UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Análisis y visualización de recursos educativos abiertos contenidos en sitios open course ware

TRABAJO DE FIN DE TITULACIÓN

AUTOR: Vire Quezada, Silvana Cecilia

DIRECTOR: Piedra Pullaguari, Nelson Oswaldo, Ing.

LOJA – ECUADOR

2014

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Ingeniero.

Nelson Oswaldo Piedra Pullaguari

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de fin de titulación: Análisis y visualización de recursos educativos abiertos contenidos en sitios open course ware realizado por Silvana Cecilia Vire Quezada, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, 17 de septiembre de 2014

f) ……..

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Silvana Cecilia Vire Quezada declaro ser autora del presente trabajo de fin de titulación: Análisis y visualización de recursos educativos abiertos contenidos en sitios open course ware, de la Titulación de Ingenieria en Sistemas Informáticos y Computación, siendo el Ing. Nelson Oswaldo Piedra Pullaguari director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos cientificos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, academico o institucional (operativo) de la Universidad””

f. ……

Autor: Silvana Cecilia Vire Quezada

Cedula: 1104616709

DEDICATORIA

El presente trabajo, paso final para alcanzar una de mis metas académicas, lo dedico a las personas más importantes en mi vida, mi amada familia.

A mi Dios toda la gloria y honra sea para ti, gracias por bendecir mi vida e iluminar cada paso que he dado para llegar a cumplir uno de tus propósitos, a mis padres especialmente a mi madre Fannicita eres mi principal ejemplo de perseverancia, constancia y fuerza incansable. A mi querido esposo Darío gracias por asumir los últimos pasos en mi carrera, por tu apoyo, tu consideración, y sobre todo por tu comprensión eres parte importante en este logro. A mi padre Máximo gracias por tus consejos y palabras de apoyo todo mi respeto y cariño hacia ti. A mis hermanos y hermanas Soraya, Max, Julissa, Stalin, Andrea gracias por confiar en mí por su amor y apoyo en todo momento. A mis corazoncitos pequeños mis sobrinos especialmente a Kevin, Joao, Daniel, Luisa, Julissa, Sofía por su ayuda en el cuidado de mi hijo cuando debía cumplir con mis obligaciones académicas, Dios los bendiga por sus lindas acciones. A mí cuñado Daniel y mi suegra Crucita gracias por su cariño, por abrirme sus brazos y quererme como una hija más y por ayudarme en el cuidado de mi hijo durante el desarrollo de mí proyecto de tesis. Al hombrecito dueño de mi corazón mi hijo Matías eres mi inspiración para ser mejor cada día te amo.

Este título lo dedico con todo mi cariño para ustedes, por darme su amor y apoyo incondicional durante mi carrera profesional, por ser mis razones para levantarme luego de cada caída, por comprenderme cuando elegía mis obligaciones académicas sobre los momentos familiares y sobre todo por ser parte de mi vida Dios los bendiga y proteja siempre.

Silvana Cecilia

AGRADECIMIENTO

A la Universidad Técnica Particular Técnica de Loja por darme todas las herramientas necesarias para mi educación profesional y poder obtener mi título de tercer nivel. A los docentes de la Escuela de Sistemas Informáticos y Computación por compartir sus conocimientos, gracias por sus consejos y exigencias. Al Ingeniero Juan Carlos Morocho por sus consejos y enseñanzas en Gestión Productiva y Beca de Responsabilidad, gracias por sus palabras de apoyo. Al Ingeniero Nelson Piedra por dirigir mi proyecto de tesis.

A mis amigos y amigas especialmente a Rommel Landívar, Rommel Agusto, Jhonny, Carlitos, David, María José, Felipe, Pablo, Christian por compartir conmigo este proceso de aprendizaje por todas las malas noches que tuvimos para terminar con los proyectos, por el apoyo que nos dimos durante ellas y por el crecimiento personal que compartimos.

A mi familia gracias por los ánimos, por la confianza depositada en mí, y por toda la paciencia que han tenido para compartir conmigo este sueño hecho realidad.

Silvana Cecilia

INDICE DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

El análisis y visualización de recursos educativos abiertos contenidos en sitios OpenCourseWare involucra tres procesos importantes los cuales son: a) extracción y procesamiento de información de OER's/OCW: encargado de obtención de data (porción de texto, tokens, tags, tokens representativos, entidades) de los OER's en idioma inglés y en formato PDF, esta data es sometida a tareas de procesamiento del lenguaje natural como la desambiguación del sentido de las palabras, también se realiza la creación de relaciones entre OER's, y finalmente la data obtenida se almacena en un triplestore en formato RDF; b) enriquecimiento de datos sobre OER's/OCW a través del enlace con la nube de datos enlazados abiertos: este proceso se enfoca en incrementar la data previamente obtenida usando principalmente a las palabras comunes entre OER's, para lo cual se utiliza el dataset más importante y robusto como lo es DBpedia; como último proceso tenemos c) visualización de información de OER/OCW: este proceso se encarga de implementar la visualización en base a las relaciones creadas entre los recursos educativos abiertos, se utiliza herramientas disponibles basadas en visualización de redes como JavaInfovisToolkit.

PALABRAS CLAVES: OER, OCW, PLN, extracción, enriquecimiento, visualización, desambiguación, RDF, triplestore, palabras, representativas, pdf, idioma, inglés, NLTK, JIT, Dbpedia, Python, relaciones, procesamiento, recursos, educativos, abiertos, lenguaje, natural, texto, tags, tokens, representativos, entidades, synsets, palabras, comunes.

ABSTRACT

The analysis and visualization of open educational resources in OpenCourseWare sites involves three important processes which are a) extraction and processing of information OER's/OCW: responsible for obtaining data (portion of text, tokens, tags, representatives words, entities) of OER's in English and in PDF format, this data is it subjected to tasks of natural language processing as disambiguation of word’s sense, creating relationships between OER 's is also performed , and finally the data obtained is stored in a triplestore in RDF format; b) enrichment of data on OER's/OCW through liaison with the cloud data linked open: this process focuses on increasing the data previously obtained using mostly common words between OER's, for which the largest and strongest dataset is used DBpedia; As a last process we have c) information display of OER/OCW: This process is responsible for implementing the display based on the relations existing between the open educational resources, available tools based on network visualization are used as JavaInfovisToolkit .

KEYWORDS : OER, OCW, PLN, extraction, enrichment, visualization, disambiguation, RDF, triplestore, words, representative, pdf, language, English, NLTK, JIT, DBpedia, Python, relations, process, resources, education, open, language, natural, text, tags, tokens, representatives, entities, synsets, words, commons.

INTRODUCCIÓN

Las tecnologías de la información (TIC's) se han convertido en una de las principales herramientas para la educación debido a que nos permiten el acceso al conocimiento de una manera rápida. A nivel mundial las Universidades se están preocupando por implementar nuevas metodologías de enseñanza – aprendizaje las mismas que contemplen material didáctico que fortalezca y desarrolle el conocimiento de un tema determinado, ademas que sea de libre acceso para todos los usuarios. Este material didáctico también se lo denomina Recurso Educativo Abierto o por sus siglas en inglés OER (Open Education Resource)

Los OER se pueden encontrar en sitios Open Course Ware (OCW) o en otros repositorios; abarcan diversos temas educativos como física, análisis matemático, química, biología, multimedia, entre otros; también se encuentran disponibles en varios tipos de archivos como multimedia, hojas de cálculo, presentaciones power point, documentos de texto, podcast, entre otros, por lo tanto existen varios formatos como pdf, doc, ppt, xls, txt, flv, odt; en varios idiomas como: Inglés, Español, Catalán, Chino Mandarín, Italiano, entre otros.

El presente trabajo se enfoca en la extracción y procesamiento del contenido de los recursos educativos abiertos pertenecientes a sitios OCW, específicamente de documentos de texto en formato PDF y en idioma inglés, con la finalidad de analizarlos y relacionarlos.

Es aquí donde radica la importancia del presente proyecto, pues se necesita mejorar la interoperabilidad entre OER's y crear relaciones con otros recursos educativos abiertos para fortalecer, mejorar, y facilitar el acceso al conocimiento.

El presente trabajo consta de 4 capítulos, el primer capítulo abarca el Estado del Arte que nos brinda un enfoque general sobre los inicios de la web, evolución semántica, introducción a los OER's, procesamiento del lenguaje natural, proceso de desambiguación del sentido de las palabras, visualización de redes, además nos da un enfoque al análisis y visualización de los recursos educativos abiertos y finalmente las herramientas que se utilizan para el desarrollo del proyecto.

El segundo capítulo trata sobre Definición de Marco de Trabajo nos presenta una descripción de los datos a utilizar y sobre cada proceso involucrado en el desarrollo del proyecto.

El tercer capítulo trata sobre el Proceso de Extracción y Procesamiento de Información de OER's/OCW es el primer proceso, encargado de obtener los recursos educativos abiertos, extraer la data de estos recursos, procesar esta data y crear las relaciones entre los OER's.

El cuarto capítulo abarca el Enriquecimiento de datos sobre OER/OCW a través de la nube de datos enlazados abiertos (Linked Data), se establece utilizar el dataset de Dbpedia mediante consultas Sparql para enriquecer los datos de los recursos educativos abiertos.

El quinto capítulo trata sobre la Visualización de Información de OER's, se utiliza visualización de redes porque permite al usuario final entender y observar las relaciones entre los elementos con facilidad, características que la convierten en la opción más idónea para el presente proyecto.

Finalmente tenemos las conclusiones y las recomendaciones del proyecto que son tan importantes y dan una pauta para nuevos proyectos.

Se espera que la información que contiene este proyecto sea de ayuda para fortalecer los procesos de investigación y desarrollo del conocimiento, así como para establecer nuevos enfoques sobre el procesamiento del lenguaje natural.

OBJETIVOS

GENERAL

Analizar y visualizar recursos educativos abiertos pertenecientes a sitios OpenCourseWare.

ESPECÍFICOS

- Extraer metadatos de los recursos educativos abiertos que se encuentren en formato .pdf.
- Aplicar métodos de desambiguación a la metadata obtenida de los recursos educativos abiertos.
- Encontrar relaciones entre los recursos educativos abiertos disponibles en sitios OpenCourseWare.
- Enriquecer los datos obtenidos de los OER mediante el enlace a Linked Open Data Cloud.
- Visualizar las relaciones entre los recursos educativos abiertos pertenecientes a sitios OCW.

CAPITULO 1: ESTADO DEL ARTE

1. Historia y evolución de la web.

Es importante recordar la historia y evolución de la web y como principal referencia se ha tomado a (Berners-Lee, 1996) que afirma: “La Web se define simplemente como el universo de información accesible desde la red global. Se trata de un espacio abstracto con el cual las personas pueden interactuar, actualmente está poblado por páginas interconectadas que contienen texto, imágenes, animaciones y vídeos. Su existencia marca el final de una era de incompatibilidades frustrantes y debilitantes entre sistemas informáticos.”

La “World Wide Web” fue creada aproximadamente a finales de los años 80, y a lo largo de los años 90 como un proyecto para un laboratorio propuesto por Tim Berners Lee en el 2007, para difundir investigaciones e ideas a lo largo de la organización y a través de la red. El objetivo de la web es ser un espacio de información compartida a través del cual las personas (y máquinas) puedan comunicarse. (Berners-Lee, 1996).

1.1. Web de datos.

La necesidad de ir resolviendo problemas como la calidad de la información, acceso a publicaciones científicas, obtención de información correcta, autorizar al usuario para que haga uso de documentos privativos, y si a estas razones le agregamos como punto extra la competencia de compañías como Google para apoderarse del mercado e ir mejorando su tecnología, han dado lugar al surgimiento de algunos cambios que dieron paso a una nueva era de la Web.

Usando como referencia a (W3C Oficina Española, 2009) la web de datos o Linked Data se refiere a la web de los datos enlazados. Linked Data permite pasar de una Web en la que los recursos son documentos HTML, a una Web de Datos Enlazados expresados en RDF, un lenguaje para representar significados sobre recursos, en la que agentes de software pueden explotar estos datos de forma automática (recopilándolos, agregándolos, interpretándolos, publicándolos, mezclándolos, etc.), potenciados por vocabularios y ontologías que usan especificaciones explícitas y formales de una conceptualización compartida.

La Web de Datos cuenta con información vinculada sobre música, autores, países, libros, organizaciones públicas, privadas, ubicaciones geográficas, universidades, entre otros. Siendo capaz de filtrar la información que es relevante de aquella que no lo es mediante el uso de tecnologías que han emergido en base a las necesidades existentes que serán mencionadas a detalle en los siguientes apartados; además ofrece resultados precisos y vinculados a búsquedas establecidas por el usuario. (Gruber, 1993)

Como mencionan (Piedra, Tovar, López, Chicaiza, & Martinez, 2011) “La Web de Datos plantea un potencial considerable para el sector educativo, tanto en uso como en la contribución de que su información esté disponible a través de sets de datos vinculados. En particular, existe un potencial de beneficios para los materiales y recursos educativos ofrecidos libre y abiertamente para que cualquiera pueda usar, mezclar, distribuir”.

1.2. Web semántica.

(Piedra, Tovar, López, Chicaiza, & Martinez, 2011) mencionan que “la visión de la Web Semántica, defendida por Sir Tim Berners-Lee, está construida entorno al concepto de la “Web de Datos” (o LinkedData), que significa pasar de una Web actualmente centrada en Documentos a una Web centra en Datos. En esta visión la Web, los datos y sus relaciones son fundamentales.”

La Web Semántica no se trata únicamente de la publicación de datos en la Web, sino que estos se pueden vincular a otros, de forma que las personas y las máquinas puedan explorar la web de los datos, pudiendo llegar a información relacionada que se hace referencia desde otros datos iniciales. (W3C Oficina Española, 2009)

Nos menciona (Castells, 2005) que “Si estudiamos la web actual se asemeja a un grafo formado por nodos del mismo tipo, e hiperenlaces entre ellos igualmente indiferenciados. Por ejemplo, no se hace distinción entre la un blog profesional de una temática concreta y el portal de una tienda on-line, como tampoco se distinguen explícitamente los enlaces de publicidad externa de la tienda con los de los productos concretos. Por el contrario en la web semántica cada nodo (recurso) tiene un tipo

(profesor, tienda, pintor, libro), y los arcos representan relaciones explícitamente diferenciadas (pintor – obra, profesor – departamento, libro – editorial).” En la siguiente figura podemos ver un ejemplo

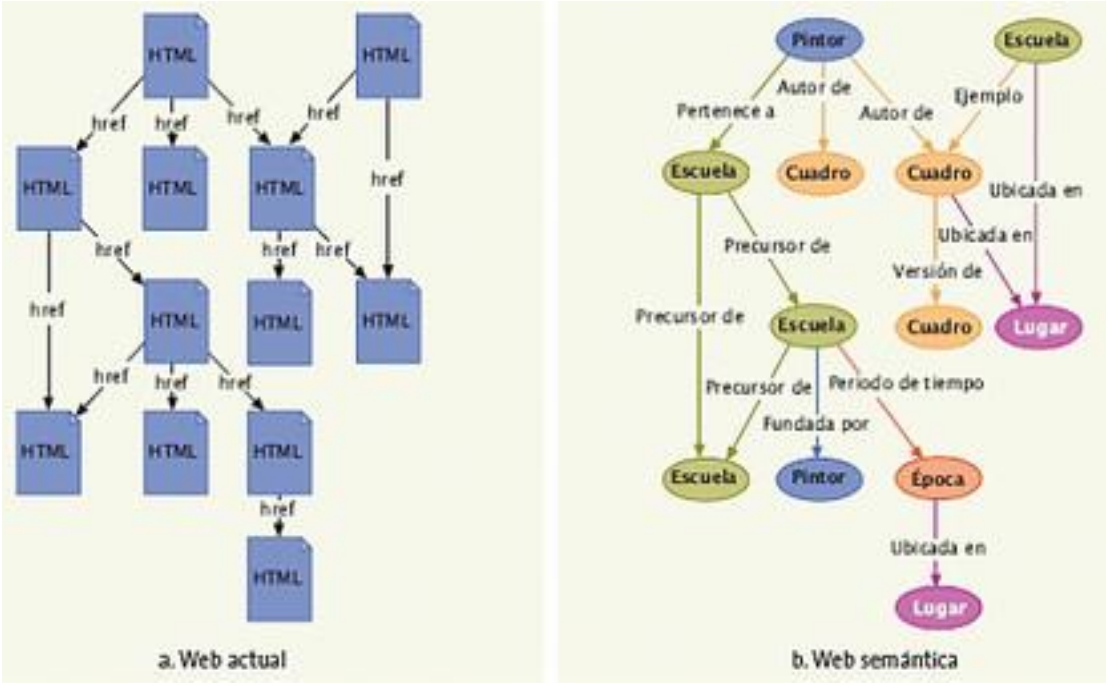


Figura 1: Ejemplo gráfico entre Web actual y Web semántica.
Fuente: (Fonseca, Hierro, & Romo, 2009)

Lograr que la Web Semántica sea una potente herramienta de acceso al conocimiento implica construir estructuras bien formadas que soporten y permitan una correcta publicación y acceso a los datos vinculados. Además es necesario seguir y cumplir ciertas normas o principios establecidos, que permitirán alcanzar un nivel coherente en la información que será publicada.

1.2.1. Estructura de la web semántica

(Berners-Lee, 2000) Presenta una estructura basada en lenguajes, estándares, plantillas, reglas y otros componentes que hacen de la Web Semántica toda una base de conocimiento. En la siguiente figura se puede apreciar la estructura de la Web Semántica y seguidamente un detalle de cada una de ellas.

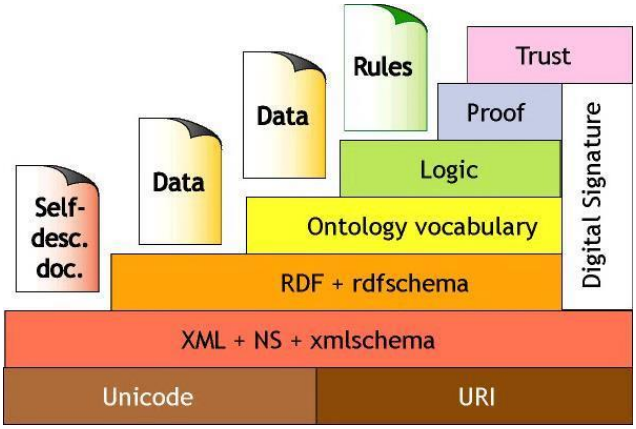


Figura 2: Estructura de la Web Semántica
Fuente: (Berners-Lee, Architecture, 2000)

- **Unicode - URI:** Unicode es un sistema de codificación que asigna un número único para identificar cada carácter sin importar la plataforma, programa ni idioma. Este es compatible con la mayoría de sistemas operativos y con todos los exploradores actuales, además es un requerimiento para estándares modernos como XML. URI proporciona un nombre para identificar de manera única a los distintos recursos de la Web.
- **XML + NS + xmlschema:** En esta capa se integran tres tecnologías que hacen posible la comunicación entre agentes.

- **XML** ofrece un formato común para intercambiar documentos de una forma estructurada, como árboles de etiquetas con atributos.

- **XML Schema** es uno de estos lenguajes para definir su estructura, donde se describen de antemano las estructuras y tipos de datos utilizados.
- **NS** proporciona un método para cualificar elementos y atributos de nombres usados en documentos XML asociándolos con espacios de nombre identificados por referencias URI's.

- **RDF + RDFSchema:** Es un lenguaje simple mediante el cual definimos sentencias en un formato con tres elementos: sujeto, predicado y objeto.

- **RDF Schema** provee un vocabulario definido sobre RDF que permite el modelo de objetos con una semántica claramente definida. Esta capa no solo ofrece descripción de los datos, sino también cierta información semántica. Ambos corresponden a las anotaciones de la información llamados metadatos.

- **OWL:** Es uno de los lenguajes de ontologías más extendidos por la Web Semántica. Este estándar W3C fue diseñado para ser compatible con estándares web existentes. Ontology Web Language añade más vocabulario para describir propiedades, clases, relaciones entre clases, cardinalidad, igualdad, características de propiedades, clases enumeradas, etc.

- **Logic, Proof, Trust, Digital Signature:** Las capas Logic (Lógica) y Proof (Pruebas) son encargadas de aplicar reglas de inferencia con sus pruebas respectivas. En la capa Trust (Confianza) encontramos agentes que realizan un análisis completo y comprobación de las fuentes de información de la Web Semántica. Finalmente Digital Signatura (Firma Digital) garantiza que la información ofrecida proviene de sitios confiables. (Berners-Lee, Architecture, 2000)

Cada una de las tecnologías que forman parte de la estructura de la Web Semántica aportan para dar sostenibilidad, coherencia, establecer un orden en los datos que serán publicados y por ende serán parte del conocimiento abierto y accesible a todos los usuarios; y sin dejar de lado el tema de seguridad punto importante y necesario para el contenido que conformará la Web Semántica.

Según (Gruber, 1993) estas tecnologías aportarán a la facilidad y reducción de costos: de captura y almacenamiento de datos, distribución de la información, comunicación y principalmente a generar verdadera inteligencia colectiva. Para Tom Gruber este último constituye el rol principal de la Web Semántica, “crear valor a partir de los datos”, e indica dos formas de hacerlo:

- **Primero**, aumentar datos estructurados a las contribuciones de los usuarios y

- **Segundo**, la apertura para compartir datos entre aplicaciones distintas y heterogéneas, gracias a los estándares e infraestructura indicada anteriormente.

La Web Semántica como se mencionó permite obtener información estructurada en base a una relación establecida entre datos, indicándonos con claridad cuál es la razón por la que se corresponden entre sí (dato, relación, dato), cabe mencionar que esta estructura es entendible tanto para los humanos como para las máquinas, marcando una gran diferencia entre la Web actual; sin embargo uno de los puntos en contra de la Web Semántica es el poco conocimiento que el usuario final tiene de ella, existe un gran porcentaje que no la utiliza , o en muchos de los hace uso de ella pero no le da los créditos que merece debido a la falta de promoción.

Recordar cómo ha ido evolucionando la web desde sus inicios hasta la actualidad nos permite ir planteando una visión en base a las nuevas necesidades del usuario, y ser capaces de crear soluciones oportunas e innovadoras que se guíen en la contribución, compartición, calidad, y estructura de los datos.

2. Linked data.

Linked Data permite construir la Web de los datos, en una gran base de datos interconectados y distribuidos en la Web. Los datos se vinculan y se exploran de una forma similar a la utilizada para vincular los documentos HTML. (W3C Oficina Española, 2009)

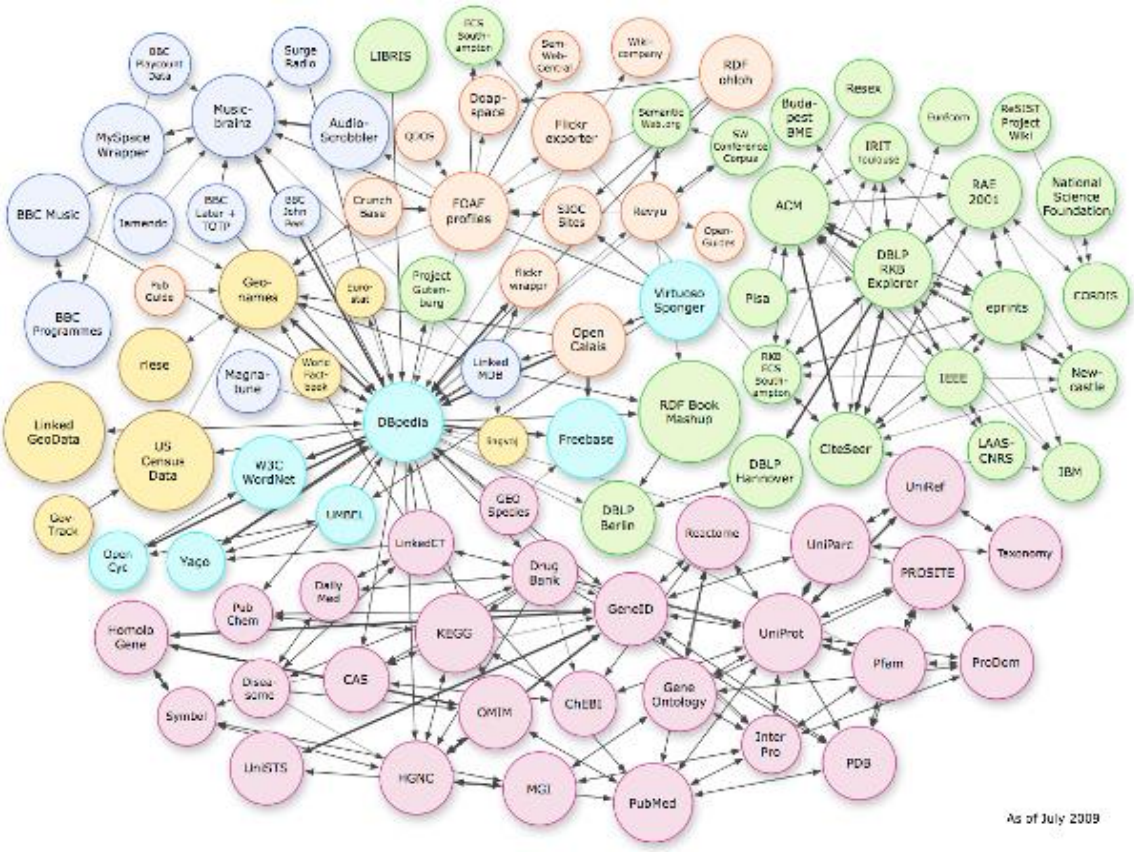


Figura 3: Este grafo representa a distintos conjuntos de datos de diversos tipos, organizados mediante colores por dominios. Estos conjuntos de datos están conectados entre sí de forma que componen la “Nube de Linked Data” o “Nube de Datos Enlazados”.
Fuente: (W3C Oficina Española, 2009)

Tim Berners-Lee presentó en TED 2009 (Technology Entertainment and Design) una conferencia, en la que redefinió los principios de Linked Open Data presentándolos como tres reglas que se resumen en:

- *Asignar a todas las cosas conceptuales nombres que comienzan con http*
- *Obtener información importante de retorno a partir de los nombres; y*
- *La información obtenida debe contener relaciones.*

Se hace énfasis en esta conferencia en publicar los datos lo más pronto posible. (Berners-Lee, 2009)

2.1. Principios de linked data.

La importancia y facilidad que Sir Tim Berners-Lee da a los principios de Linked Data permiten e invitan al usuario a contribuir y publicar sus datos, aspecto tan relevante para ir construyendo la Web de datos, conforme a nuestras necesidades y en prioridad a la información que se maneja. Aplicar estos principios esenciales y obligatorios permite impulsar el crecimiento de la Web.

Utilizando como referencia el artículo publicado por Tim Berners-Lee en julio del 2006, es conveniente complementar estos principios o reglas básicas con un mayor detalle:

- **Utilice URIs como nombre para las cosas:** Al nombrar los conceptos mediante URIs, estamos ofreciendo una abstracción del lenguaje natural para así evitar ambigüedades y brindar una forma estándar y univoca para referirnos a cualquier recurso.
- **Usar HTTP URIs para que la gente pueda buscar los nombres:** Se debe usar URIs sobre HTTP para asegurar que cualquier recurso pueda ser buscado y accedido en la Web. Se debe tener en cuenta que los URIs no son sólo direcciones, son identificadores de los recursos.
- **Cuando alguien busca un URI, proveer información útil utilizando las normas (RDF. SPARQL):** Cuando se busca y accede a un recurso identificado mediante una URI HTTP, se debe obtener información útil sobre dicho recurso representada mediante descripciones estándares en RDF. Se pretende que para cualquier conjunto de datos o vocabulario, se ofrezca información relativa a la información que representa.
- **Incluye enlaces a otros URI, para que puedan descubrir más cosas:** Esta regla es necesaria para enlazar datos que se encuentran en la Web, de tal manera que no se queden aislados y así poder compartir la información con otras fuentes externas y que otros sitios puedan enlazar sus propios datos de la misma forma que se hace con los enlaces en HTML.

Generalmente se otorga una igualdad en conceptos entre URL y URI, es necesario establecer la diferencia entre URI (Uniform Resource Identifier) es una secuencia compacta de caracteres que identifica un recurso abstracto o físico teniendo una sintaxis y un proceso para resolver estas. (Berners-Lee, 2005); Y URL (Uniform

Resource Locator) el mismo que sirve como medio de localización y acceso a los recursos en el mundo de Internet.

2.2. Beneficios de linked data.

Luego de haber revisado historia, estructura y principios sobre Linked Data llegamos al punto de preguntarnos cuales son los beneficios que otorga a sus usuarios como: investigadores, desarrolladores y usuarios finales. En relación a esto y usando como referencia lo que menciona (W3C Oficina Española, 2009) podemos citar algunos de los beneficios de los datos enlazados:

- La web de datos puede ser rastreada por los enlaces RDF.
- Contiene mecanismos de acceso único y estandarizado.
- Facilita el trabajo de los motores de búsqueda.
- Accede mediante navegadores de datos genéricos.
- Vinculación de los datos de diferentes fuentes.

2.3. Proceso para la publicación de datos.

Publicar lo datos implica seguir un proceso que permita de una forma organizada aportar con información valiosa y estructurada a la Web Semántica, para tomar en consideración la publicación de datos es necesario plantearse las siguientes preguntas como cita (W3C Oficina Española, 2010) ¿Con que cantidad de Datos se cuenta? ¿Cómo se almacenaran los Datos? ¿Cómo actualmente se encuentran los datos a vincular?

Mediante estos enfoques nos damos cuenta que es necesario saber que vamos a publicar conociendo; nuestro ámbito de trabajo, hacia quienes van dirigidos estos datos y quienes van a reutilizar nuestra información. Luego de pensar en estos ambientes podemos tomar en consideración las respuestas a las preguntas citadas anteriormente. (W3C Oficina Española, 2010)

Boris Villazón menciona en el artículo “Methodological Guidelines for Govermment Linked Data” que es necesario conocer el proceso de publicación de Datos tiene un ciclo de vida el mismo que se especifica a continuación:

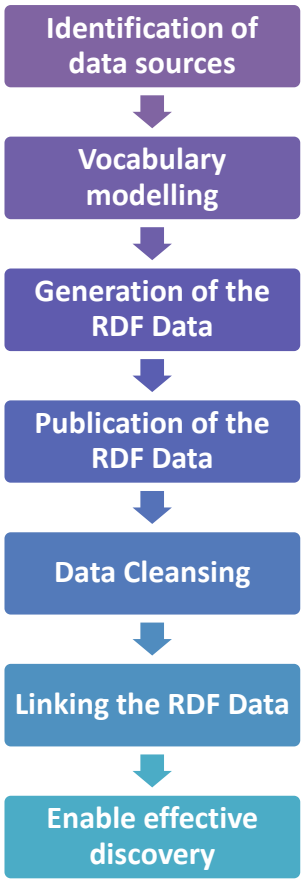


Figura 4: Proceso de publicación de datos
Fuente: (Villazón, Vilches, Corcho, & Gómez, 2011)

- **Identificación de fuentes de datos:** Identificar y seleccionar las fuentes de datos que queremos publicar. Aportar a un proyecto que ya tiene su data publicada o tomar data particular para posteriormente ser publicada. Se responde preguntas como: ¿Dónde están los datos?, ¿en qué formato?, ¿qué tipo de repositorio?, es decir se necesita conocer el entorno de trabajo a utilizar.
- **Desarrollo de vocabularios:** se recomienda la reutilización de vocabularios existente para un rápido desarrollo de ontologías, esto permite una reducción de tiempo, esfuerzo y recursos.

- **Generar datos RDF:** se toman las fuentes de datos identificadas previamente y se la transforma a formato RDF utilizando los vocabularios establecidos, de esta forma se cumple con los principios de Linked Data.
- **Publicación de datos RDF:** Se necesita almacenar y publicar la data en triplestore (base de datos para el almacenamiento en tripletas RDF).
- **Limpieza de datos:** es importante encontrar los posibles errores. Según (Hogan, Harth, Passant, & Polleres, 2010) identificar los errores comunes como por ejemplo; errores de accesibilidad, tipos de datos malformados o incompatibles, o términos de vocabularios mal definidos.
- **Enlazar datos RDF:** Se aplica el cuarto principio de Linked Data "incluir enlaces con otras URIS" para lograr interconectar con datasets externos.
- **Habilitar un descubrimiento efectivo:** Habilitar el efectivo descubrimiento y sincronización de los dataset publicados, mantener la data actualizada y en lo posible que buscadores como Google o Sindice1 sean capaces de localizarla. (Villazón, Vilches, Corcho, & Gómez, 2011)

Al momento de crear una aplicación de software es necesario el uso de metodologías de desarrollo (tradicionales o ágiles) que permiten alcanzar los objetivos propuestos, cuya función es guiar a todos los stakeholders en los pasos que deben seguir para obtener un proyecto de calidad; de igual forma y orden la Web Semántica que está orientada en el enlace de datos, merece tener una metodología que indique cuales son los pasos a seguir para publicar datos relevantes, que sean de una buena calidad y así poder contribuir a la Web.

2.4. Tecnologías de la web semántica.

Para poseer una correcta operación de los datos se necesita de tecnologías que hagan posible la web semántica, como lenguajes para la representación de ontologías, lenguajes de consulta, entornos de desarrollo, módulos de gestión de ontologías (almacenamiento, acceso, actualización), módulos de visualización; es importante mencionar cada una de ellas y los conceptos que manejan para tener un panorama

[1] http://sindice.com/main/submit

claro de su funcionamiento. La fuente de información principal en este apartado es W3C en su publicación del año 1999.

2.4.1. Representación de la información.

Siguiendo los conceptos establecidos por la (W3C Oficina Española, 2009) y menciona que la representación de la información en la web se hace posible gracias al lenguaje RDF (Resource Description Framework), es una base para procesar metadatos; proporciona interoperabilidad entre aplicaciones que intercambian información legible por máquinas en la Web. RDF destaca por la facilidad para habilitar el procesamiento automatizado de los recursos Web.

El objetivo general de RDF es definir un mecanismo para describir recursos que no cree ninguna asunción sobre un dominio de aplicación particular ni defina (a priori) la semántica de algún dominio de aplicación. La definición del mecanismo debe ser neutral con respecto al dominio, sin embargo el mecanismo debe ser adecuado para describir información sobre cualquier dominio. El modelo de datos básico consiste en tres tipos de objetos:

- **Recursos:** todas las cosas descritas por expresiones RDF se denominan recursos. Un recurso puede ser una parte de una página Web, una colección completa de páginas web, un sitio Web completo, un libro impreso. Los recursos se designan siempre por URIs.

- **Propiedades:** una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades.

- **Sentencias:** Un recurso especifico junto con una propiedad denominada, más el valor de dicha propiedad para ese recurso es una sentencia RDF.

Estas tres partes se denominan respectivamente sujeto, predicado y objeto. El objeto de una sentencia (el valor de la propiedad) puede ser otro recurso o puede ser un literal; es decir, un recurso (especificado por una URI) o una cadena simple de caracteres u otros tipos de datos primitivos definidos por XML.

Consideremos como ejemplo la sentencia: *Ora Lassila es el creador [autor] del recurso http://www.w3.org/Home/Lassila.*

Esta sentencia comprende las siguientes partes:

Tabla 1: Sentencia de Ejemplo

SUJETO (RECURSO)	http://www.w3.org/Home/Lassila
PREDICADO (PROPIEDAD)	Creator
OBJETO (LITERAL)	"Ora Lassila"

Fuente: (W3C, 1999)

Para representar gráficamente esta sentencia se usa diagrama de nodos y arcos. En estos gráficos; los nodos (óvalos) representan recursos y los arcos representan propiedades denominadas. Los nodos que representan cadenas de literales pueden dibujarse como rectángulos. La sentencia citada anteriormente se representaría gráficamente como:

Figura 5: Representación en diagrama de nodos de la sentencia.
Fuente: (W3C, 1999)

Nota: La dirección de la flecha es importante. El arco siempre empieza en el sujeto y apunta hacia el objeto de la sentencia.

RDF necesita también una sintaxis concreta para crear e intercambiar metadatos. Esta especificación de RDF utiliza XML [Lenguaje de Marcado extensible]. (W3C, 1999)

La sentencia antes mencionada como ejemplo se la representa en XML/RDF así:

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Figura 6: Sentencia representada en XML/RDF.
Fuente: (W3C, 1999)

2.4.2. Explotación de datos RDF.

Como menciona en su presentación (Qaissi, 2009) al hablar de un formato de sociabilización no tan solo es hablar de poder enlazar los datos en la web y poder ser hallados por medio de buscadores, es hablar de cómo recuperar los datos en la web tomando en consideración necesidades específicas como:

- Los datos en RDF no servirían de nada si no se pueden utilizar.
- Los lenguajes de la Web Semántica necesitan interactuar con los datos almacenados en la "base de datos" RDF.
- Necesidad parecida al lenguaje SQL de bases de datos relacionales.

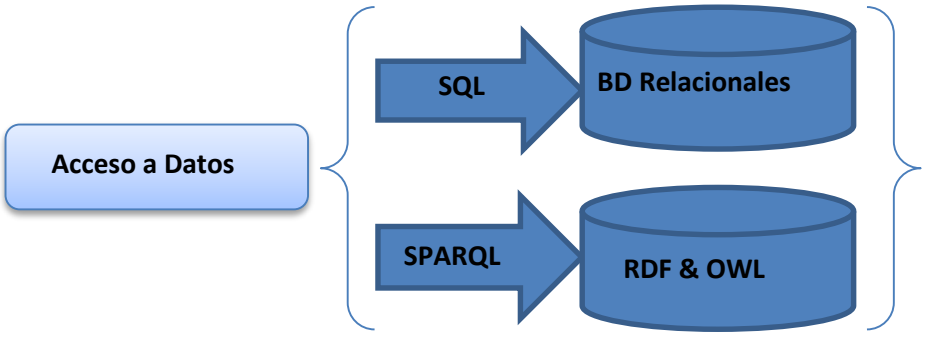


Figura 7: Acceso a Datos.
Fuente: (Qaissi, 2009)

SPARQL (Simple Protocol and RDF Query Language) es una lenguaje de consulta del ámbito de la Web Semántica de W3C. En otras palabras define la sintaxis y la semántica necesarias para una expresión de consulta sobre un grafo RDF y las diferentes formas de resultados obtenidos.

Su misión es devolver todas las tripletas o componentes solicitados basándose en la comparación de una tripleta pasada como parámetro de la consulta (grafo básico) con todas las tripletas que componen el grafo RDF. Cabe recalcar que sus sintaxis son similares al estándar SQL de bases de datos relacionales. (Qaissi, 2009)

Las consultas SPARQL cubren tres objetivos:

- *Extraer información en forma de URIs y literales.*
- *Extraer sub-estructuras RDF*
- *Construir nuevas estructuras RDF partiendo de resultados de consultas.* (Corcho & Gómez, 2010)

Al tener la información almacenada en formato RDF es posible realizar consultas utilizando SPARQL, a continuación se presenta un ejemplo de este tipo de consultas. (Rico, 2013)

(DBpedia, 2013) Nos presenta un ejemplo detallado de una sentencia Sparql: “Se desea buscar el nombre de grupos de música heavy de los años 80”, la consulta sería:

```
PREFIX esdbpp: <http://es.dbpedia.org/property/>
PREFIX esdbpr: <http://es.dbpedia.org/resource/>
SELECT ?grupo
WHERE{
  ?grupo  rdf:type  dbpedia-owl:MusicalArtist .
  ?grupo  dbpedia-owl:activeYearsStartYear ?inicio .
  ?grupo  dbpedia-owl:activeYearsEndYear   ?fin .
  ?grupo  esdbpp:estilo  esdbpr:Heavy_metal .
  FILTER ((?inicio > "1980-01-01T00:00:00Z" ^^ xsd:dateTime && ?inicio < "1990-01-
01T00:00:00Z"^^xsd:dateTime) || (?fin > "1980-01-01T00:00:00Z"^^xsd:dateTime && ?fin <
"1990-01-01T00:00:00Z"^^xsd:dateTime ) || (?inicio < "1980-01-01T00:00:00Z"^^xsd:dateTime
&& ?fin > "1990-01-01T00:00:00Z"^^xsd:dateTime ) )
}ORDER BY DESC(?inicio) LIMIT 10
```

El resultado es:

Tabla 2: Resultado de la consulta hacia DBpedia

GRUPO
http://es.dbpedia.org/resource/Bonham
http://es.dbpedia.org/resource/Argus_(banda)

http://es.dbpedia.org/resource/Logos_(banda)
http://es.dbpedia.org/resource/Shotgun_Messiah
http://es.dbpedia.org/resource/Reverend
http://es.dbpedia.org/resource/Vago_(banda_argentina)
http://es.dbpedia.org/resource/Seo_Taiji
http://es.dbpedia.org/resource/Mother_Love_Bone
http://es.dbpedia.org/resource/Saigon_Kick
http://es.dbpedia.org/resource/Tad

Fuente: (DBpedia, 2013)

Gracias a la Web y de una manera progresiva se ha logrado eliminar barreras de comunicación, comercio, y acceso a la información; beneficiando directamente a los usuarios en cada una de las actividades que realizan, otorgándole el poder de descubrir nuevas cosas. Si nos remontamos hacia una década atrás en nuestro entorno era una moda tener una cuenta de email, o ser parte de una sala de chat; se denomina un lujo tener servicio de Internet; actualmente y gracias al avance tecnológico tener estos servicios se ha convertido en una necesidad, debido a diversos factores sociales, económicos, educativos, entre otros. Es meritorio mencionar que los esfuerzos realizados por entidades públicas, privadas, sin fin de lucro, han aportado para que la brecha informática se disminuya.

Linked Data aporta una perspectiva completamente nueva y estructurada de cómo manejar la web usando como línea base el conocimiento, mejorar la capacidad de búsqueda de contenido, brindarle al usuario información útil y referente a su necesidad, y no podemos dejar de lado la importancia de mantener un nivel de estructura formal, de igual forma cumplir con los principios establecidos, seguir la metodología de publicación de datos, todo esto en beneficio del contenido que será agregado para finalmente ser parte de esta nueva ventana de conocimiento.

3. Recursos educativos abiertos.

(Unesco, 2011) Menciona “El concepto de Recursos Educativos Abiertos (OER por sus siglas en inglés Open Education Resource) describe a los recursos educativos (incluyendo mapas, materiales para cursos, vídeos, aplicaciones multimedia, podcast, libros, y otros materiales que sean diseñados su uso en la enseñanza y aprendizaje)

que sean abiertos y disponibles para el uso de educadores y estudiantes, sin la necesidad de pagar por licencias de uso o regalías.

OER surgió como un concepto con gran potencial para soportar las transformaciones en la educación. Su valor educativo radica en la idea de utilizar los recursos como un método integral de comunicación del plan de estudios en los cursos de educación (es decir aprendizaje basado en recursos), su poder de transformación reside en la facilidad con la que tales recursos, una vez digitalizados, se pueden compartir a través de Internet.

Un OER es simplemente un recurso educativo que incorpora una licencia que facilita la reutilización y, potencialmente la adaptación, sin solicitar permiso al propietario del copyright.

3.1. Elementos fundamentales de los OER

Los Recursos Educativos Abiertos son materiales y contenidos educativos ofrecidos libre y abiertamente para que cualquiera los pueda usar, se deben considerar los siguientes elementos fundamentales para los OER's:

- **Contenidos de aprendizaje:** cursos completos, materiales para cursos, módulos, contenidos, objetos de aprendizaje, programas educativos completos, publicaciones, etc.
- **Herramientas:** Software para la creación, entrega, uso y mejora del contenido de aprendizaje abierto, incluyendo búsqueda y organización de contenido, sistemas de gestión del aprendizaje (LMS), herramientas de desarrollo de contenidos, y comunidades de aprendizaje en línea.
- **Recursos de implementación:** Licencias de derechos de autor que promuevan la publicación abierta de materiales, principios de diseño y adaptación local de contenido. Frecuentemente, los recursos educativos abiertos están distribuidos bajo una licencia Creative Commons".
- **Enlaces externos:** Portal de OLCOS, el proyecto Open eLearning Content Observatory Services, que pretende construir un observatorio y centro de

información para la promoción del uso, creación y difusión de recursos educativos abiertos.

Desde el punto de vista de (López, Piedra, Sancho, Soto, & Tovar, 2012) “Los Recursos Educativos en Abierto proporcionan de esta manera un entorno de aprendizaje con un grado mayor de libertad para la innovación educativa, pilar básico de la reforma de los nuevos títulos de grado en Europa. A los alumnos les permite ver una medida de aquello a lo que se van a enfrentar en sus estudios para hacerlo con mayor garantía. A las personas en proceso de autoformación les puede resultar un medio eficaz para orientar su proceso de desarrollo personal a lo lago de la vida, y también para crecer profesionalmente en conocimientos.

Pero donde los Recursos Educativos Abiertos son especialmente útiles es en regiones donde la pobreza, la localización remota o un bajo nivel de bienestar complican el acceso al aprendizaje formal OCW proporciona un medio para aquellos profesores que dedican su vida a la diseminación del conocimiento, aun superando las preocupaciones legítimas del esfuerzo que se deben emplear.”

Un recurso educativo abierto se convierte en parte fundamental en el aprendizaje de un estudiante, por lo tanto debe contener información relevante sobre una materia o asignatura en particular. Al ser libre brinda al estudiante la capacidad de aumentar sus conocimientos con información de otras universidades.

3.2. Calidad de los recursos educativos abiertos.

Los OCW han permitido que el conocimiento sea accesible mediante la publicación del material educativo que se utiliza para impartir las cátedras universitarias, estos recursos educativos están disponibles para todo usuario con deseos de conocimiento. Sin embargo el tema de la calidad en todo ámbito es un plus que garantiza el trabajo realizado, en los recursos educativos abiertos no puede ser de otra manera.

Como referencia se utiliza la publicación “A Basic Guide to Open Educational Resources” que menciona: “este tema es posiblemente, el reflejo de una idea profundamente arraigada de la educación, materiales como “publicaciones”, cuya

calidad es controlada por los editores de la educación, esta noción ha sido y sigue siendo válida, refleja parcialmente la compresión del alcance y la diversidad de los materiales educativos utilizados en muchas enseñanzas y contextos de aprendizaje, también refleja una delegación falsa de la responsabilidad para la calidad de un tercero.

Esta mentalidad se desplaza en el espacio de los REA en la forma de un supuesto tácito de que uno o más organismos especializados deberían tener plenamente la responsabilidad de garantizar que los REA compartidos en repositorios en línea son de una alta calidad.

Además de esto, siendo prácticamente imposible, enmascarar la realidad de que la definición de calidad es subjetiva y depende contextualmente.

En última instancia, la responsabilidad de asegurar la calidad de los REA que se utilicen en la enseñanza y ambiente de aprendizaje recaerá en la institución, el programa / curso, coordinadores y educadores individuales responsables de la entrega de la educación como han hecho siempre al prescribir los libros de texto, elegir un vídeo o elegir el plan de clase de otra persona, estos agentes son los que retienen la última responsabilidad para la elección de los materiales que abierta y /o de propiedad de usar.

Por lo tanto la “calidad de los REA” dependerá de los recursos que eligen usar, como deciden adaptarlas para que sean contextualmente relevantes, y como integrarlos en las actividades de los diferentes tipos de enseñanza y aprendizaje.

Esta tarea de aseguramiento de la calidad se ha visto complicada por la explosión de disposición contenido (tanto abierto como propietario). Esto es tanto una ventaja ya que reduce la probabilidad de necesitar para desarrollar nuevos contenidos y una maldición ya que exige mayor habilidad de nivel de búsqueda de información, selección, adaptación y evaluación, tomando en cuenta que las instituciones comparten más contenido educativo en línea, van asegurarse de que le contenido refleje una buena institución y por lo tanto podrá invertir en la mejora de su calidad antes de que este a disposición en los repositorios”. (Unesco, 2011)

Para facilitar el aprendizaje sobre un tema específico los OER's se almacenan en diferentes tipos como hojas de cálculo, vídeos, documentos de texto, presentaciones, fotografías, audio, etc. Para el desarrollo del presente trabajo nos enfocaremos en los documentos de texto, específicamente en documentos en formato PDF.

4. Open course ware.

Open Course Ware es una publicación abierta y disponible para todo aquel usuario que necesita de fuentes de información confiables, actuales, científicas, y por supuesto libres; el siguiente extracto del informe sobre "Aplicación de tecnologías web emergentes para el estudio del impacto de repositorios OpenCourseWare españoles y latinoamericanos en la educación superior" nos da una perspectiva clara sobre este tema, a continuación cito.

"El modelo para compartir conocimiento en abierto que tomó el nombre de OpenCourseWare (OCW) fue propuesto por el Massachusetts Institute of Technology (MIT), quien fue el primero en implantarlo en su propia institución. Nació en el año 2000. Fue la estrategia recomendada por un comité de expertos ante el creciente impacto de Internet en la Educación Superior y, en particular, como alternativa al entonces emergente campo de la educación a distancia. Fue la estrategia recomendada por un comité de expertos ante el creciente impacto de Internet en la Educación Superior y, en particular, como alternativa al entonces emergente campo de la educación a distancia.

El éxito de esta iniciativa se alcanzó cuando el modelo comenzó a ser adoptado por muchas más universidades. En el año 2003 aparecieron los primeros sitios web OCW (el MIT OCW y el The Fulbrigth Economics Teaching Program OCW).

De una comunidad de instituciones que compartían este compromiso se hizo necesario la creación de una organización, no dependiente del MIT, que liderara y diera soporte al gran volumen de discusiones que surgieron en torno a la publicación en abierto de materiales. Así se creó el OpenCourseWare Consortium (OCWC)[2]. Cuyo

[2] Es importante mencionar que en el año 2014 se ha cambiado el nombre de OCWC a Open Education Consortium (OEC) para más información revisar http://www.oeconsortium.org/

plan estratégico se articuló a partir de la visión de un mundo en el que, para cada deseo de aprender, hay una oportunidad independientemente de quién la formule, el momento y lugar donde se encuentre. Se definió la misión de la organización como la intención de *"progresar en el aprendizaje formal e informal compartiendo y usando en todo el mundo materiales educativos gratis, abiertos y de alta calidad organizados como cursos".*

Comunidad mundial de cientos de instituciones de educación superior y organizaciones asociadas comprometidas a promover OpenCourseWare y su impacto en la educación global, siendo un órgano de coordinación para el movimiento a escala mundial y como un foro para el intercambio de ideas y planificación futura.

4.1. Beneficios de OCW

Un proyecto OCW puede atraer muchos beneficios a la institución, como por ejemplo progresar para alcanzar la misión, estimular la innovación, captar nuevos alumnos, y crear orgullo de comunidad. Cada institución debe resaltar el aspecto que más le interese.

Entre los elementos positivos que nos brinda OCW se puede mencionar:

- El uso de OCW mejora la calidad de los materiales docentes que ofrecen los profesores, al nutrirse de la experiencia de otros materiales que han sido probados con éxito.
- El OCW incentiva a hacer actividades virtuales.
- El OCW abre ventanas de colaboración con otras instituciones y otros profesionales posicionados en la materia.
- Los alumnos pueden consultar los materiales antes de matricularse con un profesor, lo que les permite tener más herramientas para tomar su decisión.
- Se hace más visible en qué medida la universidad devuelve a la sociedad los recursos que toma de ella, de manera que ésta pueda valerse de éstos directamente.

- Para los gestores se puede saber cómo orientan los profesores su docencia y el aula deja de ser una caja negra en la que no se sabe qué pasa.
- Incrementa la difusión del conocimiento, lo cual es beneficioso en sí mismo.
- Etiquetar los contenidos ha aumentado su visibilidad, incrementando el número de visitas." (López, Piedra, Sancho, Soto, & Tovar, 2012)

Los contenidos de OCW están estructurados en lo general a las necesidades de cada uno de los sitios. En la mayoría de estos se trabaja bajo el nombre de cursos y cada uno de estos se clasifican en Facultades técnicas, biológicas, comunicativas, etc.; como también en menús de cursos como guías; ejemplos, áreas de estudios tipos de materias, etc. Esto no se puede normalizar puesto que las necesidades son diferentes y las palabras o clasificadores utilizadas son propias de cada región.

Los recursos educativos abiertos básicamente se encuentra en el idioma español e inglés a nivel de Latinoamérica. En consorcios Occidentales se manejan lenguaje básico como japonés, chino tradicional, e inglés.

4.2. OCW en Iberoamérica.

Iberoamérica es un grupo de países que está buscando el desarrollo y no solo en la parte industrial sino además en la parte educativa, puesto que la educación en estos países está siendo el principal aporte luego de la sustentabilidad económica como turismo, exportaciones de materias primas, productos, entre otros.

Universia nace en base a la necesidad de establecer los principios de OCW Consortium para Iberoamérica, siendo capaz de obtener y colaborar con recursos educativos, los países que siguen el enfoque Open Data son: Argentina, México, Venezuela, Brasil, Chile, Colombia, Perú, Puerto Rico, República Dominicana, Uruguay, Ecuador.

(OpenCourseWare UTPL, 2012) Menciona "A partir del año 2009 la Universidad Técnica Particular de Loja es la primera en pertenecer al grupo de Universia y al Consorcio OCW basándose en los principios de OCW y con el propósito de: "Proporcionar un acceso libre, sencillo y coherente en los materiales de los cursos que

ofrecen sus modalidades de estudio presencial y distancia para educadores, estudiantes y autodidactas de todo el mundo. (http://.ocw.utpl.edu.ec/)"

Además de crear un modelo eficiente basado en estándares generados de OCW Consortium, Universia, las tendencias tecnológicas de la Web Social y Semántica, así como de otros organismos afines; para que otras universidades puedan acceder a la hora de publicar sus propios materiales pedagógicos."

5. Lenguaje natural.

Uno de los principales medios de comunicación de las personas para expresar o trasmitir sus ideas, conocimientos, sentimientos, emociones es de manera escrita u oral; el lenguaje tiene un gran valor expresivo y posee características especiales como las palabras que dan sentido o resaltan una oración se puede mencionar a la conjugación de los verbos, utilización de adjetivos, adverbios, artículos, preposiciones, sustantivos, entre otros; estudiar todos estos detalles que conforman su estructura implica una de las tareas más tediosas de realizar debido a la riqueza que contiene.

5.1. Niveles de lenguaje

(Bolshakov & Gelbukh, 2004) (citado por Torres, 2009) menciona que "la lingüística general comprende 5 niveles principales para el análisis de la estructura del lenguaje estos son:

- **Nivel fonológico:** trata de los sonidos que comprenden el habla, permitiendo formar y distinguir palabras.
- **Nivel morfológico:** trata sobre la estructura de las palabras y las leyes para formar nuevas palabras a partir de unidades de significado más pequeñas llamadas morfemas.
- **Nivel sintáctico:** trata como las palabras pueden unirse para construir oraciones y cuál es la función que cada palabra realiza en esa oración.
- ***Nivel semántico:*** *trata del significado de las palabras y de cómo se unen para dar significado a una oración.*

- **Nivel pragmático:** estudia la intención del hablante al producir oraciones específicas o textos en una situación específica."

El presente proyecto se enfocará únicamente en el nivel semántico que *"trata el significado de las palabras y de cómo se unen para dar significado a una oración." Para nuestro caso de estudio se realizará un análisis semántico de los recursos educativos abiertos específicamente se tomarán los documentos de texto que estén en formato PDF.*

6. Procesamiento del lenguaje natural

Según (Torres, 2009) menciona que "El estudio del lenguaje está relacionado con varias disciplinas. Una de ellas es la lingüística general, que estudia la estructura general y descubre las leyes universales de funcionalidad de los lenguajes naturales. Estas estructuras y leyes, unidas a los métodos computacionales forman la lingüística computacional."

Bolshakov & Gelbukh (citado por Torres, 2009) piensa que "la lingüística computacional puede ser considerada como un sinónimo de procesamiento de lenguaje natural, ya que su tarea principal es la construcción de programas que procesen palabras y textos en lenguaje natural."

Para realizar el procesamiento de texto en lenguaje natural la ambigüedad es un inconveniente y tema principal a resolver, es relevante establecer la conceptualización necesaria para manejar este problema con mayor formalidad, en los siguientes apartados se encontrará referencias que son trascendentales para aclarar el tema.

7. Ambigüedad en el lenguaje natural.

Al hablar de ambigüedad inmediatamente viene a nuestra mente varios significados para una palabra o una oración que no está clara; para solucionar el primer caso la forma más fácil y sencilla de hacerlo es buscando sus posibles significados en un diccionario y seguidamente elegir el que mejor se adapte, pero ¿cómo sabemos cuál de ellos se adapta mejor? Pues bien, se toma como principal referencia el contexto o

la oración en el cual se encuentra la palabra. En cuanto al segundo caso se debe realizar un análisis un poco más detallado de la estructura de la oración, podría ser que falten signos de puntuación o alguna palabra como adjetivo, artículo, complemento.

7.1.1. Tipos de ambigüedad.

Existen tres tipos principales de ambigüedad que se mencionarán a continuación:

- Sidorov (citado por Torres, 2009) sugiere que la ***ambigüedad léxica*** se presenta cuando las palabras pueden pertenecer a diferentes categorías gramaticales. Por ejemplo:
  - ✓ *bajo* puede ser una preposición, un sustantivo, un adjetivo o una conjugación del verbo bajar.
- Según (Torres, 2009) la ***ambigüedad sintáctica***, también conocida como ambigüedad estructural se presenta cuando una oración puede tener más de una estructura sintáctica. Por ejemplo de la oración "María habló con el profesor del instituto" se puede entender dos cosas diferentes:
  - ✓ el profesor pertenece al instituto, o bien,
  - ✓ el tema del que hablo María con el profesor fue el instituto.
- ***Ambigüedad semántica*** y en cual se enfoca este proyecto se mostrará a detalle en el siguiente apartado.

7.1.1.1. Ambigüedad semántica

La ambigüedad semántica es un tema crucial en el presente proyecto por lo tanto se utilizarán referencias exactas de (Torres, 2009) que señala "cualquier palabra que usamos para comunicarnos tiene dos o más posibles interpretaciones, llamadas sentidos. *Para entender correctamente el sentido adecuado para cada palabra debemos tomar en cuenta su contexto."*

La determinación automática del sentido correcto de una palabra es crucial, por lo tanto es necesario mencionar la tarea principal en la que se enfocará el presente proyecto:

- **Recuperación de información:** Al realizar búsquedas por palabras claves específicas, es necesario eliminar los documentos donde se usa la palabra o palabras en un sentido diferente al deseado; por ejemplo, al buscar referencias sobre animales con la palabra gato, es deseable eliminar los documentos que contienen dicha palabra asociada con mecánica automotriz (Salton, 1968) (citado por Torres, 2009).

El Análisis de los Recursos Educativos Abiertos necesita del procesamiento de texto por lo tanto se necesita identificar el significado correcto para las palabras que se detecten como ambiguas, con la finalidad de relacionar el contenido de los recursos con otros.

8. Desambiguación de las palabras

Galicia-Haro (citado por Torres, 2009) menciona que "La ambigüedad es el proceso lingüístico que se presenta cuando pueden admitirse distintas interpretaciones a partir de una representación dada o cuando existe confusión al tener diversas estructuras y no tener los elementos necesarios para eliminar las eventualmente incorrectas. Para desambiguar, es decir para seleccionar los significados o las estructuras más adecuadas de un conjunto conocido de posibilidades, se requieren de diversas estrategias de solución según sea el caso."

Como solución para la ambigüedad se presenta el tema de desambiguación que es la selección del mejor significado para una palabra ambigua de una serie de opciones.

(Torres, 2009) *Sugiere que la desambiguación del sentido de las palabras es el problema de seleccionar un sentido de un conjunto de posibilidades predefinidas para una palabra dado en un texto o discurso.*

Según (Grettel Barceló, 2010) *en términos generales la desambiguación involucra la asociación de una palabra ambigua dada en un texto o discurso, con una definición o significado (sentido), que es distinguible del resto de los significados potenciales atribuibles a dicha palabra.*

8.1. Etapas para la desambiguación

Ide & Véronis (citado por Torres, 2009) afirman que la desambiguación de los sentidos de las palabras consta de dos etapas fundamentales:

- **La definición del conjunto de sentidos** para la palabra ambigua o la extracción de los mismos de un diccionario. Por ejemplo, para la palabra ambigua planta el diccionario de la lengua española define, entre otros, los siguientes significados:

  - ✓ planta 1 = Parte inferior del pie.
  - ✓ planta 2 = Árbol u hortaliza que, sembrada y nacida en alguna parte, está dispuesta para trasplantarse en otra.
  - ✓ planta 3 = Fábrica central de energía, instalación industrial.

- **El desarrollo de un algoritmo** que asigne el sentido correcto a la palabra para un determinado contexto. Así, si se tiene la siguiente sentencia:

  - ✓ "Científicamente se conoció la planta de hierba mate en Europa desde principios del siglo XIX."

Siguiendo estas dos etapas se puede determinar que el sentido correcto para la palabra ambigua "planta", según su contexto corresponde a la definición 2.

Según (Grettel Barceló, 2010) "existen varios recursos que permiten obtener los sentidos predefinidos de las palabras durante la primera etapa, como diccionarios,

tesauros, corpus[3] y recientemente, textos paralelos bilingües, que incluyen las traducciones de una entrada en otro lenguaje.

En la segunda etapa, la asignación correcta de los sentidos requiere del análisis del contexto en el cual la palabra ambigua está siendo empleada a partir de fuentes de conocimiento externas, información sobre los contextos de casos previamente desambiguados derivados de un corpus o ejemplos de relaciones entre artículos léxicos de diferentes lenguajes obtenidas de textos paralelos. En cualquier caso, se utiliza algún método de asociación para determinar la mejor correspondencia entre el contexto actual y una de las fuentes de información mencionadas."

8.2. Método de desambiguación basado en diccionarios

(Torres, 2009) Afirma que los diccionarios proporcionan una lista de glosas (definición de sentido) para las palabras. Los métodos que utilizan sólo diccionarios de sentidos, buscan elegir un sentido (de esta lista) para cada palabra en un texto dado, tomando en cuenta el contexto en el que aparece.

Se cita de (Grettel Barceló, 2010) que: "*el sentido que debe ser asignado a palabras con múltiples co-ocurrencias es aquel que maximiza la relación entre los sentidos elegidos". El algoritmo más importante creado sobre la base de este razonamiento, es el* ***algoritmo de Lesk****.*

En la actualidad el diccionario más conocido, robusto y empleado en estos métodos es Princeton WordNet (PWN) incluye las definiciones para los sentidos individuales de palabras, las cuales son agrupadas en conjuntos de sinónimos (synsets) para representar un solo concepto léxico, organizados en una jerarquía.

Cuenta con 155327 palabras, 117597 synsets y 207016 sentidos; está conformado por tres bases de datos correspondientes a sustantivos, vernos y una para adjetivos y adverbios. Cada base de datos está conformada por entradas léxicas que corresponden a formas ortográficas individuales. Cada palabra se asocia con un

[3] Un corpus (corpora en plural) es una colección grande de texto, esta escrito por y para los seres humanos.

conjunto de sentidos. La siguiente tabla muestra un ejemplo basado en el sustantivo person (persona). (Torres, 2009)

Tabla 3: Sentidos dados por WordNet 2.1 para el sustantivo person (persona).

SENTIDO	DEFINICIÓN DE GLOSA
1	(7229) person, individual, someone, somebody, mortal, soul – (a human being, “there was too much for one person to do”)
2	(11) person – (a human body (usually including the clothing); “a weapon was hidden on his person”)
3	person – (a grammatical category of pronouns and verb forms; “stop walking

Fuente: (Torres, 2009)

Una muy buena razón para el amplio uso de WordNet es su condición de recurso léxico libre y ampliamente disponible.

8.2.1. Algoritmo de Lesk

Se ha usado como referencia principal a (Torres, 2009) para la explicación del algoritmo de Lesk, la autora menciona que “es uno de los primeros algoritmos exitosos usados en la desambiguación de sentidos de palabras. Este algoritmo está determinado por dos principales ideas: un algoritmo de optimización para WSD (Desambiguación del sentido de la palabras) y una medida de similitud para las definiciones de los sentidos.

- El primero es acerca de desambiguar palabras, considerando la optimización global del texto, esto es, encontrar la combinación de los sentidos que maximice la relación total entre los sentidos de todas las palabras.
- En el segundo punto, relacionado con la medida de similitud, Lesk sugiere usar el traslape entre las definiciones de los sentidos, es decir, contar el número de palabras que tienen en común.

Para reducir el espacio de búsqueda del algoritmo original de Lesk, Kilgarriff & Rosenzweig (citado por Torres, 2009) propusieron una variación del algoritmo original de Lesk, conocido como algoritmo de ***Lesk simplificado o Lesk Simple, donde los sentidos de las palabras en el texto son determinados uno a uno encontrando el***

***mayor traslape entre los sentidos de las definiciones de cada palabra con el contexto actual*.**

Tabla 4: Pseudocódigo del algoritmo de Lesk Simple.

```
for each word w
    for each sense i of w
        xi = overlap (si, context)
chose sense i for w, where xi is maximized
```

Fuente: (Torres, 2009)

Una de las aplicaciones del procesamiento del lenguaje natural y como enfoque del presente proyecto es la extracción del contenido de los recursos educativos abiertos contenidos en sitios Open Course Ware, con la finalidad de establecer relaciones entre ellos usando como principal referencia su contenido que será sometido a la identificación de palabras ambiguas y por consiguiente al proceso de desambiguación, utilizando el algoritmo de Lesk simplificado y el diccionario Wordnet.

9. Extracción de la información

(Russell & Peter, 2003) mencionan que "la extracción de la información es el proceso de crear entradas a una base de datos a partir de un texto y buscar ocurrencias de una clase particular de objeto o acontecimiento para las relaciones entre esos objetos o acontecimientos. Por ejemplo se podría intentar extraer instancias de direcciones de las paginas web, con campos para la base de datos como ciudad, calle, el estado y el código postal. Los sistemas de extración de información se sitúan a mitad de camino entre los sistemas de recuperación de datos y los programas de análisis completo del texto.

El tipo mas simple de sistema de extracción de la información usa un sistema basado en atributos porque se asume que el texto entero se refiere a un solo objeto y la tarea es extraer las cualidades de ese objeto."

10. Visualización de redes

En 1996, un grupo de investigadores de la Universidad de Konstanz (Alemania) empezaron a desarrollar una nueva herramienta para la visualización de redes. El impulso para desarrollar dicha herramienta se basó en la creencia de los investigadores de que, primero, la visualización de redes puede ser una importante herramienta por encima y más allá de la mera ilustración de los datos; segundo, que no todas las visualizaciones de redes son igualmente efectivas para ello (como señaló (Tufte, 1983): 'el diseño es una elección'); y tercero, que ninguna de las herramientas de visualización que estaban disponibles en ese momento cumplían los principios de excelencia gráfica de Tufte. Estos principios son:

- La excelencia gráfica es una presentación bien diseñada de datos interesantes una cuestión de sustancia, de estadística y de diseño.
- La excelencia gráfica consiste en la comunicación de ideas complejas con claridad, precisión y eficiencia.
- La excelencia gráfica es la que da al observador el mayor número de ideas en el más corto espacio de tiempo con menos tinta en el espacio más pequeño.
- La excelencia gráfica es casi siempre multi-variada.
- La excelencia gráfica requiere contar la verdad de los datos.

En consecuencia, se desarrolló el argumento de que para producir visualizaciones efectivas hay que identificar claramente la información relevante, es decir, filtrar, transformar y procesar la colección de actores, lazos y atributos para identificar la sustancia interesante, definir un mapa apropiado para la representación gráfica, y generar la imagen correspondiente sin introducir artefactos. (Brandes, Kenis, & Raab, 2005)

(Foley & Kibasky, 1994) definen a la visualización como "la transformación de los datos a una representación que puede ser percibida por los sentidos. Los resultados de esta transformación pueden ser visuales, auditivos, táctiles o una combinación de estos."

(Espinoza, Martinez, & Racine, 2013) mencionan que una de las técnicas mas utilizadas para visualizar la relación existente entre elementos son los grafos dirigidos los mismos que poseen dos características que son precisas mencionarlas como:

- **Aristas:** son lineas que unen los nodos de un grafo, y constituyen los caminos que pueden recorrerse.
- **Vértices:** son los puntos o nodos con los que esta conformado un grafo.

La razón por la cual se utliza visualización de redes para el presente proyecto se justifica por las características mencionadas en esta sección, además el objetivo es presentar los resultados de las relaciones encontradas entre OER's de una manerá gráfica, atractiva y fácil de comprender para los usuarios finales.

Es importante mencionar a los trabajos relacionados que nos sirven de referencia y como base de conocimiento para el presente proyecto, es importante saber y tener en claro los proyectos ya desarrollados con los cuales nos podamos apoyar en el proceso de investigación e implementación, esta información se puede observar en el anexo 19.

CAPITULO 2: DEFINICIÓN DEL MARCO DE TRABAJO

1. Problemática

A nivel mundial existen alrededor de 78 Open Course Ware (OCW) pertenecientes en su gran mayoría a universidades, que nos permiten tener acceso al conocimiento mediante Recursos Educativos Abiertos (Open Educational Resource). De acuerdo al informe sobre “Aplicación de tecnologías web emergentes para el estudio del impacto de repositorios OpenCourseWare españoles y latinoamericanos en la Educación Superior” en (López, Piedra, Sancho, Soto, & Tovar), el objetivo de estos OCW es transmitir conocimiento sin barrera alguna de nacionalidad, idioma, religión, clase social; su objetivo es claro mejorar el entorno social en el que se desarrollan un individuo.

Existe un total aproximado de 47000 recursos educativos abiertos de los cuales la mitad corresponden a documentos de texto en su gran mayoría en formato PDF, cantidades considerables de información disponible, pese a esto y hasta el momento no se la ha sometido a programas que extraigan y procesen texto en lenguaje natural; rezagando a estas fuentes tan importantes de información.

Actualmente estos recursos no cuentan con relaciones establecidas con otros OER’s respecto a un tema, se puede decir que se encuentran aislados pertenecen a un sitio OCW pero carecen de relación entre ellos.

Por lo mencionado tampoco existe una herramienta para visualización que haya sido implementada y que permita apreciar las relaciones existentes entre los recursos educativos abiertos.

Linked Data menciona algunos requisitos que la data debe cumplir como; a) se debe generar datos en formato RDF y b) se debe enlazar los datos con otros datasets; los OER’s no cumplen con estos requerimientos, carecen de formato RDF y no están enlazados a otros datasets.

2. Planteamiento de la solución

En esta sección se muestra la solución propuesta para el proyecto “Análisis y Visualización de Recursos Educativos Abiertos pertenecientes a sitios Open Course Ware”, es preciso indicar la intervención de los conceptos mencionados durante el “Estado del Arte” en el desarrollo del presente proyecto.

Linked Data es el entorno en el cual se trabajará por lo tanto se respetará su arquitectura, cada uno de sus principios, y se hará uso de los beneficios que ofrece; el establecer una “Metodología para la publicación de datos” (Ver sección 2.3 del capitulo 1) ha dado lugar al desarrollo efectivo de proyectos relacionados (Ver Anexo 19, del capitulo 1), la claridad con la que se define cada proceso y la facilidad para adaptarlo a nuestras necesidades le da un valor agregado.

El gráfico siguiente muestra 5 de las 7 fases en total que posee la “Metodología para la publicación de datos”, se adaptó a nuestra necesidad, será incluida en el primer proceso creado y denominado “Extracción y Procesamiento de Información de OER/OCW”.

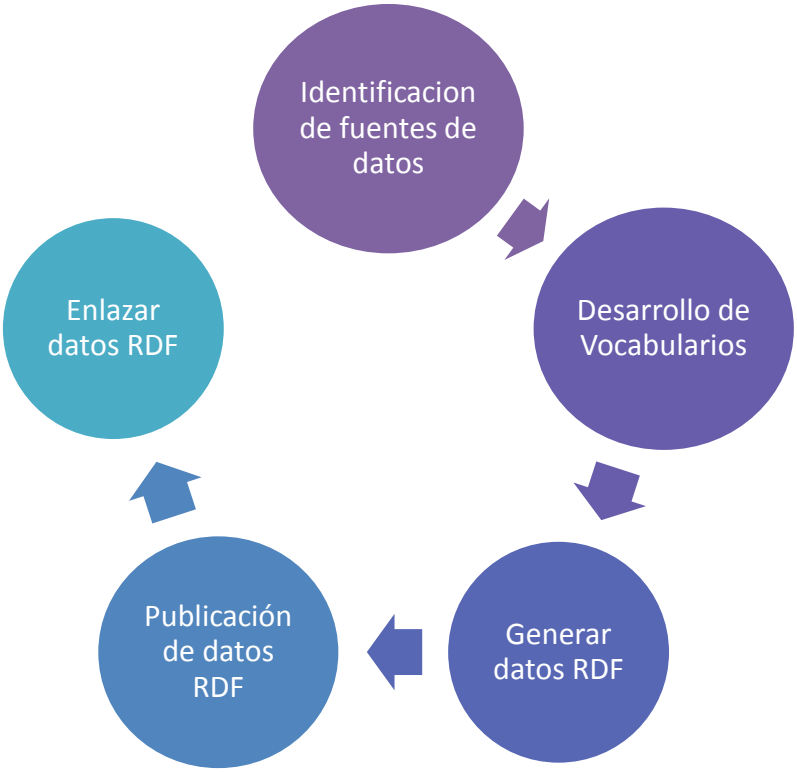


Figura 8: Metodología para la publicación de datos, que se adaptará para el presente proyecto

Para cumplir con esta metodología y los procesos que se establecerán, es necesario utilizar las tecnologías competentes necesarias y establecidas para este fin (ver sección 2.4) como:

- Lenguaje RDF para representar la información.
- Lenguaje de consultas SPARQL para acceder a los datos.
- Virtuoso triplestore para el almacenamiento de los datos en RDF.
- LOV[4] Linked Open Vocabularies posee un extenso catálogo de vocabularios, listos para ser reutilizados.
- DBpedia para enlazar la información obtenida.
- NLTK conjunto de herramientas para la extracción y procesamiento del lenguaje natural.

La información que contiene los recursos educativos abiertos pertenecientes a sitios Open Course Ware, específicamente aquellos que contienen texto se convertirá en nuestra materia prima que será extraída, procesada, almacenada y relacionada. Sin embargo esto no se puede lograr si no establecemos procesos que nos conduzcan a lograr los resultados esperados, esta es una de las principales normas en el desarrollo de aplicaciones, en nuestro ámbito no es la excepción.

El siguiente diagrama indica los procesos generales que intervienen durante el desarrollo de nuestro tema de estudio, en los siguientes capítulos se describirá a cada proceso.

[4] http://lov.okfn.org/dataset/lov/

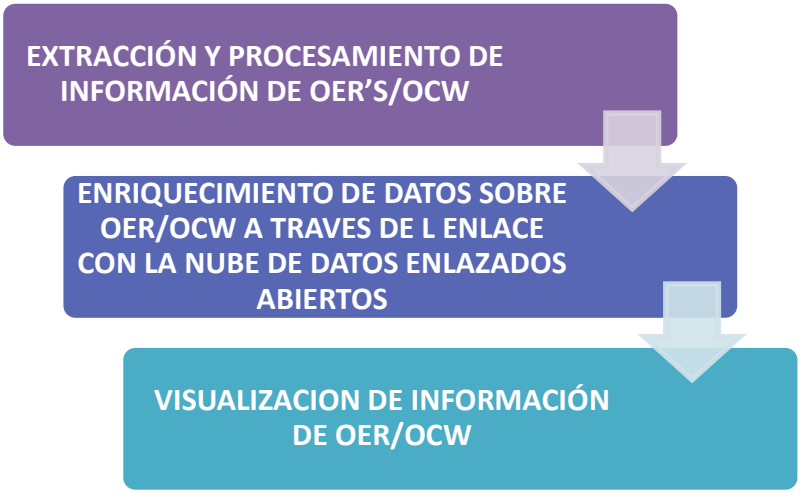


Figura 9: Procesos de Desarrollo para Análisis y Visualización de OER's

Open Course Ware con su visión de compartir conocimiento abiertamente con la finalidad de mejorar el aprendizaje mediante OER's de muy buena calidad, se la considera como la principal fuente de información para este proyecto, mediante scrapy web[5] (proceso desarrollado en un proyecto previo a este) se obtuvo la data necesaria sobre los Recursos Educativos Abiertos que serán nuestra principal fuente de datos.

El departamente de Tecnologías Avanzadas en la Web y Sistemas Basados en el Conocimiento de la UTPL, desarrolló el proceso de Scrapy Web que se encarga de la obtención de información de sitios OCW por ejemplo: la estructura del sitio web, los cursos que posee, las asignaturas, los URI's correspondientes a cada OER's, entre otros.

Es necesario darle la importancia que le corresponde a los recursos educativos abiertos por poseer contenido de alta calidad y garantizado por universidades, debería convertirse una de las primeras elecciones de consulta para los estudiantes en nuestro medio, y para los docentes ser una opción de referencia en sus cátedras.

A los Recursos Educativos se los puede obtener en *diferentes tipos* como vídeos, documentos de texto, hojas de cálculo, presentaciones...; en *diferentes formatos* .doc, .docx, .xls, .ppt, .pdf; *en diversos idiomas* español, francés, inglés, italiano...; *en varios temas* química, matemáticas, estructuras de datos entre otros. Convirtiéndolos en una fuente rica de conocimiento para todos los gustos con la finalidad de explotarlos,

[5] Extracción de información de sitios Web

compartirlos, y colaborar. La imagen siguiente muestra la cantidad de OER's clasificados según su formato.

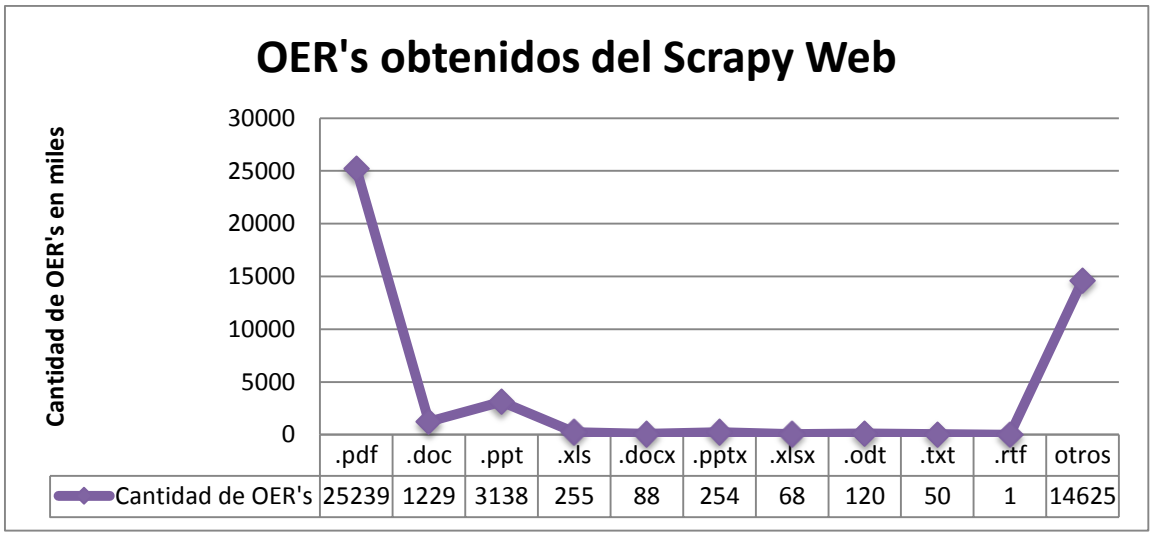


Figura 10: Cantidad de OER's disponibles por su formato

Se eligió documentos en ***.pdf*** por la existencia de gran cantidad de OER's en este formato como se observa en la gráfica en comparación con documentos .doc, .ppt, .xls.

Para el procesamiento del lenguaje natural se necesita extraer información (ver secciónes 6 y 9) de documentos de texto en nuestro ámbito de los OER's como texto, tokens, tags, palabras representativas, entidades, con el fin de analizar esta información a nivel semántico que "trata el significado de las palabras y de cómo se unen para dar significado a una oración" (Ver sección 5.1).

Durante el procesamiento del lenguaje natural nos encontramos con un problema a resolver como la *ambigüedad semántica* que trata sobre una palabra con varios significados (ver sección 7), los OER's contienen gran cantidad de estas palabras que deben ser sometidas a un proceso de desambiguación (ver sección 8).

El método de desambiguación que se utilizará *está basado en diccionarios como se menciona en la sección 8.2, usando como base el algoritmo de Lesk y el diccionario WordNet.* Es preciso indicar que únicamente se trabajará con OER's en idioma inglés debido a factores como: a) la disponibilidad del diccionario Princeton WordNet solamente en inglés para la herramienta NLTK, b) es uno de los idiomas más usados a

nivel académico. La gráfica siguiente muestra los idiomas con mayor número de hablantes.

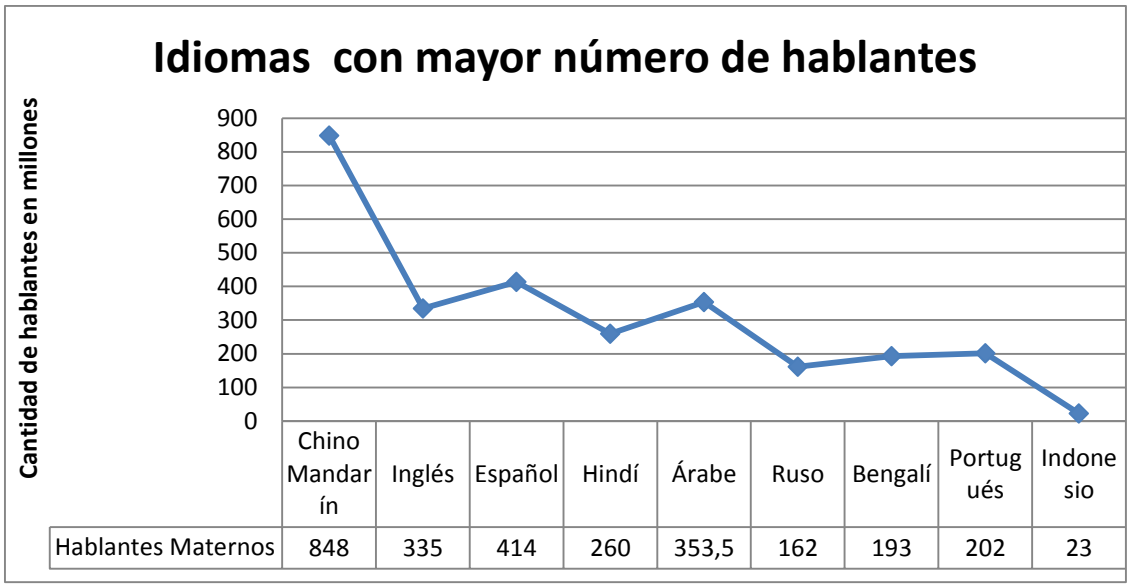


Figura 11: Gráfico estadístico de los idiomas con mayor número de hablantes, tomado de Wikipedia

Para hacer posible la relación con otros recursos educativos abiertos es necesario el establecimiento del significado correcto para la o las palabras identificadas como ambiguas, una vez solucionado este inconveniente se estableceran las relaciones usando los tokens o palabras en común detectadas entre OER's; por ejemplo:

- Si tenemos dos OER's que tratan sobre el tema de Análisis Matemático se los relacionará, pero si tenemos un tercer OER que trata sobre Multimedia obviamente no se relaciona con los dos anteriores por lo tanto no se toma como una relación, este es el proceso que se resolverá.

Finalmente los resultados obtenidos se los presentará utilizando los conceptos de visualización de redes (ver sección 10) que nos permitirá dar una visualización amigable y entendible de las relaciones establecidas y existentes entre OER's, además es una de las tendencias mas utilizadas en la actualidad para presentar información.

En los siguientes capítulos y secciones se abarcarán los procesos necesarios para el desarrollo del actual proyecto, de igual forma se mencionarán las herramientas que

intervienen en cada fase sin embargo si se desea una explicación un poco mas detallada acerca de las herramientas utilizadas se puede ver el Anexo 20, además se estableció la arquitectura y funcionalidad correspondiente a la aplicación que se desarrollará, se puede observar en los Anexos 1 y 2 respectivamente.

Tabla 5: Resumen de los problemas y soluciones encontradas

PROBLEMA DETECTADO	SOLUCIÓN PROPUESTA	IMPACTO
Información contenida en recursos educativos abiertos de documentos de texto en formato PDF sin ser explotada.	Extraer, analizar, y procesar la información que contienen los OER's.	Alto
Inexistencia de relaciones entre OER's pertenecientes a sitios OpenCourseWare.	Encontrar relaciones entre los recursos educativos abiertos.	Alto
No existe la implementación de una herramienta para visualización de relaciones entre OER's.	Implementar una herramienta basada en visualización de redes para presentar de una manera amigable y entendible las relaciones obtenidas entre OER's.	Medio
Los OER's no cumple con el requisito establecido por Linked Data, como enriquecimiento de data con otros dataset.	Enlazar la información obtenida con el dataset DBpedia.	Alto
Los OER's no cumple con el requisito establecido por Linked Data, como generar datos en formato RDF.	Generar y almacenar toda la data obtenida de los OER's en formato RDF.	Alto

CAPITULO 3: EXTRACCIÓN Y PROCESAMIENTO DE INFORMACIÓN DE OER'S

1. Introducción

Este capítulo presenta el proceso de extracción y procesamiento de información de los OER's, se enfoca en la ejecución de las siguientes tareas a) identificación de fuentes de datos, b) desarrollo de vocabularios, c) obtención de OER's, d) extracción de información, e) procesamiento de información, f) generar datos RDF, y finalmente g) publicación de datos RDF.

Se especifica el propósito que se desea alcanzar en cada tarea, además se menciona las herramientas tecnológicas utilizadas durante su ejecución y se presenta los resultados obtenidos.

2. Propósito del proceso

"Extracción y procesamiento de Información de OER's" es el principal proceso a ejecutarse para lo cual se ha establecido tres propósitos; a) obtener OER's que sean documentos de texto en formato PDF, desde sitios OCW, b) extraer información de OER's en idioma inglés c) procesar la información obtenida que involucra al proceso de desambiguación, y la creación de relaciones entre OER's.

3. Precondiciones para ejecutarlo

Aproximadamente se trabajará con unos 1200 recursos educativos abiertos por lo tanto es necesario conexión a Internet con un buen ancho de banda para que el proceso de obtención de OER's se ejecute con rapidez. Además es necesario tener acceso a la tabla o replica "CursosConsortium" proporcionada por el Departamento de Tecnologías Avanzadas en la Web y Sistemas Basados en el Conocimiento.

4. Pasos generales a ejecutar

Es importante ejecutar pasos que contribuyan al desarrollo ordenado del proceso se utiliza como referencia 5 de las 7 tareas mencionadas en "Metodología para la publicación de datos" , además es necesario agregar algunos pasos enfocados a

nuestro ámbito de trabajo. La siguiente figura muestra todos las tareas que deben ejecutarse para el proceso de “Extracción y Procesamiento de Información de OER’s”.

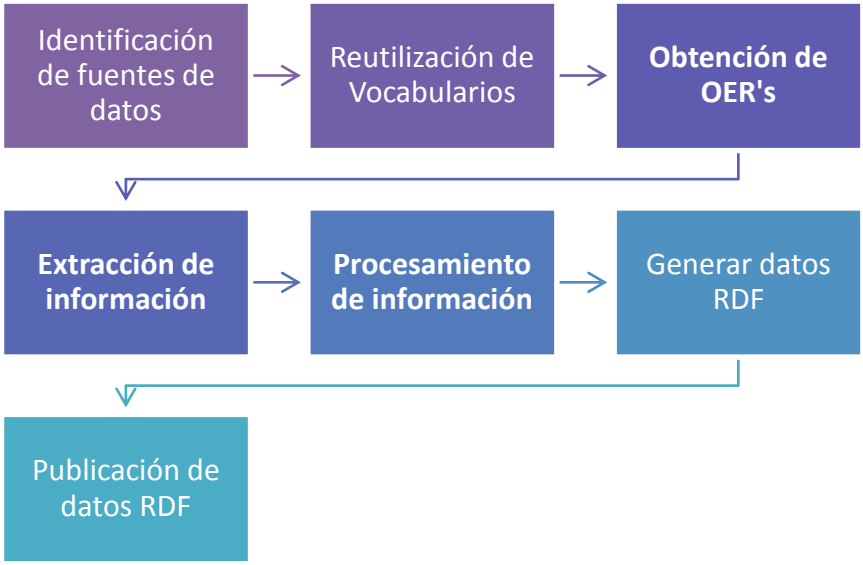


Figura 12: Pasos a ejecutar en el proceso de “Extracción y Procesamiento de Información de OER’s”

En las siguientes secciones se menciona cada una de las tareas que involucra el proceso de “Extracción y Procesamiento de Información de OER’s” mencionadas en la figura anterior, se indicará el propósito de cada tarea, su ejecución y las herramientas que se utilizan para su funcionamiento. Los artefactos técnicos como casos de uso y diagramas de clases correspondientes a este proceso de los puede observar en los Anexos 3 y 4 respectivamente.

4.1. Identificación de fuentes de datos

El propósito de esta tarea y como su nombre lo menciona es identificar la principal fuente de datos que será de utilidad para el proyecto.

4.1.1. Ejecución

Previo a este proyecto se desarrolló un proceso de scrapy web para sitios OCW por parte del grupo de TAW-SBC (Tecnologías Avanzadas de la Web y Sistemas Basados en el Conocimiento), del cual se generó una tabla con el nombre *“CursosConsortium”* perteneciente al esquema de base de datos *“scrapyconsortium”*, los datos están

almacenados en tripletas *(sujeto, predicado y objeto)*, contienen valores referentes al curso OCW como la url, titulo, menú, descripción, oer, texto, entre otros.

Por lo tanto este dataset posee la información base y necesaria para el desarrollo del presente proyecto la siguiente imagen muestra la estructura de la tabla "CursosConsortium" que será utilizada como la principal fuente de datos.

Tabla 6: Estructura de tabla "CursosConsortium"

NOMBRE ATRIBUTO	VALOR QUE RECIBE CADA ATRIBUTO	COMENTARIO
sujeto	URL	URL del OCW del cual se está realizando el proceso de scrapy web.
predicado	link, rdf:type, ***oer***, title, Description, html, error, menu	Lista de valores que se pueden utilizar para relacionar al sujeto con el objeto. Se utilizará el predicado "oer"
objeto	URL, texto	Puede ser: un URI que identifica a un OER, o un extracto de texto como una descripción, el menú que contiene el curso OCW. Depende del valor que se elija como predicado.

Fuente: (Novillo, 2013)

4.1.2. Resultados

La tabla "CursosConsortium" contiene 427802 registros en total; de la lista de valores establecida para el predicado solamente se utilizará el valor ***"oer"*** como resultado se obtienen 45067 registros sin duplicados correspondientes a este valor, la siguiente imagen muestra un ejemplo de los registros que serán utilizados:

Tabla 7: Extracto de los registros obtenidos con el predicado "oer"

SUJETO	PREDICADO	OBJETO
http://oer.avu.org/handle/123456789/267	oer	http://oer.avu.org/bitstream/handle/123456789/267/Programa%c3%a7%c3%a3o-Linear.pdf?sequence=1
http://oer.avu.org/handle/123456789/274	oer	http://oer.avu.org/bitstream/handle/123456789/274/Analysis%202.pdf?sequence=1
http://oer.avu.org/handle/12345	oer	http://oer.avu.org/bitstream/handle/1234567

6789/266		89/266/Separa%c3%a7%c3%a3o%2c%20T%c3%a9cnicas%2c%20Electroanal%c3%adtica%20e%20Espectroqu%c3%admica.pdf?sequence=2
http://oer.avu.org/handle/123456789/268	oer	http://oer.avu.org/bitstream/handle/123456789/268/Probabilidade%20e%20Estat%c3%adstica*.pdf*?sequence=1
http://oer.avu.org/handle/123456789/254	oer	http://oer.avu.org/bitstream/handle/123456789/254/TIC%27s%20em%20F%c3%adsica*.pdf*?sequence=1
http://oer.avu.org/handle/123456789/256	oer	http://oer.avu.org/bitstream/handle/123456789/256/BIOLOGIA%20CELULAR%20%26%20GENETICA%20Trad.*pdf*?sequence=1
http://oer.avu.org/handle/123456789/255	oer	http://oer.avu.org/bitstream/handle/123456789/255/F%c3%adsica%20Estat%c3%adstica.*pdf*?sequence=1

El valor que contiene *el campo "objeto" es un URI que identifica al OER, esta URI contiene el nombre del recurso y el tipo de archivo*; por ejemplo se puede encontrar .pdf, .doc, .xls, .flv.

