UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería del Software

Trabajo Fin de Grado

WordDomain 2.0. Herramienta semiautomática para la generación de diccionarios

UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería del Software

Trabajo Fin de Grado

WordDomain 2.0. Herramienta semiautomática para la generación de diccionarios

Autor: Isaac Sosa García

Tutor: Encarna Sosa Sánchez

ÍNDICE GENERAL DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En este trabajo se han aumentado las capacidades que tiene WordDomain para generar un diccionario semántico relacionado con un dominio determinado. Se ha dotado a la aplicación con la capacidad de generar un diccionario mucho más amplio al aumentar el número de ontologías usadas en la búsqueda de términos relacionados. Además, se permite que el usuario pueda añadir manualmente dos ontologías de su entera elección para obtener un mayor número de términos relacionados con dominios específicos seleccionados por el usuario.

Cabe destacar también el trabajo de refactorización realizado en WordDomain en aras de obtener un código más legible y modular que pueda facilitar el mantenimiento y ampliación de la aplicación en el futuro.

Además de las mejoras realizadas a WordDomain, se ha implementado la herramienta BPMN Comparator, esta herramienta permite realizar una comparativa semántica de modelos de procesos de negocio representados en notación BPMN según un dominio dado.

CAPÍTULO 1: INTRODUCCIÓN

A pesar de que vivimos en la era de la información con acceso a incontables cantidades de datos sobre cualquier tema que podamos llegar a imaginar, el acercamiento y tratamiento de este mar de información, desde el punto de vista de herramientas software, en algunas ocasiones se ha basado en un análisis sintáctico de los datos, dejando algo de lado al análisis semántico, es decir su significado.

La semántica se define como el campo que estudia el significado de palabras y expresiones, su finalidad es descomponer este significado en unidades más pequeñas (semas) que nos permitan diferenciar palabras de significado similar u opuesto.

En ciertas áreas de la Ingeniería del Software es de gran importancia poder realizar una comparativa semántica de información para establecer una base semántica común y posibilitar la comparación con un dominio concreto.

En el ámbito de Business Process Management (BPM)

El uso de una base semántica común en el ámbito de BPM es de gran ayuda cuando se plantean situaciones como las enumeradas a continuación:

- Mejora de la descripción de los BP de empresas, al aportar una base semántica común para ayudar a definir y describir los procesos de una forma más eficiente.
- Representar los BP de una forma más unificada y menos ambigua. Obteniendo un conjunto de sinónimos y usando los términos más representativos para cada proceso podemos promover la utilización de un lenguaje común para los BP.
- Rediseño, modernización y optimización de los BP de la empresa. Por ejemplo, en el trabajo Sosa2020 [1] se utiliza un diccionario creado ad hoc en la modernización de una aplicación web legada. A la vez que se lleva a cabo la modernización de dicha aplicación se descubren nuevas funcionalidades y se optimizan los BP.

En el ámbito de Procesamiento de Lenguaje Natural

En el ámbito de Procesamiento de Lenguaje Natural (PLN) es necesario utilizar una base léxica común que nos servirá de base tanto a la hora de extraer información

desde lenguaje natural, exportar información a documentos o realizar comparativas entre ellos.

A la hora de extracción de información desde documentos escritos en lenguaje natural, desde los términos extraídos a partir de dichos documentos se puede definir una base semántica para un dominio determinado si extraemos los términos desde dichos documentos. Podríamos crear una base semántica (diccionario semántico) relacionada con el dominio de dichos documentos y se podría realizar una comparativa de los documentos, o también detectar términos similares que podrían renombrarse o unificarse al ser estos pertenecientes al mismo dominio semántico. Esta base semántica ayudaría a mantener la consistencia semántica de dichos documentos.

Un ejemplo de esta aplicación se da en el trabajo: "Discovering Healthcare Processes from Natural Language Documents: a case study on COVID-19" [2]. En este trabajo se extraen los procesos de negocio que definen los síntomas y tratamiento del COVID-19 a partir de documentos escritos en lenguaje natural. A partir de este trabajo y utilizando una base semántica común al dominio de síntomas y tratamiento del COVID-19, podría comprobarse la consistencia de distintos documentos médicos sobre un dominio relacionado. Este es un trabajo futuro que se llevará a cabo próximamente basándose en el trabajo explicado en esta memoria.

En el ámbito de comparación semántica en BPMN y entre servicios web:

Existen varios trabajos que, utilizando una base semántica común, realizan matching semántico entre distintos modelos representados en BPMN[3] de un mismo dominio. Por su parte, otros trabajos expresan la necesidad de extracción de servicios web desde repositorios para hacer un matching entre los servicios web disponibles para comprobar qué servicios se adaptan mejor a los criterios de búsqueda. Estos son algunos ejemplos:

Pietsch et al. [4] enfatizan en su trabajo la importancia de algoritmos semánticos para llevar a cabo comparaciones de modelos en BPMN. Por su parte, Tibermacine y C. Foudil [5] utilizan en su trabajo distintas estructuras y métricas de similitud semántica para establecer una medida de similitud entre diferentes servicios web WSDL, especialmente comparando elementos de dichos servicios web como operaciones, mensajes y parámetros.

Teniendo en cuenta los trabajos relacionados y la utilidad que se le puede dar a la generación de una base de datos léxica asociada a un dominio determinado para utilizarla como base semántica a la hora de realizar comparativas entre distintos modelos, servicios web o documentos escritos en lenguaje natural, se decidió la realización de este trabajo.

Este trabajo parte del trabajo de fin de grado realizado por Celia Astorga Hurtado [6]. En su proyecto Celia implementaba un asistente para la generación semántica a partir de un dominio, llamado WordDomain. Este trabajo se ha ampliado y mejorado de las siguientes formas:

- Se realiza la obtención de un conjunto de términos inicial mayor añadiendo nuevas ontologías para realizar la primera búsqueda, siendo estas Linked Open Vocabularies [7] y DBpedia [8].
- Dado que a veces existe una ontología ya creada asociada a un dominio determinado, se permite la selección de hasta dos ontologías más cualesquiera seleccionadas por el usuario.
- Se ha llevado a cabo la refactorización del código original simplificando así la comprensión de los métodos existentes.
- Se ha mejorado la interfaz gráfica simplificando y detallando todas las partes del proceso de generación del dominio semántico para lograr una mayor compresión y sencillez para el usuario.

A parte de las mejoras realizadas también se ha desarrollado una nueva aplicación que trabaja en conjunción con WordDomain llamada "BPMN Comparator".

Esta adición utiliza las características mejoradas de la herramienta WordDomain para obtener un dominio semántico a partir de las tareas extraídas de uno o varios modelos en notación BPMN, que llamaremos conjunto A, con el objetivo de establecer una comparativa con el dominio de un conjunto B de modelos BPMN (esta comparativa puede ser uno a uno o uno a varios, comparando un modelo inicial con un conjunto de modelos) y poder establecer de manera cuantitativa una similitud semántica entre ambos conjuntos de modelos dentro de un mismo dominio. Se trata de un proceso semiautomático muy sencillo para el usuario, consta de los siguientes pasos:

- Seleccionar el tipo de comparativa deseado, habiendo disponibles:

    - Simple: Tanto el conjunto de BPMN A como el B estarán formados por un único diagrama, respectivamente. A partir de dichos conjuntos se extraerán sus tareas y se obtendrá un dominio semántico para cada uno. Tanto las tareas de los conjuntos como sus respectivos dominios serán comparados utilizando varios algoritmos de comparativa semántica.
    - Múltiple: El conjunto de BPMN A y el B podrán estar compuestos de más de un diagrama. Se extraerán las tareas de ambos conjuntos y se generará un dominio semántico único para el conjunto de A. Este será comparado individualmente por los diagramas integrantes del conjunto B con varios algoritmos de comparativa semántica.
- Selección de los conjuntos de BPMN A y B.
- De manera similar a WordDomain, seleccionar el tipo de búsqueda, de acuerdo al nivel semántico, deseado para generar el dominio (mismo nivel, nivel superior, nivel inferior o alguna combinación de las anteriores).

OBJETIVOS

Como ya se ha comentado previamente, este trabajo está basado en un Trabajo Fin de Grado (TFG) previo en el que se implementaba una primera versión del diccionario semántico WordDomain. Partiendo de dicho trabajo, los objetivos generales que se plantearon son los siguientes: implementación de varias mejoras al trabajo previo y ampliación del ámbito de aplicación y ejemplo de uso de WordDomain en el ámbito de matching de modelos definidos en notación BMPN.

Para llevar a cabo los objetivos generales, se han definido los siguientes objetivos específicos, que serán explicados comparándolo con el trabajo previo desarrollado:

WordDomain

Las funcionalidades originales del trabajo de Celia Astorga se resumen a continuación:

WordDomain generaba de forma semiautomática una base semántica que aporte un conjunto de términos relacionados con un dominio concreto y aportar un grupo de sinónimos asociados a cada término. Para ello se realizan los siguientes pasos:

1. Obtención del conjunto de términos inicial, partiendo del dominio seleccionado por el usuario. Dicho conjunto se obtiene a través de Ontobee [9], un almacén de conjuntos de ontologías que permite la ejecución de consultas SPARQL.
2. Partiendo del conjunto de términos obtenidos, realizar una nueva búsqueda sobre una base de datos estándar para la obtención de los sinónimos.
3. Generación del diccionario semántico y su exportación a un fichero .csv.

Teniendo esto en cuenta, en este trabajo se han mejorado y ampliado la funcionalidad de WordDomain de la siguiente forma:

Obtener un conjunto de términos inicial mayor.

En el trabajo original se utilizaba como base semántica Ontobee. La primera mejora consiste en ampliar esta base semántica para la búsqueda inicial. Así, se le dará al usuario la posibilidad de realizar la búsqueda inicial de términos en más de un servidor de conjuntos de términos relacionados entre sí mediante vínculos semánticos, y que permitan la ejecución de consultas SPARQL.

La primera búsqueda se hacía inicialmente en **Ontobee**, además se han añadido dos nuevas bases semánticas. Estas son como en el trabajo original de Celia, **DBpedia**, un proyecto de extracción de datos de Wikipedia, **Linked Open Vocabularies** (LOV), un catálogo de vocabulario que describe datos encontrados en la Web.

Estas dos nuevas fuentes semánticas han sido elegidas porque incluyen un número significativo de términos relacionados con dominios muy dispares, lo cual ampliará el conjunto de términos obtenidos. Además, ambas ofrecen endpoints para la ejecución de consultas SPARQL y son bases semánticas que mantienen su contenido actualizado.

Por último, el usuario tendrá la opción de introducir un endpoint a su conveniencia para aumentar aún más el conjunto de resultados, siempre y cuando permita consultas SPARQL.

Edición del diccionario semántico antes de la exportación.

En la versión original de WordDomain, se refinaban los términos obtenidos, pero no era lo suficientemente intuitivo para el usuario y tampoco se permitía la edición libre de cada uno de los términos, definiciones o sinónimos.

En este trabajo, hemos mejorado esta característica, de forma que antes de la exportación el usuario tendrá la opción de editarlo como más le convenga, esto implica la agregación, modificación o eliminación de términos, definiciones y sinónimos.

Refactorización del código base de WordDomain.

Utilizando las herramientas que proporciona Django [10] para el almacenamiento de información en bases de datos, se puede refactorizar el código original para que el proceso de guardado de los resultados y su posterior exportación se realicen de manera más simple.

Además, esto aporta un historial local de las búsquedas realizadas y diccionarios generados al que el usuario tiene fácil acceso a través de la página de administración que proporciona Django.

Mejora de la Interfaz de Usuario.

Se realizará una mejora sobre la Interfaz de Usuario de la versión original de WordDomain con el objetivo de que los pasos para la generación del diccionario semántico sean más claros para el usuario.

Opción para la exportación del diccionario en formato xml.

A la hora de exportar el diccionario semántico, originalmente únicamente se podía exportar a formato csv. En este trabajo se ha ampliado el formato, de forma que el diccionario se podrá obtener además en un fichero xml, ya que es un formato estándar qué, aunque sencillo de usar, es muy versátil y nos permite tratar un gran volumen de datos.

Comparativa semántica de modelos BPMN

Un uso que puede tener la obtención de diccionarios semánticos es la posibilidad de realizar comparativas entre procesos de negocio dentro de un mismo dominio semántico. Este tipo de comparativas pueden utilizarse para comprobar posibles similitudes o diferencias entre modelos desarrollados por distintos grupos de desarrolladores, modelos que quieren ser unificados (por ejemplo, a la hora de fusionarse distintas empresas con una misma línea de negocio) o modelos que necesitan utilizar un lenguaje similar. En este trabajo se proporciona la posibilidad de extraer los componentes fundamentales de un BPMN y poder obtener un diccionario semántico a través de ellos, una vez generado el diccionario, establecer una comparativa entre distintos modelos definidos en el lenguaje BPMN 2.0 [11] basándonos en un mismo dominio semántico.

Por tanto, los objetivos específicos definidos en este trabajo pueden resumirse como los siguientes:

Extracción de las tareas que componen los diagramas BPMN

Un BPMN (Business Process Model and Notation) es una notación gráfica estandarizada que permite de manera sencilla y legible el modelado de procesos de negocio. Dentro de los elementos que componen estos diagramas, será necesario localizar y extraer aquellos que nos permitan obtener un diccionario semántico posteriormente, analizando la composición de esta notación deberemos extraer y utilizar las tareas, puesto que representan una sola unidad de trabajo indivisible en el proceso de negocio.

Generación de diccionario semántico

Utilizando las tareas extraídas anteriormente se procederá a la generación de un diccionario semántico. Para ello se deberá implementar primeramente una API de WordDomain que permita la obtención de los términos que compondrán el diccionario de manera automática.

Comparativa de los diccionarios semánticos

Una vez obtenido el diccionario semántico se realizará una comparativa utilizando algoritmos de comparación semántica para establecer la similitud existente entre el dominio del diccionario generado y los dominios de otros BPMN.

CAPÍTULO 2: ESTADO DEL ARTE

En la Ingeniería del Software existen multitud de aplicaciones y herramientas para el tratamiento de datos (búsquedas, comparativas, extracciones, clasificaciones, etc.) sin embargo este tratamiento de la información se realiza en algunos casos a nivel sintáctico puesto que las herramientas software no contemplan el significado a nivel semántico que poseen los mismos datos con los que operan.

La primera gran propuesta de dotar de significado que fuese comprensible para herramientas software fue la Web Semántica en 1998, planteada por Tim Bernesrs-Lee, creador de la WWW[12].

La Web Semántica es un proyecto impulsado por el W3C (World Wide Web Consortium). En complementación a la llamada "Web of documents" [13], una web basada en documentos y enlaces de hipertexto específicamente diseñada para la lectura y comprensión humana, se plantea la web semántica, este término se refiere a la visión de una web de datos enlazados, datos dotados de significado con el objetivo de ser interpretados y comprendidos no solo por agentes humanos sino también por agentes computarizados.

En este trabajo se utilizan herramientas que ya se contemplaban para el desarrollo de la Web Semántica, son las siguientes:

RDF

Resource Description Framework (RDF) es un modelo estandarizado para el intercambio de datos en la web. Está basado en la identificación de recursos web haciendo uso de las Uniform Resource Identifiers (URIs) [14], se propone un conjunto de proposiciones simples llamados tripleta, compuesta por sujeto, predicado y objeto:

- **Sujeto**. Lo compone el recurso, un recurso representa cualquier elemento descrito por expresiones RDF, desde una imagen, a un artículo o incluso a una página web completa.
- **Predicado**. Compuesto por una propiedad, este es un aspecto característico que describe el recurso.

- **Objeto.** Formado por el valor de la propiedad pudiendo tratarse esta de otro recurso especificado por una URI o un literal.

Se pueden realizar consultas sobre la información definida en RDF haciendo uso de SPARQL [15], un lenguaje de consulta de grafos RDF.

Un ejemplo de uso del modelo RDF es el siguiente diagrama en el que se representa que Ora Lassila es el creador del recurso http://www.w3.org/Home/Lassila:

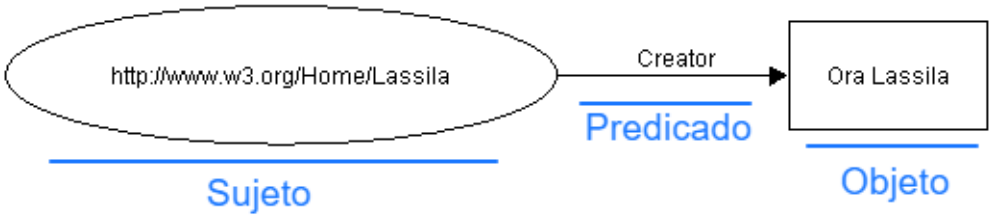

Figura 1: Diagrama RDF

OWL

Web Ontology Language (OWL) [16] es un lenguaje extendido a partir de RDF y está diseñado para definir ontologías en la web. Tiene mayor capacidad para expresar significado y semántica que RDF y por tanto puede usarse cuando la información a describir necesita ser procesada por programas y aplicaciones.

OWL ofrece tres sublenguajes con distinto nivel de expresividad para adaptarse a las necesidades del usuario:

- **OWL Lite**. Diseñado principalmente para clasificaciones jerárquicas y restricciones simples, menor complejidad.
- **OWL DL**. Diseñado para usos que requieran máxima expresividad conservando al mismo tiempo completitud computacional, garantizando así que todas las conclusiones sean computables, y resolubilidad, asegurando que todos los cálculos se resolverán en tiempo finito.
- **OWL Full**. Diseñado para mantener máxima expresividad y libertad sintáctica, pero sin garantías computacionales.

Ontologías

En la Web Semántica, los vocabularios [17] y ontologías se usan para clasificar términos que se usan en dominios concretos y expresar posibles relaciones o restricciones entre ellos. Gracias a estos bancos de información tenemos disponible una gran cantidad de datos enlazados que podemos extraer para el propósito de este trabajo:

Ontobee

Ontobee es un proyecto colaborativo mantenido por la Universidad de Michigan que ha sido diseñado para la visualización, consulta y desarrollo de términos ontológicos.

Linked Open Vocabularies

Linked Open Vocabularies (LOV) es otra plataforma que trata de dotar a la información que existe en la web de significado. Usando dialectos de RDF como OWL, LOV, en sus vocabularios, reúne definiciones de un conjunto de clases y propiedades útiles de descripción de elementos que pertenecen a un dominio concreto.

DBpedia

DBpedia es una plataforma que extrae información estructurada de Wikipedia con el objetivo de hacer público y accesible un conjunto de vocabularios y ontologías referentes a distintos dominios.

CAPÍTULO 3: METODOLOGÍA

Para la realización de este trabajo y dadas sus características, se ha seguido una metodología de desarrollo ágil, llevando a cabo un desarrollo incremental. Para ello, se ha dividido el trabajo en las fases de análisis, desarrollo, implementación, pruebas y documentación.

Análisis

En la fase de análisis se llevó a cabo el estudio del trabajo previo y la definición de mejoras que podrían implementarse. Como resultado, se identificaron los requisitos funcionales y no funcionales para la aplicación. Además, se han desarrollado diagramas de casos de usos para describir la funcionalidad de cada parte.

WORDDOMAIN

Requisitos funcionales

- El usuario podrá elegir entre varias ontologías o combinaciones de ellas para realizar la búsqueda.
- El usuario podrá añadir hasta un máximo de dos SPARQL endpoint de su elección para realizar la búsqueda.
- El usuario podrá modificar el diccionario resultante antes de la exportación de las siguientes maneras.
    - Edición de los términos, definiciones y sinónimos.
    - Adición de términos, definiciones y sinónimos.
    - Eliminación de términos, definiciones y sinónimos.
- El usuario podrá exportar el diccionario resultante en formato XML.
- El sistema llevará un registro de los términos buscados y los diccionarios generados con ellos a través del framework de bases de datos proporcionado por Django.
- El usuario podrá utilizar un fichero .csv con los términos de su preferencia para la entrada inicial de datos. Esto le permite comenzar el proceso de generación del diccionario usando más de un término.

Requisitos no funcionales

- La aplicación deberá ejecutarse de manera online para poder realizar las búsquedas y generar el diccionario de manera correcta.
- La aplicación será resistente ante errores a lo largo del proceso de generación del diccionario.
- El sistema tendrá una interfaz sencilla, explicando al usuario la manera de proceder en cada paso.
- La aplicación se desarrollará de manera modular para permitir la integración de manera más sencilla de trabajos futuros.
- La aplicación estará disponible íntegramente en inglés.

Casos de uso

1. El usuario introduce un término, o varios en formato .csv, para comenzar la creación del diccionario.
2. WordDomain busca en las ontologías seleccionadas por el usuario el término introducido. Se muestran los resultados y el usuario puede elegir a conveniencia.
3. WordDomain generará sinónimos a raíz de los términos previamente seleccionados. El usuario puede elegir los sinónimos generados a conveniencia.
4. WordDomain generará un diccionario acorde a los términos y sinónimos seleccionados.
5. El usuario podrá añadir, eliminar y/o modificar los términos del diccionario.
6. WordDomain exporta el diccionario final acorde a las selecciones hechas por el usuario. El usuario podrá decidir si el fichero resultante será un .csv y/o un .xml.

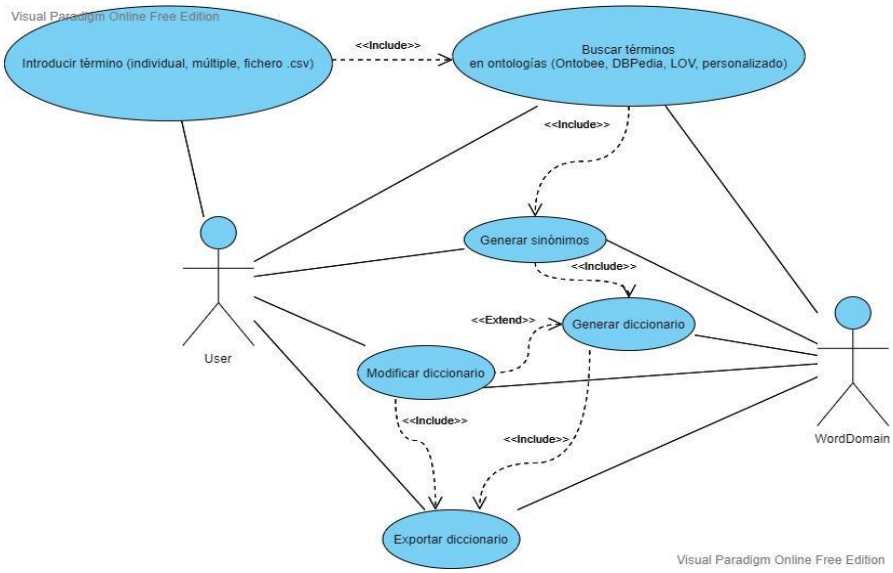

Figura 2: Diagrama de casos de uso de WordDomain

BPMN COMPARATOR

Requisitos funcionales

- El usuario podrá seleccionar los modelos BPMN que desee comparar. Dos tipos de entradas estarán disponibles:
    - Simple. Comparativa 1 a 1, se usarán dos modelos BPMN.
    - Múltiple. Comparativa M a N, se usarán dos conjuntos de modelos BPMN.
- La aplicación analizará dichos modelos para extraer el nombre de las tareas que componen los diagramas.
- Las tareas que han sido extraídas compondrán nuestro conjunto inicial de términos para generar el diccionario semántico con ayuda de WordDomain. De esta forma, el diccionario estará relacionado con el dominio del modelo representado. Este proceso será automático.
- Se creará un diccionario para cada modelo BPMN. Este diccionario estará formado por los diccionarios creados a raíz de las tareas extraídas del modelo.
- El sistema llevará un registro de los términos buscados y los diccionarios generados con ellos a través del framework de bases de datos proporcionado por Django.
- La aplicación analizará el diccionario generado desde cada modelo y los comparará entre sí para calcular la similitud que existe entre los modelos BPMN.

Requisitos no funcionales

- La aplicación deberá ejecutarse de manera online para poder realizar las búsquedas y generar el diccionario de manera correcta.
- La aplicación será resistente ante errores a lo largo del proceso de generación del diccionario.
- El sistema tendrá una interfaz sencilla, explicando al usuario la manera de proceder en cada paso.
- La aplicación se desarrollará de manera modular para permitir la integración de manera más sencilla de trabajos futuros.
- La aplicación estará disponible íntegramente en inglés.

Casos de uso

- El usuario selecciona los dos BPMN que desea comparar.
- La aplicación extrae los nombres de las tareas de cada diagrama.
- La aplicación realiza una búsqueda en WordDomain para cada tarea. Se genera un diccionario con los resultados para cada BPMN.
- La aplicación realiza la comparativa en búsqueda de similitudes e informa al usuario.

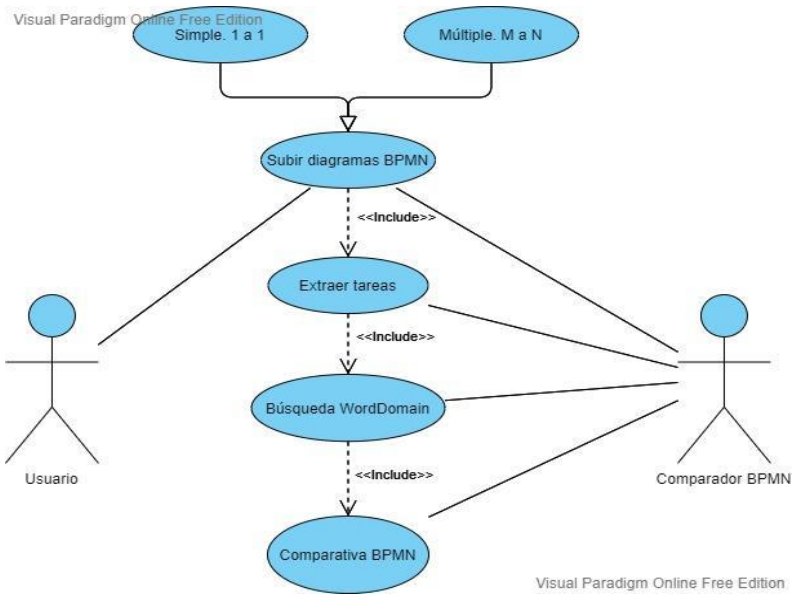

Figura 3: Diagrama de casos de uso de BPMN Comparator

Tipo de comparativa

A la hora de realizar la comparativa de los resultados, para buscar la mejor forma de medir la similitud de los BPMN se estuvieron investigando varios algoritmos de comparativa semántica. Existe un gran número de algoritmos que realizan una comparativa semántica y devuelven un resultado numérico dependiendo del nivel de similitud entre términos o frases. Finalmente, dada su facilidad de aplicación y adaptabilidad al trabajo desarrollado se decidieron utilizar el Índice de similitud de Jaccard y la similitud de coseno. Se ha planteado el uso de dos algoritmos distintos, puesto qué el Índice de similitud de Jaccard es muy sensible a la diferencia de tamaño entre los conjuntos estudiados, por tanto, se decidió implementar también la similitud de coseno para subsanar la desventaja de Jaccard contra conjuntos de diferente tamaño y así aportar más información al usuario sobre la comparativa de dominio.

- Índice de similitud de Jaccard [18]:

  Este coeficiente mide el grado de similitud entre dos conjuntos, se define como la cardinalidad de la intersección de ambos conjuntos dividida por la cardinalidad de su unión.

$$\mathcal{J}(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Figura 4: Formula del índice de similitud de Jaccard

- Similitud de coseno [19]:

  Se define como una medida de similitud entre dos vectores distintos de cero de un espacio de producto interno.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Figura 5: Formula de la similitud de coseno

CAPÍTULO 4: IMPLEMENTACIÓN Y DESARROLLO

WORDDOMAIN

Interfaz de usuario

La interfaz de usuario ha sido retocada con respecto al proyecto base para aumentar la legibilidad y sencillez para el usuario. A continuación, se muestran las distintas páginas que componen WordDomain:

Home

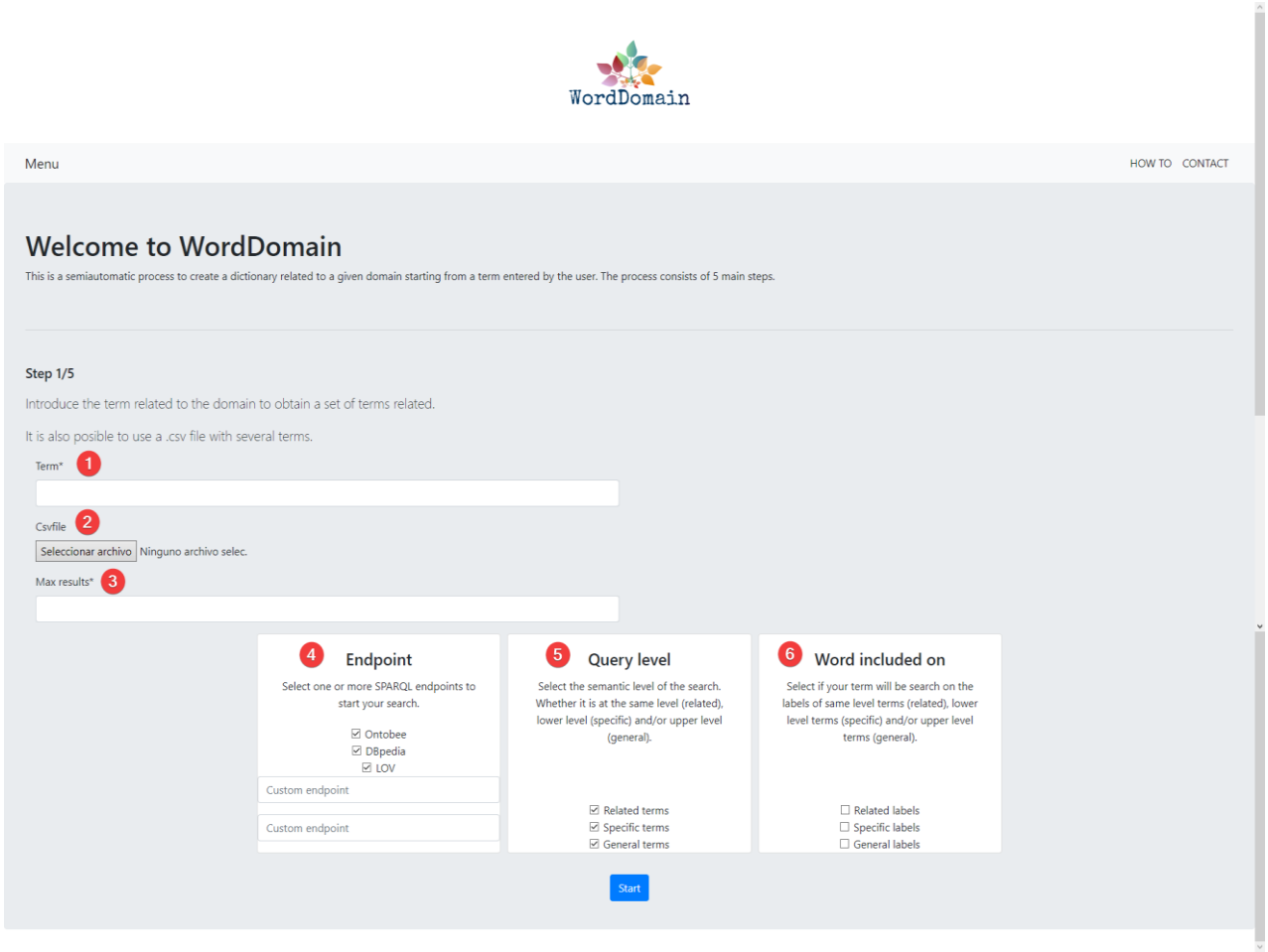

Figura 6: Página principal WordDomain

La página principal es donde comienza el proceso de generación del diccionario semántico. En 1 el usuario introducirá el término inicial de la búsqueda, si precisase de más términos con los que quiera realizar el procedimiento, en el campo 2

tiene la opción de subir un archivo csv conteniendo todos los términos con los que desee. El campo 3 marca la cantidad de resultados máximos para cada término que la aplicación obtendrá. En el espacio 4 el usuario deberá seleccionar las ontologías en las que desee comenzar la búsqueda, se contemplan dos campos de texto para incluir ontologías personalizadas que precisen de un acceso SPARQL con el que realizar las consultas. Con los elementos 5 y 6 el usuario podrá seleccionar el nivel semántico de la consulta (búsqueda en subclases, superclases o clases del mismo nivel) y el nivel semántico de la búsqueda en las "label".

Conjunto inicial

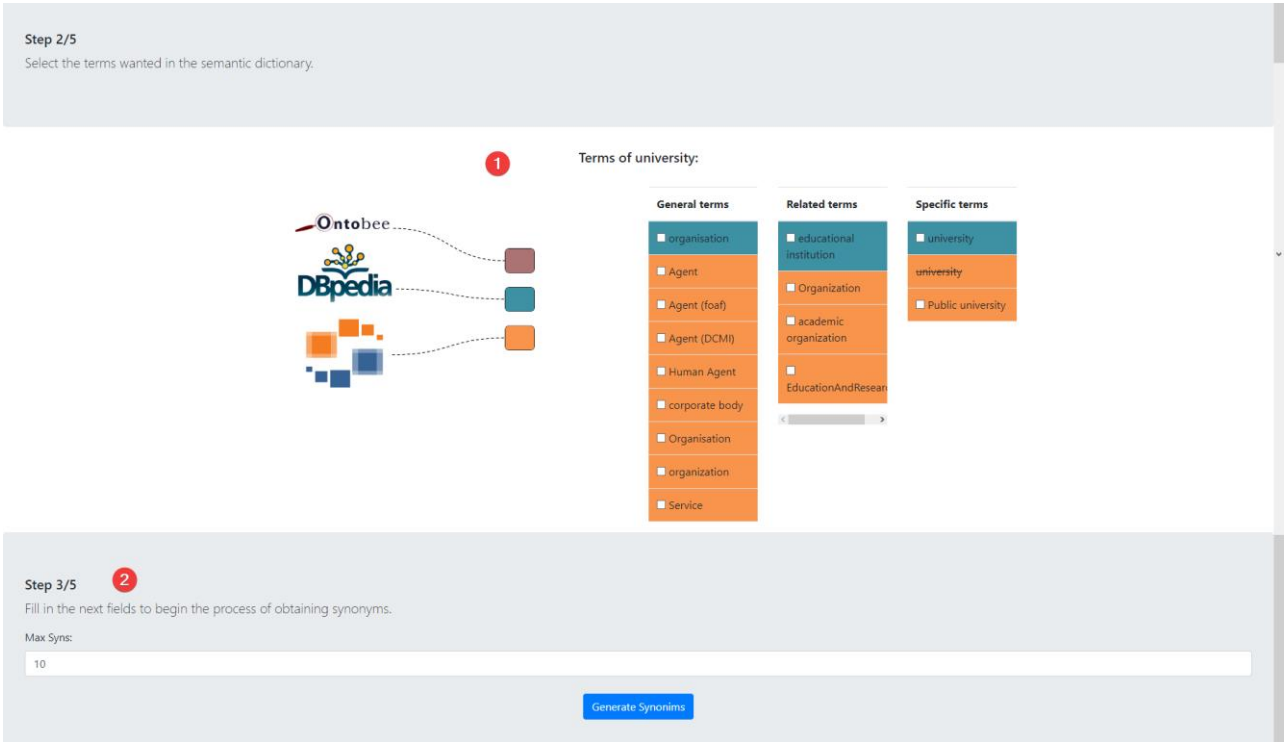

Figura 7: Resultados después de introducir el dominio

Una vez obtenido el conjunto inicial de términos, en el área 1 el usuario deberá seleccionar aquellos que quiera incluir en el diccionario semántico. En el punto 2 sencillamente se marcarán la cantidad máxima de sinónimos que la aplicación buscará para cada término elegido.

Sinónimos

Figura 8: Sinónimos obtenidos

La aplicación mostrará a continuación los sinónimos encontrados para los términos seleccionados en la pantalla anterior. En esta sección, el usuario puede seleccionar o deseleccionar aquellos resultados que desee.

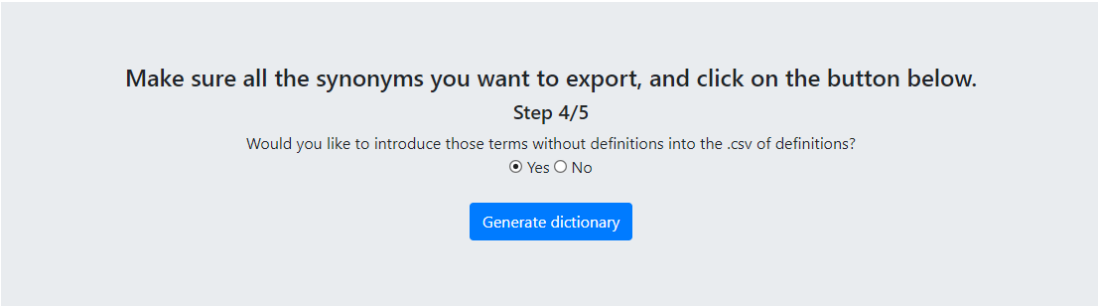

Figura 9: Formulario de generación de diccionario

Después de comprobar la selección de sinónimos solo resta seleccionar si deseamos incluir en el diccionario aquellos términos para los que no se encontraron sinónimos.

Exportar

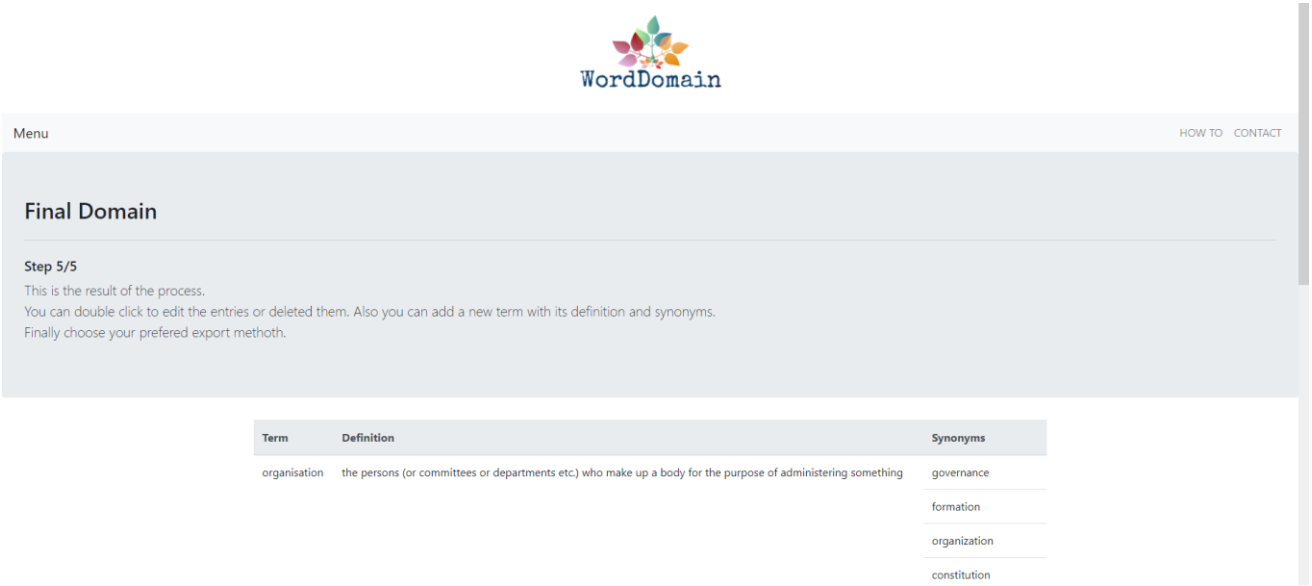

Figura 10: Diccionario resultante

En la página final se podrá ver el diccionario resultante. El usuario puede, utilizando el doble click, editar el término su definición o sus sinónimos.

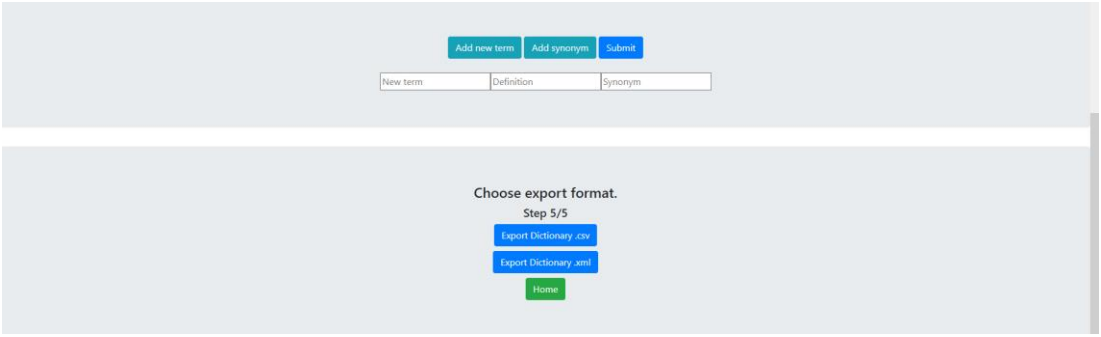

Figura 11: Formulario para añadir un término y opciones de exportación

Por último, el usuario se encontrará diversos botones que le permitirán añadir un nuevo término y seleccionar el tipo de fichero para exportar el diccionario, siendo las opciones en formato .xml o .csv.

Refactorización

Parte del desarrollo de este trabajo ha consistido en una refactorización del código base, con este proceso se ha conseguido una mayor legibilidad del código además de facilitar la integración con ampliaciones futuras.

Base de datos

Uno de los cambios más importantes en WordDomain ha sido el uso de la base de datos que proporciona Django para el almacenamiento de los resultados obtenidos por la aplicación. La configuración por defecto usa **SQLite [20]** para establecer la base de datos, estas son algunas de sus ventajas con respecto a otros sistemas de gestión de bases de datos relacionales, y las razones por las que se ha decidido su uso:

- Ligera, ocupa poco en disco.
- Auto contenida, no necesita de la instalación de dependencias externas para su funcionamiento.
- Fácil integración con la aplicación, no necesita de una gestión activa, no es necesario iniciarla o pararla.

Establecido el sistema de gestión de bases de datos así es como se utiliza con los datos de WordDomain:

Django propone el uso de *Modelos* para definir los campos y el comportamiento de cualquier tipo de información que se vaya a almacenar en la base de datos. Cada modelo se verá representado por una tabla y los atributos que contenga serán los campos de esta, lo que permite la definición de la base de datos de manera sencilla. Además, Django ofrece una API de acceso a los datos para poder crear, consultar, actualizar y borrar la información de una manera cómoda.

Teniendo en cuenta la información necesaria para el funcionamiento de WordDomain estos han sido los modelos definidos y sus atributos, que como se ha mencionado anteriormente se convertirán en las tablas de la base de datos:

- **Session**. Se creará una "Session" cada vez que iniciemos el proceso de generación de diccionario con WordDomain, con el objetivo de que todos los resultados obtenidos durante el proceso estén fácilmente identificados. Atributos:
    - id. Identificador, clave primaria.
    - value. Corresponderá al término introducido para comenzar el proceso de generación de diccionario en WordDomain.
- **Term.** Cada término obtenido de WordDomain se almacenará en esta tabla. Atributos:
    - id. Identificador, clave primaria.

  - session_id. Clave secundaria referenciando a la "Session" a la que pertenece.
  - value. El término obtenido.
  - definition. Definición del término que se obtendrá a lo largo del proceso.
  - endpoint. SPARQL endpoint elegido para la obtención de este término.
  - relation. nivel de la query elegido para la obtención de este término, es decir si la búsqueda se ha realizado en clases del mismo nivel semantico, en subclases o en superclases.
  - duplicate. Verdadero si el término ya se había obtenido a través de otra "relation" o "endpoint".
  - chosen. Verdadero si el usuario lo ha seleccionado para buscar sinónimos y añadirlo al diccionario final.
- **Synonym.** Sinónimos pertenecientes a cada "Term". Atributos:
  - id. Identificador, clave primaria.
  - session_id. Clave secundaria referenciando a la "Session" a la que pertenece.
  - term_id. Clave secundaria referenciando al "Term" al que pertenece.
  - value. El sinónimo obtenido.
  - chosen. Verdadero si el usuario lo ha escogido para añadirlo al diccionario final.

Siendo estos los modelos definidos el diagrama Entidad Relación es el siguiente:

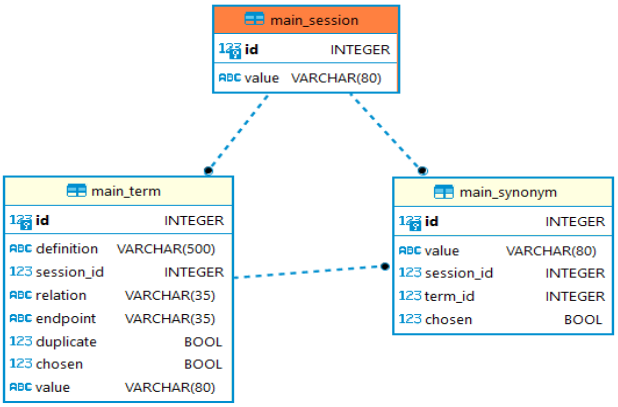

Figura 12: Diagrama Entidad Relación

Pruebas

En esta sección se mostrarán un pequeño conjunto de pruebas de la ejecución de WordDomain para el término “conference”. Las pruebas se realizarán teniendo como parámetros base: el término, el número de resultados máximo y las ontologías a usar, siendo en este caso todas las disponibles. Se mostrarán la cantidad de términos obtenidos por la ontología y los mismos. También se medirá el tiempo que tarda la aplicación en mostrar resultados, teniendo en cuenta que este es orientativo. Dado que todos los resultados se muestran a la vez y no según se vayan obteniendo, el tiempo medido marcará siempre a la ontología más lenta en acabar la consulta.

Tabla 1: Prueba conference en general y specific terms

Término: conference **Max results: 15**	**General Terms**	**Related Terms**	**Specific Terms**
Nivel consulta	✓	✓	✓
Nivel búsqueda en “labels”	✓	-	✓
Resultados Ontobee	1	1	2
	- conference	- academic conference	- medical conference - sexology conference
Resultados DBpedia	1	1	2
	Repetido con Ontobee	Repetido con Ontobee	Repetidos con Ontobee
Resultados LOV	-	-	-
	-	-	-
Tiempo (seg)	40.64		

Con esta configuración podemos ver que se han obtenido un total de 8 resultados entre Ontobee y DBpedia, habiendo resultado estos últimos repetidos.

Tabla 2: Prueba conference en related terms

Término: conference **Max results: 15**	**General Terms**	**Related Terms**	**Specific Terms**
Nivel consulta	✓	✓	✓
Nivel búsqueda en "labels"	-	✓	-
Resultados Ontobee	-	-	-
	-	-	-
Resultados DBpedia	-	-	-
	-	-	-
Resultados LOV	-	-	-
	-	-	-
Tiempo (seg)	Time Out		

Con esta configuración observamos que las consultas han tardado demasiado tiempo puesto que se ha superado el tiempo límite establecido, de un minuto, para realizarlas.

Tabla 3: Prueba conference en general y related terms

Término: conference **Max results: 15**	**General Terms**	**Related Terms**	**Specific Terms**
Nivel consulta	✓	✓	✓
Nivel búsqueda en "labels"	✓	✓	-
Resultados Ontobee	-	-	-
	-	-	-
Resultados DBpedia	-	-	-
	-	-	-
Resultados LOV	-	-	-
	-	-	-
Tiempo (seg)	Time Out		

Se ha sobrepasado el tiempo límite de ejecución, no se devuelven resultados.

Tabla 4: Prueba conference en general terms

Término: conference **Max results: 15**	**General Terms**	**Related Terms**	**Specific Terms**
Nivel consulta	✓	✓	✓
Nivel búsqueda en "labels"	✓	-	-
Resultados Ontobee	-	-	-
	-	-	-
Resultados DBpedia	-	-	-

	-	-	-
Resultados LOV	-	-	-
	-	-	-
Tiempo (seg)	Time Out		

Se ha sobrepasado el tiempo límite de ejecución, no se devuelven resultados.

Tabla 5: Prueba conference en specific terms

Término: conference **Max results: 15**	**General Terms**	**Related Terms**	**Specific Terms**
Nivel consulta	✓	✓	✓
Nivel búsqueda en "labels"	-	-	✓
Resultados Ontobee	-	-	-
	-	-	-
Resultados DBpedia	1	1	1
	- event	- societal event	- academic conference
Resultados LOV	6	5	5
	- Corporate body - Authority Resource - financial entity - expression - expression collection - publication	- Conference or Event - fee - academic proceedings - book - proceedings paper	- Series of conference or event - conference fee - conference paper - conference proceedings - conference poster
Tiempo (seg)	56.13		

En esta última prueba podemos ver como la búsqueda en las “labels” de los términos más específicos qué “conference”, es la opción más adecuada para este término concreto puesto que confiere más resultados, siendo un total de 19.

BPMN COMPARATOR

Interfaz de usuario

En este apartado del trabajo se mostrará el diseño de la interfaz gráfica de usuario explicando el procedimiento para poder realizar una comparativa entre diagramas BPMN.

Home

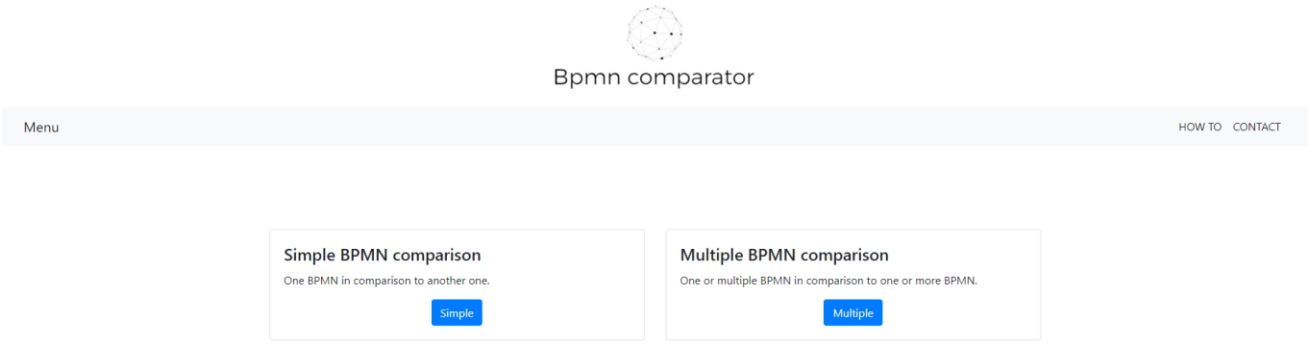

Figura 13: Página principal BPMN Comparator

En la imagen se muestra la pantalla de inicio de BPMN Comparator, en ella el usuario deberá seleccionar el tipo de comparativa que desea:

- **Simple BPMN comparison**: se realizará una comparativa de diagramas 1 a 1, primero se realizará una comparativa utilizando las tareas que componen cada uno de los diagramas. Finalmente, utilizando WordDomain, se generará un dominio para cada diagrama partiendo de las tareas extraídas anteriormente y se realizará su comparativa.
- **Multiple BPMN comparison**: se realizarán comparativas de N a M diagramas. El conjunto de diagramas N en este caso formará un solo dominio que se comparará de manera individual con cada elemento del conjunto de diagramas M. Por razones de rendimiento no se genera un diccionario semántico del conjunto M, por tanto, las comparativas se realizan utilizando el conjunto de

tareas obtenidas de N y el diccionario semántico obtenido también del mismo conjunto contra las tareas extraídas de cada elemento de M.

Simple BPMN comparison

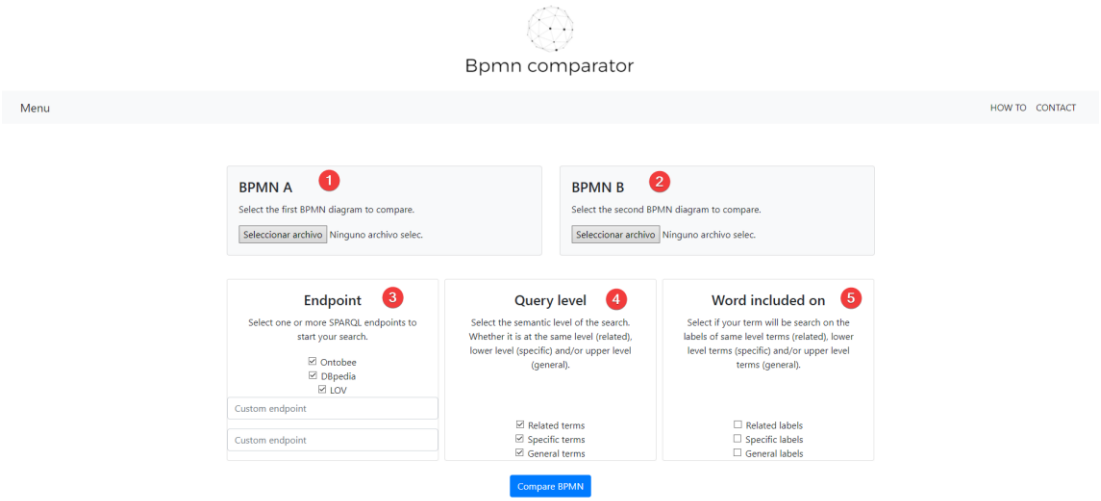

Figura 14: Página simple BPMN comparison

Como se puede apreciar la interfaz es muy similar a la escogida para WordDomain por tanto un usuario de WordDomain no tendrá complicación en utilizarla correctamente.

En **1** y **2** podremos subir con facilidad los dos diagramas que queramos comparar para comenzar el proceso. Los elementos **3, 4 y 5** nos permitirán justo como en WordDomain seleccionar la ontología deseada así como el nivel semántico de consulta y el nivel semántico de búsqueda en "labels".

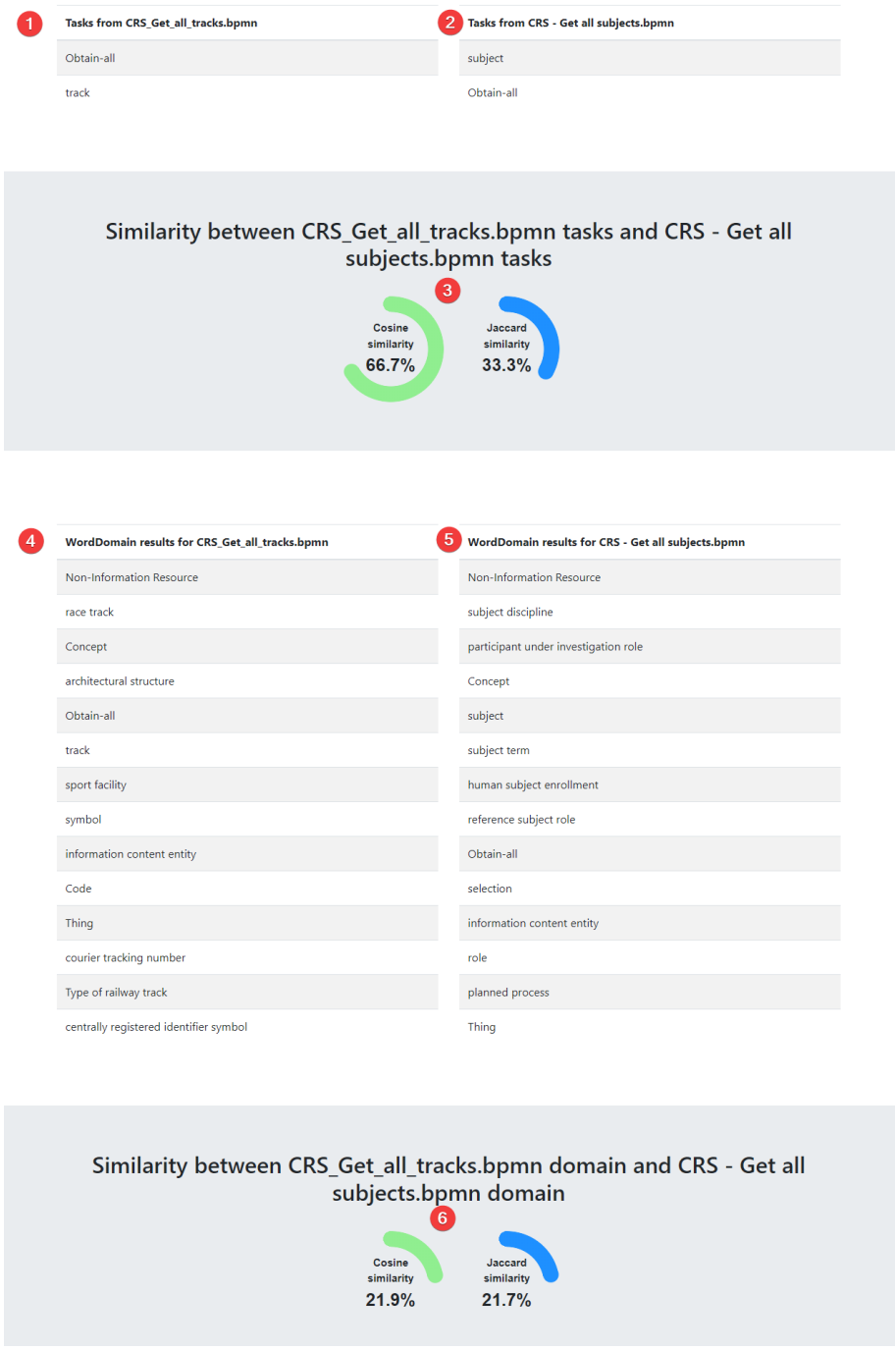

Figura 15: Resultados comparativa simple

Después del procesado de los diagramas se mostrarán los resultados de la siguiente forma:

1. Tareas obtenidas del modelo A.
2. Tareas obtenidas del modelo B.
3. Resultado de la comparativa entre las tareas obtenidas de **1** y **2.** Se proponen dos algoritmos distintos de comparativa semántica, la similitud del coseno y la similitud de Jaccard.

4. Dominio obtenido a través de WordDomain con las tareas del diagrama A.
5. Dominio obtenido a través de WordDomain con las tareas del diagrama B.
6. Resultado de la comparativa entre los dominios obtenidos de **4** y **5**. Se proponen dos algoritmos distintos de comparativa semántica, la similitud del coseno y la similitud de Jaccard.

Multiple BPMN comparison

Para el tipo de comparativa múltiple se utiliza el mismo diseño visto en la comparativa simple (imagen x), con la novedad de que el usuario ahora podrá subir más de un diagrama a la vez. Una vez procesados los diagramas la comparativa se verá de la siguiente forma:

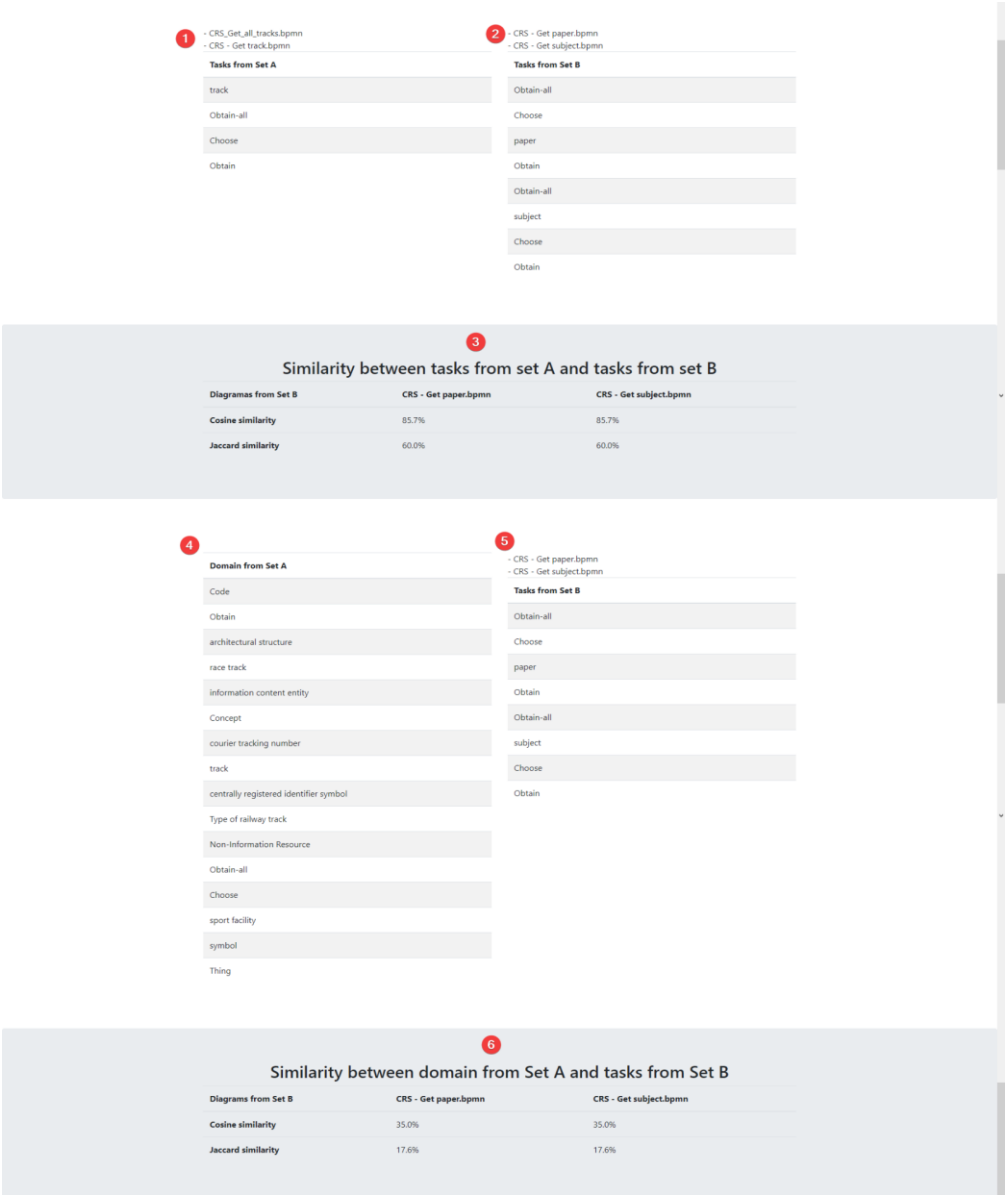

Figura 16: Resultados comparativa múltiple

1. Tareas obtenidas del modelo A.
2. Tareas obtenidas del modelo B.
3. Resultado de la comparativa entre las tareas obtenidas de **1** y **2,** se representa la similitud de manera individual para cada diagrama introducido en el conjunto B. Se proponen dos algoritmos distintos de comparativa semántica, la similitud del coseno y la similitud de Jaccard.
4. Dominio obtenido a través de WordDomain con las tareas del diagrama A.
5. Mismas tareas obtenidas del modelo B que en **2**.
6. Resultado de la comparativa entre las tareas obtenidas de **4** y **2,** se representa la similitud de manera individual para cada diagrama introducido en el conjunto B. Se proponen dos algoritmos distintos de comparativa semántica, la similitud del coseno y la similitud de Jaccard.

Herramientas Utilizadas

Python

Python [21] es un lenguaje de programación de alto nivel. Se decidió su uso en este trabajo dado que el proyecto base ya estaba desarrollado en este lenguaje, siendo esta la razón principal habría de añadir que Python muy legible y fácil de usar, además tiene una gran comunidad de desarrolladores de librerías y frameworks, como Django, que permite el desarrollo de aplicaciones web de manera sencilla.

Django

Django es un potente framework desarrollado para python que permite la creación de aplicaciones web de manera cómoda y simple.

La característica más notable para este trabajo sería la Base de Datos que nos proporciona el framework.

Bootstrap

Bootstrap [22] es uno de los frameworks de desarrollo front-end más populares, es de código abierto y posee características que hacen el desarrollo más sencillo y limpio, como el sistema de cuadrículas que posee para la colocación de los elementos en la página.

Ontobee

Ontobee es un servidor de datos diseñado específicamente para facilitar la visualización y las consultas a ontologías. Se ha decidido su uso en este trabajo por la cantidad de ontologías disponibles y por la posibilidad de realizar consultas SPARQL para la extracción de datos.

DBpedia

DBpedia es un proyecto que extrae información de wikipedia para que pueda ser compartida y accedida mediante un grafo de conocimiento. Se usa en este proyecto por su catálogo de datos y la posibilidad de extracción de los mismos mediante consultas SPARQL.

Linked Open Vocabularies

Linked Open Vocabularies (LOV) es una recopilación de vocabularios bien documentados con el objetivo de describir los datos y proveer conexiones semánticas entre ellos. Debido a dichas características se decidió su uso en este trabajo.

Wordnet

Wordnet [23] es una base de datos léxica que agrupa nombres, verbos, adjetivos y adverbios en conjuntos denominados synsets. Los synsets se interconectan entre ellos mediante relaciones semánticas y léxicas. Gracias a ella podemos extraer fácilmente sinónimos de cualquier término, y es por ello que se decidió su uso en el proyecto.

SPARQL

SPARQL es un lenguaje de consulta para RDF, es decir, datos representados por grafos dirigidos y etiquetados. Este modelo de representación es el usado por las herramientas de recopilación de datos mencionadas anteriormente (Ontobee, DBpedia, LOV), por lo tanto, su uso era necesario para el proyecto.

Javascript

Javascript [24] es un lenguaje de programación interpretado que se encarga de dotar de mayor interactividad y dinamismo a las páginas web. Ha sido utilizado principalmente para que el usuario pudiese modificar el diccionario generado por WordDomain.

Pruebas

En esta sección se mostrarán un pequeño conjunto de pruebas de la ejecución de BPMN Comparator. Durante la ejecución se usarán como parámetros base las ontologías DBPedia y LOV, se excluye Ontobee por razones de rendimiento puesto que tarda en devolver resultados. Se realizarán pruebas tanto para **Simple BPMN comparison** como para **Multiple BPMN comparison.** Los modelos seleccionados para estas pruebas son modelos diseñados en el ámbito CRS (Conference Review System). En las figuras 17 y 18 puede verse la representación gráfica de los modelos llamados "Delete paper" y "Delete Subject" respectivamente utilizados en dichas pruebas.

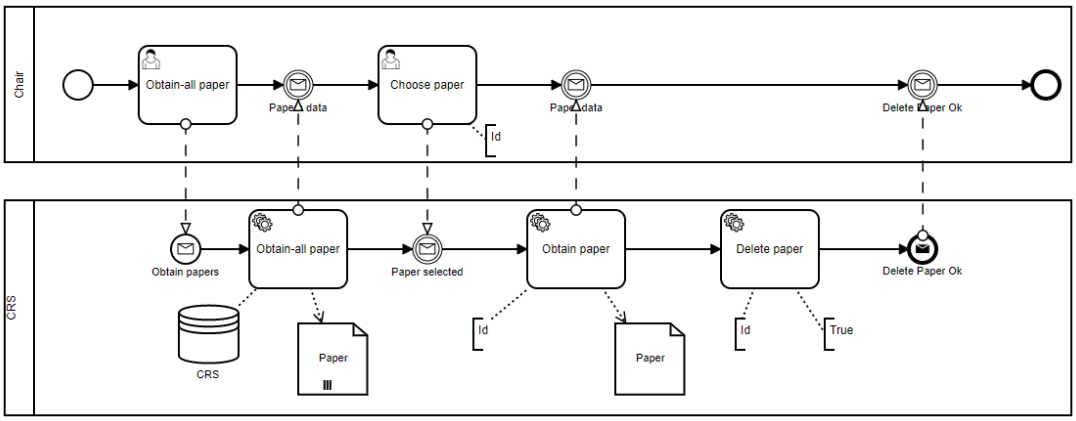

Figura 17: Modelo delete paper

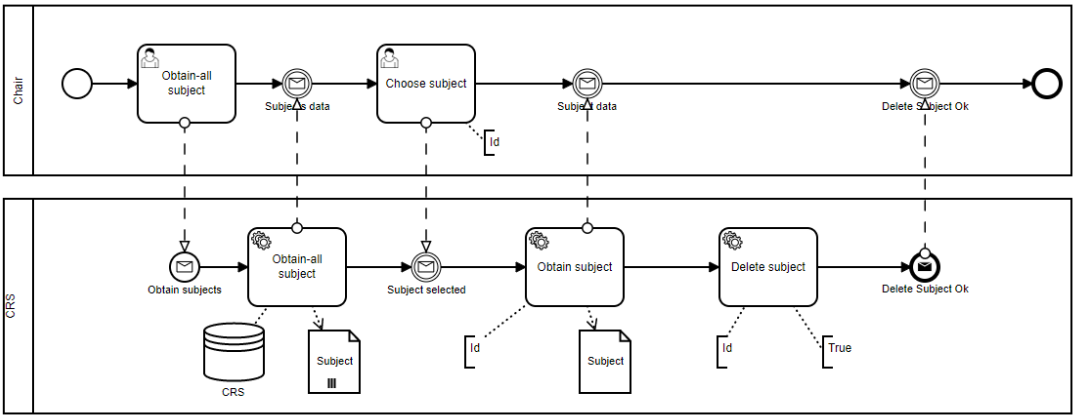

Figura 18: Modelo delete subject

Simple BPMN comparison

En este primer conjunto de pruebas se mostrarán las tareas extraídas de cada modelo BPMN, la similitud de coseno y jaccard resultante de comparar las tareas, el dominio obtenido de las tareas de cada modelo y por último la similitud de coseno y jaccard resultante de comparar los dominios.

Utilizando BPMN similares

Se realizarán tres primeras pruebas utilizando modelos similares, concretamente dos modelos que describen el proceso de borrado de un "paper" (CRS - Delete paper) y un "subject" respectivamente (CRS - Delete subject).

Tabla 6: Prueba con Delete paper y Delete subject en general terms

BPMN	**CRS - Delete paper**	**CRS - Delete subject**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✓ General Terms ✗ Related Terms ✗ Specific Terms	
Tareas Extraídas	- Obtain-all - Delete - paper - Obtain - Choose	- Obtain-all - Delete - Obtain - subject - Choose
Similitud entre tareas	Cosine Similarity: 87.5% Jaccard Similarity: 66.7%	

Dominio obtenido	- DeleteHealthInformationAction - DeleteAlarmAction - DeleteContactAction - Choose - DeleteMediaInformationAction - DeleteCalendarItemAction - paper - DeleteAction - DeleteReminderAction - Delete - Obtain - DeleteFileAction - DeletedFromStorageTrigger - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction	- DeleteHealthInformationAction - reference subject role - cohort role - DeleteAlarmAction - DeleteContactAction - Choose - DeleteMediaInformationAction - DeleteCalendarItemAction - DeleteAction - DeleteReminderAction - Delete - biological replicate role - Obtain - subject - subject role - DeleteFileAction - DeletedFromStorageTrigger - clinical subject role - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction - patient role
Similitud entre dominios	Cosine Similarity: 51.3% Jaccard Similarity: 70.4%	

Con esta configuración podemos observar que la similitud entre tareas y entre dominios es bastante alta, indicando que ambos modelos pueden pertenecer al mismo dominio semántico.

Tabla 7: Prueba con Delete paper y Delete subject en related terms

BPMN	**CRS - Delete paper**	**CRS - Delete subject**	
Nivel consulta	✓ General Terms	✓ Related Terms	✓ Specific Terms
Nivel labels	✗ General Terms	✓ Related Terms	✗ Specific Terms

Tareas Extraídas	- Obtain-all - Delete - paper - Obtain - Choose	- Obtain-all - Delete - Obtain - subject - Choose
Similitud entre tareas	Cosine Similarity: 87.5% Jaccard Similarity: 66.7%	
Dominio obtenido	- DeleteHealthInformationAction - DeleteAlarmAction - DeleteContactAction - Choose - DeleteMediaInformationAction - workshop paper - DeleteCalendarItemAction - paper - DeleteAction - DeleteReminderAction - Delete - expression - Obtain - DeleteFileAction - DeletedFromStorageTrigger - proceedings paper - Best applications paper award - DeleteTimerAction - Best student paper award - Obtain-all - conference paper - DeletedWebBookmarkTrigger - Award - Best paper award - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction	- DeleteHealthInformationAction - reference subject role - DeleteAlarmAction - DeleteContactAction - crossover population role - Choose - DeleteMediaInformationAction - participant under investigation role - technical replicate role - DeleteCalendarItemAction - DeleteAction - DeleteReminderAction - baseline participant role - Delete - biological replicate role - Obtain - subject - subject role - DeleteFileAction - DeletedFromStorageTrigger - clinical subject role - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction - patient role
Similitud entre dominios	Cosine Similarity: 19.2% Jaccard Similarity: 50%	

En esta ejecución podemos observar que los dominios resultantes son algo más dispares que en la anterior y como consecuencia la similitud es menor.

Tabla 8: Prueba con Delete paper y Delete subject en specific terms

BPMN	**CRS - Delete paper**	**CRS - Delete subject**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✗ General Terms ✗ Related Terms ✓ Specific Terms	
Tareas Extraídas	- Obtain-all - Delete - paper - Obtain - Choose	- Obtain-all - Delete - Obtain - subject - Choose
Similitud entre tareas	Cosine Similarity: 87.5% Jaccard Similarity: 66.7%	
Dominio obtenido	- DeleteHealthInformationAction - DeleteFromStorageAction - periodical literature - Thing - DeleteAlarmAction - DeleteContactAction - Choose - document - Version Class - periodical - DeleteCalendarItemAction - Document - paper - Deleted Entry - DeleteReminderAction - Delete - written work - expression - Obtain - DeleteFileAction - DeleteTimerAction - expression collection - Obtain-all - AccessPrivilege - Creative work - Entry Class - Item - Newspaper - newspaper - DeleteRemindAction - DeleteAction - DeleteWebBookmarkAction	- DeleteHealthInformationAction - reference subject role - DeleteFromStorageAction - human subject enrollment - Thing - selection - DeleteAlarmAction - DeleteContactAction - Choose - planned process - participant under investigation role - Version Class - Non-Information Resource - DeleteCalendarItemAction - Deleted Entry - DeleteReminderAction - Delete - subject term - Obtain - subject - DeleteFileAction - information content entity - subject discipline - DeleteTimerAction - Obtain-all - AccessPrivilege - Entry Class - Item - DeleteRemindAction - role - DeleteAction - DeleteWebBookmarkAction - Concept
Similitud entre dominios	Cosine Similarity: 47.2% Jaccard Similarity: 47.7%	

En esta ejecución, como en la anterior, podemos ver que los dominios son algo dispares y como consecuencia la similitud baja. Podemos llegar a la conclusión de que para estos modelos la mejor comparativa resulta en marcar General Terms en el nivel de búsqueda en las labels.

Utilizando BPMN distintos

Para las tres últimas pruebas de la comparativa simple usaremos dos modelos algo más diferentes, siendo estos, un modelo que describe el proceso de la obtención de todas las "tracks" (CRS_Get_all_tracks) y otro que describe el proceso de borrado de un "reviewer" (CRS - Delete reviewer).

Tabla 9: Prueba con Get all tracks y Delete reviewer en general terms

BPMN	**CRS_Get_all_tracks**	**CRS - Delete reviewer**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✓ General Terms ✗ Related Terms ✗ Specific Terms	
Tareas Extraídas	- Obtain-all - track	- Obtain-all - Delete - reviewer - Obtain - Choose
Similitud entre tareas	Cosine Similarity: 61.2% Jaccard Similarity: 16.7%	

Dominio obtenido	- Obtain-all - track	- DeleteHealthInformationAction - DeleteAlarmAction - DeleteContactAction - Choose - DeleteMediaInformationAction - DeleteCalendarItemAction - DeleteAction - DeleteReminderAction - Delete - reviewer - Obtain - DeleteFileAction - DeletedFromStorageTrigger - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction
Similitud entre dominios	Cosine Similarity: 36.1% Jaccard Similarity: 4.8%	

Para esta configuración podemos observar que los índices de similitud de coseno y de jaccard son bastante dispares, esto se debe a que el índice de jaccard es muy sensible a la diferencia de tamaños entre los conjuntos a comparar, además, con el nivel de búsqueda en labes para los General Terms no se han obtenido nuevos términos para el dominio del primer modelo, por tanto, los resultados son bastante bajos.

Tabla 10: Prueba con Get all tracks y Delete reviewer en related terms

BPMN	**CRS_Get_all_tracks**		**CRS - Delete reviewer**
Nivel consulta	✓ General Terms	✓ Related Terms	✓ Specific Terms
Nivel labels	✗ General Terms	✓ Related Terms	✗ Specific Terms

Tareas Extraídas	- Obtain-all - track	- Obtain-all - Delete - reviewer - Obtain - Choose
Similitud entre tareas	Cosine Similarity: 61.2% Jaccard Similarity: 16.7%	
Dominio obtenido	- sport facility - Obtain-all - track - racecourse - race track	- DeleteHealthInformationAction - DeleteAlarmAction - DeleteContactAction - Choose - DeleteMediaInformationAction - DeleteCalendarItemAction - DeleteAction - DeleteReminderAction - Delete - reviewer - Obtain - DeleteFileAction - DeletedFromStorageTrigger - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction
Similitud entre dominios	Cosine Similarity: 19.8% Jaccard Similarity: 4.2%	

Con esta configuración, seleccionando la búsqueda en labes a nivel de Related Terms, tampoco obtenemos un dominio del primer modelo muy amplio y observamos que los índices de similitud son bastante bajos.

Tabla 11: Prueba con Get all tracks y Delete reviewer en specific terms

BPMN	**CRS_Get_all_tracks**	**CRS - Delete reviewer**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✗ General Terms ✗ Related Terms ✓ Specific Terms	
Tareas Extraídas	- Obtain-all - track	- Obtain-all - Delete - reviewer - Obtain - Choose
Similitud entre tareas	Cosine Similarity: 61.2% Jaccard Similarity: 16.7%	
Dominio obtenido	- sport facility - Obtain-all - track - symbol - Non-Information Resource - centrally registered identifier symbol - Thing - information content entity - Type of railway track - architectural structure - courier tracking number - race track - Code - Concept	- DeleteHealthInformationAction - DeleteFromStorageAction - DeleteAlarmAction - DeleteContactAction - Choose - Version Class - DeleteCalendarItemAction - Deleted Entry - DeleteReminderAction - Delete - reviewer - Obtain - DeleteFileAction - DeleteTimerAction - Obtain-all - AccessPrivilege - Entry Class - Item - DeleteRemindAction - DeleteAction - DeleteWebBookmarkAction
Similitud entre dominios	Cosine Similarity: 8.7% Jaccard Similarity: 2.9%	

Con la configuración de esta ejecución se ha conseguido el dominio de mayor tamaño para el primer modelo y los índices de similitud más bajos. Podemos llegar a la conclusión de que se obtendrán mayores índices de similitud cuanto más generales sean los dominios obtenidos y obtendremos menores índices cuanto más específicos sean los dominios.

Multiple BPMN comparison

Utilizando BPMN similares

Se realizarán tres primeras pruebas utilizando modelos similares, concretamente:

- Conjunto A
  - CRS_Get_all_tracks: describe el proceso de obtener todas las "tracks"
  - CRS - Delete track: describe el proceso de eliminar una "track"
- Conjunto B
  - CRS - Create track: describe el proceso de creación de una "track"
  - CRS - Get track: describe el proceso de obtención de una "track"
  - CRS - Update track: describe el proceso de actualizar una "track"

Tabla 12: Prueba comparación multiple con modelos similares en general terms

Conjunto	**A**	**B**
BPMN	- **CRS_Get_all_tracks** - **CRS - Delete track**	- **CRS - Create track** - **CRS - Get track** - **CRS - Update track**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✓ General Terms ✗ Related Terms ✗ Specific Terms	

Tareas Extraídas	- Obtain-all - track - Choose - Obtain - Exclude	- track - Send - Obtain - Data - Create - Obtain-all - track - Obtain - Choose - Obtain-all - track - Obtain - data - Introduce - Modify - Choose
Similitud entre tareas	**CRS - Create track** Cosine Similarity: 47.4% Jaccard Similarity: 25%	**CRS - Get track** Cosine Similarity: 93.5% Jaccard Similarity: 80%	**CRS - Create track** Cosine Similarity: 78.3% Jaccard Similarity: 50%
Dominio obtenido de conjunto A	- Obtain-all - track - Choose - Obtain - Exclude		
Similitud entre dominio de A y tareas de B	**CRS - Create track** Cosine Similarity: 47.4% Jaccard Similarity: 25%	**CRS - Get track** Cosine Similarity: 93.5% Jaccard Similarity: 80%	**CRS - Create track** Cosine Similarity: 78.3% Jaccard Similarity: 50%

Para esta configuración unos índices de similitud bastante altos, sobretodo para los modelos **Get track** y **Create track.**

Tabla 13Prueba comparación multiple con modelos similares en related terms

Conjunto	**A**	**B**
BPMN	- **CRS_Get_all_tracks** - **CRS - Delete track**	- **CRS - Create track** - **CRS - Get track** - **CRS - Update track**
Nivel consulta	✓ General Terms	✓ Related Terms	✓ Specific Terms

<table>
<tr><td>Nivel labels</td><td colspan="3">✗ General Terms ✓ Related Terms ✗ Specific Terms</td></tr>
<tr><td>Tareas Extraídas</td><td colspan="2">- Obtain-all
- track
- Choose
- Obtain
- Exclude</td><td>- track
- Send
- Obtain
- Data
- Create
- Obtain-all
- track
- Obtain
- Choose
- Obtain-all
- track
- Obtain
- data
- Introduce
- Modify
- Choose</td></tr>
<tr><td>Similitud entre tareas</td><td>CRS - Create track
Cosine Similarity: 47.4%
Jaccard Similarity: 25%</td><td>CRS - Get track
Cosine Similarity: 93.5%
Jaccard Similarity: 80%</td><td>CRS - Create track
Cosine Similarity: 78.3%
Jaccard Similarity: 50%</td></tr>
<tr><td>Dominio obtenido de conjunto A</td><td colspan="3">- sport facility
- Obtain-all
- track
- racecourse
- Obtain
- race track
- Exclude
- Choose</td></tr>
<tr><td>Similitud entre dominio de A y tareas de B</td><td>CRS - Create track
Cosine Similarity: 46.2%
Jaccard Similarity: 18.2%</td><td>CRS - Get track
Cosine Similarity: 78.1%
Jaccard Similarity: 50%</td><td>CRS - Create track
Cosine Similarity: 65.3%
Jaccard Similarity: 36.4%</td></tr>
</table>

Marcando Related Terms en la búsqueda realizada en las labels podemos comprobar que obtenemos índices algo más bajos que buscando en General Terms pero todavía bastante altos.

Tabla 14Prueba comparación multiple con modelos similares en specific terms

Conjunto	**A**	**B**
BPMN	- **CRS_Get_all_tracks** - **CRS - Delete track**	- **CRS - Create track** - **CRS - Get track** - **CRS - Update track**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✗ General Terms ✗ Related Terms ✓ Specific Terms	
Tareas Extraídas	- Obtain-all - track - Choose - Obtain - Exclude	- track - Send - Obtain - Data - Create - Obtain-all - track - Obtain - Choose - Obtain-all - track - Obtain - data - Introduce - Modify - Choose
Similitud entre tareas	**CRS - Create track** Cosine Similarity: 47.4% Jaccard Similarity: 25%	**CRS - Get track** Cosine Similarity: 93.5% Jaccard Similarity: 80%	**CRS - Create track** Cosine Similarity: 78.3% Jaccard Similarity: 50%
Dominio obtenido de conjunto A	- Thing - Excluded tender - Type of railway track - architectural structure - courier tracking number - Code - Choose - track - Non-Information Resource - centrally registered identifier symbol - Tender - Offering - Obtain - information content entity - race track - Exclude - sport facility - Obtain-all - symbol - Concept	
Similitud entre dominio de A y tareas de B	**CRS - Create track** Cosine Similarity: 32.6% Jaccard Similarity: 8.7%	**CRS - Get track** Cosine Similarity: 49.6% Jaccard Similarity: 20%	**CRS - Create track** Cosine Similarity: 41.5% Jaccard Similarity: 17.4%

Para la última ejecución de este conjunto de pruebas podemos observar como han bajado los índices de similitud de coseno y de jaccard al haber obtenido un dominio que es más específico.

Utilizando BPMN distintos

Se realizarán el último conjunto de pruebas utilizando modelos más diferentes, concretamente:

- Conjunto A
  - CRS - Get all subjects: describe el proceso de obtener todas las "subjects"
  - CRS - Delete subject: describe el proceso de eliminar una "subject"
- Conjunto B
  - CRS - Create conference: describe el proceso de creación de una "conference"
  - CRS - Get conference: describe el proceso de obtención de una "conference"
  - CRS - Update conference: describe el proceso de actualizar una "conference"

Tabla 15: Prueba comparación multiple con modelos diferentes en general terms

Conjunto	**A**	**B**
BPMN	- **CRS - Get all subjects** - **CRS - Delete subject**	- **CRS - Create conference** - **CRS - Get conference** - **CRS - Update conference**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✓ General Terms ✗ Related Terms ✗ Specific Terms	

<table>
<tr><td>Tareas Extraídas</td><td>- Obtain-all
- Delete
- Obtain
- subject
- Choose</td><td>- Cancel
- Email
- Notify
- Facebook
- Send
- Obtain
- Twitter
- Conference
- Data
- Create
- Publish
- conference
- Obtain-all
- Obtain
- Choose
- Obtain-all
- Update
- Obtain
- data
- conference
- Introduce
- Choose</td></tr>
<tr><td>Similitud entre tareas</td><td colspan="2">CRS - Create track
Cosine Similarity: 21.3%
Jaccard Similarity: 6.7%

CRS - Get track
Cosine Similarity: 80.2%
Jaccard Similarity: 50%

CRS - Create track
Cosine Similarity: 67.1%
Jaccard Similarity: 33.3%</td></tr>
<tr><td>Dominio obtenido de conjunto A</td><td colspan="2">- DeleteHealthInformationAction
- reference subject role
- cohort role
- DeleteAlarmAction
- DeleteContactAction
- Choose
- DeleteMediaInformationAction
- DeleteCalendarItemAction
- DeleteAction
- DeleteReminderAction
- Delete
- biological replicate role
- Obtain
- subject
- subject role
- DeleteFileAction
- DeletedFromStorageTrigger
- clinical subject role
- DeleteTimerAction
- Obtain-all
- DeletedWebBookmarkTrigger
- DeleteRemindAction
- DeleteFromStorageAction
- DeletedTrigger
- DeleteWebBookmarkAction
- patient role</td></tr>
<tr><td>Similitud entre dominio de A y tareas de B</td><td colspan="2">CRS - Create track
Cosine Similarity: 6.7%
Jaccard Similarity: 2.8%

CRS - Get track
Cosine Similarity: 25.4%
Jaccard Similarity: 11.1%

CRS - Create track
Cosine Similarity: 21.2%
Jaccard Similarity: 10%</td></tr>
</table>

Podemos observar en esta ejecución como, aunque se haya marcado la opción de búsqueda en General Terms en las label no obtenemos buenos índices de similitud puesto que los modelos aunque tienen elementos comunes, pertenecen a dominios semánticos distintos.

Tabla 16: Prueba comparación multiple con modelos diferentes en related terms

Conjunto	**A**	**B**
BPMN	- **CRS - Get all subjects** - **CRS - Delete subject**	- **CRS - Create conference** - **CRS - Get conference** - **CRS - Update conference**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✗ General Terms ✓ Related Terms ✗ Specific Terms	
Tareas Extraídas	- Obtain-all - Delete - Obtain - subject - Choose	- Cancel - Email - Notify - Facebook - Send - Obtain - Twitter - Conference - Data - Create - Publish - conference - Obtain-all - Obtain - Choose - Obtain-all - Update - Obtain - data - conference - Introduce - Choose
Similitud entre tareas	**CRS - Create track** Cosine Similarity: 21.3% Jaccard Similarity: 6.7%	**CRS - Get track** Cosine Similarity: 80.2% Jaccard Similarity: 50%
	CRS - Create track Cosine Similarity: 67.1% Jaccard Similarity: 33.3%	

Dominio obtenido de conjunto A	- DeleteHealthInformationAction - reference subject role - DeleteAlarmAction - DeleteContactAction - crossover population role - Choose - DeleteMediaInformationAction - participant under investigation role - technical replicate role - DeleteCalendarItemAction - DeleteAction - DeleteReminderAction - baseline participant role - Delete - biological replicate role - Obtain - subject - subject role - DeleteFileAction - DeletedFromStorageTrigger - clinical subject role - DeleteTimerAction - Obtain-all - DeletedWebBookmarkTrigger - DeleteRemindAction - DeleteFromStorageAction - DeletedTrigger - DeleteWebBookmarkAction - patient role		
Similitud entre dominio de A y tareas de B	**CRS - Create track** Cosine Similarity: 5.2% Jaccard Similarity: 2.6%	**CRS - Get track** Cosine Similarity: 19.4% Jaccard Similarity: 10%	**CRS - Create track** Cosine Similarity: 16.2% Jaccard Similarity: 9.1%

En esta ejecución podemos observar como los índices de similitud de coseno y de jaccard son algo más bajos, debido a que estamos realizando la búsqueda en las labels de Related Terms y estamos perdiendo generalidad en el dominio obtenido.

Tabla 17: Prueba comparación multiple con modelos diferentes en specific terms

Conjunto	**A**	**B**
BPMN	- **CRS - Get all subjects** - **CRS - Delete subject**	- **CRS - Create conference** - **CRS - Get conference**

		- **CRS - Update conference**
Nivel consulta	✓ General Terms ✓ Related Terms ✓ Specific Terms	
Nivel labels	✗ General Terms ✗ Related Terms ✓ Specific Terms	
Tareas Extraídas	- Obtain-all - Delete - Obtain - subject - Choose	- Cancel - Email - Notify - Facebook - Send - Obtain - Twitter - Conference - Data - Create - Publish - conference - Obtain-all - Obtain - Choose - Obtain-all - Update - Obtain - data - conference - Introduce - Choose
Similitud entre tareas	**CRS - Create track** Cosine Similarity: 21.3% Jaccard Similarity: 6.7%	**CRS - Get track** Cosine Similarity: 80.2% Jaccard Similarity: 50% **CRS - Create track** Cosine Similarity: 67.1% Jaccard Similarity: 33.3%

Dominio obtenido de conjunto A	- DeleteHealthInformationAction - reference subject role - DeleteFromStorageAction - human subject enrollment - Thing - selection - DeleteAlarmAction - DeleteContactAction - Choose - planned process - participant under investigation role - Version Class - Non-Information Resource - DeleteCalendarItemAction - Deleted Entry - DeleteReminderAction - Delete - subject term - Obtain - subject - DeleteFileAction - information content entity - subject discipline - DeleteTimerAction - Obtain-all - AccessPrivilege - Entry Class - Item - DeleteRemindAction - role - DeleteAction - DeleteWebBookmarkAction - Concept		
Similitud entre dominio de A y tareas de B	**CRS - Create track** Cosine Similarity: 6.6% Jaccard Similarity: 2.3%	**CRS - Get track** Cosine Similarity: 24.9% Jaccard Similarity: 8.8%	**CRS - Create track** Cosine Similarity: 20.8% Jaccard Similarity: 8.1%

Para la última ejecución, marcando el nivel de búsquedas en labels para Specific Terms podemos comprobar como los índices de similitud de coseno y jaccard se mantienen bajos, indicando que los modelos pertenecen a dominios semánticos distintos.

Analizando las pruebas podemos llegar a la conclusión de que, con frecuencia, cuanto más general es el dominio obtenido mayores resultan los índices de similitud de coseno y de jaccard, y por el contrario cuanto más específico se vuelve el dominio más bajos resultan los índices de similitud.

CAPÍTULO 5: MANUAL DE INSTALACIÓN

En este apartado se tratarán los pasos a seguir para poder instalar y usar la aplicación web. Será necesaria una memoria mínima de 4GB de RAM y 10GB de espacio libre en disco.

Se puede descargar el proyecto desde el siguiente enlace:

WordDomain 2.0. Herramienta semiautomática para la generación de diccionarios

Python

El primer paso será instalar una versión de python 3.7 o superior, si ya tiene instalada una versión que cumpla los requisitos puede ir directamente a la sección de librerías.

Nos dirigiremos a la página oficial de python y allí descargaremos la versión más reciente que haya hasta el momento y que se adecue a nuestro sistema operativo.

Al abrir el instalador nos aparecerá la siguiente pantalla:

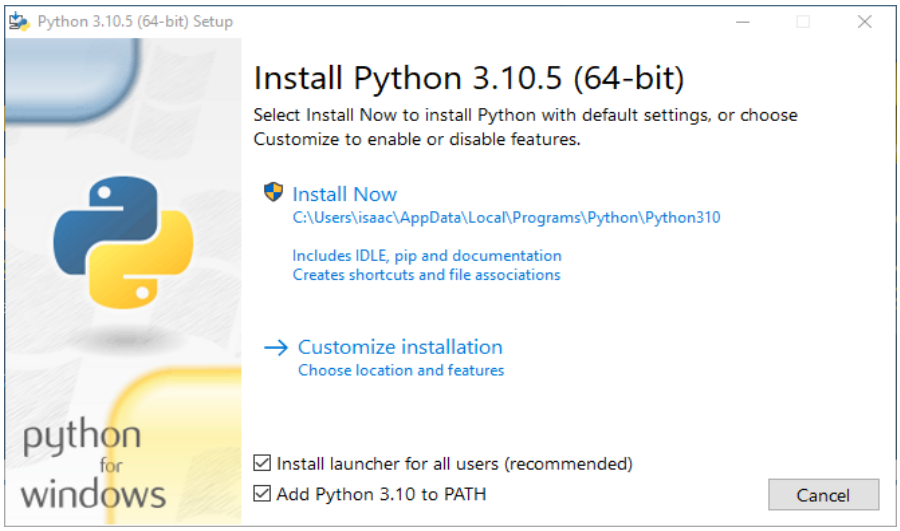

Figura 19: Pantalla principal instalador python

Antes de pulsar en “Install Now” es muy importante marcar la casilla “Add Python 3.10 to PATH” puesto que si no deberemos hacerlo manualmente después de la instalación.

Cuando acabe el proceso nos aparecerá esta pantalla en la que deberemos pulsar la opción “Disable path lenght limit” y ya tendremos instalado correctamente Python en nuestro equipo.

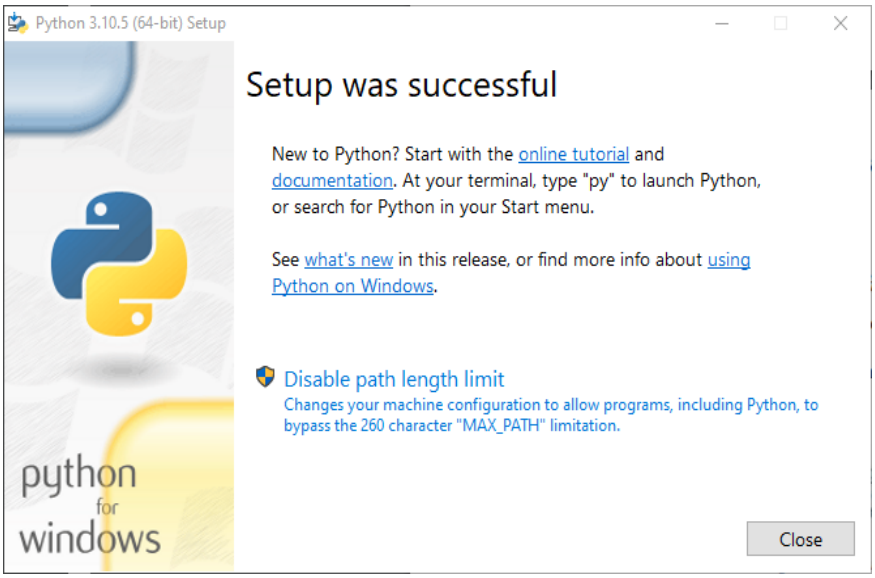

Figura 20: Pantalla final instalador python

Librerías

Una vez instalado python deberemos instalar las siguientes librerías necesarias para el correcto funcionamiento de la aplicación.

La instalación es sencilla, en un terminal (si estamos en windows lo podremos abrir fácilmente escribiendo cmd en la barra de búsqueda de windows) deberemos escribir **pip install nombreLibreria** e inmediatamente comenzará el proceso de instalación de la librería correspondiente.

Las librerías necesarias son las siguientes:

- django==2.2
- django-import-export
- nltk
- SPARQLWrapper
- isodate
- pyparsing
- rdflib
- datetime
- celery_progress
- djangorestframework
- dict2xml
- django-crispy-forms

- django-extensions
- django-schema-graph
- django-spaghetti-and-meatballs
- numpy
- sklearn

Pasos finales

Teniendo ya instalado python y las librerías mencionadas comenzamos con los últimos pasos antes de poder utilizar la aplicación.

Django BBDD

Deberemos de iniciar la base de datos que proporciona Django y que es usada en el proyecto. Para ello debemos abrir un terminal en la ruta donde se encuentre el fichero **manage.py** de nuestro proyecto. Este archivo se encuentra dentro de la carpeta **worddomain** de la aplicación. Para abrir fácilmente un terminal apuntando a esa dirección podemos ir directamente a la carpeta deseada a través del explorador de archivos y una vez en nuestro destino, bastará con escribir **cmd** en la barra de direcciones del explorador y pulsar intro.

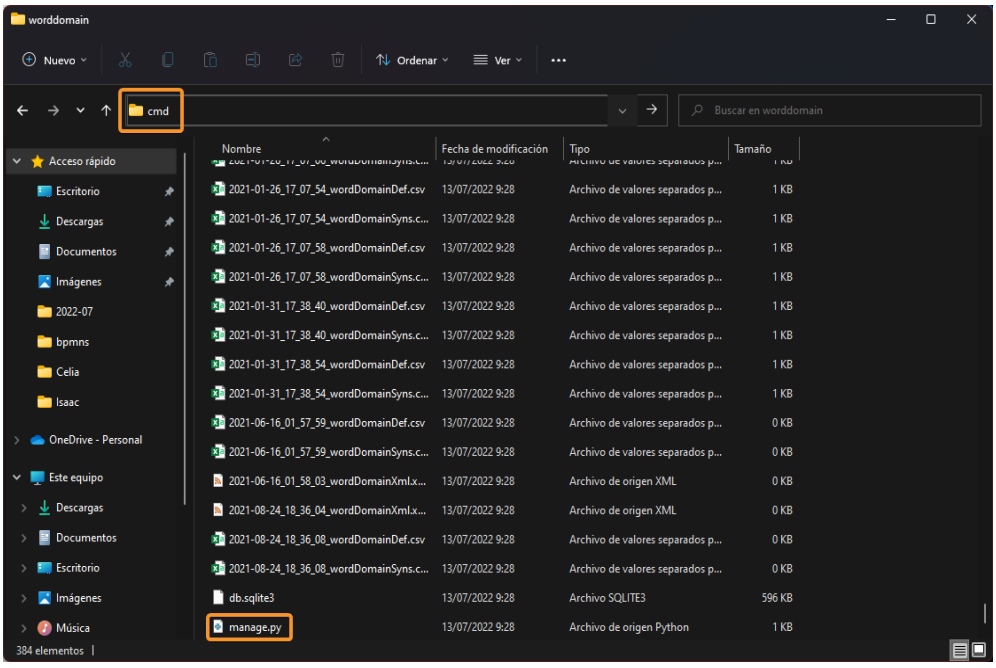

Figura 21: Ruta manage.py

Una vez abierto el terminal en la ruta de manage.py escribiremos los siguientes comandos uno a uno:

- python manage.py makemigrations
- python manage.py migrate

Dichos comandos crearán la base de datos en base a los modelos que hayan sido definidos en Django.

NLTK

Para acabar de instalar este paquete deberemos abrir una terminal y escribir **python** en ella para abrir correctamente una terminal de python.

En la nueva terminal escribiremos y ejecutaremos los siguientes comandos:

- import nltk
- nltk.download()

Una vez ejecutado se abrirá la siguiente pantalla:

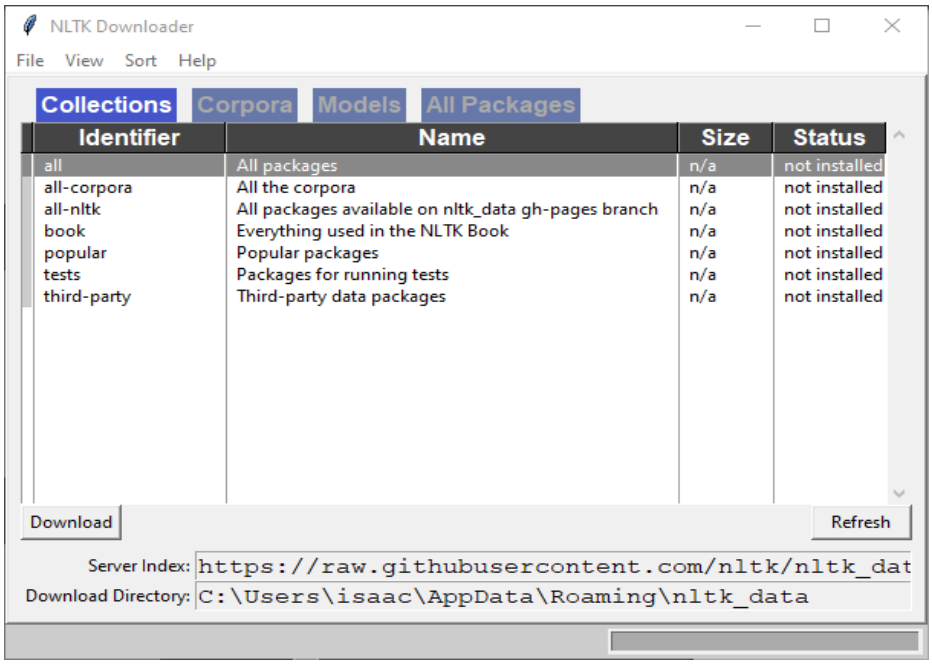

Figura 22: Instalador NLTK

Aquí, nos aseguramos de que está marcada la opción “all” y pulsamos el botón “Download”. Comenzará el proceso de instalación que puede tardar unos minutos, cuando todos los paquetes cambien su color a verde la instalación habrá terminado y podremos cerrar la ventana.

Inicio de la aplicación

Disponemos ya de todo lo necesario para iniciar la aplicación y poder usarla. Para iniciarla abriremos un terminal en la ruta del archivo **manage.py** igual que en el apartado Django BBDD y escribiremos el siguiente comando:

- python manage.py runserver

Dicho comando lanzará el servidor local en el que se aloja la aplicación para poder ser utilizada. Tan solo tendremos que ir a la siguiente url desde cualquier navegador:

http://127.0.0.1:8000/

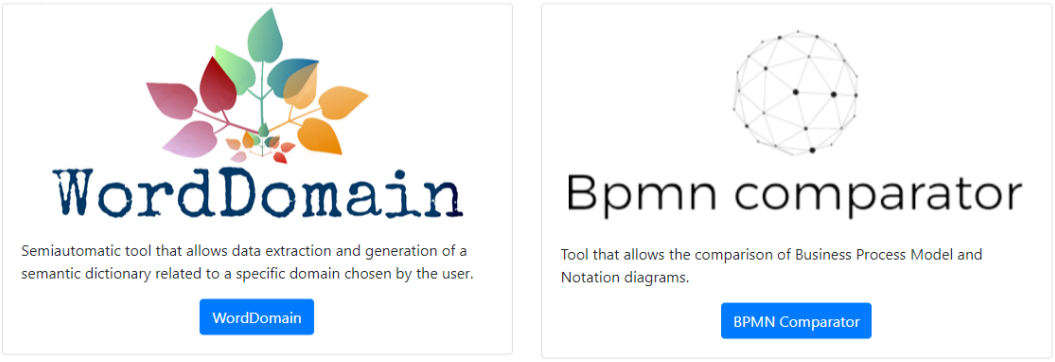

Figura 23: Página principal aplicación

Con el servidor activo, aparte de usar la aplicación también tendremos acceso a la interfaz que proporciona Django para visualizar y gestionar la base de datos.

Para acceder a esta característica tendremos que crear primero un super usuario. Desde la terminal que usamos antes en la ruta de **manage.py** escribiremos el siguiente comando:

- python manage.py createsuperuser

Esto permitirá la creación de nuestro perfil de super usuario. Una vez creado podremos acceder al entorno de administrador de Django desde la siguiente url:

http://127.0.0.1:8000/admin/

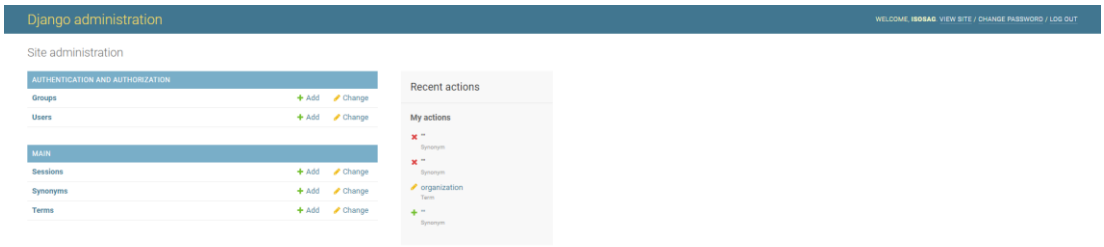

Figura 24: Página administración Django

Desde aquí tendremos acceso a los términos y sinónimos que hemos ido generando conforme usamos la aplicación.

Troubleshooting

En esta sección se informará sobre errores comunes ocurrentes durante la instalación y cómo resolverlos:

- ImportError: cannot import name 'url' from 'django.conf.urls'

  Este error ocurre si no tenemos la versión correcta de django. La aplicación ha sido desarrollada en django 2.2, si se tiene instalada una versión superior algunas funciones darán error por encontrarse obsoletas.

  Para solucionarlo basta con ejecutar nuevamente **pip install django==2.2**

- ModuleNotFoundError: No module named 'nombreLibreria'

  Este error ocurre cuando alguna librería no ha sido instalada. Para solucionarlo solo hay que instalar la librería que falte con **pip install nombreLibreria**

CAPÍTULO 6: CONCLUSIONES

En Ingeniería del Software en ocasiones es necesario el uso de una base léxica para utilizarla como base semántica en ciertos trabajos que requieren una base o dominio común.

Este trabajo planteó dos objetivos principales, la mejora del trabajo previo "ASISTENTE PARA LA GENERACIÓN AUTOMÁTICA DE DICCIONARIOS SEMÁNTICOS DE DOMINIO" y su ampliación en el uso del matching de modelos BPMN.

Como se ha explicado en la documentación, ambos objetivos se han llevado a cabo en este trabajo, mediante los objetivos específicos que se definieron inicialmente. Estos objetivos específicos fueron para **WordDomain**:

- Obtener un conjunto de términos inicial mayor.

  Este objetivo se ha logrado con la inclusión de un mayor número de ontologías para la búsqueda del conjunto inicial, concretamente se han añadido DBpedia y LOV, además de hasta dos ontologías personalizadas seleccionadas por el usuario.

- Edición del diccionario semántico antes de la exportación.

  Este objetivo ha sido cumplido al permitir al usuario que añada, edite o elimine tanto los términos como sus definiciones y sinónimos antes de realizar la exportación.

- Refactorización del código base de WordDomain.

  Este objetivo se ha cumplido al simplificar partes del código ya existentes y al utilizar en más profundidad la base de datos proporcionada por Django.

- Mejora de la interfaz de usuario.

  Este objetivo ha sido llevado a cabo y se ha modificado la interfaz aportando más legibilidad y claridad para el usuario.

- Opción para la exportación del diccionario en formato xml.

  Este objetivo ha sido cumplido añadiendo dicha opción junto con la ya existente de exportación a formato csv.

Estos objetivos específicos fueron para **BPMN Compare**:

- Extracción de las tareas que componen los diagramas BPMN

  Este objetivo ha sido cumplido, se ha conseguido la extracción de tareas de los modelos de negocio para su posterior procesado.

- Generación de diccionario semántico

  Este objetivo ha sido logrado al implementar una API de WordDomain que permite la generación del diccionario semántico de manera automática.

- Comparativa de los diccionarios semánticos

  Este objetivo ha sido llevado a cabo implementando dos algoritmos diferentes de comparativa, siendo los elegidos el Índice de similitud de Jaccard y la similitud de coseno.

También se han realizado una serie de pruebas para verificar que se habían cumplido todos los objetivos propuestos, las pruebas se han dividido en pruebas de funcionamiento de WordDomain definiendo distintas consultas y pruebas sobre matching semántico entre varios modelos BPMN.

Observando el apartado de pruebas de WordDomain podemos llegar a la conclusión de que al aumentar el conjunto inicial de términos podremos obtener un diccionario semántico más amplio y completo.

Analizando el apartado de pruebas de BPMN Comparator llegamos a la conclusión de que cuanto más general sean los dominios a comparar mayores resultarán los índices de similitud, por el contrario, cuanto más específicos sean los dominios peores resultados obtendremos.

CAPÍTULO 7: TRABAJOS FUTUROS

Como posibles trabajos futuros que puedan añadir valor al presente proyecto se proponen las siguientes ideas:

- Mejorar el tiempo de búsquedas en ontologías, por ejemplo, observando cual sería el rendimiento con las ontologías almacenadas de manera local.
- Modificar BPMN Comparator para que acepte modelos diseñados en otras herramientas además de Signavio
- Mejorar la interfaz de BPMN Comparator para que muestre de manera clara de que ontología procede cada término del dominio resultante y para el caso de la comparación múltiple, marcar de que modelo procede cada tarea extraida.
- Implementar un instalador para simplificar el proceso de instalación y evitar posibles errores por parte del usuario.

CAPÍTULO 9: REFERENCIAS BIBLIOGRÁFICAS

[1] Sosa2020. Encarna Sosa Sánchez, Pedro J. Clemente, José M. Conejero Manzano and Alvaro E. Prieto. Business Process Execution From the Alignment Between Business Processes and Web Services: A Semantic and Model-Driven Modernization Process. IEEE Access. Vol 8. Pág: 93346-93368. May 2020. DOI: 10.1109/ACCESS.2020.2993883. https://ieeexplore.ieee.org/document/9091182

[2] Discovering Healthcare Processes from Natural Language Documents: a case study on COVID-19. Moreira Bohnenberger, Nicolas Mauro; Ceolin Schmitt, Alessandra; and Thom, Lucinéia Heloisa, "Discovering Healthcare Processes from Natural Language Documents: a case study on COVID-19" (2021). PACIS 2021 Proceedings. 175. https://aisel.aisnet.org/pacis2021/175

[3] BPMN About the Business Process Model and Notation. OMG | Object Management Group [en línea]. Enero de 2014 [consultado el 12 de julio de 2022]. Disponible en: https://www.omg.org/spec/BPMN/

[4] P. Pietsch, K. Müller, and B. Rumpe, "Model matching challenge: Benchmarks for ecore and bpmn diagrams," Softwaretechnik-Trends, 2013.

[5] O. C. Tibermacine and C. Foudil, "A practical approach to the measurement of similarity between wsdl-basedweb services," in 6ème Conférence francophone sur les Architectures Logicielles, CAL 2012, Montpellier, France, 30-21 Mai 2012, ser. RNTI, E. Exposito and M. Zouari, Eds., vol. L-7. Hermann-Éditions, 2012, pp. 3–18. [Online]. Available: http://editions-rnti.fr/?inprocid=1002037

[6] ASTORGA HURTADO, Celia. ASISTENTE PARA LA GENERACIÓN AUTOMÁTICA DE DICCIONARIOS SEMÁNTICOS DE DOMINIO. Trabajo de fin de grado, Escuela Politécnica, Universidad de Extremadura. [consultado el 12 de julio de 2022].

[7] Linked Open Vocabularies (LOV). [consultado el 12 de julio de 2022]. Disponible en: https://lov.linkeddata.es/dataset/lov/

[8] DBpedia Association. [consultado el 12 de julio de 2022]. Disponible en: https://www.dbpedia.org

[9] Ontobee [consultado el 12 de julio de 2022]. Disponible en: https://ontobee.org

[10] Django. [consultado el 12 de julio de 2022]. Disponible en: https://www.djangoproject.com

[11] About the Business Process Model And Notation Specification Version 2.0. OMG | Object Management Group. [consultado el 14 de julio de 2022]. Disponible en: https://www.omg.org/spec/BPMN/2.0/

[12] World Wide Web (WWW) [consultado el 12 de julio de 2022]. Disponible en: https://es.wikipedia.org/wiki/World_Wide_Web#Enlaces_externos

[13] Web of Documents. Daniel Janus's blog. [consultado el 12 de julio de 2022]. Disponible en: https://blog.danieljanus.pl/2019/10/07/web-of-documents/

[14] Identificador de recursos uniforme (URI) [consultado el 12 de julio de 2022]. Disponible en: https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme

[15] SPARQL Query Language. World Wide Web Consortium (W3C). [consultado el 12 de julio de 2022]. Disponible en: https://www.w3.org/TR/sparql11-query/

[16] OWL Web Ontology Language Guide. World Wide Web Consortium (W3C) [en línea]. [consultado el 12 de julio de 2022]. Disponible en: https://www.w3.org/TR/owl-guide/

[17] Vocabularies - W3C. World Wide Web Consortium (W3C) [consultado el 12 de julio de 2022]. Disponible en: https://www.w3.org/standards/semanticweb/ontology

[18] Algoritmo de similaridad de Jaccard. [consultado el 14 de julio de 2022]. Disponible en: https://www.grapheverywhere.com/algoritmo-de-similaridad-de-jaccard/

[19] Algoritmo de similitud de coseno. [consultado el 14 de julio de 2022]. Disponible en: https://www.grapheverywhere.com/algoritmo-de-similitud-de-coseno/

[20] SQLite. [consultado el 14 de julio de 2022]. Disponible en: https://www.sqlite.org/index.html

[21] Python. [consultado el 14 de julio de 2022]. Disponible en: https://www.python.org/about/

[22] Bootstrap. [consultado el 14 de julio de 2022]. Disponible en: https://getbootstrap.com

[23] WordNet | A Lexical Database for English. WordNet. [consultado el 14 de julio de 2022]. Disponible en: https://wordnet.princeton.edu

[24] Javascript. [consultado el 14 de julio de 2022]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript