UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIONES DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

Desarrollo de Servicios Web para el proceso de Enlace y Enriquecimiento de Datos Enlazados

TRABAJO DE FIN DE TITULACIÓN

AUTOR: Montaño Sozoranga, Wilmer Fabricio

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: Desarrollo de Servicios Web para el proceso de Enlace y Enriquecimiento de Datos Enlazados. Piloto: dominio de datos Universitarios, realizado por Montaño Sozoranga Wilmer Fabricio , ha sido orientado y revisado durante su ejecución, por se aprueba la presentación del mismo.

Loja, noviembre de 2014

f) ……..

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

"Yo Montaño Sozoranga Wilmer Fabricio declaro ser autor (a) del presente trabajo de fin de titulación: Desarrollo de Servicios Web para el proceso de Enlace y Enriquecimiento de Datos Enlazados. Piloto: dominio de datos Universitarios, de la Titulación de Ingeniería 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 científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad"

f. ……

Autor: Montaño Sozoranga Wilmer Fabricio

Cédula: 11104634421

Contenido

INDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPITULO I: MARCO TEÓRICO

1. Datos Enlazados

1.1. **Introducción.**

En sus inicios la web en su primera versión 1.0, donde web era rígida en cuanto a la entrega de información, además de poco actualizada, convertía al visitante de un sitio web un simple lector, restringido de cualquier interacción. Se puede decir que la web no era más que paginas enlazadas mediante hipervínculos.

La web que siempre está creciendo y evolucionado, alcanza su versión conocida como la web 2.0 en donde usuario juega el papel más importante, es quien evalúa, puede calificar, compartir, rectificar, alimentar y subir su propia información a la web. Esto producto de le aparición de nuevas tecnologías y estandarización[1].

Los datos enlazados llegan para dar forma a la siguiente versiona de la web, la web semántica. La W3C[2] los define así : "Linked Data se refiere a la utilización de las mejores prácticas para publicación, estructuración de los datos en la web, de tal forma que puedan ser enlazados entre sí, utilizando tecnología propias de web semántica como RDF, OCW, SPARQL, etc."

Se refiere en si a la estructura de la de la siguiente generación de la web, como es la web semántica, que en si busca que la información que se en publica en internet pueda no solo ser entendida por seres humanos sino también por la maquinas que navegan en la web. En donde a partir de un dato se puede descubrir otros datos por sus relaciones.

1.2. **Principios de Datos Enlazados.**

Tim Berners Lee en su publicación Linked Data - Design Issues (Berners-Lee, Linked Data - Design Issues, 2006)describe cuatro reglas base para la publicación de datos enlazados:

1. Usar URIs como nombre de las cosas
2. Usar URIs HTTP par que esas cosas puedan ser referenciadas
3. Representar los datos en RDF y SPARQL como lenguaje de consulta
4. Incluir enlaces hacia otra cosas, para descubrir más cosas

[1] http://www.evolutionoftheweb.com
[2] http://www.w3.org/

La utilizar de estas reglas para la publicación de datos, permite que estos por las características propias de las tecnologías sobre las cuales se construyen como:

- Las cosas nombradas por URIs son inequívocas y estos serán recursos.
- Los detalles o atributos y las relaciones de los datos van a estar descritos y estructurados en formato RDF
- Se pude acceder o realizar consultas sobre estos mediante SPARQL
- La cosas que se publiquen estarán relacionados

1.3. **Tecnologías.**

1.3.1. ***URI.***

URI (Uniform Resource Identifier) ha sido desarrollado por el IETF[3] y pretende crear un sistema mundial para identificar recursos de todo tipo en la web: documentos, imágenes, programas, servicios, correos electrónicos, etc. Este método combina URNs y URLs, esto es, nombres/direcciones. Se trata de identificar los documentos mediante una secuencia de sintaxis controlada que identifica cada documento de una forma única. (...). Los URIs hacen posible encontrar los recursos bajo una gran variedad de esquemas definidos y métodos de acceso tales como HTTP, FTP, Gopher, news, telnet o correos electrónicos localizables siempre de la misma manera, ya que a un mismo documento se puede acceder desde distintos protocolos. Ya se han establecido una serie de schemes o esquemas direccionados. Los esquemas definidos URI coinciden con los protocolos más usados de Internet. (Lapuente, 2013)

En (Berners-Lee, 2005) el RFC[4] que trata sobre este tema dice lo siguiente: “Un identificador uniforme de recursos (URI) proporciona un medio simple y extensible para la identificación de un recurso”. Es base a los términos que lo conforman se explica como:

“Uniforme: Uniformidad ofrece varios beneficios. Permite diferentes tipos de identificadores de recursos que se utilizarán en el mismo contexto, aun cuando los mecanismos utilizados para acceder a esos recursos pueden ser diferentes. Permite la interpretación semántica uniforme de convenciones sintácticas comunes a través de diferentes tipos de identificadores de recursos (…).

[3] http://www.ietf.org/
[4] http://en.wikipedia.org/wiki/Request_for_Comments

Recurso: Esta especificación no limita el alcance de lo que podría ser un recurso; más bien, el término "recurso" se utiliza en un sentido general de lo que pudiera ser identificado por un URI. Ejemplos conocidos incluyen un documento electrónico, una imagen, una fuente de información con un propósito consistente (por ejemplo, "parte meteorológico de hoy para Los Ángeles"), un servicio (por ejemplo, una puerta de enlace HTTP a SMS), y una colección de otros recursos. Un recurso no es necesariamente accesible a través de Internet; por ejemplo, los seres humanos, las empresas y los libros encuadernados en una biblioteca también pueden ser recurso (…).

Identificador: Un identificador encarna la información necesaria para distinguir lo que se identificó a partir de todas las otras cosas dentro de su ámbito de aplicación de la identificación. Nuestro uso de los términos "identificar" y "identificación" se refieren a este fin de distinguir un recurso de todos los demás recursos, independientemente de cómo se logra ese propósito (ejemplos, nombre, dirección, o el contexto). Estos términos no deben confundirse con la presunción de que un identificador define o encarna la identidad de lo que se hace referencia, aunque esto puede ser el caso de algunos identificadores. Tampoco debe asumirse que un sistema que utiliza los URI tendrá acceso al recurso identificado: en muchos casos, los URI se utilizan para referirse a los recursos sin ninguna intención de que se puede acceder. Del mismo modo, "un" recurso identificado podría no ser singular en la naturaleza (ejemplo, un recurso puede ser un conjunto con nombre o una asignación que varía con el tiempo).” (Berners-Lee, 2005)

Suele existir confusión entre URI, URN y URL, que se decriben a continuación de acuerdo a (Berners-Lee, 2005): “El término "Uniform Resource Locator" (URL[5]) se refiere al subconjunto de URIs que, además de la identificación de un recurso, proporcionan los medios para localizar el recurso mediante la descripción de su mecanismo de acceso primario (por ejemplo, su red de "ubicación"). El término "Uniform Resource Name" (URN[6]) se ha utilizado históricamente para referirse tanto a los URI en el marco del esquema de "URN" [RFC2141], que son necesarios para permanecer globalmente único y persistente, incluso cuando el recurso deja de existir o no está disponible, y a cualquier otro URI con las propiedades de un nombre”. La diferencia en grafico de estos conceptos se la puede ver en el Figura 1.

[5] http://www.ietf.org/rfc/rfc1738.txt
[6] http://www.ietf.org/rfc/rfc3406.txt

Figura 1: Relación entre URI, URL y URN.

Fuente: propio

La estructura de una URI es explicada en el Figura 1 de (Albahari & Albahari, 2012), donde se observan los componentes: Scheme, Authority (Host, Port), PathAndQuery (AbsolutyPath, Query y Fragment), UserInfo.

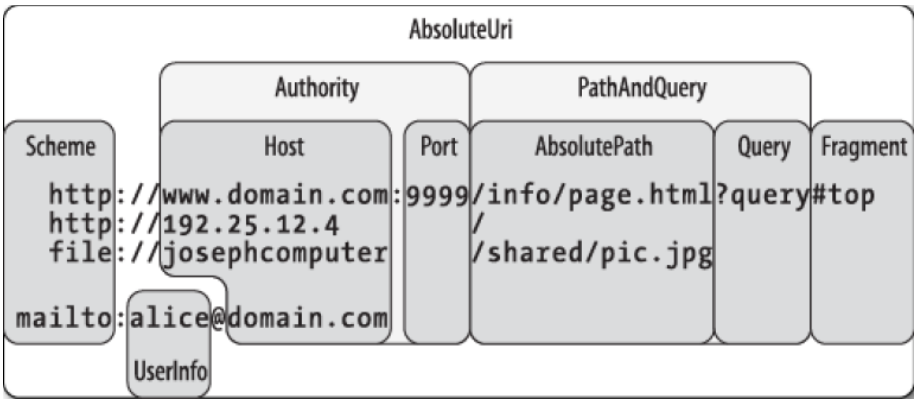


Figura 2: Estructura de la URI (Scheme URI).
Fuente: (Albahari & Albahari, 2012)

(Lapuente, 2013) recoge alguna definiciones sobre las URIs de Architecture of the World Wide Web, Volume One especificación publicada por W3C, que se exponen a continuación:

- URI: Acrónimo para Uniform Resource Identificer.
- URI aliases: dos o más URIs que son -carácter por carácter-, diferentes, pero que identifican el mismo recurso.
- URI overloading: uso del mismo URI para referirse a más de un recurso en el contexto de los protocolos y formatos de la Web.
- URI ownership: la relación entre el agente que asigna y el URI que es definido por un esquema URI.

- URI persistence: la expectación social que desde hace algún tiempo identifica URI a un recurso particular, este podría continuar indefinidamente para referirse al recurso.

- URI reference: un apunte operacional para un URI.

- Uniform Resource Identifier (URI): un identificador global en el contexto de la World Wide Web.

- Namespace document: el recurso de información identificado por un namespace URI en XML.

- Link o enlace: una relación entre dos recursos cuando un recurso (representación) se refiera al otro recurso mediante el significado de un URI.

1.3.2. ***RDF***

El Marco de Descripción de Recursos (RDF) es un lenguaje para representar la información acerca de los recursos en la World Wide Web. En particular, se pretende para la representación de metadatos sobre recursos web, como el título, el autor y la fecha de modificación de una página Web, los derechos de autor y la información de licencia de un documento Web, o el calendario de disponibilidad de algún recurso compartido. Sin embargo, al generalizar el concepto de un "recurso de la Web", RDF puede también ser utilizado para representar la información acerca de las cosas que se pueden identificar en la red, incluso cuando no pueden recuperarse directamente en la Web. Los ejemplos incluyen información acerca de productos disponibles de servicios en línea de la compra (por ejemplo, información acerca de las especificaciones, los precios y la disponibilidad), o la descripción de las preferencias del usuario Web para la entrega de información. (McBride, 2004)

Desde la primera publicación de RDF en su versión 1.0 en febrero del 2004 han pasado 10 años para que en la primera mitad del 2014 poder conocer la versión 1.1, las novedades de esta nueva versión y las diferencias entre las versiones se encuentran en *What's New in RDF 1.1*[7]*,* entre las cuales se resalta dos, la primera que ahora se utiliza IRIs en lugar de URIs para identificar a los recursos y segundo nuevos formatos se serialización, este último se visualizan en la Figura 3.

[7]

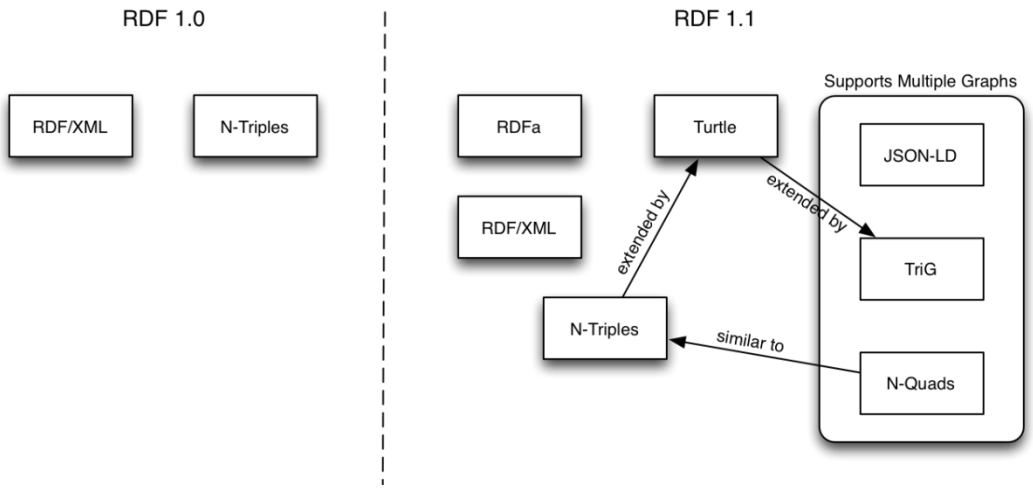


Figura 3. RDF 1.0 y 1.1 formatos de serialización.
Fuente: (Wood, 2014)

1.3.2.1. RDF Data Model.

La estructura básica de la sintaxis abstracta es un conjunto de triples, cada una consiste de un sujeto, un predicado y un objeto. Un conjunto de triples se llama grafo RDF. Un grafo RDF se puede visualizar como un nodo y el diagrama de arco dirigido, en el que cada uno de triple se representa como un enlace nodo-arco-nodo. (Cyganiak, Wood, & Lanthaler, 2014)

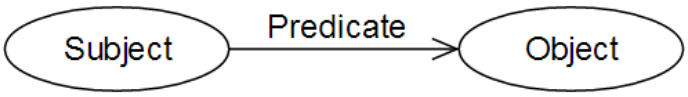


Figura 4: RDF con dos nodos
Fuente: (Cyganiak, Wood, & Lanthaler, 2014)

1.3.2.2. RDF Vocabularios y Espacio de Nombres IRIs

Un RDF vocabulario es una colección de IRIs destinados para ser usados en grafos RDF. Por ejemplo, los IRIs documentados en [RDF11-SCHEMA] [8] son el Vocabulario RDF Esquema (RDF Schema Vocabulary). (Cyganiak, Wood, & Lanthaler, 2014)

Una colección de "términos" para un propósito en particular. Los vocabularios pueden ir desde simples el ampliamente usado Schema RDF utilizado, FOAF [9] y Dublin Core

[8] http://www.w3.org/TR/2014/REC-rdf-schema-20140225/

Metadata Element Set[10] para vocabularios complejos con miles de términos, tales como los utilizados en la asistencia sanitaria para describir síntomas, enfermedades y tratamientos. Vocabularios juegan un papel muy importante en Linked Data, específicamente para ayudar con la integración de datos. El uso de este término se superpone con la Ontología. (W3C, 2013)

El iris de un vocabulario RDF a menudo comienzan con una subcadena común conocido como un espacio de nombres IRI. Algunos IRIs de espacio de nombres se asocian por convención con un nombre corto conocido como un prefijo de espacio de nombres (namespace prefix). (W3C, 2013)

El término "espacio de nombres" por sí no tiene un significado bien definido en el contexto de la RDF, pero a veces se utiliza de manera informal en el sentido de "espacio de nombres IRI" o "vocabulario RDF". (W3C, 2013)

Tabla 1. Ejemplos de Prefijos de Espacios de Nombres e IRIs

Namespace prefix	Namespace IRI	RDF vocabulary
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	The RDF built-in vocabulary [RDF11-SCHEMA][11]
rdfs	http://www.w3.org/2000/01/rdf-schema#	The RDF Schema vocabulary [RDF11-SCHEMA][12]
xsd	http://www.w3.org/2001/XMLSchema#	The RDF-compatible XSD types[13]

Fuente: (Cyganiak, Wood, & Lanthaler, 2014)

1.3.2.3. Especificación de formatos de serialización de RDF

N-Triples

Tripletas N-Triples son una secuencia de términos RDF que representan al sujeto, predicado y objeto de una Tripleta RDF. Estos pueden estar separados por espacios en blanco (espacios U +0020 o tabulaciones U +0009). Esta secuencia es terminada por un '.' y una nueva línea (opcional al final de un documento). (Carothers & Seaborne, 2014)

[9] http://www.foaf-project.org/
[10] http://dublincore.org/documents/dces/
[11] http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#bib-RDF11-SCHEMA
[12] http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#bib-RDF11-SCHEMA
[13] http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/#dfn-rdf-compatible-xsd-types

N-triple en formato en texto plano para grafos RDF, en la Tabla 2 se ejemplifica su estructura.

Tabla 2. Ejemplo N-triple

01	<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
02	<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice#me> .
03	<http://example.org/bob#me> <http://schema.org/birthDate> "1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date> .
04	<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/topic_interest> <http://www.wikidata.org/entity/Q12418> .
05	<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/title> "Mona Lisa" .
06	<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/creator> <http://DBpedia.org/resource/Leonardo_da_Vinci> .
07	<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/terms/subject> <http://www.wikidata.org/entity/Q12418>.

