustantivos) concretos pertenecientes a diferentes clases semánticas (vivos, no vivos, etc.) con conjuntos asociados de otras palabras para relaciones semánticas específicas, tales como "atributo", "categoría coordinada", "evento" o "metonimia".

En específico este banco de pruebas incluye 200 nombres concretos (100 nombres animados y 100 nombres inanimados) de 17 dominios diferentes (por ejemplo, ropa, herramientas, vehículos, animales, etc.). Cada nombre objetivo está asociado con un conjunto de otras palabras (nombres, verbos o adjetivos) a través de una relación semántica (hiperonimia, cohiponimia, meronimia, atributo, evento o aleatorio).

3.2 UKB: Graph Based Word Sense Disambiguation and Similarity

UKB (2) es una colección de programas que permitirá realizar la desambiguación de los sentidos de las palabras basada en gráficos y la similitud léxica. Desempeña esta tarea utilizando una base de conocimiento preexistente.

UKB ha sido desarrollado por el grupo *IXA* en la Universidad del País Vasco. Aplica el denominado *PageRank* personalizado en una Base de conocimiento léxico (LKB) para clasificar los vértices de la *LKB* y así realizar la desambiguación. El algoritmo también se puede usar para calcular la similitud o relación léxica de palabras u oraciones.

Bless - <https://sites.google.com/site/geometricalmodels/shared-evaluation>

UKB - <https://github.com/asorooa/ukb>

IXA - <http://ixa.si.ehu.es/ixa>

3.3 WordNet

WordNet (2) es una base de datos léxica del idioma inglés que se encuentra disponible online de forma gratuita.

Contiene información codificada manualmente sobre nombres, verbos, adjetivos y adverbios del inglés, y está organizada entorno a la noción de *synset*. Un *synset* es un conjunto de palabras de la misma categoría morfosintáctica que pueden ser intercambiados en un contexto dado. Un *synset* es comúnmente descrito por una glosa o definición y por un conjunto explícito de relaciones semánticas con otros *synsets*.

Cada *synset* representa un concepto que está relacionado con otros conceptos mediante una gran variedad de relaciones semánticas, incluyendo hiperonimia/hiponimia, meronimia/holonimia, antonimia, etc. WordNet también codifica 26 tipos diferentes de dichas relaciones.

3.4 SUMO y Adimen-SUMO

SUMO (*Suggested Upper Merged Ontology*) (4), es una ontología que fue creada por el *IEEE Standard Upper Ontology Working Group*. Su objetivo era desarrollar una ontología estándar de alto nivel para promover el intercambio de datos, la búsqueda y extracción de información, la inferencia automática y el procesamiento del lenguaje natural.

SUMO provee definiciones para términos de propósito general resultantes de fusionar diferentes ontologías libres de alto nivel.

Adimen-SUMO (3) es una reconversión de *SUMO* a una ontología de primer orden operativa. Está escrita en el lenguaje *Adimen*, un lenguaje ontológico basado en lógica de primer orden diseñado originalmente para especificar la ontología *Adimen-SUMO*. Ha sido desarrollado conjuntamente por los Grupos *LoRea* e *IXA* de la Universidad del País Vasco (UPV/EHU).

Así, *Adimen-SUMO* puede ser utilizada para el razonamiento formal por demostradores automáticos de teoremas (*ATP*) en lógica de primer orden, como puede ser Vampire.

WordNet - <https://wordnet.princeton.edu/>
LoRea - <http://www.sc.ehu.es/jiwnagom/PaginaWebLorea>
Vampire - <https://vprover.github.io>
Adimen-SUMO - <http://adimen.si.ehu.es/web/AdimenSUMO>
SUMO - <http://www.adampease.org/OP/>

3.5 Mapping de WordNet a Adimen-SUMO

WordNet y *SUMO* están conectados mediante un *mapping* semántico (7), el cual será utilizado en este proyecto. Es importante comprender las relaciones que contiene:

- *Equivalence*: Ambos conceptos son semánticamente similares.
- *Subsumption*: El concepto de *SUMO* es más general que el *synset* de *WordNet*.
- *Instance*: El concepto de *WordNet* es un caso particular del concepto de *SUMO*.

Dicho *mapping* ha sido portado a Adimen-SUMO manteniendo las relaciones. En particular, se utilizará una versión del mapping entre *WordNet* y *Adimen-SUMO* en formato *Prolog* dividida en varios ficheros que contienen el mapeo correspondiente de cada *synset* y su relación.

El formato de este *mapping* es el siguiente:

```
verbMapping2AdimenSUMO('01482075-v',[ ('PhysicalAttribute','Attribute',subsumption)]).
verbMapping2AdimenSUMO('01482285-v',[ ('Motion',class,subsumption)]).
verbMapping2AdimenSUMO('01482449-v',[ ('Putting',class,subsumption)]).
```

3.6 Relaciones Adimen-SUMO

Por cada relación definida en *WordNet*, se van a utilizar varias relaciones definidas en la ontología *Adimen-SUMO*. Estas relaciones tienen limitado su uso mediante restricciones de dominio, por las cuáles se define el subconjunto de conceptos de *Adimen-SUMO* a los que se puede aplicar la relación.

Teniendo en cuenta las relaciones que contiene *Bless*, se van a utilizar ciertas relaciones de la ontología *Adimen-SUMO*. Para la posterior generación de preguntas, será necesario comprobar ciertas restricciones por cada término en la ontología.

Para que un *synset* cumpla la restricción de dominio de una relación de la ontología *Adimen-SUMO*, será necesario que el concepto de *Adimen-SUMO* al cual está relacionado mediante el *mapping* sea subtipo del concepto indicado en la restricción de dominio o viceversa.

A continuación, se detallarán las restricciones que debe cumplir cada *synset* para que pueda ser generada su pregunta correspondiente:

Si la relación en *Bless* es *attri*:

- Cada *mapping* del *synset* 2 debe cumplir las restricciones del concepto *Attribute* de la ontología.

Si la relación en *Bless* es *coord*:

- No se debe cumplir ninguna restricción especial.

Si la relación en *Bless* es *event*:

- Cada *mapping* del *synset* 1 debe cumplir las restricciones del concepto *Object* de la ontología.
- Cada *mapping* del *synset* 2 debe cumplir las restricciones del concepto *Process* de la ontología.

Si la relación en *Bless* es *hyper*:

- No se debe cumplir ninguna restricción especial.

Si la relación en *Bless* es *mero*:

- Cada *mapping* del *synset* 1 y del *synset* 2 deben cumplir las restricciones del concepto *Object* de la ontología.

3.7 Razonador automático

Un razonador automático es una herramienta automatizada que sirve para decidir si una fórmula es consecuencia lógica de otra. Únicamente se utilizarán razonadores basados en lógica de primer orden, y concretamente el razonador utilizado en este proyecto es *Vampire* (6).

3.8 Metodologías de evaluación

Como método de evaluación se propone la división del conjunto de preguntas en dos clases. De esta manera:

- Las conjeturas que se esperan que se deduzcan de la ontología se denominarán ***truth-tests***.
- Las conjeturas que se esperan que no se deduzcan de la ontología se denominarán ***falsity-tests***.

Un ejemplo de *truth-test* puede ser la conjetura “Los hermanos tienen la misma madre”, porque se espera que esté correctamente relacionada.

Por el contrario, un ejemplo de *falsity-test* puede ser “Los herbívoros comen animales”, ya que se espera que no esté implicada por la ontología.

Para superar el problema de decidir si las preguntas generadas están implicadas o no por la ontología usando ATP, proponemos la clasificación como *passing*, *nonpassing* o *unknown* usando los siguientes criterios:

- Si el ATP encuentra una prueba para un ***truth-test***, se dirá que el test es ***passing*** (se esperaba que estuviesen relacionados).
- Si el ATP encuentra una prueba para un ***falsity-test***, se dirá que el test es ***nonpassing*** (no se esperaba que estuviesen relacionados).
- Si el ATP no encuentra pruebas para un test se clasifica como ***unknown*** (no se sabe si las conjeturas están relacionadas.)

4 Captura de requisitos

En el diagrama de casos de uso, se pueden ver las acciones disponibles para los usuarios en todo momento. En esta sección se hará un breve resumen de cada uno de ellos. En el anexo se pueden consultar los casos de uso extendidos completamente detallados de las principales funcionalidades.

Además, se describirán las entidades que compondrán las funcionalidades mediante el modelo de dominio.