Por lo tanto al filtrarlos por el tipo de archivo "***.pdf***" se obtienen *25239 registros, se utilizará una muestra de* ***1200*** *registros en este formato.* El script que contiene esta tarea se lo puede observar en los Anexos 5 y 6.

4.2. Reutilización de vocabularios

El propósito de esta tarea es definir el vocabulario RDF a reutilizar para identificar la data que se obtendrá en las tareas siguientes.

4.2.1. Ejecución

Es importante definir el vocabulario RDF que se re-utilizará y que se acople a nuestra necesidad, se utilizó LOV (Linked Open Vocabularies) como una de las mejores opciones para buscar vocabulario que describa la metadata.

4.2.2. Resultados

El vocabulario seleccionado y el valor que identifica se presentan en la siguiente tabla, (la información fue tomada de cada una de las descripciones originales de cada vocabulario).

Tabla 8: Vocabulario RDF para el proceso de Extracción y Procesamiento de Información

VOCABULARIO RDF	IDENTIFICA A
http://xmlns.com/foaf/0.1/Document	Documento
http://www.aktors.org/ontology/portal#has-page-numbers	Número de páginas
http://purl.org/spar/doco/Title	Una palabra, frase, oración que indique el nombre de un documento o un componente de un documento. Por ejemplo: libro, reporte, articulo de noticia, capitulo, sección, o una tabla de datos.
http://purl.org/dc/elements/1.1/language	Idioma
http://www.lexinfo.net/ontology/2.0/lexinfo#standardText	Texto estándar
http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word	Representa una cadena de caracteres, como un token o palabra, pronombre, signo de puntuación, apostrofe, o que este separado por un espacio u otro carácter. Utilizado para el proceso de tokenización
http://www.w3.org/ns/dcat#keyword	Una palabra clave o un tag que describa al dato.
http://purl.org/spar/doco/Glossary	Definición de palabras o frases por su importancia, normalmente ordenadas alfabéticamente.
http://purl.org/spar/doco/ListOfAgents	Lista de ítems que denoten un agente como por ejemplo un autor, contribuyente, organización, relacionado con una publicación en particular.
http://purl.org/dc/elements/1.1/relation	Relación
http://www.w3.org/2002/07/owl#NamedIndividual	Nombre Individual
http://www.w3.org/2006/03/wn/wn20/schema/word	Un Synset contiene uno o más significados de palabras, específicamente una palabra que contenga más de un significado.

	Synset es el concepto específico usado por WordNet.
http://www.w3.org/2006/03/wn/wn20/instances/	Instancia de un Synset contiene la palabra y el número que identifica el significado que le correspondiente. Utilizado para especificar valores relacionados con el diccionario de Princeton WordNet.
http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense	Contiene el significado de palabras
http://www.w3.org/2002/07/owl#sameas	Similar a

Fuente: http://lov.okfn.org/dataset/lov/

El vocabulario mencionado en la tabla anterior identificará a la data que se obtendrá en las tareas siguientes. Adicionalmente a los valores establecidos fue necesario crear dos vocabularios en la siguiente tabla se pueden observar.

Tabla 9: Valores adicionales para el vocabulario RDF

VOCABULARIO RDF	IDENTIFICA A
http://dataoers.org/oer/commonentity/	Las entidades comunes encontradas entre los OER's.
http://dataoers.org/oer/commonword/	Las palabras comunes encontradas entre los OER's.

4.3. Obtención de OER's

El propósito principal de este proceso es descargar los OER's desde los sitios OCW.

Se trabaja con los registros de la tabla "CursosCosortium" que correspondan al predicado **"oer"** y al objeto que posee el URI que identifica al OER (ver tabla 5 y 6). El siguiente gráfico indica las tareas que se realizaran para obtener los recursos educativos abiertos.

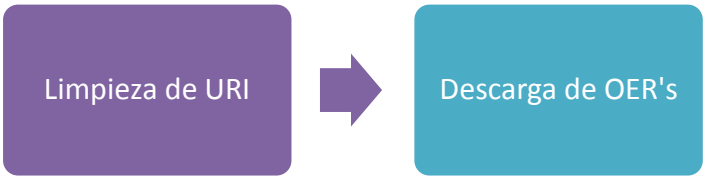


Figura 13: Tareas de Ejecución para el Proceso de Obtención de OER's

4.3.1. Limpieza de URI

El propósito de esta tarea es obtener una URI válida por los protocolos de red, para su posterior descarga.

4.3.1.1. Ejecución

Un URI identifica de manera única a un recurso de la Web en este caso a un OER, al momento de realizar una consulta SQL a la base de datos la respuesta que retorna se altera agregándose caracteres especiales (comillas, paréntesis, coma) al inicio y al final del registro. Como ejemplo tenemos el siguiente valor **'**http://oer.avu.org/handle/123456789/264/Analysis%202.pdf?sequence=1**')** que es un URI invalida para los protocoles de red, por lo tanto estos caracteres deben ser eliminados para obtener una URI que sea aceptada al momento de acceder a la Web.

4.3.1.2. Resultados

Como resultado del proceso de limpieza tenemos un URI sin ningún carácter especial y lista para ser utilizada por protocolos de red, la imagen siguiente presenta un extracto de URIs limpias.

Tabla 10: Extracto de URI's limpias

URL LIMPIAS
http://oer.avu.org/bitstream/handle/123456789/267/Programa%c3%a7%c3%a3o-Linear.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/274/Analysis%202.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/266/Separa%c3%a7%c3%a3o%2c%20T%c3%a9cnicas%2c%20Electroanal%c3%adtica%20e%20Espectroqu%c3%admica.pdf?sequence=2
http://oer.avu.org/bitstream/handle/123456789/219/Microbiologie%20et%20Mycologie.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/221/Mecanique%20II.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/225/Methode%20d%27enseignement.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/237/Philosophie%20de%20l%27education.pdf?sequence=1

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7 .

4.3.2. Descarga de OER's

El propósito de esta tarea es descargar los OER's desde sitios OCW al equipo en el cual se ejecuta el script, e identificar el nombre del OER para posteriomente acceder a ellos.

4.3.2.1. Ejecución

La descarga de los OER's se la realiza directamente en el servidor que almacena nuestra aplicación, para esta tarea se considera:

- **Obtener el nombre original del OER:** se obtiene el nombre con la finalidad de mantener datos originales, para adquirir esta propiedad se utiliza la URI limpia, de la cual se toma los caracteres después de la última barra (/) del URI hasta el último carácter o hasta después de la letra f.

Tabla 11: Ejemplo de URI del OER a descargar

URI LIMPIA	NOMBRE ORIGINAL DEL OER
http://oer.avu.org/handle/123456789/264/*Analysis%202.pdf?sequence=1*	Analysis%202.pdf
http://oer.avu.org/bitstream/handle/123456789/219/Microbiologie%20et%20Mycologie.pdf?sequence=1	Microbiologie%20et%20Mycologie.pdf
http://oer.avu.org/bitstream/handle/123456789/237/Philosophie%20de%20l%27education.pdf?sequence=1	Philosophie%20de%20l%27education.pdf

- **Agregar Identificador:** existen OER's que tienen el mismo nombre a pesar de pertenecer a diferentes OCW, para evitar una omisión o sobre escritura de archivos se agrega al inicio de cada OER un identificador numérico continuo, empezando en 1.

- **Realizar descarga:** para realizar la descarga del recurso se ejecuta 4 operaciones: 1) acceder al recurso, 2) crear el archivo en el disco destino con

el nombre correspondiente, 3) escribir su contenido, y finalmente 4) cerrar la conexión.

Se utiliza métodos para la manipulación de archivos que permitan el acceso a objetos almacenados en la web, Python ofrece la librería:

- **Urllib** idónea para esta tarea de descarga de archivos de la web.

4.3.2.2. Resultados

Se descargaron *2865* OER's en formato PDF, en la siguiente imagen se pueden observar un extracto de los documentos almacenados en el servidor "apolo" en la carpeta OERs.

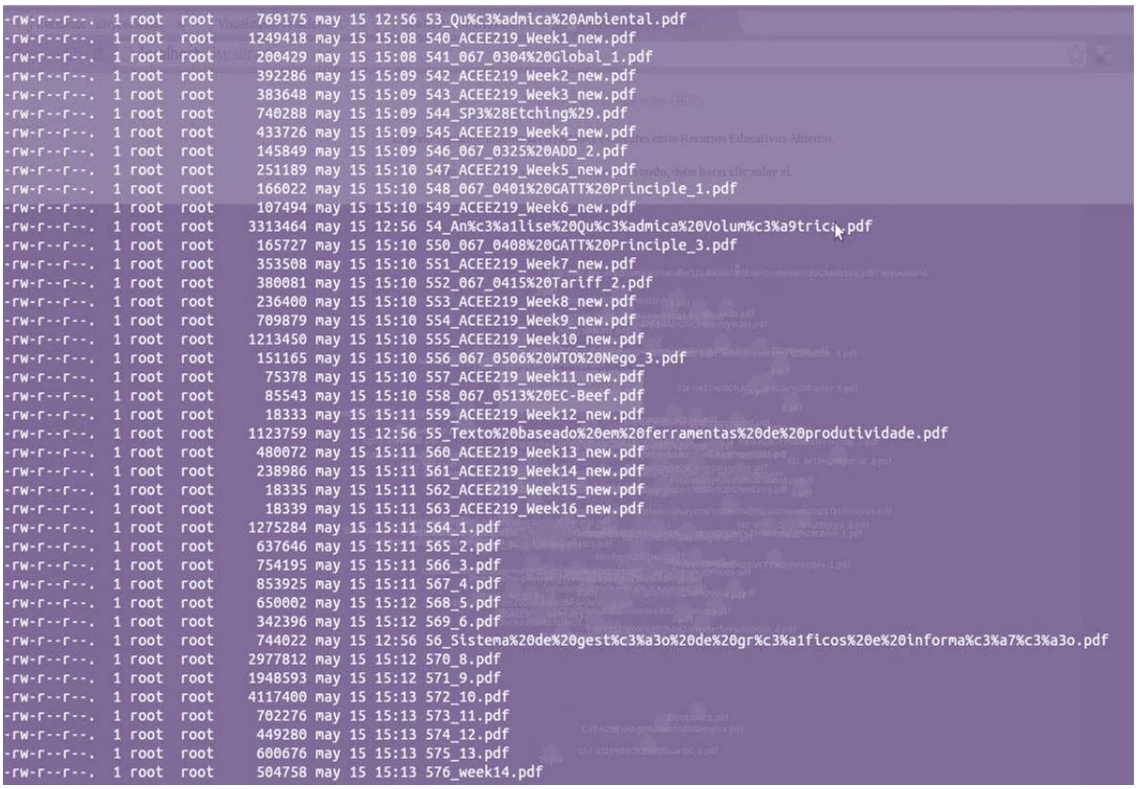


Figura 14: OER's descargados en el servidor apolo.utpl.edu.ec

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4. Extracción de información

El propósito de esta etapa es extraer la información del OER como el número de páginas, idioma, porción de texto, entidades, tokens, tags, palabras representativas.

Para el proceso de extracción de información de los recursos educativos abiertos se estableció tareas de ejecución que permitirá obtener ordenadamente los metadatos necesarios, la gráfica indica las tareas que se realizarán.

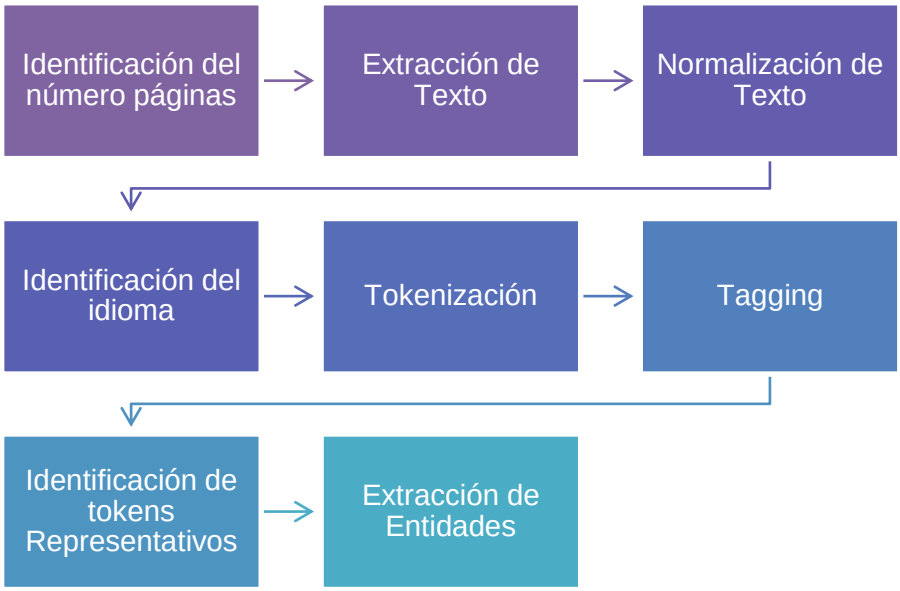


Figura 15: Tareas para la ejecución del proceso de Extracción de data

4.4.1. Identificación del número de páginas

El propósito de esta tarea consiste en establecer la cantidad de paginas de las cuales se extraerá el texto dependiendo directamente del total de paginas que posee el OER.

4.4.1.1. *Ejecución*

Para establecer la cantidad de paginas de las cuales se extraerá el texto se necesita hacer un análisis sobre este aspecto; así pues:

- Si tenemos un OER que posee 100 páginas en total y extraemos texto de dos paginas este cantidad de texto no representará la idea que abarca el recurso, se podrá realizar su análisis pero sus resultados serán vagos y sin relevancia; al contrario

- Si tenemos un OER que posee 2 páginas en total es válido extraer texto de dos páginas porque representará el contenido del OER y sus resultados serán consistentes.

Por lo tanto el número de páginas que contiene un OER se convierte en un parámetro importante para establecer la cantidad de texto que debe ser extraído del recurso.

La siguiente tabla muestra el número de páginas de las cuales se extraerá el texto, dependiendo directamente del número total de páginas que posee el OER, para lo cual se ha establecido un rango de valores.

Tabla 12: Número de páginas de las cuales se extraerá el texto

RANGO TOTAL DE PÁGINAS DEL OER	**CANTIDAD DE PÁGINAS DE LAS QUE SE EXTRAERÁ EL TEXTO**
1 – 3 páginas en total	2 páginas
4 – 6 páginas en total	4 páginas
7 – 9 páginas en total	5 páginas
10 – 19 páginas en total	6 páginas
20 – 39 páginas en total	10 páginas
40 – 69 páginas en total	17 páginas
70 – 99 páginas en total	24 páginas
100 – 149 páginas en total	34 páginas
150 – 200 o más páginas en total	47 páginas

Al especificar el número de páginas de las cuales se extraerá el texto se puede realizar el análisis y procesamiento del mismo con mayor seguridad, ya que representará o abarcará el tema que trata el OER.

Para identificar este parámetro utilizamos la librería *pyPDF* la misma que nos ayuda en la manipulación de archivos PDF como la extracción de información como: título, autor, *número de páginas*, texto de una página, entre otros.

4.4.1.2. Resultados

La figura 17 muestra un extracto del número total de páginas que tienen los OER's.

sujeto	numero_paginas
http://oer.avu.org/bitstream/handle/123456789/40/Industrial%20Chemistry.pdf?sequence=4	168
http://oer.avu.org/bitstream/handle/123456789/55/Mechanics.pdf?sequence=4	165
http://oer.avu.org/bitstream/handle/123456789/56/Mathematical%20Physics%202%20-%20Readings.pdf?sequence=1	165
http://oer.avu.org/bitstream/handle/123456789/25/Electronics.pdf?sequence=4	165
http://oer.avu.org/bitstream/handle/123456789/66/ICT%20Integration%20in%20Biology.pdf?sequence=1	163
http://oer.avu.org/bitstream/handle/123456789/52/Volumetric%20Chemical%20Analysis.pdf?sequence=5	149
http://oer.avu.org/bitstream/handle/123456789/32/Cell%20Biology%20and%20Genetics.pdf?sequence=4	144
http://oer.avu.org/bitstream/handle/123456789/50/Plant-animal-physiologyVFreadings.pdf?sequence=1	142
http://oer.avu.org/bitstream/handle/123456789/58/Geometrical%20Optics%20and%20Physical%20Optics%20-%20Readings.pdf?sequence=1	138
http://oer.avu.org/bitstream/handle/123456789/51/Animal-diversityWreadings.pdf?sequence=1	137
http://oer.avu.org/bitstream/handle/123456789/41/Inorganic%20Chemistry.pdf?sequence=4	133
http://oer.avu.org/bitstream/handle/123456789/53/Calculus.pdf?sequence=3	132

Figura 16: Número de páginas de cada OER

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.2. Extracción de texto

El propósito de la extracción de texto es tomar parte del contenido del OER para su posterior procesamiento.

4.4.2.1. Ejecución

Extraer el texto de los recursos educativos abiertos es la base fundamental de este proceso para esto se utiliza el número de páginas de las cuales se obtendrá el texto, valores mencionados en la tabla 11.

Para realizar la extracción del texto se utiliza la librería pyPDF mencionada anteriormente, esta librería nos permite extraer el texto de la página que se especifique.

4.4.2.2. Resultados

Es conveniente presentar resultados que sustenten la creación de los rangos de valores como se estableció en la tabla 11. (Ver sección 4.4.1)

Se han tomado los OER's "Analysis%202.pdf" y "Calculus" del OCW del OCW "African Virtual University", en las figuras siguientes se muestra la cantidad de paginas de las cuales se extraerá el texto y una porción del texto extraído.

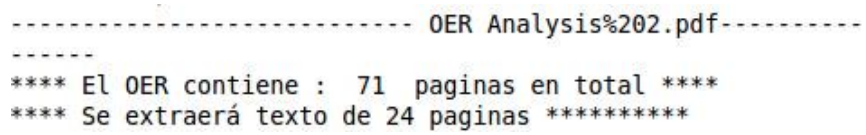


Figura 17: Cantidad de páginas de las cuales se extraerá texto del OER Analysis%202.pdf

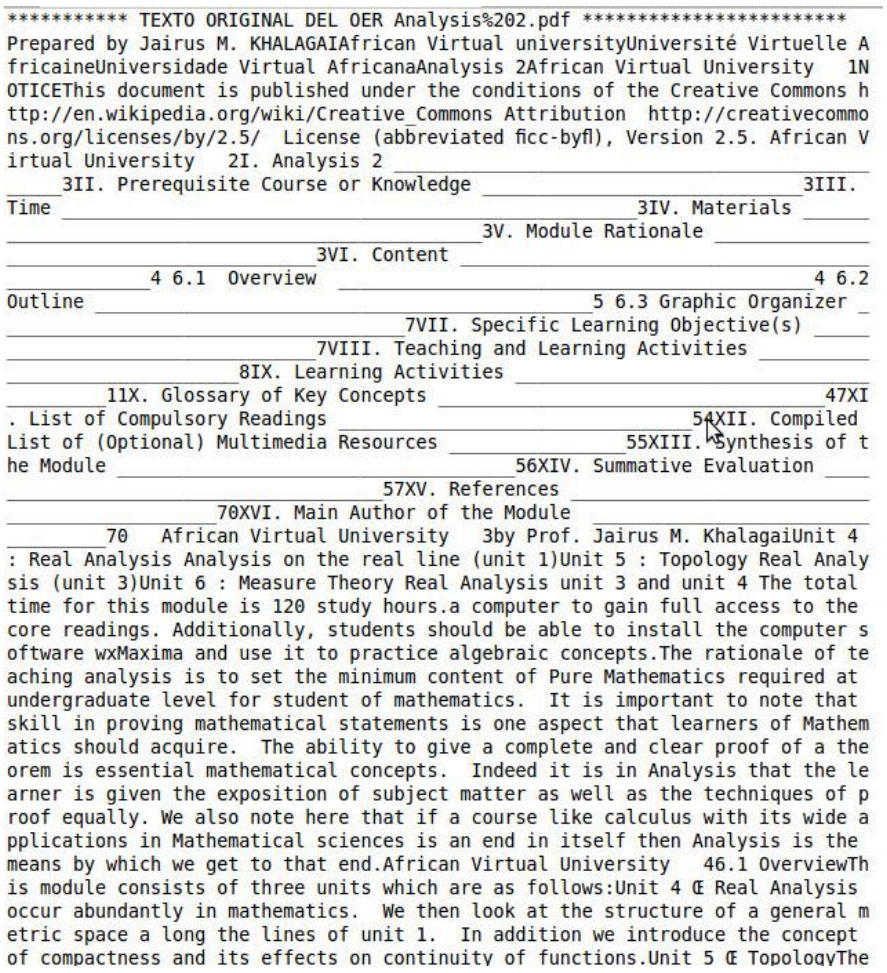


Figura 18: Texto extraído del OER "Analysis%202.pdf"

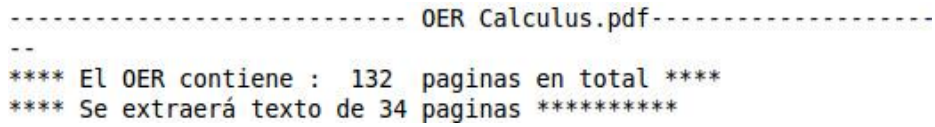
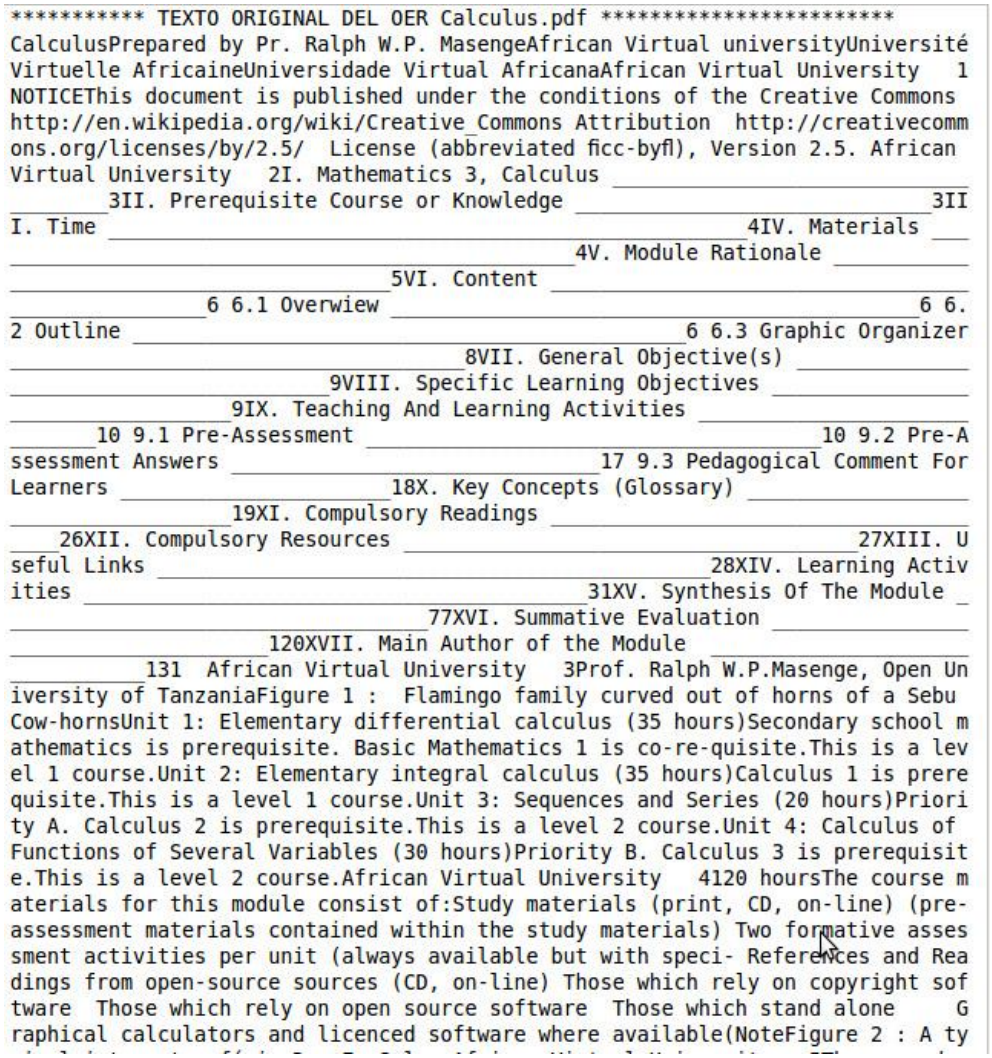


Figura 19: Cantidad de paginas de las cuales se extraerá texto del OER Calculus.pdf



********** TEXTO ORIGINAL DEL OER Calculus.pdf **********
CalculusPrepared by Pr. Ralph W.P. MasengeAfrican Virtual universityUniversité Virtuelle AfricaineUniversidade Virtual AfricanaAfrican Virtual University 1 NOTICEThis document is published under the conditions of the Creative Commons http://en.wikipedia.org/wiki/Creative_Commons Attribution http://creativecommons.org/licenses/by/2.5/ License (abbreviated ficc-byfl), Version 2.5. African Virtual University 2I. Mathematics 3, Calculus _____ _____3II. Prerequisite Course or Knowledge _____3III. Time _____4IV. Materials _____4V. Module Rationale _____ _____5VI. Content _____ _____6 6.1 Overwiew _____6 6.2 Outline _____6 6.3 Graphic Organizer _____8VII. General Objective(s) _____ _____9VIII. Specific Learning Objectives _____ _____9IX. Teaching And Learning Activities _____ _____10 9.1 Pre-Assessment _____10 9.2 Pre-Assessment Answers _____17 9.3 Pedagogical Comment For Learners _____18X. Key Concepts (Glossary) _____ _____19XI. Compulsory Readings _____ _____26XII. Compulsory Resources _____27XIII. Useful Links _____28XIV. Learning Activities _____31XV. Synthesis Of The Module _____77XVI. Summative Evaluation _____ _____120XVII. Main Author of the Module _____ _____131 African Virtual University 3Prof. Ralph W.P.Masenge, Open University of TanzaniaFigure 1 : Flamingo family curved out of horns of a Sebu Cow-hornsUnit 1: Elementary differential calculus (35 hours)Secondary school mathematics is prerequisite. Basic Mathematics 1 is co-re-quisite.This is a level 1 course.Unit 2: Elementary integral calculus (35 hours)Calculus 1 is prerequisite.This is a level 1 course.Unit 3: Sequences and Series (20 hours)Priority A. Calculus 2 is prerequisite.This is a level 2 course.Unit 4: Calculus of Functions of Several Variables (30 hours)Priority B. Calculus 3 is prerequisite.This is a level 2 course.African Virtual University 4120 hoursThe course materials for this module consist of:Study materials (print, CD, on-line) (pre-assessment materials contained within the study materials) Two formative assessment activities per unit (always available but with speci- References and Readings from open-source sources (CD, on-line) Those which rely on copyright software Those which rely on open source software Those which stand alone G raphical calculators and licenced software where available(NoteFigure 2 : A ty

Figura 20: Texto extraído del OER “Calculus.pdf”

El script que contiene esta tarea se lo puede observar en el Anexo 4.

4.4.3. Normalización de texto

El propósito de la normalización consiste en convertir la codificación del texto original a codificación Unicode, y cambiar el texto extraído a minúsculas con el propósito de eliminar posibles errores relacionados con estas características (tipo de codificación del texto, sensibilidad a mayúsculas ó minúsculas).

4.4.3.1. Ejecución

Al mencionar el tipo de codificación de texto nos referimos al método que permite convertir un caracter de un lenguaje natural (alfabeto o silabario) en un símbolo de otro sistema de representación con la finalidad de facilitar el almacenamiento de texto en computadoras, la tabla siguiente menciona los tres más utilizados, cabe resaltar que la codificación que se utilizará es *Unicode*:

Tabla 13: Tipos de codificación más usados

CODIFICACIÓN	DESCRIPCIÓN	CANTIDAD DE CARACTERES
ASCII	Es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.	Utiliza 7 bits para representar caracteres
ASCII Extendido	Se denomina ASCII extendido a cualquier juego de caracteres de 8 bits en el cual los códigos 32 a 126 coinciden con los caracteres imprimibles de ASCII, así como los caracteres comúnmente llamados "de espacio", estos son los códigos de control de 8 a 13.	Juego de caracteres de 8 bits
Unicode	*Es un estándar industrial cuyo objetivo es proporcionar el medio por el cual un texto en cualquier forma o idioma puede ser codificado para el uso informático. Cubre la mayor parte de las escrituras usada actualmente.*	*90000 caracteres codificados*

Fuente: http://techtastico.com/post/tipos-de-codificacion-de-caracteres/

Para realizar la normalización del texto se utiliza funciones como:

- *Unicodedata* para codificación y
- *lower* para convertir las palabras a minúsculas.

4.4.3.2. Resultados

La imagen siguiente muestra un contraste entre una parte del texto extraído y luego de haber sido normalizado.

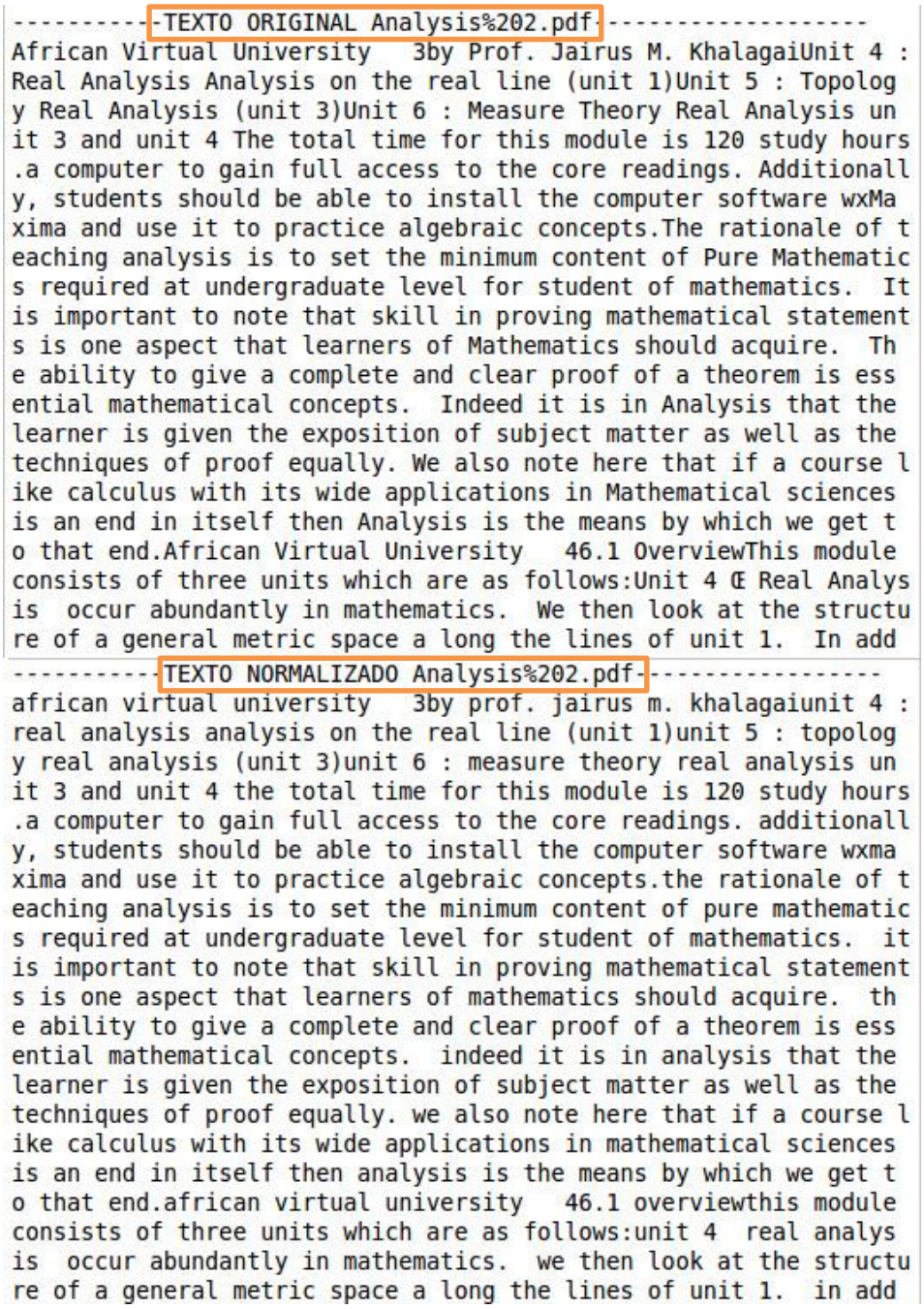


Figura 21: Extracto del Texto original, y texto normalizado del OER "Analysis%202.pdf"

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.4. Identificación del idioma

El propósito de esta tarea es identificar el idioma del OER con la finalidad de tomar en cuenta solamente aquellos que se encuentren en idioma ingles.

4.4.4.1. Ejecución

Los repositorios OCW al igual que los OER se encuentran en distintos idiomas como español, catalán, inglés, italiano, chino mandarín entre otros; se utilizarán únicamente los recursos que se encuentren en idioma *inglés.*

La herramienta principal que interviene es NLTK específicamente con el corpus Stopwords el mismo que contiene adjetivos, adverbios, pronombres en 11 idiomas, características que contribuyen en la identificación del idioma del OER.

Tabla 14: Corpus Wordnet

CORPUS	COMPILER	CONTENTS
Stopwords Corpus	Porter et al	2,400 stopwords for 11 languages

Fuente: (Natural Language Processing with Python, 2012)

4.4.4.2. Resultados

En la imagen se puede observar el idioma correspondiente a cada OER.

s	o
http://oer.avu.org/bitstream/handle/123456789/84/Graphics%20and%20Information%20Management%20Systems.pdf?sequence=3	english
http://oer.avu.org/bitstream/handle/123456789/59/Quantum%20Mechanics.pdf?sequence=1	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-food-trade-theory/314_0506%20Ch4%20Heckscher%20Ohlin_4.pdf	english
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_root.pdf	english
http://oer.avu.org/bitstream/handle/123456789/27/Properties%20of%20Matter.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/23/Atomic%20Physics.pdf?sequence=4	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/b2e8bc31c9c8-d569c131-bc0f-bd84d574b860/2.pdf	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604c870c808b860/5.pdf	english
http://oer.avu.org/bitstream/handle/123456789/76/Educational%20Communication.pdf?sequence=1	english
http://oer.avu.org/bitstream/handle/123456789/45/Organic%20Chemistry%201.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/25/Electronics.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/78/Educational%20Evaluation%20and%20Testing.pdf?sequence=1	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604c870c808b860/4.pdf	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-trade-negotiations-1/067_global_0304.pdf	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-trade-negotiations-1/067_0325%20ADD_2.pdf	english
http://oer.avu.org/bitstream/handle/123456789/46/Organic%20Chemistry%202.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/18/Basic%20Mathematics.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/38/Separation%2c%20Electroanalytical%20and%20Spectrometric%20Techniques.pdf?sequence=4	english
http://oer.avu.org/bitstream/handle/123456789/56/Mathematical%20Physics%202%20-%20Readings.pdf?sequence=1	english
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-food-trade-theory/314_0415%20Special_2.pdf	english
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_intro_2nd.pdf	english

Figura 22: Idioma de los OER's

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.5. Tokenización

El propósito de la tokenización es dividir el texto en tokens o palabras.

4.4.5.1. Ejecución

El objetivo de la tokenización es facilitar el procesamiento del texto extraído dividiendo el texto en tokens o palabras, aunque también se puede tokenizar a nivel de párrafos, u oraciones (Natural Language Processing with Python, 2012); el siguiente ejemplo nos aclara este concepto:

Oración = "Cuando alguien busca un URI, se debe proveer información útil utilizando las normas (RDF, SPARQL)"

Tokens = ['Cuando', 'alguien', 'busca', 'un', 'URI', ',', 'se', 'debe', 'proveer', 'información', 'útil', 'utilizando', 'las', 'normas', '(', 'RDF', '.', 'SPARQL', ')']

Como se puede observar la tokenización divide la oración en palabras incluidos los signos de puntuación, admiración, etc., acción que facilita y da pauta a muchas tareas de procesamiento de texto.

Anteriormente se mencionó sobre los corpus disponibles, para esta tarea se utiliza uno de los corpus más comunes TreeBank.

Tabla 15: Corpus TreeBank

CORPUS	COMPILER	CONTENTS
Dependency Treebank	Narad	Dependency parsed version of Penn Treebank sample

Fuente: (Natural Language Processing with Python, 2012)

4.4.5.2. Resultado

Un extracto del texto tokenizado del OER "Analysis%202.pdf" se observa en el siguiente gráfico.

Figura 23: Extracto de la tokenización del texto extraído del OER “Analysis%202.pdf”

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.6. Tagging

El propósito de tagging o etiquetado es asignar a cada palabra o token una etiqueta que indique el tipo de palabra que es.

4.4.6.1. Ejecución

El objetivo del etiquetado o tagging es asignar a cada token una etiqueta que indique la clasificación de la palabra en clases, como clases de palabras se puede mencionar : adjetivos, adverbios, sustantivos, verbos, conjunciones, pronombres, preposiciones, entre otros. Las clases de palabras existentes en el idioma inglés son:

Tabla 16: Clasificación de las palabras en clases

TAG	SIGNIFICADO	EJEMPLOS
ADJ	Adjective	new, good, high, special, big, local
ADV	Adverb	really, already, still, early, now
CNJ	Conjunction	and, or, but, if, while, although
DET	Determiner	the, a, some, most, every, no
EX	Existential	there, there's
FW	foreignword	dolce, ersatz, esprit, quo, maitre
MOD	modal verb	will, can, would, may, must, should
N	Noun	year, home, costs, time, education
NP	propernoun	Alison, Africa, April, Washington
NUM	Number	twenty-four, fourth, 1991, 14:24
PRO	Pronoun	he, their, her, its, my, I, us
P	Preposition	on, of, at, with, by, into, under
TO	the word to	to
UH	Interjection	ah, bang, ha, whee, hmpf, oops
V	Verb	is, has, get, do, make, see, run
VD	past tense	said, took, told, made, asked
VG	present participle	making, going, playing, working
VN	past participle	given, taken, begun, sung
WH	wh determiner	who, which, when, what, where, how

Fuente: (Natural Language Processing with Python, 2012)

Con la finalidad de aclarar este tema se presenta el siguiente ejemplo tomado de (Natural Language Processing with Python, 2012), el etiquetado se lo realiza utilizando las clases de palabras del idioma inglés:

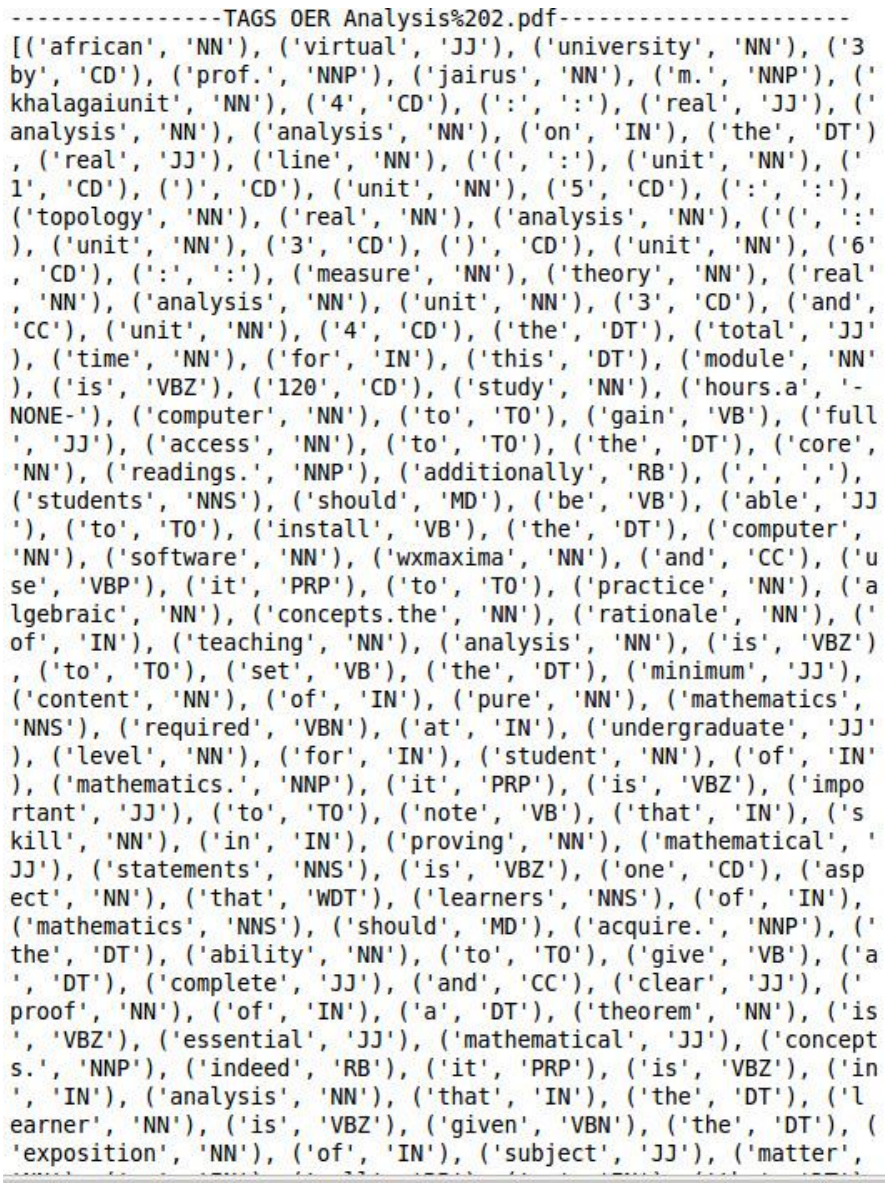
Tokens = ['At', 'eight', "o'clock", 'on', 'Thursday', 'morning','Arthur', 'did', "n't", 'feel', 'very', 'good', '.']

Tags= [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), ('Arthur', 'NNP'), ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')]

Para realizar el tagging o etiquetado se utiliza ***nltk.pos_tag***, el proceso se desarrolla utilizando los tokens obtenidos previamente, a estos se les agrega el tag dependiendo de la clase a la que pertenece.

4.4.6.2. Resultados

Un extracto del tagging del texto extraído del OER "Analysis%202.pdf" se observa a continuación:



```
----------------TAGS OER Analysis%202.pdf----------------------
[('african', 'NN'), ('virtual', 'JJ'), ('university', 'NN'), ('3
by', 'CD'), ('prof.', 'NNP'), ('jairus', 'NN'), ('m.', 'NNP'), ('
khalagaiunit', 'NN'), ('4', 'CD'), (':', ':'), ('real', 'JJ'), ('
analysis', 'NN'), ('analysis', 'NN'), ('on', 'IN'), ('the', 'DT')
, ('real', 'JJ'), ('line', 'NN'), ('(', ':'), ('unit', 'NN'), ('
1', 'CD'), (')', 'CD'), ('unit', 'NN'), ('5', 'CD'), (':', ':'),
('topology', 'NN'), ('real', 'NN'), ('analysis', 'NN'), ('(', ':'
), ('unit', 'NN'), ('3', 'CD'), (')', 'CD'), ('unit', 'NN'), ('6'
, 'CD'), (':', ':'), ('measure', 'NN'), ('theory', 'NN'), ('real'
, 'NN'), ('analysis', 'NN'), ('unit', 'NN'), ('3', 'CD'), ('and',
'CC'), ('unit', 'NN'), ('4', 'CD'), ('the', 'DT'), ('total', 'JJ'
), ('time', 'NN'), ('for', 'IN'), ('this', 'DT'), ('module', 'NN'
), ('is', 'VBZ'), ('120', 'CD'), ('study', 'NN'), ('hours.a', '-
NONE-'), ('computer', 'NN'), ('to', 'TO'), ('gain', 'VB'), ('full
', 'JJ'), ('access', 'NN'), ('to', 'TO'), ('the', 'DT'), ('core',
'NN'), ('readings.', 'NNP'), ('additionally', 'RB'), (',', ','),
('students', 'NNS'), ('should', 'MD'), ('be', 'VB'), ('able', 'JJ
'), ('to', 'TO'), ('install', 'VB'), ('the', 'DT'), ('computer',
'NN'), ('software', 'NN'), ('wxmaxima', 'NN'), ('and', 'CC'), ('u
se', 'VBP'), ('it', 'PRP'), ('to', 'TO'), ('practice', 'NN'), ('a
lgebraic', 'NN'), ('concepts.the', 'NN'), ('rationale', 'NN'), ('
of', 'IN'), ('teaching', 'NN'), ('analysis', 'NN'), ('is', 'VBZ')
, ('to', 'TO'), ('set', 'VB'), ('the', 'DT'), ('minimum', 'JJ'),
('content', 'NN'), ('of', 'IN'), ('pure', 'NN'), ('mathematics',
'NNS'), ('required', 'VBN'), ('at', 'IN'), ('undergraduate', 'JJ'
), ('level', 'NN'), ('for', 'IN'), ('student', 'NN'), ('of', 'IN'
), ('mathematics.', 'NNP'), ('it', 'PRP'), ('is', 'VBZ'), ('impo
rtant', 'JJ'), ('to', 'TO'), ('note', 'VB'), ('that', 'IN'), ('s
kill', 'NN'), ('in', 'IN'), ('proving', 'NN'), ('mathematical', '
JJ'), ('statements', 'NNS'), ('is', 'VBZ'), ('one', 'CD'), ('asp
ect', 'NN'), ('that', 'WDT'), ('learners', 'NNS'), ('of', 'IN'),
('mathematics', 'NNS'), ('should', 'MD'), ('acquire.', 'NNP'), ('
the', 'DT'), ('ability', 'NN'), ('to', 'TO'), ('give', 'VB'), ('a
', 'DT'), ('complete', 'JJ'), ('and', 'CC'), ('clear', 'JJ'), ('
proof', 'NN'), ('of', 'IN'), ('a', 'DT'), ('theorem', 'NN'), ('is
', 'VBZ'), ('essential', 'JJ'), ('mathematical', 'JJ'), ('concept
s.', 'NNP'), ('indeed', 'RB'), ('it', 'PRP'), ('is', 'VBZ'), ('in
', 'IN'), ('analysis', 'NN'), ('that', 'IN'), ('the', 'DT'), ('l
earner', 'NN'), ('is', 'VBZ'), ('given', 'VBN'), ('the', 'DT'), (
'exposition', 'NN'), ('of', 'IN'), ('subject', 'JJ'), ('matter',
```

Figura 24: Asignación de Tag para cada token del OER "Analysis%202.pdf"

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.7. Identificación de tokens representativos

Al momento de leer un texto ya sea digital o impreso se hace una revisión rápida o búsqueda de palabras que nos interesan, de esta manera se obtiene una idea general sobre el tema que aborda el documento. Tomando este principio como base podemos establecer *el propósito de esta tarea; identificar los tokens representativos del texto que permita adquirir una idea sobre el tema que aborda el OER.*

4.4.7.1. Ejecución

Los tokens o palabras representativas es un parámetro importante durante el proceso de desambiguación; para considerar un token como representativo debe cumplir los siguientes lineamientos:

- No ser un stopword conocidos también como: adjetivos, pronombres, artículos, etc.
- Se considera tokens representativos aquellos que superen 3 repeticiones en el texto extraído.