Fuente: (Schreiber & Raimond, 2014)

Turtle

Permite a un grafo RDF a ser completamente escrito en un formulario de texto compacto y natural, con las abreviaturas para los patrones y tipos de datos de uso común. Turtle ofrece niveles de compatibilidad con el formato N-Triples, así como con la sintaxis de patrón de tripletas de la Recomendación de la W3C de SPARQL[14]. (Beckett, Berners-Lee, Prud'hommeaux, Carothers, & Machina., 2014)

Turtle es una extensión del N-Triples. Además de la sintaxis básica N-Triples, Turtle introduce una serie de atajos sintácticos, como el soporte para prefijos de espacio de nombres, listas y abreviaturas para datos tipo literales. Turtle ofrece una compensación entre la facilidad de la escritura, la facilidad de análisis y facilidad de lectura. (Schreiber & Raimond, 2014)

Tabla 3. Ejemplo Turtle

```
01  <http://example.org/>
02  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
03  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
04  PREFIX schema: <http://schema.org/>
05  PREFIX dcterms: <http://purl.org/dc/terms/>
06  PREFIX wd: <http://www.wikidata.org/entity/>
07
08  <bob#me>
09      a foaf:Person ;
```

[14] http://www.w3.org/TR/sparql11-query/

```
      foaf:knows <alice#me> ;
      schema:birthDate "1990-07-04"^^xsd:date ;
      foaf:topic_interest wd:Q12418 .

wd:Q12418
      dcterms:title "Mona Lisa" ;
      dcterms:creator <http://DBpedia.org/resource/Leonardo_da_Vinci> .

   <http://data.europeana.eu/item/04802/243FA8618938F4117025F
                  17A8B813C5F9AA4D619>
      dcterms:subject wd:Q12418 .
```

Fuente: (Schreiber & Raimond, 2014)

TriG

La sintaxis de la Turtle sólo soporta la especificación de grafos simples sin un medio para "nombrarlos". TriG es una extensión de la Turtle que permite la especificación de múltiples grafos en forma de un conjunto de datos RDF. (Schreiber & Raimond, 2014)

Un documento TriG permite escribir un conjunto de datos RDF en una forma textual compacta. Se consiste de una sucesión de directivas, declaraciones triples, declaraciones de grafos que contienen declaraciones triple-generación y líneas en blanco opcionales. (Bizer & Cyganiak, 2014)

Tabla 4. Ejemplo TriG

```
BASE   <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <http://schema.org/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>

GRAPH <http://example.org/bob>
  {
    <bob#me>
    a foaf:Person ;
    foaf:knows <alice#me> ;
    schema:birthDate "1990-07-04"^^xsd:date ;
    foaf:topic_interest wd:Q12418 .
  }

GRAPH <https://www.wikidata.org/wiki/Special:EntityData/Q12418>
  {
    wd:Q12418
      dcterms:title "Mona Lisa" ;
      dcterms:creator <http://DBpedia.org/resource/Leonardo_da_Vinci> .

<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
        dcterms:subject wd:Q12418 .
  }

<http://example.org/bob>
    dcterms:publisher <http://example.org> ;
    dcterms:rights <http://creativecommons.org/licenses/by/3.0/> .
```

Fuente: (Schreiber & Raimond, 2014)

N-Quads

N-Quads es una simple extensión de N-Triples para permitir el intercambio de RDF Datasets. N-Quads le permite a uno agregar un cuarto elemento a una línea, capturando en la gráfica IRI la tripleta descrito en esa línea. (Schreiber & Raimond, 2014)

Tabla 5. Ejemplo N-Quads

01	<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> <http://example.org/bob> .
02	<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice#me> <http://example.org/bob> .
03	<http://example.org/bob#me> <http://schema.org/birthDate> "1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date> <http://example.org/bob> .
04	<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/topic_interest> <http://www.wikidata.org/entity/Q12418> <http://example.org/bob> .
05	<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/title> "Mona Lisa" <https://www.wikidata.org/wiki/Special:EntityData/Q12418> .
06	<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/creator> <http://DBpedia.org/resource/Leonardo_da_Vinci> <https://www.wikidata.org/wiki/Special:EntityData/Q12418> .
07	<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/terms/subject> <http://www.wikidata.org/entity/Q12418> <https://www.wikidata.org/wiki/Special:EntityData/Q12418> .
08	<http://example.org/bob> <http://purl.org/dc/terms/publisher> <http://example.org> .
09	<http://example.org/bob> <http://purl.org/dc/terms/rights> <http://creativecommons.org/licenses/by/3.0/> .

Fuente: (Schreiber & Raimond, 2014)

JSON-LD (JSON-based RDF syntax);

Proporciona una sintaxis JSON para gráficos y conjuntos de datos RDF. JSON-LD puede ser utilizado para transformar documentos JSON a RDF con cambios mínimos. JSON-LD ofrece identificadores universales de objetos JSON, un mecanismo en el que un documento JSON se puede referir a un objeto descrito en otro documento JSON en otros lugares en la Web, así como el tipo de datos y el lenguaje de manipulación.

Tabla 6. Ejemplo JSON-LD

01	{

```
  "@context": "example-context.json",
  "@id": "http://example.org/bob#me",
  "@type": "Person",
  "birthdate": "1990-07-04",
  "knows": "http://example.org/alice#me",
  "interest": {
    "@id": "http://www.wikidata.org/entity/Q12418",
    "title": "Mona Lisa",
    "subject_of": "http://data.europeana.eu/item/04802/
243FA8618938F4117025F17A8B813C5F9AA4D619",
    "creator": "http://DBpedia.org/resource/Leonardo_da_Vinci"
  }
}
```

Fuente: (Schreiber & Raimond, 2014)

RDFa

Es una sintaxis de RDF que se puede utilizar para insertar datos RDF dentro de los documentos HTML y XML. Esto permite, por ejemplo, a los motores de búsqueda agregar estos datos al rastrear la Web y enriquecer los resultados de búsqueda. (Schreiber & Raimond, 2014)

Tabla 7. Ejemplo RDFa

```
<body prefix="foaf: http://xmlns.com/foaf/0.1/
            schema: http://schema.org/
            dcterms: http://purl.org/dc/terms/">
  <div resource="http://example.org/bob#me" typeof="foaf:Person">
    <p>
      Bob knows <a property="foaf:knows"
href="http://example.org/alice#me">Alice</a>
      and was born on the <time property="schema:birthDate"
datatype="xsd:date">1990-07-04</time>.
    </p>
    <p>
      Bob is interested in <span property="foaf:topic_interest"
      resource="http://www.wikidata.org/entity/Q12418">the Mona Lisa</span>.
    </p>
  </div>
  <div resource="http://www.wikidata.org/entity/Q12418">
    <p>
      The <span property="dcterms:title">Mona Lisa</span> was painted by
      <a property="dcterms:creator"
href="http://DBpedia.org/resource/Leonardo_da_Vinci">Leonardo da Vinci</a>
      and is the subject of the video
      <a
href="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813
C5F9AA4D619">'La Joconde à Washington'</a>.
    </p>
  </div>
```

```
22   <div
     resource="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8
     B813C5F9AA4D619">
23       <link property="dcterms:subject"
     href="http://www.wikidata.org/entity/Q12418"/>
24   </div>
25 </body>
```

Fuente: (Schreiber & Raimond, 2014)

RDF/XML

Proporciona una sintaxis XML para grafos RDF. Cuando RDF fue desarrollado originalmente en la década de 1990, esta fue su única sintaxis, y algunas personas siguen llamando esta sintaxis "RDF". En 2001, se propuso un precursor de la Tortuga llamado "N3", y poco a poco los otros idiomas que aparecen aquí se han adoptado y normalizado. (Schreiber & Raimond, 2014)

Tabla 8. Ejemplo RDF/XML

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <rdf:RDF
03       xmlns:dcterms="http://purl.org/dc/terms/"
04       xmlns:foaf="http://xmlns.com/foaf/0.1/"
05       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
06       xmlns:schema="http://schema.org/">
07   <rdf:Description rdf:about="http://example.org/bob#me">
08     <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
09     <schema:birthDate
     rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1990-07-
     04</schema:birthDate>
10     <foaf:knows rdf:resource="http://example.org/alice#me"/>
11     <foaf:topic_interest
     rdf:resource="http://www.wikidata.org/entity/Q12418"/>
12   </rdf:Description>
13   <rdf:Description rdf:about="http://www.wikidata.org/entity/Q12418">
14     <dcterms:title>Mona Lisa</dcterms:title>
15     <dcterms:creator
     rdf:resource="http://DBpedia.org/resource/Leonardo_da_Vinci"/>
16   </rdf:Description>
17   <rdf:Description
     rdf:about="http://data.europeana.eu/item/04802/243FA8618938F4117025F1
     7A8B813C5F9AA4D619">
18     <dcterms:subject
     rdf:resource="http://www.wikidata.org/entity/Q12418"/>
19   </rdf:Description>
20 </rdf:RDF>
```

Fuente: (Schreiber & Raimond, 2014)

1.3.3. ***SPARQL Query Language for RDF***

SPARQL se puede utilizar para expresar consultas que permiten interrogar diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son definidos mediante vistas RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricciones del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF. (Prud'hommeaux & Seaborne, 2008)

La mayoría de las formas de consulta en SPARQL contienen un conjunto de patrones de tripleta (triple patterns) denominadas patrón de grafo básico. Los patrones de tripleta son similares a las tripletas RDF, excepto que cada sujeto, predicado y objeto puede ser una variable. Un patrón de grafo básico concuerda con un subgrafo de datos RDF cuando los términos RDF (RDF terms) de dicho subgrafo pueden ser sustituidos por las variables y el resultado es un grafo RDF equivalente al subgrafo en cuestión. (Prud'hommeaux & Seaborne, 2008)

A continuación se redacta un ejemplo de una consulta SPARQL, tomado de (Prud'hommeaux & Seaborne, 2008).

Datos:

<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .

Consulta:

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title.
}
```

Resultado de la consulta:

Tabla 9. Resultado consulta SPARQL

title
"SPARQL Tutorial"

Fuente: (Prud'hommeaux & Seaborne, 2008).

1.4. **Acerca de DBpedia**

DBpedia[15] da la siguiente definición sobre si misma: “Es un esfuerzo de la comunidad crowd-sourced[16] para extraer información estructurada de Wikipedia[17] y hacer esta información disponible en la web. DBpedia permite que hacer consultas sofisticadas contra Wikipedia.” El conocimiento extraído de Wikipedia es publicado cumpliendo los estándares de la Web Semántica y las mejores prácticas de Linkend Data.

1.4.1. ***Framework extracción***

Los artículos de Wikipedia consisten sobre todo en texto libre, pero también comprenden diversos tipos de información estructurada en forma de wiki markup[18]. Dicha información incluye plantillas infobox, información de categorización, imágenes geo-coordenadas, enlaces a páginas web externas, páginas de desambiguación, redirecciones entre páginas y vínculos a través de las diferentes ediciones lingüísticas de Wikipedia. El marco de la extracción DBpedia extrae esta información estructurada de Wikipedia y lo convierte en una rica base de conocimientos (Lehmann, y otros, 2012)

En la figura 5 se observa el marco de trabajo necesario para lograr que todo el proceso partiendo de la extracción de información desde Wikipedia hasta poder disponer de ella como datos enlazados.

[15] http://dbpedia.org/About
[16] http://es.wikipedia.org/wiki/Crowdsourcing
[17] http://www.wikipedia.org/
[18] http://en.wikipedia.org/wiki/Help:Wiki_markup

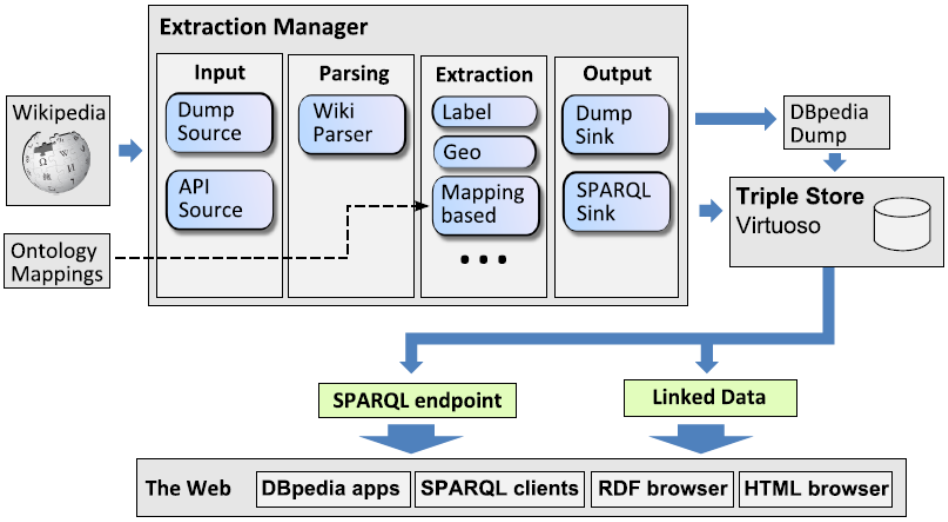


Figura 5: Representación gráfica RDF
Fuente: (Lehmann, y otros, 2012)

1.4.2. *DBpedia Dataset*

DBpedia se trata de una base de conocimiento (en inglés knowledge base) que se encuentra distribuida 119 idiomas que en total describen 12.6 millones de cosas únicas, 24.6 millones de enlaces a imágenes 27.6 millones enlaces a fuentes externas 45 millo es de enlaces a fuentes externas de datos RDF y 67 millones de enlaces a categorías Wikipedia 42.1 millones a categorías YAGO[19].

Lo cual la establece como una fuente muy basta de información sobre cualquier ámbito de conocimiento, esto gracias a al continuo crecimiento de la Wikipedia, su fuente de información. Pero no esto no quiere decir que única base de conocimiento disponible en la web, se encuentran disponibles otras como YAGO.

1.4.3. *Acceso a DBpedia Dataset*

El Dataset de DBpedia se almacena y publica mediante OpenLink Virtuoso. La infraestructura de Virtuoso permite el acceso a los datos RDF de DBpedia a través de SPARQL endpoint, junto al soporte HTTP para cualquier GET estándar de cliente Web para HTML o representación RDF de un recurso DBpedia. (Bizer, Dbpedia, 2009).

[19] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

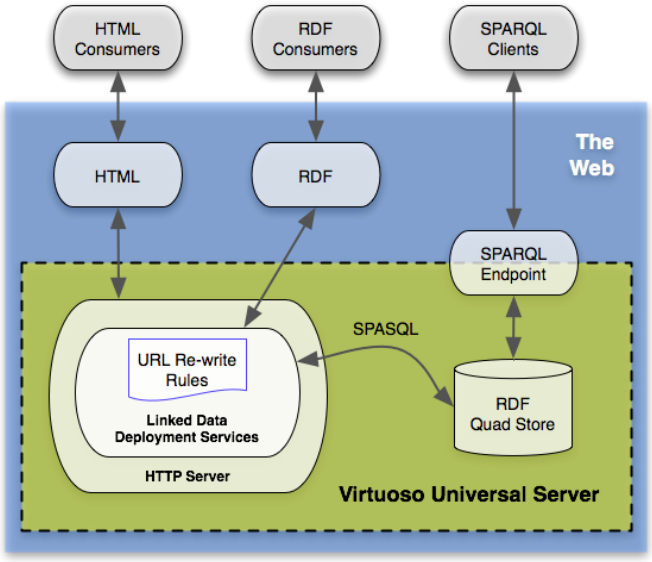


Figura 6. Arquitectura de provisión de Datos de Dbpedai.
Fuente: (Bizer, Dbpedia, 2009)

Se expones dos formas para acceder a la a la Dataset de DBpedia:

1. Querying DBpedia: se puede accederé a través del SPARQL endpoint público http://DBpedia.org/sparql proporcionado por OpenLink Virtuoso. Por este método se puede acceder enviando query Sqparl para hacer consultar sobre Dataset.

2. Linked Data: se refiere a la aplicación de los principios de datos enlazados revisados en 1.1.2. para nombrar y referenciar los recursos dentro de Dataset de DBpedia como por ejemplo: http://DBpedia.org/resource/The_Lord_of_the_Rings

Además de esta opciones se puede descargar el Dataset de DBpedia en diferentes idiomas teniendo en cuenta de que el número de recursos puede cambiar de idioma a idioma puesto que no se trata de una traducción sino de una recopilación de información de Wikipedia la cual se encuentra más extendida en ingles que otros lenguas,

2. Procesamiento de Lenguaje Natural (PLN)

2.1. Introducción

El procesamiento de lenguaje natural se preocupara por entender el lenguaje humano, la comunicación una tarea que para las personas e inclusive animales es tan natural y cotidiana, se vuelve un reto al tratar de interpretarlo mediante procesos computacionales a fin de comprenderlo y poder replicarlo.

La dificultada de la construcción de una aplicación de la ingeniería lingüística variara de acuerdo a objetivo que se persiga, esto explicado por (García, 2005) en donde ejemplifica: “un sistema de generación de cartas personalizado no precisa ningún tratamiento de comprensión, o un sistema de identificación de la lengua (o un detector de errores ortográficos) no necesitan generar lenguaje humano. La mayoría de las aplicaciones incluyen, sin embargo, alguna forma más o menos precisa de comprensión. Así, un sistema de consulta en lenguaje humano a una base de datos precisa un nivel muy alto de comprensión de las expresiones del interlocutor humano para que la respuesta del sistema sea de utilidad. En cambio, en un sistema de traducción o de resumen automáticos se pueden lograr niveles de corrección muy notables con niveles de comprensión bajos. Es decir, no es preciso comprender totalmente una oración para ser capaz de traducirla correctamente.”

2.2. **Part of Speech Tagger**

Permite distinguir la función de una palabra en un determinada contexto mediante la asignación de una etiqueta predefinida. “Una part of speech tagger es un sistema que usa el contexto para asignar parte de un discurso a una palabra”, (Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P., 1992).

Es te etiquetado de palabra ya permite un primera desambiguación en cuanto a la función de la palabra en un sentencia o contexto. Así se puede por ejemplo[20] ver que la palabra “*dado”* que si bien es nombre en singular, también puede ser una forma del verbo dar.

Pero antes de poder etiquetar una palabra por su función es necesaria una Tokenización del texto que va a analizar, que consiste en separarlo en palabras individuales reconociendo un token para palabra o carácter extraído.

2.3. **Chunking**

Text Chunking consiste en dividir un texto en frases de tal manera que palabras sintácticamente relacionadas sean miembros de la misma clase. Estas frases no se superponen es decir que una sola palabra puede ser miembro de un chunk. (Tjong Kim Sang, E. F., & Buchholz, S., 2000)

Este proceso es básico al momento de detectar entidades dentro de un texto, este proceso lo se puede observar en la figura 1 en donde la sentencia, *We saw the yellow*

[20] http://es.wikipedia.org/wiki/Ambig%C3%BCedad

dog, está separada en cuadros en los más pequeños se observa etiquetas de POS Tag y las más grades al nivel de chunking. Una vez la frase ha pasado por el proceso de chunking se puede rescatar dos entidades dentro de la sentencia como *We* y *the yellow dog.*

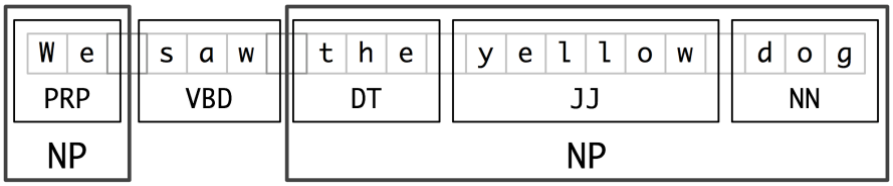


Figura 7: Ejemplo POS Tag y Chunking
Fuente: http://www.nltk.org/book/ch07.html

Las etiquetas utilizadas el idioma inglés se encuentran descritas en el anexo 10.

2.4. **Desambiguación**

La polisemia es un fenómeno muy común, que se refiere a cuando una palabra tiene varios significados, la desambiguación busca descifrar que significado una palabra está siendo utilizado de acuerdo a un contexto en específico, se denomina Desambguacion del sentido de la palabra (Word Sense Disambiguation (WSD), en inglés), este es un problema propio del procesamiento de lenguaje natural (PLN). El descifrar estos distintos significados para los seres humanos es muy común, lo resolvemos de forma cotidiana y pasa casi desapercibida.

La desambiguación de recursos en contexto con Linked Open Data segun (Peláez, Morocho, & Malla, 2012) encontramos “Al realizar la desambiguación estamos consiguiendo un paso clave en la recuperación de la información ya que numerosas palabras cambian de sentido según el contexto en el que nos encontremos trabajando y es sumamente necesario aclarar con el usuario cuál es el sentido que le otorga al término que está utilizando.”