4.1 Casos de uso

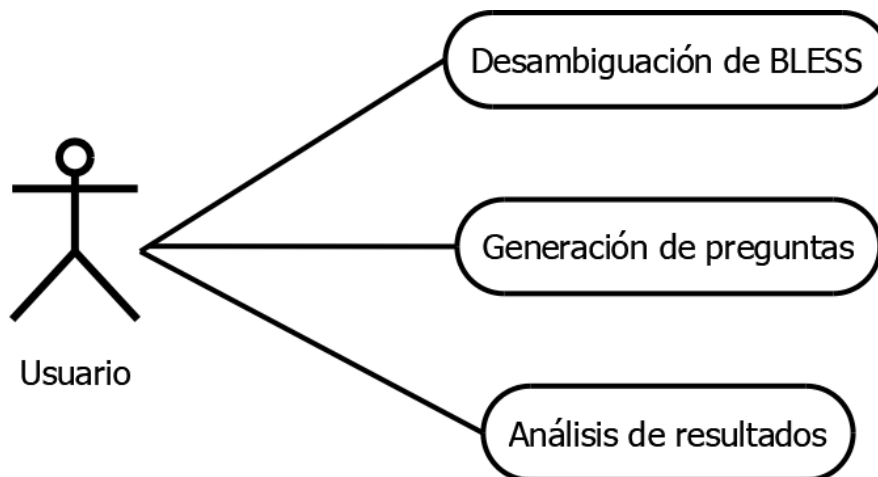


Ilustración 4 Casos de uso

- **Desambiguación de *Bless*:** El usuario puede desambiguar los términos del banco de pruebas *Bless* para utilizarlos posteriormente en la generación de preguntas y análisis de resultados.
Para ello es necesario el fichero con los datos de *Bless* para obtener la desambiguación de cada palabra.
- **Generación de preguntas:** El usuario, mediante los patrones de preguntas definidos puede generar las preguntas que se utilizarán en los análisis.
Para ello es necesario el acceso a todos los recursos (*Bless*, resultado de la desambiguación, *Adimen-SUMO*, *mappings*, patrones definidos). Y una vez ejecutada la aplicación se obtendrán las preguntas y sus negaciones.

- **Análisis de resultados:** El usuario puede obtener los resultados a partir de los elementos anteriores para realizar un análisis. Para ello es necesario disponer de los resultados obtenidos por el razonador automático a partir de las preguntas generadas. Tras la ejecución de la aplicación el usuario obtiene 4 ficheros de texto con los resultados. Cada fichero contiene los resultados a distintos niveles de análisis.

4.2 Modelo de dominio

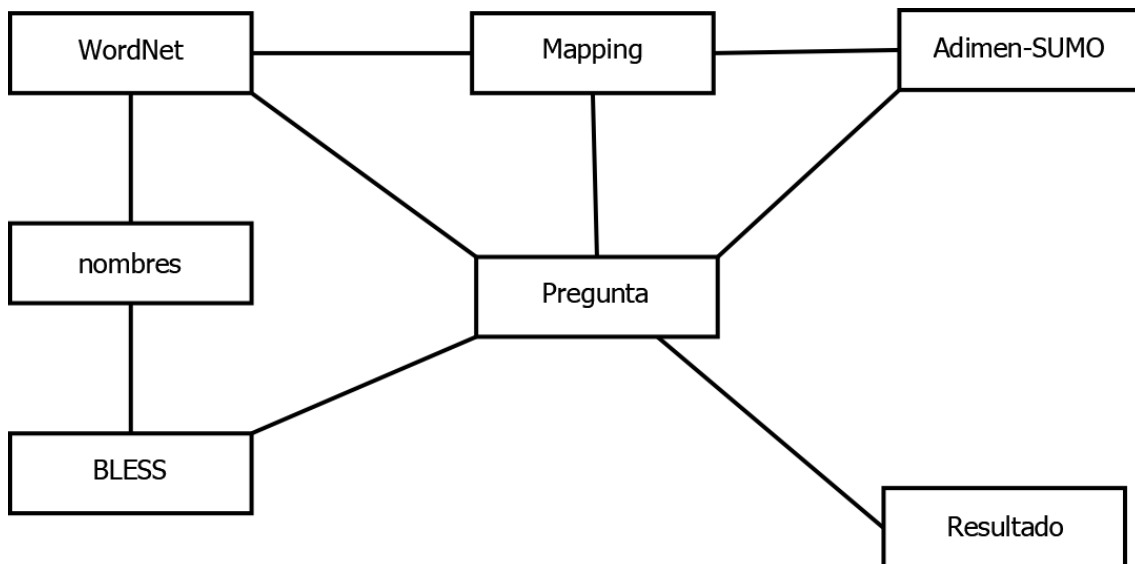


Ilustración 5 Modelo de dominio

- **WordNet:** Entidad que guarda la información de las relaciones de *synsets* de *WordNet*.
- **Adimen-SUMO:** Entidad que guarda la información relacionada con la ontología *Adimen-SUMO*.
- **Mapping:** Entidad que guarda la información del *mapping* de *WordNet* con la ontología *Adimen-SUMO*.
- **BLESS:** Entidad que contiene las palabras que serán utilizadas del banco de pruebas *Bless*.

- **Nombres:** Entidad que hace referencia a la primera palabra de cada par de *Bless* (200 nombres diferentes).
- **Pregunta:** Entidad que guarda las preguntas generadas.
- **Resultado:** Entidad que contiene los resultados de cada test (una pregunta y su negación).

5 Análisis y diseño

5.1 Diagrama de clases

A continuación, se muestra el diagrama de clases que se utilizará en el proyecto. En él aparecen las entidades definidas y se describe brevemente lo que representa cada una.

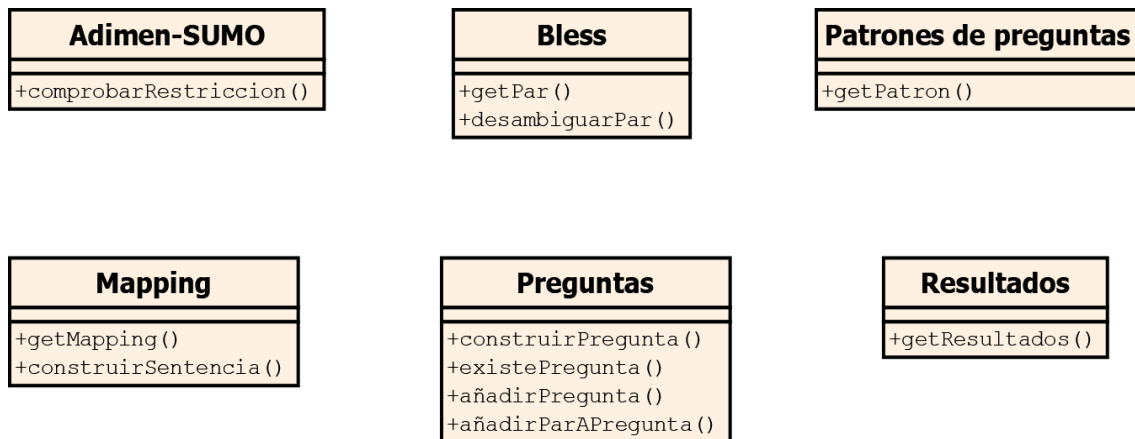


Ilustración 6 Diagrama de clases

- Adimen-SUMO:**
 Entidad que contiene toda la información necesaria para operar con la información de la ontología. Contiene los datos almacenados en un diccionario de *Python* para poder acceder a ellos rápidamente. Es necesaria para comprobar las restricciones de las relaciones que tienen los *synsets* de *Bless* a la hora de generar las preguntas.
- Mapping:**
 Entidad que proporciona el acceso a los *mappings* de *WordNet* a *Adimen-SUMO* a través de un diccionario de *Python*. Es necesaria para construir la sentencia necesaria por cada *synsets* de *Bless* a la hora de generar las preguntas.
- Bless:**
 Entidad que permite acceder, a través de una lista, a los pares de palabras que se quieren evaluar del banco de pruebas *Bless*.

- **Patrones de preguntas:**
Entidad que guarda la estructura de las preguntas que serán generadas. Se almacenarán los patrones en ficheros individuales para cada tipo de pregunta.
- **Preguntas:**
Entidad que contiene todas las preguntas generadas en un diccionario de *Python*. De esta manera se generan los ficheros de preguntas correspondientes, y a la vez se obtiene un acceso rápido para el análisis posterior.
- **Resultados:**
Entidad que conteniendo los resultados obtenidos por el razonador proporciona acceso, mediante un diccionario de *Python*, para realizar el análisis correspondiente.

Usando las clases descritas anteriormente, a continuación, se diseña en detalle el caso de uso "Generación de preguntas" mediante un diagrama de secuencia.