Se utiliza la clase **FreqDist** (Natural Language Processing with Python, 2012), que utiliza conceptos de estadística básica, para este caso en particular empleamos una distribución de frecuencias. Se realiza un conteo de las repeticiones de cada token.

Word Tally

the	𝍸 𝍸 𝍸 ||||
been	𝍸 𝍸 |
message	||||
persevere	|
nation	𝍸 |||

Figura 25: Número de repeticiones de un token.
Fuente: (Natural Language Processing with Python, 2012)

4.4.7.2. Resultados

Las siguientes imágenes muestran los tokens representativos de los OER's "Analysis%202.pdf", "Calculus,pdf" y "Multimedia%20Design.pdf"

```
********** PALABRAS REPRESENTATIVAS DEL OER Analysis%202.pdf **********
['1', '2', '3', '4', '5', '6', 'a.', 'able', 'abstract', 'activity', 'african'
, 'also', 'analysis', 'b', 'based', 'bn', 'c', 'called', 'case', 'closed', 'cl
osure', 'compact', 'concept', 'concepts', 'contains', 'continuity', 'continuou
s', 'convergence', 'copies', 'cover', 'd', 'd.', 'denoted', 'development', 'di
fferent', 'distance', 'distance.', 'dx', 'dy', 'end', 'endowed', 'essential',
'even', 'every', 'examples', 'f', 'following', 'function', 'function.', 'furni
ture', 'general', 'geometric', 'give', 'given', 'gives', 'hours', 'house', 'if
f', 'ii', 'iii', 'important', 'integral', 'interior', 'iv', 'learner', 'learni
ng', 'lebesgue', 'let', 'level', 'like', 'limit', 'line', 'look', 'mathematica
l', 'mathematics', 'measurable', 'measure', 'metric', 'module', 'n', 'neighbou
rhood', 'non-empty', 'note', 'o', 'one', 'open', 'p', 'particular', 'point', '
points', 'priority', 'properties', 'readings', 'real', 'riemann', 'said', 'set
', 'sets', 'show', 'space', 'spaces', 'spaces.', 'statements', 'story', 'struc
ture', 'study', 'subset', 'subsets', 'symmetrical', 'terms', 'theorem', 'theor
y', 'thus', 'topological', 'topology', 'true', 'unit', 'university', 'virtual'
, 'walls', 'well', 'without', 'x', 'xx', 'y', 'yx']
```

Figura 26: Palabras representativas del OER "Analysis%202.pdf"

```
********** PALABRAS REPRESENTATIVAS DEL OER Calculus.pdf **********
['0', '07.11.06', '1', '1,1', '1,2', '100', '2', '2.', '3', '35', '4', '5', '6
', '9', 'a.', 'ab', 'able', 'access', 'activities', 'activities.', 'activity',
'african', 'also', 'always', 'applications', 'approaches', 'area', 'available'
, 'b', 'ba', 'basic', 'brigham', 'c', 'calculus', 'calculus.', 'called', 'cd',
'center', 'check', 'choose', 'click', 'clicking', 'commands', 'complete', 'com
puter', 'concept', 'concepts', 'contain', 'continuity', 'continuous', 'converg
e', 'convergence', 'cos', 'course', 'course.unit', 'critical', 'curve', 'cxlim
lxf', 'd', 'derivative', 'determine', 'different', 'differentiability', 'diffe
rential', 'differentiation', 'differentiation.', 'discontinuity', 'domain', 'd
ouble', 'elementary', 'entry', 'equal', 'equation', 'example', 'explore', 'f',
'following', 'function', 'function.', 'functions', 'functions.', 'fx', 'get',
'given', 'graph', 'graphs', 'hand', 'help', 'hours', 'however', 'http', 'ict',
'including', 'independent', 'install', 'integral', 'integration', 'integration
.', 'interactive', 'interval', 'introduce', 'key', 'knowledge', 'known', 'l',
'large', 'learner', 'learning', 'level', 'lim', 'limit', 'limit.', 'limits', '
link', 'links', 'look', 'make', 'manual', 'material', 'materials', 'mathematic
al', 'mathematics', 'maths', 'mathworld', 'maxima', 'maximum', 'may', 'module'
, 'multipliers', 'need', 'number', 'numbers', 'obtained', 'often', 'one', 'ope
n', 'p', 'p.', 'page', 'partial', 'point', 'points', 'power', 'pp', 'pre-asses
sment', 'prerequisite', 'prerequisite.this', 'press', 'pressure', 'priority',
'provided', 'q', 'r', 'radius', 'range', 'read', 'readings', 'real', 'relative
', 'right', 'said', 'school', 'secondary', 'sections', 'see', 'sequence', 'seq
uences', 'series', 'set', 'several', 'shown', 'sides', 'sin', 'single', 'softw
are', 'software.', 'solution', 'solve', 'source', 'started', 'sum', 'sure', 's
ystem', 'tangent', 'temperature', 'tests', 'thn', 'three', 'time', 'trapezium'
, 'triangle', 'try', 'two', 'type', 'understanding', 'unit', 'units', 'univers
ity', 'use', 'used', 'using', 'v', 'value', 'values', 'variable', 'variables',
'version', 'virtual', 'visited', 'whenever', 'without', 'wxmaxima', 'x', 'xf',
'y', 'young', 'yxf', 'z']
```

Figura 27: Palabras representativas del OER "Calculus.pdf"

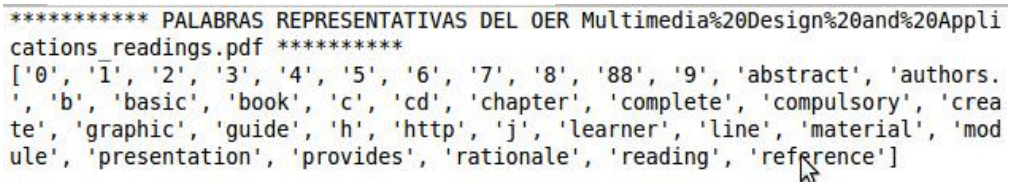


Figura 28: Palabras representativas del OER “Multimedia%20Design.pdf”

Si se observa las palabras representativas de cada OER mostradas en las imágenes se puede identificar palabras que son comunes específicamente entre los recursos “Analysis%202” y “Calculus” como: “real”, “mathematics”, “unit”, “function”, “integral” al leer estas palabras se puede tener una idea sobre el tema que tratan; sin embargo se necesita conocer si sus significados son los mismos, para esto se realizará el procesamiento de información tareas que se abordarán mas adelante.

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 7.

4.4.8. Extracción de Entidades

Una entidad es un objeto real o abstracto del cual se puede recoger información de fuentes externas, *el propósito de la extracción de entidades es identificar aquellas palabras que son parte de uno de los grupos presentados en la tabla 15, con la finalidad de utilizar estos valores para crear relaciones entre OER’s:*

Tabla 17: Tipos de Entidades más usadas

TIPO DE ENTIDAD	EJEMPLOS
ORGANIZATION	Georgia-Pacific Corp., WHO
PERSON	Eddy Bonte, President Obama
LOCATION	Murray River, Mount Everest
DATE	June, 2008-06-29
TIME	two fifty a m, 1:30 p.m.
MONEY	175 million Canadian Dollars, GBP 10.40
PERCENT	twenty pct, 18.75 %
FACILITY	Washington Monument, Stonehenge
GPE	South East Asia, Midlothian

Fuente: (Natural Language Processing with Python, 2012)

4.4.8.1. Ejecución

Se utiliza como valores de entrada a los tokens y tags obtenidos anteriormente; como resultado nos retorna un árbol estructurado en base a los tags, al encontrar una entidad la incluye al grupo al que pertenece, por ejemplo:

Oración = "At eight o'clock on Thursday morning**.** Arthur didn't feel very good."

Tags = [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), ('Arthur', 'NNP'), ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')]

Entidades = Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'), Tree('PERSON', [('Arthur', 'NNP')]), ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')])

Los grupos de entidades mencionados anteriormente están creados en la clase ***ne_chunck*** de NLTK (Natural Language Processing with Python, 2012) librería utilizada para extraer entidades.

4.4.8.2. Resultados

En nuestro caso las entidades extraídas de los OER's "Analysis%202.pdf", "Calculus.pdf", "Multimedia%20Design" se muestran en las gráficas siguientes.

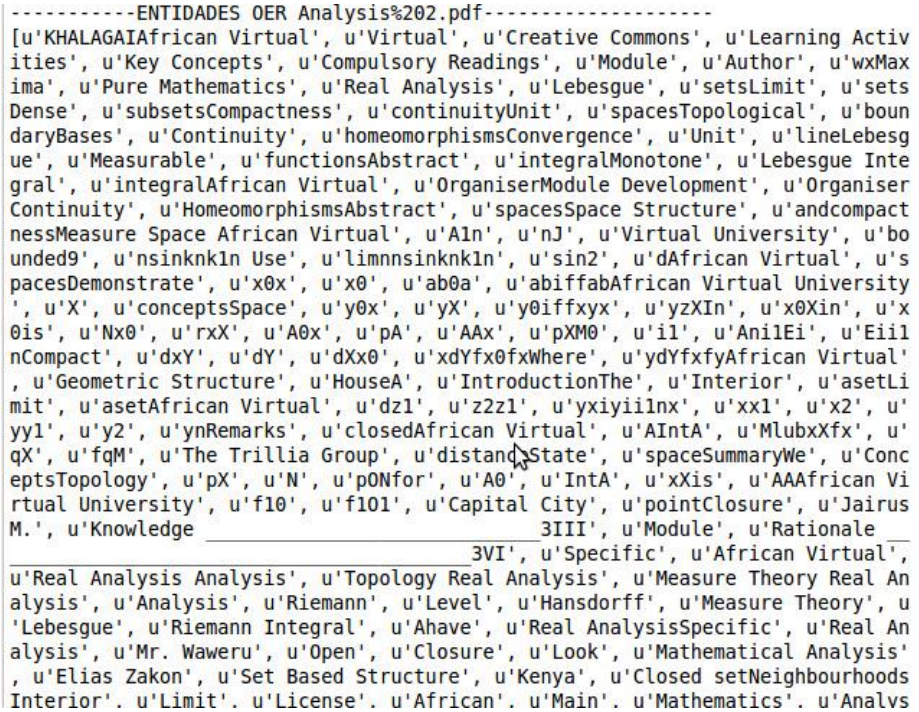


Figura 29: Entidades extraídas del OER “Analysis%202.pdf”

Figura 30: Entidades extraídas del OER "Calculus.pdf”

```
----------ENTIDADES OER Multimedia%20Design%20and%20Applications_readings.pdf
--------------------
[u'COMPULSORY', u'READINGS Reading', u'chapter8', u'ICT', u'Carrera', u'Adiels
son', u'Barnes', u'Bonde', u'Worthington', u'Save Line', u'Line', u'Belzunce',
u'Hall', u'Kampa', u'Rationale']
```

Figura 31: Entidades extraídas del OER "Multimedia%20Design"

Al igual que las palabras representativas, las entidades tambien permiten establecer una idea sobre el tema que trata el OER, existen entidades en común entre los OER's "Calculus" y "Analysis%202" como "African Virtual", "Creative Commons", "Riemman", entre otros; el script que realiza esta tarea se lo puede observar en el Anexo 8.

4.5. Procesamiento de Información

Una vez extraída la data de los OER's el siguiente proceso a desarrollar es el procesamiento de esta información, *este proceso contempla y tiene como propósito resolver el problema de ambigüedad semántica, y posteriormente crear relaciones entre los recursos educativos abiertos usando como base la data desambigua.* Las tareas establecidas para la resolución de este problema son:

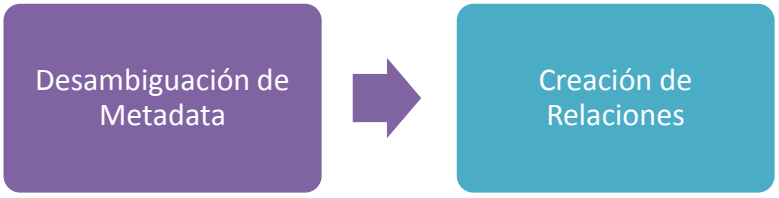


Figura 32: Tareas a Realizar para Crear Relaciones entre OER's

4.5.1. Desambiguación de metadata

El propósito de esta tarea es establecer el correcto significado para una palabra ambigua tomando en cuenta su contexto.

4.5.1.1. Ejecución

Si tenemos un OER que aborda el tema de *aplicaciones* de software, se debe establecer los parámetros necesarios para que dicho OER pueda ser relacionado con otro que aborde el mismo tema, y eliminar las posibilidades de relacionarse con OER's

que aborden el tema de *aplicaciones* de formulas químicas; ambos temas tratan sobre aplicaciones pero sus contextos son completamente diferentes y sin relación.

Los tokens representativos al igual que las entidades extraídas del OER nos dan una idea sobre el tema que trata, sin embargo al ser solamente palabras sueltas se convierten en palabras ambiguas, lo que permite abordar el tema de ambigüedad semántica que se presenta cuando las palabras tienen múltiples significados (ver secciones 1.5.2 y 1.5.3), este tema es el core de esta tarea.

Obligatoriamente se necesita tener un contexto para identificar el significado idóneo de las palabras y poseer una perspectiva clara y concisa para posteriormente relacionar al OER. Es preciso someter al texto extraído a un proceso de desambiguación con la finalidad de establecer el significado correcto para cada palabra, el siguiente gráfico indica el proceso para la desambiguación de palabras.

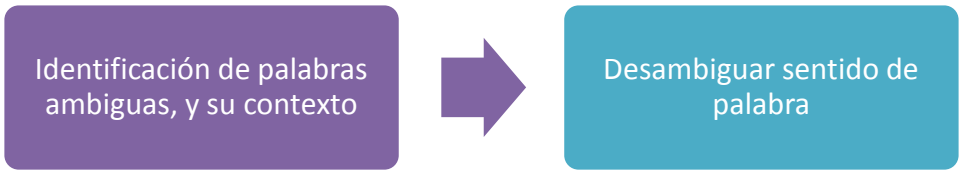


Figura 33: Proceso Desambiguación de metadata del OER

4.5.1.2. Identificación de palabras ambiguas y su contexto

El propósito de esta tarea y como su nombre lo menciona es identificar las palabras ambiguas en el texto extraído y el contexto correspondiente.

4.5.1.2.1. *Ejecución*

Como menciona (Torres, 2009) “Cualquier palabra que se utilice para comunicarse tiene dos o más posibles interpretaciones, llamadas sentidos. Para entender correctamente un texto, el lector (humano) o programa de computadora debe ser capaz de determinar el sentido adecuado para cada palabra en el texto.

Debido a que existe ambigüedad aun para los humanos, su solución no es solo lograr la asignación del sentido único por palabra en el análisis de textos, sino eliminar la gran cantidad de variantes que normalmente existen."

Para el desarrollo de la fase de identificación de palabras ambiguas y su contexto es necesario:

- **Dividir el texto:** *es necesario dividir el texto en párrafos, los párrafos en oraciones, y las oraciones en tokens,* ***con el propósito de establecer el contexto para la palabra ambigua****.*
- **Identificar la palabra ambigua:** para identificar la palabra ambigua se debe verificar que esta palabra posea más de un significado y sea parte de los tokens representativos.

Las herramientas que nos facilita la realización de esta tarea son:

- **nltk.sent_tokenize** que permite la división del texto extraído en párrafos, y este en oraciones. Para dividir en tokens se reutiliza funciones ya desarrolladas (ver sección 3.5.4).
- **wordnet.synsets() y len()** para comprobar si posee más de un significado

4.5.1.2.2. *Resultados*

Las imágenes siguientes muestran un extracto de la división del texto extraído en párrafos, oraciones, tokens y las palabras ambiguas identificadas de los OER's "Analysis%202.pdf", "Calculus" y "Multimedia%20Design".

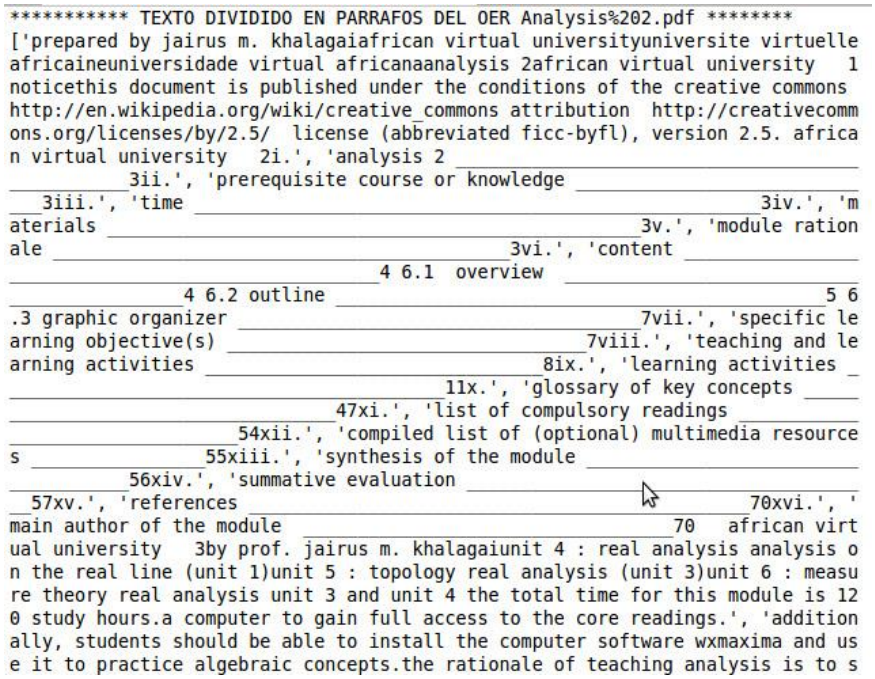


Figura 34: División del texto en párrafos del OER “Analysis%202.pdf”

Figura 35: División en oraciones y tokens del OER “Analysis%202.pdf”

```
********** ORACION **********
['finally a brief comparison of the lebesgue integral and the well known riema
nn integral is also essential in this unit.african virtual university   56.2
outline : syllabusunit 4 - real analysis (30 hours)level priority unit 1 is th
e pre-requisitenieghbourhoods, interior points and open setslimit points and c
losed setsdense and compact subsetscompactness and continuityunit 5  topology
(30 hours)level priority unit 3 is the pre-requisitereview of metric spacesto
pological spaces, neighbourhoods, interior and open setslimit points closed s
ets, closure and boundarybases, relative and product topologies continuity and
homeomorphismsconvergence and hansdorff axiom.unit 6 - measure theory (40 hou
rs)level priority unit 3 and unit 4 are the pre-requisitelebesgue outer measu
re and lebesgue measure on real linelebesgue measurable subsets of measurable
spaces and measurable functionsabstract measure spaces and abstract integralm
onotone convergence theorem, fatoutms lemma and lebesgue dominated convergenc
e theoremrelation between riemann integral and lebesgue integralunit 7 - the l
ebesgue integral (20 hours)level priority unit 6 is the pre-requisitestate so
me properties of lebesgue integralverify some properties of the lebesgue integ
ralafrican virtual university   66.3 graphic organisermodule development templ
ate5graphic organisercontinuity and homeomorphismsabstract integral measurabl
e spacemeasurablefunctions structure of atopological spacefunctions ontopolog
ical spacesspace structure of ametric spacefunctions onmetric spacescontinuit
y andcompactnessmeasure space african virtual university   7you should be abl
e to:1. demonstrate understanding of basic concepts and principles of mathemat
ical analysis.2.']
~~~~~~ Palabra ambigüa detectada --> integral
~~~~~~ Palabra ambigüa detectada --> well
~~~~~~ Palabra ambigüa detectada --> riemann
~~~~~~ Palabra ambigüa detectada --> integral
~~~~~~ Palabra ambigüa detectada --> also
~~~~~~ Palabra ambigüa detectada --> essential
~~~~~~ Palabra ambigüa detectada --> virtual
~~~~~~ Palabra ambigüa detectada --> university
~~~~~~ Palabra ambigüa detectada --> real
~~~~~~ Palabra ambigüa detectada --> analysis
~~~~~~ Palabra ambigüa detectada --> hours
~~~~~~ Palabra ambigüa detectada --> level
~~~~~~ Palabra ambigüa detectada --> priority
~~~~~~ Palabra ambigüa detectada --> unit
~~~~~~ Palabra ambigüa detectada --> interior
~~~~~~ Palabra ambigüa detectada --> points
```

Figura 36: Identificación de palabras ambiguas del OER “Analysis%202.pdf”

```
********** TEXTO DIVIDIDO EN PARRAFOS DEL OER Calculus.pdf ********
['calculusprepared by pr.', 'ralph w.p.', 'masengeafrican virtual universityun
iversite virtuelle africaineuniversidade virtual africanaafrican virtual unive
rsity   1noticethis document is published under the conditions of the creative
commons http://en.wikipedia.org/wiki/creative_commons attribution  http://crea
tivecommons.org/licenses/by/2.5/  license (abbreviated ficc-byfl), version 2.5
. african virtual university   2i.', 'mathematics 3, calculus _______________
_____________________3ii.', 'prerequisite course or knowledge _______________
_____________3iii.', 'time _____________________________________________
_4iv.', 'materials _____________________________________________4v.', 'mod
ule rationale _____________________________________5vi.', 'content ______
_____________________________6 6.1 overwiew _____________________
_____________________6 6.2 outline _____________________________
____6 6.3 graphic organizer _____________________________8vii.', 'gen
eral objective(s) _____________________________9viii.', 'specific l
earning objectives _____________________________9ix.', 'teaching and lear
ning activities _____________________10 9.1 pre-assessment _____________
_____________10 9.2 pre-assessment answers _____________________
_______17 9.3 pedagogical comment for learners _____________________18x.', '
key concepts (glossary) _____________________19xi.', 'compulsor
y readings _____________________26xii.', 'compulsory resource
s _____________________27xiii.', 'useful links _____________
_____________________28xiv.', 'learning activities _____________
_____________________31xv.', 'synthesis of the module _____________
_____________77xvi.', 'summative evaluation _____________
___120xvii.', 'main author of the module _____________131
african virtual university   3prof.', 'ralph w.p.masenge, open university of t
anzaniafigure 1 :  flamingo family curved out of horns of a sebu cow-hornsunit
1: elementary differential calculus (35 hours)secondary school mathematics is
prerequisite.', 'basic mathematics 1 is co-re-quisite.this is a level 1 course
.unit 2: elementary integral calculus (35 hours)calculus 1 is prerequisite.thi
s is a level 1 course.unit 3: sequences and series (20 hours)priority a. calcu
lus 2 is prerequisite.this is a level 2 course.unit 4: calculus of functions o
f several variables (30 hours)priority b. calculus 3 is prerequisite.this is a
level 2 course.african virtual university   4120 hoursthe course materials for
this module consist of:study materials (print, cd, on-line) (pre-assessment ma
terials contained within the study materials) two formative assessment activit
ies per unit (always available but with speci- references and readings from op
en-source sources (cd, on-line) those which rely on copyright software  those
which rely on open source software  those which stand alone     graphical calc
```

Figura 37: División del texto en párrafos del OER “Calculus.pdf"

```
********** ORACIÓN **********
['calculusprepared by pr.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['calculusprepared', 'by', 'pr', '.']
global name 'tokens_representativos' is not defined
--------------------------------------------------------------------------------
********** ORACIÓN **********
['ralph w.p.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['ralph', 'w.p', '.']
--------------------------------------------------------------------------------
********** ORACIÓN **********
['masengeafrican virtual universityuniversite virtuelle africaineuniversidade virtu
al africanaafrican virtual university   1noticethis document is published under the
conditions of the creative commons http://en.wikipedia.org/wiki/creative_commons at
tribution  http://creativecommons.org/licenses/by/2.5/  license (abbreviated ficc-b
yfl), version 2.5. african virtual university   2i.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['masengeafrican', 'virtual', 'universityuniversite', 'virtuelle', 'africaineuniver
sidade', 'virtual', 'africanaafrican', 'virtual', 'university', '1noticethis', 'doc
ument', 'is', 'published', 'under', 'the', 'conditions', 'of', 'the', 'creative', '
commons', 'http', ':', '//en.wikipedia.org/wiki/creative_commons', 'attribution', '
http', ':', '//creativecommons.org/licenses/by/2.5/', 'license', '(', 'abbreviated'
, 'ficc-byfl', ')', ',', 'version', '2.5.', 'african', 'virtual', 'university', '2i
', '.']
global name 'tokens_representativos' is not defined
--------------------------------------------------------------------------------
********** ORACIÓN **********
['mathematics 3, calculus _____3ii.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['mathematics', '3', ',', 'calculus', '_____3ii', '
.']
global name 'tokens_representativos' is not defined
--------------------------------------------------------------------------------
********** ORACIÓN **********
['prerequisite course or knowledge _____3iii.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['prerequisite', 'course', 'or', 'knowledge', '_____3iii',
'.']
global name 'tokens_representativos' is not defined
```

Figura 38:División en oraciones y tokens del OER “Calculus.pdf"

```
********** ORACIÓN **********
['we then look at the structure of a general metric space a long the lines of
unit 1.  in addition we introduce the concept of compactness and its effects
on continuity of functions.unit 5  topologythe structures of topological spac
es are studied along side those of metric spaces.']
~~~~~~ Palabra ambigüa detectada --> look
~~~~~~ Palabra ambigüa detectada --> structure
~~~~~~ Palabra ambigüa detectada --> general
~~~~~~ Palabra ambigüa detectada --> metric
~~~~~~ Palabra ambigüa detectada --> space
~~~~~~ Palabra ambigüa detectada --> unit
~~~~~~ Palabra ambigüa detectada --> concept
~~~~~~ Palabra ambigüa detectada --> continuity
~~~~~~ Palabra ambigüa detectada --> topological
~~~~~~ Palabra ambigüa detectada --> spaces
~~~~~~ Palabra ambigüa detectada --> metric
~~~~~~ Palabra ambigüa detectada --> spaces
--------------------------------------------------------------------------------
********** ORACIÓN **********
['space the concept of distance is absent.in particular the study of the twin
concepts of convergence and continuity brings out this difference very well.']
~~~~~~ Palabra ambigüa detectada --> space
~~~~~~ Palabra ambigüa detectada --> concept
~~~~~~ Palabra ambigüa detectada --> distance
~~~~~~ Palabra ambigüa detectada --> particular
~~~~~~ Palabra ambigüa detectada --> study
~~~~~~ Palabra ambigüa detectada --> convergence
~~~~~~ Palabra ambigüa detectada --> continuity
~~~~~~ Palabra ambigüa detectada --> well
--------------------------------------------------------------------------------
********** ORACIÓN **********
['finally a look at different topologies like product or quotient topology end
owed on a set is essential in this unit.unit 6/7  measure theoryin this unit
we start with the study of both the lebesgue outer measure and the real line
before we look at the lebesgue measurable subsets of the real line.']
~~~~~~ Palabra ambigüa detectada --> look
~~~~~~ Palabra ambigüa detectada --> different
~~~~~~ Palabra ambigüa detectada --> topology
~~~~~~ Palabra ambigüa detectada --> endowed
~~~~~~ Palabra ambigüa detectada --> set
```

Figura 39: Identificación de palabras ambiguas del OER “Calculus.pdf”

```
********** TEXTO DIVIDIDO EN PARRAFOS DEL OER Multimedia%20Design%20and%20App
lications_readings.pdf ********
['            compulsory compulsory readingsreadings11
1 according to the author of the module, the compulsory readings do not infrin
ge known copyright.', 'compulsory readings   reading #1 complete reference:bel
zunce, a.,carrera,d.,hall, l.m., et al (2006).draw guide.openoffice.org:author
s.', 'http://oooauthors.org/en/authors/userguide2/published  abstract:  this b
ooks provides all the reading you need on using graphic tools.chapter one intr
oduces the toolbars.chapter two discusses how to draw basic shapes such a line
,a circle ellipse etc.chapters three and four deal with editing objects.chapte
r five is about managing 3d objects and bitmaps.chapter 6 is about combining m
ultiple objects.chapetr 7 provides useful  tips and tools for editing objects
.', 'this book is on cd for you rationale: the book provides material on how t
o create and edit graphics in an easy to follow format.', 'it contains good il
lustrations and guidance to the user to practice with the computer.', 'reading
#2 complete reference: adielsson, m., barnes,r., belzunce, a.,bonde, c.,et al
(2005).getting started.openoffice.org:authors.', 'http://oooauthors.org/en/aut
hors/userguide2/published  abstract: this book not only provides material rele
vant to this module but also helps the learner to revise material covered in t
he previous modules.', 'the topics and chapters directly linked to this module
include menus and toolbars (chapter 4), getting started with draw (chapter8),
creating presentation (chapter 9), working with templates (chapter 12), workin
g with gallery (chapter 14), and creating web pages (chapter 16).', 'this book
is on cd for you rationale: the book provides   comprehensive information in b
asic ict as an open source.', 'it has good illustrations for the reader and pr
ovides reading tips.', 'its coverage of web pages design makes it appropriate
text for this module.', 'reading #3 complete reference:worthington, l.,weber,j
.h.,kampa(2005).quickstart guide  for impress.openoffice.org:authors.', 'http:
//oooauthors.org/en/authors/userguide2/published  abstract :  rationale: this
is a concise presentation guide which shows the learner how to create a graphi
c presentation, how to format a presentation, choosing the layout, inserting n
ew slides, and running the slide show.this book is on cd for you.', 'rationale
: the presentation guide shows the learner in a practical way how to create an
d run aa graphic presentation.', 'its clarity makes it a must read for every s
tudent of basic ict.', 'reading(s) #1reading(s) #1      #     $    &  ()    ,.
', ',  ) (.1 3 1 1, 5    / ,   /  & 8   ,)78&  ,       9.', ')&9; 1<=  &??',
'@)&99)&)&9<)&   9<&9=/9=09!', '9!<)9!', '9b&9b9?', ': 9@<,#<.>  >   ) > >
>   *  c())(.', ':)+&(:<9=,#<!', '/:<b<b)&&<?', '(& <@( <@( :  =9  =97  7  7
7  =< &=<   =<&  =<&==&=!', '&  (&=b =b:( =?,(=?=@ $=@2 &   +&  >   !9!97 !9<)
)!=)!=!*8a!b*!?)!', '?<)  !?;>.', "/* *'   b!", ';;; *: dbb  bb*b?', '*b?>b?.b
@, / b@ *%  b%?', '!, a2ee?b2?b.cf?b>????2?@.?', '@,2a*/@9  )5 @<1g@=./@!', ')
```

Figura 40:División del texto en párrafos del OER “Multimedia%20Design"

```
********************* ORACION *********************
['            compulsory compulsory readingsreadings11
1 according to the author of the module, the compulsory readings do not infringe kn
own copyright.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['compulsory', 'compulsory', 'readingsreadings11', '1', 'according', 'to', 'the', '
author', 'of', 'the', 'module', ',', 'the', 'compulsory', 'readings', 'do', 'not',
'infringe', 'known', 'copyright', '.']
--------------------------------------------------------------------------------
********************* ORACIÓN *********************
['compulsory readings   reading #1 complete reference:belzunce, a.,carrera,d.,hall,
l.m., et al (2006).draw guide.openoffice.org:authors.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['compulsory', 'readings', 'reading', '#', '1', 'complete', 'reference', ':', 'belz
unce', ',', 'a.', ',', 'carrera', ',', 'd.', ',', 'hall', ',', 'l.m.', ',', 'et', '
al', '(', '2006', ')', '.draw', 'guide.openoffice.org', ':', 'authors', '.']
--------------------------------------------------------------------------------
********************* ORACIÓN *********************
['http://oooauthors.org/en/authors/userguide2/published  abstract:  this books prov
ides all the reading you need on using graphic tools.chapter one introduces the too
lbars.chapter two discusses how to draw basic shapes such a line ,a circle ellipse
etc.chapters three and four deal with editing objects.chapter five is about managin
g 3d objects and bitmaps.chapter 6 is about combining multiple objects.chapetr 7 pr
ovides useful  tips and tools for editing objects .']
~~~~~~~~ Tokens de la oración~~~~~~~~
['http', ':', '//oooauthors.org/en/authors/userguide2/published', 'abstract', ':',
'this', 'books', 'provides', 'all', 'the', 'reading', 'you', 'need', 'on', 'using',
'graphic', 'tools.chapter', 'one', 'introduces', 'the', 'toolbars.chapter', 'two',
'discusses', 'how', 'to', 'draw', 'basic', 'shapes', 'such', 'a', 'line', ',', 'a',
'circle', 'ellipse', 'etc.chapters', 'three', 'and', 'four', 'deal', 'with', 'editi
ng', 'objects.chapter', 'five', 'is', 'about', 'managing', '3d', 'objects', 'and',
'bitmaps.chapter', '6', 'is', 'about', 'combining', 'multiple', 'objects.chapetr',
'7', 'provides', 'useful', 'tips', 'and', 'tools', 'for', 'editing', 'objects', '.'
]
--------------------------------------------------------------------------------
********************* ORACIÓN *********************
['this book is on cd for you rationale: the book provides material on how to create
and edit graphics in an easy to follow format.']
~~~~~~~~ Tokens de la oración~~~~~~~~
['this', 'book', 'is', 'on', 'cd', 'for', 'you', 'rationale', ':', 'the', 'book', '
provides', 'material', 'on', 'how', 'to', 'create', 'and', 'edit', 'graphics', 'in'
```

Figura 41: División en oraciones y tokens del OER “Multimedia%20Design"

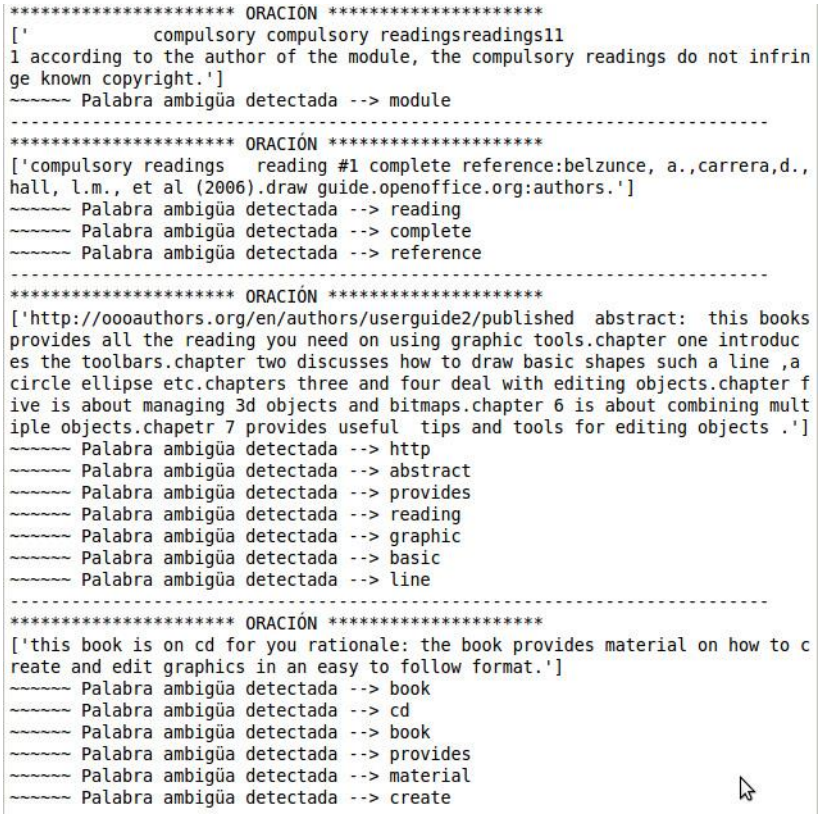


Figura 42: Identificación de palabras ambiguas del OER “Multimedia%20Design"

El script que contiene esta tarea se lo puede observar en el Anexo 9.

4.5.1.3. Desambiguar sentido de palabra

El propósito de esta tarea es asignar un correcto significado a una palabra ambigua utilizando su contexto.

4.5.1.3.1. *Ejecución*

En la sección 1.9 del estado del arte se establece el método basado en diccionarios para la desambiguación del sentido de las palabras, recordando este punto importante es meritorio también citar a (Torres, 2009) que menciona “el uso de un diccionario en la desambiguación del sentido de las palabras es de suma importancia, ya que esta consiste en asignar, a cada palabra en un texto dado, un sentido que se relaciona con una lista de sentidos en un diccionario”.

Para la ejecución de esta tarea los pasos que se siguieron son:

- Descartar los tokens que sean signos de puntuación y Stopwords como artículos o pronombres.
- Utilizar como filtro principal los tokens representativos del texto extraído, con la finalidad de ingresar al proceso de desambiguación solamente las palabras ambiguas y representativas.
- Tomar todos los significados de la palabra a evaluar para someterlos al algoritmo de Lesk.
- Recorrer cada palabra de cada significado, para comprobar si esa palabra se encuentra en el contexto dado.
- Tomar el significado que tenga mayor frecuencia de palabras dentro del contexto u oración a evaluar.

En el método propuesto se utiliza el corpus **WordNet** como diccionario para obtener los sentidos de una palabra; y además se utiliza los procesos establecidos anteriormente como tokenización, y tagging.

Tabla 18: Diccionario WordNet

CORPUS	COMPILER	CONTENTS
WordNet 3.0 (English)	Miller, Fellbaum	145k synonym sets

Fuente: (Natural Language Processing with Python, 2012)

Para la eliminar el problema de ambigüedad se utiliza el algoritmo de Lesk Simplificado véase sección 1.5.4.5, implementándolo sobre el lenguaje de programación Python en su versión 2.7, tomando en cuenta que cada palabra ambigua para pasar al proceso de desambiguación necesariamente debe ser parte de los tokens representativos caso contrario se la omite; al pasar por este filtro se ahorra tiempo en el proceso de ejecución

4.5.1.3.2. *Resultados*

Las siguientes gráficas muestran la oración que es el contexto, la palabra ambigua con los significados disponibles y finalmente la asignación del significado correcto para la palabra ambigua, se utilizan los OER's "Analysis%202.pdf", "Calculus.pdf" y "Multimedia%20Design".

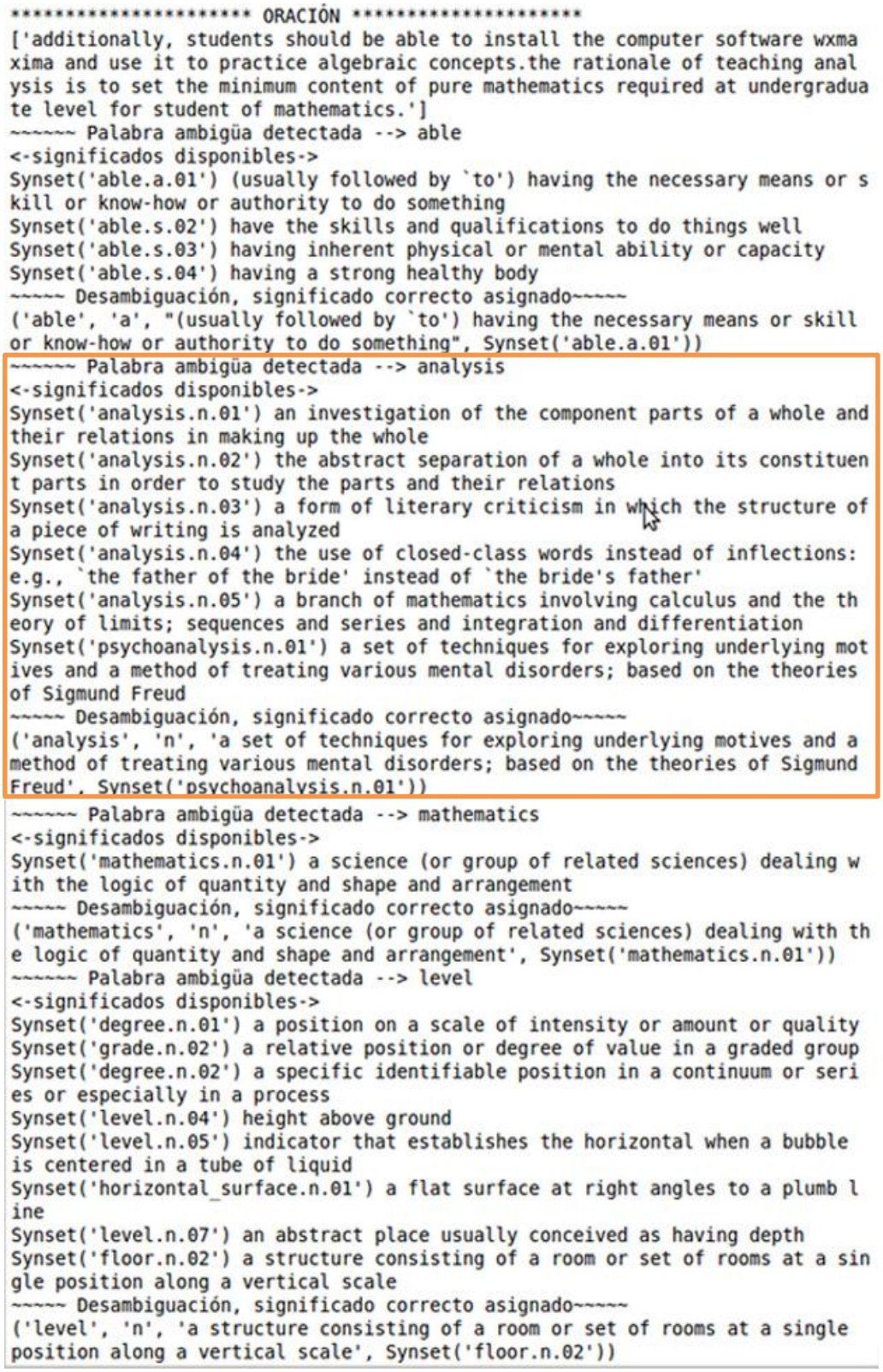


Figura 43: Desambiguación de palabras ambiguas del OER “Analysis%202.pdf”

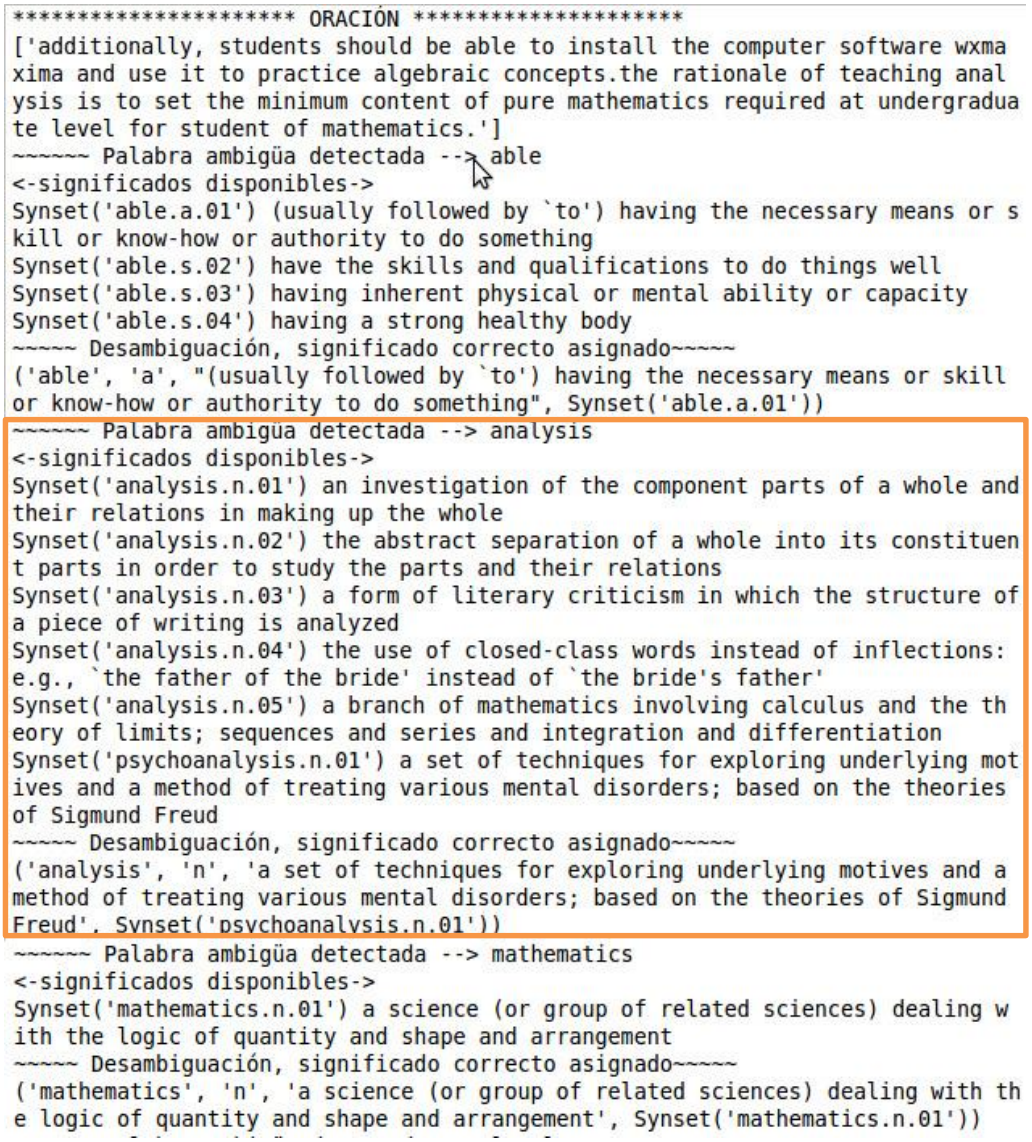


Figura 44: Desambiguación de palabras ambiguas del OER “Calculus"

```
********** ORACION **********
['http://oooauthors.org/en/authors/userguide2/published  abstract:  this books
provides all the reading you need on using graphic tools.chapter one introduc
es the toolbars.chapter two discusses how to draw basic shapes such a line ,a
circle ellipse etc.chapters three and four deal with editing objects.chapter f
ive is about managing 3d objects and bitmaps.chapter 6 is about combining mult
iple objects.chapetr 7 provides useful  tips and tools for editing objects .']
~~~~~~ Palabra ambigüa detectada --> http
<-significados disponibles->
Synset('hypertext_transfer_protocol.n.01') a protocol (utilizing TCP) to trans
fer hypertext requests and information between servers and browsers
~~~~~ Desambiguación, significado correcto asignado~~~~~
('http', 'n', 'a protocol (utilizing TCP) to transfer hypertext requests and
information between servers and browsers', Synset('hypertext_transfer_protocol
.n.01'))
~~~~~~ Palabra ambigüa detectada --> abstract
<-significados disponibles->
Synset('abstraction.n.01') a concept or idea not associated with any specific
instance
Synset('outline.n.02') a sketchy summary of the main points of an argument or
theory
~~~~~ Desambiguación, significado correcto asignado~~~~~
('abstract', 'n', 'a sketchy summary of the main points of an argument or the
ory', Synset('outline.n.02'))
~~~~~~ Palabra ambigüa detectada --> provides
<-significados disponibles->
Synset('supply.v.01') give something useful or necessary to
Synset('provide.v.02') give what is desired or needed, especially support, fo
od or sustenance
Synset('provide.v.03') determine (what is to happen in certain contingencies),
especially by including a proviso condition or stipulation
Synset('put_up.v.02') mount or put up
Synset('leave.v.06') make a possibility or provide opportunity for; permit to
be attainable or cause to remain
Synset('provide.v.06') supply means of subsistence; earn a living
Synset('provide.v.07') take measures in preparation for
~~~~~ Desambiguación, significado correcto asignado~~~~~
('provides', 'v', 'determine (what is to happen in certain contingencies), es
pecially by including a proviso condition or stipulation', Synset('provide.v.
03'))
~~~~~~ Palabra ambigüa detectada --> reading
<-significados disponibles->
Synset('reading.n.01') the cognitive process of understanding a written lingu
istic message
Synset('reading.n.02') a particular interpretation or performance
Synset('reading.n.03') a datum about some physical state that is presented to
a user by a meter or similar instrument
Synset('reading.n.04') written material intended to be read
Synset('interpretation.n.01') a mental representation of the meaning or signif
icance of something
Synset('reading.n.06') a city on the River Thames in Berkshire in southern Eng
land
Synset('recitation.n.02') a public instance of reciting or repeating (from me
mory) something prepared in advance
Synset('reading.n.08') the act of measuring with meters or similar instruments
~~~~~ Desambiguación, significado correcto asignado~~~~~
('reading', 'n', 'a datum about some physical state that is presented to a us
er by a meter or similar instrument', Synset('reading.n.03'))
~~~~~~ Palabra ambigüa detectada --> graphic
<-significados disponibles->
Synset('graphic.s.01') written or drawn or engraved
Synset('graphic.s.02') describing nudity or sexual activity in graphic detail
Synset('graphic.a.03') of or relating to the graphic arts
Synset('graphic.a.04') relating to or presented by a graph
Synset('graphic.s.05') evoking lifelike images within the mind
~~~~~ Desambiguación, significado correcto asignado~~~~~
('graphic', 'a', 'of or relating to the graphic arts', Synset('graphic.a.03'))
```

Figura 45: Desambiguación de palabras ambiguas del OER “Multimedia%20Design"

Se puede observar en cada imagen la asignación del significado correcto para cada palabra ambigua de cada OER, podemos apreciar que los recursos “Analysis%202.pdf” y “Calculus.pdf” tienen palabras en común con iguales siginificados, mientras tanto el tercer OER “Multimedia%20Design” no tiene ninguna palabra en común con los dos recursos anteriores. En la tabla siguiente se puede observar con mas claridad un extracto de las palabras representativas de cada OER, y los significados asignados una vez concluida la tarea de desambiguación.

Se observará: un extracto de la lista de palabras representativas ambiguas, y dos significados asignados a la palabra según el contexto encontrado; a los significados comunes entre los OER’s se los resaltó con otro color para una mejor apreciación

Tabla 19: Palabras representativas y significados comunes entre OER'S “Analysis%202.pdf” y “Calculus.pdf”

OER Analysis%202.pdf			OER Calculus.pdf			OER MultimediaDesign.pdf		
Palabra representativa ambigua	**Asignación Significado 1**	**Asignación Significado 2**	**Palabra representativa ambigua**	**Asignación Significado 1**	**Asignación Significado 2**	**Palabra representativa ambigua**	**Asignación Significado 1**	**Asignación Significado 2**
Palabras y su significados en común			**Palabras y su significados en común**			**Palabras y su significados en común**		
continuity *	continuity', 'n', 'the property of a continuous and connected period of time', Synset('continuity.n.03')	continuity', 'n', 'a detailed script used in making a film in order to avoid discontinuities from shot to shot', Synset('continuity.n.02')	**continuity ***	continuity', 'n', 'the property of a continuous and connected period of time', Synset('continuity.n.03')	continuity', 'n', 'a detailed script used in making a film in order to avoid discontinuities from shot to shot', Synset('continuity.n.02'			
convergence *	convergence', 'n', 'the approach of an infinite series to a finite limit', Synset('convergence.n.02')		**convergence ***	convergence', 'n', 'the approach of an infinite series to a finite limit', Synset('convergence.n.02')				
integral *	integral', 'a', 'existing as an essential constituent or characteristic', Synset('built-in.s.01')		**integral ***	'integral', 'a', 'of or denoted by an integer', Synset('integral.a.03')	'integral', 'a', 'existing as an essential constituent or characteristic', Synset('built-in.s.01')			
limit *	limit', 'n', 'the mathematical value toward which a function goes as the independent variable approaches infinity', Synset('limit.n.05')	limit', 'n', 'the greatest amount of something that is possible or allowed', Synset('limit.n.06')	**limit ***	limit', 'n', 'the mathematical value toward which a function goes as the independent variable approaches infinity', Synset('limit.n.05')	limit', 'n', 'the greatest amount of something that is possible or allowed', Synset('limit.n.06')) ,			

mathematical *	mathematical', 'a', 'characterized by the exactness or precision of mathematics', Synset('mathematical.s.05')	'mathematical', 'a', 'of or pertaining to or of the nature of mathematics', Synset('mathematical.a.01')	**mathematical ***	mathematical', 'a', 'of or pertaining to or of the nature of mathematics', Synset('mathematical.a.01')	mathematical', 'a', 'relating to or having ability to think in or work with numbers', Synset('numerical.a.03')			
mathematics *	mathematics', 'n', 'a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement', Synset('mathematics.n.01')		**mathematics ***	mathematics', 'n', 'a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement', Synset('mathematics.n.01')				
palabras adicionales y su significado			**palabras adicionales y su significado**			**palabras adicionales y su significado**		
analysis	analysis', 'n', 'a form of literary criticism in which the structure of a piece of writing is analyzed', Synset('analysis.n.03')	analysis', 'n', 'the abstract separation of a whole into its constituent parts in order to study the parts and their relations', Synset('analysis.n.02')	**calculus**	calculus', 'n', 'a hard lump produced by the concretion of mineral salts; found in hollow organs or ducts of the body', Synset('calculus.n.01')	calculus', 'n', 'the branch of mathematics that is concerned with limits and with the differentiation and integration of functions', Synset('calculus.n.03')	**basic**	basic', 'a', 'reduced to the simplest and most significant form possible without loss of generality', Synset('basic.s.02')	basic', 'a', 'of or denoting or of the nature of or containing a base', Synset('basic.s.04')
distance	distance', 'n', 'the property created by the space between two objects or points', Synset('distance.n.01')	distance', 'n', 'size of the gap between two places', Synset('distance.n.03')	**curve**	curve', 'n', 'a pitch of a baseball that is thrown with spin so that its path curves as it approaches the batter', Synset('curve.n.03')		**create**	create', 'v', 'invest with a new title, office, or rank', Synset('create.v.04')	create', 'v', 'pursue a creative activity; be engaged in a creative activity', Synset('create.v.03')

geometric	geometric', 'a', 'characterized by simple geometric forms in design and decoration', Synset('geometric.s.01')	geometric', 'a', 'of or relating to or determined by geometry', Synset('geometric.a.02')	**discontinuity**	discontinuity', 'n', 'lack of connection or continuity', Synset('discontinuity.n.01')		**http**	http', 'n', 'a protocol (utilizing TCP) to transfer hypertext requests and information between servers and browsers', Synset('hypertext_transfer_protocol.n.01')	
theorem	theorem', 'n', 'an idea accepted as a demonstrable truth', Synset('theorem.n.02')		**trapezium**	trapezium', 'n', 'a quadrilateral with no parallel sides', Synset('trapezium.n.01')	trapezium', 'n', 'the wrist bone on the thumb side of the hand that articulates with the 1st and 2nd metacarpals', Synset('trapezium.n.03')	***line***	*line', 'n', 'mechanical system in a factory whereby an article is conveyed through sites at which successive operations are performed on it', Synset('production_line.n.01')*	*line', 'n', 'a slight depression in the smoothness of a surface', Synset('wrinkle.n.01')*
function	function', 'n', 'the actions and activities assigned to or required or expected of a person or group', Synset('function.n.03')		**functions**	functions', 'n', '(mathematics) a mathematical relation such that each element of a given set (the domain of the function) is associated with an element of another set (the range of the function)', Synset('function.n.01')				

Para una mejor compresión de los datos mostrados en la tabla 18 se presenta en los siguientes literales una explicación sobre los puntos principales que muestran los datos mostrados.

- Las palabras representativas que poseen un asterisco (*) se repiten en los dos recursos educativos abiertos “Analysis%202.pdf” y “Calculus.pdf”, de los significados asignados para cada palabra algunos se repiten en ambos OER’s, demostrando que los dos OER’s tienen conceptos en común (palabra representativa e igual significado).
- En cuanto a las palabras representativas adicionales se puede observar que no existe frecuencia de palabras entre los OER’s, por lo tanto no existen conceptos en común.
- Si observamos la ultima palabra representativa en la tabla “functions” esta palabra se repite en ambos OER’s sin embargo sus significados no son iguales, por lo tanto no es un concepto en común para los OER’s.
- Por otro lado tenemos al OER “MultimediaDesign.pdf” el cual no tiene palabras representativas ni significado que se repitan entre los otros dos OER’s, por lo tanto este recurso educativo abierto no es común con ninguno de los dos OER’s.

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 9.

4.5.2. Crear Relaciones

El propósito de crear las relaciones entre OER’s implica identificar las palabras comunes entre OER’s.

4.5.2.1. Ejecución

Al hablar de palabras comunes es necesario mencionar que deben referirse al mismo tema; por ejemplo:

- Si tenemos la palabra “lenguaje” en el OER1 y esta palabra se refiere al tema “Lenguaje de Programación”, el OER32 que también tiene la palabra “lenguaje” esta debe referirse al mismo tema “Lenguaje de Programación”.
- Dado el caso de que el OER32 con la palabra “lenguaje” se refiera al tema “Idioma”, este OER se descarta para la relación.

Al abordar el tema de relación de los recursos educativos abiertos nos encontramos con premisas válidas y necesarias que nos ayudaran a enfocarnos correctamente.

- El OER se puede relacionar con uno o varios OER’s.
- Un OER puede relacionarse con otro siempre y cuando aborden el mismo tema, por ejemplo: Estadística, Biología, Programación, etc.
- Un OER puede relacionarse con otro tomando en cuenta las entidades en común.

Este proceso tiene un nivel de complejidad considerable, debido a los factores mencionados anteriormente, para una mejor comprensión de este tema se puede observar la tabla 18 y la explicación mencionada después de la tabla.

Para crear las relaciones entre OER’s se necesita como base las palabras comunes o entidades; al tener varios OER’s para evaluar es posible que exista más de una relación, por lo tanto es necesario definir la siguiente regla:

- Una relación entre OER’s prevalece ante otra siempre y cuando tenga mayor número de palabras comunes o entidades, o al menos la cuarta parte del mayor número de palabras comunes o entidades.

La creación de una relación entre OER’s depende directamente de que se cumpla la regla mencionada anteriormente, para este proceso se emplea el principio del “Método de Ordenación Burbuja”; se toma un elemento del array y se lo compara con los elementos siguientes.

Cada OER tiene un array que contiene las palabras desambiguas o entidades, cada palabra/entidad del array se compara con cada palabra/entidad del array perteneciente al siguiente OER a evaluar.

Al encontrar una palabra/entidad común durante la evaluación se la almacena en otro array, al culminar el proceso se realiza un conteo con el número de palabras comunes para verificar que cumpla la regla mencionada anteriormente, caso contrario se lo descarta.

El desarrollo del proceso que se menciona se debe ejecutar por separado para las palabras desambiguas y para las entidades. Para esta tarea es necesario:

- El manejo de estructuras de datos como listas, y tuplas.
- Consultas Sparql para obtener las entidades, se utilizan los predicados: http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense (palabras desambiguas), http://purl.org/spar/doco/ListOfAgents (entidades), y http://xmlns.com/foaf/0.1/Document (URI del OER), mediante consultas Sparql se los extrae de la base de datos Virtuoso.

4.5.2.2. Resultados

La siguiente imagen muestra las relaciones establecidas para el OER Analysis%202.pdf, se observará a cuatro OER's que se relacionan con el mismo tema.

relaciones_Analysis202_pdf
http://oer.avu.org/bitstream/handle/123456789/153/GUIDANCE%20AND%20COUNSELING.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/53/Calculus.pdf?sequence=3
http://oer.avu.org/bitstream/handle/123456789/17/Linear%20Programming.pdf?sequence=4
http://oer.avu.org/bitstream/handle/123456789/14/Analysis%201.pdf?sequence=8

Figura 46: Relaciones con otros OER's para el recurso Analysis%202.pdf

Como se puede observar los OER's "Analysis202.pdf" y "Calculus.pdf" se relacionan; el script que realiza esta tarea se lo puede observar en los Anexos 5, 10 y 11.

4.6. Generar datos RDF

El propósito de este proceso consiste en "tomar las fuentes de datos identificadas previamente y transformarla a formato RDF utilizando los vocabularios establecidos, de esta forma se cumple con los principios de Linked Data." (Villazón, Vilches, Corcho, & Gómez, 2011)

4.6.1. Ejecución

Durante la ejecución de cada una de las tareas pertenecientes a cada proceso se obtuvo data relevante que debe ser identificada utilizando vocabulario RDF que fue definido en la sección 4.2.

Con el vocabulario definido se debe realizar los siguientes pasos:

- Establecer tripletas *sujeto, predicado y objeto* con la data obtenida.
- Definir el tipo de valor que tomará el "objeto" puede ser un recurso (representado por un URI) o un literal como un valor numérico o cadena de caracteres.

La siguiente tabla presenta las tripletas con los valores que contendrá cada uno.

Tabla 20: Definición de tripletas con sus respectivos valores

SUJETO	PREDICADO	OBJETO
URL del OCW	http://xmlns.com/foaf/0.1/Document	URI del OER (Recurso)
URI del OER	http://www.aktors.org/ontology/portal#has-page-numbers	Número de páginas (Literal)
URI del OER	http://purl.org/spar/doco/Title	Nombre del OER (Literal)
URI del OER	http://purl.org/dc/elements/1.1/language	Idioma (Literal)
URI del OER	http://www.lexinfo.net/ontology/2.0/lexinfo#standardText	Texto extraído (Literal)
URI del OER	http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word	Tokens obtenidos del texto (Literal)
URI del OER	http://www.w3.org/ns/dcat#keyword	Tags asignado a cada token (Literal)
URI del OER	http://purl.org/spar/doco/Glossary	Tokens representativos (Literal)

URI del OER	http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense	Palabras ambiguas con sus significado (Literal)
URI del OER	http://purl.org/spar/doco/ListOfAgents	Entidades extraídas (Literal)
URI del OER	http://purl.org/dc/elements/1.1/relation	URI de OER relacionado (Recurso)
URI del OER	http://www.w3.org/2002/07/owl#NamedIndividual	URI de la entidad (Recurso)
URI del OER	http://www.w3.org/2006/03/wn/wn20/schema/word	URI de la palabra común (Recurso)
URI de la palabra común	http://www.w3.org/2006/03/wn/wn20/instances/	Significado de WordNet (Literal)
URI de la palabra o entidad común	http://www.w3.org/2002/07/owl#sameas	URI de DBpedia

Las herramientas utilizadas para la correcta ejecución de esta tarea son:

- **rdflib.term:** para definir el tipo de valor del “objeto”
- **rdflib.namespace:** para utilizar el vocabulario con DublinCore.

4.6.2. Resultados

La imagen siguiente muestra el modelo conceptual de los datos en formato RDF, representa los valores que se utilizarán durante al obtención de los mismos.

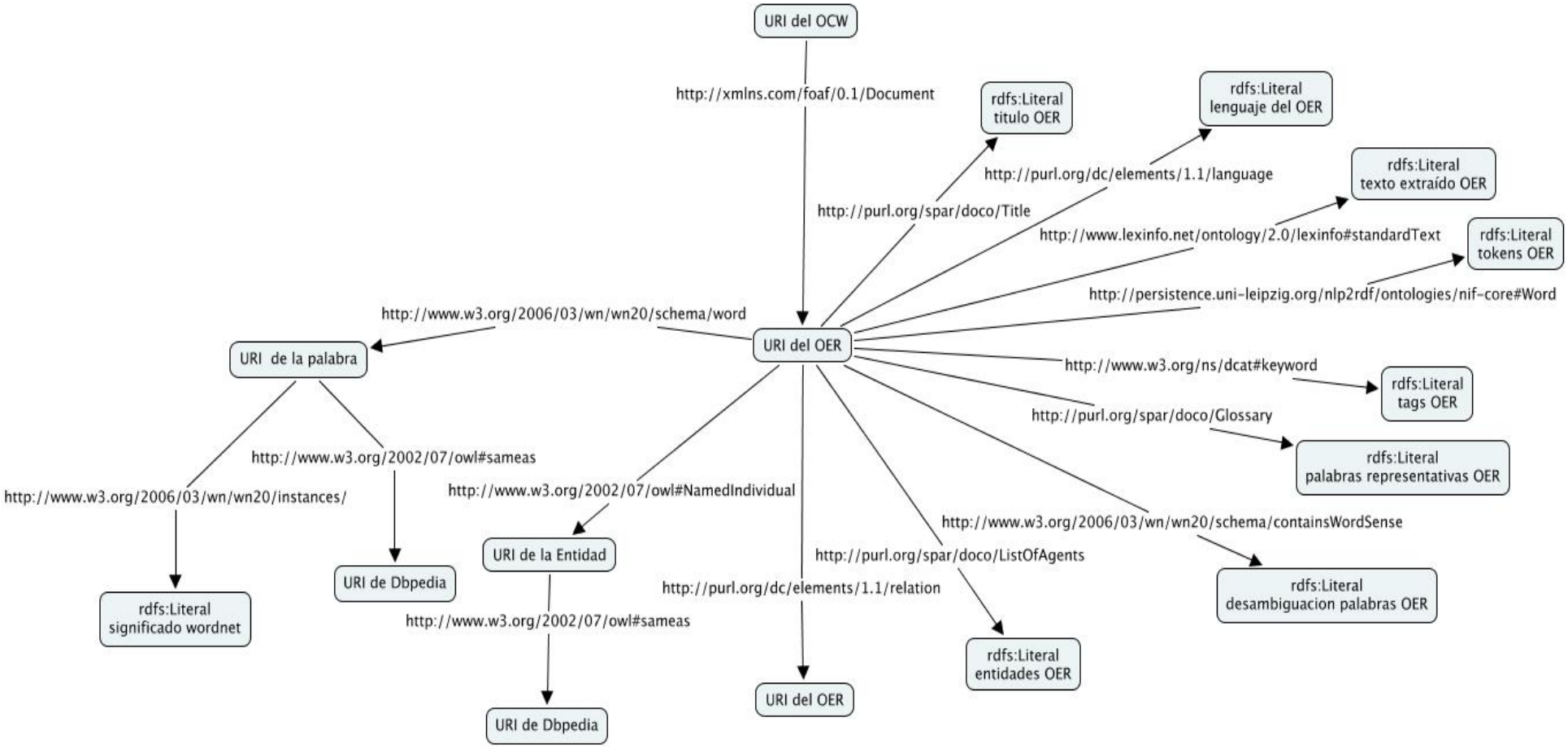


Figura 47: Modelo Conceptual de metadata de OER's

El script que contiene esta tarea se lo puede observar en el Anexo 12.

4.7. Publicación de datos RDF

El propósito principal es almacenar y publicar la data en triplestore es decir en una base de datos para el almacenamiento en tripletas RDF.

4.7.1. Ejecución

Cabe mencionar que la aplicación fue desarrollada sobre el sistema operativo Linux en su distribución Ubuntu 12.10, sin embargo se la implementó sobre la distribución Centos 5.6 distro que posee el servidor “apolo” funciona correctamente en ambas distribuciones. Para el almacenamiento de los datos RDF obtenidos en la sección anterior se debe realizar.

- **Establecer el triplestore a utilizar:** Virtuoso es el triplestore elegido para nuestro fin, las características (ver anexo 20) que posee lo convierte en uno de los repositorios más utilizados.
- **Instalar las librerías**: este paso es uno de los principales y el que más atención merece, es muy importante que cada librería que interviene para la ejecución correcta de este proceso sea instalada adecuadamente, recordando que las librerías que se utilizan son del lenguaje de programación Python en su versión 2.7. No podemos dejar de lado las configuraciones que se deben realizar a nivel del sistema.
- **Crear conexión:** para este paso se necesita credenciales de acceso para Virtuoso (usuario y contraseña), con estos datos se puede establecer la conexión entre Python 2.7 y Virtuoso.

- **Crear el Graph IRI:** implica establecer un nombre descriptivo para el IRI[6] que es el identificador del grafo que contiene la data obtenida de los OER's. El nombre que se asignó para el IRI es: ***http://dataoers.org/***
- **Insertar las tripletas:** Las tripletas definidas en la sección anterior se deben ir almacenando conforme se vaya obteniendo la data en cada tarea.

Las herramientas utilizadas durante la ejecución de esta tarea presentaron un grado considerable de atención en su instalación, debido a las dependencias de paquetes que necesitan y la compatibilidad entre versiones.

- **pyodbc 2.11, con el parche 2.12:** es un conector de datos entre Python y Virtuoso, la aplicación funciona específicamente con esta versión.
- **Virtuoso:** es el repositorio elegido para el almacenamiento de la data obtenida en formato RDF.
- **virtuosos-python-master:** librería para conexión entre Python 2.7 con el triplestore Virtuoso.
- Configuraciones a nivel del sistema para el correcto funcionamiento de Virtuoso, y pyodbc.

4.7.2. Resultados

Es importante mostrar los resultados con algunas preguntas sobre la data obtenida de los OER's, las mismas que deben ser solucionadas mediante consultas SPARQL. Los siguientes ejemplos muestran las preguntas, la consulta SPARQL, y el resultado que retorna. Para realizar pruebas de las consultas mencionadas a continuación se puede acceder al endpoint de virtuoso con la ruta: ***http://apolo.utpl.edu.ec:8890/sparql***

Pregunta: ¿Cuántas tripletas disponibles existen?

Consulta SPARQL: select distinct count(*) where {?s ?p ?o}

[6] Internationalized Resource Identifiers (IRI) es un nuevo elemento de protocolo, un complemento para los URIs, para mas información revisar http://www.w3.org/International/O-URL-and-ident.html

Tabla 21: Cantidad de tripletas existentes en Virtuoso

CALLRET-0
11392

Pregunta: ¿Cuáles son los OER's de los que se ha extraído y procesado su data?

Consulta SPARQL: select distinct * where {?sujeto <http://xmlns.com/foaf/0.1/Document > ?objeto}

apolo.utpl.edu.ec:8890/sparql?default-graph-uri=http%3A%2F%2Fdataoers.org%2F&query=select+distinct+*+where+%7B%3Fsujeto+<htt

sujeto	objeto
http://ocw.uc3m.es/ingenieria-informatica/accesibilidad-universal/bibliografia	http://www.itu.int/ITU-D/sis/PwDs/Documents/ITU-G3ict%20Making_TV_Accessible_Report_November_2011.pdf
http://ocw.uc3m.es/periodismo/teoria-de-la-comunicacion-mediatica/bibliografia	http://psyc604.stasson.org/Milgram2.pdf
http://ocw.uc3m.es/tecnologia-electronica/electronic-instrumentation-and-laboratory-of-electronic-instrumentation/laboratory-tests	http://www.analog.com/static/imported-files/data_sheets/AD620.pdf
http://ocw.uc3m.es/ingenieria-quimica/environmental-engineering/basic-bibliography	http://www.pedz.uni-mannheim.de/daten/edz-bn/gdu/02/waterguide_en.pdf
http://ocw.tufts.edu/Course/4/Lecturenotes	http://ocw.tufts.edu/data/4/531943.pdf
http://oer.avu.org/handle/123456789/251	http://oer.avu.org/bitstream/handle/123456789/251/Mathematical-Physics1vf-Readings.pdf?sequence=1
http://oer.avu.org/handle/123456789/154	http://oer.avu.org/bitstream/handle/123456789/154/managing-school.pdf?sequence=1
http://oer.avu.org/handle/123456789/152	http://oer.avu.org/bitstream/handle/123456789/152/CONTEMPORARY-ISSUES-EDUCATION.pdf?sequence=1
http://oer.avu.org/handle/123456789/153	http://oer.avu.org/bitstream/handle/123456789/153/GUIDANCE%20AND%20COUNSELING.pdf?sequence=1
http://oer.avu.org/handle/123456789/157	http://oer.avu.org/bitstream/handle/123456789/157/SPECIAL-NEEDS.pdf?sequence=1
http://oer.avu.org/handle/123456789/85	http://oer.avu.org/bitstream/handle/123456789/85/Multimedia%20Design%20and%20Applications.pdf?sequence=3
http://oer.avu.org/handle/123456789/38	http://oer.avu.org/bitstream/handle/123456789/38/Separation%2c%20Electroanalytical%20and%20Spectrometric%20Techniques.pdf?sequence=4
http://oer.avu.org/handle/123456789/84	http://oer.avu.org/bitstream/handle/123456789/84/Graphics%20and%20Information%20Management%20Systems.pdf?sequence=3
http://oer.avu.org/handle/123456789/83	http://oer.avu.org/bitstream/handle/123456789/83/Text-Based%20Productivity%20Tools.pdf?sequence=3
http://oer.avu.org/handle/123456789/82	http://oer.avu.org/bitstream/handle/123456789/82/Introduction%20to%20ICT.pdf?sequence=3
http://oer.avu.org/handle/123456789/65	http://oer.avu.org/bitstream/handle/123456789/65/Integrating%20ICT%20in%20Mathematics%20Education.pdf?sequence=2
http://oer.avu.org/handle/123456789/75	http://oer.avu.org/bitstream/handle/123456789/75/Classroom%20Management%20and%20Supervision.pdf?sequence=1
http://oer.avu.org/handle/123456789/76	http://oer.avu.org/bitstream/handle/123456789/76/Educational%20Communication.pdf?sequence=1
http://oer.avu.org/handle/123456789/55	http://oer.avu.org/bitstream/handle/123456789/55/Mechanics.pdf?sequence=4
http://oer.avu.org/handle/123456789/78	http://oer.avu.org/bitstream/handle/123456789/78/Educational%20Evaluation%20and%20Testing.pdf?sequence=1
http://oer.avu.org/handle/123456789/72	http://oer.avu.org/bitstream/handle/123456789/72/Developmental%20Psychology.pdf?sequence=1
http://oer.avu.org/handle/123456789/69	http://oer.avu.org/bitstream/handle/123456789/69/Comparative%20Education.pdf?sequence=1

er.avu.org/bitstream/handle/123456789/251/Mathematical-Physics1vf-Readings.pdf?sequence=1

Figura 48: Extracto de los OER's de los cuales se ha extraído y procesado su data.

Pregunta: ¿Cuáles son las entidades extraídas del OER Calculus.pdf?

Consulta SPARQL: select distinct * where {<http://oer.avu.org/bitstream/handle/123456789/53/Calculus.pdf?sequence=3> <http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense> ?objeto} limit 100

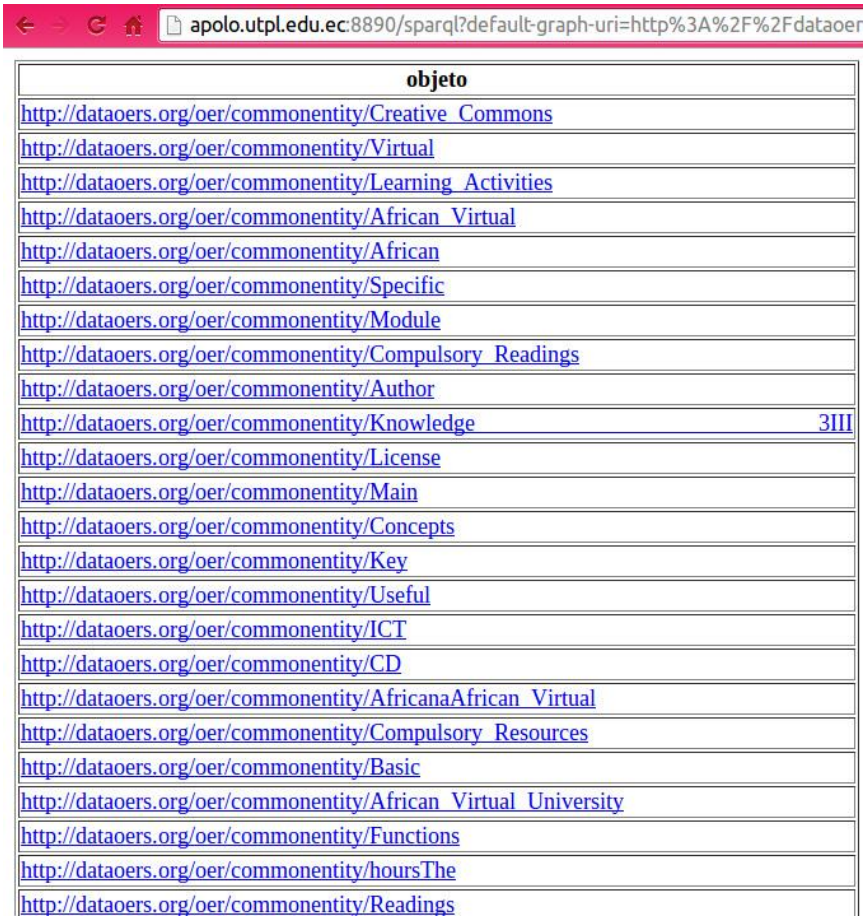
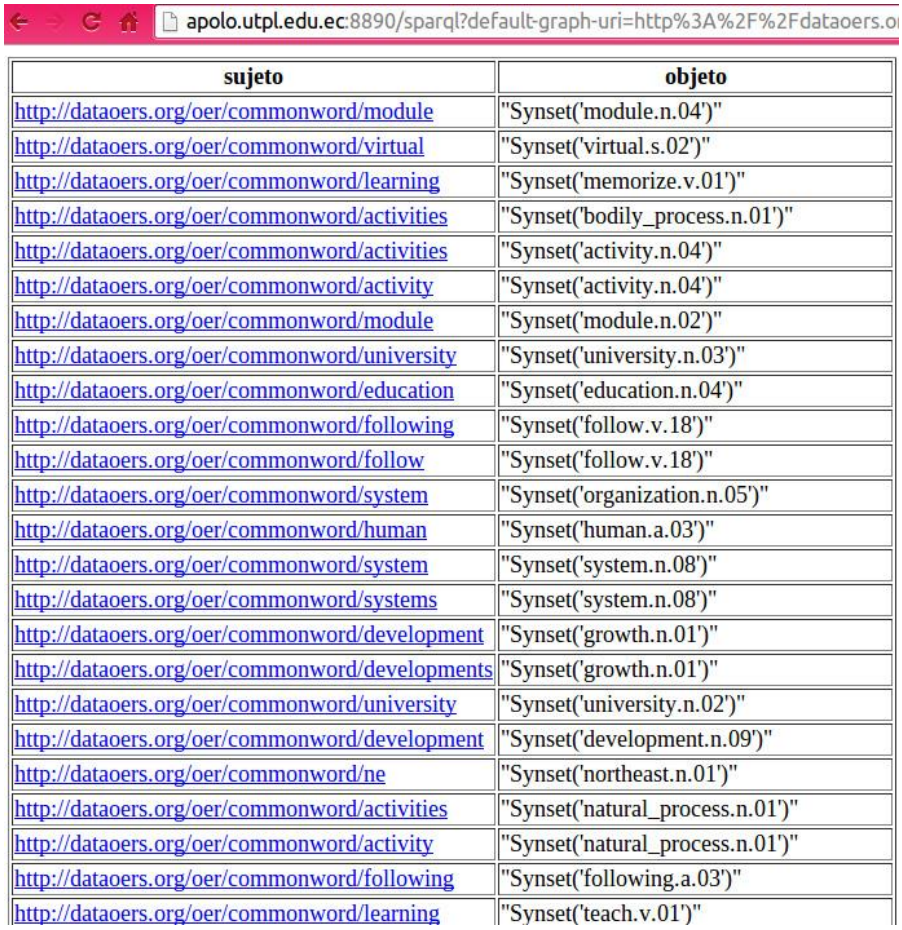


Figura 49: Extracto de las entidades del OER Calculus.pdf

Pregunta: ¿Cuáles son los significados establecidos para las palabras ambiguas en común entre OER's?

Consulta SPARQL: select distinct * where {?sujeto < http://www.w3.org/2006/03/wn/wn20/instances> ?objeto}



sujeto	objeto
http://dataoers.org/oer/commonword/module	"Synset('module.n.04')"
http://dataoers.org/oer/commonword/virtual	"Synset('virtual.s.02')"
http://dataoers.org/oer/commonword/learning	"Synset('memorize.v.01')"
http://dataoers.org/oer/commonword/activities	"Synset('bodily_process.n.01')"
http://dataoers.org/oer/commonword/activities	"Synset('activity.n.04')"
http://dataoers.org/oer/commonword/activity	"Synset('activity.n.04')"
http://dataoers.org/oer/commonword/module	"Synset('module.n.02')"
http://dataoers.org/oer/commonword/university	"Synset('university.n.03')"
http://dataoers.org/oer/commonword/education	"Synset('education.n.04')"
http://dataoers.org/oer/commonword/following	"Synset('follow.v.18')"
http://dataoers.org/oer/commonword/follow	"Synset('follow.v.18')"
http://dataoers.org/oer/commonword/system	"Synset('organization.n.05')"
http://dataoers.org/oer/commonword/human	"Synset('human.a.03')"
http://dataoers.org/oer/commonword/system	"Synset('system.n.08')"
http://dataoers.org/oer/commonword/systems	"Synset('system.n.08')"
http://dataoers.org/oer/commonword/development	"Synset('growth.n.01')"
http://dataoers.org/oer/commonword/developments	"Synset('growth.n.01')"
http://dataoers.org/oer/commonword/university	"Synset('university.n.02')"
http://dataoers.org/oer/commonword/development	"Synset('development.n.09')"
http://dataoers.org/oer/commonword/ne	"Synset('northeast.n.01')"
http://dataoers.org/oer/commonword/activities	"Synset('natural_process.n.01')"
http://dataoers.org/oer/commonword/activity	"Synset('natural_process.n.01')"
http://dataoers.org/oer/commonword/following	"Synset('following.a.03')"
http://dataoers.org/oer/commonword/learning	"Synset('teach.v.01')"

Figura 50: Extracto de los significados establecidos para las palabras ambiguas de los OER's

Pregunta: ¿Cuáles son las palabras representativas de los OER's?

Consulta SPARQL: select distinct * where {?sujeto <http://purl.org/spar/doco/Glossary> ?objeto}

Tabla 22: Extracto de las palabras representativas de los OER's

Sujeto	Objeto
http://oer.avu.org/bitstream/handle/123456789/154/managing-school.pdf?sequence=1	"(lp0 S'activities' p1 aS'actors' p2 aS'african' p3 aS'available' p4 aS'chart' p5 aS'data' p6 aS'de' p7 aS'development' p8 aS'e' p9 aS'education' p10 aS'educational' p11 aS'explanation' p12 aS'following' p13 aS'functions' p14 aS'human' p15 aS'identify' p16 aS'knowledge' p17 aS'learning' p18 aS'list' p19

	aS'manage' p20 aS'management' p21 aS'managing' p22 aS'material' p23 aS'module' p24 aS'nancial' p25 aS'ne' p26 aS'organization' p27 aS'principal' p28 aS'required' p29 aS'resources' p30 aS'role' p31 aS'roles' p32 aS'school' p33 aS'schoolos' p34 aS'sources' p35 aS'structure' p36 aS'system' p37 aS'teaching' p38 aS'university' p39 aS'various' p40 aS'virtual' p41 aS'within' p42 a."
http://oer.avu.org/bitstream/handle/123456789/274/Analysis%202.pdf?sequence=1	"(lp0 S'1' p1 aS'3' p2 aS'4' p3 aS'5' p4 aS'6' p5 aS'able' p6 aS'abstract' p7 aS'african' p8 aS'also' p9 aS'analysis' p10 aS'b' p11 aS'bn' p12 aS'c' p13 aS'called' p14 aS'closed' p15 aS'concepts' p16 aS'continuity' p17 aS'continuous' p18 aS'convergence' p19 aS'd' p20 aS'd.' p21 aS'dx' p22 aS'dy' p23 aS'essential' p24 aS'every' p25 aS'following' p26 aS'function' p27 aS'geometric' p28 aS'given' p29 aS'hours' p30 aS'integral' p31 aS'interior' p32 aS'learning' p33 aS'lebesgue' p34 aS'let' p35 aS'level' p36 aS'like' p37 aS'limit' p38 aS'line' p39 aS'look' p40 aS'mathematical' p41 aS'measurable' p42 aS'measure' p43 aS'metric' p44 aS'module' p45 aS'neighbourhood' p46 aS'note' p47 aS'open' p48 aS'p' p49 aS'point' p50 aS'points' p51 aS'priority' p52 aS'real' p53 aS'riemann' p54 aS'said' p55 aS'set' p56 aS'sets' p57 aS'space' p58 aS'spaces' p59 aS'statements' p60 aS'structure' p61 aS'study' p62 aS'subset' p63 aS'subsets' p64 aS'theory' p65 aS'topology' p66 aS'true' p67 aS'unit' p68 aS'university' p69 aS'virtual' p70 aS'well' p71 aS'x' p72 aS'xx' p73 aS'yx' p74 a."

Pregunta: ¿Cuáles son todos los valores que tiene el OER “Analysis%202.pdf”?

Consulta SPARQL: select distinct * where {<http://oer.avu.org/handle/123456789/264/Analysis%202.pdf?sequence=1> ?p ?o}

Tabla 23: Propiedades correspondientes al OER Analysis%202.pdf

P	O
http://purl.org/dc/elements/1.1/relation	http://oer.avu.org/bitstream/handle/123456789/153/GUIDANCE%20AND%20COUNSELING.pdf?sequence=1
http://purl.org/dc/elements/1.1/relation	http://oer.avu.org/bitstream/handle/123456789/53/Calculus.pdf?sequence=3
http://purl.org/dc/elements/1.1/relation	http://oer.avu.org/bitstream/handle/123456789/14/Analysis%201.pdf?sequence=8
http://purl.org/dc/elements/1.1/relation	http://oer.avu.org/bitstream/handle/123456789/17/Linear%20Programming.pdf?sequence=4
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/virtual
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/module
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/learning
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/set
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/structure
http://www.w3.org/2006/03/wn/wn20/schema/word	http://dataoers.org/oer/commonword/closed
http://purl.org/spar/doco/Title	Analysis%202.pdf
http://www.aktors.org/ontology/portal#has-page-numbers	71
http://www.lexinfo.net/ontology/2.0/lexinfo#standardText	prepared by jairus m. khalagaiafrican virtual universityuniversite virtuelle africaineuniversidade virtual africanaanalysis 2african virtual university 1noticethis document is published under the conditions of the creative commons http://en.wikipedia.org/wiki/creative_commons attribution http://creativecommons.org/licenses/by/2.5/ license (abbreviated ficc-byfl), version 2.5. african virtual university 2i. analysis 2 _____3ii.
http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word	(lp0 S'prepared' p1 aS'by' p2 aS'jairus' p3 aSp1379 aS'from' p1380 ag233 aS'metric' p1381 aS'such' p1437 aS'that' p1438 aS'for' p1439 aS'any' p1440 aS'pair' p1441 aS'of' p1442 aS'points' p1443 ag1029 ag37 aS'yx'

	p1444 aS'we' p1445 aS'have' p1446 ag27 aS'dxx' p1447 ag37 aS'ydyfxfy' p1448 a.
http://www.w3.org/ns/dcat#keyword	(lp0 (S'prepared' p1 S'VBN' p2 tp3 a(S'paying' p2265 g156 tp2266 a(S'special' p2267 g16 tp2268 a(S'attention' p2269 g8 tp2270 a(S'to' p2271 g352 tp2272 a(S'the' p2273 g47 tp2274 a(S'struc-ture' p2275 g16 tp2276 a(S'itself.' p2277 g11 tp2278 a(S'in' p2279 g5 tp2280 a(S'modern' p2281 g16 tp2282 a(S'analysis' p2283 g8 tp2284 a(S'this' 2285 g47 tp2286 a(S'is' p2287 g
http://purl.org/spar/doco/Glossary	(lp0 S'1' p1 aS'3' p2 aS'4' p3 aS'5' p4 aS'6' p5 aS'point' p50 aS'points' p51 aS'priority' p52 aS'real' p53 aS'riemann' p54 aS'said' p55 aS'set' p56 aS'sets' p57 aS'space' p58 aS'spaces' p59 aS'statements' p60 aS'structure' p61 aS'study' p62 aS'subset' p63 aS'subsets' p64 aS'theory' p65 aS'topology' p66 aS'true' p67 aS'unit' p68 aS'university' p69 aS'virtual' p70 aS'well' p71 aS'x' p72 aS'xx' p73 aS'yx' p74 a.
http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense	(lp0 (S'virtual' p1 S'a' p2 S'existing in essence or effect though not in actual fact' p3 a(S'said' p309 g16 S'give instructions to or direct somebody to do something with authority' p310 S"Synset('order.v.01')" p311 tp312 a(S'said' p313 g16 S'have or contain a certain wording or form' p314 S"Synset('read.v.02')" p315 tp316 a(S'continuous' p317 g2 S'of a function or curve; extending without break or irregularity' p318 S"Synset('continuous.a.02')" p319 tp320 a.
http://purl.org/spar/doco/ListOfAgents	(lp0 VKHALAGAIAfrican Virtual p1 aVVirtual p2 aVCreative Commons p3 aVLearning Activities p4 aVMeasure Theory Real Analysis p76 aVAnalysis p77 aVRiemann p78 aVLevel p79 aVHansdorff p80 aVMeasure Theory p81 aVLebesgue p82 aVRiemann Integral p83 aVAhave p84 aVReal AnalysisSpecific p85 aVLicense p86 aVAfrican p87 aVMain p88 aVMathematics p89 aVAnalysis p90 aVMathematical p91 aVLebesgue p92 aVDemonstrate p93 aVDevelop p94 aVBn p95 aVRiemann p96 aVab0iffab p97 aVNp p98 a.
http://purl.org/dc/elements/1.1/language	english

La tabla 22 muestra las propiedades que posee el OER “Analysis%202.pdf”, los valores correspondientes a http://purl.org/spar/doco/ListOfAgents, http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense, http://www.w3.org/ns/dcat#keyword, http://purl.org/spar/doco/Glossary, http://www.lexinfo.net/ontology/2.0/lexinfo#standardText, http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word representan un extracto del valor total que se encuentra en el triplestore Virtuoso.

El script que contiene esta tarea se lo puede observar en los Anexos 12 y 13.

5. Limitaciones y condiciones críticas de fallo

Necesita conexión a Internet para la descarga de los OER’s, disponibilidad y correcta instalación de todas las librerías y herramientas mencionadas en cada apartado.

CAPITULO 4: ENRIQUECIMIENTO DE DATOS SOBRE OER/OCW A TRAVÉS DE ENLACE CON LA NUBE DE DATOS ENLAZADOS ABIERTOS

1. Introducción

Obtener la data de los recursos educativos abiertos implica involucrar varias tareas que permitieron establecer las relaciones entre OER's, sin embargo es necesario enriquecer los datos obtenidos utilizando fuentes de datos externas como DBpedia.

Para sustentar esta premisa es necesario citar uno de los principios de Linked Data: *"Incluye enlaces a otros URI, para que puedan descubrir más cosas: Esta regla es necesaria para enlazar datos que se encuentran en la Web, de tal manera que no se queden aislados y así poder compartir la información con otras fuentes."* (W3C, 1999)

Este principio nos da las pautas necesarias, además de abarcar el objetivo al cual se desea llegar en el presente proceso; este capituló incluye en primera instancia las caracteristicas del dataset DBpedia, luego aborda la creación de la consulta Sparql y obtención de resultados, y finalmente el almacenamiento de data.

2. Propósito del proceso

Este proceso tiene como propósito enriquecer los tokens y entidades comunes obtenidas en el proceso anterior, mediante consultas Sparql se accederá al dataset DBpedia con la finalidad de tomar los enlaces que pertenezcan a Wikipedia.

3. Precondiciones para ejecutarlo

Se necesita que el proceso de "Extracción y Procesamiento de Información de OER's" se haya ejecutado previamente, para usar los tokens y entidades comunes de los OER's valores que serán utilizados para la búsqueda de datos en DBpedia.

4. Características del dataset DBpedia

DBpedia es el dataset más completo actualmente, razón más que necesaria para utilizarlo en el enriquecimiento de los datos extraídos de los Recursos Educativos Abiertos, para sustentar esta aseveración es importante mencionar sus características.

(DBpedia, 2013) Menciona que “en DBpedia solo en la versión en inglés se describen 3,77 millones de entidades, entre ellas al menos 764 mil personas, 563 mil lugares, 112 mil álbumes de música, 72 mil películas y 18 mil videojuegos.

Con todas las versiones se tienen 8 millones de enlaces a imágenes, 24,4 millones de enlaces a páginas externas, 27,2 millones de enlaces a datasets externos y 55,8 millones categorías de Wikipedia. Además de la información extraída de la versión en inglés, en junio de 2011 se implementó la extracción de información de otras versiones de Wikipedia, comenzando por 15 de estas, como la versión en: español, alemán, francés, entre otras. Para el 2013 dispone de 111 versiones en distintos idiomas.

DBpedia cumple los principios de Linked Data, por lo tanto se utilizan URIs para identificar una entidad por ejemplo: http://xx.dbpedia.org/resource/Name, donde “xx” es el código del lenguaje de Wikipedia y “Name” es el nombre de la entidad a buscar tomada de la URL original http://xx.wikipedia.org/wiki/Name.”

Mencionadas estas características sobre DBpedia podemos concluir que es un dataset completo, estructurado, y el más idóneo para enriquecer la data que se ha obtenido de los OER’s.

5. Pasos generales a ejecutar

Los pasos que se debe ejecutar durante este proceso son pocos y con un grado de dificultad mínimo, estos son:

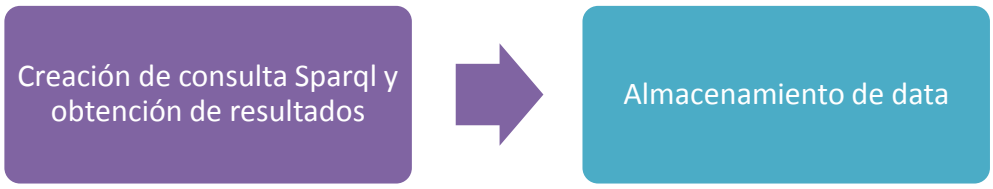


Figura 51: Proceso de enriquecimiento

5.1. Creación de la consulta SPARQL y Obtención de resultados

Enriquecer los datos obtenidos de los recursos educativos abiertos implica cumplir con los principios de Linked Data específicamente con el siguiente: “Incluir enlaces a otros URI, para que puedan descubrir más cosas: Esta regla es necesaria para enlazar datos que se encuentran en la Web, de tal manera que no se queden aislados y así poder compartir la información con otras fuentes.” (W3C, 1999) Cumplir con este objetivo es el proposito durante esta tarea.

5.1.1. Ejecución

Los valores que se utilizarán para realizar las consultas son los tokens y entidades comunes obtenidas del proceso de “Extracción y Procesamiento de Información de OER’s/OCW”; la imagen siguiente muestra un extracto de los valores disponibles para enriquecer.

Figura 52: Extracto de los tokens y entidades comunes disponibles para enriquecer

Por lo tanto es preciso recordar que todo concepto que se desea consultar es un recurso en la web (Resource), para acceder a información relacionada sobre un recurso en DBpedia mediante consultas SPARQL se utiliza la URI ***http://dbpedia.org/resource/*** más el nombre del recurso que se desea consultar en donde el recurso será cada palabra o entidad común. Como ejemplo se toma la palabra *"Human".*

Una de las opciones de consulta sparql se presenta acontinuación, y su respectivo resultado se muestra en la tabla 24.

```
prefix dbpedia: <http://dbpedia.org/resource/>
select distinct *
where {dbpedia:Human ?propiedad ?objeto}
```

Tabla 24: Extracto del resultado de la consulta SPARQL sobre la palabra "Human"

PROPIEDAD	OBJETO
http://dbpedia.org/property/species	"Homo sapiens"@en
http://dbpedia.org/property/status	"LC"@en
http://dbpedia.org/property/statusSystem	"iucn3.1"@en
http://dbpedia.org/property/subdivision	"Homo sapiens sapiens"@en
http://dbpedia.org/property/subdivision	"Homo sapiens idaltu White et al., 2003"@en
http://dbpedia.org/property/subdivisionRanks	http://dbpedia.org/resource/Subspecies
http://dbpedia.org/property/taxon	"Homo sapiens"@en
http://dbpedia.org/property/upper	2.1
http://dbpedia.org/property/v	"no"@en
http://xmlns.com/foaf/0.1/isPrimaryTopicOf	http://en.wikipedia.org/wiki/Human
http://www.w3.org/ns/prov#wasDerivedFrom	http://en.wikipedia.org/wiki/Human?oldid=548414736
http://dbpedia.org/property/hasPhotoCollection	http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/photos/Human
http://dbpedia.org/ontology/wikiPageInLinkCount	5607
http://dbpedia.org/ontology/wikiPageOutLinkCount	841

Fuente: DBpedia

Otra opción de consulta Sparql se puede observar en la figura 53 y su correspondiente resultado en la figura 54, los resultados que retorna esta consulta no son los mas idóneos para utilizarlos, sin embargo si se selecciona uno de los valores que se listan por ejemplo “http://dbpedia.org/resource/Human” nos presentaran los datos en formato rdf como se muestra en la tabla 24.

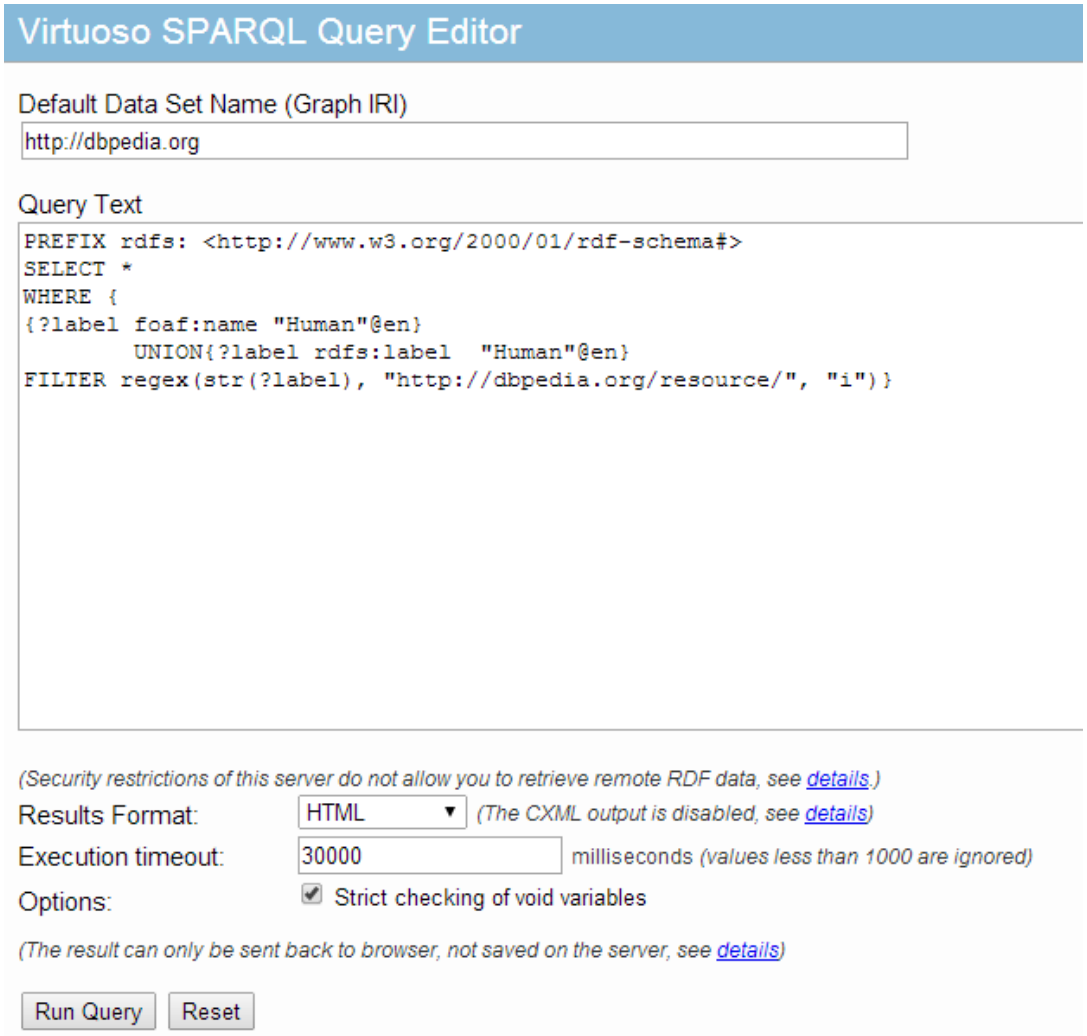


Figura 53: Consulta Sparql a Dbpedia

label
http://dbpedia.org/resource/Human_(Goldfrapp_song)
http://dbpedia.org/resource/Human_(The_Human_League_song)
http://dbpedia.org/resource/Human_(Brandy_Norwood_album)
http://dbpedia.org/resource/Human_(Gary_Numan_album)
http://dbpedia.org/resource/Human_(Rod_Stewart_album)
http://dbpedia.org/resource/Human_(Skye_Sweetnam_song)
http://dbpedia.org/resource/Human_(1971_film)
http://dbpedia.org/resource/Human_(Rachael_Lampa_album)
http://dbpedia.org/resource/Human_(The_Killers_song)
http://dbpedia.org/resource/Human_(EP)
http://dbpedia.org/resource/Human_(Stargate_Universe)
http://dbpedia.org/resource/Human_(Death_album)
http://dbpedia.org/resource/Human_(Projected_album)
http://dbpedia.org/resource/Human

Figura 54: Resultado de la consulta sparql a Dbpedia

Se debe acotar que estas consultas son sensibles a mayúsculas y minúsculas, si se desea realizar la misma búsqueda con la palabra “human” (h en minúscula), no devolverá ningún resultado. Por lo tanto se necesita consultar la palabra en mayúscula y en minúscula para mayor seguridad.

De las dos opciones de consultas sparql presentadas se toma la primera opción porque es mas simple y concreta para usar, además contribuirá a mejorar el rendimiento del proceso de enlazado.

De las propiedades y objetos que retorna la primera consulta ver tabla 24, se utiliza la propiedad **“http://xmlns.com/foaf/0.1/isPrimaryTopicOf”** cuyo objeto es la URI a Wikipedia en este caso “*http://en.wikipedia.org/wiki/Human*”.

Con esta aclaración la consulta SPARQL final que se utiliza para enriquecer las palabras y entidades comunes de los recursos educativos abiertos se indica a continuación, y su resultado se puede observar en la tabla 25:

```
prefix dbpedia: <http://dbpedia.org/resource/>
select distinct *
```

where{dbpedia:Human <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?o}

Tabla 25: Resultado de la consulta SPARQL

O
http://en.wikipedia.org/wiki/Human

Con esta URI obtenida se está enriqueciendo nuestra data con un enlace externo a Wikipedia, cabe recalcar que realizar consultas al dataset de DBpedia da al proceso un nivel de lentitud considerable, debido a la dependencia de conectividad y disponibilidad de este repositorio.

Como principales herramientas que intervienen en este proceso tenemos:

- **SPARQLWrapper:** es de gran ayuda para realizar las consultas Sparql hacia repositorios externos como DBpedia.
- **Conexión a Internet:** la consulta se realiza hacia un repositorio externo, para lo cual se necesita una buena conexión para minimizar el tiempo de respuesta.

5.1.2. Resultados

Los siguientes ejemplos muestran las consultas realizadas a DBpedia, con su respectivo resultado.

Consulta SPARQL: prefix dbpedia: <http://dbpedia.org/resource/> select distinct ?o where {dbpedia:**n**eed <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?o}

Esta consulta no devuelve ningún valor, porque la primera letra de la palabra a consultar esta en minuscula. Se puede ver la diferencia en la siguiente consulta.

Consulta SPARQL: prefix dbpedia: <http://dbpedia.org/resource/> select distinct ?o where {dbpedia:**N**eed <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?o}

Figura 55: Consulta realizada a DBpedia con el token Need

Consulta SPARQL: prefix dbpedia: <http://dbpedia.org/resource/> select distinct ?o where {dbpedia:Computers <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?o}

Figura 56: Consulta realizada a DBpedia con el token Computers

El script que contiene esta tarea se lo puede observar en los Anexos 5 y 14.

5.2. Almacenamiento de data

El propósito de esta tarea es guardar los datos a enriquecer en el triplestore Virtuoso.

5.2.1. Ejecución

La URI obtenida se la almacena en nuestro triplestore Virtuoso, la tripleta que representa a este valor se puede observar en la tabla siguiente:

Tabla 26: Tripleta que se utiliza para el almacenamiento de la URI obtenida de DBpedia

SUJETO	PREDICADO	OBJETO
http://dataoers.org/oer/commonword/ o http://dataoers.org/oer/commonentity/	http://www.w3.org/2002/07/owl#sameas	URI de DBpedia

En donde a la URI del sujeto se le agrega la palabra o entidad en común de la cual se esta realizando el proceso de enriquecimiento, la URI del predicado corresponde a la reutilización del vocabulario RDF (ver capitulo 3, sección 4.6), finalmente la URI del objeto corresponde al valor obtenido de la consulta a DBpedia (URI de Wikipedia).

5.2.2. Resultados

La siguiente imagen muestra un extracto de los resultados obtenidos.

s	o
http://dataoers.org/oer/commonword/need	http://en.wikipedia.org/wiki/Need
http://dataoers.org/oer/commonword/study	http://en.wikipedia.org/wiki/Study
http://dataoers.org/oer/commonword/closed	http://en.wikipedia.org/wiki/Closed
http://dataoers.org/oer/commonword/square	http://en.wikipedia.org/wiki/Square
http://dataoers.org/oer/commonword/computer	http://en.wikipedia.org/wiki/Computer
http://dataoers.org/oer/commonword/reduction	http://en.wikipedia.org/wiki/Reduction
http://dataoers.org/oer/commonword/tool	http://en.wikipedia.org/wiki/Tool
http://dataoers.org/oer/commonword/sites	http://en.wikipedia.org/wiki/Sites
http://dataoers.org/oer/commonword/potential	http://en.wikipedia.org/wiki/Potential
http://dataoers.org/oer/commonword/exercise	http://en.wikipedia.org/wiki/Exercise
http://dataoers.org/oer/commonword/characteristics	http://en.wikipedia.org/wiki/Characteristics
http://dataoers.org/oer/commonword/lesson	http://en.wikipedia.org/wiki/Lesson
http://dataoers.org/oer/commonword/mole	http://en.wikipedia.org/wiki/Mole
http://dataoers.org/oer/commonword/family	http://en.wikipedia.org/wiki/Family
http://dataoers.org/oer/commonword/matter	http://en.wikipedia.org/wiki/Matter
http://dataoers.org/oer/commonword/sources	http://en.wikipedia.org/wiki/Sources
http://dataoers.org/oer/commonword/follow	http://en.wikipedia.org/wiki/Follow
http://dataoers.org/oer/commonword/heat	http://en.wikipedia.org/wiki/Heat
http://dataoers.org/oer/commonword/include	http://en.wikipedia.org/wiki/Include
http://dataoers.org/oer/commonword/numerical	http://en.wikipedia.org/wiki/Numerical
http://dataoers.org/oer/commonword/systems	http://en.wikipedia.org/wiki/Systems
http://dataoers.org/oer/commonword/learning	http://en.wikipedia.org/wiki/Learning
http://dataoers.org/oer/commonword/make	http://en.wikipedia.org/wiki/Make

Figura 57: Extracto del enriquecimiento de los tokens o entidades comunes

El script que contiene esta tarea se lo puede observar en el Anexo 14.

6. Limitaciones y condiciones críticas de fallo

Una de las limitaciones es la necesidad de la ejecución previa del proceso de “Enriquecimiento y Procesamiento de Información de OER’s”, sin esta información no sería posible el proceso de enriquecimiento y su posterior visualización.

El proceso de enriquecimiento depende de la disponibilidad del dataset Dbpedia para su correcto funcionamiento, caso contrario los datos no serán enriquecidos.

En cuanto al rendimiento del proceso de enriquecimiento la ejecución es relativa a la conexión a Internet y la disponibilidad de Dbpedia; al ejecutar el proceso de enriquecimiento en el servidor el tiempo que utilizó fueron de alrededor 15 minutos para una cantidad aproximada de 2383 palabras disponibles a enriquecer.

CAPITULO 5: VISUALIZACIÓN DE INFORMACIÓN DE OER/OCW

1. Introducción

El presente capitulo comprende el proceso de visualización de la información obtenida de los OER's, aborda también la implementación de un web service necesario para el correcto funcionamiento de este proceso.

Se indicará la ejecución, las herramientas a utilizar y los resultados obtenidos durante el proceso de Visualización de Información de OER's pertenecientes a OCW.

Además para concluir el desarrollo del proyecto sobre "Análisis y Visualización de Recursos Educativos Abiertos contenidos en sitios OCW" se establece una sección de discusión sobre el trabajo realizado.

2. Propósito del proceso

El propósito del procesos de Visualización es presentar al usuario final las relaciones existentes entre los recursos educativos abiertos, resultados que se presentaran utilizando visualización de redes.

3. Precondiciones para ejecutarlo

Se necesita de un servicio web que retorne el resultado en formato JSON de la consulta SPARQL que se realiza sobre los predicados existentes y almacenados en el triplestore Virtuoso. Además se necesita que la herramienta para graficar JIT esté disponible para ser utilizada.

4. Pasos generales a ejecutar

Se utiliza la visualización de redes porque permite al usuario final entender y observar las relaciones entre los elementos con facilidad, características que la convierten en una opción idónea para alcanzar el proposito de este proceso.

Es necesario establecer las tareas que deben ejecutarse para el proceso de Visualización, en la gráfica siguiente se las menciona.

Figura 58: Proceso de Visualización

4.1. Implementación de un Web Service

El objetivo de esta tarea es crear un web service que retorne un JSON bien estructurado, que contenga el resultado de la consulta Sparql que utiliza como parámetro los predicados disponibles (ver tabla 18).

4.1.1. Ejecución

Un servicio web sirve para intercambiar datos entre aplicaciones, aporta gran independencia entre la aplicación que usa el servicio y el propio servicio, utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Los valores que utiliza el servicio web, y el valor que retorna son:

- **Entrada de datos:** recibe los datos obtenidos de la consulta SPARQL sobre las relaciones existentes entre los OER's, sin embargo se puede hacer consultas filtrando los datos por el predicado que se desee.
  - ✓ La consulta sparql utilizada es: select distinct * where {?sujeto < http://purl.org/dc/elements/1.1/relation> ?objeto} Limit 300

    Se limita el resultado a 300 relaciones debido a la lentitud de carga de todos los datos, la tabla siguiente muestra un extracto del resultado de la consulta.

s	o
http://oer.avu.org/bitstream/handle/123456789/85/Multimedia%20Design%20and%20Applications.pdf?sequence=3	http://oer.avu.org/bitstream/handle/123456789/84/Graphics%20and%20Information%20Man sequence=3
http://oer.avu.org/bitstream/handle/123456789/251/Mathematical-Physics1vf-Readings.pdf?sequence=1	http://oer.avu.org/bitstream/handle/123456789/59/Quantum%20Mechanics.pdf?sequence=1
http://oer.avu.org/bitstream/handle/123456789/55/Mechanics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/59/Quantum%20Mechanics.pdf?sequence=1
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_stat-ch01.pdf	http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_ro
http://oer.avu.org/bitstream/handle/123456789/31/Thermal%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/27/Properties%20of%20Matter.pdf?sequence=
http://oer.avu.org/bitstream/handle/123456789/29/Statistical%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/23/Atomic%20Physics.pdf?sequence=4
http://oer.avu.org/bitstream/handle/123456789/31/Thermal%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/23/Atomic%20Physics.pdf?sequence=4
http://oer.avu.org/bitstream/handle/123456789/26/Nuclear%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/23/Atomic%20Physics.pdf?sequence=4
http://oer.avu.org/bitstream/handle/123456789/28/Solid%20State%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/23/Atomic%20Physics.pdf?sequence=4
http://oer.avu.org/bitstream/handle/123456789/50/Plant-animal-physiologyVFreadings.pdf?sequence=1	http://oer.avu.org/bitstream/handle/123456789/78/Educational%20Evaluation%20and%20Te
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604c870c808b860/3.pdf	http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-food-trade-theory/314_0415%20Special_2.pdf	http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-trade-nego 1/067_0325%20ADD_2.pdf
http://oer.avu.org/bitstream/handle/123456789/251/Mathematical-Physics1vf-Readings.pdf?sequence=1	http://oer.avu.org/bitstream/handle/123456789/56/Mathematical%20Physics%202%20-%20R
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-trade-negotiations-1/067_0408%20GATT%20Principle_3.pdf	http://oer.avu.org/bitstream/handle/123456789/56/Mathematical%20Physics%202%20-%20R
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_stat-ch03.pdf	http://oer.avu.org/bitstream/handle/123456789/56/Mathematical%20Physics%202%20-%20R
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-food-trade-theory/314_0408%20Ch4%20Heckscher%20Ohlin_1.pdf	http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/international-food-trade theory/314_0415%20Special_2.pdf
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_intro.pdf	http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_in
http://oer.avu.org/bitstream/handle/123456789/46/Organic%20Chemistry%202.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/38/Separation%2c%20Electroanalytical%20and%20Spectrometric%20Techniques.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/47/Physical%20Chemistry%201.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/43/Chemistry%202%20-%20Introductory%20General.pdf?sequence=6	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/36/Microbiology%20and%20Mycology.pdf?sequence=3	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/44/Macromolecules%20in%20Biological%20Systems.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/26/Nuclear%20Physics.pdf?sequence=4	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://oer.avu.org/bitstream/handle/123456789/52/Volumetric%20Chemical%20Analysis.pdf?sequence=5	http://oer.avu.org/bitstream/handle/123456789/48/Physical%20Chemistry%202.pdf?sequence
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604c870c808b860/4.pdf	http://oer.avu.org/bitstream/handle/123456789/50/Plant-animal-physiologyVFreadings.pdf?se
http://oer.avu.org/bitstream/handle/123456789/33/Diversity%20of%20Algae%20and%20Plants.pdf?sequence=8	http://oer.avu.org/bitstream/handle/123456789/50/Plant-animal-physiologyVFreadings.pdf?se
http://ocw.korea.edu/ocw/college-of-life-sciences-and-biotechnology/c720c804c790bc1cd604c870c808b860/12.pdf	http://oer.avu.org/bitstream/handle/123456789/50/Plant-animal-physiologyVFreadings.pdf?se
http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_root.pdf	http://ocw.korea.edu/ocw/college-of-science/c804c0b0bb3cb9acd559-bc0f-c2e4d5d8/cphy_st

Figura 59: Extracto de las relaciones existentes entre OER's

- **Salida de datos:** los datos que retorna este servicio web es un JSON válido y bien estructurado, que contiene:
    - ✓ La URI de cada OER denominado "*node*" y el nombre correspondiente.
    - ✓ Sus respectivas relaciones denominadas "*adjacencies*" y el color que se utilizará para la gráfica.
    - ✓ Datos para gráficar los nodos como: el color, la dimensión y la forma que se utilizará para el nodo. (ver figura 44).

  Cabe acotar que para la creación de la estructura del JSON se siguió el formato establecido por la herramienta JIT (encargada de crear la gráfica con estos datos).

Para la creación del Web Service se necesita:

- **Instalar el entorno virtual para Python:** este paso es recomendable y básico en un entorno de desarrollo Python. Permite instalar todas las librerías que se utilizaran durante el desarrollo de una aplicación, y realizar configuraciones sin afectar al sistema.
- **Instalación de Flask:** es el framework de Python que se utiliza para crear el servicio web.
- **Configuración para Apache:** esta acción se la realiza en función del módulo mod_wsgi que se utiliza para el web service Flask.
- **Asignar el nombre para el virtual host:** este parámetro es la ruta con la que se accederá públicamente al web service, el valor establecido es: ***taw02.utpl.edu.ec/anavisoers/webservice*** con este ruta se puede acceder al webservice desde cualquier navegador.

Las herramientas utilizadas para esta tarea son:

- **Virtualenv:** entorno virtual idóneo para desarrollo de aplicaciones, permite realizar configuraciones e instalaciones de librerías sin afectar al sistema.

- **Flask:** framework de Python con licencia BSD, los datos que retorna este servicio web es un JSON que contiene el resultado de la consulta Sparql, en este caso de las relaciones existentes entre los OER's; sin embargo se puede hacer una consulta filtrando los datos por el predicado que se desee graficar.
- **Mod_wsgi para Apache:** configuración necesaria para el correcto funcionamiento de Flask.
- **Configuración de Apache:** creación de virtual host para el web service y también se creó un puerto específico para el funcionamiento de Flask.

4.1.2. Resultados

Se puede acceder al web service desde cualquier navegador con conexión a internet con la ruta ***http://taw02.utpl.edu.ec/anavisoers/webservice***. En la imagen siguiente se puede observar al JSON debidamente estructurado y listo para ser utilizado por la herramienta JIT (JavaScript Infovis Toolkit)

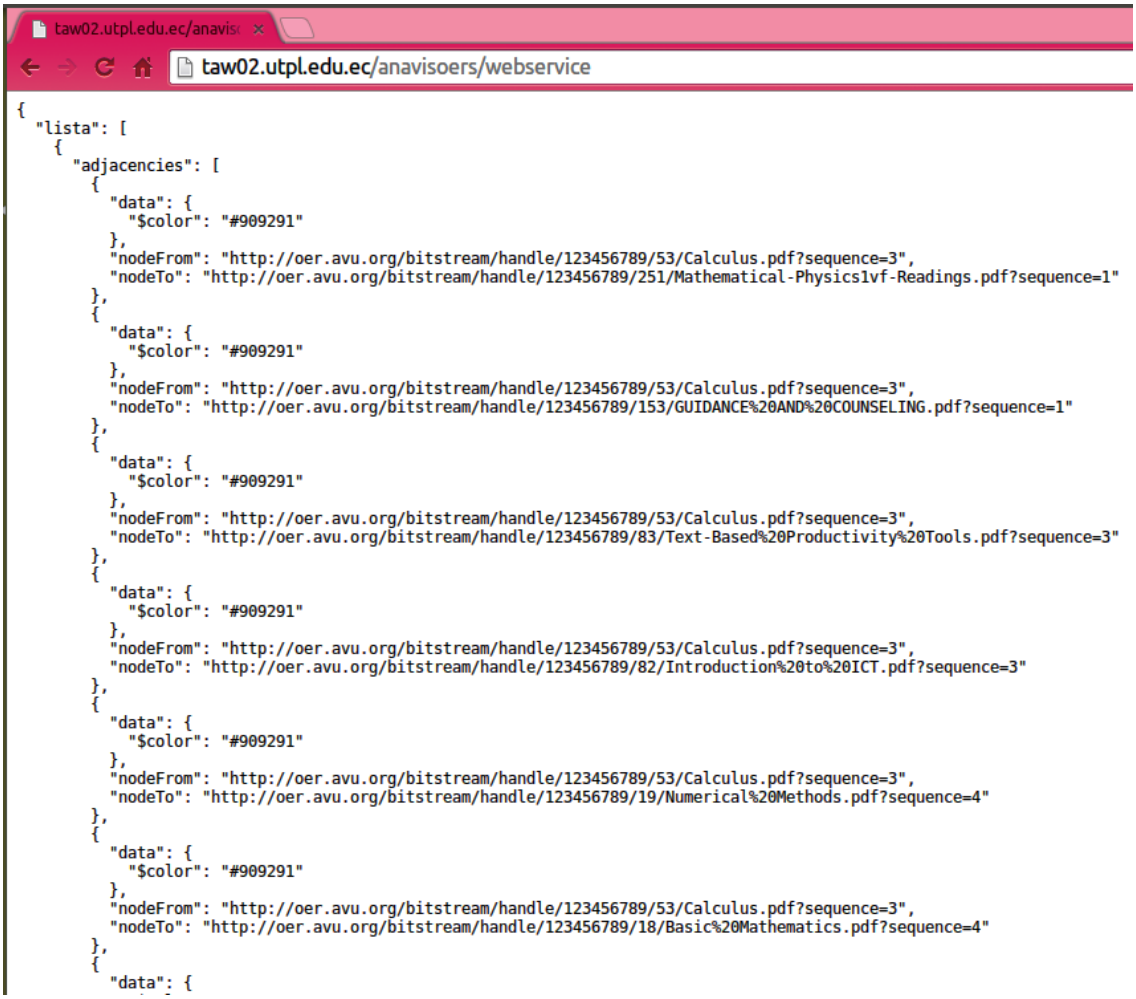


Figura 60: JSON Resultado del Web Service.

El script que contiene esta tarea se lo puede observar en los Anexos 15 y 16.

4.2. Visualización de Información de OER's

El propósito de este proceso es visualizar las relaciones entre OER's usando el valor que retorna el web service; mediante la herramienta JIT.

4.2.1. Ejecución

Por la gran cantidad de relaciones existentes entre OER's solamente se visualizará un extracto de estos datos.

Durante la ejecución de este paso es necesario contar con la disponibilidad de:

- **Web service:** debe retornar los valores bien estructurados y en formato JSON que serán utilizados por la herramienta JIT.
- **Librería JIT - ForceDirected:** Se utiliza la herramienta JIT (JavaScript InfoVis Toolkit) concretamente el demo ForceDirected, la particularidad que nos presenta esta herramienta es la interacción con el grafo.

El demo ForceDirected utiliza en la grafica dos principales elementos:

- **Nodos:** representados con circulos que identifican a un OER.
- **Aristas:** son lineas que sirven para relacionar un nodo con otro.

Si se hace clic en uno de los nodos se presenta en la parte derecha el OER seleccionado y la lista correspondiente a las conexiones ó relaciones que posee, al hacer clic en una de las relaciones listadas se podrá descargar el OER con el cual se relaciona.

Las herramientas utilizadas para esta tarea son:

- JIT: Para implementar la visualización de información de los recursos educativos abiertos se utilizó la herramienta JIT (JavaScript InfoVis Toolkit) concretamente el demo ForceDirected.
- Web Service: correcto funcionamiento del servicio web y disponibilidad de los datos que retorna.

4.2.2. Resultados

La figura 44 muestra las relaciones existentes entre OER's, es necesario mencionar que por la gran cantidad de relaciones solamente se visualizará un extracto de las mismas.

Para una mejor apreciación desde cualquier navegador con acceso a Internet se puede observar este visualizador, usando la siguiente ruta ***http://taw02.utpl.edu.ec/anavisoers/visualizador***.

Los scripts adaptados y creados para realizar esta tarea se lo puede observar en los Anexos 17 y 18.

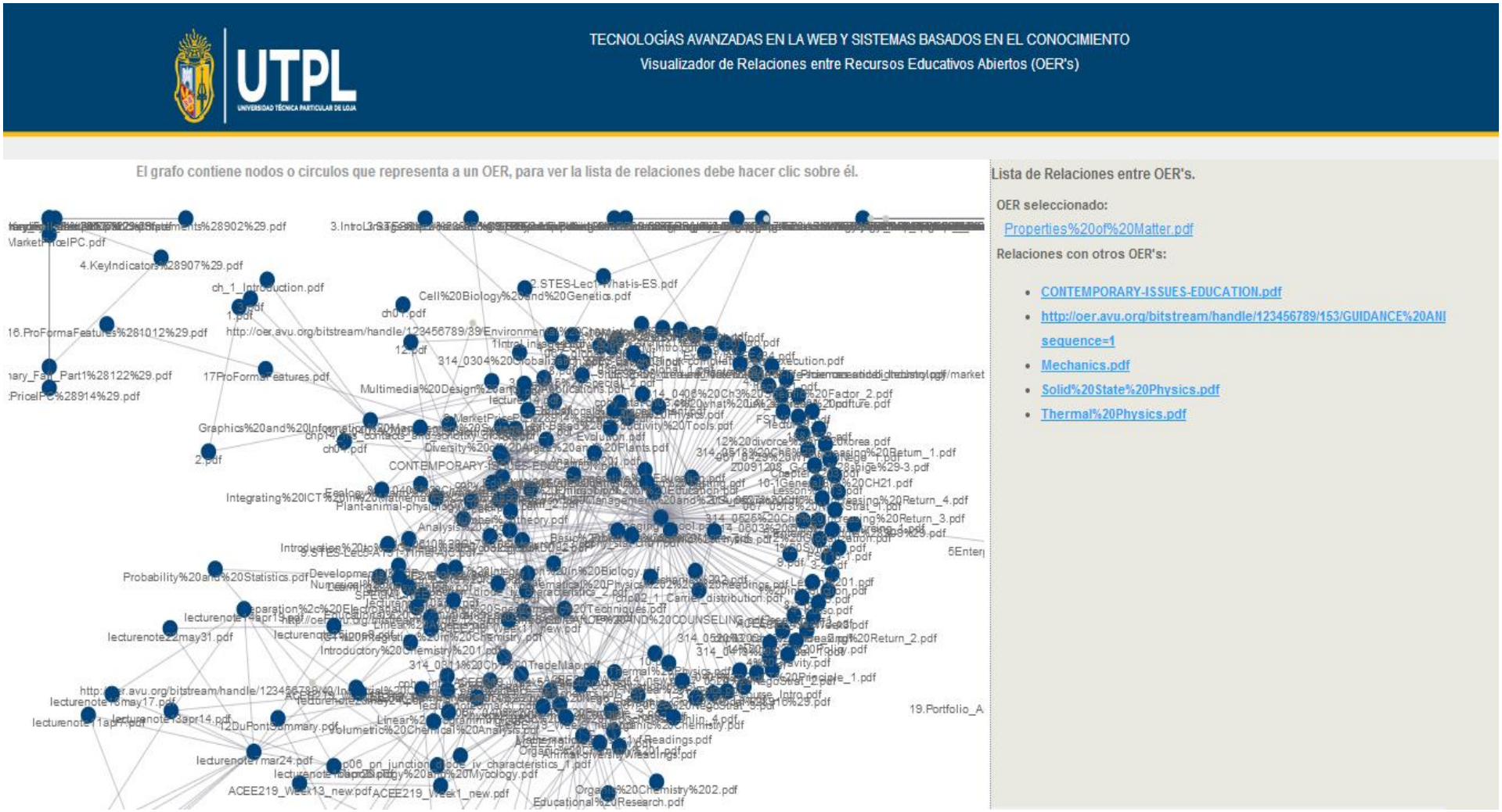


Figura 61: Visualizador de las relaciones entre OER's

DISCUSIÓN FINAL

Mediante la extracción de texto de los recursos educativos abiertos se pudo tokenizar, taggear, identificar las palabras más representativas, obtener entidades, además se estableció el vocabulario RDF que identificará a la data obtenida, se adquirió también la metadata como el título del OER, el número de páginas, y el idioma; información que permite enriquecer a la base de datos inicial sobre OCW, agregando información detallada de cada recurso educativo abierto.

Se estableció parámetros para a) la extracción de texto utilizando el número de páginas del OER, con la finalidad definir la cantidad necesaria de texto que será extraída; también se utiliza b) el idioma del OER para utilizar solamente aquellos que estén en inglés, y finalmente c) las palabras representativas que se utilizan para el procesamiento de información.

Durante el desarrollo del procesamiento de información se pudo resolver el principal problema de ambigüedad semántica, mediante el uso del diccionario en inglés Princeton WordNet y la implementación del algoritmo simplificado de Lesk; sin embargo el tiempo de respuesta del proceso es muy lenta dando lugar a la intervención de las palabras representativas del OER, cuando se identifica una palabra ambigua esta debe ser parte del grupo de palabras representativas para poder pasar al proceso de desambiguación caso contrario se la ignora de esta forma el tiempo de respuesta del proceso mejora.

Al finalizar el proceso de desambiguación se logró la creación de relaciones mediante el establecimiento de palabras o entidades comunes, se considera una palabra o entidad común cuando dos o mas OER's la contienen con el mismo significado establecido previamente en el proceso de desambiguación y se repite mas de 2 veces; a la relación se la creará siempre y cuando se cumpla lo mencionado.

Mediante el uso del dataset DBpedia se consiguió enriquecer los datos sobre OER's, esto utilizando las palabras y entidades comunes identificadas en el proceso de "Extracción y Procesamiento", mediante consultas SPARQL.

Mediante la implementación de la herramienta JIT se pudo obtener la visualización, esto con la creación de un web service que que retorna las relaciones existentes entre OER’s en formato JSON.

CONCLUSIONES

Mediante la presentación de información a partir de visualizaciones sean grafos, diagramas, o gráficas, se pude fácilmente inferir el contexto, las tendencias o la naturaleza del OER que se selecciona, por ejemplo: al seleccionar un nodo que representa a un OER llamado “Properties%20of%20Matter.pdf” las relaciones hacia el mismo nos permiten inferir que el contexto de dicho recurso seria en el ámbito educativo.

Mediante la visualización se puede analizar la tendencia de un conjunto de recursos educativos abiertos dentro de una base de datos específica, por ejemplo los OER’s de una organización particular como la UTPL tienden a estar orientados hacia la educación y la investigación, aquellos nodos que se encuentran alejados del conjunto central nos dan una idea sobre aquellos temas que no son tan recurrentes dentro de la universidad. De esta forma se puede desde una base de datos correspondiente a una institución fabricar visualizaciones que nos permitan inferir rápidamente su ámbito de trabajo o el enfoque de trabajo que poseen.

Mediante pln (procesamiento de lenguaje natural) se puede enriquecer la data que se encuentra en una base de datos, puesto que mediante métodos de extracción y procesamiento se puede adicionar información sobre la ya existente, por ejemplo, mediante el análisis (tokenización y taggeo) de un texto se pueden encontrar conceptos, entidades, personas, países, entre otros, que permiten ampliar el contexto de un recurso o también en el caso de consultas especializar aún más la búsqueda.

Al extraer conceptos a partir del texto de los OER’s se tiene una probabilidad más alta de relacionar los mismos mediante el consumo de otros servicios, por ejemplo, del OER llamado “Analysis%201.pdf” al principio se podría decir que no tenía relación con alguna otra entidad, luego de haber sido procesado se extrajeron palabras comunes que sirven como tags de relación entre este OER y otros, además a estas palabras o tags se los enriqueció mediante consultas a la Dbpedia lo que permitió ampliar el contexto de dicho OER.

RECOMENDACIONES

Establecer un estándar para asignar los nombres a los OER's, de tal manera que se pueda evitar nombres vagos y sin referencia alguna al recurso, además incentivar a los estudiantes el uso de Recursos Educativos Abiertos como su principal fuente de información, y acceso al conocimiento.

Realizar un estudio previo de los vocabularios RDF disponibles, con la finalidad de seleccionar los más idóneos y que se adapten a nuestra necesidad, tomando en cuenta que deben ser de fuentes confiables.

Es de suma importancia crear un ambiente de desarrollo Python para la creación de algoritmos, de esta manera se podrá manipular con mayor confianza las librerías sin afectar al Sistema Operativo, todas las librerías que se utilicen en el desarrollo de una aplicación en el lenguaje de programación Python deben estar correctamente instaladas y siempre trabajando con la versión correcta, igualmente es importante adquirir un conocimiento previo sobre la estructura del lenguaje natural para posteriormente empezar con el uso de la herramienta NLTK.

Se recomienda como trabajo futuro desarrollar una visualización mas avanzada que contemple filtros de búsqueda para los datos que se deseen mostrar, que incluya por ejemplo: los predicados disponibles, las palabras comunes, entidades, palabras representativas, entre otros; con la finalidad de permitirle al usuario acceder de una manera mas detallada y gráfica a la data obtenida a partir de los OER's.

Como trabajo futuro se propone el desarrollo de un plugin con las funcionalidades desarrolladas en este proyecto y las recomendaciones mencionadas, con la finalidad de poderlo agregar en sitios OpenCourseWar, para que el usuario pueda visualizar otras opciones referentes al tema que este investigando, ahorrándole tiempo en búsqueda, y generando un servicio extra para cada OCW.

BIBLIOGRAFÍA

Berners-Lee, T. (agosto de 1996). *www.w3c.org.* Recuperado el 8 de agosto de 2013, de Actas de la V Conferencia Internatonal World Wide Web: www.w3c.org

Berners-Lee, T. (6 de diciembre de 2000). *Architecture*. Recuperado el 12 de agosto de 2013, de Semantic Web on XML: http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

Berners-Lee, T. (2005). *Uniform Resource Identifier (URI).* Recuperado el 12 de agosto de 2013, de http://tools.ietf.org/html/rfc3986

Berners-Lee, T. (febrero de 2009). *Tim Berners-Lee on the next web.* Recuperado el 3 de febrero de 2014, de http://www.ted.com/talks/tim_berners_lee_on_the_next_

Bird, S., & Loper, E. (15 de octubre de 2012). *Natural Language Processing with Python*. Obtenido de http://nltk.org/book

Bolshakov, I., & Gelbukh, A. (2004). Computational Linguistics. Models, Resources, Applications. *Ciencia de la Computacion Primera Edición*.

Brandes, U., Kenis, P., & Raab, J. (diciembre de 2005). *REDES- Revista hispana para el análisis de redes sociales.* Recuperado el 8 de marzo de 2014, de http://revista-redes.rediris.es

Castells, P. (2005). *La Web Semántica.* Madrid.

Corcho, O., & Gómez, A. (2010). *Mini-Curso sobre Linked Data.* Recuperado el 16 de febrero de 2014, de slideshare: http://www.slideshare.net/ocorcho/linked-data-tutorial-florianpolis

DBpedia. (17 de septiembre de 2013). *The DBpedia Data Set*. Recuperado el 28 de febrero de 2014, de http://wiki.dbpedia.org/Datasets

Espinoza, K., Martinez, Y., & Racine, K. (2013). *Visualización de redes: herramientas y tecnicas para la creacion y evalucion visual de las redes.* Panamá.

Foley, & Kibasky. (1994). *Visualización de Redes.*

Fonseca, J. M., Hierro, J. J., & Romo, P. Á. (2009). La Web Semántica, la siguiente generación de Webs. *Telos, Cuadernos de Comunicación e Innovación*, 2.

Galicia-Haro, S. (2000). *Análisis sintático conducido por un diccionario de patrones de manejo sintáctivo para lenguaje español.* Mexico: Tesis doctoral.

Grettel Barceló, A. (junio de 2010). *Desambiguación de los sentidos de las palabras en español usando texto paralelos.* Recuperado el julio de 2013, de http://www.gelbukh.com/thesis/Grettel%20Barcelo%20Alonso%20-%20PhD.pdf

Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition Journal Vol. 5*, 199-200.

Gruber, T. (1993). Towards Principles for the Design of Ontologies used for Knowledge Sharing. *Proc. Of International Workshop on Formal Ontology* (págs. 93-94). Italy: Knowledge Systems Laboratory, Standford University.

Hogan, A., Harth, A., Passant, S., & Polleres, A. (2010). Weaving the Pedantic web. *Linked Data on the Web Workshop (LDOW2010).*

Ide, N., & Véronis, J. (1998). *Word sense disambiguation: the state of the art. Computational Linguistics.*

Kilgarriff, A., & Rosenzweig, J. (2000). Framework and results for English SENSEVAL. *Computers and the Humanities*, (págs. 1 - 2).

López, J., Piedra, N., Sancho, E., Soto, Ó., & Tovar, E. (2012). *Aplicación de tecnologías web emergentes para el estudio del impacto de repositorios OpenCourseWare españoles y latinoamericanos en la Educación Superior.* España.

Novillo, A. (2013). *Aplicación de técnicas de Linked Data para la publicación de datos enlazados de Open Education Resources encontrados en Opencourseware.* Loja: UTPL.

OpenCourseWare UTPL. (2012). *OpenCourseWare UTPL.* Recuperado el 22 de 8 de 2013, de http://ocw.utpl.edu.ec/

Piedra, N., Tovar, E., López, J., Chicaiza, J., & Martinez, O. (4 de abril de 2011). *Next Generation "Open" Learning.* Recuperado el 8 de Agosto de 2013, de sitio web de OpenCourseWare Consortium Global Meeting: http://conference.ocwconsortium.org/index.php/2011/cambridge/paper/view/162

Qaissi, H. (17 de junio de 2009). Recuperado el 18 de marzo de 2014, de http://sinbad.dit.upm.es/docencia/doctorado/curso0809/HichamSPARQLrevCarmen.pdf

Rico, M. (24 de marzo de 2013). *Ejemplos de consultas SPARQL*. Recuperado el 26 de febrero de 2014, de http://es.dbpedia.org/Wiki.jsp?page=Ejemplos%20de%20consultas%20SPARQL

Russell, S. J., & Peter, N. (2003). *Inteligencia Artificial un enfoque moderno (Segunda edición).* Person Education.

Salton, G. (1968). Automatic Information organization and Retrival. New Cork: MacGraw-Hill.

Sidorov, G. (2005). Etiquetador Morfológico y Desambiguador Manual: Dos Aplicaciones del Analizador Morfológico Automático para el Español. *En Memorias del VI encuentro internacional de computacion*, (págs. 147-149). Mexico.

Torres, S. (2009). *Optimización global de coherencia en la desambiguacion del sentido de las palabras.* México.

Tufte, E. (1983). Graphics Press. *The Visual Display of Quantitative Information.* Connecticut.

Unesco. (2011). *A Basic Guide to Open Educational Resources.* Recuperado el 22 de 8 de 2013, de www.unesco.org/education

Villazón, B., Vilches, L., Corcho, O., & Gómez, A. (2011). *Methodological Guidelines for Publishing Goverment Linked Data.* Recuperado el 23 de abril de 2014, de https://www.lri.fr/~hamdi/datalift/tuto_inspire_2012/Suggestedreadings/egovld.pdf

W3C. (22 de febrero de 1999). *Resource Description Framework*. Recuperado el 14 de agosto de 2013, de http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

W3C Oficina Española. (14 de julio de 2009). *Guías Breves: Linked Data*. Recuperado el 9 de agosto de 2013, de sitio web de Oficina Española W3C: http://www.w3c.es/Divulgacion/GuiasBreves/LinkedData

W3C Oficina Española. (2010). *Guia Breve: Web Semántica.* Recuperado el 15 de agosto de 2013, de sitio de W3C Oficina Española: http://www.w3c.es/divulgacion/guiasbreves/linkeddata

ANEXOS

1. Arquitectura la Aplicación

La arquitectura de la aplicación se indica en la siguiente gráfica.

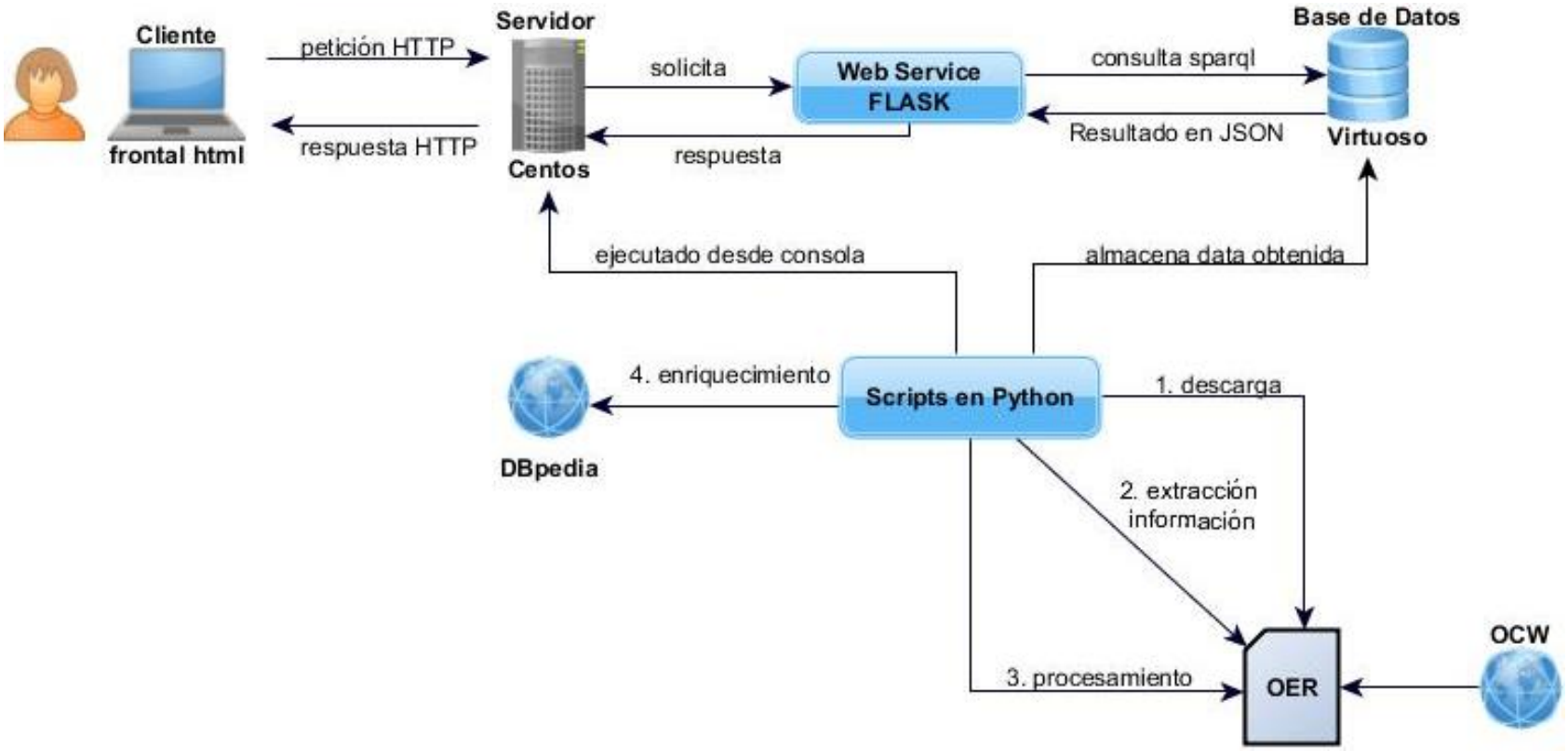


Figura 62: Arquitectura de la aplicación

2. Funcionalidad de la Aplicación

- **Extracción y Procesamiento de Información de OER's/OCW:** en este proceso se tomaron en consideración las siguientes funcionalidades.

  - ✓ DescargaClasifica (obtiene los OER's desde sitios OCW)
  - ✓ ObtenerNombre (adquiere el nombre del OER)
  - ✓ LimpiezaUrl (elimina los caracteres especiales de la URL)
  - ✓ ExtraerTextoPdfs (extrae texto de OER's en formato PDF)
  - ✓ DetectarIdioma (identifica el idioma del OER)
  - ✓ Tokeniza(divide en palabras el texto exraído)
  - ✓ GuardaTripletas (almacena en tripletas la data obtenida de cada tarea)
  - ✓ CrearConexion (crea la conexión entre Python y Virtuoso)
  - ✓ CrearGrafo (crea el grafo que contendrá las tripletas)
  - ✓ PdfsMetadatosPaginas (extrae el número de páginas del OER)
  - ✓ AgregaTag (agrega un tag a cada token)
  - ✓ TokensRepresentativos (identifica los tokens representativos del OER)
  - ✓ ObtenerEntidades (adquiere las entidades que contiene el OER)
  - ✓ ProcesoDesambiguación (desambigua aquellas palabras identificadas como ambiguas)
  - ✓ RelaciónOersGuardar (crea las relaciones utilizando las palabras comunes)
  - ✓ RelaciónEntidadesOersGuardar (crea las relaciones utilizando las entidades comunes)

- **Enriquecimiento de Información de OER's/OCW:** en este proceso se tomó en cuenta las siguientes funcionalidades.

- ✓ ConsultarTokenComun (envía una consulta Sparql al triplestore Virtuoso de los tokens comunes)
- ✓ ObtenerURLEnriquecer (adquiere la URL de DBpedia correspondiente al token comun)
- ✓ ConsultarEntidadComun (envía una consulta Sparql al triplestore Virtuoso de las entidades comunes)
- ✓ ObtenerURLEnriquecer (adquiere la URL de DBpedia correspondiente a la entidad comun)

- **Visualización de Información de OER's/OCW:** en este proceso se tomó en cuenta las siguientes funcionalidades.

  - ✓ VisualizarGrafo (solicita al web service el resultado en JSON)
  - ✓ Service (contiene parámetros para el funcionamiento de servicio web)
  - ✓ Visualizaroerjit (web service)
  - ✓ CrearJsonOers (construye el JSON en base a las relaciones entre OER's,)

3. Diagramas de Casos de Usos

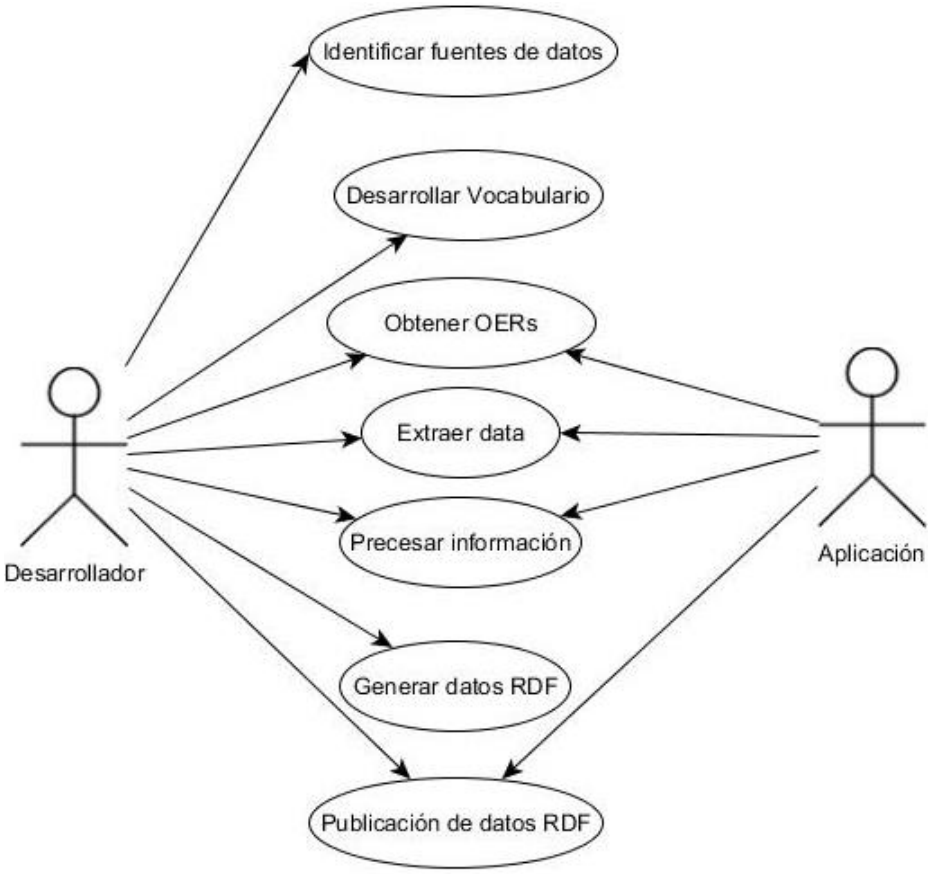


Figura 63: Diagrama de Casos de Uso del proceso de Extracción y Procesamiento de información

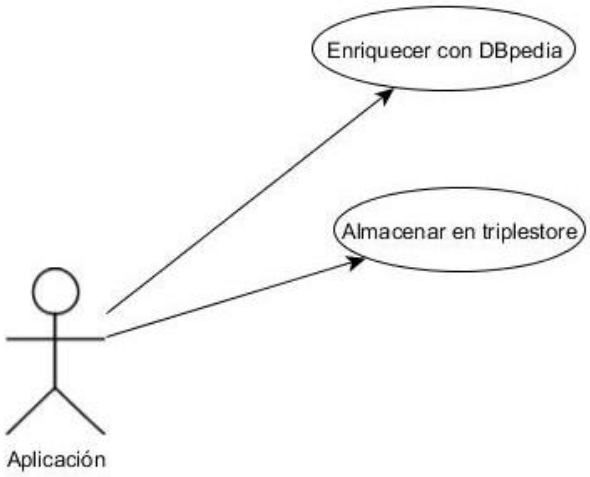


Figura 64: Diagrama de Casos de Uso del proceso de Enriquecimiento de Información

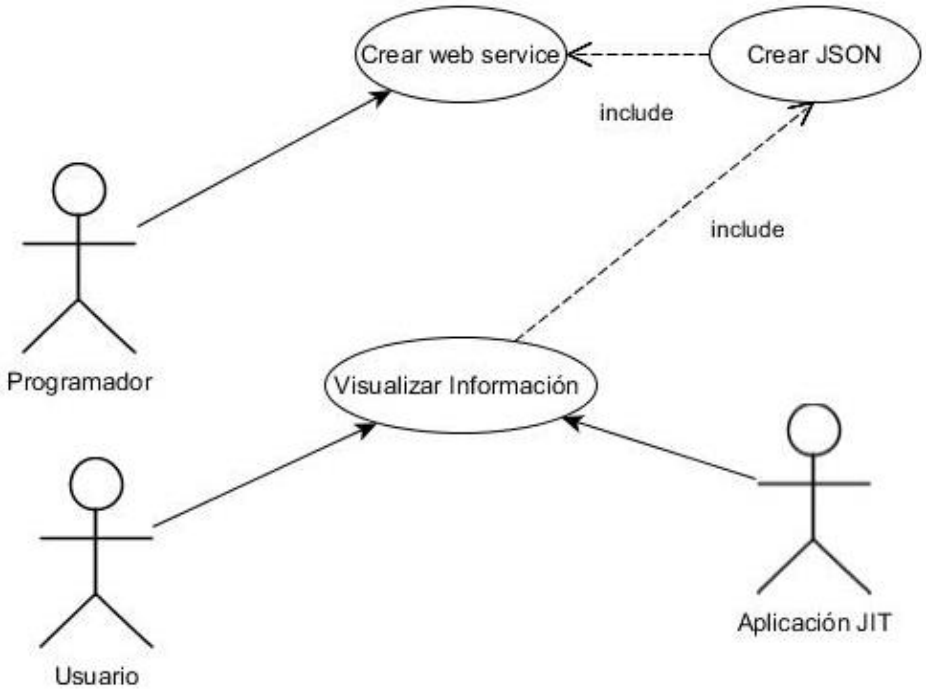


Figura 65: Diagrama de casos de uso del proceso de Visualización

4. Diagramas de Clases

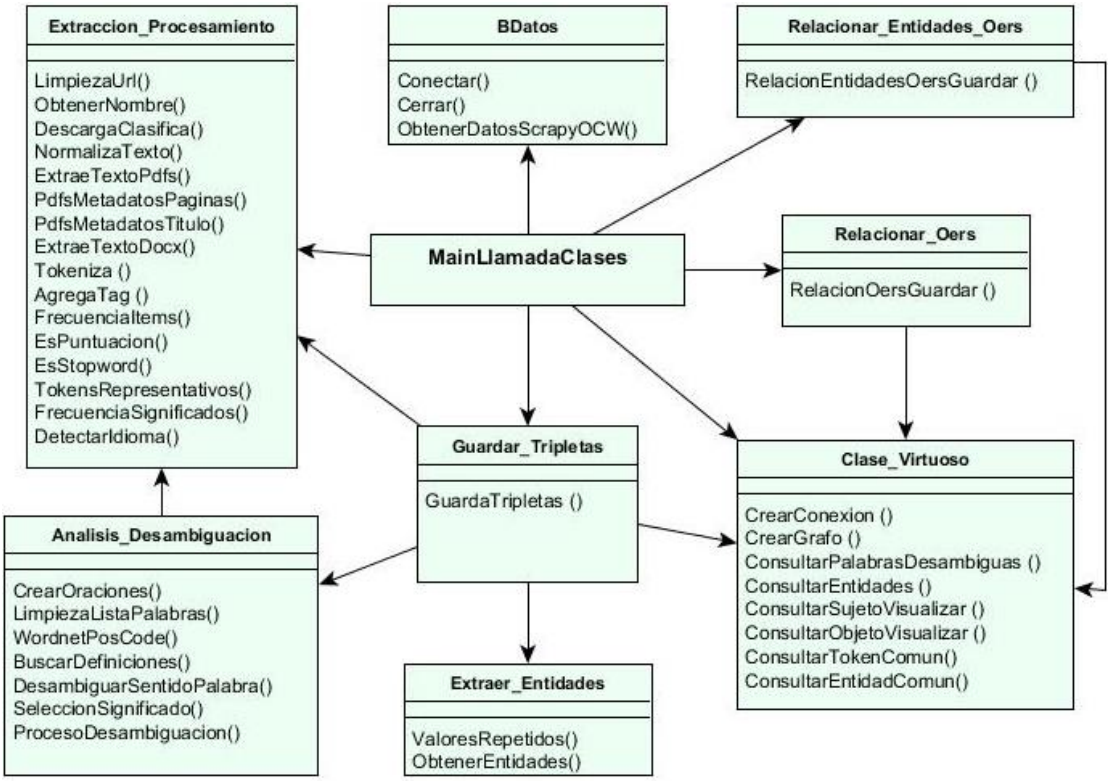


Figura 66: Diagrama de Clases del proceso de Extracción y Procesamiento de información

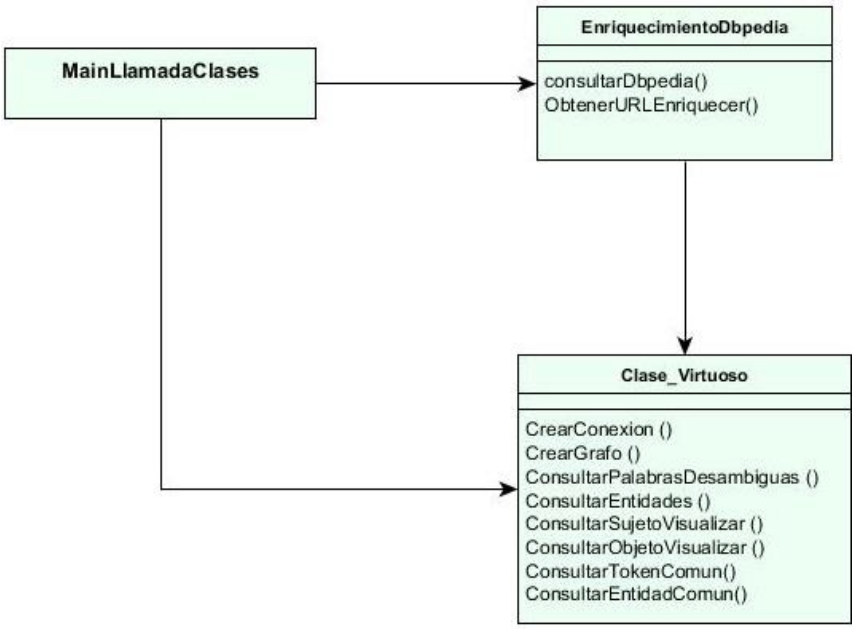


Figura 67: Diagrama de clases del proceso de Enriquecimiento de Información

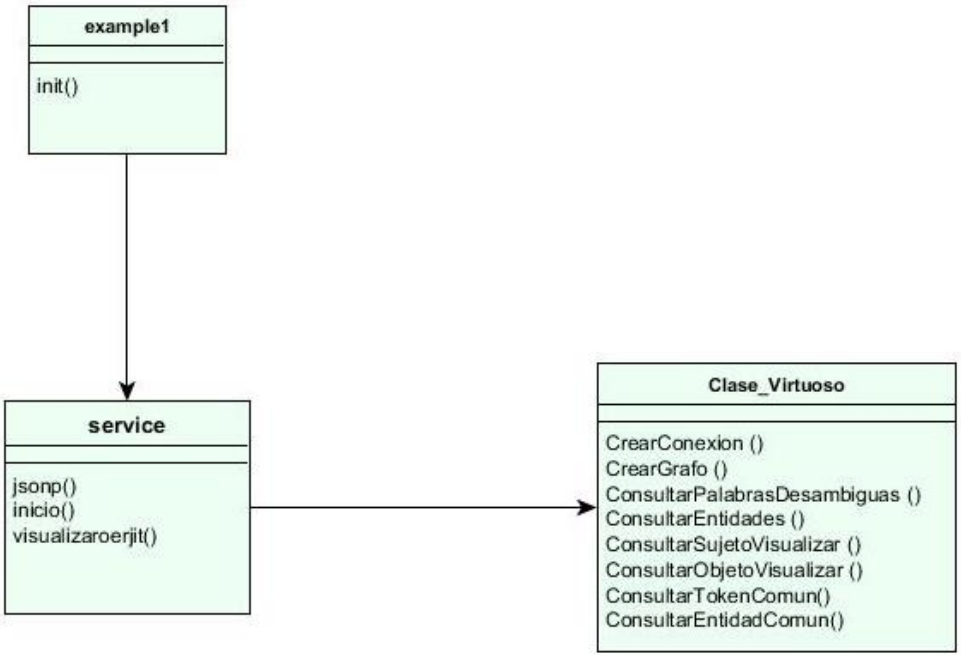


Figura 68: Diagrama de clases del proceso de Visualización de Información de OER's

5. MainLlamadaClases.py

```
# -*- coding: utf-8 -*-
# Duracion de ejecucion 10 minutos con 43 segundos - 28 de febrero 2014
"""
Created on Mon Oct  7 15:25:51 2013
@author: scecy
"""
from ClaseBaseDatos import *
from ClaseVirtuosoOersGuardar import *
from ClaseRelacionarPalabrasOers import *
from ClaseRelacionarEntidadesOers import *
from ClaseExtraccionProcesamientoOersV01 import *
from ClaseManipulacionVirtuoso import *
from ClaseEnriquecerDbpedia import *
##obtener datos de la base de datos
ObjBd = BDatos() #instancia objeto
ObjRO= Relacionar_Oers() #llama al metodo para relacionar las palabras del oer
ObjRE=Relacionar_Entidades_Oers() #llama al metodo para relacionar las entidades del oer
ObjGuaTri= Guardar_Tripletas()
ObjExtPro=Extraccion_Procesamiento()
ObjManipVirtu=Clase_Virtuoso()
ObjEnriqDbp=Clase_EnriquecerDbpedia()
#asignar nombre de las tablas de la bd
tabla = "scrapy_consortium.CursosConsortium" #tabla que contiene los Oers de los OCW
#crear cursores de la bd
cursorBd=ObjBd.ObtenerDatosScrapyOCW(tabla) #llama al metodo de obtencion de datos de la tabla en
la que se encuentra los OER's extraidos del scrapy
try:
    #Descargar OER's
    #ObjExtPro.DescargaClasifica()
    cont=0
    #Extraccion y Procesamiento OER's
    for tupla in cursorBd:
        cont=cont+1
        nombrearch=ObjExtPro.ObtenerNombre(tupla)
        nombredd=str(cont)+"_"+nombrearch
        print nombredd
        textoer=ObjExtPro.ExtraeTextoPdfs(nombredd)
        idioma=ObjExtPro.DetectarIdioma(textoer)
        if idioma=="english":
            ObjGuaTri.GuardaTripletas(tupla,nombredd)
    #Relacionar OER's
    ObjRO.RelacionOersGuardar()
    ObjRE.RelacionEntidadesOersGuardar()
    #Enriquecer tokens y entidades comunes con Dbpedia
    resulttokens=ObjManipVirtu.ConsultarTokenComun()
    uripalabra="http://dataoers.org/oer/commonword/"
    ObjEnriqDbp.ObtenerURLEnriquecer(resulttokens,uripalabra)
    resultentidades=ObjManipVirtu.ConsultarEntidadComun()
    urientidad="http://dataoers.org/oer/commonentity/"
    ObjEnriqDbp.ObtenerURLEnriquecer(resultentidades,urientidad)
except Exception, e:
   print e
```

6. ClaseBaseDatos.py

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 31 17:17:47 2013
@author: scecy
"""
import codecs
import MySQLdb
import MySQLdb as mdb
class BDatos():
    """        Clase para la Base de Datos
    """
    def Conectar(self):
        """            conectarme a la bd
        """
        db = None
        try:
            db =
MySQLdb.connect(host="localhost",user="root",passwd="admin",db="enlaces_oers")#,charset="utf8",use_
unicode="True"
        except Exception, e:
            print "error de coneccion", e
        return db
    def Cerrar(self, db):
        """            cerrar la base de datos
        """
        db.close()
    def ObtenerDatos(self, tabla):
        """            conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT * FROM %s;"%tabla # sql obtener datos
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
    def ObtenerDatosScrapyOCW(self, tabla):
        """            conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        lik="'%.pdf%'"
        sql="SELECT DISTINCT * FROM "+tabla+" WHERE predicado='oer' AND objeto like "+lik+" LIMIT
26000;"#%tabla
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos

    def ObtenerDatosRelacionar(self, tabla):
        """        conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT sujeto, objeto FROM %s WHERE predicado='palabrasdesambiguas' LIMIT
2000;"%tabla # sql para obtener las palabras ambiguas
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
```

```
    def ObtenerDatosEntidades(self, tabla):
        """             conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT sujeto, objeto FROM %s WHERE predicado='entidades' LIMIT
2000;"%tabla # sql para obtener las palabras ambiguas
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
    def ObtenerPropiedadesOers(self):
        """             conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT * FROM AnalisisVisualizacionOers.OersExtraccionProcesamiento WHERE
(predicado='dc:title' or predicado='cantidadpaginas' or predicado='idioma')"
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
    def ObtenerToknsRepresentativos(self, tabla):
        """             conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT sujeto, objeto FROM %s WHERE predicado='tokensrepresentativos' LIMIT
2000;"%tabla # sql para obtener las palabras ambiguas
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
    def InsertarTripletas(self, s, p, o, tabla):
        db=self.Conectar()
        cur=db.cursor()
        cur.execute(u"""'INSERT INTO '"""+ tabla +""" (sujeto, predicado, objeto) VALUES (%s, %s, %s)""" ,
(s, p, o))
        print "Datos guardados satisfactoriamente..."
        db.commit()
        db.close()


    def InsertarRelaciones(self, s, p, o, tabla):
        db=self.Conectar()
        cur=db.cursor()
        cur.execute(u"""'INSERT INTO '"""+ tabla +""" (sujeto, predicado, objeto) VALUES (%s, %s, %s)""" ,
(s, p, o))
        print "Relaciones guardadas satisfactoriamente..."
        db.commit()
        db.close()
    def ConsultarSujetoVisualizar(self):
        """             conectarme a la bd, para sacar los datos necesarios
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT DISTINCT sujeto FROM AnalisisVisualizacionOers.RelacionesEntreOers limit 300;" #
sql obtener datos
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
    def ConsultarObjetoVisualizar(self, suj):
        """             conectarme a la bd, para sacar los datos necesarios
```

```
        """
        db=self.Conectar()
        cursor=db.cursor()
        sql = "SELECT objeto FROM AnalisisVisualizacionOers.RelacionesEntreOers where sujeto='"+suj+"'
and predicado='relatedto' limit 300;" # sql obtener datos
        cursor.execute(sql)
        datos = cursor.fetchall()
        db.close()
        return datos
```

7. ClaseExtraccionProcesamientoOersV01.py

```
# -*- coding: utf-8 -*-
"""
Created on Fri May 31 17:52:59 2013
@author: scecy
"""
#librerias necesarias
from pyPdf import PdfFileReader
import unicodedata
import urllib2
import nltk
from nltk import *
#import os.path
from ClaseBaseDatos import *
from ClaseManipulacionVirtuoso import *
class Extraccion_Procesamiento():
    """Metodos para extraccion y manipulacion de texto"""
    def LimpiezaUrl(self,tup):
        """Toma la tupla que contiene sujeto, predicado, objeto
           y Obtiene la url que corresponde al objeto"""
        try:
            stupla=str(tup) #convertimos en str la tupla contiene sujeto, predicacdo y objeto
            url=stupla.split(',')[2] #extraemos el objeto-url
            longitud = len(str(url))
            url_limpia = str(url)[2:longitud-2] #quitamos ',) para obtener la url limpia del objeto
            return (url_limpia) #devuelve una url limpia y lista para poder utilizarla en un browser
        except  Exception, e:
            print e
    def ObtenerNombre(self,tupla):
        """Obtiene el nombre del oer contenido en el objeto """
        url = self.LimpiezaUrl(tupla)
        nombre_oer = url.split('/')[-1] #extraemos el nombre del archivo de la url
        if nombre_oer.split('.')[-1] == "pdf":
            return (nombre_oer)#devuelve el nombre del archivo
        else:
            nombre_oer_limpio =  nombre_oer[0:-11]
            return(nombre_oer_limpio)#devuelve el nombre del archivo
    def DescargaClasifica(self):
        """Realiza la descarga de los OERs de sitios OCW y los clasifica dependiendo del tipo de archivo"""
        try:
            cont=1
            ObjBd = BDatos()
            tabla = "scrapy_consortium.CursosConsortium" #tabla que contiene los Oers de los OCW
            cursorBd=ObjBd.ObtenerDatosScrapyOCW(tabla) #consulta a la bd los oers del scrapeo
            for tupla in cursorBd:
                print "--archivo: ",cont
                url = self.LimpiezaUrl(tupla) #llama al metodo de limpieza de url
                nombre_archivo= self.ObtenerNombre(tupla) #obtiene el nombre del archivo mediante el
metodo
                try:
                    peticion = urllib2.urlopen(url)
                #creamos y escribimos el archivo y con with nos ayuda a cerrar el archivo automaticamente
                    print "** Descargando archivo: " + nombre_archivo
                    nombre_archivo_dd=str(cont)+"_"+nombre_archivo
                    with open("/home/scecy/Oers/Pdfs/"+nombre_archivo_dd, "wb") as code:
                        code.write(peticion.read())
                except urllib2.HTTPError, e:
                    print 'Could not download page, %s.'%e.code
                    if e.code==404:
                        pass
                cont=cont+1
```

```
        except Exception, e:
            print e
    def NormalizaTexto(self,texto):
        "Normaliza texto, limpia tildes, caracteres especiales y lo convierte a minusculas"
        try:
            texto_cod= unicodedata.normalize('NFKD', texto).encode('ascii','ignore') #normaliza la
codificaciÃ³n a unicode
            texto_listo=texto_cod.lower()
            return (texto_listo)
        except  Exception, e:
            print e
    def ExtraeTextoPdfs(self,nomb_archivo):
        """EXTRAE PORCION DE TEXTO DE UNA PAGINA DEL ARCHIVO PDF, """
        try:
            archivo_pdf = PdfFileReader(file("/home/scecy/Oers/Pdfs/"+nomb_archivo, "rb"))    #Capturar el
archivo pdf a leer
            cant_paginas = archivo_pdf.getNumPages()
            i=1
            textoextraido =''
            if cant_paginas>150 and cant_paginas>200 :
                for i in range (47):
                    textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                return(textoextraido) #devuelve el texto limpio y listo para utilizar
            else:
                if cant_paginas>100 and cant_paginas <=150:
                    for i in range (34):
                        textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                    return(textoextraido) #devuelve el texto limpio y listo para utilizar
                else:
                    if cant_paginas>70 and cant_paginas <=100:
                        for i in range (24):
                            textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                        return(textoextraido) #devuelve el texto limpio y listo para utilizar
                    else:
                        if cant_paginas>40 and cant_paginas <=70:
                            for i in range (17):
                                textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                            return(textoextraido) #devuelve el texto limpio y listo para utilizar
                        else:
                            if cant_paginas>20 and cant_paginas <=40:
                                for i in range (10):
                                    textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                                return(textoextraido) #devuelve el texto limpio y listo para utilizar
                            else:
                                if cant_paginas>10 and cant_paginas <=20:
                                    for i in range (6):
                                        textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                                    return(textoextraido) #devuelve el texto limpio y listo para utilizar
                                else:
                                    if cant_paginas>7 and cant_paginas <=10:
                                        for i in range (5):
                                            textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                                        return(textoextraido) #devuelve el texto limpio y listo para utilizar
                                    else:
                                        if cant_paginas>4 and cant_paginas <=7:
                                            for i in range (4):
                                                textoextraido += archivo_pdf.getPage(i).extractText()+ ""
                                            return(textoextraido) #devuelve el texto limpio y listo para utilizar
                                        else:
                                            if cant_paginas>=1 and cant_paginas <=4:
                                                textoextraido +=
archivo_pdf.getPage(0).extractText()+""+archivo_pdf.getPage(1).extractText()
                                            return(textoextraido) #devuelve el texto limpio y listo para utilizar
        except  Exception, e:
```

```
            print e
    def PdfsMetadatosPaginas(self,na):
        """Extrae metadatos de archivos pdfs"""
        try:
            archivo_pdf = PdfFileReader(file("/home/scecy/Oers/Pdfs/"+na, "rb"))
            numpag = archivo_pdf.getNumPages()      #Capturar la cantidad de paginas que tiene el
documento
            return (numpag)
        except  Exception, e:
            print e
    def ExtraeTextoDocx(nombrearchivo):
        """Extrae tecto de archivos docx"""
        try:
            docx = zipfile.ZipFile('/home/scecy/Oers/Docs/'+nombrearchivo) #lee el archivo docx
            content = docx.read('word/document.xml')#lee el contenido del documento
            cleaned = re.sub('<(.|\n)*?>','',content) #limpia el texto extraido
            texto_docx_listo=self.NormalizaTexto(cleaned) #normaliza texto
            return(texto_docx_listo)
        except  Exception, e:
            print e
    def Tokeniza (self,txt):
        """Tokeniza el texto extraido de un archivo"""
        tokens_cadena = TreebankWordTokenizer().tokenize(txt) #tokeniza el texto
        return (tokens_cadena)
    def AgregaTag (self,tokens):
        """Agrega un tag a la teokenizacion, y guarda el archivo con extension .txt"""
        tags = nltk.pos_tag(tokens) #agrega un tag a los tokens
        return(tags)
    def FrecuenciaItems(self,tokens):
        "Devuelve la frecuencia de los tokens"
        try:
            fdist1 = FreqDist(tokens)
            frec_items= fdist1.items()
            return (frec_items)
        except  Exception, e:
            print e

    def EsPuntuacion(self,string): #identifica si es un signo, o digito de puntuacion
        for char in string:
            if char.isalpha() or char.isdigit():  #or char.printable():or char.isprintable():
                return False
        return True
    def EsStopword(self,string): #identifica si es un stopwords
        if string.lower() in nltk.corpus.stopwords.words('english'):
            return True
        else:
            return False
    def TokensRepresentativos(self,tokens):
        "Devuelve los tokens mas representativos "
        toknsrep_limpios=[]
        try:
            fdist2 = FreqDist(tokens)
            tokns_significativos= sorted ([w for w in set(tokens) if fdist2[w]>=3]) #muestra los tokens que dan
un significado del texto en una lista
            for token in tokns_significativos:
                if self.EsPuntuacion(token): #identifica si es un token de puntuacion
                    pass
                elif self.EsStopword(token): #si encuentra un stopword no lo toma en cuenta
                    pass
                elif token not in toknsrep_limpios: #verifica el valor de retorno, para no guardar repeticiones de
palabras
                    toknsrep_limpios.append(token)
                else:
                    pass
```

```
            return (toknsrep_limpios)
        except  Exception, e:
            print e
    def FrecuenciaSignificados(self,significados):
        try:
            fdist=FreqDist(significados)
            frec_significados= sorted ([w for w in set() if fdist[w]>3])
            return (frec_significados)
        except  Exception, e:
            print e
    def DetectarIdioma(self, txt):
        "Detecta el Idioma del texto, enviado en tokens"
        # Lista de idiomas disponibles en la nltk
        try:
            txt_tokens= self.Tokeniza(txt)
            languages =
["spanish","english","dutch","finnish","german","italian","portuguese","turkish","danish","french","hungarian",
"norwegian","russian","swedish"]
            # Creamos un dict donde almacenaremos la cuenta de las stopwords para cada idioma
            lang_count = {}
            # Por cada idioma
            for lang in languages: # Obtenemos las stopwords del idioma del mÃ³dulo nltk
                stop_words = unicode(nltk.corpus.stopwords.words(lang))
                lang_count[lang] = 0 # Inicializa a 0 el contador para cada idioma
                for word in txt_tokens: # Recorremos las palabras del texto a analizar
                    if word in stop_words: # Si la palabra se encuentra entre las stopwords, incrementa el
contador
                        lang_count[lang] += 1
            detected_language = max(lang_count, key=lang_count.get) # Obtiene el idioma con el nÃºmero
mayor de coincidencias
            return (detected_language)
        except  Exception, e:
            print e
```

8. ClaseExtraerEntidadesV01.py

```
# -*- coding: utf-8 -*-

import re
import unicodedata
import codecs
from nltk import *
from textblob import TextBlob, Word
from dateutil.parser import parse
from ClaseExtraccionProcesamientoOersV01 import *
ObjExPr = Extraccion_Procesamiento() #instancia objeto metodos
class Extraer_Entidades():
    """Extraer Entidades"""
    def ValoresRepetidos(self,lista, valor):
        for orp in xrange(0, len(lista)):
            if valor == lista[orp]:
                valor = ""
        return valor
    def ObtenerEntidades(self,texto):
        try:
            entidades=[]
            entities = []
            person=[]
            organization=[]
            gpe=[]
            otherents=[]
            indexent=0
            indexorg=0
            indexper=0
            indexgpe=0
            indexoth=0
            while len(entities)>indexent:
                del entities[indexent]
            while len(organization)>indexorg:
                del organization[indexorg]
            while len(person)>indexper:
                del person[indexper]
            while len(gpe)>indexgpe:
                del gpe[indexgpe]
            while len(otherents)>indexoth:
                del otherents[indexoth]
            for sentence in sent_tokenize(texto):
              chunks = ne_chunk(pos_tag(word_tokenize(sentence)))
              entities.extend([chunk for chunk in chunks if hasattr(chunk, 'node')])
            for jp in entities:
              cadentidad = ""
              for c in jp.leaves():
                        cadentidad =  cadentidad + c[0] + " "
              cadentidad = cadentidad.strip()
              if ObjExPr.EsStopword(cadentidad):
                        cadentidad=""
              if cadentidad !="":
                    if ObjExPr.EsPuntuacion(cadentidad):
                        pass
                    elif jp.node == 'ORGANIZATION':
                        cadentidad = self.ValoresRepetidos(organization, cadentidad)
                        if cadentidad != "":
                            organization.append(cadentidad)
                    elif jp.node== 'PERSON':
                        cadentidad = self.ValoresRepetidos(person, cadentidad)
                        if cadentidad != "":
```

```
                                    person.append(cadentidad)
            elif jp.node== 'GPE':
                cadentidad = self.ValoresRepetidos(gpe, cadentidad)
                if cadentidad != "":
                                    gpe.append(cadentidad)
            else:
                cadentidad = self.ValoresRepetidos(otherents, cadentidad)
                if cadentidad != "":
                    otherents.append(cadentidad)
        index=0
        while len(entidades)>index:
            del entidades[index]
        entidades=organization+person+gpe+otherents
        return (entidades)
    except Exception, e:
        print e
```

9. ClaseAnalisisDesambiguacionV1.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov  7 22:40:59 2013
@author: scecy
"""
import nltk
from ClaseExtraccionProcesamientoOersV01 import *
from nltk.corpus import wordnet
lista_palabras=[]
lista_sigwn=[]
valor_retorno=[]
token={}
tokens_representativos=''
cont=0
class Analisis_Desambiguacion():
    """Metodos para el analisis y desambiguacion del texto"""
    def CrearOraciones(self,texto): #crea oracione
        texto_oraciones = nltk.sent_tokenize(texto)
        return (texto_oraciones)
    def LimpiezaListaPalabras(self,lista):
        try:
            for index in lista:
                lista.remove[index]
                print "limpieza lista ",lista
        except Exception, e:
            print e
    def WordnetPosCode(self,tag): #agrega un tag a cada palabra usando wordnet
        try:
            if tag.startswith('NN'):
                return wordnet.NOUN
            elif tag.startswith('VB'):
                return wordnet.VERB
            elif tag.startswith('JJ'):
                return wordnet.ADJ
            elif tag.startswith('RB'):
                return wordnet.ADV
            else:
                return ''
        except Exception, e:
            print e
    def BuscarDefiniciones(self,texto,tokens_representativos):
        try:
            ObjMetTes = Extraccion_Procesamiento()
            for parrafo in texto:  # utiliza cada parrafo de un texto
                oraciones_parrafo = self.CrearOraciones(parrafo) #tokeniza el parrafo en oraciones
                for oracion in oraciones_parrafo: #recorre las oraciones de un parrafo
                    sentence = []
                    tokens = ObjMetTes.Tokeniza(oracion) #tokeniza la oracion llamando al metodo
                    tag_tuples = ObjMetTes.AgregaTag(tokens) #agrega tag a cada token de la oracion,
llamando a metodo
                    for (string, tag) in tag_tuples:
                        token = string,tag #utiliza un arreglo con el token, tag: facilita el proceso de
desambiguacion
                        sentence.append(token)
                self.SeleccionSignificado (sentence, oracion, token,tokens_representativos) #llama al metodo
envia los parametros correspondientes, de cada oracion en un parrafo
        except Exception, e:
            print e
    def DesambiguarSentidoPalabra(self,word, wn_pos, sentence, token): #desambigua el siginificado de la
palabra segun el contexto
```

```
        try:
            senses = wordnet.synsets(word, wn_pos) #toma todos los significados de la palabra
            #toma el numero que cumplen con esta condicion: recorre todos los siginificados, y de cada
significado toma las palabras de dicho significado, para comparar si esa palabra del significado se
encuentra en la oracion de analisis.
            cfd = nltk.ConditionalFreqDist((sense, def_word) for sense in senses for def_word in
sense.definition.split() if def_word in sentence)
            best_sense = senses[0] # start with first sense
            for sense in senses:
                if cfd[sense].N > cfd[best_sense].N: #toma el mejor significado, verificando el significado que
tenga mayor frecuencia dentro de una oracion
                    best_sense = sense
                    token=word,wn_pos,str(best_sense.definition),str(best_sense)
            return token
        except Exception, e:
            print e
    def SeleccionSignificado(self,sentence,oracion,token,tokens_representativos):
        ObjMetTes = Extraccion_Procesamiento()  #instancia de objeto
        try:
            valor_retorno=[]
            for token in sentence: #toma cada token de una oracion
                word = token[0]
                wn_pos = self.WordnetPosCode(token[1]) #agrega el tag a cada token
                if ObjMetTes.EsPuntuacion(word): #identifica si es un token de puntuacion
                    pass
                elif ObjMetTes.EsStopword(word): #si encuentra un stopword no lo toma en cuenta
                    pass
                elif len(wordnet.synsets(word, wn_pos)) > 0: #verifica que la palabra tenga almenos un
significado
                    if word in tokens_representativos: #verifica si la palabra esta dentro de los tokens
representativos
                        valor_retorno=self.DesambiguarSentidoPalabra(word,wn_pos, oracion, token) #llama al
metodo de desambiguacion envia: palabra,tag,oracion
                        if len(valor_retorno)>3: #verifica que el valor de retorno del metodo de desambiguacion
tenga 3 elementos (palabra,tag,definicion)#
                            if valor_retorno not in lista_palabras: #verifica el valor de retorno, para no guardar
repeticiones de palabras
                                lista_palabras.append(valor_retorno) #palabras desambiguas
                else:
                    pass
        except Exception, e:
            print e
    def ProcesoDesambiguacion(self,texto): #proceso principal de desambiguacion
        ObjMetTes = Extraccion_Procesamiento()  #instancia de objeto
        try:
            index=0
            while len(lista_palabras)>index:
                del lista_palabras[index]
            tokens_texto = ObjMetTes.Tokeniza(texto)
            texto_parrafos=self.CrearOraciones(texto)
            tokens_representativos=ObjMetTes.TokensRepresentativos(tokens_texto)
            self.BuscarDefiniciones(texto_parrafos,tokens_representativos)
            return lista_palabras #lista de palabras desambiguas
        except Exception, e:
            print e
```

10. ClaseRelacionarEntidadesOers.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 19 08:16:21 2013

@author: scecy
"""

import pickle
from ClaseManipulacionVirtuoso import *
from ClaseExtraccionProcesamientoOersV01 import *
from ClaseVirtuosoOersGuardar import *

contE=0
cantidad_entidades=0
mitadcpE=0
cuartacpE=0
entidades_comunes=[]

class Relacionar_Entidades_Oers():
    """Crea la relaciones entre los Oers"""

    def VerificarDatosEnVirtuoso(self, s, p, o):
        try:
            ObjManipVirt = Clase_Virtuoso()
            store = ObjManipVirt.CrearConexion()
            graph = ObjManipVirt.CrearGrafo(store)
            resultvalidar=ObjManipVirt.consultaValidacionRelacion(s, p, o)
            if len(resultvalidar["results"]["bindings"])==0:
                #Agrega datos nuevos
                if o.startswith("http:")==True: #verifica si es un uri o un literal
                    graph.add([URIRef(s),URIRef(p),URIRef(o)])
                    print "El objeto es un URI, datos de relaciÃ³n entre palabras guardados en el grafo
http://dataoers.org/"
                else:
                    graph.add([URIRef(s),URIRef(p),Literal(o)])
                    print "El objeto es un Literal, datos de relaciÃ³n entre palabras guardados en el grafo
http://dataoers.org/"
            else:
                #Actualiza datos existentes
                print "Â¡Los datos de relaciÃ³n entre palabras ya se encuentran almacenados en el grafo
http://dataoers.org/!"
            store.commit()
        except Exception, e:
            print e


    def RelacionEntidadesOersGuardar (self):
        ObjVirtGuardar=Guardar_Tripletas()
        ObjVirt= Clase_Virtuoso()
        listaE=ObjVirt.ConsultarEntidades()
        try:
            for tuplaE in listaE: #recorre cada tupla del cursor que contiene todos los elemetos de la bd...
                valorcomparar=0
                valoresarreglar=[]
                cE=0
                indexec=0
                lista_entidadesA1=tuplaE[1]
                plista_entidadesA1=pickle.loads(lista_entidadesA1) #carga el tipo de dato original del elemento,
en este caso lista
                suj= tuplaE[0]
```

```
            while len(listaE)>cE:
                while len(entidades_comunes)>indexec:
                    del entidades_comunes[indexec]

                lista_entidadesA2_sig=listaE[cE] #toma la lista de palabras siguientes de la lista de palabras
eje
                plista_entidadesA2_sig=pickle.loads(lista_entidadesA2_sig[1]) #carga los datos originales de
la bd
                cnprE=0 #contador del numero de palabras que se relacionan con el oer a evaluar
                for lentidadesA1 in plista_entidadesA1: #toma cada palabra de la lista de palabras que se
estan evaluando
                    if lentidadesA1 in plista_entidadesA2_sig: #verifica si la palabra a evaluar se encuentra en
la lista de palabras siguientes
                        cnprE+=1 #contador del numero de palabras que se relacionan con el oer a evaluar
                        entidades_comunes.append(lentidadesA1) #almacena las palabras comunes
                        valores=[len(entidades_comunes), lista_entidadesA2_sig[0]] #guarda la cantidad de
palabras comunes y la url relacionada

                cantidad_entidades=len(plista_entidadesA1) #numero de palabras del objeto
                mitadcpE=cantidad_entidades/2
                cuartacpE=mitadcpE/2
                if lista_entidadesA2_sig[0]!=tuplaE[0]:
                    if cantidad_entidades<=cnprE or cnprE>=cuartacpE:
                        cantvalores=valores[0]
                        if cantvalores>valorcomparar:
                            valorcomparar=cantvalores
                            valoresarreglar+=valores

self.VerificarDatosEnVirtuoso(suj,"http://purl.org/dc/elements/1.1/relation",lista_entidadesA2_sig[0])
#                           graph.add([URIRef(suj),DC["relation"],Literal(lista_entidadesA2_sig[0], lang="en")])
#                           print "Relacion entre OER's almacenada exitosamente.."

                        for entidad in entidades_comunes:
                            urientidad="http://dataoers.org/oer/commonentity/"+entidad.replace(" ","_")

self.VerificarDatosEnVirtuoso(suj,"http://www.w3.org/2002/07/owl#NamedIndividual",urientidad)
#
graph.add([URIRef(suj),URIRef("http://www.w3.org/2002/07/owl#NamedIndividual"),Literal(urientidad,
lang="en")]) #relacion tokenxtoken
#                               print "Entidad comun almacenada exitosamente.."
#                               store.commit()
                cE=cE+1
        except Exception, e:
            print e
```

11. ClaseRelacionarPalabrasOers.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 19 08:16:21 2013

@author: scecy
"""
import pickle
from ClaseManipulacionVirtuoso import *
from ClaseExtraccionProcesamientoOersV01 import *
from ClaseVirtuosoOersGuardar import *

contE=0
cantidad_entidades=0
mitadcpE=0
cuartacpE=0
entidades_comunes=[]

class Relacionar_Entidades_Oers():
    """Crea la relaciones entre los Oers"""

    def VerificarDatosEnVirtuoso(self, s, p, o):
        try:
            ObjManipVirt = Clase_Virtuoso()
            store = ObjManipVirt.CrearConexion()
            graph = ObjManipVirt.CrearGrafo(store)
            resultvalidar=ObjManipVirt.consultaValidacionRelacion(s, p, o)
            if len(resultvalidar["results"]["bindings"])==0:
                #Agrega datos nuevos
               if o.startswith("http:")==True: #verifica si es un uri o un literal
                    graph.add([URIRef(s),URIRef(p),URIRef(o)])
                    print "El objeto es un URI, datos de relaciÃ³n entre palabras guardados en el grafo
http://dataoers.org/"
                else:
                    graph.add([URIRef(s),URIRef(p),Literal(o)])
                    print "El objeto es un Literal, datos de relaciÃ³n entre palabras guardados en el grafo
http://dataoers.org/"
            else:
                #Actualiza datos existentes
                print "Â¡Los datos de relaciÃ³n entre palabras ya se encuentran almacenados en el grafo
http://dataoers.org/!"
            store.commit()
        except Exception, e:
            print e
    def RelacionEntidadesOersGuardar (self):
        ObjVirtGuardar=Guardar_Tripletas()
        ObjVirt= Clase_Virtuoso()
#         store=ObjVirt.CrearConexion()
#         graph=ObjVirt.CrearGrafo(store)
        listaE=ObjVirt.ConsultarEntidades()
        try:
            for tuplaE in listaE: #recorre cada tupla del cursor que contiene todos los elemetos de la bd...
                valorcomparar=0
                valoresarreglar=[]
                cE=0
                indexec=0
                lista_entidadesA1=tuplaE[1]
                plista_entidadesA1=pickle.loads(lista_entidadesA1) #carga el tipo de dato original del elemento,
en este caso lista
                suj= tuplaE[0]
```

```
            while len(listaE)>cE:
                while len(entidades_comunes)>indexec:
                    del entidades_comunes[indexec]
                lista_entidadesA2_sig=listaE[cE] #toma la lista de palabras siguientes de la lista de palabras
eje
                plista_entidadesA2_sig=pickle.loads(lista_entidadesA2_sig[1]) #carga los datos originales de
la bd
                cnprE=0 #contador del numero de palabras que se relacionan con el oer a evaluar
                for lentidadesA1 in plista_entidadesA1: #toma cada palabra de la lista de palabras que se
estan evaluando
                    if lentidadesA1 in plista_entidadesA2_sig: #verifica si la palabra a evaluar se encuentra en
la lista de palabras siguientes
                        cnprE+=1 #contador del numero de palabras que se relacionan con el oer a evaluar
                        entidades_comunes.append(lentidadesA1) #almacena las palabras comunes
                        valores=[len(entidades_comunes), lista_entidadesA2_sig[0]] #guarda la cantidad de
palabras comunes y la url relacionada
                cantidad_entidades=len(plista_entidadesA1) #numero de palabras del objeto
                mitadcpE=cantidad_entidades/2
                cuartacpE=mitadcpE/2
                if lista_entidadesA2_sig[0]!=tuplaE[0]:
                    if cantidad_entidades<=cnprE or cnprE>=cuartacpE:
                        cantvalores=valores[0]
                        if cantvalores>valorcomparar:
                            valorcomparar=cantvalores
                            valoresarreglar+=valores

self.VerificarDatosEnVirtuoso(suj,"http://purl.org/dc/elements/1.1/relation",lista_entidadesA2_sig[0])
#                           graph.add([URIRef(suj),DC["relation"],Literal(lista_entidadesA2_sig[0], lang="en")])
#                           print "Relacion entre OER's almacenada exitosamente.."

                        for entidad in entidades_comunes:
                            urientidad="http://dataoers.org/oer/commonentity/"+entidad.replace(" ","_")

self.VerificarDatosEnVirtuoso(suj,"http://www.w3.org/2002/07/owl#NamedIndividual",urientidad)
#
graph.add([URIRef(suj),URIRef("http://www.w3.org/2002/07/owl#NamedIndividual"),Literal(urientidad,
lang="en")]) #relacion tokenxtoken
#                               print "Entidad comun almacenada exitosamente.."
#                               store.commit()
                cE=cE+1
        except Exception, e:
            print e
```

12. ClaseVirtuosoOERsGuardar.py

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 26 15:59:22 2014

@author: scecy
"""
#import pickle #ayuda a encapsular tipos de datos que no son validos para una bd, como una lista

from ClaseAnalisisDesambiguacionV1 import *
from ClaseExtraccionProcesamientoOersV01 import *
from ClaseBaseDatos import *
from ClaseExtraerEntidadesV01 import *
from ClaseManipulacionVirtuoso import *
import pickle
from rdflib.graph import Graph
from rdflib.store import Store
from rdflib.plugin import get as plugin
from rdflib.namespace import DC, XSD
from rdflib.term import URIRef, Literal
from virtuoso.vstore import Virtuoso
class Guardar_Tripletas():
    """Guarda los datos en tripletas del proceso de extraccion y del proceso de análisis"""
    def VerificarDatosEnVirtuoso(self, s, p, o):
        try:
            ObjManipVirt = Clase_Virtuoso()
            store = ObjManipVirt.CrearConexion()
            graph = ObjManipVirt.CrearGrafo(store)

#           print tipo
            resultvalidar=ObjManipVirt.consultaValidacion(s, p, o)
            if len(resultvalidar["results"]["bindings"])==0:
                #Agrega datos nuevos
                if p=="http://www.aktors.org/ontology/portal#has-page-numbers":
                    graph.add([URIRef(s),URIRef(p),Literal(o)])
                    print "El objeto es un Literal int, datos guardados en el grafo http://dataoers.org/"
                else:
                    if o.startswith("http:")==True: #verifica si es un uri o un literal
                        graph.add([URIRef(s),URIRef(p),URIRef(o)])
                        print "El objeto es un URI, datos guardados en el grafo http://dataoers.org/"
                    else:
                        graph.add([URIRef(s),URIRef(p),Literal(o)])
                        print "El objeto es un Literal, datos guardados en el grafo http://dataoers.org/"
            else:
                #Actualiza datos existentes
                print "¡Los datos ya se encuentran almacenados en el grafo http://dataoers.org/!"
            store.commit()
        except Exception, e:
            print e
    def GuardaTripletas (self,tupla, nombrearchdd):
        ObjManipVirt = Clase_Virtuoso()
        ObjManipOers= Extraccion_Procesamiento()
        ObjAnDesam = Analisis_Desambiguacion()
        ObjEnti=Extraer_Entidades()
#       ObjManipVirt = Clase_Virtuoso()
        try:
            nombre_oer_ocw=ObjManipOers.ObtenerNombre(tupla) #obtner nombre del oer desde la tupla
            stringbd=str(tupla) #convertir a string
            objetotxt=ObjManipOers.ExtraeTextoPdfs(nombrearchdd)
            objetotxtnorma=ObjManipOers.NormalizaTexto(objetotxt)
            objetoidioma=ObjManipOers.DetectarIdioma(objetotxt)
```

```
        if objetoidioma=='english':
            print "-- Extrayendo data del OER: "+nombre_oer_ocw+"--"
            #obtener sujeto y objeto de la tabla origen para enviar a la tabla destino
            sujetobd= stringbd.split(',')[-3] #toma el sujeto de la tabla origen con los caracteres ('
            sujeto_limpio=sujetobd[2:len(sujetobd)-1] #limpia el sujeto, elimina los caracteres ('
            objetobd= stringbd.split(',')[-1] # toma el objeto de la tabla origen con los caracteres ')
            #insertar en la tabla
            print "Obteniendo URL del OER.."
            objeto_limpio=objetobd[2:len(objetobd)-2] #limpia el objeto, elimina los caracteres ')
            resultadovalidar=ObjManipVirt.consultaValidacion(sujeto_limpio,
"http://xmlns.com/foaf/0.1/Document","")
            if len(resultadovalidar["results"]["bindings"])==0:
self.VerificarDatosEnVirtuoso(sujeto_limpio,"http://xmlns.com/foaf/0.1/Document",objeto_limpio )
                #
graph.add([URIRef(sujeto_limpio),URIRef("http://xmlns.com/foaf/0.1/Document"),URIRef(objeto_limpio)])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Titulo.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://purl.org/spar/doco/Title","")
            if len(resultadovalidar["results"]["bindings"])==0:
                self.VerificarDatosEnVirtuoso(objeto_limpio,"http://purl.org/spar/doco/Title",nombre_oer_ocw )
    #
graph.add([URIRef(objeto_limpio),URIRef("http://purl.org/spar/doco/Title"),Literal(nombre_oer_ocw,
lang="en")])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Páginas.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://www.aktors.org/ontology/portal#has-page-numbers","")
            if len(resultadovalidar["results"]["bindings"])==0:
                objetopaginas=ObjManipOers.PdfsMetadatosPaginas(nombrearchdd)
    #             pickle_objetopaginas=pickle.dumps(objetopaginas) #encapsulamiento del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio,"http://www.aktors.org/ontology/portal#has-page-
numbers",objetopaginas)
    #             graph.add([URIRef(objeto_limpio),URIRef("http://www.aktors.org/ontology/portal#has-page-
numbers"),Literal(objetopaginas)]) #, datatype=XSD["integer"]
            else:
                print "¡Datos existentes!"
            print "Obteniendo Idioma.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://purl.org/dc/elements/1.1/language","")
            if len(resultadovalidar["results"]["bindings"])==0:
                self.VerificarDatosEnVirtuoso(objeto_limpio, "http://purl.org/dc/elements/1.1/language",
objetoidioma)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://purl.org/dc/elements/1.1/language"),Literal(objetoidioma)]
)
            else:
                print "¡Datos existentes!"
            print "Obteniendo Texto.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://www.lexinfo.net/ontology/2.0/lexinfo#standardText","")
            if len(resultadovalidar["results"]["bindings"])==0:
                self.VerificarDatosEnVirtuoso(objeto_limpio,
"http://www.lexinfo.net/ontology/2.0/lexinfo#standardText", objetotxtnorma)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://www.lexinfo.net/ontology/2.0/lexinfo#standardText"),Liter
al(objetotxtnorma, lang="en")])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Tokens.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio, "http://persistence.uni-
leipzig.org/nlp2rdf/ontologies/nif-core#Word","")
```

```
            if len(resultadovalidar["results"]["bindings"])==0:
                objetotokns=ObjManipOers.Tokeniza(objetotxtnorma)
                pickle_objetotokns=pickle.dumps(objetotokns) #encapsulamiento del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio, "http://persistence.uni-
leipzig.org/nlp2rdf/ontologies/nif-core#Word", pickle_objetotokns)
    #               graph.add([URIRef(objeto_limpio),URIRef("http://persistence.uni-
leipzig.org/nlp2rdf/ontologies/nif-core#Word"),Literal(pickle_objetotokns)])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Tags.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://www.w3.org/ns/dcat#keyword","")
            if len(resultadovalidar["results"]["bindings"])==0:
                objetotag=ObjManipOers.AgregaTag(objetotokns)
                pickle_objetotag=pickle.dumps(objetotag) #encapsulamiento del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio,"http://www.w3.org/ns/dcat#keyword",
pickle_objetotag)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://www.w3.org/ns/dcat#keyword"),Literal(pickle_objetotag)])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Tokens representativos.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://purl.org/spar/doco/Glossary","")
            if len(resultadovalidar["results"]["bindings"])==0:
                objetotoknsrepres=ObjManipOers.TokensRepresentativos(objetotokns)
                pickle_objetotoknsrepres=pickle.dumps(objetotoknsrepres) #encapsulamiento del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio, "http://purl.org/spar/doco/Glossary",
pickle_objetotoknsrepres)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://purl.org/spar/doco/Glossary"),Literal(pickle_objetotoknsre
pres)])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Palabras desambiguas.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense","")
            if len(resultadovalidar["results"]["bindings"])==0:
                objetopalabrasambiguas=ObjAnDesam.ProcesoDesambiguacion(objetotxtnorma)
                pickle_objetopalabrasambiguas=pickle.dumps(objetopalabrasambiguas) #encapsulamiento
del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio,
"http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense", pickle_objetopalabrasambiguas)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://www.w3.org/2006/03/wn/wn20/schema/containsWordSen
se"),Literal(pickle_objetopalabrasambiguas)])
            else:
                print "¡Datos existentes!"
            print "Obteniendo Entidades.."
            resultadovalidar=ObjManipVirt.consultaValidacion(objeto_limpio,
"http://purl.org/spar/doco/ListOfAgents","")
            if len(resultadovalidar["results"]["bindings"])==0:
                objetoentidades=ObjEnti.ObtenerEntidades(objetotxt)
                pickle_objetoentidades=pickle.dumps(objetoentidades) #encapsulamiento del objeto lista
                self.VerificarDatosEnVirtuoso(objeto_limpio, "http://purl.org/spar/doco/ListOfAgents",
pickle_objetoentidades)
    #
graph.add([URIRef(objeto_limpio),URIRef("http://purl.org/spar/doco/ListOfAgents"),Literal(pickle_objetoenti
dades)])
            else:
                print "¡Datos existentes!"
#               store.commit()
        except Exception, e:
            print e
```

13. ClaseManipulacionVirtuoso.py

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 26 19:16:37 2014
@author: scecy
"""
from rdflib.graph import Graph
from rdflib.store import Store
from rdflib.plugin import get as plugin
from rdflib.namespace import RDF, RDFS, XSD, DC
from rdflib.term import URIRef, Literal
from datetime import datetime
from virtuoso.vstore import Virtuoso
from SPARQLWrapper import SPARQLWrapper, JSON, XML, N3, RDF

class Clase_Virtuoso():
    """Conexion con Virtuoso"""
    def CrearConexion (self):
        V = plugin("Virtuoso", Store)
        assert V is Virtuoso
        store = Virtuoso("DSN=VOS;UID=silvana;PWD=12345;WideAsUTF16=Y") #parametros de conexion
con la bd
#       store = Virtuoso("DSN=VOS;UID=dba;PWD=admin;WideAsUTF16=Y") #parametros de conexion
con la bd
        return store
    def CrearGrafo (self, store):
        graph = Graph(store, identifier=URIRef("http://dataoers.org/")) #creacion del grafo que alamacenará
la data de los oers
        return graph
    def ConsultarPalabrasDesambiguas (self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""
                SELECT DISTINCT ?sujeto, ?objeto
                FROM <http://dataoers.org/>
                WHERE {?sujeto <http://www.w3.org/2006/03/wn/wn20/schema/containsWordSense> ?objeto}
                Limit 200
                """)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        lista=[]
        for result in results["results"]["bindings"]:
            tripleta=result["sujeto"]['value'], result["objeto"]['value']
            lista.append(tripleta)
        return (lista)
    def ConsultarEntidades (self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""
                SELECT DISTINCT ?sujeto, ?objeto
                FROM <http://dataoers.org/>
                WHERE {?sujeto <http://purl.org/spar/doco/ListOfAgents> ?objeto}
                Limit 200
                """)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        lista=[]
        for result in results["results"]["bindings"]:
            tripleta=result["sujeto"]['value'], result["objeto"]['value']
            lista.append(tripleta)
        return (lista)
    def ConsultarTokenComun(self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
```

```
        sparql.setQuery("""SELECT DISTINCT ?o FROM <http://dataoers.org/> WHERE { ?s
<http://www.w3.org/2006/03/wn/wn20/schema/word> ?o }""")
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        return(results)
    def ConsultarEntidadComun(self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""SELECT DISTINCT ?o FROM <http://dataoers.org/> WHERE { ?s
<http://www.w3.org/2002/07/owl#NamedIndividual> ?o }""")
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        return(results)
    def ConsultarSujetoVisualizar (self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""
                SELECT DISTINCT ?sujeto
                FROM <http://dataoers.org/>
                WHERE {?sujeto <http://purl.org/dc/elements/1.1/relation> ?objeto} limit 300

                """) #se pueede cambiar la consulta sparql, el predicado debe ser el mismo para
ConsultarObjetoVisualizar
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        lista=[]
        for result in results["results"]["bindings"]:
            tripleta=result["sujeto"]['value']
            lista.append(tripleta)
        return (lista)
    def ConsultarObjetoVisualizar (self,sujeto):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""
                SELECT ?objeto WHERE { <'"+sujeto+"'> <http://purl.org/dc/elements/1.1/relation> ?objeto} limit
30
                """)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        lista=[]
        for result in results["results"]["bindings"]:
            tripleta=result["sujeto"]['value'], result["objeto"]['value']
            lista.append(tripleta)
        return (lista)
    def ConsultarNombresOers(self,sujeto):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""SELECT ?objeto
                        FROM <http://dataoers.org/>
                        WHERE { <"""+sujeto+"""> <http://purl.org/spar/doco/Title> ?objeto}""")
        #se pueede cambiar la consulta sparql, el predicado debe ser el mismo para
ConsultarSujetoVisualizar
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        for result in results["results"]["bindings"]:
            nombre=str(result["objeto"]['value'])
        return (nombre)
    def ConsultarOers(self):
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        sparql.setQuery("""
                SELECT DISTINCT * WHERE { ?s <http://xmlns.com/foaf/0.1/Document> ?o } limit 25000
                """)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        lista=[]
        for result in results["results"]["bindings"]:
            tripleta=result["sujeto"]['value'], result["objeto"]['value']
            lista.append(tripleta)
```

```
    return (lista)
def consultaValidacion(self, s, p, o):
    try:
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        txtQuery = """select * where {<%s> <%s> ?s} """%(s,p)
        sparql.setQuery(txtQuery)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        return results
    except Exception, e:
        print e
def consultaValidacionRelacion(self, s, p, o):
    try:
        sparql = SPARQLWrapper("http://localhost:8890/sparql")
        txtQuery = """select * where {<%s> <%s> <%s>} """%(s,p,o)
        sparql.setQuery(txtQuery)
        sparql.setReturnFormat(JSON)
        results = sparql.query().convert()
        return results
    except Exception, e:
        print e
```

14. ClaseEnriquecerDbpedia.py

```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar 29 15:22:10 2014

@author: scecy
"""
from SPARQLWrapper import SPARQLWrapper, JSON
from ClaseManipulacionVirtuoso import *
class Clase_EnriquecerDbpedia():
    def consultarDbpedia(self, token):
        try:
            sparql = SPARQLWrapper("http://dbpedia.org/sparql")
            txtQuery = """prefix dbpedia: <http://dbpedia.org/resource/> select distinct ?o where
{dbpedia:"""+token+""" <http://xmlns.com/foaf/0.1/isPrimaryTopicOf> ?o} """
            sparql.setQuery(txtQuery)  #"+token+"
            sparql.setReturnFormat(JSON)
            results = sparql.query().convert()
            return (results)
        except Exception, e:
            print e
    def ObtenerURLEnriquecer(self, resultslh, uripalabra):
        try:
            for resultlh in resultslh["results"]["bindings"]:
                urltokn=str(resultlh["o"]['value'])
                objtokn=urltokn.split("/")[-1]
                print "Se buscará datos en DBpedia con la palabra:",objtokn
                resultsdbp=self.consultarDbpedia(objtokn)

                if len(resultsdbp["results"]["bindings"])==0:
                    Objtokn=objtokn.capitalize()
                    print "No se han encontrado datos en DBpedia, se buscará con la palabra:",Objtokn
                    resultsDbp = self.consultarDbpedia(Objtokn)

                    if len(resultsDbp["results"]["bindings"])==0:
                        print "No se han encontrado datos en DBpedia.."
                        pass

                    if len(resultsDbp["results"]["bindings"])!=0:
                        for resultDbp in resultsDbp["results"]["bindings"]:
                            urlenriqDbp=str(resultDbp["o"]['value'])
                        print "Datos encontrados en DBpedia",urlenriqDbp
                        self.GuardarUrlDbpedia(urlenriqDbp,objtokn,uripalabra)

    #                   return (enriqDbp)
                else:
#                 if len(resultsdbp["results"]["bindings"])!=0:
                    for resultdbp in resultsdbp["results"]["bindings"]:
                        urlenriqdbp=str(resultdbp["o"]['value'])
                    print "Datos encontrados en DBpedia", urlenriqdbp
                    self.GuardarUrlDbpedia(urlenriqdbp, objtokn, uripalabra)
    #               return(enriqdbp)
        except Exception, e:
            print e

    def GuardarUrlDbpedia(self, urldbp, token, uripalabra):
        try:
            ObjManipVirt = Clase_Virtuoso()
            urisujeto=uripalabra+token #sujeto uri  + token
            print urisujeto
```

```
        store = ObjManipVirt.CrearConexion()
        graph = ObjManipVirt.CrearGrafo(store)
        resultvalidar=ObjManipVirt.consultaValidacion(urisujeto, "http://www.w3.org/2002/07/owl#sameas",
urldbp)
        print len(resultvalidar["results"]["bindings"])

        if len(resultvalidar["results"]["bindings"])==0:
           graph.add([URIRef(urisujeto),URIRef("http://www.w3.org/2002/07/owl#sameas"),URIRef(urldbp)])
            print "Datos guardados en Virtuoso en el grafo http://dataoers.org/"
        else:
#             graph.update("INSERT { <"+urisujeto+"> <http://www.w3.org/2002/07/owl#sameas>
<"+urldbp+">.}")
            print "Datos actualizados en Virtuoso en el grafo http://dataoers.org/"
      except Exception, e:
        print e
```

15. service.py

```
from flask import Flask, render_template, request, redirect, url_for, abort, session, jsonify, Response,
current_app
from functools import wraps
from ClaseCrearJsonOers import *

# Initialize the Flask application
app = Flask(__name__)

def jsonp(func):
    """Wraps JSONified output for JSONP requests."""
    @wraps(func)
    def decorated_function(*args, **kwargs):
        callback = request.args.get('callback', False)
        if callback:
            data = str(func(*args, **kwargs).data)
            content = str(callback) + '(' + data + ')'
            mimetype = 'application/javascript'
            return current_app.response_class(content, mimetype=mimetype)
        else:
            return func(*args, **kwargs)
    return decorated_function

@app.route('/')
def inicio():
    return "Servicio web en python - flask"

# This route will return a list in JSON format
@app.route('/visualizaroerjit')
@jsonp
def visualizaroerjit():
    try:
        objCrearJson= CrearJsonOers()
        JsonRetorno= objCrearJson.JsonJit()
        # lst=JsonRetorno
         ## jsonify will do for us all the work, returning the
    ## previous data structure in JSON
        #r = jsonify({'lista':lst})
        r = jsonify({'lista':JsonRetorno})
        return r
    except Exception, e:
        print e
if __name__ == '__main__':
    app.debug = True
    app.run()
```

16. ClaseCrearJsonOers.py

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar  4 14:31:11 2014
@author: scecy
"""
from ClaseManipulacionVirtuoso import *
class CrearJsonOers():
    """crear el json para graficar"""
    def JsonJit(self):
        try:
            #ObjBd = BDatos()
            ObjBd=Clase_Virtuoso()
            cursorSujeto = ObjBd.ConsultarSujetoVisualizar()
            ##    cursorSujeto = BDatos.ConsultarSujetoVisualizar()
            JSON_retorno=[]
            for tuplasujeto in cursorSujeto:
                strtuplasujeto=str(tuplasujeto)
            ##        strtuplasujeto=str(tuplasujeto)
            ##        aux_sujeto=strtuplasujeto[2:-3]
            #    #            print aux_sujeto
                strJSON_retorno={
                            "adjacencies": [],
                            "data": {"$color": "83548B",
                                    "$type": "circle",
                                    "$dim": 10
                                    },
                            "id":"",
                            "name":""
                            }
                cursorObjeto = ObjBd.ConsultarObjetoVisualizar(strtuplasujeto)
                if cursorObjeto:
                    for tuplaobjeto in cursorObjeto:
                        strtuplaobjeto=str(tuplaobjeto)
            #                aux_objeto=strtuplaobjeto[2:-3]
            #    #                print aux_objeto
                        adiacencia={"nodeTo": strtuplaobjeto,
                                    "nodeFrom": strtuplasujeto,
                                    "data": {"$color": "#909291"}
                                    }
                        strJSON_retorno["adjacencies"].append(adiacencia)
                nombreoer=ObjBd.ConsultarNombresOers(strtuplasujeto)
                strJSON_retorno["id"]=strtuplasujeto
                #strJSON_retorno["name"]=strtuplasujeto
                strJSON_retorno["name"]=nombreoer
                JSON_retorno.append(strJSON_retorno)
            return JSON_retorno
            #print JSON_retorno
        except Exception, e:
            print e
```

17. Visualizargrafo.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 TRANSITIONAL//EN"
"HTTP://WWW.W3.ORG/TR/XHTML1/DTD/XHTML1-TRANSITIONAL.DTD">
<HTML XMLNS="HTTP://WWW.W3.ORG/1999/XHTML" XML:LANG="EN" LANG="EN">
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-8" />
<!--A HREF="/" TITLE="UNIVERSIDAD TECNICA PARTICULAR DE LOJA - U T P L | ECUADOR "></A--
>
<TITLE>TAWSBC - GRAFO DE RELACIONES ENTRE OER'S</TITLE>

<!-- CSS FILES -->
<LINK TYPE="TEXT/CSS" HREF="BASE.CSS" REL="STYLESHEET" />
<LINK TYPE="TEXT/CSS" HREF="FORCEDIRECTED.CSS" REL="STYLESHEET" />

<!--[IF IE]><SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT"
SRC="../../EXTRAS/EXCANVAS.JS"></SCRIPT><![ENDIF]-->

<!-- JIT LIBRARY FILE -->
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT" SRC="JIT.JS"></SCRIPT>

<!-- EXAMPLE FILE -->
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT"
SRC="VISUALIZARGRAFO.JS"></SCRIPT>
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT" SRC="JQUERY-
2.1.0.MIN.JS"></SCRIPT>
</HEAD>

<BODY ONLOAD="INIT();">
        <DIV ID="CABECERA_TITULO">
                <IMG
SRC="HTTP://WWW.UTPL.EDU.EC/SITES/ALL/THEMES/UTPL/IMAGES/LOGO.PNG"
ALT="UNIVERSIDAD TECNICA PARTICULAR DE LOJA - U T P L | ECUADOR " ID="HOMELOGO"/>
    <DIV CLASS="TEXT">
        <BR />
        TECNOLOGÍAS AVANZADAS EN LA WEB Y SISTEMAS BASADOS EN EL
CONOCIMIENTO<BR />
            VISUALIZADOR DE RELACIONES ENTRE RECURSOS EDUCATIVOS ABIERTOS
(OER'S)<BR />
    </DIV>
                <DIV ID="ID-LIST"></DIV>
        </DIV>
        <DIV ID="INFOVIS-CONTAINER">
                <B/>
                EL GRAFO CONTIENE NODOS O CIRCULOS QUE REPRESENTA A UN OER, PARA
VER LA LISTA DE RELACIONES DEBE HACER CLIC SOBRE ÉL.</B>
                <DIV ID="INFOVIS"></DIV>
        </DIV>
        <DIV ID="LOG"></DIV>

        <DIV ID="RELACIONES-CONTAINER">
                <B/>
                <B> LISTA DE RELACIONES ENTRE OER'S.</B> <BR/>
                <DIV ID="INNER-DETAILS"></DIV>
        </DIV>
</BODY>
</HTML>
```

18. visualizargrafo.js

```
var labelType, useGradients, nativeTextSupport, animate;

(function() {
  var ua = navigator.userAgent,
      iStuff = ua.match(/iPhone/i) || ua.match(/iPad/i),
      typeOfCanvas = typeof HTMLCanvasElement,
      nativeCanvasSupport = (typeOfCanvas == 'object' || typeOfCanvas == 'function'),
      textSupport = nativeCanvasSupport
        && (typeof document.createElement('canvas').getContext('2d').fillText == 'function');
  //I'm setting this based on the fact that ExCanvas provides text support for IE
  //and that as of today iPhone/iPad current text support is lame
  labelType = (!nativeCanvasSupport || (textSupport && !iStuff))? 'Native' : 'HTML';
  nativeTextSupport = labelType == 'Native';
  useGradients = nativeCanvasSupport;
  animate = !(iStuff || !nativeCanvasSupport);
})();

var Log = {
  elem: false,
  write: function(text){
    if (!this.elem)
      this.elem = document.getElementById('log');
    this.elem.innerHTML = text;
    this.elem.style.left = (500 - this.elem.offsetWidth / 2) + 'px';
  }
};

function init(){
  $.ajax({
    url: "http://localhost:310/visualizaroerjit", //URL del servicio
    //url: "http://taw02.utpl.edu.ec/anavisoers/webservice",
    contentType: "application/json; charset=utf-8",
    dataType: "jsonp", // sirve para que retorne en JSON
    success: function(data){
      // init data
      var json = data.lista;

      // end
      // init ForceDirected
      var fd = new $jit.ForceDirected({
        //id of the visualization container
        injectInto: 'infovis',
        //Enable zooming and panning
        //by scrolling and DnD
    Navigation: {
      enable: true,
      //Enable panning events only if we're dragging the empty
      //canvas (and not a node).
      panning: 'avoid nodes',
      zooming: 10 //zoom speed. higher is more sensible
    },
    // Change node and edge styles such as
    // color and width.
    // These properties are also set per node
    // with dollar prefixed data-properties in the
    // JSON structure.
    Node: {
      overridable: true,
    },
    Edge: {
```

```
    overridable: true,
    color: '#23A4FF',
    lineWidth: 0.4
  },
  //Native canvas text styling
  Label: {
    type: labelType, //Native or HTML
    size: 10,
    //style: 'bold',
    color: '#666'
  },
  //Add Tips
  Tips: {
    enable: true,
    onShow: function(tip, node) {
      //count connections
      var count = 0;
      node.eachAdjacency(function() { count++; });
      //display node info in tooltip
      tip.innerHTML = "<div class=\"tip-title\"><b>Nombre del OER:  </b>" + node.name + "</div>"
        + "<div class=\"tip-text\"><b>Cantidad de Relaciones:</b> " + count + "</div>";
    }
  },
  // Add node events
  Events: {
    enable: true,
    type: 'Native',
    //Change cursor style when hovering a node
    onMouseEnter: function() {
      fd.canvas.getElement().style.cursor = 'move';
    },
    onMouseLeave: function() {
      fd.canvas.getElement().style.cursor = '';
    },
    //Update node positions when dragged
    onDragMove: function(node, eventInfo, e) {
        var pos = eventInfo.getPos();
        node.pos.setc(pos.x, pos.y);
        fd.plot();
    },
    //Implement the same handler for touchscreens
    onTouchMove: function(node, eventInfo, e) {
      $jit.util.event.stop(e); //stop default touchmove event
      this.onDragMove(node, eventInfo, e);
    },
    //Add also a click handler to nodes
    onClick: function(node) {
      if(!node) return;
      // Build the right column relations list.
      // This is done by traversing the clicked node connections.
      var html = "<b>OER seleccionado:</b><h4><a href="' + node.id +'">" + node.name +
"</a></h4><b>Relaciones con otros OER's:</b><ul><li>",
          list = [];
      node.eachAdjacency(function(adj){
        list.push("<a href="'+adj.nodeTo.id+'">"+adj.nodeTo.name);
      });
      //append connections information
      $jit.id('inner-details').innerHTML = html + list.join("</li><li>") + "</li></ul>";
    }
  },
  //Number of iterations for the FD algorithm
  iterations: 200,
  //Edge length
  levelDistance: 130,
```

```
    // Add text to the labels. This method is only triggered
    // on label creation and only for DOM labels (not native canvas ones).
    onCreateLabel: function(domElement, node){
      domElement.innerHTML = node.name;
      var style = domElement.style;
      style.fontSize = "1em";
      style.color = "#666";
    },
    // Change node styles when DOM labels are placed
    // or moved.
    onPlaceLabel: function(domElement, node){
      var style = domElement.style;
      var left = parseInt(style.left);
      var top = parseInt(style.top);
      var w = domElement.offsetWidth;
      style.left = (left - w / 2) + 'px';
      style.top = (top + 10) + 'px';
      style.display = '';
      style.color = "#666";
    }
  });
  // load JSON data.
  fd.loadJSON(json);
  // compute positions incrementally and animate.
  fd.computeIncremental({
    iter: 40,
    property: 'end',
    onStep: function(perc){
      Log.write(perc + '% cargando...');
    },
    onComplete: function(){
      Log.write('');
      fd.animate({
        modes: ['linear'],
        transition: $jit.Trans.Elastic.easeOut,
        duration: 2500
      });
    }
  });  // end);
}
```

19. Trabajos Relacionados

Podemos citar algunos proyectos que están siendo centro de compartición de recursos educativos libres y de calidad.

- **MitOpenCourseWare (Massachusetts Institute of Technology):** Nace como un primer Proyecto editorial OER/OCW en el Instituto de Tecnología de Massachusetts en el año 2001 con vocación internacional y de creación de un movimiento de difusión libre de materiales docentes, con el fomento del trabajo cooperativo.
- **Universia:** Fundación Universia persigue favorecer la inclusión educativa y laboral de las personas con discapacidad, con especial atención en el ámbito universitario y en el empleo de calidad, utilizando como instrumento las Nuevas Tecnologías de la Información y las Comunicaciones. Basándose en el OpenCourseWare (OCW) que es una iniciativa editorial electrónica a gran escala y vincula recursos por medio de la coperación de universidades a nivel de Latinoamérica.
- **OCWConsortium:** Este proyecto nace en el 2006 como una comunidad mundial de cientos de instituciones de educación superior y organizaciones asociadas se comprometieron a avanzar en OpenCourseWare y su impacto en la educación global. Servimos como un recurso para iniciar y mantener proyectos OCW, como un órgano de coordinación del movimiento a escala mundial, y como un foro para el intercambio de ideas y la planificación futura.

A continuación podemos citar algunos proyectos que se han formado a partir de la necesidad de vinculación de datos libres a través de la web, sirviendo de esta manera como punto de encuentro de dichos recursos.

- **Open Data:** Es el mejor de los ejemplos de vinculación de datos por medio de Linked Data, objetivo es impulsar la web de datos mediante la identificación de conjuntos de datos existentes que estén disponibles bajo licencias abiertas, convirtiéndolas en RDF bajo los principios de Linked Data y publicarlos en la web

- **Course Ware:** Es un repositorio semántico que contiene y publica datos vinculados RDF, los datos presentes son provistos desde el proyecto ReSIST, el mismo que posee más de 50 millones de ítems, recomendados para todos los involucrados con la enseñanza de los temas relacionados.

Podemos encontrar también algunos sitios de interés que aporten información acerca de calidad y contenidos OER/OCW como los citados a continuación:

- **Opal:** Nace por la necesidad de obtener una mejora en cantidad y calidad de recursos educativos abiertos que pueden ser incorporados a la educación superior. En 2012, la Iniciativa OPAL ha establecido las bases para el concepto emergente de las prácticas educativas abiertas. Se ha construido para ser fácil de utilizar.
- **Plataforma OER:** A partir de noviembre del 2011 está puesta en marcha la Plataforma REA es una nueva e innovadora plataforma en línea que ofrecerá una selección de publicaciones de la UNESCO como REA plenamente autorizados. La Plataforma se ha creado con software libre de la Universidad de Witwatersrand (Sudáfrica) en el marco del consorcio AVOIR (Africa Virtual Open Initiatives and Resources) integrado por once universidades africanas.

20. Herramientas Utilizadas

Las herramientas que se utilizarán sé eligieron en base a sus prestaciones, funcionalidades, usabilidad, dando prioridad al software libre con la finalidad de mantener los principios de Open Course Ware Consortium.

- **PYTHON:** Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de comics ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que ofrece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

- **NLTK:** Es un Kit de Herramientas del Lenguaje Natural, define una infraestructura que puede ser utilizado para construir programas basados en el Procesamiento de Lenguaje Natural. Fue diseñado con cuatro objetivos principales:
  - ✓ **Simplicidad:** proporciona un framework intuitivo con importantes bloques de construcción, dando a los usuarios un conocimiento práctico de PLN (Programación en Lenguaje Natural) sin atascarse en el tedioso mantenimiento asociado con el procesamiento del idioma de los datos.
  - ✓ **Consistencia:** proporciona un framework uniforme con interfaces y estructuras de datos consistentes los nombres de los métodos son fáciles de recordar.
  - ✓ **Extensibilidad:** proporciona una estructura en la cual los nuevos módulos de software puedan ser adaptados con facilidad, incluyendo implementaciones alternativas y enfoques que compitan para la misma tarea.
  - ✓ **Modularidad:** proporciona componentes que se puedan utilizar de forma independiente sin necesidad de entender la herramienta en su totalidad.

- **Pypdf:** biblioteca basada en Python construida con el propósito de manejar archivos PDF, también puede funcionar con objetos StringIO en lugar de secuencias de archivo, lo que permite la manipulación de PDF en la memoria. Por tanto, es una herramienta útil para los sitios web que manejan o manipulan Pdf, es capaz de:
    - ✓ Extraer información de documentos (título, autor, etc.)
    - ✓ Divide el documento en página por página
    - ✓ Combina el documento página por página
    - ✓ Corta paginas
    - ✓ Funciona varias páginas en una sola
    - ✓ Encriptado y des encriptado de archivos PDF

- **Unicodedata:** este módulo permite el acceso a la base de datos de caracteres Unicode que define las propiedades de carácter para todos los caracteres Unicode. Los datos de esta base de datos se basan en la versión del archivo UnicodeData.txt 5.2.2. El modulo utiliza los mismos nombres y los símbolos definidos por el formato de archivo Unicodedata 5.2.0

- **Urllib2:** este módulo define las funciones y clases que ayudan abrir las URL (principalmente HTTP) autenticación implícita, redirecciones, cookies y más.

- **Zipfile:** El formato de archivo ZIP es un archivo común y estándar de compresión. Este módulo proporciona herramientas para crear, leer, escribir, añadir y listar un archivo ZIP. Cualquier uso avanzado de este módulo requerirá un entendimiento del formato está definido en las notas de las aplicación PKZIP.

- **JIT:** InfoVis Toolkit JavaScript proporciona herramientas para crear visualizaciones interactivas de datos para la Web. El kit de herramientas implementa funciones avanzadas de visualización de la información como

TreeMaps, una visualización adaptada de árboles basado en SpaceTree, una técnica de enfoque-marco para trazar arboles hiperbólicos, una disposición radial de los arboles con el llamado animaciones avanzadas RGraph y otras visualizaciones.

- **DBpedia:** DBpedia es proyecto para la extracción de datos de Wikipedia para proponer una versión Web semántica. Este proyecto es realizado por la Universidad de Leipzig, Universidad Libre de Berlín y la compañía OpenLink Software. El contenido de la base de datos está disponible bajo licencia CC-BY-SA3.0 y GFDL (ya que el contenido se basa en la Wikipedia). El motor de extracción de datos se realiza con Scala5 un software libre publicado bajo el GNU General Public License. Su código fuente se distribuye, se alberga en sourceforge y disponible a través de Subversion.

- **Virtuoso:** Es una innovadora herramienta de procesamiento en tiempo real, es un servidor de datos multi-modelo plataforma incomparable para la gestión de datos, acceso e integración. La arquitectura de servidor híbrido único de Virtuoso ofrece la funcionalidad de servidor tradicional cubre las siguientes áreas:
  - ✓ Gestión de Datos Relacional
  - ✓ Gestión de datos RDF
  - ✓ XML Data Management
  - ✓ Gestión Texto libre Contenido e indización de texto completo
  - ✓ Document Server Web
  - ✓ Linked Data Server
  - ✓ Servidor de aplicaciones Web
  - ✓ Servicios de Desarrollo Web (SOAP o REST)

- **Flask:** Flask es un framework minimalista escrito en Python y basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2. Tiene la licencia BSD.

21. Herramientas para DBpedia

Es importante mencionar herramientas adicionales para utilizar Dbpedia, a continuación se presentan dos opciones.

21.1. Live Dbpedia

Partiendo de la visión del usuario final que desea utilizar la Web Semántica, se necesita una herramienta que nos facilite realizar consultas sobre cualquier concepto que se desee, además que cuente con una interfaz sencilla y amigable de usar. DBpedia nos ofrece “Live Dbpedia” (http://live.dbpedia.org/page) que cumple con estos requisitos, la imagen siguiente muestra un extracto del resultado para la consulta realizada sobre la palabra “Human”.

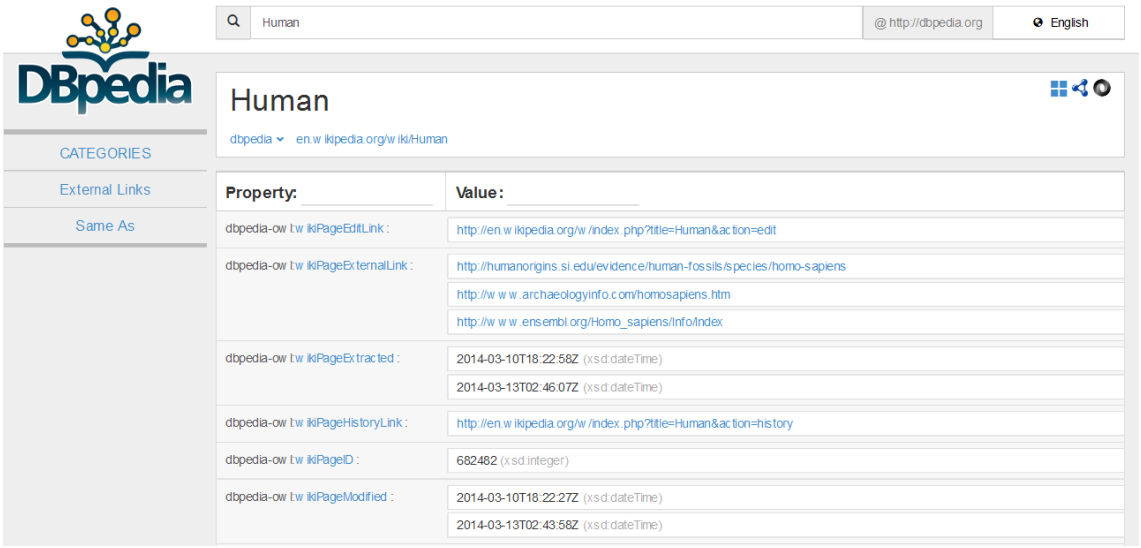


Figura 69: Frontal LiveDbpedia
Fuente: http://live.dbpedia.org/page

Este frontal nos genera toda la información semántica relacionada con esta palabra sin necesidad de tener conocimientos en consultas SPARQL, es de facil uso y orientada al usuario final.

21.2. Dbpedia Spotlight

Dbepdias es un datset muy potente para ser utilizado en tareas relacionadas con el procesamiento del lenguaje natural, pensando en esta característica se creó el proyecto DBpedia Spotlight[7]. La imagen siguiente muestra una de las aplicaciones que se le puede dar.

Figura 70: Demo sobre el uso de Dbpedia Spotlight

Fuente: http://live.dbpedia.org/page

Este frontal permite ingresar un texto y realiza el reconocimiento de las entidades que han sido mencionadas para posteriomente enlazarlos con sus identificadores unicos en la web.

Spotlight puede ser utilizado para tareas de extracción de información, para mostrar información complementaria en paginas web ó para mejorar las tareas de recuperación de información; con esta herramienta y siguiendo los enlaces desde DBpedia, se puede tener un acercamiento más factible a la Web de Datos.

[7] http://spotlight.dbpedia.org