2.4.1. ***Métodos basados en el conociendo.***

Utilizan fuentes de léxicas estructuras existentes para resolver el significado de las palabras, (Tello Leal, 2009) lo define de la siguiente forma: “Estos métodos utilizan un conocimiento lingüístico previamente adquirido. La idea básica consiste en utilizar recursos externos para desambiguar las palabras, tales como diccionarios, tesauros

(vocabularios controlados que representan las relaciones semánticas con otros palabras y sus significados), textos sin ningún tipo de etiquetado e incluso recursos de la Web"

Algunos recursos lingüísticos que se utilizan para la desambiguación de sentido de la palabra en PNL se describen a continuación.

2.4.1.1. WordNet:

"WordNet es una base de datos léxico-conceptual del inglés estructurada en forma de red semántica y construida manualmente. Es el lexicón relacional en formato electrónico más completo y extenso existente, comparable sólo con el diccionario bilingüe para el japonés y el inglés EDR Electronic Dictionary[21]. La unidad básica en que se estructura WordNet es el synset[22], un conjunto de sinónimos representando un concepto." (Mihaela, 2004)

2.4.1.2. EuroWordNet

"El propósito de EuroWordNet ha sido construir una base de datos léxica multilingüe para diferentes lenguas europeas, siguiendo la metodología de WordNet. EuroWordNet es una base de datos multilingüe, con wordnets para varias lenguas (holandés, italiano, español, inglés, alemán, francés, estoniano y checo), compatibles entre sí en cobertura e interpretación de las relaciones." (Mihaela, 2004)

2.4.1.3. Extended WordNet

"Debido a que WordNet ha sido construida como una base de datos léxica, hay limitaciones en su uso para ciertas aplicaciones del procesamiento del conocimiento; por ejemplo, no es posible extraer palabras relacionadas temáticamente. El propósito del proyecto Extended WordNet es transformar WordNet en un formato que permita la derivación de relaciones semánticas y lógicas adicionales." (Mihaela, 2004)

2.4.1.4. MindNet

"MindNet. Una alternativa a WordNet y EuroWordNet es MindNet (Dolan et al., 2000). Aunque tiene un núcleo derivado a partir de diccionarios, la red se construye a base de oraciones nuevas de un corpus que, una vez analizadas, se incorporan a la red. En la visión de sus autores, la dinamicidad y la continua ampliación, permitirán a MindNet perfilarse como un sistema de amplia cobertura." (Mihaela, 2004)

[21] Para detalles, consúltese el sito: http://www.iijnet.or.jp/edr/.
[22] De synonym set 'conjuntos de sinónimos'.

Algoritmo de Lesk

Este algoritmo se basa en diccionarios para resolver la ambigüedad, en (Pérez, 2009) una descripción de la Algoritmo de Lesk 1986, “es uno de los primeros algoritmo desarrollado para la desambiguación semántica de toda las palabras en cualquier texto. El único recurso requerido por el algoritmo es un conjunto de entradas en un diccionario, una por cada posible sentido y conocimiento sobre el contexto inmediato donde se desarrolla la desambiguación”

2.5. **Servicios Web**

2.5.1. ***Introducción***

La W3C[23] (World Wide Web Consortium) encarda de estandarización de las tecnologías en la web aborda este tema de la siguiente forma: “Los servicios web proporcionan un medio estándar de interoperabilidad entre las distintas aplicaciones de software, que se ejecuta en una variedad de plataformas y/o marcos de trabajo. Los servicios Web se caracterizan por su gran interoperabilidad y extensibilidad. Se pueden combinar en una forma de acoplamiento flexible con el fin de lograr operaciones complejas. Programas que prestan servicios simples pueden interactuar entre sí con el fin de ofrecer servicios de valor añadido sofisticados.” Los servicios web permiten la colaboración entre aplicaciones independientemente de la plataforma en las que están desarrolladas, utiliza protocolos y normas estandarizadas en la web, además esto permite la reutilización de código, además de disminuir el coste de integración.

2.5.2. ***Tipos de servicios web***

Dos tipos de servicios web se pueden encontrar de acuerdo con la forma en que se puede implementar abarcado diferentes tecnologías: *RESTful Web Services* y *“Big”*[24] *Web Services* (o también, The “Big” Web services technology stack, debido a la diversas tecnologías en las que se implementar como: SOAP, WSDL, WS-Addressing, WS-ReliableMessaging, WSSecurity, etc), estos dos tipos son expuesto en (Pautasso, Zimmermann, & Leymann, 2008)

[23] http://www.w3.org/
[24] Nombrado así en (Richardson & Ruby, RESTful Web Services, 2007)

2.5.2.1. SOAP AND THE WS- STACK*

Proporcionar interoperabilidad sin fisuras entre los heterogéneos pilas de tecnología de middleware y el fomento de la articulación flexible de servicio al consumidor (solicitante, cliente) y proveedor de servicios son los principales objetivos de diseño de arquitectura orientada a servicios (SOA) conceptos y tecnologías de servicios Web. (Pautasso, Zimmermann, & Leymann, 2008)

En el plano conceptual, un servicio es un componente de software que se proporciona a través de un endpoint[25] accesible en la red. Consumidores de servicios y proveedores usan mensajes para intercambiar solicitudes e información de respuesta en forma de *documentos self-containing*[26] que hacen muy pocas suposiciones sobre las capacidades tecnológicas del receptor. En particular, no hay noción de una referencia de objeto remoto que requeriría un corredor de objeto para gestionar un espacio distribuido dirección de memoria. En el nivel de la tecnología, SOAP es un lenguaje XML que define una arquitectura de mensajes y formatos de mensaje, por lo tanto, proporcionar un protocolo de procesamiento rudimentario. El documento SOAP define un elemento XML de nivel superior llamada sobre, que contiene un encabezado y un cuerpo. El encabezado SOAP es un contenedor de información de infraestructura extensible de capa de mensajes que se puede utilizar para fines de enrutamiento (por ejemplo, hacer frente) y Calidad de Servicio (QoS) de configuración (por ejemplo, las transacciones, la seguridad, la fiabilidad). El cuerpo contiene la carga útil del mensaje. Esquema XML se usa para describir la estructura del mensaje SOAP, por lo que los motores de jabón en los dos puntos finales pueden Marshall y Resolver referencia el contenido del mensaje y la ruta a la aplicación apropiada. (Pautasso, Zimmermann, & Leymann, 2008)

2.5.2.2. REST

Transferencia de estado representacional (REST) se introdujo originalmente como un estilo de arquitectura para la construcción de sistemas hipermedia distribuidos a gran escala. Este estilo arquitectónico es una entidad más abstracta, cuyos principios se han utilizado para explicar la excelente escalabilidad del protocolo HTTP 1.0 y también han limitado el diseño de su siguiente versión, HTTP 1.1. Por lo tanto, el término REST muy a menudo se utiliza junto con HTTP. (Pautasso, Zimmermann, & Leymann, 2008)

El estilo arquitectónico REST se basa en cuatro principios:

[25] http://www.w3.org/TR/ws-gloss/#endpoint
[26] http://www.thefreedictionary.com/self-contained

Identificación de recursos a través de URI. Un servicio web RESTful expone un conjunto de recursos que identifican los objetivos de la interacción con sus clientes. Los recursos son identificados por URI, que proporcionan un espacio de direccionamiento global de los recursos y de descubrimiento de servicios.

Interfaz uniforme. Los recursos son manipulados utilizando un conjunto fijo de cuatro crear, leer, actualizar, eliminar operaciones: PUT, GET, POST y DELETE. PUT crea un nuevo recurso, que puede ser luego borrar con DELETE. GET recupera el estado actual de un recurso en alguna representación. POSTE transfiere un nuevo estado sobre un recurso.

Mensajes de auto-descriptivo. Recursos están desconectados de su representación para que su contenido se puede acceder en una variedad de formatos (por ejemplo, HTML, XML, texto plano, PDF, JPEG, etc.) Metadatos sobre el recurso está disponible y se utiliza, por ejemplo, para controlar el almacenamiento en caché, detectar errores de transmisión, negociar el formato de representación adecuada, y llevar a cabo la autenticación o controlar el acceso. Interacciones con estado a través de hipervínculos. Cada interacción con un recurso no tiene estado, es decir, los mensajes de solicitud son autónomos.

Interacciones con estado se basan en el concepto de transferencia de estado explícito. Existen varias técnicas para el intercambio de estado, por ejemplo, la reescritura de URI, cookies, y los campos de formulario ocultos. Estado puede ser embebido en los mensajes de respuesta para señalar válidos estados futuros de la interacción.

2.5.3. ***Recursos y representaciones***

(Richardson & Amundsen, RESTful Web APIs, 2013) Rest denomina recursos a los datos estructurados que son objetos de las interacciones entre métodos de HTTP, y se dice que cualquier cosa que pueda ser almacenado de un computador puede ser un recurso, como documento electrónico, una fila de una base de datos o el resultado de un algoritmo

No solo las cosas almacenadas en un computador pueden ser llamados recursos también pueden ser recursos artículos tangibles como frutas por ejemplos, y es posible representarlo como recursos a través de la web como por ejemplo como un artículo en de venta o una imagen binaria depende de la aplicación así que por eso decimos sobre las

representaciones que puede ser cualquier documento legible que contenga información acerca de un recurso.

CAPITULO 2: PROBLEMÁTICA

1. Estado actual

La documentación dentro del desarrollo de trabajos educativos universitarios es indispensable para la difusión de los avances realizados en las distintas ramas y disciplinas de las ciencias, estas publicaciones se encuentran en texto plano el cual está diseñado para la fácil compresión por parte del usuario humano que acceda a estos, y contienen datos relevantes dentro de sus líneas los cuales se pierden puesto que no son explotados. Y que a pesar que estos documentos de encontrasen almacenados y publicados de tal forma que sean alcanzados por motores de búsquedas, no se puede acceder por este métodos a los datos relevantes que representa recursos disponibles en la Web a los cuales se requiere enlazar.

Se busca que las fuentes de datos que contengan estos recursos “objetivos”, sean estructuradas y publicadas de acuerdo con los principios de Datos Enlazados y permita acceder a Linking Open Data Cloud.

En la estructura de estos documentos existe resumen inicial en el que se explica el tema que se abarca en la publicación, etiquetado en inglés como “*abstract*” que en español significa “resumen” que es donde se centraran los esfuerzos para descubrir datos, esto a pesar de la existencia otros campos como “keywords” (en español, palabras claves), que exponen los temas que expone los abordan, pero que no son tan descriptivos.

2. Justificación

Teniendo en cuanta la tendencia actual de web, la web semántica que se basa en los principios de datos enlazados, los datos toma un factor importante, por lo cual que estos se encuentren “ocultos” dentro del texto no hace posible su enlace e impide la apertura hacia otras fuentes de información.

En el contexto educativo existen esfuerzos para la publicación de Datos Enlazados, uno de los problemas es las fuentes y sus estructura, como lo exponen (Valverde, Morocho, & Piedra, 2012), “La información que se posee no se encuentra en un formato estructurado, y la encontramos en documentos de tipo pdf, hojas de excel, word, texto plano y medios digitales como DVD. Adicionalmente, se completó información faltante, a través de búsquedas manuales en páginas web de organismos de educación de cada país. Al contar con información en diversos formatos y no estructurada, se dificulta realizar una extracción automática de la información sobre el ámbito que nos interesa, la Legislación de Educación Superior.”

Esto hace necesario medios que permitan extraer y relacionar estos datos dentro de las publicaciones, de acuerdo a los principio de la web semántica que se encuentra en construcción y que iniciativas como esta ayudan a su expansión.

La información escrita es de fácil compresión para las seres humanos, se puede entender palabras por palabra su significado, aun cuando este puede varias de acurdo al contexto en que se encuentre y a la vez modificando en significado de otras palabras.

Dentro de un texto existen palabras que son más representativas que otras al momento de dar sentido a toda una sentencia o frase, esto puede ocurrir debido a que una palabra o varias palabras, más allá de tener un sentido pueden ser representaciones que entidades existentes en mundo real, como: personas, lugares, eventos, organizaciones etc. o representen entidades abstractas como la Web y diferentes tecnologías existentes, en sí, un texto plano como tal pude estar relacionado con diferentes representaciones de entidades del mundo real alojadas en la web por pedio de las palabras.

Pero el sentido de una frase descansa en todas las palabras siendo unas más representativas que otras como ya se ha visto, aunque no necesariamente estas tengas representaciones en la web.

3. Objetivo General.

Implementar servicios web para descubrimiento, enlace y enriquecimiento de Datos Enlazados, aplicado a publicaciones universitarias que permitan el enlace con nuevos datos vinculados a LOD Cloud.

4. Objetivos Específicos

- Crear Base de Conocimiento que sirva como fuente de información para el descubrimiento de datos.
- Desarrollar Servicios Web para Desambiguación, Enlace, Descubrimiento y Enriquecimiento Datos LOD-Cloud
- Implementar Frontal Piloto para Integración de Servicios Web.

CAPITULO 3: SOLUCIÓN

1. Propuesta

Los datos que se encuentran dentro del texto tanto de las publicaciones como de fuentes en general, se encurtan relacionados con otros temas y fuentes de datos, a los que por medio este proyecto se tratara de acceder, extraer, relacionar y enlazar con fuentes de información abierta como lo es DBpedia (que se basa en los principios de los Datos Enlazados), esto permitirá el enriquecimiento del contenido.

Esta propuesta utiliza Servicios Web Rest lo que permite una independencia de la fuente de origen de texto a ser analizado, con la lógica de la aplicación propuesta y esta a su vez devuelve un resultado en que en este caso será formato JSON[27] (JavaScript Object Notation - Notación de Objetos de JavaScript) que es ampliamente conocido y utilizado para el intercambio de datos, hacia el cliente que consume el servicio.

La lógica que se propone es explicada en la Figura 10, en donde las interacciones inician con el ingreso del texto a ser analizado, en el cual se aplican las diferentes tecnologías revisadas en capítulos 1 de este documento, para obtener como resultado entidades y keywords estructurados en formato JSON, que estarán desambiguadas y enlazadas, de existir un recurso al cual corresponda dentro del Dataset de DBpedia, es decir que no todos las entidades que se encuentren dentro del texto de una publicación es referenciado en DBpedia.

Puesto que los esfuerzos se concentraran en los *abstracts* de las publicaciones y que esto se redactados en idioma ingles a pesar de que se trate de una publicación en español, se limitara el desarrollo de la soluciona este idioma.

Esta propuesta se surge como solución para el descubrimiento de datos en el texto de los *abstracts* de las publicaciones universitarias, pero debido a la gran cantidad de recursos de diversos temas que se encuentran actualmente disponibles en DBpedia, se puede aplicar a cualquier texto (en idioma ingles) para esto se implementa un interfaz gráfica web donde el usuario puede insertar su texto y ver el resultado.

2. Metodología

Para el desarrollo del proyecto de software se propone una metodología de desarrollo en prototipos, con el fin que en el desarrollo del producto de software adicionar

[27] http://json.org/json-es.html

funcionalidades al sistema de forma incremental, los cuales se dispondrán en forma de Servicios Web de acuerdo con los objetivos del proyecto.

Por lo cual se decide por el "proceso" de desarrollo ICONIX, que está entre la complejidad de RUP (Rational Unified Processes) y la simplicidad y pragmatismo de XP (Extreme programming), sin eliminar las tareas de análisis y de diseño que XP no completa. (San Martin Oliva)

2.1. **Fases de desarrollo**

Se establece las siguientes fases de desarrollo de acuerdo con metodología ICONIX, con las respectivas tareas y resultados esperados de las iteraciones de estas fases.

Tabla 10: Fases de desarrollo del proyecto

Fase	Tarea	Resultado
Análisis de requerimientos	▪ Identificar objetos del mundo real que interviene en el proceso ▪ Identificar actores involucrados ▪ Identificar casos de uso del sistema en interacciones con actores identificados	▪ Requerimientos ▪ Modelo de dominio ▪ Modelo de caso de uso
Diseño preliminar	▪ Describir los casos de uso como un flujo de acciones ▪ Verificar el diseño	▪ Especificación de casos de uso
Diseño	▪ Especificar comportamiento a través de diagrama de secuencias ▪ Verificar el diseño	▪ Diagrama de secuencias
Implementación	▪ Escribir código ▪ Realizar pruebas	▪ Código ▪ Pruebas

Fuente (propio)

3. Desarrollo

3.1. Análisis de requerimientos

A partir de las primeras reuniones se determina los requerimientos funcionales con los debe contar la propuesta de software a desarrollar y se inicia con la formalización de estos requerimientos y luego con su necesario análisis.

3.1.1. *Requerimientos*

Los requerimientos descritos a continuación definen el comportamiento del sistema para obtener los resultados esperados.[28] Estos requerimientos forman la base del desarrollo y se complementa con requerimientos no funcionales que los cuales serán implementados con la finalidad de agregar calidad al producto frente al usuario que lo va a utilizar.

Tabla 11: Resumen de requerimientos funcionales

Código	Requerimiento	Descripción
REQ001	Extraer entidades y palabra relevantes	Descubrir datos relevantes en el texto, a quien se describe y las palabra relevantes que lo acompañan
REQ002	Enlazar entidades y palabra relevantes con LOD Cloud	Se enlazara los términos encontrados en caso de que sea posible con la LOD Cloud
REQ003	Desambiguar entidades y palabra relevantes	Se determinara el sentido con que las palabras estas siendo usadas en caso de que estas sean ambiguas.
REQ004	Levantar servicios REST separados para los procesos relevantes.	Para que los procesos relevantes dentro del sistema pueda ser consumidos de forma individual y así reutilizados se levantaran servicios separados
REQ005	Frontal UI Web	Construir una interfaz web que permita visualizar el comportamiento del sistema, es decir, la integración de los servicios y su funcionamiento individual

[28] Ver en anexos documento de especificación de requerimientos

Fuente: (propio)

Los requerimientos redactados establecen las tecnologías necesarias para el desarrollo del producto de software así como las áreas de conocimiento con las que esta ligados siendo el procesamiento de lenguaje natural (PLN) uno de los puntos más fuertes a resolver junto el levantamiento de servicios web y la construcción de un cliente web.

3.1.2. ***Modelo de Dominio***

Uno de los ámbitos más importantes a resolver para el desarrollo de este sistema es el tratamiento de texto que es enviado por un usuario a través de un cliente, que será la entrada y base del procesamiento para descubrir datos relevantes dentro de este.

Se ha procedido a separar en servicios web distintos los procesos de relevancia del producto de software, en vista de que este uno de los requerimientos (REQ004), los servicios web que serán levantados perteneces a fases importantes dentro de las técnicas de procesamiento de lenguaje natural (PLN) aplicadas al texto, así como funciones dadas por los requerimientos, lo cual se observa en la figura 8.