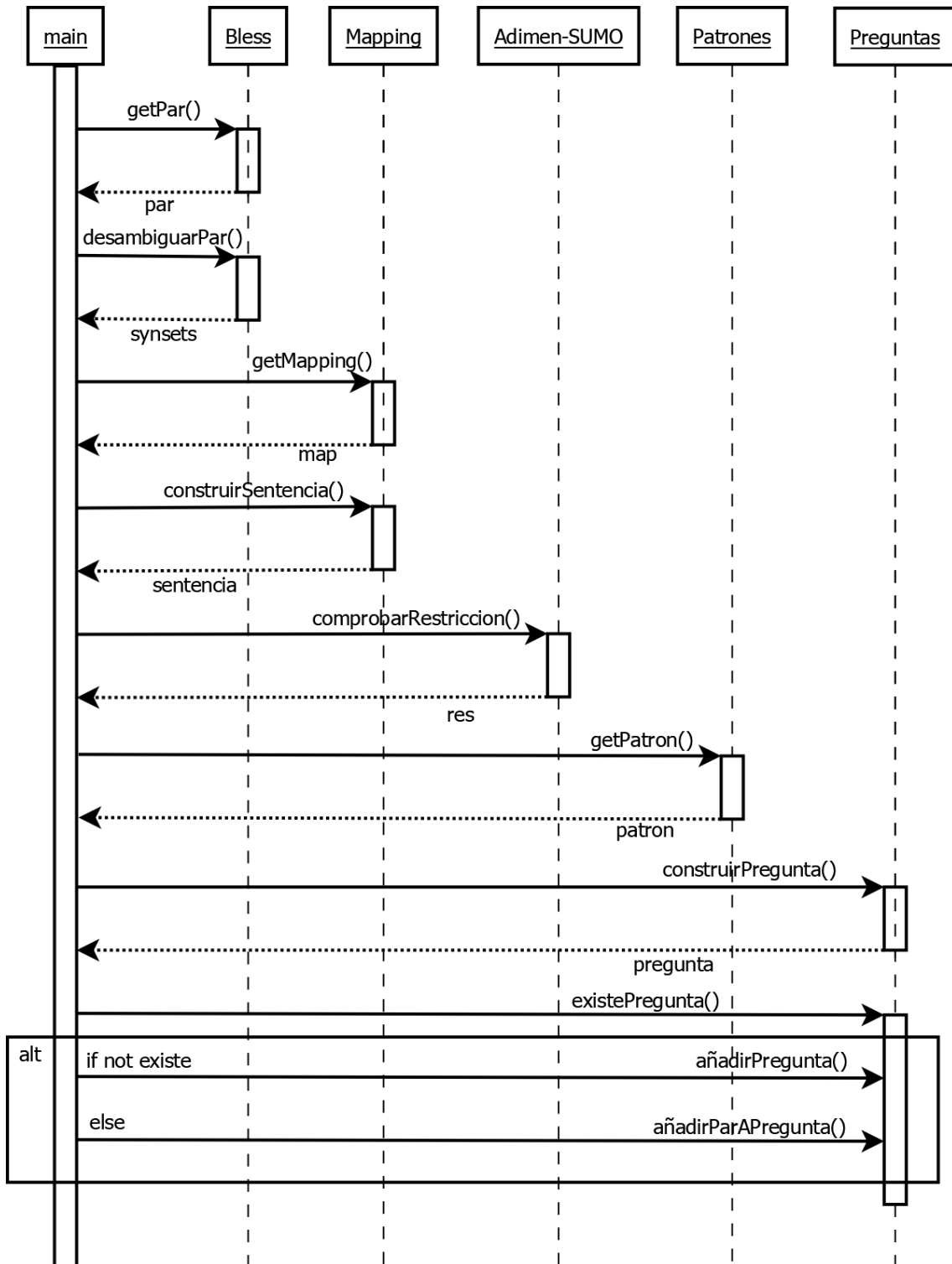


Ilustración 7 Secuencia obtener una pregunta

6 Desarrollo

Para llevar a cabo el desarrollo de este proyecto se ha utilizado el lenguaje de programación *Python* (7) debido a las facilidades que ofrecen alguna de sus estructuras de datos.

6.1 Estructuras de datos

Para almacenar la información que se obtiene a través de los ficheros de texto que se nos proporcionan, se han utilizado las siguientes estructuras de datos.

6.1.1 Listas

Se han utilizado listas de *Python* para almacenar toda la información directamente desde los ficheros proporcionados, y poder, a partir de ellas, extraer la información de una manera más rápida.

Uno de los principales objetivos del uso de listas es facilitar la carga de la información en los diccionarios.

6.1.2 Diccionarios

Como principal estructura de datos de este proyecto se ha utilizado el diccionario de *Python* por su velocidad de acceso a datos concretos. Estos diccionarios nos facilitan la búsqueda de elementos identificándolos con un valor único para cada uno de ellos.

De esta manera, por ejemplo, para la generación de las preguntas que se utilizan en la evaluación de *Bless* se puede acceder en un instante a una pregunta concreta, accediendo a su diccionario correspondiente mediante su identificador.

Esto permite no tener que recorrer una estructura de datos por completo en búsqueda de un elemento. Y de esta manera, reducir en gran medida el tiempo de ejecución del programa.

Listas - <https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

Diccionarios - <https://docs.python.org/3/tutorial/datastructures.html#dictionaries>

6.2 Patrones de preguntas

Para la aplicación que genera las preguntas a partir de los datos de *Bless* es necesario definir los patrones que serán utilizados. En este proyecto serán definidos 18 patrones en total debido al número de relaciones que pueden tener los pares de palabras en *Bless* y en el *mapping* de *WordNet* a *Adimen-SUMO*.

Las relaciones que pueden tener los pares de palabras en *Bless* son las siguientes:

- *attri*: La segunda palabra del par expresa un atributo de la primera.
- *coord*: Ambas palabras pertenecen a la misma clase semántica.
- *event*: La segunda palabra del par expresa una acción que puede ser realizada por la primera.
- *hyper*: La segunda palabra es un hiperónimo de la primera.
- *mero*: la segunda palabra hace referencia a una parte o componente de la primera. Es decir, la primera contiene a la segunda.

Por cada una de estas hay que tener en cuenta las relaciones de *WordNet* de la siguiente manera:

Synset1	Synset2
subsumption o instance	subsumption o instance
subsumption o instance	equivalence
equivalence	subsumption o instance
equivalence	equivalence

Tabla 30 Relaciones 1 mapping *WordNet*

Excepto en la relación “*hyper*” de *Bless* que sólo hay dos combinaciones posibles:

Synset1
subsumption o instance
equivalence

Tabla 31 Relaciones 2 mapping *WordNet*

Por ejemplo, un patrón podría tener la siguiente estructura:

```
(forall (?Y)
  (= >
    <<sentencia2>>
    (exists (?X)
      (and
        <<sentencia1>>
        (exists (?Z)
          (and
            ($instance ?X ?Z)
            ($instance ?Y ?Z)
          )
        )
      )
    )
  )
)
```

6.3 Generación de preguntas

En este apartado se describirá como se generan las preguntas y, además, se completará con la obtención concreta de una pregunta referente a un par de palabras de *Bless*.

El ejemplo que se desarrollará en este apartado surge del siguiente par de palabras:

Palabra1	Dominio	Relación	Palabra2
alligator-n	amphibian_reptile	attri	aggressive-j

Tabla 32 Ejemplo generación de pregunta

En primer lugar, es necesario recoger los pares del banco de pruebas para desambiguarlos. La desambiguación de los términos se realiza mediante *UKBs* utilizando el fichero que contiene toda la información de *Bless*, previamente eliminando aquellos pares de palabras que están relacionados por *random*. Estos términos se eliminan debido a que este tipo de relación no está contemplada en la ontología y por lo tanto no se generará una pregunta para los pares de palabras con dicha relación en *Bless*. De esta manera se reduce el tiempo de ejecución de la aplicación.

Los pares de palabras que permanecen se desambiguan para obtener el *synset* (código de 8 dígitos más una letra) correspondiente y poder buscarlos posteriormente, ya sea en el *mapping* a *WordNet* o *Adimen-SUMO*.

Ejemplo:

alligator-n → 01698434-n
aggressive-j → 00082241-a

Una vez desambiguados se almacenan en un fichero de texto, guardando cada par de palabras identificados por un mismo código. Este fichero es el utilizado para la generación de las preguntas.

Recorriendo este fichero se comprueba el tipo de relación de cada par de palabras en *Bless*, y de acuerdo a ella se genera una pregunta concreta haciendo referencia al *mapping* de cada término en *Adimen-SUMO*.

Ejemplo:

Relación de Bless: attri
Mapping *synset1*: ('Reptile',class,subsumption)
Mapping *synset2*: ('PhychologicalAttribute','Attribute',subsumption)

El proceso continúa con la comprobación de las restricciones que deben cumplir los términos en cuanto a *Adimen-SUMO*.

Ejemplo:

El *mapping2* debe ser del tipo 'Attribute' o 'attribute' y 'PhychologicalAttribute' debe ser sub-clase o instancia de 'Attribute'.

Posteriormente se elige el patrón correspondiente al par de palabras que se está tratando. Para ello, se tiene en cuenta la relación de *Bless* y la relación del *mapping* de cada *synset*.

Ejemplo:

Como la relación de Bless es 'attri' y la de los mappings es 'subsumption' el patrón correspondiente es el siguiente.


```
(exists (?X ?Y)
  (and
    <<sentencia1>>
    <<sentencia2>>
    (equal ?X ?Y)
  )
)
```

Una vez seleccionado el patrón a utilizar y cumpliéndose las restricciones, se crean las sentencias correspondientes a cada mapping. Para ello se asocian a una variable.

Ejemplo:

Sentencia 1, variable ‘?X’:

```
(&instance ?X Reptile)
```

Sentencia 2, variable ‘?Y’:

```
(exists (?W)
  (and
    ($instance ?W PsychologicalAttribute)
    (attribute ?Y ?W)
  )
)
```

El último paso para generar la pregunta final consiste en completar el patrón seleccionado con las sentencias creadas.

Ejemplo:

```
(exists (?X ?Y)
  (and
    ($instance ?X Reptile)
    (exists (?W)
      (and
        ($instance ?W PsychologicalAttribute)
        (attribute ?Y ?W)
      )
    )
    (equal ?X ?Y)
  )
)
```

Por otro lado, para evitar duplicados de una pregunta se almacena toda la información en un diccionario. Se comprueba en cada pregunta si ya existe, y en caso de que ya se haya generado previamente una pregunta igual, se añaden los *synsets* en una lista relacionada con dicha pregunta.

Una vez obtenido el diccionario con todas las preguntas obtenidas sin repeticiones, se genera el fichero resultante. Dicho fichero será empleado por el razonador para obtener el resultado de cada pregunta realizada.

Esta información sobre Bless y las preguntas generadas permanece almacenada en un diccionario de Python. Ya que, posteriormente, será utilizada en el análisis de los resultados que nos proporcione el razonador automático.

7 Análisis experimental de resultados

Con los resultados proporcionados por el razonador sobre cada pregunta se realizarán análisis en diferentes niveles:

- **De pregunta:** Cuantas preguntas han resultado *passing*, cuantas *nonpassing* y cuantas *unknown* de todas las realizadas.
- **De pares:** Cuantos pares (de los cuales se obtienen las preguntas) han resultado *passing*, cuantos *nonpassing* y cuantos *unknown*.
- **De palabra:** Por cada nombre de los 200 que forman el banco de pruebas qué porcentaje de *passing*, *nonpassing* y *unknown* ha obtenido cada uno de ellos. Y de esta manera se utilizarán 2 criterios:

Criterio 1: clasificar como *passing* una palabra si alguno (al menos uno) de sus test es *passing*.

Criterio2: clasificar como *passing* una palabra si al menos 1/3 de sus test es *passing*.

- **De dominio:** De los 17 dominios a los cuales pertenecen los nombres de *Bless*, que porcentaje de *passing*, *nonpassing* y *unknown* ha obtenido cada uno.

7.1 A nivel de pregunta

El primer análisis que se ha realizado ha sido comprobar los resultados de cada pregunta. Así, podemos ver cómo de efectivas son las preguntas generadas a partir de *Bless* para evaluar la ontología *Adimen-SUMO*.

	Cantidad	Porcentaje
Passing	1179	18,122 %
nonpassing	1	0,015 %
unknown	5326	81,863 %
total	6506	100 %

Tabla 33 Resultados a nivel de pregunta

Se puede observar que una única pregunta ha sido rechazada y que la mayor parte no se ha podido responder. Por lo que podemos deducir que el banco de pruebas no es el ideal para evaluar *Adimen-SUMO*. O bien que los recursos asignados al razonador no son suficientes para encontrar la demostración.

7.2 A nivel de pares

El segundo análisis realizado buscaba comprobar los resultados de cada pregunta, pero en este caso observando los pares de palabras del banco de pruebas que dan lugar a dicha pregunta.

	Cantidad	Porcentaje
passing	3156	22,958 %
nonpassing	1	0,007 %
unknown	10590	77,035 %
total	13747	100 %

Tabla 34 Resultados a nivel de pares

Aquí podemos ver como la proporción en cuanto a los resultados se mantiene similar al análisis anterior. Aunque subiendo levemente el resultado *passing*.

7.3 A nivel de palabra

El tercer análisis busca comprobar los resultados de cada nombre de *Bless*. Para ello se han analizado los *synsets* resultantes de la desambiguación de cada nombre. Como los nombres tienen diferente contexto para cada par relacionado en *Bless*, la desambiguación no es necesariamente la misma para una palabra concreta. Por ello se han obtenido 222 *synsets* en lugar de 200, que son los nombres de *Bless*.

Resultados según el criterio 1:

	Cantidad	Porcentaje
passing	208	93,697 %
nonpassing	0	0,000 %
unknown	14	6,303 %
total	222	100 %

Tabla 35 Resultados a nivel de palabra criterio 1

Resultados según el criterio 2:

	Cantidad	Porcentaje
passing	47	21,172 %
nonpassing	0	0,000 %
unknown	175	78,828 %
total	222	100 %

Tabla 36 Resultados a nivel de palabra criterio 2

7.4 A nivel de dominio

El último análisis realizado busca comprobar los resultados obtenidos a nivel de dominio de los nombres del banco de pruebas. Es decir, por cada par de palabras de *Bless* se especifica cual es el dominio de la primera del par (el nombre). Así, podemos ver de una manera más precisa los resultados obtenidos por cada grupo de los nombres que forman *Bless*.

Dominio	passing	nonpassing	unknown	Total
amphibian_reptile	21	0	236	257
	8,171 %	0 %	91,829 %	100 %
clothing	324	0	794	1118
	28,980 %	0 %	71,020 %	100 %
appliance	77	0	607	684
	11,257 %	0 %	88,743 %	100 %
furniture	74	0	426	500
	14,800 %	0 %	85,200 %	100 %
tool	343	0	718	1061
	32,328 %	0 %	67,672 %	100 %
vehicle	380	0	1732	2112
	17,992 %	0 %	82,008 %	100 %
bird	303	0	519	822
	36,861 %	0 %	63,139 %	100 %
building	131	0	767	898
	14,588 %	0 %	85,412 %	100 %
container	84	0	375	459
	18,300 %	0 %	81,7 %	100 %
fruit	160	0	460	620
	25,806 %	0 %	74,194 %	100 %
vegetable	131	0	510	641
	20,437 %	0 %	79,563 %	100 %
tree	94	0	241	335
	28,059 %	0 %	71,941 %	100 %
ground_mammal	469	1	1220	1690
	27,751 %	0,059 %	72,190 %	100 %
water_animal	96	0	477	573
	16,754 %	0 %	83,246 %	100 %
insect	104	0	261	365
	28,493 %	0 %	71,507 %	100 %
musical_instrument	173	0	365	538
	32,156 %	0 %	67,844 %	100 %
weapon	192	0	882	1074
	17,877 %	0 %	82,123 %	100 %

Tabla 37 Resultados a nivel de dominio

Una vez más se puede ver en la tabla anterior como las proporciones de cada resultado son similares a los análisis previamente expuestos. Una gran parte de cada dominio no se ha podido responder y el resto ha sido respondida satisfactoriamente.

7.5 Análisis de resultados: conclusiones

Que no haya apenas (solo 1) test *nonpassing* (probablemente debido a un problema de los recursos asignados) reduce la posibilidad de analizar exhaustivamente los resultados.

Según resultados experimentales anteriores, se deberían haber resuelto (clasificar como *passing* o *nonpassing*) alrededor de un 50% de los *tests*. Y como se puede comprobar, los resultados obtenidos rondan el 20%.

Sin embargo, los resultados obtenidos en ciertos dominios mejoran sensiblemente el resultado promedio. Por ejemplo, los dominios *clothing*, *tool*, *bird* o *musical_instrument* superan el 28%.

Esto nos lleva a pensar que el conocimiento de la ontología acerca de estos dominios semánticos es claramente mejor que el resto.

Además, si analizamos los resultados a nivel de palabra, el número de palabras validadas según alguna de sus relaciones es relativamente alto. En este caso, la conclusión que se puede extraer es que la ontología proporciona información semántica acerca de la mayoría de palabras, pero que lo hace parcialmente y sin reflejar todas las relaciones semánticas que se cumplen.

8 Pruebas unitarias

Se ha sometido al proyecto a una serie de pruebas con el fin de encontrar y corregir fallos existentes en las aplicaciones. De esta manera se busca conseguir el correcto funcionamiento ante el uso que pueda realizar un usuario.

A continuación, se recogen algunas de las principales pruebas de las funcionalidades clave:

8.1 Generación de preguntas

Descripción	Resultado esperado	Resultado obtenido	Acciones realizadas
Generar el contexto para desambiguar mediante <i>UKBs</i> con un fichero con el formato de <i>Bless</i>	Se genera el fichero context.txt correctamente	Se genera el fichero context.txt correctamente	Ejecutar el script <i>bleess2ukb.sh</i>
Generar el contexto para desambiguar mediante <i>UKBs</i> con un fichero sin el formato de <i>Bless</i>	No se genera el fichero context.txt	La ejecución falla y no se obtiene el fichero	Ejecutar el script <i>bleess2ukb.sh</i>
Ejecutar la desambiguación mediante <i>UKBs</i> con el fichero context.txt correcto	Se genera el fichero desambiguado.txt correctamente	Se genera el fichero desambiguado.txt correctamente	Ejecutar la aplicación de <i>UKB</i>
Ejecutar la desambiguación mediante <i>UKBs</i> con el fichero context.txt incorrecto	No se genera el fichero desambiguado.txt	La ejecución falla y no se obtiene el fichero	Ejecutar la aplicación de <i>UKB</i>
Ejecutar la generación de preguntas con los ficheros de patrones correctamente indicados	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se indican las rutas correctamente en el código

Ejecutar la generación de preguntas con el fichero de la ontología correctamente indicado	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se indican las rutas correctamente en el código
Ejecutar la generación de preguntas con los ficheros de los mappings correctamente indicados	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se generan los ficheros preg.txt y negatedPreg.txt correctamente	Se indican las rutas correctamente en el código
Comprobar que se genera el diccionario de preguntas correctamente	Se añaden los pares de palabras correspondientes por cada pregunta	Se añaden los pares de palabras correspondientes por cada pregunta	Comprobar la estructura de datos
Comprobar que no se duplican preguntas	No existen preguntas repetidas en el fichero generado	No existen preguntas repetidas en el fichero generado	Revisar el fichero de preguntas
Comprobar que las preguntas tienen el formato indicado en los patrones	Se generan las preguntas con la estructura definida en los patrones	Se generan las preguntas con la estructura definida en los patrones	Revisar el fichero de preguntas
Comprobar que las sentencias para un tipo de <i>mapping</i> se crean correctamente	La sentencia creada es correcta de acuerdo al <i>mapping</i>	La sentencia creada es correcta de acuerdo al <i>mapping</i>	Comprobar la creación de la sentencia para ciertos mapeos
Comprobar que las preguntas generadas son las esperadas.	La pregunta contiene las sentencias y el patrón correspondiente a cada par de preguntas	La pregunta contiene las sentencias y el patrón correspondiente a cada par de preguntas	Analizar un par de palabras concreto

Tabla 38 Pruebas unitarias

9 Conclusiones

9.1 Seguimiento

En la planificación del proyecto, se calcularon las horas de esfuerzo que supondría cada proceso. Pero lo cierto es que el resultado final ha variado respecto al cálculo inicial.

A continuación, se recogen las horas reales dedicadas a cada tarea del proyecto, y se compararán con las iniciales planteadas en el Documento de Objetivos de Proyecto.

Proceso	Inicial	Real
Inicio y aprendizaje		
Aprendizaje de <i>Wordnet</i> y <i>Adimen-SUMO</i>	16	12
Aprendizaje <i>Bless</i> y desambiguación	16	16
Instalación de herramientas y conexión al servidor	2	1
Redacción del DOP	16	16
Captura de requisitos		
Identificación de casos de uso	8	6
Análisis y diseño		
Diseño de casos de uso	6	6
Diseño de modelo de dominio	6	8
Diseño de diagramas de secuencia	6	8
Desarrollo		
Lectura del banco de pruebas <i>Bless</i> y desambiguación mediante <i>UKBs</i>	8	12
Lectura de las relaciones de <i>Wordnet</i>	3	4
Lectura de la ontología <i>Adimen-SUMO</i>	3	4
Lectura de <i>Mappings</i>	3	4
Definición de patrones de preguntas	30	35
Generación de preguntas	30	40
Análisis de soluciones	20	20
Análisis de los <i>Tests</i>	20	20
Pruebas unitarias	15	15
Documentación		
Memoria	75	80
Cierre		
Validación y entrega	2	2
Exposición del proyecto	6	6
Total	291	315

Tabla 39 Resultado seguimiento

A causa de estas variaciones en los esfuerzos es necesario recalcular la evaluación económica inicial planteada en el DOP, debido a que la inversión de horas en el proyecto es diferente.

- **Salario:**

Horas totales: 315 h

Coste hora: 32 €/h

Total coste (€) = 315 h * 32 €/h = 10.080 €

- **Amortización del equipo:**

Precio: 750 €

Unidades: 1.

Tiempo estimado para su total amortización: 6 años (72 meses).

Horas jornada: 4 h/día de Lunes a Viernes.

Tiempo de uso en el proyecto: $\frac{315 h}{4 h/día} \times \frac{1}{5 días/sem} \times \frac{1}{4 sem/mes} = 3,94$ meses.

Amortización equipo = (Coste/Duración estimada)*Tiempo de uso*Unidades
 $(750/72) * 3,94 * 1 = 41,04$ €

- **Servidor:**

Inicialmente se estimó que las preguntas que se obtendrían de *Bless* serían el 20% del número de pares de palabras que lo componen. Pero finalmente este valor es superior, concretamente 6506 preguntas.

Tiempo de ejecución:

$$6506 \times 2 \times 5 \text{ min/pregunta} = 65060 \text{ minutos} = 1084.33 \text{ horas}$$

Coste de ejecución: $1084,33 \text{ horas} \times 0.27 \text{ €/hora} = 292.77$ €

El coste de ejecución se ha calculado teniendo en cuenta el coste por hora de *Microsoft Azure Cloud Services*.

- **Total:**

Concepto	Inicial	Final
Salario	9.312 €	10.080 €
Amortización del equipo	37,89 €	41.04 €
Alquiler del lugar de trabajo	0 €	0 €
Software	0 €	0 €
Servidor	238,95 €	292,77 €
Subtotal	9.588,84 €	10.413,81 €
Gastos comunes	526,03 €	496,59 €
TOTAL	10.068,28 €	10.934,50 €

Tabla 40 Total recalculado coste del proyecto

9.2 Reflexión personal

Tras completar todos los créditos necesarios y encontrarme realizando ya prácticas en empresa, llegaba la hora escoger y comenzar el TFG.

La elección de este proyecto surgió tras acudir a varias reuniones con diferentes tutores para conocer sus propuestas. Cuando escuché por primera vez que se trataba de realizar un trabajo con ontologías he de admitir que no me convenció por completo. Pero conforme el tutor me explicaba de que se trataba concretamente, el tema fue interesándome cada vez más.

Además, este proyecto ofrecía total libertad a la hora de llevarlo a cabo. Es decir, no estaba ligado a un lenguaje de programación, ni a herramientas específicas para completarlo. Esto me llamó la atención, ya que meses atrás realicé alguna tarea con el lenguaje *Python* y ciertamente me apetecía hacer algo más completo.

Al principio, como desconocía el tema, invertía el tiempo en comprender de que se trataba y que era lo que yo realmente iba a realizar. Tenía la sensación de que no avanzaba y eso no me gustaba del todo. Sentía que no avanzaba nada. Pero poco a poco se me aclaraban las ideas, y las tareas que han llegado después las he completado cada vez con mayor rapidez.

Este proyecto también me ha servido para comprobar de primera mano lo que los profesores siempre nos dicen sobre la planificación previa a realizar el trabajo. Y es que es realmente importante dividir los tiempos que se van a emplear para cada objetivo, y planear las tareas antes de realizarlas. De esta manera se controlan mejor las entregas y el avance del proyecto. En particular, durante las últimas semanas he notado mucha más presión. Se acercaba la fecha de entrega y veía el fin más lejos de lo que pensaba. Pero aplicando el tiempo necesario se han conseguido los objetivos.

Por otro lado, la decisión de realizar el trabajo en el lenguaje *Python* me ha servido para iniciarme en él. Ahora comprendo mejor su funcionamiento y las posibilidades que ofrecen sus estructuras de datos.

Siendo crítico con el trabajo realizado, por supuesto pienso que la parte de programación es mejorable. Es decir, el código desarrollado podría ser más claro y preciso. Pero principalmente por falta de tiempo cuando conseguía un objetivo pasaba al siguiente para obtenerlos todos cuanto antes.

Por último, uno de los fines de mi TFG era que posteriormente se pudiera utilizar para realizar la evaluación con diferentes patrones de preguntas. Espero que esto facilite el trabajo para aquellos que decidan continuar con la evaluación.

9.3 Líneas futuras

El objetivo principal de este proyecto es evaluar la ontología *Adimen-SUMO* utilizando el banco de pruebas *Bless*. Pero una funcionalidad que se le puede atribuir a este proyecto es poder evaluar la ontología con diferentes bancos de pruebas. Por ello, opino que para continuar con el proyecto se podría realizar la evaluación aportando diferentes recursos.

Otra posibilidad es poder definir los patrones de preguntas que genera la aplicación. De esta manera se podría realizar la evaluación definiendo patrones diferentes para analizar los diferentes resultados.

Por último, se puede continuar con la evaluación de los resultados obtenidos por el razonador. Por ejemplo, analizando la cobertura en la ontología de los *tests* utilizando las demostraciones proporcionadas por el razonador.

10 Bibliografía

1. **J. Álvarez, P. Lucio y G. Rigau.** Adimen-SUMO: Reengineering an ontology for first-order reasoning. 8(4) (págs. 80-116). s.l. : Int. J. Semantic Web Inf. Syst., 2012, págs. 1 -- 2.
2. **C. Fellbaum.** *WordNet: An Electronic Lexical Database.* s.l. : MIT Press, 1998.
3. **M. Baroni y A. Lenci.** How we BLESSED distributional semantic evaluation. *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics.* s.l. : Association for Computational Linguistics, 2011, págs. 1-10.
4. **J. Álvarez, P. Lucio y G. Rigau.** Improving the Competency of First-Order Ontologies. *Proc. of the 8 th Int. Conf. on Knowledge Capture (K-CAP 2015).* s.l. : ACM, 2015, págs. 15:1-15:8.
5. **Eneko Agirre y Aitor Soroa.** *Personalizing PageRank for Word Sense Disambiguation.* s.l. : EACL, 2009. págs. 33-41.
6. **I. Niles y A. Pease.** Towards a standard upper ontology. [aut. libro] Guarino N. et al. *Proc. of the 2 nd Int. Conf. on Formal Ontology in Information Systems (FOIS 2001).* s.l. : ACM, 2001, págs. 2-9.
7. **I. Niles y A. Pease.** Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. [aut. libro] H. R. Arabnia. *Proc. of the IEEE Int. Conf. on Inf. and Knowledge Engin.* s.l. : CSREA Press, 2003, págs. 412-416.
8. **L. Kovács y A. Voronkov.** First-order theorem proving and Vampire. [aut. libro] N. Sharygina y H. Veith. *Computer Aided Verification.* s.l. : Springer, 2013, págs. 1-35.
9. **Guttag, John.** *Introduction to computation and programming using Python.* s.l. : MIT Press, 2013.

11 Casos de uso extendidos

La finalidad de este apartado es ofrecer más detalles sobre las funciones que se realizan en este proyecto. De esta manera el usuario podrá conocer cuáles son los requerimientos de una tarea, que se realiza en cada una de ellas y que se obtiene tras su ejecución.

A continuación, se indicará una descripción, los actores, las precondiciones de la tarea, sus requisitos no funcionales, el flujo de eventos que realiza y las Postcondiciones.

11.1 Desambiguación de Bless

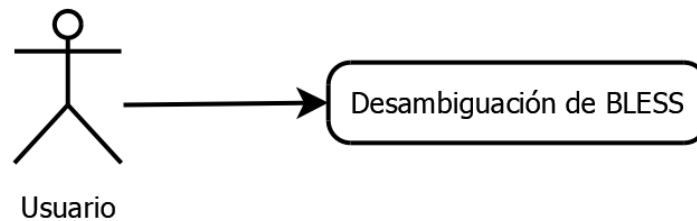


Ilustración 8 Caso de uso Desambiguación de Bless

Descripción	Desambiguación de los términos de <i>Bless</i> para poder generar las preguntas necesarias.
Actores	Usuario.
Entradas	Se han tratado los datos de <i>Bless</i> para obtener los contextos que necesita UKB para desambiguar.
Requisitos no funcionales	Ninguno.
Flujo de eventos	1.- El usuario introduce el comando necesario para la desambiguación de los términos mediante UKB. 2.- La aplicación recorre los contextos proporcionados para desambiguar los términos.
Salidas	Se obtiene un fichero <i>txt</i> con la desambiguación de cada palabra por contexto.

Tabla 41 Caso de uso extendido Desambiguación de Bless

11.2 Generación de preguntas

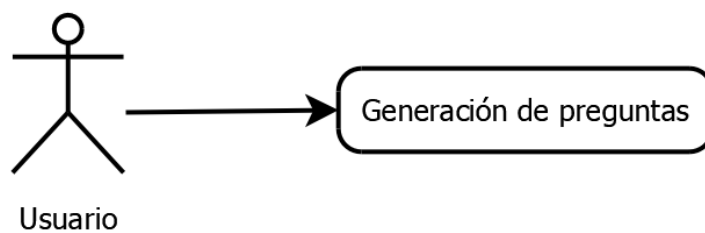


Ilustración 9 Caso de uso Generación de preguntas

Descripción	Generación de las preguntas necesarias para el razonador automático.
Actores	Usuario.
Entradas	Tener los patrones de preguntas definidos, los datos de <i>Bless</i> , los resultados de la desambiguación y acceso a los recursos (<i>Adimen-SUMO</i> , <i>mapping</i> de <i>WordNet</i>).
Requisitos no funcionales	Ninguno.
Flujo de eventos	<ol style="list-style-type: none"> 1.- El usuario ejecuta la aplicación para generar las preguntas y su negación. 2.- Se carga la información desde los ficheros necesarios (entradas). 3.- Se generan las preguntas a partir de los patrones definidos.
Salidas	Se obtienen dos ficheros <i>txt</i> , uno con las preguntas y otro con la negación de cada una de ellas.

Tabla 42 Caso de uso extendido Generación de preguntas

11.3 Análisis de resultados

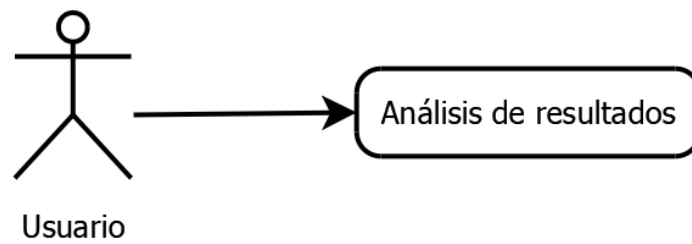


Ilustración 10 Caso de uso Análisis de resultados

Descripción	Obtención del análisis de los resultados del razonador.
Actores	Usuario.
Entradas	Resultados del razonador y preguntas generadas.
Requisitos no funcionales	Ninguno.
Flujo de eventos	<ol style="list-style-type: none"> 1.- El usuario ejecuta la aplicación para realizar el análisis de los resultados. 2.- Se carga la información mediante los ficheros de entrada. 3.- Se calculan los resultados y se generan 4 ficheros.
Salidas	Se obtienen 4 ficheros <i>txt</i> , uno por cada nivel de análisis realizado.

Tabla 43 Caso de uso extendido Análisis de resultados

12 Diagramas de secuencia

En este apartado se mostrarán los diagramas de secuencia de la aplicación desarrollada. Describiendo cómo se tratan los datos de *Bless*, cómo se obtienen las preguntas, y cómo se realizan los análisis de resultados.