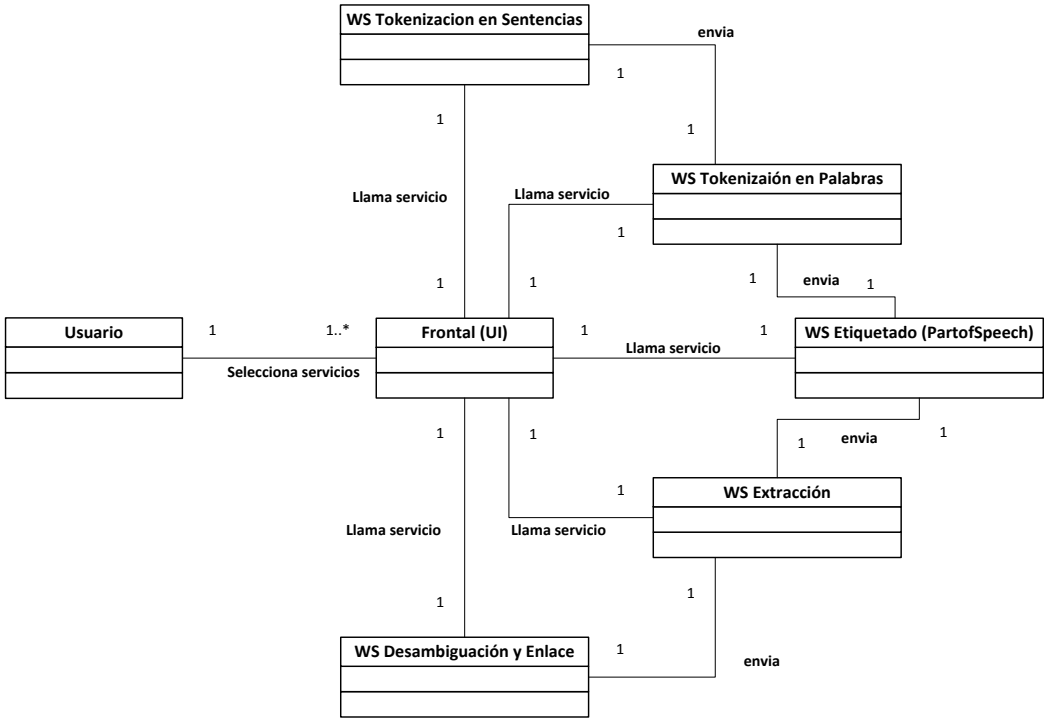


Figura 8: Modelo de dominio
Fuente: (propio)

3.1.3. ***Modelo de caso de Uso***

El comportamiento de los objetos del mundo real, los componentes del sistema a desarrollar y las interacciones que entre estos se puedes realizar son dados por los casos de uso aplicables al sistema dadas las funciones que incorporara basados en el análisis de los requisitos, estas interacciones son visualizados en figura 9.

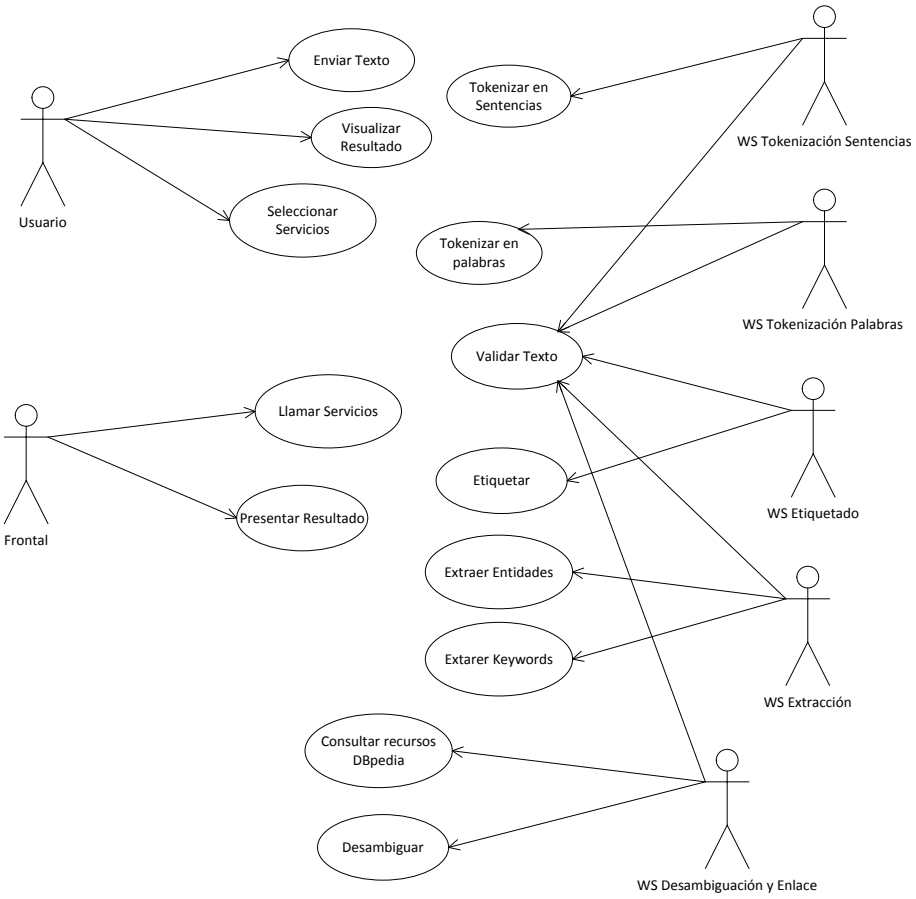


Figura 9: Modelo de casos de uso
Fuente: (propio)

3.2. **Análisis y diseño preliminar**

3.2.1. ***Especificación de casos de uso***

Una vez determinadas los casos de uso que resolverá el software se presenta a continuación es especificaciones de cada uno.

3.2.1.1. Tokenización en Sentencias

Tabla 12: Requerimiento de tokenización de sentencias

Número	**ECS-01**	
Nombre	Tokenización en Sentencias	
Actores	Usuario, Cliente	
Descripción	Divide el texto de entrada en sentencias cortas separadas por un punto y parte, la salida es una lista de estas sentencias.	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado ▪ Texto segmentado en sentencias	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web. **SA1**
	2	Verifica el número de sentencias que comenten al texto, que estén separadas por un punto seguido (.)
	3	Divide cada una teniendo en cuenta la terminación con punto (.) estructura las sentencias dentro de una lista. **SA1**
	4	Estructura la lista de elementos formato JSON.
	5	Devuelve el JSON resultante.
Poscondición	▪ El texto dividido en sentencias.	
Secuencia alternativo	**SA1 el número de sentencias es 1** Se estructura una lista de un solo elemento con la sentencia.	
Prioridad	Media	
Requerimientos Especiales	Idioma de texto ingles	
Asunciones y Dependencias		
Notas adicionales		

Fuente: (propio)

3.2.1.2. Tokenización en palabras

Tabla 13. Requerimiento de tokenización en palabras

Número	**ECS-02**	
Nombre	Tokenización en palabras	
Actores	Cliente, Servicio Web	
Descripción	Divide cada sentencia en palabras validas, tokens.	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Segmentación del texto en sentencias. **ECS-01**
	3	Se recorre la lista de sentencias segmentadas.
	4	Se divide palabra por palabra de la sentencia en una lista, se obtiene una lista de listas.
	5	Se estructura en formato JSON
	6	Retorna el JSON con las sentencias divididas en “tokens”
Poscondición	▪ Texto tokenizado por sentencias y estos a la vez tokenizados en palabras	
Secuencia alternativo		
Prioridad	Baja	
Requerimientos Especiales	Del funcionamiento del Servicio web de Tokenización en Sentencias	
Asunciones y Dependencias		
Notas adicionales		

Fuente: (propio)

3.2.1.3. Etiquetado de palabra

Tabla 14: Requerimiento de etiquetado

Número	**ECS-03**	
Nombre	Etiquetado	
Actores	Cliente, Servicio Web	
Descripción	Este servicio permite la tokenización de cada palabra y etiquetación de las mismas de acuerdo a la función que cumplen en el contexto que se encuentra, para hacerlo se apoya en el servicio web de tokenización en sentencias	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Tokenización del texto en sentencias. **ECS-01**
	3	Recorrido de la lista de sentencias
	4	Etiquetado de las palabras que conforman cada sentencia
	5	Estructura y retorna data en JSON
Poscondición	▪ Texto tokenizado a nivel de palabras y etiquetado.	
Secuencia alternativo		
Prioridad	Alta	
Requerimientos Especiales		
Asunciones y Dependencias		

Notas adicionales	Depende del funcionamiento del servicio web de Etiquetado en Sentencias (ECS-01)

Fuente:(propio)

3.2.1.4. Extracción de Entidades

Tabla 15. Requerimiento de extracción de entidades.

Número	**ECS-04**	
Nombre	Extracción de Entidades	
Actores	Cliente, Servicio Web	
Descripción	Permite reconocer y extraer, las entidades y palabras relevantes o claves (keywords) que se encuentran dentro del texto, para lograr se apoya en el servicio web de Etiquetado (y en los que este a su vez , servicio web de tokenización en sentencias)	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Tokenización del texto en sentencias. **ECS-01**
	3	Tokenización y Etiquetado de palabra **ECS-03**
	4	Reconocimiento de estructuras de Entidades y Keywords
	5	Extracción de Entidades y Keywords
	6	Estructuración de retorno de resultado en formato JSON
Poscondición	▪ Entidades y palabra importantes que las acompañan reconocidos y extraídos	

Secuencia alternativo	
Prioridad	Alta
Requerimientos Especiales	
Asunciones y Dependencias	
Notas adicionales	Este servicio depende del funcionamiento del servicio web de Tokenización en Entidades (ECS-01) y Servicio web de Etiquetado (ECS-03)

Fuente: (propio)

3.2.1.5. Desambiguación y Enlace

Tabla 16. Especificación del requerimiento de desambiguación y enlace

Número	**ECS-05**	
Nombre	Desambiguación y Enlace	
Actores	Cliente, Servicio Web	
Descripción	Enlaza las entidades y palabras relevantes (keywords) hacia LOD Cloud, más específicamente DBpedia, esto de existir un recurso al cual vincular, en caso de que una entidad o keyword tuviese más de uno posible recurso al cual enlazar, se realizara un proceso de desambiguación y luego de enlace.	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Tokenización del texto en sentencias. **ECS-01**
	3	Tokenización y Etiquetado de palabra. **ECS-03**

	4	Extracción de Entidades y keywords. **ECS-04**
	5	Consulta de recursos a DBpedia.
	6	Consulta de “Abstract” de recurso a DBpedia
	7	Verificar si existen Entidades o keywords ambiguas
	8	Desambiguar Entidades y keywords ambiguos. **SA1**
	9	Estructurara resultado
	10	Retornar resultado
Poscondición	▪ Entidades y palabra importantes que las acompañan reconocidos y extraídos	
Secuencia alternativo	**SA1 Entidades y keywords no ambiguos** Se enlaza con los recursos únicos encontrados a las entidades y keywords del texto.	
Prioridad	Alta	
Requerimientos Especiales		
Notas adicionales	Este servicio depende de los servicios web de tokenización en sentencias (ECS-01), etiquetado (ECS-03), extracción de entidades (ECS-04).	

Fuente: (propio)

3.3. **Diseño**

3.3.1. ***Arquitectura***

Después de haber realizado el análisis y posterior investigación sobre los puntos más relevantes dentro del desarrollo, se propone la siguiente propuesta lógica para el funcionamiento de los componentes a desarrollar y en base a los requerimientos funcionales, casos uso a los que va a responder el software y tecnologías disponibles que se pueden implementar para la resolución de la problemática planteada.

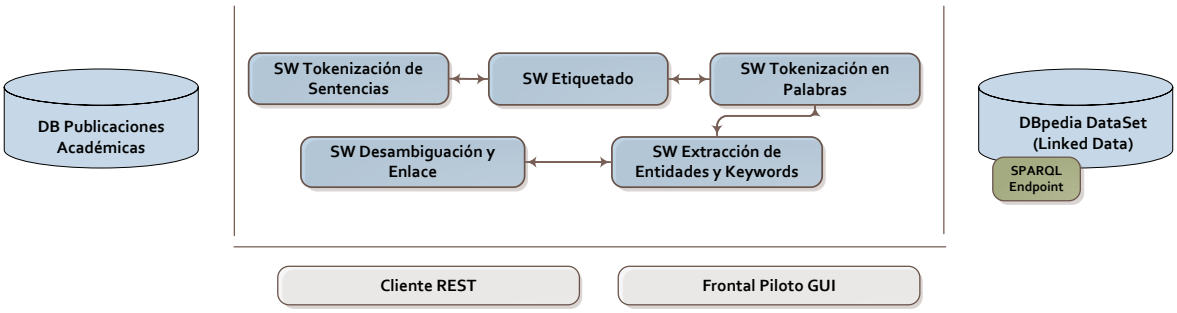


Figura 10. Lógica propuesta para la Aplicación
Fuente: (Propio)

La distribución de los compones implementados se resume en la gráfica 10, haciendo una división de los componentes de la aplicación que interactúan para dar solución a la problemática. Las capas que componen la arquitectura de aplicación se detallan a continuación.

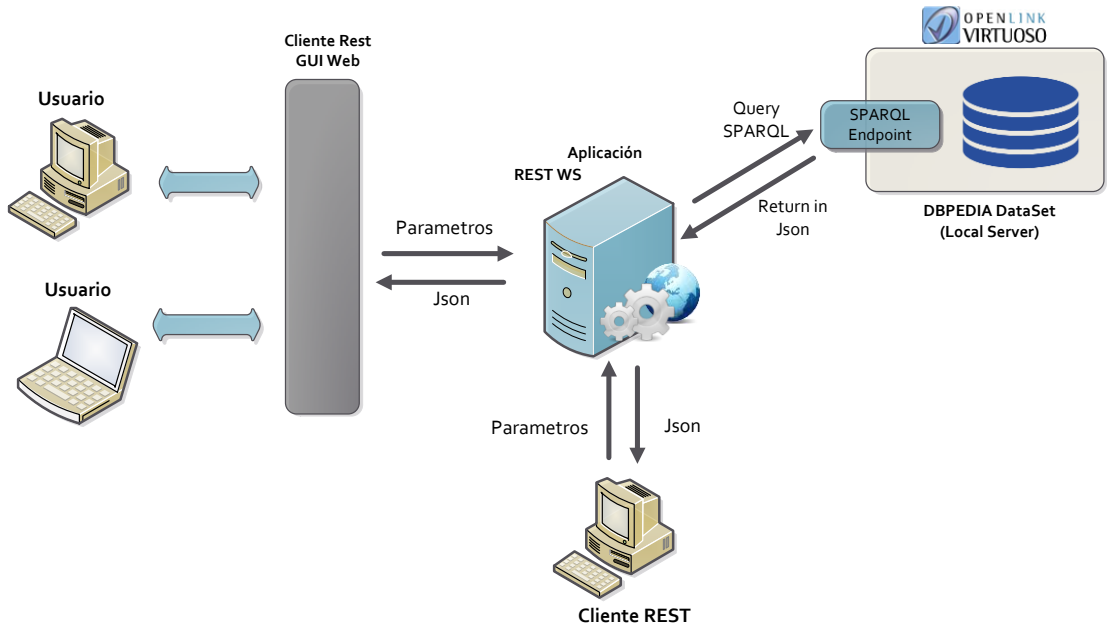


Figura 11. Arquitectura.
Fuente: (Propio)

3.3.2. ***Componentes***

3.3.2.1. Servidor

La lógica de la aplicaciones se la desarrollará en el Lenguaje de alto nivel Python versión 2.7 la elección se ha basado en las familiaridad con la librería, *Natural Language Toolkit* (NLTK), que permite realizar procesamiento de leguaje natural (PNL) que es una parte fundamental a resolver y de igual forma cuenta con las librerías para satisfacer los

requerimientos con los que debe cumplir el software. Las funcionalidades destacadas dentro del desarrollo se resumen a continuación.

Validar texto

Que el texto sea legible para el sistema en funciones siguientes a fin de evitar errores. Comprobar que el texto contenga caracteres a fin de evitar trabajar sobre texto vacío. Es origen obligatorio por el cual los todos los módulos deben pasar para la comprobación de los requisitos del texto que ingresa como parámetro para ser procesado.

Tokenización en sentencias

Dividir el texto ingresado en sentencias u oraciones, generalmente separados por un punto (.) significando el final de esta. Este componente forma parte de los procesos relevantes dentro de procesamiento de lenguaje natural siendo el inicio de estos procesos.

Tokenización en palabras

Una vez divididos el texto en sentencias, se realiza en mismo proceso para las palabras que lo conforman así como los signos de computación, obteniendo un “token” por cada palabra o carácter reconocido. Parte del procesamiento de lenguaje natural.

Etiquetado

Este componente realiza etiquetado de las palabras, de acuerdo al contexto dentro de la sentencia en la que se encuentra, tomando una función específica como verbo, sustantivo, etc. la etiquetación de palabras (Part of Speech, por su nombre en inglés) forma parte del procesamiento de lenguaje natural.

Extracción de entidades y palabras claves

Permite el reconociendo y extracción de entidades y palabra claves dentro del contexto de una oración, para lo cual realiza un reconocimiento de los etiquetas de las palabras y un análisis de su estructura para determinar que palabra es una entidad o una palabra relevante, este análisis puede extraer una palabra o un conjunto de estas formando una entidad o palabras claves. Se basa en componentes posteriores para poder realizar sus operaciones, es decir, necesita de un texto que haya sido tokenizado y etiquetado.

Extracción recursos DBpedia

Una vez que los entidades y palabra claves hayan sido extraídos de las sentencias. Se realizan consultas con estos términos hacia el servidor local de DBpedia, para encontrar los recursos que se denominen igual que estos términos, así como las descripciones rápidas de estos recursos.

Desambiguación de recursos

A través del algoritmo de **Lesk** se analiza el contexto del término y las descripciones de los recursos de DBpedia, para determinar que cual de estos concuerda mejor con el uso que se le está dando al término en la sentencia.

Servicios web

De conformidad con los requerimientos funcionales se levantan servicios diferentes para algunos de los proceso relevantes, permitiendo que puedan ser consumidos y analizados de forma individual. Los servicios levantados serán:

- Tokenización en sentencias
- Tokenización en palabras
- Etiquetado
- Extracción de entidades
- Desambiguación y enlace

Existe una fuerte dependencia entre los servicios, puesto que la salida de unos se convierte en la entrada de otros en forma de secuencia a través de los servicios, el servicio de Tokenización en sentencias, después de proceso de validación, es el primeo en trabajas sobre el texto hasta llegar al servicios de Desambiguación y enlace. La gráfica 12, presenta la forma en que los servicios se relacionan e interactúan entre sí.