12.1 `getBLESSsinRandom.py`

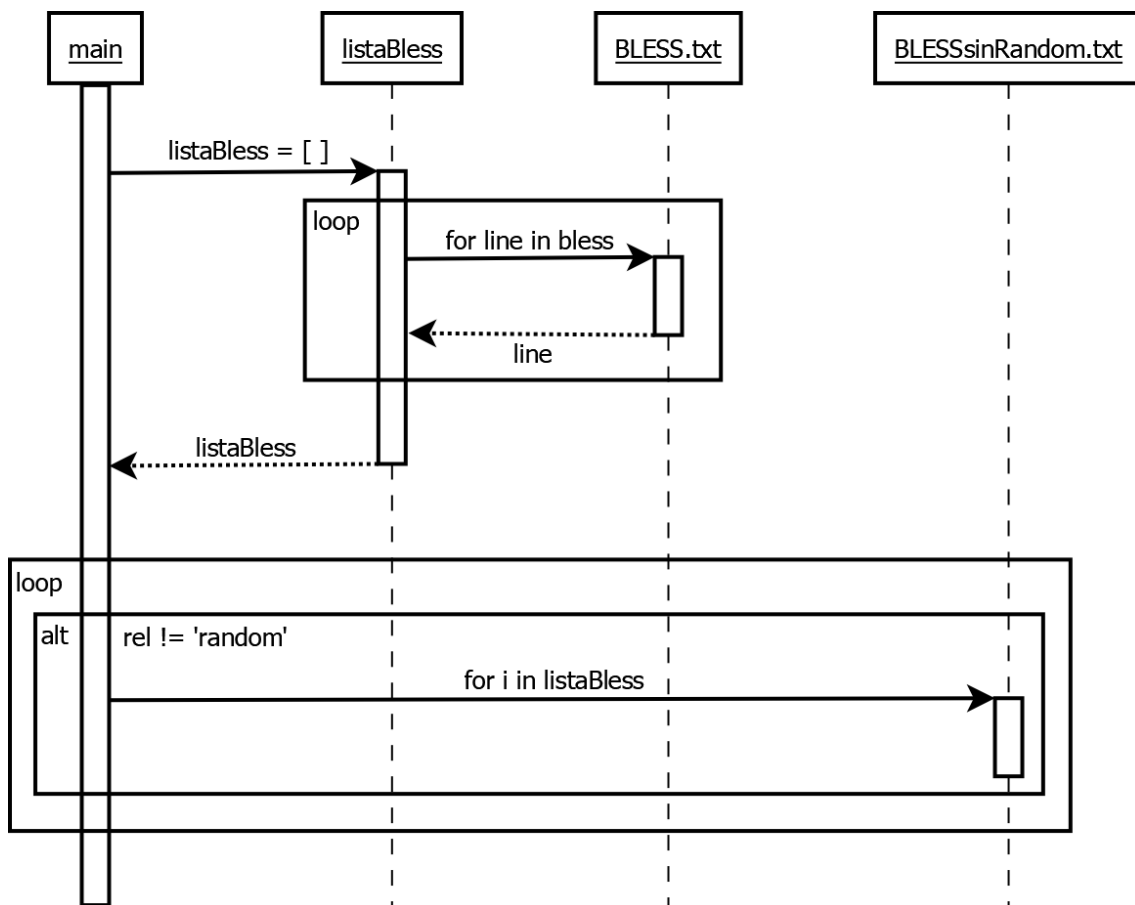


Ilustración 11 Secuencia `getBLESSsinRandom.py`

12.2 bless2ukb.sh

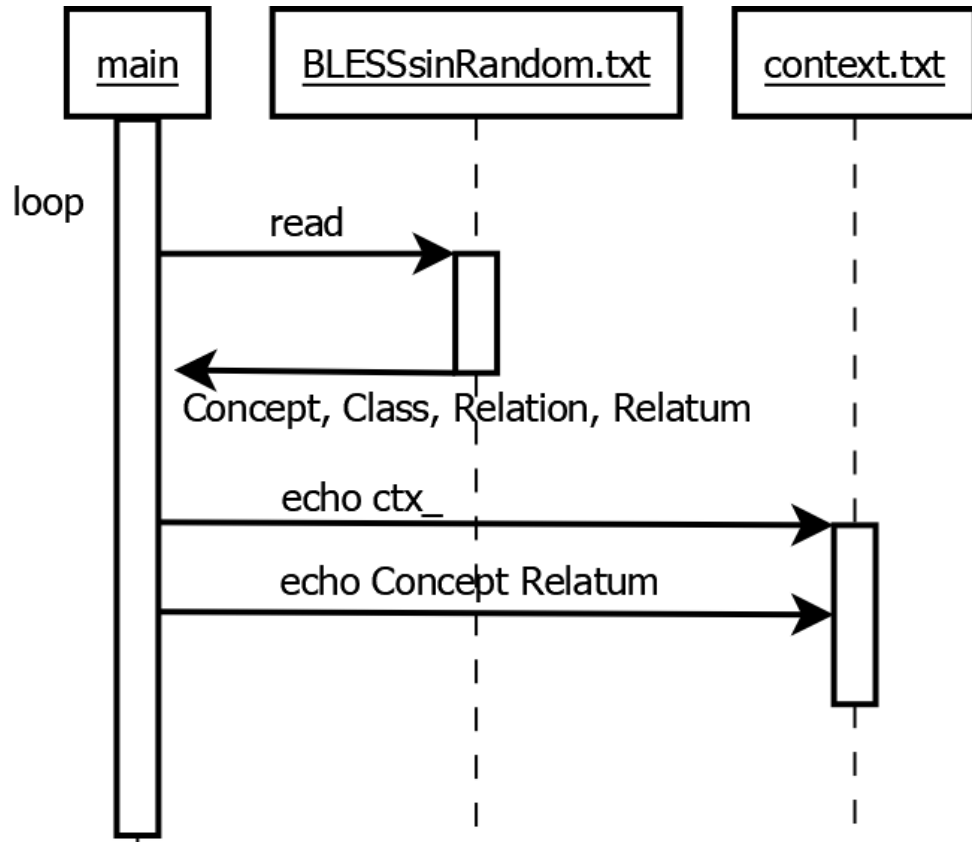


Ilustración 12 Secuencia `bless2ukb`

12.3 getPreguntas.py

Se dividirá el diagrama en 4 diagramas para tener mayor claridad.

12.3.1 Carga de *Bless* y sus términos desambiguados

En este primer diagrama se muestra la carga de las listas con los datos de *Bless* y sus términos desambiguados a partir de los ficheros de texto.

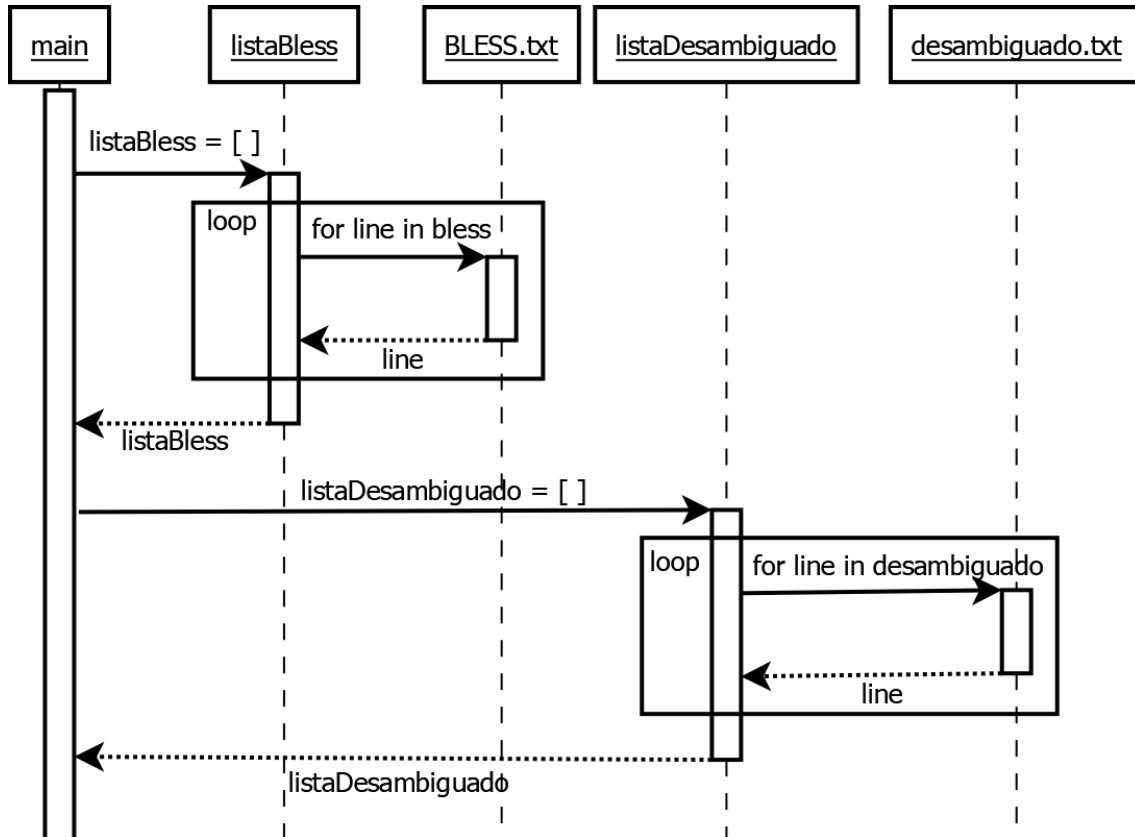


Ilustración 13 Secuencia carga de Bless y desambiguado

12.3.2 Carga de los mappings de WordNet y la ontología Adimen-SUMO

En esta segunda parte se muestra la carga de las listas que contienen la información de los mappings utilizados y de la ontología.

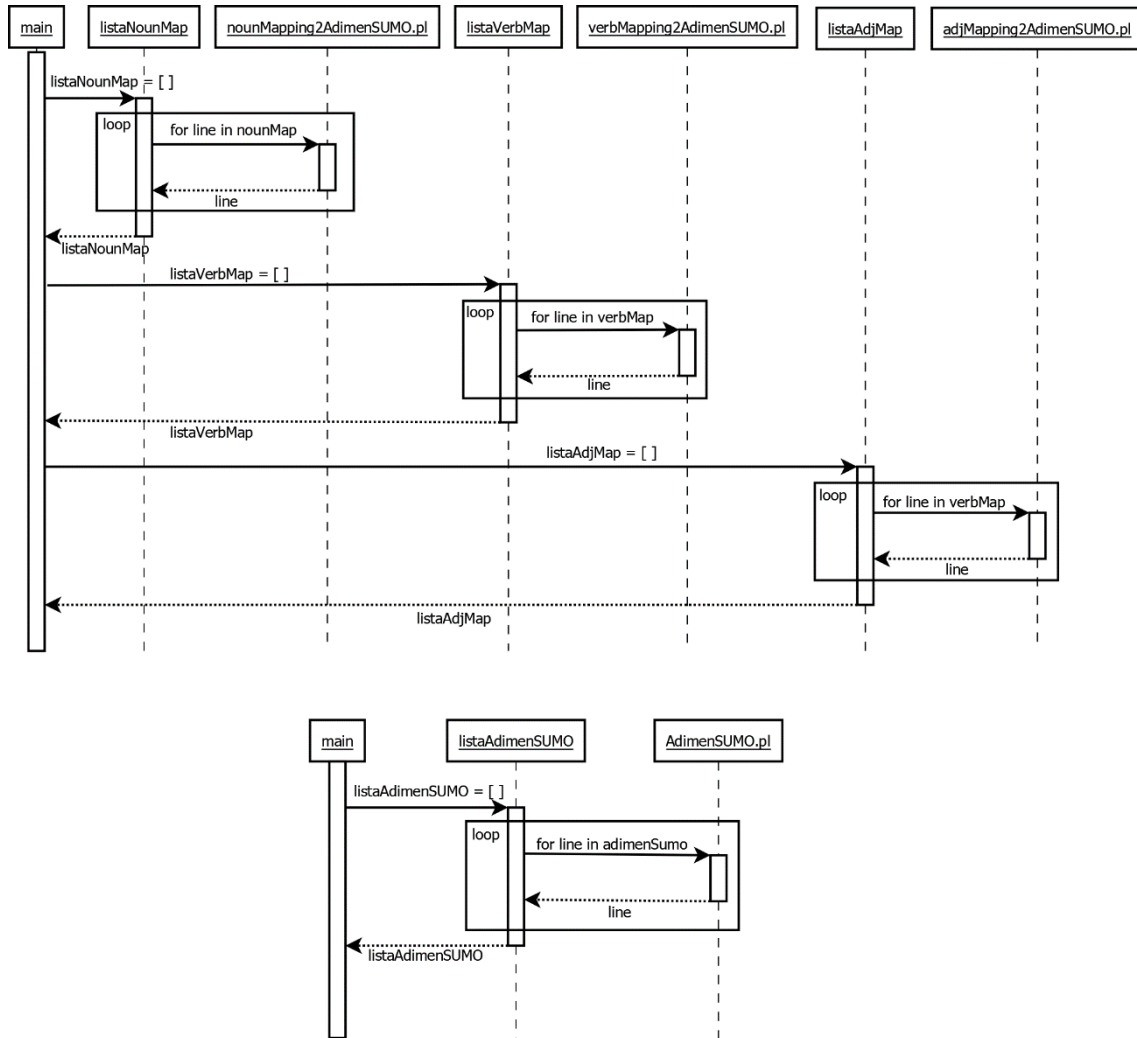


Ilustración 14 Secuencia carga mapping y ontología

12.3.3 Carga de los diccionarios de mappings y ontología

En esta tercera parte se muestra la carga de los diccionarios que serán utilizados a partir de las listas previamente cargadas.

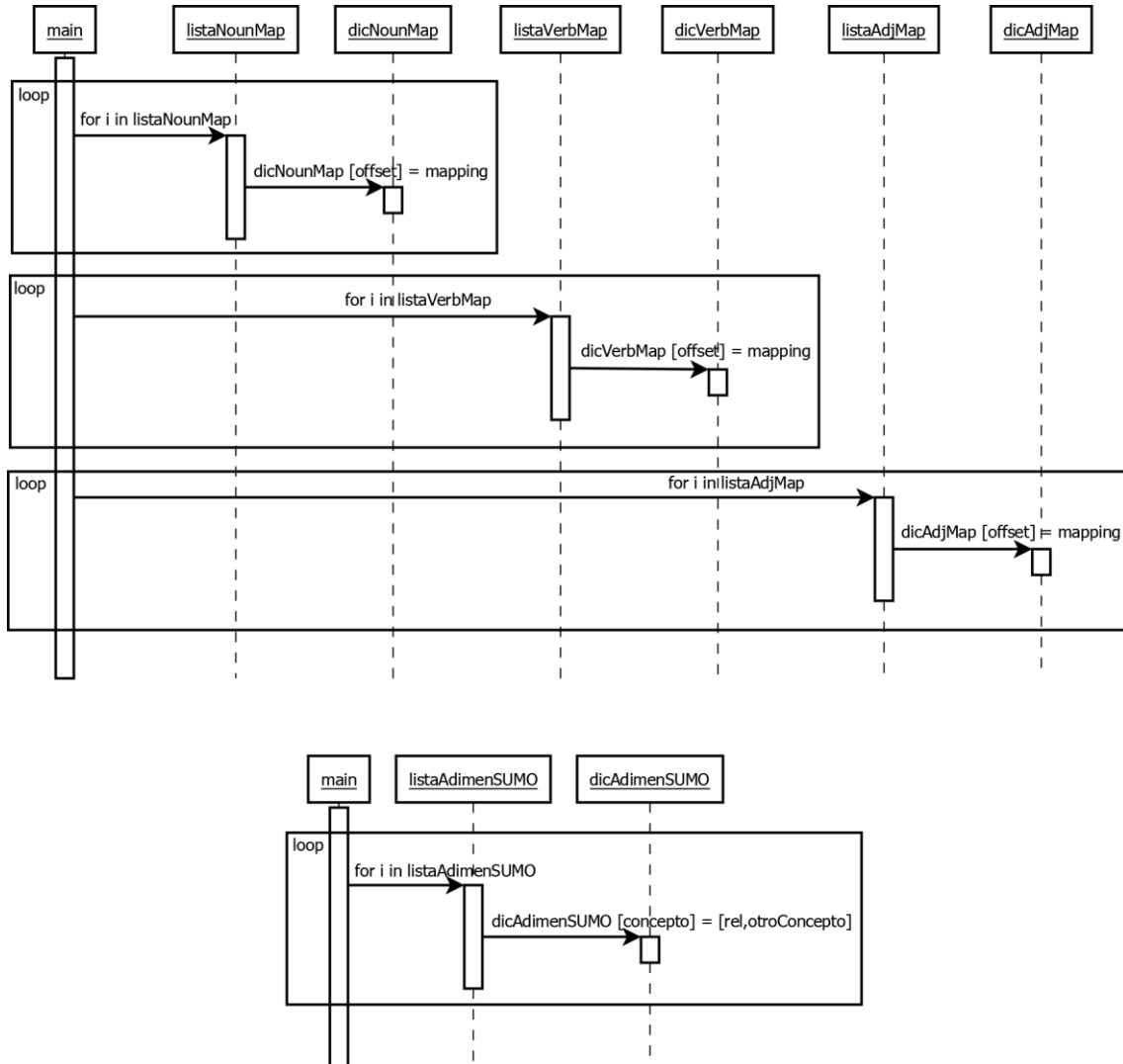


Ilustración 15 Secuencia carga de diccionarios

12.3.4 Creación de preguntas y carga en el diccionario

En esta cuarta parte se muestra la creación de una pregunta correspondiente a un patrón concreto y la carga en el diccionario de preguntas comprobando si está duplicada o no.

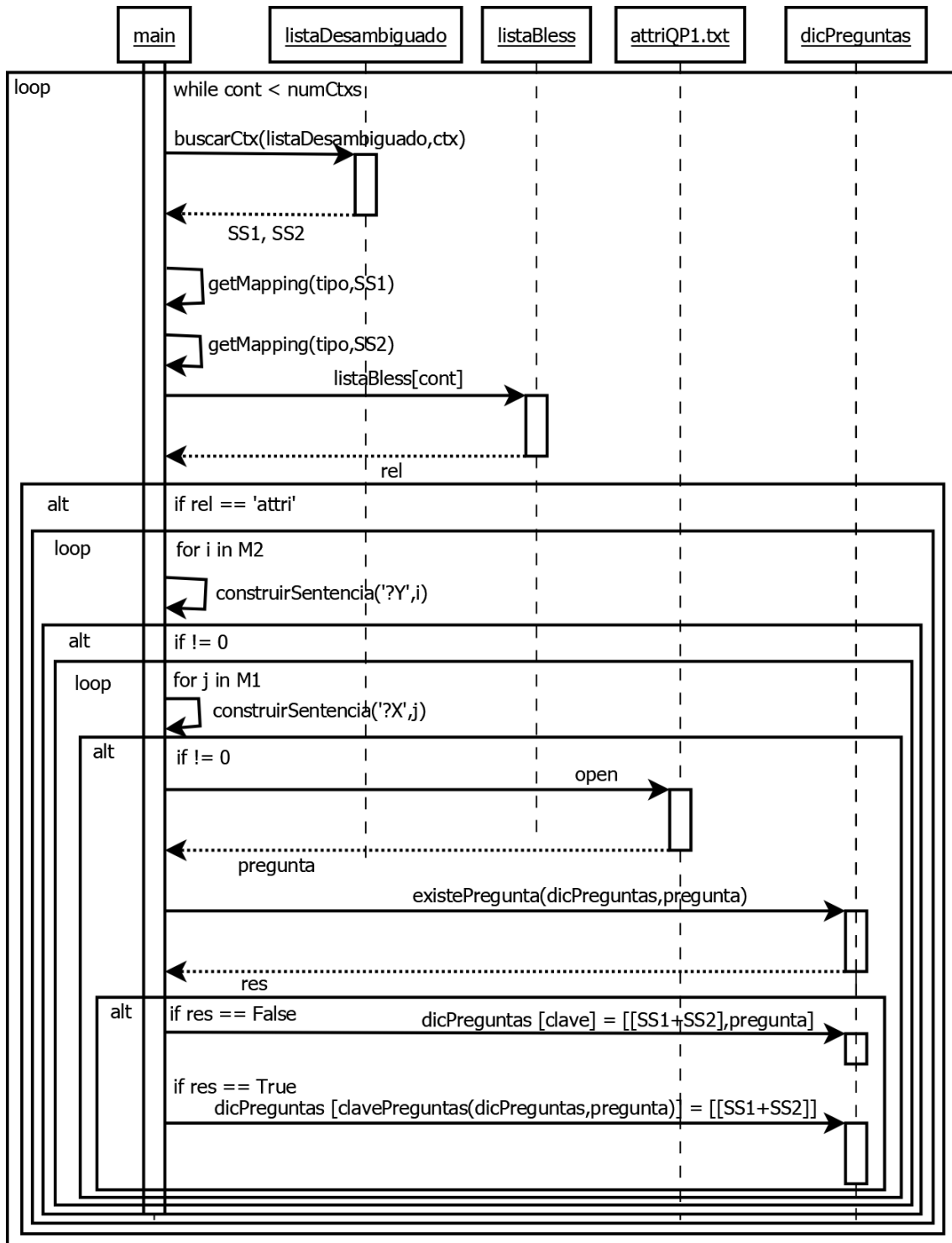


Ilustración 16 Secuencia creación y carga de preguntas