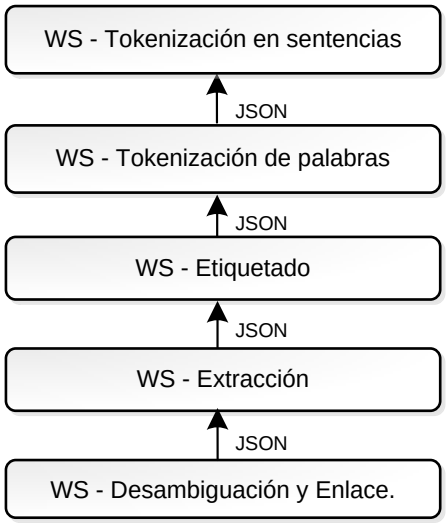


Figura 12. Dependencia de servicios web
Fuente: (propio)

3.3.2.2. Servidor Dataset DBpedia Local (SPARQL endpoint)

Parte de la propuesta consiste en acceder a los recursos de DBpedia por lo cual se implementará un servidor local con los datos, de estos recursos, necesarios para desarrollar la propuesta y así evitar cualquier fallo por problemas de conexión recurrentes al tratar de consultar directamente con su servidor, esto es posible gracias a que DBpedia misma proporciona los medios para descargar sus recursos.

3.3.2.3. Cliente

La vista es un Interfaz Web que permite al usuario la facilidad de la comunicación entre la aplicación y el usuario. Instruye al usuario sobre el uso de la herramienta, identificando con facilidad los parámetros necesarios y en especial la facilidad de la presentación de los resultados. Permite la integración de los compontes del sistema en un entorno amigable para el usuario.

Prototipo de interfaz

3.3.3. ***Diagrama de secuencia***

Al observar el modelado de las interacciones entre los componentes del sistema se evidencia una clara dependencia e interoperabilidad entre los estos, iniciando siempre con el módulo validación del texto a ser procesado.

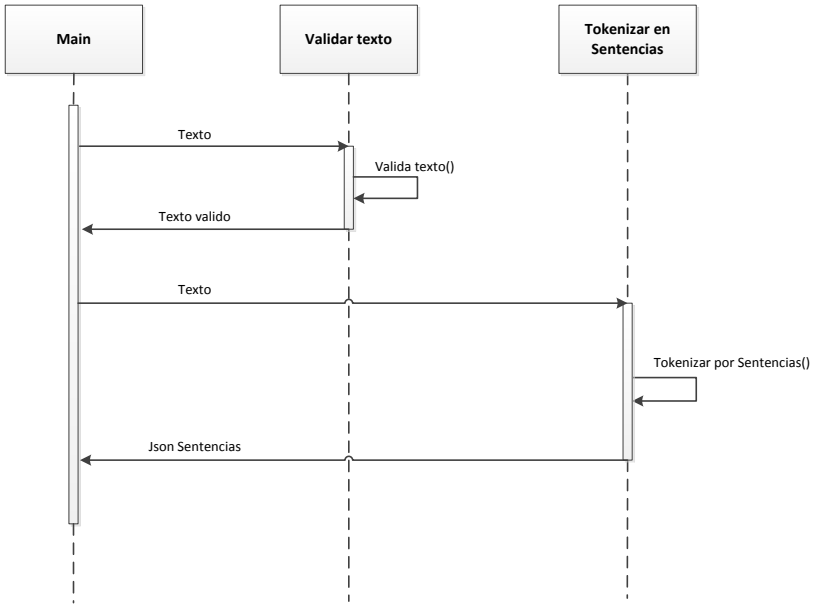


Figura 13. diangrama de secuencias de tokenización de sentencias
Fuente: (propio)

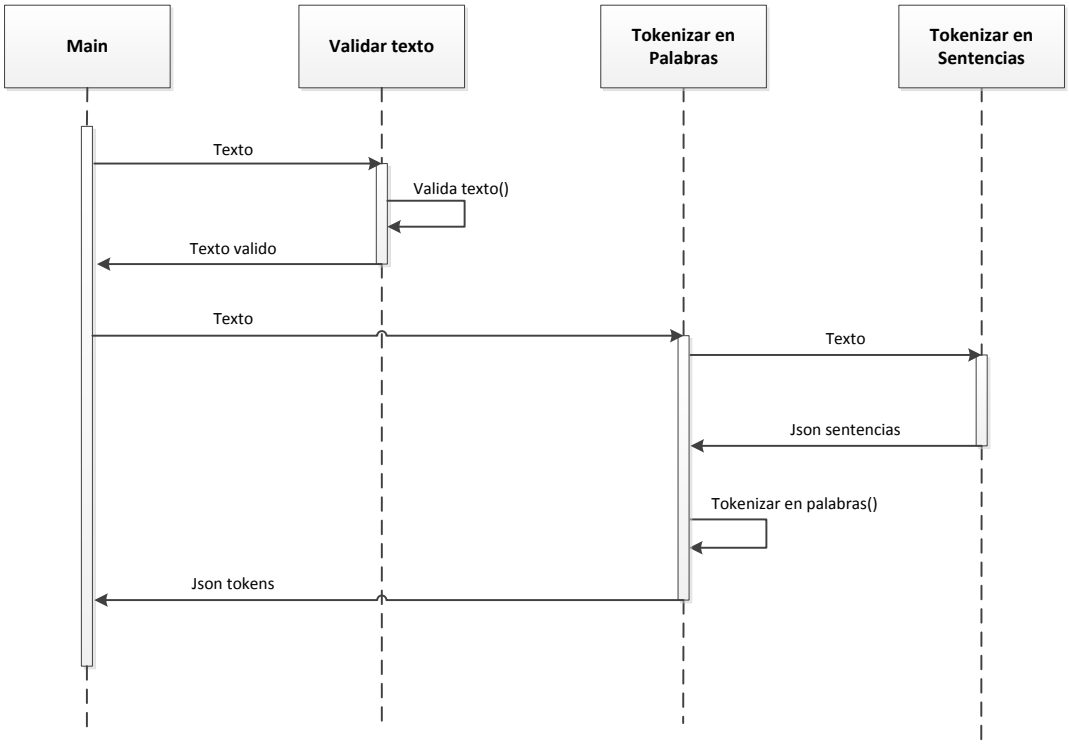


Figura 14. Diagrama de secuencias de tokenización en palabras
Fuente: (propio)

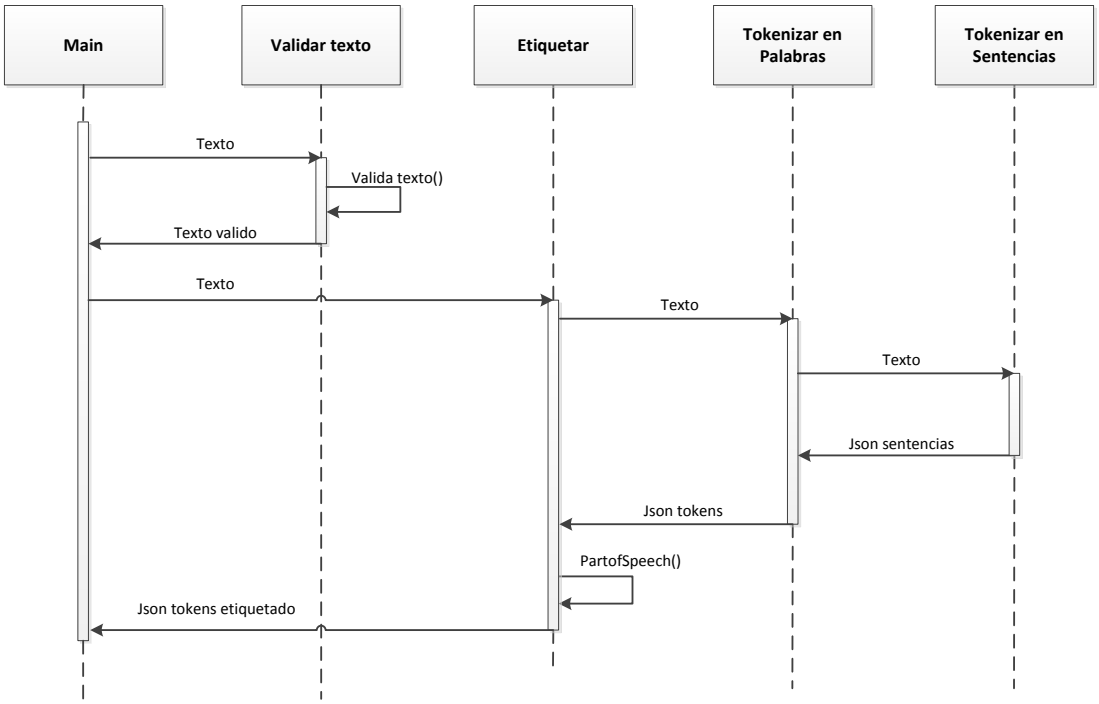


Figura 15. Diagrama de secuencias de etiquetado de palabra
Fuente: (propio)

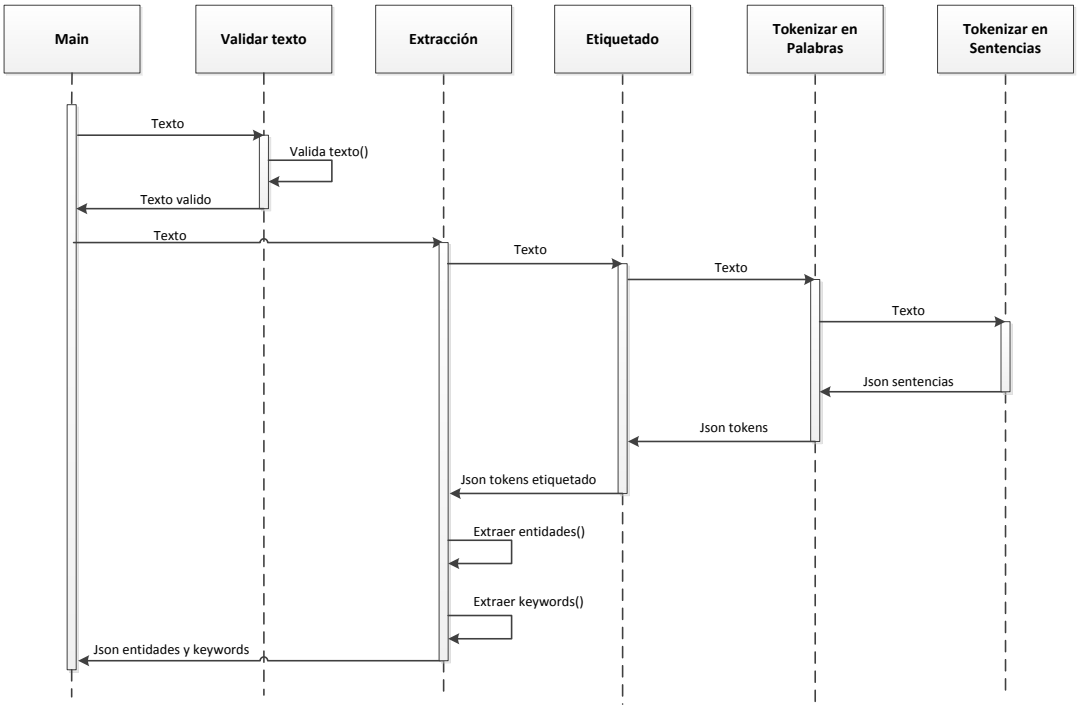


Figura 16. Diagrama de secuencias de extracción
Fuente: (propio)

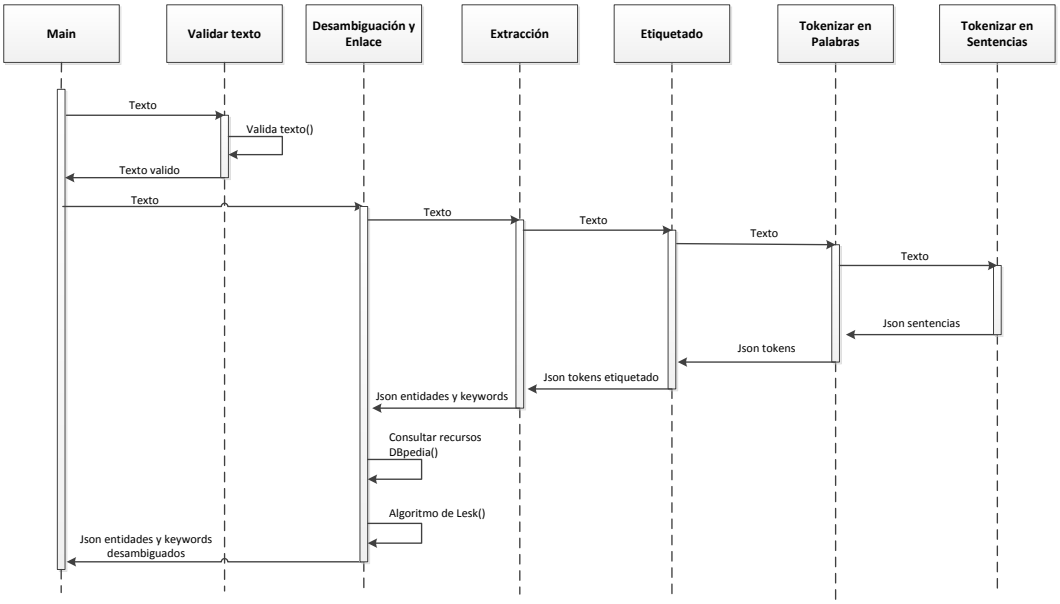


Figura 17. Diagrama de secuencia de desambiguación y enlace
Fuente: (propio)

3.4. **Implementación**

3.4.1. ***Servidor***

Producto del análisis realizado en el diseño de la software, Python es el lenguaje de programación en el que se desarrollan los componentes de la lógica de la aplicación, gracias a las facilidades que proporciona la librería NLTK en el procesamiento del lenguaje natural, pero debido a que esta herramienta presenta limitaciones al trabajar con idiomas distinto al inglés, las tares relacionada con POS-Tagging se las realiza con TreeTagger permitiendo la posibilidad de trabajos futuros relacionados con otros idiomas distintos a inglés.

El código generado para el desarrollo del servidor del producto de software se encuentra en anexos.

Para extraer todos los recursos de DBpedia que a través de uso de sus atributos sea referenciado como el término que se ha encontrado del texto se realiza la consulta SPARQL capturada en la figura 18.

```
SELECT distinct ?x ?amb ?redir ?amb1
WHERE {
    ?x ?predicado '"""+term+"""'@en.
    OPTIONAL { ?x dbpedia-owl:wikiPageDisambiguates ?amb }
    OPTIONAL { ?x dbpedia-owl:wikiPageRedirects ?redir }
    OPTIONAL { ?redir dbpedia-owl:wikiPageDisambiguates ?amb1 }
}
```

Figura 18. Captura de consulta de recursos de DBpedia
Fuente: (propio)

De los recursos de DBpedia resultantes, se requiere descripciones de estos para los procesos de desambiguación, para esto a todos los recursos se les realiza la consulta de la figura 19.

```
select ?abstract
where
{
    <"""+uri+"""> <http://www.w3.org/2000/01/rdf-schema#comment> ?abstract.
    FILTER (lang (?abstract)="en")
}
```

Figura 19. Consulta de abstracts de recursos de DBpedia
Fuente: (propio)

Para determinar el tipo de recurso según la ontología de DBpedia que se ha encontrado se realiza el siguiente consulta:

```
select distinct *
where {
    <"""+term+"""> rdf:type ?typesself.
}
```

Figura 20. Consulta para extraer el tipo de recurso
Fuente: (propio)

3.4.1.1. Servicios

Los servicios levantados en base a los requerimientos funcionales del producto de software, corresponden a las funcionalidades del sistema disponibles por separado pero integrados entre sí para satisfacer las solicitudes por parte del usuario, quien va a poder elegir a que servicio quiere acceder.

Para la construcción de los servicios web se siguieron los principios de arquitectura REST, separando de esta forma las operaciones del servidor con las del cliente, los cuales se podrían construir utilizando tecnologías distintas.

La respuesta de los servicios se encuentra en formato JSON, la estructura del objeto cambia de acuerdo al servicio llamado, mediante un ejemplo en la figura 21 se muestra el resultado del servicio ***Tokenización en Sentencias*** mientras que le figura 22, el resultado de servicio ***Desambiguación y Enlace,*** en la figura 12 se da una idea clara por qué la estructura del objeto JSON aumenta notablemente conforme los servicios que intervienen para dar respuesta a la solicitud del usuario.

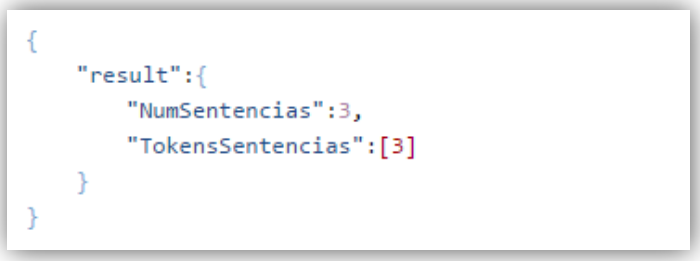


Figura 21. Resultado de servicio de tokenizacion en sentencias
Fuente: (propio)

Figura 22. Resultado del servicio web de desambiguación y enlace
Fuente: (propio)

Cada servicio adiciona una nueva propiedad al resultado, en la tabla 17 se presenta una descripción de las propiedades que cada servicio suma al JSON resultante.

Tabla 17. Propiedades del JSON resultado de los servicios web

Servicios Web	Propiedad adicionada	Descripción propiedad
Tokenización en Sentencias	"TokensSentencias"	Contiene en una lista las sentencias en las que divide en texto.
	"NumSentencias"	En número de sentencias encontradas
Tokenización en Palabras	"TokensPalabras"	Las palabras y signos de puntuación que compones en texto, denominado tokens
	"NumTokensPalabras"	La cantidad de tokens encontradas
Etiquetado	"EtiquetadoPalabras"	Los token con las etiquetas de acuerdo a la función que realizan en la sentencias, del mismo número de “NumTokensPalabras”
Extracción	"NumKeywordsSimples"	La cantidad de palabras claves extraídas del texto
	"NumKeywordsCompuestas	El número de palabras compuestas
	"KeywordsCompuestas"	Contiene las palabras claves que se descubrieron en el texto.
	"KeywordsSimples"	Las palabras claves simples del texto.
	"Entidades"	Las entidades extraídas del texto.
	"NumEntidades"	La cantidad de entidades extraídas.
Desambiguación y Enlace	"EntidadesDesambiguadas"	Los términos enlazados a DBpedia
	"NumEntidadesDesambiguadas"	Numero de términos Enlazados

Fuente: (Propio)

3.4.2. ***Servidor Dataset DBpedia Local***

Archivos necesarios para la implantación de un repositorio local de recursos DBpedia para funcionalidad de desambiguación y enlace.

- Label de recursos:
    - labels_en.nt.bz2
- Datos personales de los recursos tipos Persona:
    - persondata_en.nt.bz2
- Resúmenes Corto de los recursos
    - short_abstracts_en.nt.bz2
- Links de Desambiguación de Wikipedia
    - disambiguations_en.nt.bz2
- Redirecciones entre Recursos
    - redirects_en.nt.bz2

El tamaño total de las importaciones es de 3.8 GB, de espacio en disco.

3.4.3. ***Cliente web***

Presenta el resultado transparente para el usuario, cumple con el requerimiento de integrar los servicios levantados, además permite la selección de los servicios a los que se desea acceder de forma individual, respetando las dependencias que establecidas entre ellos para su funcionamiento.

Desarrollado en HTML, JavaScript y CSS interpreta el JSON recibido por parte de los servicios o el servicio invocado y procesa para que sea agradable al usuario además permite visualizar el JSON tal como se lo recibe desde el servicio, esto para usuarios interesados puedan analizar el resultado. Para introducir el texto a ser procesado se dispone a de un área donde colocarlo visible y amigable para el usuario.

Para seleccionar el servicio al que se requiere acceder existe un menú que los expone como fucnionalidades del sistema, en este menú se encuntran:

- Segementación en sentencias
- Tokenización
- Etiquetado
- Extración de Etidades
- Desambiguación y Enlace

Las cuales puede ser seleccionados o deseleccionados por el usuario, una captura de la interfaz se puede visualizar en la figura 23.

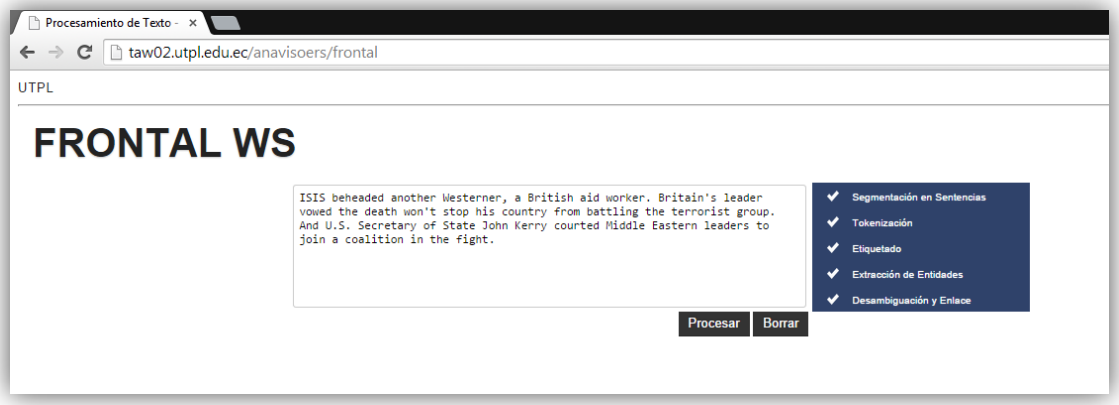


Figura 23. Captura de la interfaz es su estado inicial.
Fuente: (propio)

Cada uno de los ítems del menú de funcionalidad accede a un servicio diferente, pero en vista de que los servicios dependen entre ellos para su funcionamiento, cuando un usuario seleccionara un servicio se seleccionaran automáticamente las funcionalidad que acceden a los servicios de los cuales depende para realizar su funcionamiento, por ejemplo, en determinado momento todos las funcionalidades deseleccionadas y el usuario decide seleccionar la funcionalidad de ***Etiquetado*** (este caso esta capturado en la figura 24), en este caso de forma automática se seleccionará las funciones de ***Segmentación en Sentencias*** y ***Tokenización*** que acceden a los servicios necesario para poder realizar al etiquetado (el resultado de esta interacción se presenta en la figura 25).

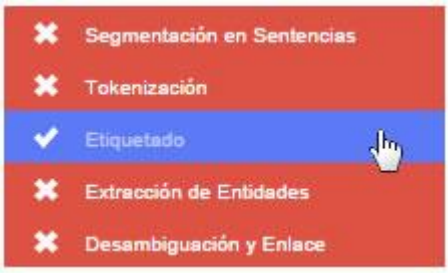


Figura 24. Momento previo a la selección de la funcionalidad de etiquetado
Fuente: (propio)

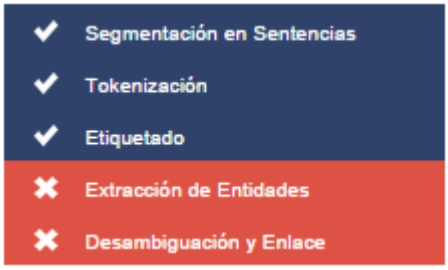


Figura 25. Función etiquetado seleccionado junto a las funcionalidades dependientes
Fuente: (propio)

De igual manera al deseleccionar un servicio al cual no se desea acceder se deseleccionaran las funcionalidades que no son necesarios para la resolución de esta petición, un ejemplo de esto lo se puede observar en la figura 26 donde se tiene todos los servicios seleccionados y se decide deseleccionar la funcionalidad de Tokenización y el resultado se observa en la figura 27.

Figura 26. Momento previo a la deselección de la funcionalidad de tokenización
Fuente: (propio)

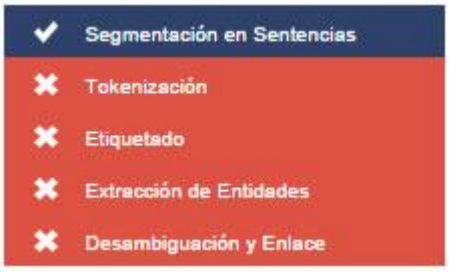


Figura 27. Función de tokenización deseleccionada junto las funciones que depende de esta.
Fuente: (propio)

Para iniciar el procesamiento se envía el texto base al presionar el botón ***Procesar*** disponible en la interfaz y se espera mientras se devuelve y procesa una respuesta. Las respuesta es interpretado y se coloca en par te inferior de la interfaz, esto se puede visualizar en la figura 28.

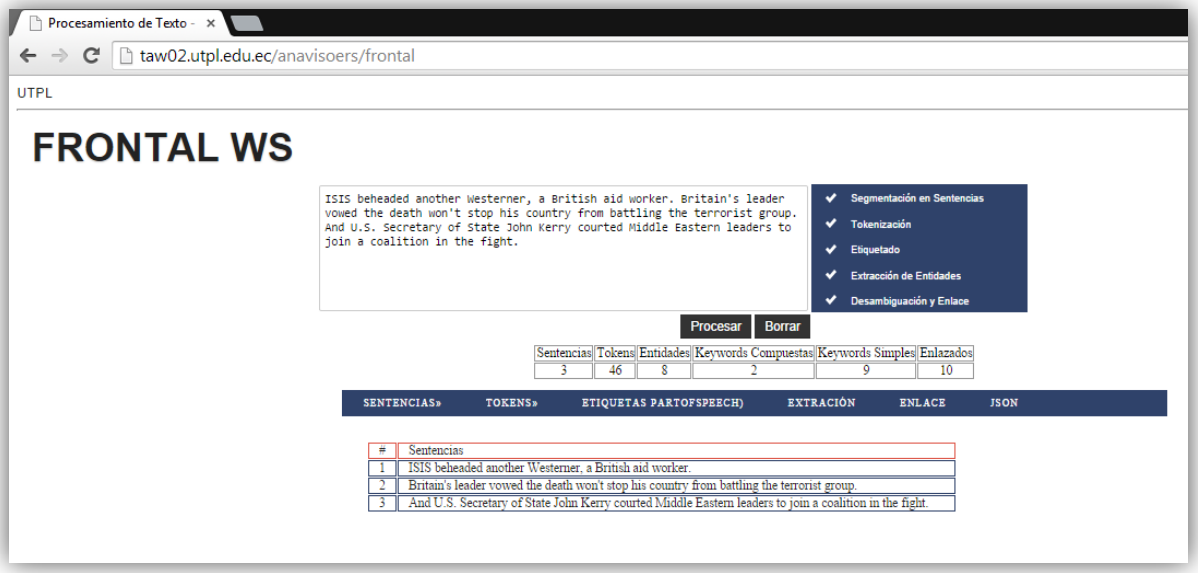
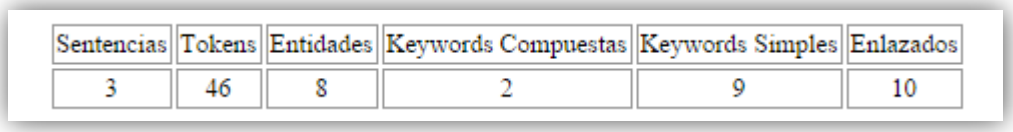


Figura 28. resultado del procesamiento del texto

Fuente: (propio)

En forma de resumen un conteo de lo encontrado en el texto mediante los procesos realizados sobre el texto y dependiendo de los servicios invocados se presenta en forma de tabla, la cual se encuentra capturada en la figura 29.



Sentencias	Tokens	Entidades	Keywords Compuestas	Keywords Simples	Enlazados
3	46	8	2	9	10

Figura 29. Captura de la tabla con datos cuantitativos de los servicios invocados.

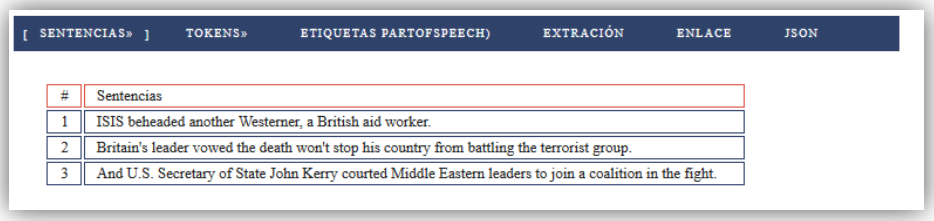
Fuente: (propio)

Con la finalidad de una fácil interacción entre el usuario y la interfaz se presenta un menú correspondiente a cada funcionalidad del sistema invocado, esto se visualiza en la figura 30, este dependerá de los servidos accedidos para su construcción, es decir, cada ítem dentro de este menú corresponde a cada una de respuestas de los servicio, interpretados por el cliente y presentados por separado.

Figura 30. Menú construido con todos los servicios

Fuente: (propio)

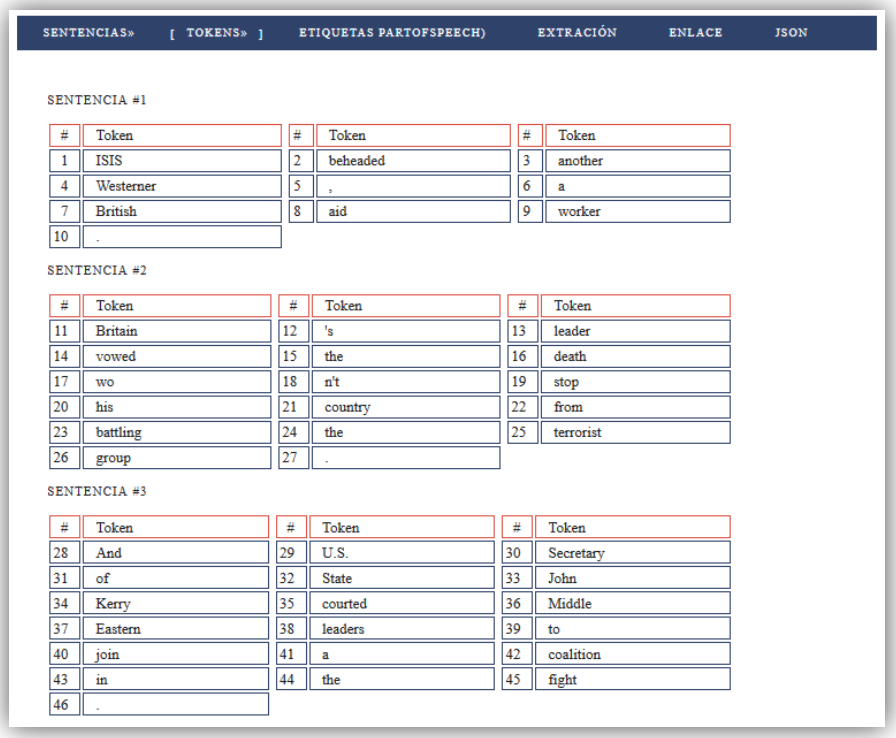
Al igual que el menú con las funcionalidades, este menú de resultado se encuentra en orden de forma que se pude evidenciar cómo evoluciona el tratamiento del texto a través de los servicios invocados. En la figura 31 se puede visualizar las sentencias separadas resultado del servicio de Tokenización en Sentencias, las cuales se encuentran numeradas.



#	Sentencias
1	ISIS beheaded another Westerner, a British aid worker.
2	Britain's leader vowed the death won't stop his country from battling the terrorist group.
3	And U.S. Secretary of State John Kerry courted Middle Eastern leaders to join a coalition in the fight.

Figura 31. Resultado de la función de tokenización
Fuente: (propio)

Separadas igualmente por sentencias y numeradas se encuentran los resultados de la Tokenización en la figura 32 y del Etiquetado, en las figura 33, donde se diferencias las funciones de las palabras dentro de las sentencias base para el servicio de Extracción.



SENTENCIA #1

#	Token	#	Token	#	Token
1	ISIS	2	beheaded	3	another
4	Westerner	5	,	6	a
7	British	8	aid	9	worker
10	.				

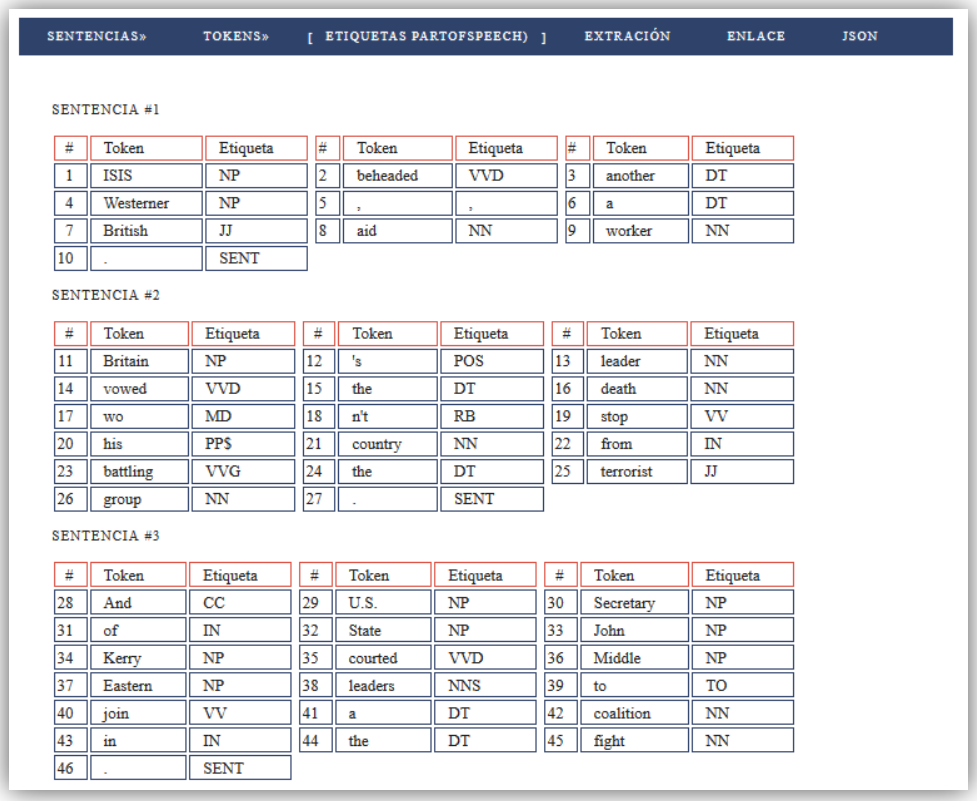
SENTENCIA #2

#	Token	#	Token	#	Token
11	Britain	12	's	13	leader
14	vowed	15	the	16	death
17	wo	18	n't	19	stop
20	his	21	country	22	from
23	battling	24	the	25	terrorist
26	group	27	.		

SENTENCIA #3

#	Token	#	Token	#	Token
28	And	29	U.S.	30	Secretary
31	of	32	State	33	John
34	Kerry	35	courted	36	Middle
37	Eastern	38	leaders	39	to
40	join	41	a	42	coalition
43	in	44	the	45	fight
46	.				

Figura 32. Resultado de la funcionalidad de tokenización.
Fuente: (propio)



SENTENCIAS» TOKENS» [ETIQUETAS PARTOFSPEECH)] EXTRACIÓN ENLACE JSON

SENTENCIA #1

#	Token	Etiqueta	#	Token	Etiqueta	#	Token	Etiqueta
1	ISIS	NP	2	beheaded	VVD	3	another	DT
4	Westerner	NP	5	,	,	6	a	DT
7	British	JJ	8	aid	NN	9	worker	NN
10	.	SENT						

SENTENCIA #2

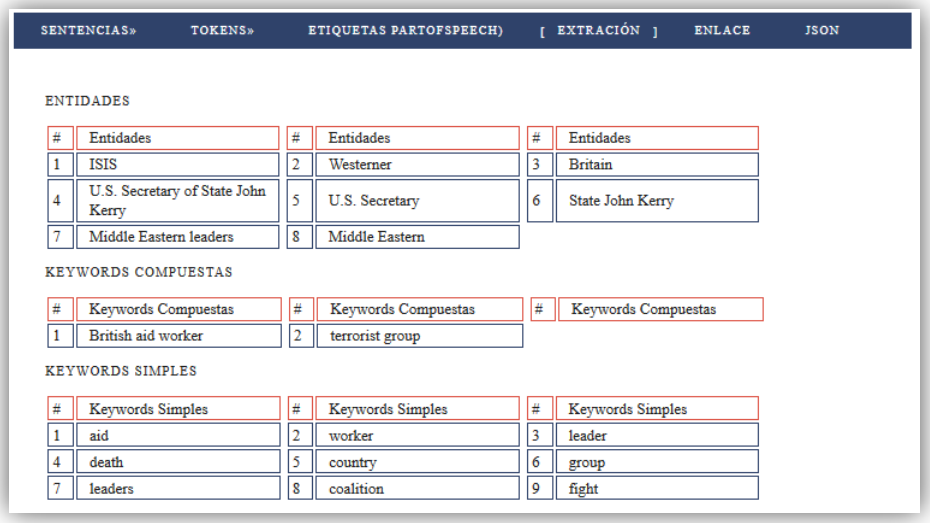
#	Token	Etiqueta	#	Token	Etiqueta	#	Token	Etiqueta
11	Britain	NP	12	's	POS	13	leader	NN
14	vowed	VVD	15	the	DT	16	death	NN
17	wo	MD	18	n't	RB	19	stop	VV
20	his	PP$	21	country	NN	22	from	IN
23	battling	VVG	24	the	DT	25	terrorist	JJ
26	group	NN	27	.	SENT			

SENTENCIA #3

#	Token	Etiqueta	#	Token	Etiqueta	#	Token	Etiqueta
28	And	CC	29	U.S.	NP	30	Secretary	NP
31	of	IN	32	State	NP	33	John	NP
34	Kerry	NP	35	courted	VVD	36	Middle	NP
37	Eastern	NP	38	leaders	NNS	39	to	TO
40	join	VV	41	a	DT	42	coalition	NN
43	in	IN	44	the	DT	45	fight	NN
46	.	SENT						

Figura 33. Resultado de la funcionalidad de etiquetado.
Fuente: (propio)

Los resultados del servicio de extracción se presentan numeradas y no separados por sentencias, divididos en entidades keywords simple y keywords compuestas (keyword, en español palabra claves). Como lo visualización en la figura 34.



SENTENCIAS» TOKENS» ETIQUETAS PARTOFSPEECH) [EXTRACIÓN] ENLACE JSON

ENTIDADES

#	Entidades	#	Entidades	#	Entidades
1	ISIS	2	Westerner	3	Britain
4	U.S. Secretary of State John Kerry	5	U.S. Secretary	6	State John Kerry
7	Middle Eastern leaders	8	Middle Eastern		

KEYWORDS COMPUESTAS

#	Keywords Compuestas	#	Keywords Compuestas	#	Keywords Compuestas
1	British aid worker	2	terrorist group		

KEYWORDS SIMPLES

#	Keywords Simples	#	Keywords Simples	#	Keywords Simples
1	aid	2	worker	3	leader
4	death	5	country	6	group
7	leaders	8	coalition	9	fight

Figura 34. Resultado del servicio de extracción.
Fuente: (propio)

Los enlaces a los recursos de DBpedia enlazados, después de ser desambiguados se muestran en una tabla como se observan en la figura 35.

#	Entidad	Tipo	Enlace
1	ISIS		DBpedia
2	Westerner	http://www.w3.org/2002/07/owl#Thing http://dbpedia.org/ontology/TelevisionShow http://dbpedia.org/ontology/Work http://schema.org/CreativeWork	DBpedia
3	British aid worker	http://xmlns.com/foaf/0.1/Person	DBpedia
4	Britain		
5	terrorist group		
6	U.S. Secretary of State John Kerry		
7	U.S. Secretary		
8	State John Kerry		
9	Middle Eastern leaders		
10	Middle Eastern		DBpedia

Figura 35. Resultado del servicio de enlace.
Fuente: (propio)

Se puede visualizar el JSON resultante de los servicios invocados, antes de ser procesado por el cliente para esto se accede al último de los ítems del menú como lo se puede ver en la figura 36.

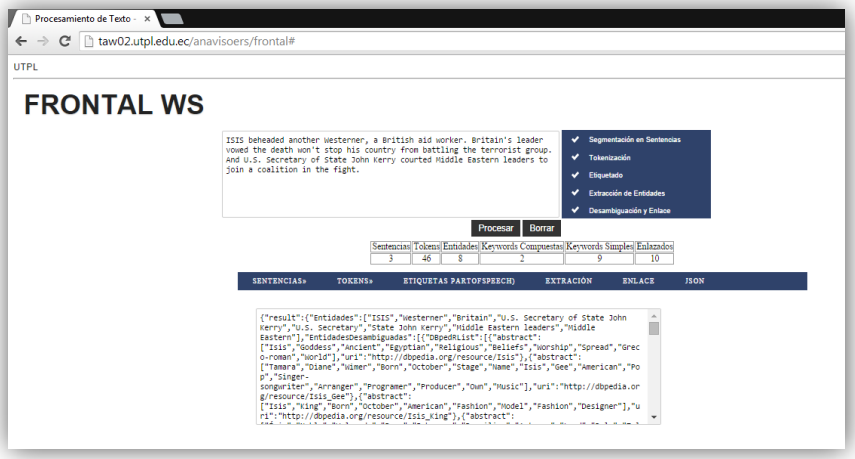


Figura 36. Captura de la visualización del JSON.
Fuente: (propio)

A continuación se presenta un ejemplo en el cual no se accede a todas las funcionalidades del sistema, sino solo a la función de etiquetado y funciones de las cuales depende, una captura del resultado se visualiza en la figura 37.

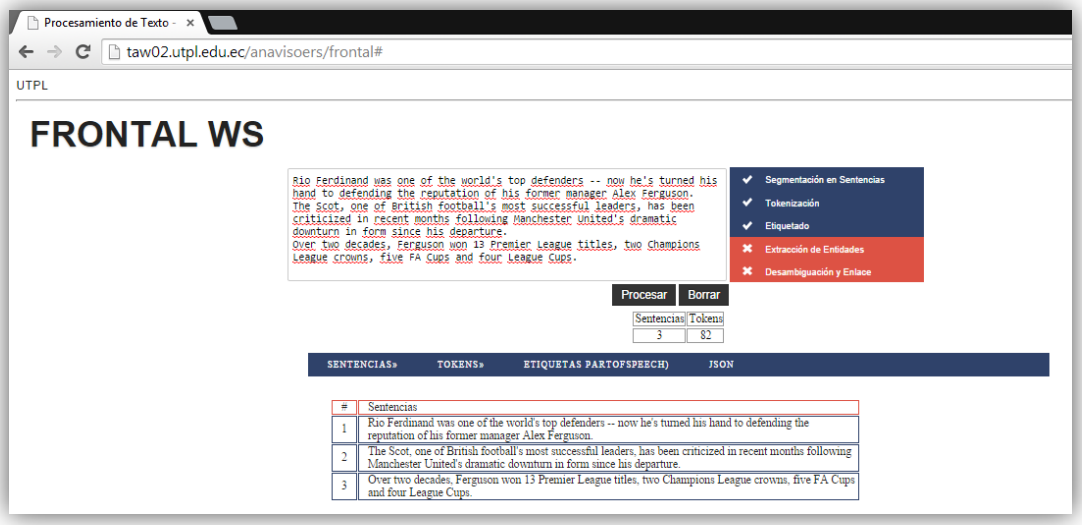


Figura 37. Captura del resultado del servicio de etiquetado de palabra
Fuente: (propio)

La presentación de la tabla resumen de los procesos así como el menú de que permite navegar por los resultados se ven afectados puesto que solo muestran los concernientes a los resultados invocados además del menú para la visualización de del JSON, una ampliación de esto se observa en la figura 38.

Figura 38. Tabla y menú generado de la llamada al servicio de etiquetado.
Fuente: (propio)

3.4.4. *Resumen de prototipos*

Prototipo 1

Descripción

En inicio de se trató de utilizar herramientas abiertas disponibles en la web para el proceso de enlace y desambiguación de los elementos trascedentes en un texto, así se consumió el servicio ofrecido por DBpedia, spotlight el cual produce el resultado esperado pero con problemas de conexión debido a que el servidor no se encontraba siempre disponible, para lo cual se construyó un cliente REST en Python. A partir de este intento y conforme se fue avanzando en la investigación de soluciones para los requerimientos presentados, se decidió desarrollar un propuesta de software propia.

Tabla 18. Tabla resumen del prototipo 1

Funcionalidad especifica	Aprendizaje	Herramienta	Tiempo
Desambiguación y enlace	Consumir servicios libres existentes para procesos de Tiempo excesivos de estará para espera respuesta	DBpedia Spotlight	3 días
Construcción de cliente para servicios	Conexión con servidores REST	Python	2 días

Fuente: (propio)

Prototipo 2

Descripción

A través de Procesamiento de Lenguaje Natural (PLN), se trabaja con el texto de entrada para obtener las entidades de quien se habla en las oraciones, así como las palabras claves que los acompañan. Este procesamiento se lo realiza especializado en el idioma ingles utilizando la librería *Natural Language Toolkit* (NLTK) de Python.

Tabla 19. Tabla resumen del prototipo 2

Funcionalidad especifica	**Aprendizaje**	**Herramienta**	**Tiempo**
Tokenización en sentencias	Procesamiento de lenguaje natural o PLN	NLTK (Natural Language Toolkit) librería de Python.	2 semanas
Tokenización en palabras	PLN	NLTK	2 horas
Etiquetado (Part of speech) en idioma ingles	PLN	NLTK	2 horas
Extracción de entidades y palabras claves idioma ingles	PLN	NLTK	1 día

Fuente: (propio)

Prototipo 3

Descripción

Todos los esfuerzos realizados se concentran en el idioma inglés, a través de NLTK para su procesamiento, para en otros idiomas como español por ejemplo no se puede realizar el mismo proceso que en el idioma ingles debido a lo diferente de su estructura. Para poder trabajar en otros idiomas se debe partir de un etiquetado (Part of Speech) propio del idioma, por lo cual se decidió trabajar con TreeTagger, que permite etiquetado en diferentes idiomas.

Tabla 20. Tabla resumen del prototipo 3

Funcionalidad especifica	**Aprendizaje**	**Herramienta**	**Tiempo**
Etiquetado (Part of speech) en idioma ingles	PLN	TreeTagger	5 días
Extracción de entidades y palabras claves idioma ingles	PLN	NLTK	2 día

Fuente: (propio)

Prototipo 4

Descripción

Una vez que se ha logrado extraer los datos importantes de un texto, se procede a enlazar esto con recursos disponibles en BDpedia.org, para esto se realizan consultas Sparql para obtener los recursos que coinciden con los datos extraídos. Una vez obtenidos los recurso de DBpedia, pueden existir más de una recurso que pueda coincidir para un dato del texto, este caso elque este término es ambiguo y es necesario romper esta ambigüedad para enlazarlo con un solo recurso de DBpedia, para lo cual se implementa el algoritmo de Lesk introducido por Michael E. Lesk en 1986.

Tabla 21. Tabla resumen del prototipo 4

Funcionalidad especifica	**Aprendizaje**	**Herramienta**	**Tiempo**
Enlazar entidades y palabras claves encontrados con recursos similares en DBpedia	Sparql	Python Sparql	4 días
Desambiguación entre los recursos encontrados	Algoritmo de desambiguación de Lesk	Python Sparql	5 días

Fuente: (propio)

Prototipo 5

Descripción

Se levantan servicio **Rest** para ofrecer la propuesta y una interfaz para que pueda interactuar con usuarios.

Tabla 22. Tabla resumen del prototipo 5

Funcionalidad especifica	**Aprendizaje**	**Herramienta**	**Tiempo**
Servicio Web REST	Levantar servicios Web Rest	REST Python	2 semanas
Interfaz de usuario (Web)	Construcción de	JavaScript	3 días

prototipo inicial	cliente Web JavaScript	HTML	

Fuente: (propio)

Prototipo 6

Descripción

Se levantan servicios diferenciados por los procesos realizado de esta forma el resultado de un servicio o es la entrada de toro servicio y así puedes ser consumido por otros servicios externos a esta propuesta.

Para poder visualizar el resultado de la interacción de los servicios Web de forma agradable para usuarios se mejora la interfaz gráfica en donde se diferencia el resultado de cada servicio.

Tabla 23. Tabla resumen del prototipo 6

Funcionalidad especifica	**Aprendizaje**	**Herramienta**	**Tiempo**
Servicios web diferenciados por Procesos	Levantar servicios Web Rest	REST Python	3 semanas
Interfaz de usuario final	Mejoramiento de interfaz Web	Css JavaScript	7 días

Fuente: (propio)

CAPITULO 4: VALIDACIÓN

1. Validación de resultados con servicios similares

Es necesario realizar una comparación de los resultados que se obtienen de este producto de software con opciones libres disponibles en la web, por lo cual se usara el servicio de DBpedia Spotlight.

Para esta validación se contabilizaran los enlaces a los recursos DBpedia, del servicio de Desambiguación y Enlace con los resultados de la versión demo con parámetros por defecto del servicio DBpedia Spotlight.

Se procesará el texto de los *abstracts* perteneciente a publicaciones científicas reales que describimos previo los resultados:

A pesar de ser "productos" que en esencia buscan el descubrimiento y enlace de datos, los resultados se estructuran de diferente manera, por lo cual solo se contabilizara los enlaces finales realizados sin tener en cuenta la clasificación interna de cada servicio a los que estos pertenezcan.

Titulo:	**Maximising the phytochemical content and antioxidant activity of Ecuadorian brown rice sprouts through optimal germination conditions**	
Autores:	Patricio J. Cáceres; Cristina Martínez Villaluenga; Lourdes Amigo; Juana Frias.	
Año:	2013	
Enlaces:	WS Desambiguación:	12
	DBpedia Spotlight:	29

Titulo:	**Productivity and management of Phytelephas aequatorialis (Arecaceae) in Ecuador**	
Autores:	Grischa Brokamp; H. Borgtoft Pedersen; Rommel Montúfar; Janice Jacome; Maximilian Weigend; Henrik Balslev	
Año:	2014	
Enlaces:	WS Desambiguación:	12

	DBpedia Spotlight:	22
Titulo:	Antimicrobial and antioxidant chitosan solutions enriched with active shrimp (Litopenaeus vannamei) waste materials	
Autores:	Autores: Mirari Y. Arancibia; Ailén Alemán, Marta M. Calvo; M. Elvira López-Caballero; Pilar Montero; M. Carmen Gómez-Guillén	
Año:	2013	
Enlaces:	WS Desambiguación:	3
	DBpedia Spotlight:	22

Titulo:	**Diversification across the New World within the blue cardinalids (Aves: Cardinalidae)**	
Autores:	Robert W. Bryson Jr.; Jaime Chaves; Brian Tilston Smith; Matthew J. Miller; Kevin Winker; Jorge L. Pérez-Emán; John Klicka;	
Año:	2014	
Enlaces:	WS Desambiguación:	16
	DBpedia Spotlight:	32

Titulo:	Population genetics of the Federally Threatened Miccosukee gooseberry (Ribes echinellum), an endemic North American species	
Autores:	Nora H. Oleas; Eric J von Wettberg; V. Negrón-Ortiz	
Año:	2014	
Enlaces:	WS Desambiguación:	19
	DBpedia Spotlight:	33

DISCUSIÓN

La propuesta presentada basada en procesamiento de lenguaje natural para descubrimiento de datos relevante e información dentro del texto de publicaciones científicas y su enlace hacia fuentes de datos abiertos, como LOD Cloud, es un esfuerzo por difundir la web semántica a través del procesamiento de fuentes de datos actuales en la web.

Si bien el desarrollo de este proyecto así como su aplicación es un entorno académico la web semántica no se limita a este, partiendo de esto se puede decir, que en la actualidad existe una baja o nula preocupación (o simple desconocimiento) por parte de quienes suben contenido a la web al momento de usar los principios de Datos Enlazados o vincular sus datos con fuentes LOD Cloud, pensando en esto puede considerarse como una herramienta que busca reducir las actividades que una persona interesada en relacionar sus artículos con fuentes de datos enlazados, convirtiéndose además de un facilitador de estas tareas en un motivador para realizarlas.

Los recurso encontrados en las los artículos científicos así como en textos en los que se han aplicado en esta propuesta existen errores, debido a que tecnologías en las que se basa, como el procesamiento del lenguaje natural pueden no ser precisas en un principio por las libertades que se general al momento de redactar escritos que son las base del procesamiento, además de esto en los textos se pueden encontrar entidades o palabras importantes en este, pero que no se encuentren mencionados en DBpedia o que se refieren a un recurso distinto del existente en este Dataset, produciéndose así un problema de ambigüedad (que desde un principio fueron analizados y aceptados) . La mejora en las diferentes tecnologías en las que se basa la web semántica impulsara nuevas herramientas y propuestas.

CONCLUSIONES

La implementación de una Dataset local con los recursos de DBpedia en el idioma inglés, ha permitido evitar posibles problemas ocasionados por conexión pérdidas o lentas con el servidor de DBpedia y su implementación es posible gracias a que facilita sus recursos, y las tecnologías libres para levantar un Triplestore con SPARQL endpoint y se ha utiliza un espacio no mayor en disco a 4 GB en la versión de DBpedia 3.9.

El descubrimiento de datos los textos de las publicaciones y la vinculación de estos a LOD Cloud permite descubrir nuevos datos relacionados al tema que se trata, gracias al beneficio de los principios de los Datos Enlazados.

La desambiguación del sentido de la palabra (WSD, por sus siglas en ingles) en base a los recursos disponibles en DBpedia es posible gracias a sus descripciones, pero al mismo tiempo al no tratarse de descripción con este fin pueden ocurrir errores en el proceso.

Al utilizar el Dataset de DBpedia para procesos de desambiguación permite poder enlazar con facilidad las entidades y palabras claves descubiertos en el texto de las publicaciones con los recursos disponibles en DBpedia.

Las técnicas de procesamiento de lenguaje natural (PNL) permiten descubrir datos relevantes automatizando procesos que consumirían mucho tiempo y recursos valiosos.

El disponer de la lógica de la aplicación a través de los principios de desarrollo REST para servicios web han permitido una indecencia en las tecnologías de construcción del cliente que este caso han sido HTML, JavaScript y CSS.

RECOMENDACIONES

La creación de recursos propios dentro del Dataset local con los recursos de DBpedia, permitiría poder enlazar entidades y palabras claves que no tengan recursos en DBpedia, y que posiblemente pertenezcan a un entorno académico local, como un profesor por ejemplo.

Realizar un análisis previo de la estructura ontológica de los recursos de DBpedia antes de trabajar con estos, para facilitar el "moverse" mediante consultas SPARQL a través de sus propiedades y relaciones.

La utilización de lenguaje de programación de alto nivel, Python ya que está provisto de las librerías *Natural Language Toolkit (NLTK)* y *Treetagger* que permiten el procesamiento de lenguaje natural de forma fácil y potente, y que al igual que Python su curva de aprendiza relativamente corta.

La utilización de la librería TreeTagger para el etiquetado (POS tagging) en el procesamiento de lenguaje natural permite tener la opción de procesar textos en diferentes al idioma ingles que una limitación que la librería NLTK no puede romper.

Utilizar las recomendaciones de la W3C para la publicación de datos enlazados, y enlazar los contenidos actuales publicados en la web a fuentes de datos estructurados con los principios de Datos Enlazados en pro de la difusión de la web semántica como estructura de la web.

Si bien es cierto se sentó la base para trabajar con otros idiomas distintos al inglés, esto no se encuentra desarrollado, así que se lo puede considerar como trabajo futuro.

Como trabajo futuro el desarrollo de un cliente que permita gestionar mejor los resultados y agregar funcionalidades como guardarlos con fines específicos como parte de una ontología.

Bibliografía

Albahari, J., & Albahari, B. (2012). *C# 5.0 IN A NUTSHELL.* O'Reilly.

Beckett, D., Berners-Lee, T., Prud'hommeaux, E., Carothers, G., & Machina., L. (25 de 02 de 2014). *RDF 1.1 Turtle.* Obtenido de W3C Recommendation: http://www.w3.org/TR/2014/REC-turtle-20140225/

Berners-Lee, T. (01 de 2005). *Uniform Resource Identifier (URI): Generic Syntax.* Recuperado el 24 de 06 de 2014, de http://tools.ietf.org/html/rfc3986

Berners-Lee, T. (23 de Julio de 2006). Linked Data - Design Issues.

Bizer, C. (09 de 11 de 2009). *Dbpedia*. Recuperado el 10 de 06 de 2014, de The DBpedia Data Provision Architecture: http://wiki.dbpedia.org/Architecture

Bizer, C., & Cyganiak, R. (25 de 02 de 2014). *RDF 1.1 TriG.* Obtenido de W3C Recommendation: http://www.w3.org/TR/2014/REC-trig-20140225/

Carothers, G., & Seaborne, A. (25 de 02 de 2014). *RDF 1.1 N-Triples.* Recuperado el 25 de 06 de 2014, de W3C Recomendations: http://www.w3.org/TR/2014/REC-n-triples-20140225/

Clark, K. G., Feigenbaum, L., & Torres, E. (01 de 15 de 2008). *SPARQL Protocol for RDF.* Recuperado el 24 de 06 de 2014, de W3C Recommendation 15 January 2008: http://www.w3.org/TR/rdf-sparql-protocol/

Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P. (Marzo de 1992). A Practical Part of Speech Tagger.

Cyganiak, R., Wood, D., & Lanthaler, M. (25 de 02 de 2014). *RDF 1.1 Concepts and Abstract Syntax.* Recuperado el 25 de 06 de 2014, de W3C Recommendation: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

Lapuente, M. J. (08 de 12 de 2013). *HIPERTEXTO: EL NUEVO CONCEPTO DE DOCUMENTO EN LA CULTURA DE LA .* Recuperado el 24 de 06 de 2014, de Tesis doctoral. Universidad Complutense de Madrid.: http://www.hipertexto.info/

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., y otros. (2012). *DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia*. Obtenido de http://semantic-web-journal.net/system/files/swj499.pdf

McBride, B. (10 de 02 de 2004). *W3C Recommendation.* Recuperado el 26 de 06 de 2014, de RDF Primer: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

Mihaela, I. N. (2004). *EL CONOCIMIENTO LINGÜÍSTICO EN LA DESAMBIGUACIÓN SEMÁNTICA AUTOMÁTICA.*

Miller, E. (Mayo de 1998). *An Introduction to the Resource Description Framework*. Recuperado el 27 de 06 de 2014, de http://www.dlib.org/dlib/may98/miller/05miller.html

Miller, E. (1998). Wiley Online Library. *Bulletin of the American Society for Information Science and Technology*, 15-19.

Pautasso, C., Zimmermann, O., & Leymann, F. (Abril de 2008). RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision.

Peláez, A. R., Morocho, J. C., & Malla, P. (2012). *Desambiguación de URI's en el Contexto de Linked Open Data para Linked Universities Data.*

Prud'hommeaux, E., & Seaborne, A. (15 de 01 de 2008). *SPARQL Lenguaje de consulta para RDF.* Recuperado el 25 de 06 de 2014, de Recomendación del W3C de 15 de enero de 2008 : http://skos.um.es/TR/rdf-sparql-query/

Richardson, L., & Amundsen, M. (2013). *RESTful Web APIs.* O'REiLLY.

Richardson, L., & Ruby, S. (2007). RESTful Web Services. En L. Richardson, & S. Ruby, *RESTful Web Services* (pág. 299). O'Reilly.

Ruby, L. R. (2007). *RESTful Web Services.*

San Martin Oliva, C. R. (s.f.). *COMPLEJO UNIVERSITARIO ISLAS MALVINAS.* Recuperado el 07 de 2014, de COMPLEJO UNIVERSITARIO ISLAS MALVINAS: http://www.unsj-cuim.edu.ar/portalzonda/seminario08/archivos/MetodologiaICONIX.pdf

Sandeep Chatterjee, j. W. (2004). *Developing Enterprise Web Services: An Architect's Guide.* Person Education Inc.

Satanjeev, B. (2002). *Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet.*

Schreiber, G., & Raimond, Y. (24 de 06 de 2014). *RDF 1.1 Primer.* Obtenido de W3C Working Group: http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. (2007). *In The semantic web.*

Tello Leal, E. (17 de abril de 2009). *La Desambiguación del Sentido de las Palabras: revisión metodológica*. Obtenido de http://www.nosolousabilidad.com/articulos/desambiguacion.htm

Tjong Kim Sang, E. F., & Buchholz, S. (Septiembre de 2000). Introduction to the CoNLL-2000 shared task: Chunking.

Valverde, M. F., Morocho, J. C., & Piedra, N. (2012). *Estudio sobre la aplicación de Linked Data a la Legislación de Educación Superior en Latinoamérica.*

W3C. (01 de 2005). *Uniform Resource Identifier (URI): Generic Syntax.* Recuperado el 22 de 02 de 2014, de January 2005

W3C. (2013). *W3C.* Obtenido de http://www.w3.org/standards/semanticweb/ontology

Wood, D. (25 de 02 de 2014). *What's New in RDF 1.1.* Obtenido de W3C: http://www.w3.org/TR/2014/NOTE-rdf11-new-20140225/

ANEXOS

3. Anexo 1: Especificación de Requerimientos de Software (ERS)

Especificación de Requerimientos de Software (ERS)

WS para descubrimiento, desambiguación y enlace en Datos Enlazados

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web de Extracción de Entidades
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_ExtracciónEntidades
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Especificación de Requerimientos de Software (ERS)

Introducción

Descripción

El presente documente tiene como finalidad redactar las funcionalidades con las que debe contar el sistema de descubrimiento, desambiguación y enlace en datos enlazados.

Problemas Conocidos

Después de un análisis inicial se detectan los siguientes problemas:

- Los *abstract* de las publicaciones universitarias que contienen datos relevantes a los que se desea acceder se encuentran en texto plano entendible solo para humanos.
- Los datos en texto no son referenciados hacia fuentes externas.
- Los datos en texto plano pueden ser ambiguos y tener más de un enlace posible en LOD Cloud específicamente DBpedia.
- No existe un proceso estándar para procesos de enlace y desambiguación lingüística.
- Las conexiones y/o consultas hacia el DataSet de DBpedia pueden demorar o fallar.

Referencias

ANSI/IEEE Std. 830-1984 Guía del IEEE para la Especificación de Requerimientos Software.[29]

Descripción General

[29] http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=2228

El fin del sistema es el descubrimiento de datos relevantes dentro del texto plano en los *abstracts* en las publicaciones universitarias, y si en caso un término extraído es ambiguo determinar el significado usado, para luego ser enlazado a DBpedia (LOD Cloud) si en caso existiera un recurso al cual referencie.

Para acceder al sistema se levantara servicios web diferentes para cada proceso relevante dentro del sistema que interactuaran entre sí. Se desarrollara un interfaz web para usuarios que permitirá visualizar los resultados individuales e integrales de los servicios.

Perspectiva del Proyecto

Esto se pretende desarrollar en base a la relevancia que toman los datos en la web semántica, buscando enlazar las publicaciones científicas a las fuentes de Datos Enlazados, donde se ubican los recursos a los cuales hacen referencia y permitiendo de esta forma ampliar la información y descubrir nuevos enlaces.

El sistema que permita extraer datos relevantes dentro del texto de las publicaciones científicas, desambiguar estos términos de ser necesario y enlazarlos a LOD Cloud. Se construirá separando e integrando los procesos relevantes mediante servicio web.

Características del Producto

El producto a desarrollar constara de las siguientes características:

- **Servidor**
  - Servicios web REST
  - Separados en los procesos importantes:
    - Procesamiento de lenguaje natural,
    - Desambiguación y
    - Enlace
  - Integrados entre si
  - Servidor local con DataSet de DBpedia (SPARQL EndPoint)

- **Cliente**
  - REST web
  - Visualizar resultados individuales de los servicios
  - Visualizar resultado integrado de los servicios
  - Permitir ver el JSON resultante del servicio consumido

Características del Usuario

Usuarios anónimos: a través de la construcción del cliente Rest web cualquier usuario podrá interactuar con el sistema.

Usuarios clientes Rest: los servicios web implementaos podrán responder al cualquier servicio que se pueda construir a partir de esos.

Limitaciones Generales

A continuación se detallan limitaciones en cuanto al software:

- Todos los enlaces que se puedan realizar se los hará con recursos disponibles en el DataSet de DBpedia, esto significa que pueden existir recursos en otro u otros repositorios a los cuales no se los enlazara directamente.
- De no existir el término extraído en DBpedia, no podrá ser enlazado.
- La desambiguación de un "término" extraído de una publicación se realizara en base a los recursos disponibles en la DBpedia que son nombrado mediante este "termino".

Asunciones y Dependencias

Asunciones

- Posibles errores en extracción de términos del texto de las publicaciones debido a faltar tipeado o error humano en la escritura del texto.
- Posibles errores el en enlace de a DBpedia producto de no existir recurso o error de desambiguación.

Dependencias

- Se desarrollara en lenguaje de programación de alto nivel Python 2.7 y algunas de sus librerías especializadas en procesamiento de lenguaje natural, levantamiento de servicios, consultas SPARQL, etc.
- Cliente desarrollara en base a HTML, CSS, JavaScript, etc.
- De navegadores web que soporte tecnologías en las que se construirá en cliente para poder acceder a este.

Requerimientos Funcionales.

REQ001 Extraer entidades y palabra relevantes

Descripción
Descubrir datos relevantes en el texto, a quien se describe y las palabra relevantes que lo acompañan

Entrada

- Texto

Proceso

1. Tokenización del texto en sentencias (oraciones), separa todas las sentencias.
2. Tokenización de las sentencias en palabras.
3. Etiquetar (Part of Speech).
4. Extracción en base a etiquetas.

Salida

- JSON con entidades y palabras claves extraídas

REQ002 Enlazar entidades y palabra relevantes con LOD Cloud

Descripción
Se enlazara los términos encontrados en caso de que sea posible con la LOD Cloud

Entrada

- JSON estructurado por procesos anteriores con entidades a enazar

Proceso

1. Consultar a DBpedia por recursos que sean nombrados con las entidades y palabras relevantes extraídas del texto de entrada.

Salida

- JSON estructurado con los enlaces de los recursos de DBpedia.

REQ002 Desambiguar entidades y palabra relevantes

Descripción
Se determinara el sentido con que las palabras estas siendo usadas en caso de que estas sean ambiguas

Entrada

- JSON procesos anteriores

Proceso

1. Aplicar algoritmo s de desambiguación en base a contexto.
2. Determinar sentido utilizado en base a mejor resultado de coincidencia.

Salida

- JSON con términos desambiguados.

REQ004 Levantar servicios REST separados para los procesos relevantes.

Descripción
Para que los procesos relevantes dentro del sistema puedan ser reutilizados se levantaran servicios individuales.

Entrada

- JSON.
- Texto.

Proceso

1. Se invoca las funciones necesarias del sistema para resolver la petición del servicio llamado.

2. Se estructurará la data en JSON.
3. Se devuelve el JSON ya sea al servicio que lo invocó o al cliente si este fuese el origen de la invocación del servicio.

Salida

- JSON estructurado de acuerdo al servicio invocado.

REQ005 Frontal UI Web

Descripción
Construir una interfaz web que permita visualizar el comportamiento del sistema, es decir, la integración de los servicios y su funcionamiento individual

Entrada

- Texto

Proceso

1. Introducir texto a ser procesado
2. Seleccionar servicios a ser invocados
3. Esperar resultado
4. Procesar resultado
5. Presentar resultado procesado

Salida

- Resultado gráfico de servicios invocados.

4. Anexo 2: Especificación de Caso de Uso (ECS) - Tokenización en Sentencias

Especificación de Caso de Uso (ECS)

Tokenización en Sentencias

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web - Tokenización en Sentencias
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_TokenizaciónSentecias
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Número	**ECS-01**	
Nombre	Tokenización en Sentencias	
Actores	Usuario, Cliente	
Descripción	Divide el texto de entrada en sentencias cortas separadas por un punto y parte, la salida es una lista de estas sentencias.	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado ▪ Texto segmentado en sentencias	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web. **SA1**
	2	Verifica el número de sentencias que comenten al texto, que estén separadas por un punto seguido (.)
	3	Divide cada una teniendo en cuenta la terminación con punto (.) estructura las sentencias dentro de una lista. **SA1**
	4	Estructura la lista de elementos formato JSON.
	5	Devuelve el JSON resultante.
Poscondición	▪ El texto dividido en sentencias.	
Secuencia alternativo	**SA1 el número de sentencias es 1** Se estructura una lista de un solo elemento con la sentencia.	
Prioridad	Media	
Requerimientos Especiales		
Asunciones y Dependencias		
Notas adicionales		

5. Anexo 3: Especificación de Caso de Uso (ECS) - Tokenización en Palabras

Especificación de Caso de Uso (ECS)

Tokenización en Palabras

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web de Tokenización
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_Tokenización
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Número	**ECS-02**	
Nombre	Tokenización en palabras	
Actores	Cliente, Servicio Web	
Descripción	Divide cada sentencia en palabras validas, tokens.	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Segmentación del texto en sentencias. **ECS-01**
	3	Se recorre la lista de sentencias segmentadas.
	4	Se divide palabra por palabra de la sentencia en una lista, se obtiene una lista de listas.
	5	Se estructura en formato JSON
	6	Retorna el JSON con las sentencias divididas en "tokens"
Postcondición	▪ Texto tokenizado por sentencias y estos a la vez tokenizados en palabras	
Secuencia alternativo		
Prioridad	Baja	
Requerimientos Especiales	Del funcionamiento del Servicio web de Tokenización en Sentencias	
Asunciones y Dependencias		
Notas adicionales		

6. Anexo 4: Especificación de Caso de Uso (ECS) - Etiquetado

Fabricio Montaño
Analista - Desarrollador

Ing. Nelson Piedra

Especificación de Caso de Uso (ECS)
Etiquetado

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web de Etiquetado
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_Etiquetado
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Número	**ECS-03**	
Nombre	SW-Etiquetado	
Actores	Cliente, Servicio Web	
Descripción	Este servicio permite la tokenización de cada palabra y etiquetación de las mismas de acuerdo a la función que cumplen en el contexto que se encuentra, para hacerlo se apoya en el servicio web de tokenización en sentencias	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Tokenización del texto en sentencias. **ECS-01**
	3	Recorrido de la lista de sentencias
	4	Etiquetado de las palabras que conforman cada sentencia
	5	Estructura y retorna data en JSON
Poscondición	▪ Texto tokenizado a nivel de palabras y etiquetado.	
Secuencia alternativo		
Prioridad	Alta	
Requerimientos Especiales		
Asunciones y Dependencias		
Notas adicionales	Depende del funcionamiento del servicio web de Etiquetado en Sentencias (ECS-01)	

7. Anexo 5: Especificación de Caso de Uso (ECS) - Extracción de Entidades

Especificación de Caso de Uso (ECS)

Extracción de Entidades

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web de Extracción de Entidades
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_ExtracciónEntidades
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Número	**ECS-04**	
Nombre	Extracción de Entidades	
Actores	Cliente, Servicio Web	
Descripción	Permite reconocer y extraer, las entidades y palabras relevantes o claves (keywords) que se encuentran dentro del texto, para lograr se apoya en el servicio web de Etiquetado (y en los que este a su vez , servicio web de tokenización de sentencias)	
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado	
Secuencia Normal	Paso	Acción
	1	Entrada del texto validado, como parámetro para el servicio web.
	2	Tokenización del texto en sentencias. **ECS-01**
	3	Tokenización y Etiquetado de palabra **ECS-03**
	4	Reconocimiento de estructuras de Entidades y Keywords
	5	Extracción de Entidades y Keywords
	6	Estructuración de retorno de resultado en formato JSON
Poscondición	▪ Entidades y palabra importantes que las acompañan reconocidos y extraídos	
Secuencia alternativo		
Prioridad	Alta	
Requerimientos Especiales		
Asunciones y Dependencias		
Notas adicionales	Este servicio depende del funcionamiento del servicio web de Tokenización en Entidades (ECS-01) y Servicio web de Etiquetado (ECS-03)	

8. Anexo 6: Especificación de Caso de Uso (ECS) - Desambiguación y Enlace

Especificación de Caso de Uso (ECS)

Desambiguación y Enlace

Versión [1.0]

Información de Documento

TÍTULO:	Especificación de casos de Uso
SUBTÍTULO:	Servicio Web – Desambiguación y Enlace
VERSIÓN:	[1.0]
ARCHIVO:	ECS_SW_ DesambiguaciónEnlace
AUTOR:	Fabricio Montaño
ESTADO:	Borrador

Lista de Cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0	17/07/2014	Fabricio Montaño	Emisión inicial

Firmas y Aprobaciones

ELABORADO POR:	Fabricio Montaño		
FECHA:	17/07/2014	FIRMA:	
REVISADO POR:	Ing. Nelson Piedra		
FECHA:		FIRMA:	

Número	**ECS-05**
Nombre	Desambiguación y Enlace
Actores	Cliente, Servicio Web
Descripción	Enlaza las entidades y palabras relevantes (keywords) hacia LOD Cloud, más específicamente DBpedia, esto de existir un recurso al cual vincular, en caso de que una entidad o keyword tuviese más de uno posible recurso al cual enlazar, se realizara un proceso de desambiguación y luego de enlace.
Precondición	▪ Ingreso texto como parámetro de la aplicación ▪ Texto ha sido validado y procesado
Secuencia Normal	Paso | Acción 1 | Entrada del texto validado, como parámetro para el servicio web. 2 | Tokenización del texto en sentencias. **ECS-01** 3 | Tokenización y Etiquetado de palabra. **ECS-03** 4 | Extracción de Entidades y keywords. **ECS-04** 5 | Consulta de recursos a DBpedia. 6 | Consulta de "Abstract" de recurso a DBpedia 7 | Verificar si existen Entidades o keywords ambiguas 8 | Desambiguar Entidades y keywords ambiguos. **SA1** 9 | Estructurara resultado 10 | Retornar resultado
Poscondición	▪ Entidades y palabra importantes que las acompañan reconocidos y extraídos
Secuencia alternativo	**SA1 Entidades y keywords no ambiguos** Se enlaza con los recursos únicos encontrados a las entidades y keywords del texto.
Prioridad	Alta
Requerimientos Especiales	

Notas adicionales	Este servicio depende de los servicios web de tokenización en sentencias (ECS-01), etiquetado (ECS-03), extracción de entidades (ECS-04).

9. Anexo 6: Pseudocódigo del algoritmo de Lesk encontrado en (Satanjeev, 2002)

```
for every word w[i] in the phrase
  let BEST_SCORE = 0
  let BEST_SENSE = null
  for every sense sense[j] of w[i]
    let SCORE = 0
    for every other word w[k] in the phrase, k != i
      for every sense sense[l] of w[k]
        SCORE = SCORE + number of words that occur in the gloss of
                        both sense[j] and sense[l]
      end for
    end for
    if SCORE > BEST_SCORE
      BEST_SCORE = SCORE
      BEST_SENSE = w[i]
    end if
  end for
  if BEST_SCORE > 0
    output BEST_SENSE
  else
```

10. Anexo 6: Pseudocódigo del algoritmo de Lesk encontrado en (Satanjeev, 2002)

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NP	Proper noun, singular
15.	NPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PP	Personal pronoun
19.	PP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb