ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Curso Académico 2012/2013

Proyecto de Fin de Carrera

APLICACIÓN BIBLIOGRÁFICA USANDO LINKED DATA

Autor: Ariadna Gómez Ruiz

Tutor: Alberto Fernández Gil

RESUMEN

Tras la revolución que ha supuesto Internet como medio de almacenamiento, difusión, compartición y búsqueda de información, nos encontramos en el momento actual, entre otros, con un problema doble provocado por el mismo exceso de información y la forma en que se presenta a los usuarios: por un lado acceder a la información útil en los archivos de hipertexto de la red, puede ser a menudo laborioso y complicado, es decir, para entresacar los datos precisos que necesitamos, hay que leer cientos o miles de páginas que pueden contener la información buscada o hacer una referencia tangencial a la misma o no disponer de ella.

Por otra parte, esta búsqueda de información no puede ser procesada por máquinas, lo que sin duda es una grave carencia, resuelta la cual, la evolución del mundo de las comunicaciones web, y por tanto de todas aquellas disciplinas que las utilicen –o lo que es lo mismo, todas- avanzará de manera exponencial.

El proyecto presentado en esta memoria solventa estas cuestiones en una parcela del conocimiento que es uno de los pilares del saber humano: los libros (incluyendo publicaciones y textos de diversa índole). Pues los libros son los contenedores de la mayor parte del conocimiento, están llenos de historias, de ideas de sentimientos, de sensaciones, de propuestas, deducciones, relaciones, sueños, proyectos…de todo.

Sin duda, como decía Umberto Eco "*Hay una satisfacción deportiva en dar caza a un texto que no se encuentra*" [1]. Y ese, precisamente, ha sido el objetivo de este proyecto, poner a disposición de los usuarios los datos principales que sobre los escritores y sus obras existen. Así el proyecto consiste en una aplicación web para cuya realización se han analizado diferentes fuentes de datos, estudiando su documentación, modelo de datos, problemas y virtudes de las mismas para establecer una relación entre ellas, que permita al usuario disponer de información conectada de tres importantes fuentes de datos acerca de escritores de cualquier rama, sus fichas biográficas, los libros escritos por ellos y las influencias de sus obras en otros escritos.

Esta aplicación realizada es una pequeña muestra de las inmensas posibilidades de Linked Data [2].

AGRADECIMIENTOS

Quiero dar las gracias a las personas que me ayudaron a encontrar el apasionante camino de la informática y a los que me han implicado en él, especialmente al tutor de este proyecto, Alberto Fernández Gil, por sus consejos e indicaciones.

Ari Gómez

ÍNDICE

TABLA DE FIGURAS

1. INTRODUCCIÓN

La primera red interconectada de ordenadores se creó en septiembre de 1969 cuando se estableció la comunicación entre las universidades de UCLA y Stanford por medio de una línea telefónica conmutada. Desde entonces, la enorme red de comunicaciones que supone internet ha puesto a disposición de los usuarios multitud de servicios entre los que se encuentra la World Wide Web (WWW, o "la Web").

La WWW o Red informática mundial, software de navegación que se emplea actualmente para el uso de Internet y que fue puesto en funcionamiento en 1990 por Tim Berners-Lee y Robert Cailliau, es un sistema de distribución de información que utiliza archivos de hipertexto. Esta información contenida en la Web es extensísima ya que es una web no solo de consulta sino que permite a los usuarios el aporte de datos, lo que ha provocado un crecimiento exponencial de la información contenida.

Al margen de otros problemas como puedan ser la veracidad de las fuentes y la calidad de la información, la magnitud de la misma resulta excesiva en ocasiones, porque, además de carecer de estructura de contenidos, contiene un abanico de posibilidades tan amplio, que cada vez es más factible perderse al buscar los mejores y adecuados, resultados pues se hace necesario saltar, enlazando textos asociados, de un contenido a otro, en la investigación de la información que nos proporciona por cada una de las cuestiones que le planteemos.

Resumiendo, para encontrar la información ajustada a cada pregunta que lancemos a la web, nos vemos obligados a poner en marcha un lento proceso de búsqueda y lectura. Y así, se ha de encontrar la información útil entre otra que no lo es tanto, de forma tediosa y manual. Porque otro de los problemas de la web tradicional es que las máquinas no pueden procesar y usar la información que contiene, ésta ha de extraerse de manera manual mediante el análisis de los documentos que la contienen (o no).

Tim Berners-Lee, el precursor de la Web Semántica, en el congreso TED (*Technology Entertainment and Design)* de 2009 [3], ilustró este problema refiriéndose a un ejemplo concreto: la búsqueda de qué proteínas participan en la transmisión de señales y están relacionadas con las neuronas piramidales da 223.000 resultados en Google y ninguno

de ellos de interés ya que no existe página web alguna que responda exactamente a esa cuestión, porque nadie lo ha preguntado antes.

Sin embargo, si consultásemos una web en que, en lugar de los documentos, fueran los datos enlazados las bases de la misma (Web de datos o Linked Data), obtendríamos 32 resultados, cada uno de los cuales es una proteína con esas propiedades.

Este proyecto que aquí se presenta es una pequeña muestra del potencial que supone utilizar Linked Data, pues se trata de una aplicación inteligente capaz de ofrecer un valor añadido al usuario:

Partiendo de una parcela del conocimiento, la información existente en una fuente de datos pública referida a escritores, se ha tratado de conseguir un nuevo producto, que podemos consultar, incorporando información procedente de otras fuentes y estructurando su contenido. Debemos hacer la consideración de que el término "escritor" lo utilizamos en un sentido amplio, pues incluye a personas que han podido publicar un solo texto, sin que podamos afirmar que eso les de estatus de escritor.

Los datos utilizados en este proyecto están alojados en servidores distintos y son mantenidos por entidades independientes, pero se encuentran referenciados entre ellos lo que ha hecho posible su publicación estructurada según un criterio de usabilidad.

A lo largo del proyecto veremos con detalle el proceso para generar esta aplicación desde el análisis, herramientas y lenguajes utilizados, hasta el diseño y la implementación incluyendo las características de las fuentes de datos, las dificultades y soluciones halladas así como posibles mejoras para el desarrollo más amplio de la aplicación.

1.1. LA WEB SEMÁNTICA

En mayo de 2001, Tim Berners Lee, James Hendler y Ora Lassila publicaron un artículo en la revista "Scientific American" titulado *The Semantic Web: a new form of Web content that is meaninful to computers will unleash a revolution of new possibilities* [4], con el que popularizaron la idea de la Web Semántica cuya arquitectura fue representada en la siguiente figura:

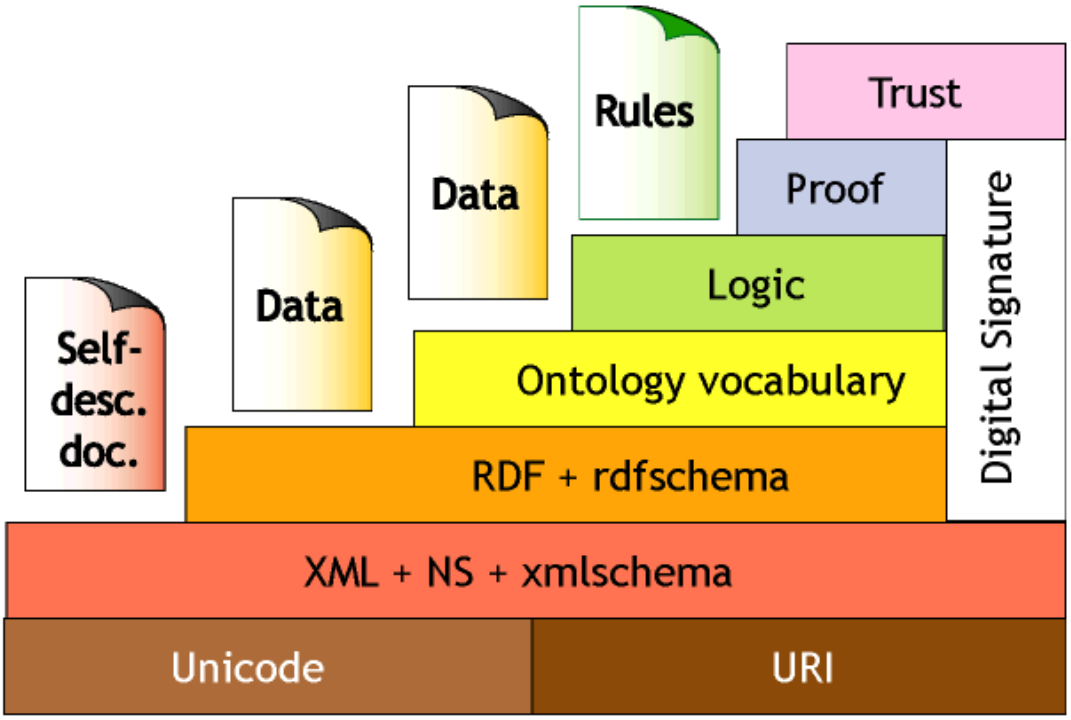

FIG 1.- TIM BERNERS-LEE. *SEMANTIC WEB -XML2000. ARCHITECTURE* [5]

La Web Semántica sería una extensión de la Web actual dotada de significado, esto es, un espacio donde la información tendría un significado bien definido, de manera que pudiera ser interpretada tanto por agentes humanos como por aplicaciones software.

Esta nueva web, concebida como una nueva forma de contenido, que toma datos puros de la red, se configura mediante los aportes de múltiples fuentes, instituciones públicas, empresas privadas y usuarios, y a su vez esos datos abiertos enlazados pueden ser reutilizados y combinados para generar nueva información, siendo mayor su utilidad cuanto más interconectados estén con otros datos en la Web de Datos, beneficiándose del "*efecto red*".

Estos datos, además de por las personas, pueden ser analizados y utilizados, como decíamos, por máquinas de forma automática. Es decir, la semántica permite a las máquinas procesar su contenido, combinarlo y realizar deducciones lógicas para aportar, de forma automática, las respuestas buscadas.

Esta Web, basada en el significado o semántica, muestra el camino a las máquinas para que resuelvan problemas "*bien definidos, a través de operaciones bien definidas que se llevarán a cabo sobre datos existentes bien definidos*" [6].

Resumiendo, por tanto, podemos decir que la Web Semántica se basa en dos conceptos fundamentales:

1) La descripción del significado que tiene los contenidos en la Web.

   Esta descripción requiere además que:

   (a) La Semántica posea un significado procesable por las máquinas.

   (b) Los Metadatos sean contenedores de información semántica sobre los datos.

2) La manipulación automática de estos significados.

   Esta manipulación automática de los significados se hace a través de:

   (a) La lógica matemática, que permite establecer reglas para tratar el contenido semántico.

   (b) Los motores de inferencia, que permiten combinar conocimientos conocidos para elaborar otros nuevos conocimientos.

Así pues, para que las máquinas puedan operar con las informaciones que se proporcionan a la web, ya sean documentos de texto, archivos de video o sonido, etc., éstas han de describir el contenido, el significado y la relación de los datos y la forma en que ha de catalogarse la información de los recursos, es mediante el significado de las palabras, no mediante palabras claves.

1.2. LENGUAJES DE ONTOLOGÍAS

Desde que en 1729 Christian von Wolff escribiese *Filosofía primera u Ontología (Philosophia prima sive Ontologia)* el concepto "Ontología" se ha adaptado a distintos contextos como por ejemplo en la biblioteconomía y la documentación como método para anticipar el contenido y/o interés de los registros bibliográficos mediante el desarrollo de catálogos, compuestos por ficheros y organizados alfabéticamente por campos, hasta la utilización, desde finales de los años 80, con el propósito principal de representar un conjunto de conceptos jerárquicamente organizados, descritos en algún sistema informático.

Una década después, las ontologías se convirtieron en un eje fundamental en las nuevas tecnologías para la Web semántica. Actualmente definen vocabularios "legibles por máquinas" y que son especificados con la suficiente exactitud como para permitir

diferenciar términos y referenciarlos de manera precisa [7]. Mediante las ontologías ampliamos la red de la información (WWW) hasta la red del conocimiento (Web Semántica).

Existen numerosas definiciones de ontologías, entre las que podemos destacar:

> *"Una ontología es un vocabulario acerca de un dominio: términos + relaciones + reglas de combinación para extender el vocabulario"*. Neches, 1991.
>
> *Una ontología necesariamente incluirá un vocabulario de términos y una especificación de su significado (definiciones e interrelaciones entre conceptos) que impone estructura al dominio y restringe las posibles interpretaciones.* Uschold-Jasper, 1993.
>
> *Una ontología es la especificación de una conceptualización*. Gruber, T, 1993.
>
> *Una ontología es "un instrumento de organización y representación del conocimiento que permite hacer explícitas las reglas implícitas de una parte de la realidad. Idealmente, su presentación formalizada permite que estas declaraciones explícitas sean independientes del sistema que las utiliza y que, a su vez, pueda reutilizarse por otros sistemas"*. Bosch M., 2004.

Según el científico e ingeniero informático, autor de una de las más conocidas definiciones de ontologías, Tom R. Gruber [8], las ontologías se componen de:

> Conceptos: clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
> Relaciones: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
> Funciones: asignar-fecha, categorizar-clase, etc.
> Instancias: se utilizan para representar objetos determinados de un concepto.
> Reglas de restricción o axiomas: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc.

Y así, nos permiten:

- Compartir conocimiento común sobre la estructura de las cosas
- Reusar el conocimiento del dominio

- Explicitar suposiciones sobre el dominio
- Separar el conocimiento del dominio del conocimiento operacional
- Posibilitar el análisis del conocimiento del dominio

En resumen, una ontología es un sistema de representación del conocimiento que resulta de seleccionar un dominio o ámbito del conocimiento, y aplicar sobre él un método con el fin de obtener una representación formal de los conceptos que contiene y de las relaciones que existen entre dichos conceptos. Son numerosos los proyectos desarrollados en Internet con lenguajes de codificación de ontologías. El servidor Protégé [9] ofrece herramientas para crear ontologías, integrarlas con otras existentes e incorporarlas a nuevos productos de software.

Además, una ontología contiene definiciones que nos suministran el vocabulario para referirse a un dominio. Estas definiciones dependen del lenguaje que usemos para describirlas.

1.2.1. RDF

A pesar de que el HyperText Markup Language (HTML) es, todavía hoy, el lenguaje por excelencia de la web, su utilización ha hecho de él un lenguaje de formato, en lugar de un código semántico como pareciera en un principio. Gracias a que se ajusta a normas muy estandarizadas todos los ordenadores pueden reproducir correctamente los documentos HTML de la red, pero orientado como está a la presentación de datos, ofrece escasa información y un pequeño número de etiquetas. En el camino hacia la Web Semántica, se realizaron algunas mejoras y se añadieron a la Web otros lenguajes que permitieran ofrecer una información más estructurada, entre ellos el Extensible Markup Language (XML), creado para enriquecer la estructura de los documentos que pueden ser usados en la Web. El XML aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado. La palabra "Extensible" del nombre hace alusión a que no limita el número de etiquetas pudiéndose crear las que fuesen necesarias.

El XML Schema (XMLS) suministra un significado para definir la estructura, contenido y semántica de los documentos XML, permitiendo la definición de gramáticas y etiquetas significativas. Los esquemas definen qué elementos pueden contener los

documentos XML, cómo están organizados, y qué atributos y de qué tipo pueden tener sus elementos, número mínimo y máximo de ocurrencias, si debe ser un número entero, una cadena de texto, una fecha, etc. y otras características más específicas; es decir jerarquizan, validan y estructuran el contenido; representan pues un paso más en la construcción de la Web Semántica.

Según el consorcio internacional, World Wide Web Consortium (W3C) [10], que produce recomendaciones para la WWW, en su especificación *XML Schema* [11], los esquemas expresan vocabularios compartidos que permiten a las máquinas extraer las reglas hechas por las personas.

Pero ni el XML ni el XMLS son suficientes ya que aportan una estructura, pero no una semántica. Para superar esa carencia se creó el lenguaje Resource Description Framework (RDF) [12] como lenguaje para especificar metadatos frente al XML que es un lenguaje para componer datos.

El RDF surge en agosto de 1997 en el seno del W3C y está recogido en sus recomendaciones: Primer, Concepts, Syntax, Semantics, Vocabulary (Schema) y Test Cases [13]:

> **W3C.** ***RDF Primer*** [14]: RDF es un lenguaje para referenciar la información de los recursos de la World Wide Web. RDF Primer ofrece los conocimientos básicos requeridos para usar RDF, introduce los conceptos básicos de RDF y describe su sintaxis XML.
>
> **W3C.** ***RDF Concepts and Abstract Syntax*** [15]: define la sintaxis abstracta en la que está basada RDF y explica para qué sirve enlazar una sintaxis concreta a una semántica formal.
>
> **W3C.** ***RDF/XML Syntax Specification (Revised)*** [16]: define la sintaxis XML para RDF llamada RDF/XML.
>
> **W3C.** ***RDF Semantics*** [17]: especifica una semántica precisa y ofrece un completo sistema de reglas de inferencia para RDF y RDFS.
>
> **W3C.** ***RDF Vocabulary Description Language 1.0: RDF Schema*** [18] describe cómo usar RDF para describir vocabularios RDF.

W3C. *RDF Test Cases* [19]: describe el RDF Test Cases ofrecido por el RDF Core Working Group

RDF fue diseñado como un mecanismo para posibilitar que los agentes software interpretasen la información disponible en Internet, asociando información sobre el contenido de los recursos web, describiendo esos recursos en términos de propiedades simples y valores y capacitándolo para referenciar prácticamente cualquier cosa, ya sea física o abstracta, mediante los Uniform Resource Identifiers o URIs.

URI o Identificador Uniforme de Recursos, es el identificador único que permite la localización de un recurso al que puede accederse vía Internet. Se trata del Uniform Resource Locator (URL) o descripción de la ubicación más el Uniform Resource Name (URN) o descripción del espacio de nombre.

El lenguaje RDF permite una representación explícita de los datos mediante tres tipos de objetos [W3C]:

Recursos: cualquier objeto web identificable unívocamente por un URI.
Propiedades: aspectos específicos, características, atributos o relaciones utilizadas para describir recursos.
Sentencias: conjunto de un recurso, un nombre de propiedad y el valor de esa propiedad.

Una declaración RDF toma la forma de una tripla compuesta de un sujeto, un objeto, y un predicado que determina la relación que une sujeto y objeto.

Estas sentencias se pueden representar por tanto:

- Sujeto - Recurso
- Predicado - Propiedad
- Objeto – Valor de la propiedad

Pero RDF también nos permite representar los recursos, y sus propiedades y valores como un grafo de nodos y arcos siendo los sujetos y objetos, nodos, mientras que los predicados son arcos.

Así pues la tripla se representa mediante nodos conectados por líneas con etiquetas. Los nodos representan recursos y las líneas con etiquetas las propiedades de esos recursos. Los 3 elementos de una tripleta se representan mediante URIs.

FIG 2.- SENTENCIA RDF

Veamos un ejemplo concreto, extraído de la especificación Primer RDF [14] donde se muestran una serie de declaraciones o sentencias:

"Hay una persona identificada por http://www.w3.org/People/EM/contact#me, cuyo nombre es Eric Miller, cuya dirección de correo electrónico es em@w3.org, y cuyo título es "Dr." que podría representarse como el grafo RDF de la siguiente figura:

FIG 3.- GRAFO RDF QUE DESCRIBE A ERIC MILLER

Esta figura ilustra que RDF usa URIs para identificar:

- individuos, por ejemplo, **Eric Miller**, identificado por **http://www.w3.org/People/EM/contact#me**
- clases de cosas, por ejemplo, **Person**, identificado por **http://www.w3.org/2000/10/swap/pim/contact#Person**
- propiedades de estas cosas, por ejemplo, **mailbox**, identificado por **http://www.w3.org/2000/10/swap/pim/contact#mailbox**
- valores de estas propiedades, por ejemplo, **mailto:em@w3.org** como el valor de la propiedad **mailbox** (RDF también usa cadenas de caracteres tales como "Eric Miller", y valores de otros tipos de datos como enteros y datos, o valores de propiedades)

RDF también provee una sintaxis basada en XML (llamada RDF/XML) para guardar e intercambiar estos grafos. Este ejemplo es una pequeña muestra de RDF en RDF/XML correspondiente al grafo de la ilustración anterior.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

        <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
                <contact:fullName>Eric Miller</contact:fullName>
                <contact:mailbox rdf:resource="mailto:em@w3.org"/>
                <contact:personalTitle>Dr.</contact:personalTitle>
        </contact:Person>

</rdf:RDF>
```

Este RDF/XML también contiene URIs, y propiedades como mailbox y fullName (en forma abreviada), y sus respectivos valores em@w3.org, y Eric Miller

También hemos de citar el formato llamado ***Terse RDF Triple Language*** **o Turtle** [20] que es un estándar de la W3C para escribir RDF y como tal ha sido utilizado en esta aplicación para el estudio de las distintas fuentes de datos. Es la notación más importante de RDF, captura de manera clara el grafo abstracto, provee un mecanismo

para manejo de espacios de nombres, permite abreviaturas para triplas con el mismo sujeto e introduce también abreviaturas para colecciones.

El ejemplo anterior en Turtle sería:

```
@prefix contact:  <http://www.w3.org/2000/10/swap/pim/contact#>.
<http://www.w3.org/People/EM/contact#me>     contact: fullName      “Eric Miller” ;
                                             contact:mailbox        “mailto:em@w3.org” ;
                                             contact:personalTitle  “Dr.” .
```

1.2.2. RDF SCHEMA

Al igual que con XML, con RDF también podemos utilizar esquemas (*schemas*) que describen las propiedades y las clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases, es decir, describen tanto el contenido como la estructura de la información.

La especificación *RDF Vocabulary Description Language 1.0: RDF Schema* [18] describe cómo usar el lenguaje RDF Schemas (RDFS) y ofrece un vocabulario concreto para este propósito.

Un esquema RDF es un conjunto de informaciones relativas a las clases de recursos que sirve para explicitar relaciones jerárquicas que se establecen entre ellos, o bien para matizar el carácter obligatorio u opcional de las propiedades y otras restricciones como tipos de dato, el número mínimo y máximo de ocurrencias y otras características más específicas.

En otras palabras, puede pensarse en un esquema como en una especie de diccionario o vocabularios para utilizar con RDF. Un esquema define los términos que se utilizarán en una declaración RDF y le otorgará significados específicos. Con RDF se pueden utilizar una gran variedad de formas de esquema, incluso la forma específica RDF Schema, que tiene algunas características específicas para ayudar a la automatización de tareas con RDF.

En los esquemas pueden definirse nuevos recursos como una "especialización" de los anteriores. Esta es una importante característica de RDFS dado que en ella radica la extensibilidad en cuanto a elaboración de nuevos esquemas.

Los recursos siguientes son las clases y propiedades principales que se definen como parte del vocabulario del esquema RDF.

- rdfs:Resource: todas las cosas que se describan por expresiones RDF se denominan recursos y se consideran como instancias de la clase *rdfs:Resource*.
- rdfs:Class: las Clases son recursos que denotan conjuntos de recursos. Pueden definirse para representar cualquier cosa, como páginas web, personas, tipos de documentos, bases de datos o conceptos abstractos.
- rdfs:Literal: es la clase de todos los valores literales, por ejemplo, cadenas de texto y números enteros.
- rdf:type: el sujeto es una instancia de una clase.
- rdfs:subPropertyOf: se aplica a las propiedades que pueden ser interpretadas como un subconjunto de otras propiedades. Permite definir jerarquías de propiedades.
- rdfs:subClassOf: el sujeto es una subclase de una clase; permite definir jerarquías.
- rdfs:range: especifica un rango de la propiedad del sujeto.
- rdfs:domain: especifica el dominio de la propiedad del sujeto.

El modelo RDF, extendido con el RDF Schema, es eficaz, dado que puede utilizarse como un modelo general para expresar metadatos sobre recursos Web. RDF puede ser expresado en una sintaxis XML que permite utilizarlo en muchos ambientes y plataformas. Es posible conjugar múltiples grafos RDF en uno solo y utilizar los vocabularios definidos en múltiples esquemas.

1.2.3. OWL

El Web Ontology Language (OWL) es un mecanismo para desarrollar temas o vocabularios específicos en los que asociar recursos tales como relaciones entre clases, cardinalidad, igualdad, tipologías de propiedades más complejas, caracterización de propiedades (simetría) o clases enumeradas de la Web Semántica. Lo que hace OWL es

proporcionar un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas.

En realidad, OWL es una extensión del lenguaje RDFS y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste; posee más funcionalidades para expresar el significado y semántica que XML, RDF, y RDFS, es decir, va más allá que estos lenguajes pues ofrece la posibilidad de representar contenido de la Web interpretable por máquina [21].

El Web Ontology Language OWL es, en realidad, un lenguaje de etiquetado semántico para publicar y compartir ontologías en la World Wide Web y es parte de las recomendaciones del W3C relacionadas con la Web Semántica desde el 10 de febrero de 2004 [22].

1.3. CONSULTAS DE INFORMACIÓN EN LA WEB SEMÁNTICA

Según recoge la Guía Breve que sobre la Web Semántica ha publicado el W3C [6], para obtener una adecuada definición de los datos, la Web Semántica utiliza esencialmente RDF, SPARQL y OWL, mecanismos que ayudan a convertir la Web en una infraestructura global en la que es posible compartir, y reutilizar datos y documentos entre diferentes tipos de usuarios.

Si RDF proporciona información descriptiva simple sobre los recursos que se encuentran en la Web y OWL es un mecanismo para desarrollar temas o vocabularios específicos en los que asociar esos recursos, SPARQL es un lenguaje estandarizado para la consulta.

La enorme cantidad de contenidos que alberga la web hacen necesarios lenguajes de recuperación (query languages) que permitan la consulta y recuperación de la información almacenada, es decir, un conjunto de órdenes, operadores y estructuras que, organizados según unas normas lógicas, permitan la consulta de fuentes y recursos de información electrónica.

1.3.1. SPARQL

En los últimos años los investigadores han tratado de desarrollar lenguajes de consulta que permitan ejecutar búsquedas complejas sobre un grafo RDF, mediante una sintaxis sencilla. Diferentes iniciativas han puesto en marcha diversos tipos de lenguaje que permiten ejecutar búsquedas en grafos RDF utilizando distintas fuentes de datos. Entre ellos se encuentra el ***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage* (SPARQL), clave en el desarrollo de la Web Semántica y que se constituyó como Recomendación oficial del W3C en 2008 [23], porque para que la Web Semántica sea una realidad se necesita un lenguaje de consulta estándar y un protocolo de recuperación. Esta recomendación dispone de una nueva versión desde el 21 de marzo de 2013, SPARQL 1.1 [24].

SPARQL consiste en tres especificaciones separadas, que contienen diferentes partes de su funcionalidad: un lenguaje de *query*, un formato para las respuestas, y un medio para el transporte de consultas y respuestas:

- SPARQL Query Language for RDF [25]: núcleo de SPARQL que explica la sintaxis para la composición de sentencias y su concordancia.
- SPARQL Protocol for RDF [26]: formato utilizado para la recuperación de los resultados de las búsquedas (queries SELECT o ASK), a partir de un esquema de XML.
- SPARQL Query XML Results Format [27]: Describe el acceso remoto de datos y la transmisión de consultas de los clientes a los procesadores. Utiliza WSDL para definir protocolos remotos para la consulta de bases de datos basadas en RDF.

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.

Algunas fuentes de datos tienen disponible un SPARQL endpoint, una URI para hacer consultas SPARQL sobre los datos que contienen dicha fuente. Además pueden tener una interfaz HTML para que los usuarios puedan consultar la fuente de datos sin tener que descargar toda la información de la fuente de datos.

Veamos un ejemplo del uso de SPARQL para las búsquedas sobre RDF [28]:

Este caso trata de obtener recursos cercanos al municipio de Madrid, a una distancia de 10Km (0.1) y con etiquetas en español. La consulta se limita a 50 recursos.

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
SELECT  ?subject ?label ?latitude2 ?longitude2
WHERE {
          < http://geo.linkeddata.es/resource/Municipio/Madrid> geo:geometry ?g.
          ?g geo:lat ?latitude.
          ?g geo:long ?longitude.
          ?subject geo:geometry ?g2.
          ?g2 geo:lat ?latitude2.
          ?g2 geo:long ?longitude2.
          ?subject rdfs:label ?label.
          FILTER(xsd:double(?latitude2) - xsd:double(?latitude) <= 0.1 &&
xsd:double(?latitude) - xsd:double(?latitude2) <= 0.1 &&
          xsd:double(?longitude2) - xsd:double(?longitude) <= 0.1 &&
xsd:double(?longitude) - xsd:double(?longitude2) <= 0.1 &&
          lang(?label) = "es").
} limit 50
```

1.4. LINKED DATA

La Web Semática es una web de datos.

Los datos que conforman la Web Semántica son denominados Linked Open Data (LOD) o Datos Abiertos Enlazados y reúnen unas características determinadas que les permiten ser combinados, enlazados y utilizados aún proviniendo de fuentes diferentes y ser de tipologías distintas.

Linked Data es una red mundial de datos que acumula millones de referencias organizadas por burbujas temáticas relacionadas entre sí a través de Internet. El mayor proveedor de datos de la Web de Datos es el sector público, en especial los de Reino Unido y Estados Unidos, pero se están sumando a la iniciativa otros sectores privados como los medios de comunicación y el mundo universitario y científico en especial.

El último objetivo de la Web de los datos es permitir que los equipos informáticos hagan un trabajo más útil y desarrollar sistemas que puedan soportar interacciones de confianza sobre la red. El término "Web Semántica" se refiere a la visión del W3C sobre la Web de los Datos Enlazados (Linked Data).

Para conseguir este objetivo se pueden seguir dos caminos para dotar de información a esta web, o bien podríamos enriquecer los textos HTML con anotaciones RDF que permitirían ser procesados automáticamente y que aportarían semántica y, naturalmente, información “legible por máquina” o podríamos decantarnos por utilizar Linked Open Data. La primera opción, que supone un cambio menor, tiene, sin embargo, un inconveniente y es que complica tanto la creación como el mantenimiento de las páginas web.

Linked Data se basa, a grandes rasgos, en la creación de recursos con información expresada directamente en RDF, ligados entre sí, capaces de ofrecer una representación distinta de los contenidos según el tipo de usuario que la solicita. Cada objeto dentro de un recurso Linked Data cuenta con un nombre único, su URI, que nos permite referenciarlo de forma unívoca.

La iniciativa Linked Data basa su funcionamiento en tecnologías y estándares ampliamente aceptados, cimentándose en 4 principios básicos que fueron definidos en 2006 por Berners-Lee [29] [30]:

1. Uso de URIs para referenciar todo objeto de información.

2. Utilización del protocolo HTTP para acceder a la información almacenada en las URIs.

3. Descripción de los recursos de información mediante RDF y utilización del lenguaje de consultas SPARQL para la búsqueda sobre estos repositorios.

4. Incluir enlaces a otras entidades mediante URI para potenciar el descubrimiento de nuevos elementos de información que puedan ser relevantes para el usuario.

La gran ventaja de Linked Data y su potencial residen en que posibilita utilizar y combinar datos procedentes de diferentes fuentes (otros recursos Linked Data) y, a partir de su integración, extraer nuevo conocimiento que nos daría la respuesta a las

cuestiones que le planteásemos a la Web. Como ejemplos de fuentes de datos con datos abiertos, podemos citar: DBpedia [31], Biblioteca Nacional Española [32], British National Bibliography [33], Cambridge University Library [34], Conseil Européen pour la Recherche Nucléaire, actual Organización Europea para la Investigación Nuclear (CERN) [35], DBLP (Digital Bibliography & Library Project) [36].

En la siguiente figura podemos ver los pasos del proceso en el contexto del desarrollo de una aplicación que utiliza datos públicos españoles de tres temas recogidos en los Anexos de la Directiva INSPIRE ((INfrastructure for SPatial InfoRmation in Europe), concretamente datos administrativos, hidrográficos y estadísticos que se relacionan con el fin, como decíamos, de extraer un nuevo conocimiento, en este caso concreto, establecer la relación existente entre la zona costera nacional e información relacionada con la población, desempleo e industria [37]:

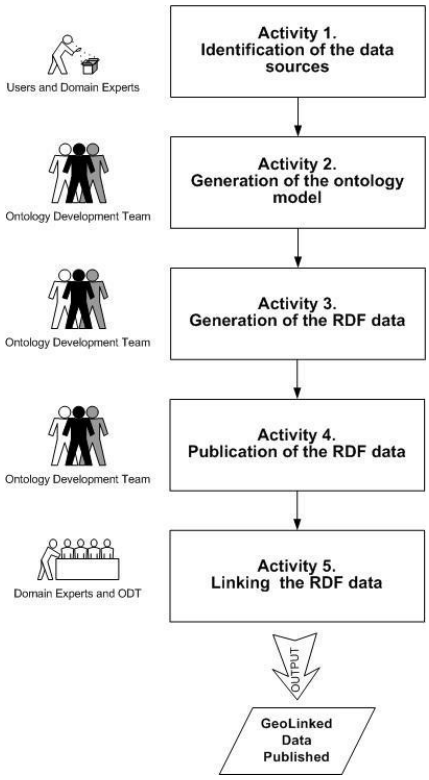

FIG 4.- EL PROCESO PARA GENERAR Y PUBLICAR GEOLINKED DATA

Así pues, la Web Semántica cuenta ya con una nutrida infraestructura de lenguajes y tecnologías que permiten publicar datos legibles por aplicaciones informáticas. La sintaxis se basa en el lenguaje XML y derivados, la semántica en los lenguajes RDF (S)

y OWL, el lenguaje de Ontologías ofrece un criterio para catalogar y clasificar la información, y también están presentes otras muchas aplicaciones y tecnologías ya desarrolladas como los URIs, etc. y numerosas empresas y centros de investigación están trabajando en ella con el fin de que la Web se convierta en la Web del conocimiento. Estamos trabajando para proporcionar una infraestructura que permita que las páginas web, las bases de datos, los programas y aplicaciones, los dispositivos, tanto los utilizados en el ámbito laboral o docente como los utilizados en el hogar, puedan consumir y producir datos, sin los problemas causados por los diferentes protocolos de acceso a la información que hacen de la transferencia de contenidos una tarea ardua y difícil.

2. OBJETIVOS

Dicen que fue el lenguaje lo que produjo el salto evolutivo de la especie humana. La escritura fue el primer método de almacenamiento y transmisión de la información. La imprenta hizo que la información pudiera distribuirse de forma masiva. Internet confiere a cada ser humano la capacidad para intervenir en este proceso desde cualquier rincón del globo. Linked Data es un valioso instrumento de gestión del conocimiento, una especie de conector inteligente entre millones de neuronas cuyas capacidades se abrirán para poder ser vistas por dentro por todas las que puedan entenderse entre sí.

El impacto de este hecho sobre la ciencia, la educación, los negocios, el arte, la vida, parece casi de ciencia ficción. Es como una especie de cerebro colectivo en donde los seres humanos podrán "pensar juntos" utilizando sus ordenadores y avanzar más deprisa en cualquier dirección utilizando una inteligencia puesta en común y esas interacciones de confianza garantizada por aplicaciones.

El almacenaje de datos es un factor que ha revolucionado el conocimiento, pero la verdadera inteligencia reside en las conexiones, conexiones que además no se limitan a "poner en contacto" u organizar en listas la información disponible, sino que sirven también para entenderse y para dar lugar a nuevas conclusiones y planteamientos que incrementen el conocimiento. Ahí está la clave, en poner en común todas las cosas y sus relaciones. Esas capacidades harán del mundo un lugar mejor, de los seres humanos

personas más libres y del futuro un tiempo apasionante, del que no sólo me gustaría beneficiarme sino al que también me gustaría contribuir. Yo veo este proyecto como mi primer paso.

Con el presente trabajo se pretende poner en valor el concepto de datos abiertos enlazados, Linked Open Data (LOD), y su relevancia en el proceso de construcción de un conocimiento cada vez más global. Para ello se ha trabajado con los siguientes objetivos concretos:

- Obtener una visión general de la evolución de la web hasta el desarrollo de la Web Semántica.

- Introducir el concepto de datos abiertos enlazados (LOD) y trabajar con ellos.

- Analizar las distintas ontologías, que sobre un mismo tema, definen diferentes fuentes de datos.

- Estudiar los vocabularios propuestos por las instituciones que definen el desarrollo de la Web Semántica y los que usan distintas fuentes de datos.

- Buscar la relación entre distintas fuentes de datos.

- Analizar fuentes de datos, estudiar su documentación, modelo de datos, problemas, virtudes…

- Como resultado de lo anterior, llevarlo a la práctica mediante la realización de una aplicación real.

Y los objetivos personales bajo los que he elegido este proyecto han sido:

- Trabajar en un campo que es interesante (o me lo parece). Interés personal en tener información útil.

- Contribuir con mi trabajo a la mejora de la documentación de las fuentes de datos de forma que el siguiente usuario disponga de un mejor conocimiento de esas fuentes.

- Mejorar la calidad de la información existente

- Colaborar al progreso de las ciencias, desde la educación primaria hasta la universidad más reputada podrían obtener ventajas usando datos más correctos, más fáciles de encontrar.

- Apreciar y poner en valor el potencial de esta tecnología.

3. DESCRIPCIÓN INFORMÁTICA

Para la realización de esta aplicación se han utilizado los lenguajes, recursos, consultas, etc., referidas en la Introducción y la utilización concreta de los activos en la aplicación desarrollada, se relata a continuación junto con el estudio de las fuentes de datos.

3.1. DEFINICIÓN DE LA APLICACIÓN: REQUISITOS

El proyecto a que hace referencia esta memoria consiste en una aplicación web de consulta de información de escritores. La información que podemos consultar se refiere a datos biográficos y bibliográficos sobre autores de textos escritos, así como las obras, de otros autores, escritas por su influencia

La utilización se realiza escribiendo el usuario el nombre del autor en la caja de texto. Según vaya escribiendo el nombre, la aplicación irá mostrando una lista de los posibles autores a los que se puede referir.

Una vez el usuario elija un autor, la aplicación buscará toda la información en las fuentes de datos, la formateará y la mostrará en la web.

Además el usuario tiene la posibilidad de actualizar el listado de autores. Para esto con sólo pulsar un botón la aplicación realizará una petición a un servicio web que se conectará a la DBpedia, fuente de datos de la que obtenemos dicho listado.

El proyecto había de cumplir un requisito obligatorio del cual derivan el resto de funcionalidades: *El proyecto consiste en construir una aplicación (Web o móvil) que haga uso de varias fuentes de datos públicamente disponibles en la Web de datos (Web of Linked Data). La aplicación debe mezclar datos procedentes de al menos tres fuentes de la Web de datos.*

Requisitos funcionales:

- Aplicación de escritorio o web: las aplicaciones web presentan una serie de ventajas con respecto a las aplicaciones de escritorio que fueron decisivas a la hora de elegir qué tipo de aplicación realizar, podemos destacar entre ellas la compatibilidad multiplataforma, el mejor aprovechamiento del hardware, así como el menor tiempo de desarrollo.
- Usar varias fuentes de datos de la web de datos: requisito impuesto por las normas para la realización del proyecto y así elegiremos tres fuentes de datos.
- Relacionar las fuentes de datos: se relacionarán las tres fuentes de datos entre sí obteniendo una visión más completa del mundo bibliográfico. Esta es una de las virtudes de Linked Data.
- Listado de autores: la lista de autores de una de las fuentes de datos tiene un tamaño de 100.000 elementos. A pesar de ser un listado tan extenso se buscará una forma eficaz de mostrar tantos elementos.
- Actualización de la lista de autores: para que la aplicación mantenga su vigencia se dotará a la misma de un mecanismo de incorporación de nuevos autores.
- Asimismo se mostrará información actualizada y útil de cada uno de los autores.

Requisitos no funcionales

- La aplicación deberá ser muy eficiente, con un tiempo de reacción mínimo.
- Para una mejor experiencia del usuario (UX), la aplicación cargará dinámicamente la información.
- Será una aplicación fácil de operar.

3.2. FUENTES DE DATOS

Una vez seleccionado el tema del que trataría la aplicación, escritores en un sentido amplio de la expresión como vimos en la Introducción, el siguiente paso fue buscar las fuentes de datos de donde se extraería dicha información.

Para la búsqueda de las fuentes de datos que se utilizarían en el proyecto, se usó la web *The Data Hub* [38]. En esta web, buscando por temas, podemos encontrar todas las fuentes de datos disponibles. Además muestra información sobre los formatos en que se

pueden encontrar, si tienen disponible SPARQL endpoint y también si la información es pública; es decir, satisface los principios de Open Data.

Las fuentes de datos seleccionadas fueron DBpedia, Biblioteca Nacional Española y British National Bibliography. No fue necesario buscar DBpedia en *The Data Hub*; y al ser la principal fuente de datos resultaba indispensable incluirla en el proyecto. La British National Bibliography (BNB) fue seleccionada por su excelente documentación. En su web se puede encontrar el modelo de datos y el esquema que se utilizan para la representación de la información, además de los vocabularios que contienen las propiedades que describen los datos.

La Biblioteca Nacional Española (BNE) fue elegida por lo completo que es su extensísimo catálogo que incluye todos los libros de su fondo bibliográfico.

Se estudiaron otras fuentes de datos pero resultaban demasiado especializadas en un campo como por ejemplo la The Internet Speculative Fiction Data que contiene abundante bibliografía pero exclusivamente de temas relacionados con la ciencia ficción y el terror y la English Language Books listed in Printed Book Auction Catalogues from 17th Century Holland que sólo reúne libros holandeses del siglo XVII y otras; por tanto, fueron descartadas.

3.2.1. DBPEDIA

Wikipedia se ha convertido en una de las fuentes de conocimiento fundamentales de la humanidad, mantenida y alimentada por miles de colaboradores.

El proyecto DBpedia [31], o versión semántica de la Wikipedia, aprovecha la gigantesca fuente de conocimientos que esta contiene posibilitando la extracción de la información estructurada que contiene en los *infoboxes*. DBpedia permite realizar consultas sofisticadas a la Wikipedia.

Surge en 2007 de la colaboración de la University of Berlin, la University of Leizpig y OpenLink Software y ofrece de manera abierta sus datos para que puedan ser utilizados y enlazados con otras fuentes creando vínculos entre las distintas fuentes de datos de la Web y los datos de la Wikipedia.

Las consultas realizadas a la DBpedia son de dos tipos:

- Listado de todas las personas de tipo *autor* o de tipo *subclase de autor*. De esta consulta obtenemos el nombre del autor, además de su URI en DBpedia.
- Datos biográficos acerca del autor, como por ejemplo fechas y lugares de nacimiento y defunción, descripción de la vida del autor y foto.

El siguiente gráfico muestra la parte del modelo de datos de la DBpedia utilizado para la realización del proyecto.

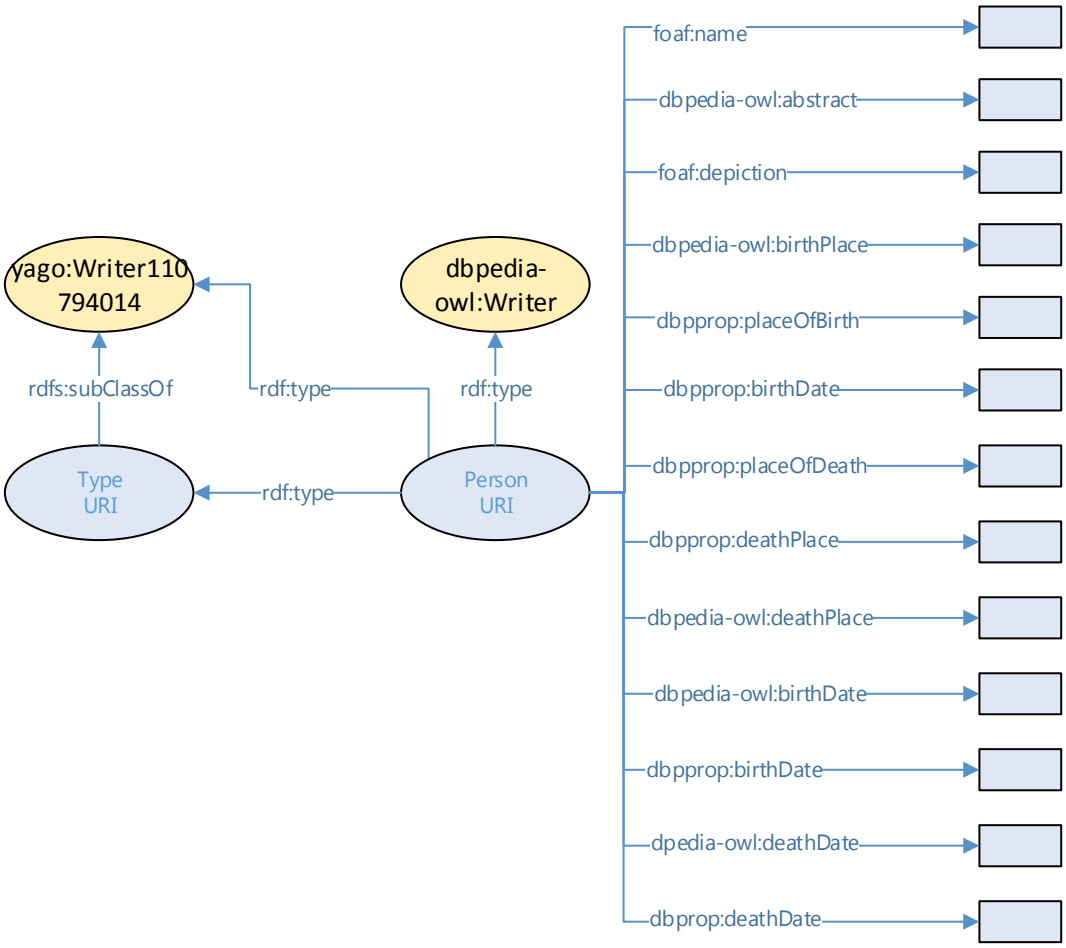

FIG 5.- SUBCONJUNTO DEL MODELO DE DATOS DE LA DBPEDIA

La primera dificultad encontrada fue cómo filtrar todas las personas que aparecen en la DBpedia para averiguar cuáles son escritores ya que existen varias formas en las que se puede encontrar el tipo *autor de libro*. Concretamente se encontraron escritores bajo más de 1800 tipos: escritores ingleses de relatos, poetas gallegos, escritores en lengua española, escritores mejicanos…

La propiedad utilizada para filtrar todos los autores es *rdf:type*. Una persona será un autor cuando *rdf:type* tome el valor: *dbpedia-owl:Writer*[1] ó *yago:Writter110794014*[2].

Sin embargo al hacer comprobaciones con diversos autores conocidos, éstos no aparecían en el listado. Por ejemplo, Valle-Inclán no aparecía como escritor. Para encontrar la causa se procedió a buscar el valor de la propiedad *rdf:type* del recurso Valle-Inclán, encontrado lo siguiente:

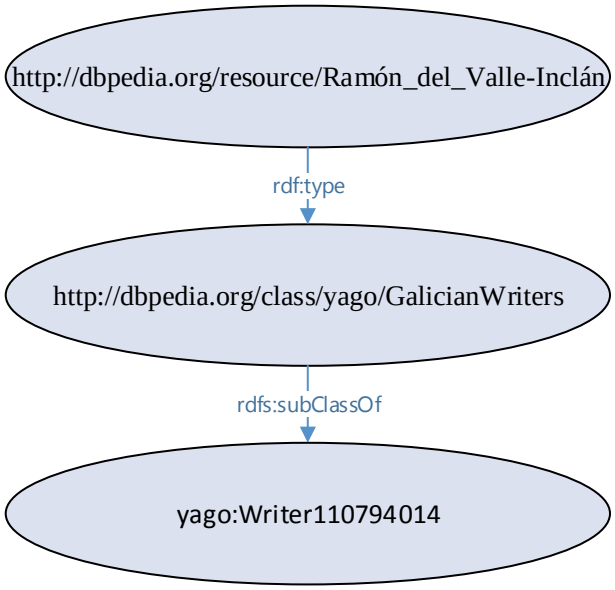

FIG 6.- PROPIEDAD *rdf:type* DEL RECURSO VALLE-INCLÁN

Repetido el proceso con varios autores llegamos a la conclusión de que el tipo *yago:Writer110794014* tiene 1843 subclases que, unidas al *dbpedia-owl:Writer*, encuadraban a todos los escritores consultados.

Ejemplos de estas subclases son: *yago:EnglishShortStoryWriters*, *yago:Spanish-languAgeWriters*, *yago:MexicanWritters*, *etc.*

[1] dbpedia-owl: http://dbpedia.org/ontology/
[2] yago: http://dbpedia.org/class/yago/

3.2.2. BIBLIOTECA NACIONAL ESPAÑOLA

La Biblioteca Nacional Española [32] contribuye a la Web Semántica con el programa *Datos enlazados en la BNE.* Esta iniciativa consiste en la transformación y publicación de los catálogos bibliográfico y de autoridades de la Biblioteca Nacional en formato RDF y conecta además, los mencionados catálogos con los registros del proyecto *Virtual International Authority File* (VIAF) [39], en el que participan una veintena de instituciones internacionales, entre ellas las bibliotecas nacionales de Francia, Alemania y del Congreso de los Estados Unidos, y al que la Biblioteca Nacional española se adhirió en el año 2009. Los recursos de la BNE se vinculan con otros conjuntos de información de la "nube" de Linked Open Data, como DBpedia.

Con esta iniciativa, la BNE se suma al reto de publicar los datos bibliográficos en RDF, siguiendo los principios de Linked Data y bajo la licencia abierta de CCo (Creative Commons Public Domain Dedication). Además, estos datos se interrelacionan con otras bases de conocimiento existentes en la iniciativa Linking Open Data.

La Biblioteca Nacional de España ha extraído 3.900.000 de registros de recursos bibliográficos, pertenecientes a monografías modernas, antiguas, recursos electrónicos, manuscritos, publicaciones periódicas, mapas, grabados, fotografías, música impresa, grabaciones sonoras y audiovisuales.

Una característica de esta fuente de datos es que utiliza las ontologías de la Federación Internacional de Asociaciones de Bibliotecarios y Bibliotecas (IFLA en sus siglas inglesas). Dichos vocabularios, a su vez, se han elaborado a partir de la Descripción Bibliográfica Internacional Normalizada (ISBD) y de los Requerimientos Funcionales para Registros Bibliográficos (FRBR). Esta normalización catalográfica para la adecuada gestión de los recursos bibliográficos, hace que los datos de la BNE no tengan una compresión directa para un usuario común sino limitada a los profesionales del ámbito bibliotecario ya que utiliza códigos reglados. En este sentido hay que señalar que el W3C consciente del importantísimo papel que los datos bibliotecarios pueden jugar en el marco de Linked Data, ha mantenido entre 2010 y 2011 el W3C Library Linked Data Incubator Group, cuya actividad, recomendaciones y resultados publicó en agosto de 2011 [40]. Esta publicación contiene claras y contundentes afirmaciones sobre

las acciones que deben desarrollar las bibliotecas si quieren integrar sus conjuntos de datos en la Web Semántica a través de Linked Data.

Así por ejemplo la tripla de Miguel de Cervantes en tanto autor de la obra *El Quijote* es:

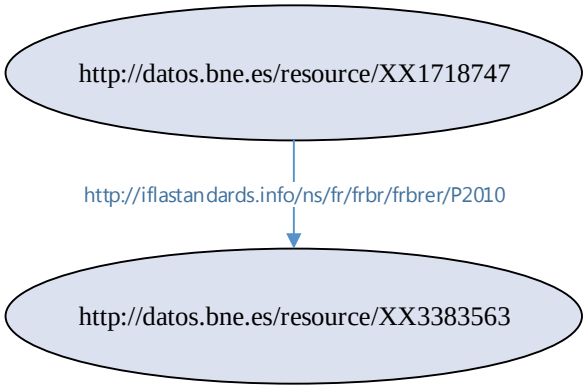

FIG 7.- EJEMPLO DE TRIPLETA DE LA BNE

La tripla anterior representa a:

- Miguel de Cervantes (http://datos.bne.es/resource/XX1718747)
- es el creador de (http://iflastandards.info/ns/fr/frbr/frbrer/P2010)
- El Quijote (http://datos.bne.es/resource/XX3383563).

En el cuadro siguiente vemos como al no mostrarse las etiquetas de los nombres de las propiedades, no resulta directamente comprensible.

Property	Value
isbd:P1004	El ingenioso hidalgo Don Quijote de la Mancha
isbd:P1007	Miguel de Cervantes
isbd:P1008	Ed. crema
isbd:P1016	Madrid
isbd:P1018	[1946?]
isbd:P1019	Madrid
isbd:P1020	Imp. de Federico Domenech
isbd:P1022	504 p., [1] h. de lám.
isbd:P1024	15 cm
isbd:P1068	Don Quijote de la Mancha
isbd:P1117	Librería Beltrán
isbd:P1185	[Texto impreso]
ifla-frbr:P2004	bne:resource/XX3383563spa
dcterms:language	<http://lexvo.org/id/iso639-3/spa>
rdf:type	ifla-frbr:C1003

FIG 8.- PROPIEDADES DE *DON QUIJOTE* EN LA BNE

De todos los recursos y propiedades que posee la BNE, el siguiente grafo representa la parte del modelo de datos que se ha utilizado para la realización de este proyecto:

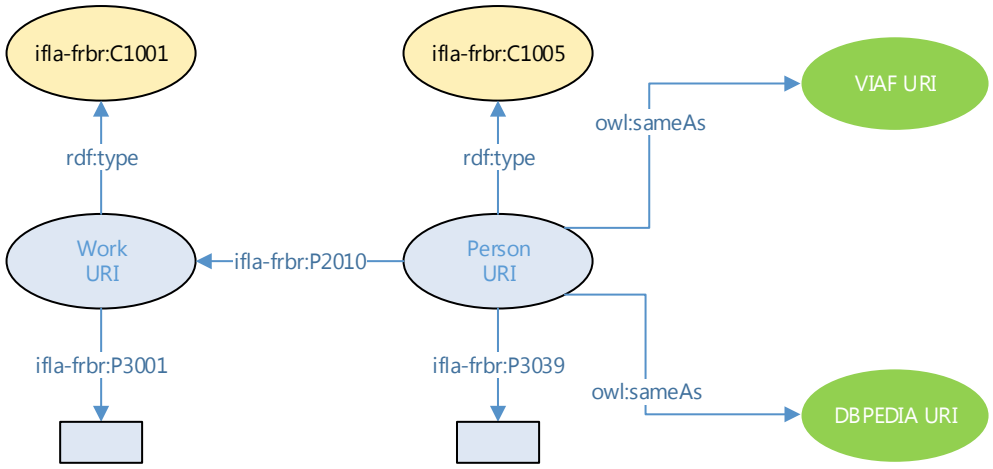

FIG 9.- SUBCONJUNTO DEL MODELO DE DATOS DE LA FUENTE DE DATOS BNE

Código	Significado
ifla-frbr:C1001	Obra
ifla-frbr:C1005	Persona
ifla-frbr:P3001	Tiene de título
ifla-frbr:P3039	Tiene de nombre
ifla-frbr:P2010	Relaciona una persona con la obra que ha creado
owl:sameAs	Los identificadores se refieren al mismo recurso.

FIG 10.- CÓDIGOS DE LA FIGURA 9

Aunque existe la posibilidad de descargar todos los datos de esta fuente y trabajar en local [41], al tener demasiada información los ficheros descargados, la aplicación era ineficiente, por lo que se decidió utilizar su SPARQL endpoint para la consulta de datos. Este método proporciona otra ventaja al permitirnos trabajar con información siempre actualizada.

3.2.3. BRITISH NATIONAL BIBLIOGRPHY

La fuente de datos British National Bibliography [33] contiene los libros publicados, incluyendo monografías y publicaciones periódicas con futuras entregas, publicadas en Reino Unido desde 1950. Contiene aproximadamente 2.8 millones de registros, generando 89 millones de triplas.

La BNB posee numerosos enlaces a recursos externos incluidos los que nos permiten conectarnos a la VIAF [39], a la *Library of Congress Authorities* (LCSH) [43], a Lexvo [44], GeoNames, RDF Book Mashup, y otros. El modelo de datos de la BNB se encuentra muy viene bien definido [45].

El siguiente grafo representa la parte de este modelo de datos utilizado para el desarrollo de la aplicación:

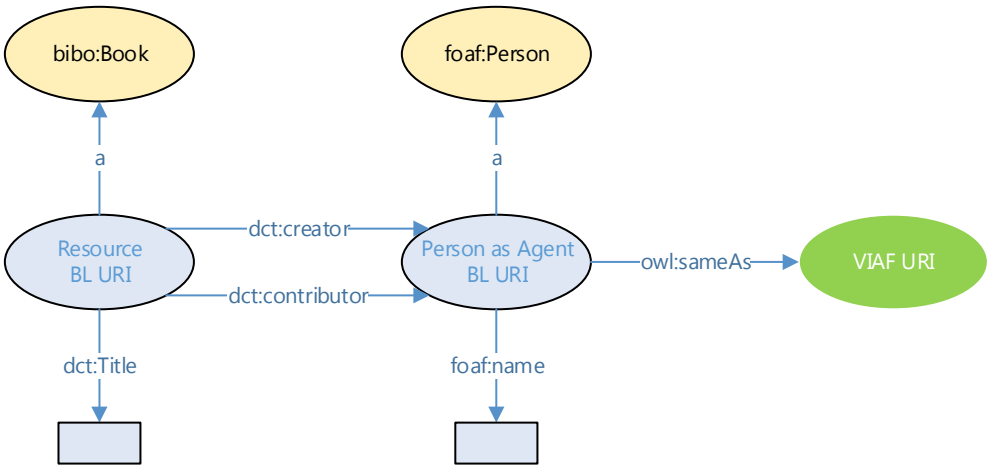

FIG 11.- SUBCONJUNTO DEL MODELO DE DATOS DE LA FUENTE DE DATOS BNB

3.2.4. RELACIÓN ENTRE LAS FUENTES DE DATOS

La propiedad clave que se ha utilizado para relacionar las tres fuentes de datos usadas en la aplicación es *owl:sameAs*. Esta propiedad determina que dos recursos de la web de datos son el mismo. Aparece tanto en BNE como en BNB.

El proceso de relación es el siguiente: En un primer paso obtenemos todos los escritores que contiene la DBpedia, con su identificador de recurso en esta fuente de datos. Al

obtener este identificador buscamos en la BNE, el autor cuya propiedad *owl:sameAs* toma el valor del identificador de DBpedia. Con esto sabremos que el autor de la DBpedia con identificador X es el mismo que el autor de la BNE cuya propiedad *owl:sameAs* toma valor X. En tercer lugar para relacionar estas dos fuentes de datos con la tercera usamos su identificador en VIAF.

VIAF es un proyecto internacional, liderado por la Library of Congress, la OCLC, y las bibliotecas nacionales de Francia y Alemania y al que la Biblioteca Nacional Española se adhirió en 2009. VIAF ha creado un identificador único, un URI, por cada uno de los autores que aparecen en las bibliotecas nacionales. Este identificador además se conecta a los registros individuales que para ese autor tienen todas las bibliotecas nacionales. En VIAF únicamente se accede al contenido del autor, no a registros bibliográficos.

Este identificador de VIAF lo contiene la propiedad *owl:sameAs* tanto de la BNE como de la BNB. En el proyecto hemos utilizado este identificador para relacionar estas dos fuentes de datos: Obtenemos el identificador en VIAF de la BNE, y con él hacemos una consulta en BNB buscando el autor cuya propiedad *owl:sameAs* coincide con el identificador en VIAF anteriormente obtenido. La representación gráfica de esta relación sería:

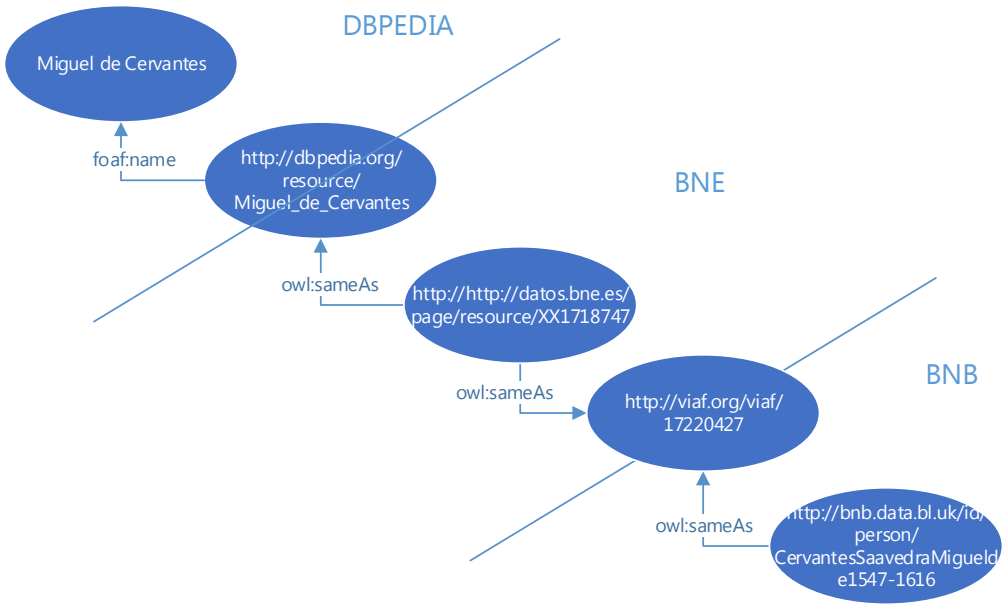

FIG 12.- RELACIÓN ENTRE LAS FUENTES DE DATOS DEL PROYECTO

3.3. ANÁLISIS

En el proceso de análisis, antes de definir claramente la aplicación que pretendíamos crear y los componentes que la iban a integrar, se realizó un diagrama de flujo de datos (DFD) con el fin de delimitar la frontera entre el sistema y el mundo exterior, y para definir sus interfaces, es decir los flujos de datos de entrada y salida del sistema con el entorno.

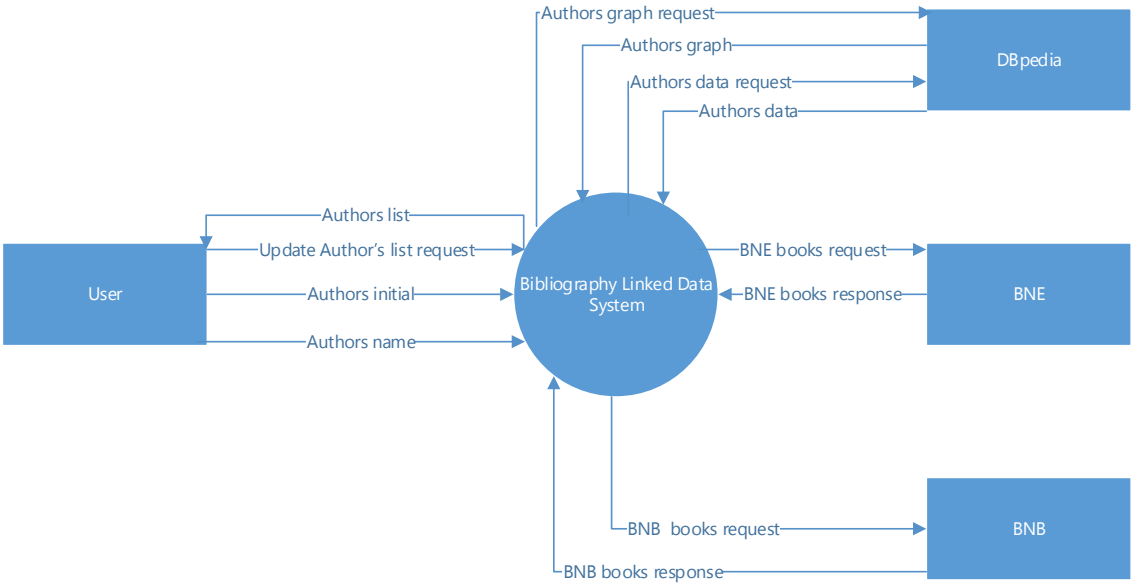

FIG 13.- DIAGRAMA DE FLUJO DE DATOS

También se realizó el diagrama de sistemas, obtenido de la descomposición del diagrama de contexto en el que podemos observar las funciones principales del sistema entre las que destacamos:

- Descargar la lista de autores
- Crear el listado de autores
- Obtener la información completa del autor, su información biográfica y bibliográfica.

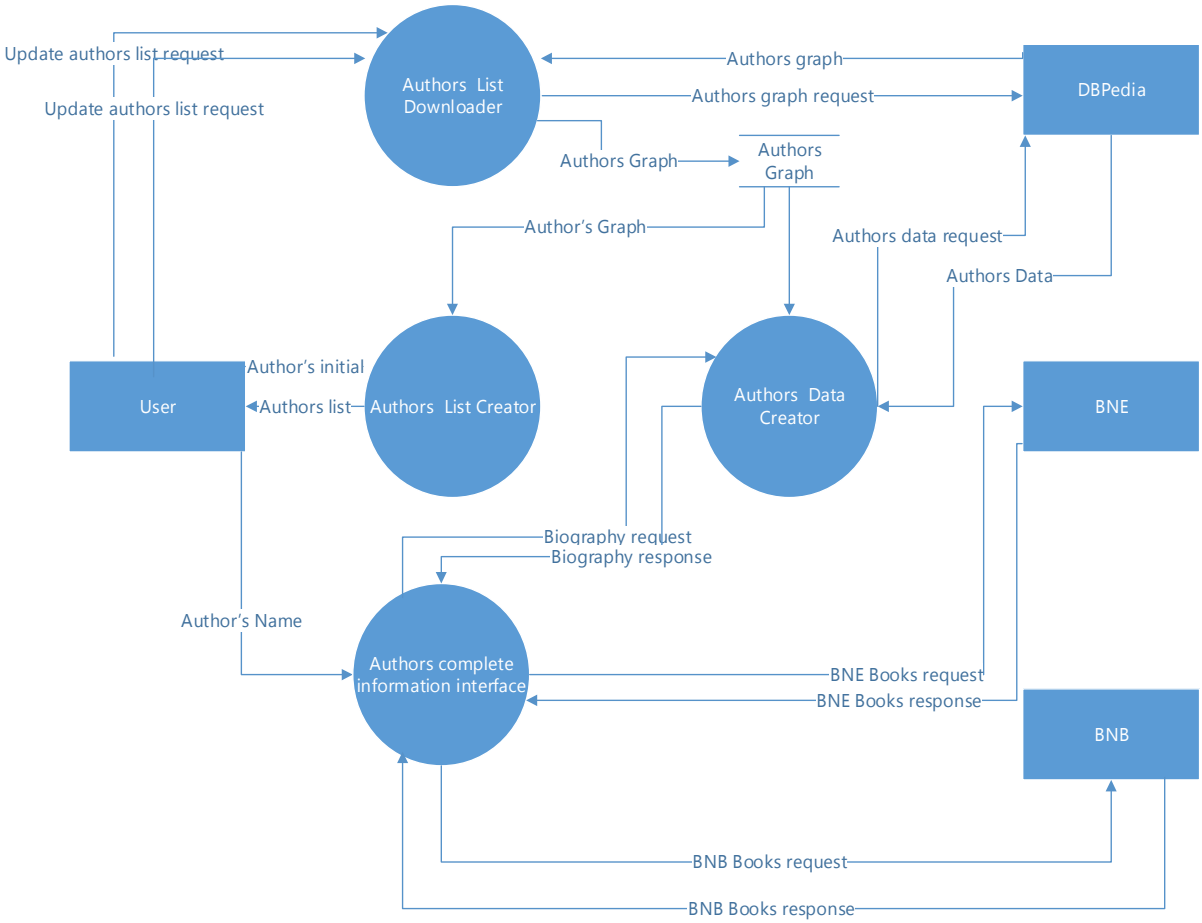

FIG 14.- DIAGRAMA DE SISTEMAS

Para definir claramente el sistema que deseamos crear así como para identificar los componentes principales que lo integrarán, es importante analizar los casos de uso.

De la captura de requisitos podemos concluir tres casos de uso:

- Descargar la lista de autores.
- Mostrar la lista de autores.
- Buscar un autor concreto.

Camino básico	
ACTOR (Usuario de la aplicación)	**SISTEMA**
1. El usuario hace click en "Update author's list"	**2.** Sistema hace una petición al subsistema encargado de la descarga.
	3. Subsistema hace la petición a DBpedia para construir el esquema RDF que contendrá todos los identificadores de escritores + sus nombres
	4. Subsistema recibe la información y la almacena en ficheros distintos dependiendo de la inicial del nombre.
	5. Subsistema manda un mensaje de OK al usuario.
Camino alternativo	
	4. Subsistema no recibe la información de los autores
	5. Subsistema envía un mensaje de error al usuario

FIG 15.- FLUJOS DE EVENTOS DEL CASO DE USO *DESCARGA DE LA LISTA DE AUTORES*

Camino básico	
ACTOR (Usuario de la aplicación)	**SISTEMA**
1. El usuario introduce el nombre del autor o parte de él	**2.** El sistema selecciona la primera letra introducida.
	3. El sistema abre el fichero que corresponde a los nombres de autor cuya inicial coincide con la introducida por el usuario y lo almacena en caché para futuras búsquedas.
	4. El sistema muestra en la caja de texto los posibles autores a los que se puede referir el usuario.
Camino alternativo	
1. La información introducida no coincide con ningún autor. P. ej. introduce números.	**2.** El sistema selecciona el primer carácter introducido.
	3. El sistema busca el fichero de esa inicial, pero no lo encuentra.
	4. El sistema no muestra ninguna lista.

FIG 16.- FLUJO DE EVENTOS DEL CASO DE USO *MOSTRAR LA LISTA DE AUTORES*

Camino básico	
ACTOR (Usuario de la aplicación)	**SISTEMA**
1. El usuario introduce un nombre completo de autor o selecciona un nombre de la lista y pulsa en buscar.	**2.** El sistema busca el identificador de DBpedia en el fichero con la lista de autores o en caché.
	3. El sistema busca en DBpedia la información biográfica del autor.
	4. Con el identificador de DBpedia, el sistema busca en BNE la lista de libros que ha escrito, además recupera el identificador en VIAF de dicho autor.
	5. El sistema busca en BNB el autor cuyo identificador en VIAF coincide con el anteriormente obtenido y la información de los libros en los que dicho autor ha influido.
	6. El sistema formatea toda la información obtenida y la muestra al usuario.
Camino alternativo	
1. El usuario introduce un nombre completo de autor o selecciona un nombre de la lista y pulsa en buscar	**2.** El sistema busca el identificador de DBpedia en el fichero con la lista de autores o en caché y no lo encuentra.
	3. El sistema muestra un mensaje *No matches found* (No se han encontrado coincidencias.)

FIG 17.- FLUJO DE EVENTOS DEL CASO DE USO *BUSCAR UN AUTOR CONCRETO*

3.4. DISEÑO

Para dar soporte a todos los requisitos tanto funcionales como no funcionales de la aplicación, se han desarrollado diversos diagramas que explicaremos en esta sección. Estos diagramas nos permitirán identificar las interfaces entre los subsistemas, y nos facilitarán el primer paso hacia la implementación.

3.4.1. CASO DE USO *DESCARGA DE LA LISTA DE AUTORES*

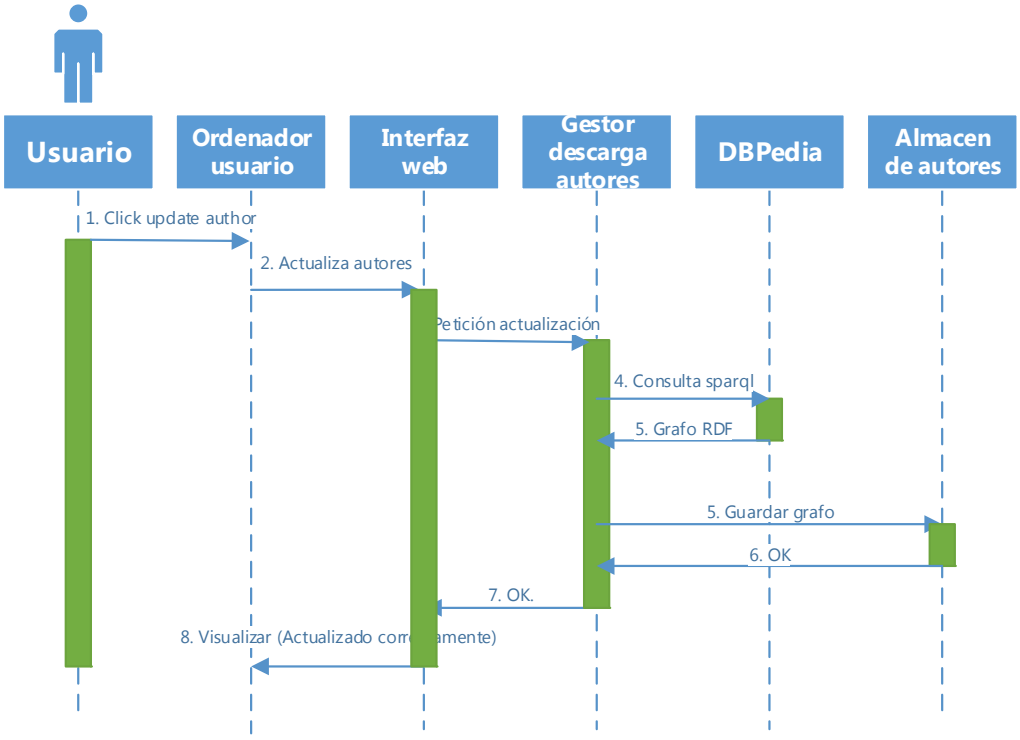

FIG 18.- DIAGRAMA DE SECUENCIA DEL CASO DE USO *DESCARGA DE LA LISTA DE AUTORES*

Este caso de uso se inicia cuando el usuario hace click en el botón *Update author's list*. La aplicación llamará al módulo *Gestor descarga autores*. Será un servicio web que tendrá disponible un endpoint. Al llamar a este endpoint el servicio web se encargará de hacer una consulta SPARQL a la DBpedia [46], construyendo un modelo en RDF con los identificadores del recurso autor, además de la propiedad nombre. Este módulo también se encargará de transformar este modelo en varios submodelos, uno por cada

letra del alfabeto. En cada submodelo se guardarán los autores cuyo nombre empieza por esa inicial. Los modelos se guardarán en un almacén de autores, que será un fichero de texto (uno por cada modelo). Así el acceso a la lista de autores será muy rápido aún teniendo gran cantidad de datos (100.000 autores).

En este diagrama aparece el primer subsistema que deberemos implementar, la interfaz web, que será la forma de comunicación con el usuario. Además aparece el gestor de descarga de autores, que nos permitirá llevar a cabo 3 de los requisitos:

- Obtener un listado de autores,
- Tener un mecanismo de actualización de la lista
- Guardar estos datos en local.

3.4.2. CASO DE USO *MOSTRAR LA LISTA DE AUTORES*

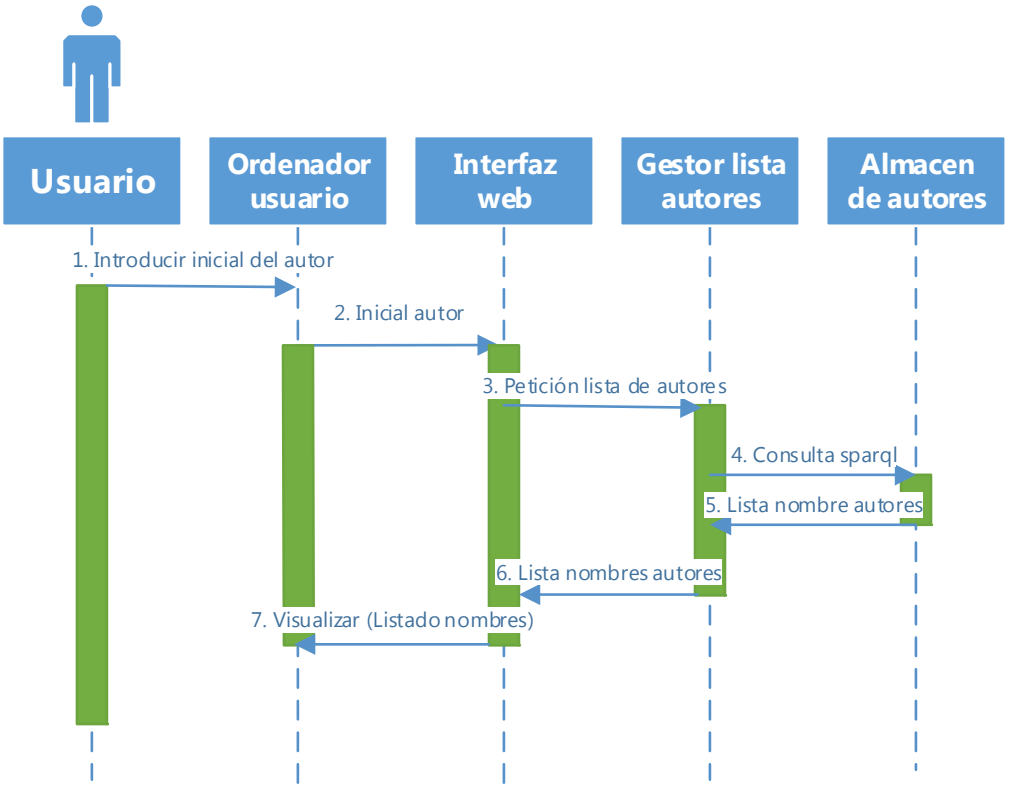

FIG 19.- DIAGRAMA DE SECUENCIA DEL CASO DE USO *MOSTRAR LA ISTA DE AUTORES*

Para llevar a cabo este caso de uso creamos un nuevo subsistema, *Gestor de la lista de autores*, cuya funcionalidad principal sería recuperar la lista del almacén de autores para que la interfaz pueda, posteriormente, darle el formato adecuado y mostrarla al usuario.

El caso de uso *Mostrar lista de autores*, comienza cuando el usuario teclea un carácter en el cuadro de texto. En ese momento la interfaz pedirá al módulo *Gestor lista de autore*s la lista de autores cuya inicial comienza con el carácter introducido por el usuario. El *Gestor lista de autores* se encargará primero de buscar en caché el listado. En caso de encontrarlo lo devolverá a la interfaz y esta se encargará de ir autocompletando el nombre introducido por el usuario con los posibles valores de la lista. En caso de que el listado no se encuentre en caché, el módulo *Gestor lista de autores*, abrirá el fichero de la inicial correspondiente y hará una consulta SPARQL pidiendo todos los nombres que contiene. Una vez se obtenga esta lista se guardará en caché para futuras consultas, y se devolverá la lista a la función de autocompletado siguiendo los pasos del caso anterior.

3.4.3. CASO DE USO *BÚSQUEDA DE UN AUTOR CONCRETO*

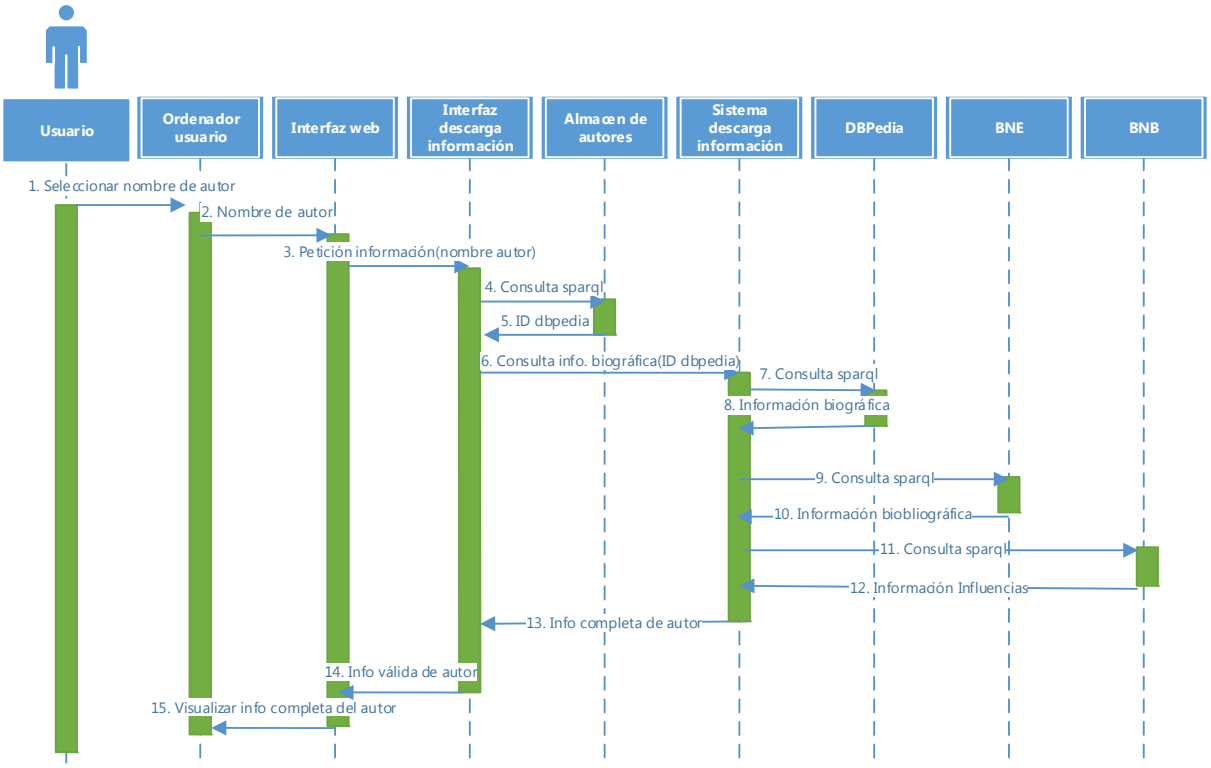

FIG 20.- DIAGRAMA DE SECUENCIA DEL CASO DE USO *DESCARGA DE UN AUTOR CONCRETO*

Una vez el usuario ha seleccionado un nombre de autor, el módulo *interfaz de descarga de información* se encargará de recuperar el identificador de DBpedia del autor. Para ello primero consultará la caché. En caso de no encontrarlo, hará una consulta SPARQL al almacén de autores (ficheros de texto correspondiente). Una vez obtenga este identificador, hará una petición al *sistema de descarga de información* que se encargará de recuperar la información biográfica mediante una consulta SPARQL a la DBpedia. Además este *sistema de descarga de información* será el encargado de recuperar la información bibliográfica de la Biblioteca Nacional Española, también consultando el SPARQL endpoint [47].

Por último recuperará la información de libros escritos bajo la influencia de este autor y el nombre de la persona que los escribió. Cuando ha recogido toda la información, esta será devuelta a la interfaz de descarga, que será la encargada de formatearla y filtrarla para finalmente mostrársela al usuario.

Del diseño de este caso de uso, observamos dos subsistemas:

- *interfaz de descarga de información*, encargada de recuperar el identificador de DBpedia y del formateo y tratamiento de los datos.
- *sistema de descarga de información,* encargado de la recuperación de la información de diversas fuentes.

3.5. DISTRIBUCIÓN DE LAS FUENTES DE DATOS EN LA APLICACIÓN

En este apartado se especificará de qué fuente de datos proviene la información que se muestra en la aplicación.

El listado de autores proviene de la DBpedia. En la aplicación desarrollada el listado de autores aparece donde se señala en la siguiente imagen, apareciendo junto al listado el nombre de la propiedad.

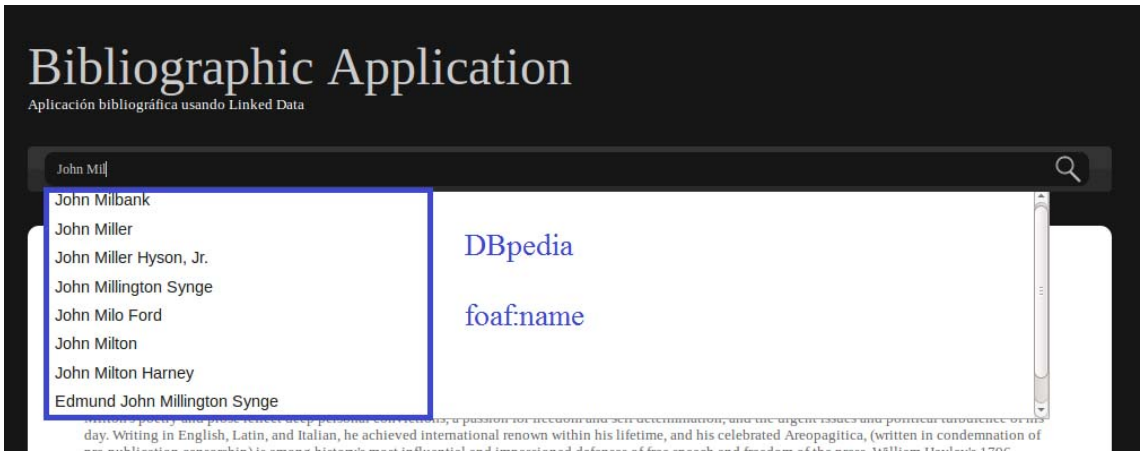

FIG 21.- LISTA DE AUTORES PROVENIENTE DE LA DBPEDIA

La información biográfica también proviene de la DBpedia. Para obtenerla se hace una consulta a su SPARQL endpoint, consultando las siguientes propiedades:

- foaf:name: nombre del autor
- dbpedia-owl:abstract breve descripción del autor
- foaf:depiction: foto del autor
- dbpprop:placeOfBirth: lugar de nacimiento del autor
- dbpprop:birthPlace: lugar de nacimiento del autor
- dbpedia-owl:birthPlace: lugar de nacimiento del autor
- dbpprop:placeOfDeath: lugar dónde murió el autor
- dbpprop:deathPlace: lugar dónde murió el autor
- dbpedia-owl:deathPlace: lugar dónde murió el autor
- dbpedia-owl:birthDate: fecha de nacimiento del autor
- dbpprop:birthDate: fecha de nacimiento del autor
- dbpedia-owl:deathDate: fecha de la muerte del autor
- dbpprop:deathDate: fecha de la muerte del autor

Como se observa en este listado, se pide la misma información por medio de propiedades distintas. Esto se debe a que no existe una uniformidad de criterio a la hora de adjudicar la misma propiedad a la misma información. Por ejemplo la fecha de nacimiento de algunos autores aparece como *dbpprop:birthDate*, mientras que en otros

aparece como *dbpedia-owl:birthDate*. Para poder obtener la información de todos los autores se les requiere bajo todas las posibilidades encontradas para asegurarnos de que la aplicación siempre nos la devolverá.

En la siguiente imagen podemos observar donde aparece en la aplicación la información biográfica consultada a la DBpedia.

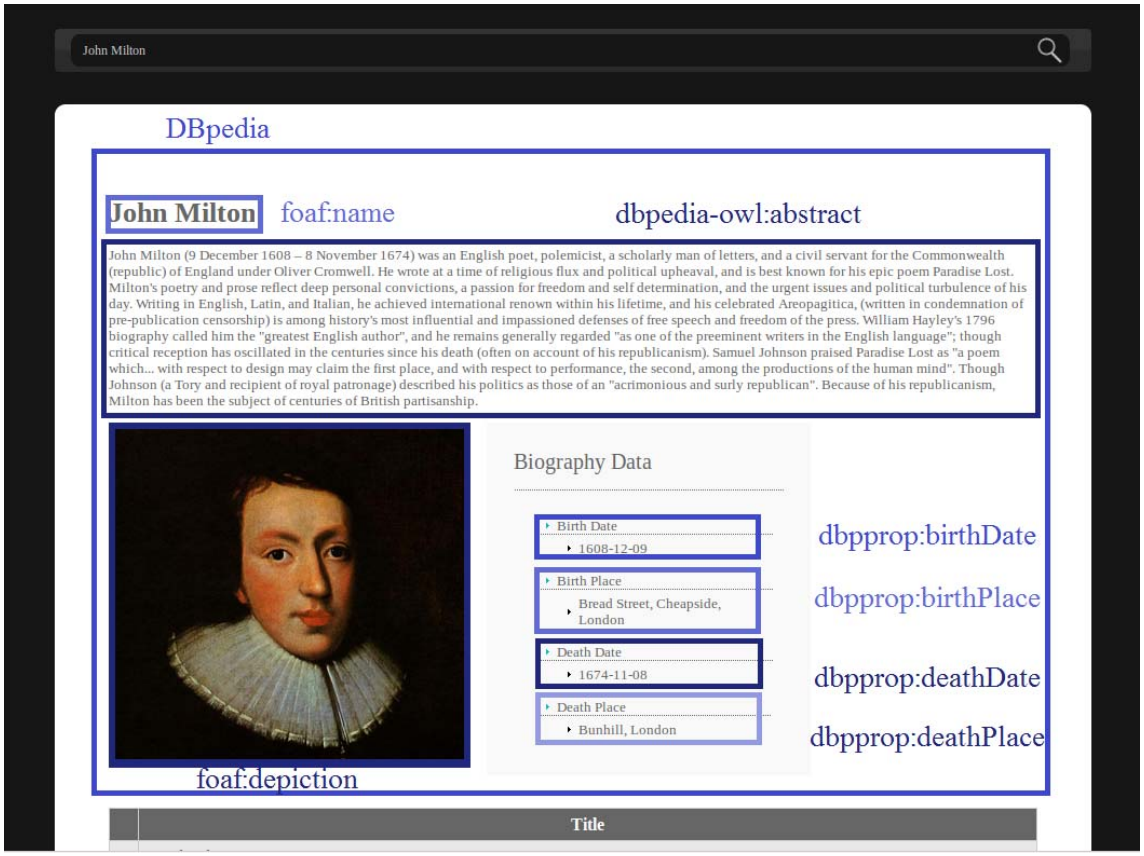

FIG 22.- INFORMACIÓN DE AUTOR PROVENIENTE DE LA DBPEDIA

La segunda fuente de datos, BNE, nos proporciona la bibliografía del autor, es decir, todos los libros que, para cada autor, contiene el catálogo de la Biblioteca Nacional Española.

La información de la biblioteca nacional española se obtiene mediante la consulta a su SPARQL endpoint.

En el siguiente cuadro se reflejan las propiedades de esta fuente usadas en este proyecto y su traducción al lenguaje natural.

IFLA Standard ID	Etiqueta	Significado
http://iflastandards.info/ns/fr/frbr/frbrer/P2010	Is creator person of	Es la persona creadora de
http://iflastandards.info/ns/fr/frbr/frbrer/P3039	Has name of person	Tiene nombre de persona
http://iflastandards.info/ns/fr/frbr/frbrer/P3001	Has title of work	Tiene título de obra
owl:sameAs	The property that determines that two given individuals are equal.	Dos individuos son el mismo (dos autores de distintas fuentes de datos son el mismo).

FIG 23.- CÓDIGOS IFLA Y SU SIGNIFICADO

En la siguiente imagen aparece el lugar en la aplicación donde se muestra la información consultada a la BNE. En este caso también aparece el nombre de la propiedad de donde se sacaron los títulos de las obras.

FIG 24.- INFORMACIÓN BIBLIOGRÁFICA PROVENIENTE DE LA BNE

La tercera fuente de datos consultada es la British National Bibliography.

La información que usamos de esta fuente de datos son las obras en las que los autores han influido y los autores de esas obras. Por ejemplo “Stories from Don Quixote” escrito por John Lang, escrito, evidentemente bajo la influencia de Cervantes.

Las propiedades utilizadas para la consulta de la BNB son:

Propiedad	Significado	Valor
blt:hasContributedTo	Ha contribuido a	Obras en las que el autor ha contribuido
owl:sameAs	Equivale a	Identificador VIAF
foaf:name	nombre	Nombre del autor
bibo:book	libro	Tipo libro
dct:title	título	Título del libro
foaf:Person	Persona	Tipo persona

FIG 25.- PROPIEDADES PROVENIENTES DE LA BNB

En este caso también se ha utilizado el SPARQL endpoint [48] para la consulta de datos. Asimismo es posible la descarga de esta fuente de datos en *The Data Hub* [49].

La siguiente imagen muestra qué parte de la información mostrada en la aplicación proviene de esta fuente de datos, y de qué propiedades provienen.

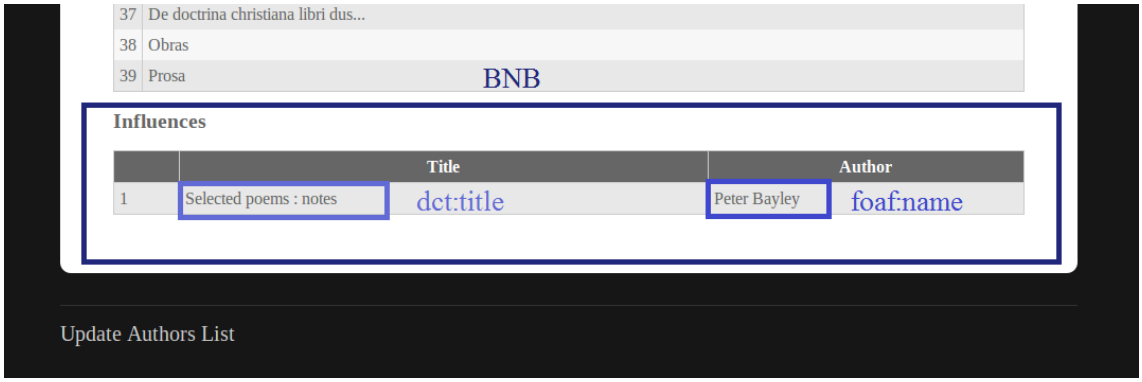

FIG 26.- INFORMACIÓN BIBLIOGRÁFICA PROVENIENTE DE LA BNB

3.6. IMPLEMENTACIÓN

En este apartado podremos ver la arquitectura del sistema desarrollado, la estructura del código, las tecnologías empleadas para su desarrollo y algunos datos relevantes sobre la implementación del sistema como por ejemplo, las consultas SPARQL realizadas a las distintas fuentes de datos.

3.6.1. ESQUEMA TECNOLOGÍAS

A continuación haremos una breve descripción de las tecnologías empleadas, sus sitios web y dónde o para qué fueron usadas dentro del código.

- RAP API [50]: RAP es una librería desarrollada en PHP para parsear, consultar, manipular serializar y servir modelos RDF. Además tiene un cliente para consultar SPARQL endpoints. Esta librería se ha utilizado en el proyecto para hacer consultas a SPARQL endpoints, para leer grafos RDF y hacer consultas sobre ellos.
- Javascript: con el fin de conseguir páginas web dinámicas se añade código javascript.
- Git [51] y Github [52]: El código fuente de la aplicación está disponible en Github. Github es un repositorio de código que podría utilizarse en un posible trabajo en equipo o para la continuación de este proyecto. Alojar el repositorio de código es ya imprescindible para cualquier proyecto software, tanto para colaboración entre desarrolladores, como para tener una copia de seguridad. El código de este proyecto se puede encontrar en el siguiente enlace https://github.com/ariadnaGomez/PFC. Además se ha utilizado el control de versiones Git.
- PHP/Java: Al ser una aplicación web se tuvieron en cuenta varios aspectos para la elección entre los lenguajes PHP y Java. Fue elegido PHP por su fácil integración con HTML además de por ser mucho menos pesado, lo que produce al usuario una sensación de rapidez y mayor usabilidad.
- Autocomplete (jQuery) [53]: Para implementar el autocompletado de la caja de texto se ha usado un *widget* de *jQuery*. Autocomplete permite al usuario

encontrar y seleccionar de forma rápida, desde una lista preconfigurada, mientras escribe. Este *widget* viene integrado en código de *jQuery UI* por lo que su instalación es muy simple: basta con añadir las librerías.

- Servicio web: se ha desarrollado un servicio web en Java para la descarga de la lista de autores. Se implementa en Java ya que al ser un lenguaje compilado era mucho más eficiente que PHP. Se investigaron otras tecnologías como por ejemplo *php/java bridge* [54] que sirve para integrar código java en PHP. Al final lo más eficiente era el servicio web, por lo que se seleccionó esta tecnología.
- Jena: Este servicio web realiza una petición a la DBpedia preguntando por todos las personas de tipo escritor. Para realizarla se usa la librería Jena. Se recupera un subconjunto (modelo RDF) con solo los identificadores de DBpedia y los nombres de autores. Este grafo es consultado una vez por cada inicial. La información se guarda por la inicial del autor en 28 ficheros distintos. Para que la búsqueda en el listado de autocompletar se pueda buscar por apellido además también se guarda por la inicial del apellido. Aparecen duplicadas las triplas pero era lo más eficiente para poder buscar tanto por nombre como por apellido.
- Ajax [55]: para recuperar la información del autor de forma dinámica se ha optado por realizar peticiones Ajax. De esta forma se puede mostrar la información del autor sin recargar la página.
- Css [56]: para la maquetación de la aplicación se utilizó un *css* libre ya creado.
- Apache [57]: PHP se ejecuta sobre un servidor apache. Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.
- Tomcat 7 [58]: el servicio web *downloader.war* implementado en este proyecto para la descarga de la lista de autores se ejecuta sobre Tomcat 7. Tomcat es un servidor web con soporte de *servlets* y JSPs.
- Memcached: es un sistema distribuido de propósito general para caché basado en memoria, muy usado en la actualidad por múltiples sitios web.

  Su funcionamiento se basa en una tabla *hash* distribuida a lo largo de varios equipos. Conforme ésta se va llenando, los datos que más tiempo han

permanecido sin ser utilizados se borran para dar espacio a los nuevos. La aplicación primero comprueba si puede acceder al listado de autores a través de Memcached antes de recurrir al fichero de texto donde se guarda el listado de autores, cuyo acceso es mucho más lento.

3.6.2. ORGANIZACIÓN DEL CÓDIGO

En este apartado se explica la estructura interna del proyecto. Contiene seis directorios principales entre los que se encuentran:

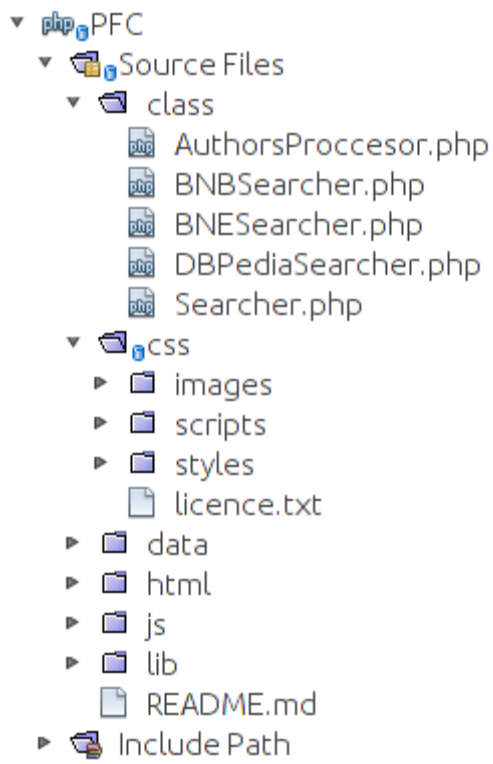

FIG 27.- ORGANIZACIÓN DEL CÓDIGO

- Class: contiene las clases necesarias para la comunicación con las distintas fuentes de datos y el almacén de autores. Las explicaremos en más detalle a continuación.
- Css: Contiene los scripts, archivos css e imágenes.
- Data: Contiene la lista de autores, corresponde a los ficheros que en modelo de diseño llamamos almacén de autores.
- Html: código html para la representación de la aplicación
- Js: código javascript necesario para carga dinámica de las páginas, y también para el autocompletado el cuadro de texto.
- Lib: contiene las librerías externas utilizadas, así como el archivo *downloader.war*. Es el servicio web que se usa para la descarga de la lista de autores. El archivo deberá estar desplegado en un servidor para que la actualización de autores funcione. Se guarda en esta carpeta para que esté en Github el código completo del proyecto.

Clases

Searcher: clase padre encargada de instanciar las clases de RAP, la librería necesaria para consultar las fuentes de datos.

DBpediaSearcher: clase hija de Searcher, encargada de hacer las consultas a DBpedia.

BNESearcher: clase hija de Searcher, encargada de hacer las consultas a BNE.

BNBSearcher: clase hija de Searcher, encargada de hacer las consultas a BNB.

AuthorsProccesor: Clase encargada de cargar el modelo RDF a la librería RAP y hacer las consultas necesarias para recuperar la lista de todos los autores que contiene.

3.6.3. CONSULTA DE LA INFORMACIÓN EN LAS FUENTES DE DATOS

En esta sección analizaremos las consultas que se realizan a las distintas fuentes de datos.

La primera necesidad es obtener una lista de autores, para lograrlo realizaremos una consulta CONSTRUCT al SPARQL endpoint de la DBpedia. Con esta consulta obtenemos un grafo RDF, que contiene los nombres e identificadores de todos los escritores que contiene la DBpedia.

```
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
CONSTRUCT {?person yago:hasName ?name.}
WHERE {
        {
                ?person a yago:Writer110794014 .
        } UNION {
                ?person a ?type.
                ?type rdfs:subClassOf yago:Writer110794014 . "
        } UNION {
                ?person a dbpedia-owl:Writer.
        } UNION {
                ?person a ?type1.
                ?type1 rdfs:subClassOf ?type2 . "
                ?type2 rdfs:subClassOf yago:Writer110794014 . "
        }
        ?person foaf:name ?name.
}
```

Una vez obtenido este grafo lo filtraremos. Para ello hacemos una consulta SPARQL por letra del abecedario, usando un filtro (FILTER) y la expresión regular "^A". Así obtendremos un subgrafo con todos los autores cuyo nombre comience por A, por B y así sucesivamente.

Estos subgrafos los iremos guardando en ficheros de texto, para consultarlos posteriormente.

Además, para que el autocompletado funcione correctamente (en un principio al escribir el apellido del autor no ayudaba autocompletando), hubo que realizar el proceso de filtrado también por apellidos. En este caso se hace un filtrado muy parecido al anterior pero con la expresión regular " A" (antes del primer carácter, A, debe haber un espacio). De esta forma además de por apellido la aplicación también permite la búsqueda por el segundo nombre, en caso de nombres compuestos.

Para realizar la función de autocompletado necesitamos recuperar el listado de autores que anteriormente guardamos en los ficheros de texto en formato RDF. Para ello cargamos el modelo con la librería RAP y hacemos una consulta SELECT para recuperar todos los nombres.

La siguiente consulta que se realiza también es a la fuente de datos DBpedia. En este caso lo que se solicita es información acerca del autor. Teniendo el ID del autor (ya que tenemos un listado de todos los escritores con sus IDs), hacemos una consulta SELECT de todas las propiedades que se necesitaban pues como vimos, es necesario pedir la misma información por medio de propiedades diferentes al no existir uniformidad de criterio al adjudicar a la misma información igual propiedad. En el apartado 3.5 vimos el listado de todas estas propiedades que hacían referencia al nombre del autor, breve descripción del mismo, su foto, lugar y fechas de nacimiento y defunción.

Para recuperar la información bibliográfica haremos dos consultas SELECT a la BNE y a la BNB.

3.6.4 DIAGRAMA DE COMPONENTES

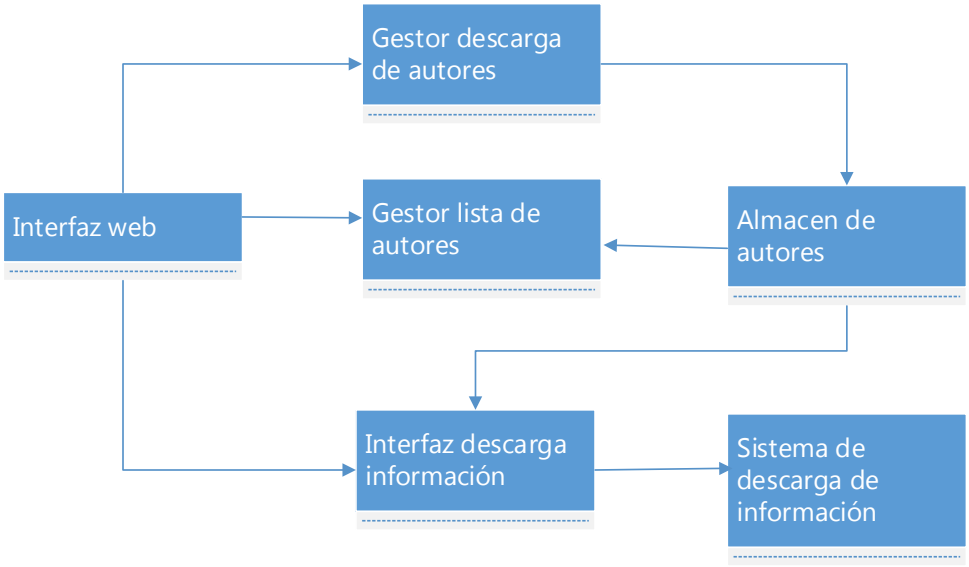

FIG 28.- DIAGRAMA DE COMPONENTES

3.6.5. DIAGRAMA DE DESPLIEGUE

Para una mejor comprensión de la arquitectura se muestra en la siguiente imagen un esquema:

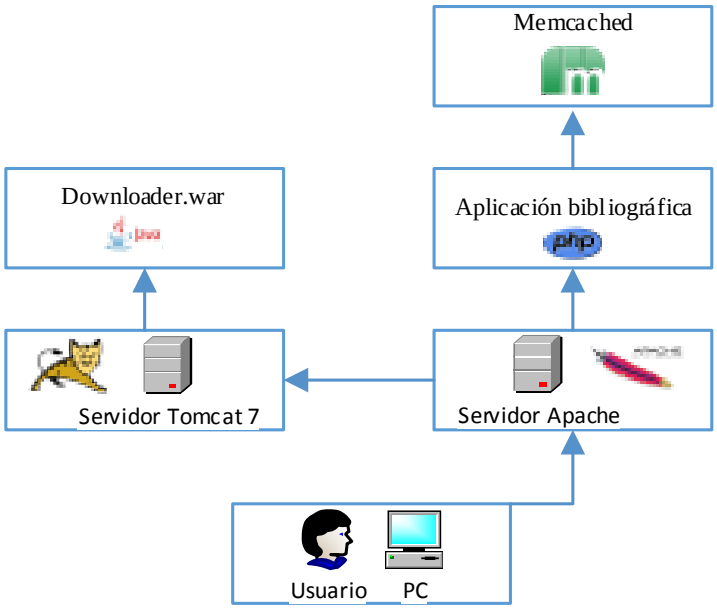

FIG 29.- DIAGRAMA DE DESPLIEGUE

4. CONCLUSIONES

Prácticamente todo el conocimiento de la humanidad se había acumulado exclusivamente en los libros hasta finales del siglo XX. La tecnología de la información vino a terminar con un monopolio sólo roto por los soportes audiovisuales.

La unión de libros y tecnología resulta en la actualidad algo natural. Desde la aparición de los libros electrónicos, los soportes digitales se han convertido en un sistema de lectura y difusión de originales muy popular. La facilidad de edición y publicación de libros en ediciones electrónicas significa que prácticamente el contenido de cualquier libro será preservado. Pero el auténtico potencial de esta relación entre libros (textos en general) y tecnología reside en la posibilidad de relacionar referencias, autores, contenidos, que en un mundo analógico habrían permanecido aislados o sólo vinculados de una forma circunstancial o por determinados especialistas.

Este trabajo ha tratado de establecer precisamente los fundamentos tecnológicos de esa capacidad de relación basándose en el paradigma más antiguo de la comunicación humana: el significado de las palabras, su semántica.

Hemos visto cómo la Web Semántica y el Linked Data permiten entrelazar los datos de modo que contenidos ocultos puedan abrirse y ser relacionados entre sí, sumando informaciones capaces de incrementar tanto el conjunto de conocimientos como su disponibilidad.

Para llegar a ello hemos situado el concepto de datos abiertos enlazados, Linked Open Data (LOD), y su relevancia como elemento base en los procesos de preservación, recuperación e intercambio de información en el ámbito de la web y hemos visto el avance que supone la publicación de esta información en LOD en un contexto de acceso abierto.

Hemos tratado de comprender el funcionamiento de la Web Semántica, sus ventajas, su capacidad para servir mejor al usuario, así como vislumbrar de una forma general lo que sin duda será un brillante futuro impulsado por las recientes tecnologías.

Se ha intentado realizar un análisis de la situación actual de la web, con la finalidad de comprender mejor el camino hacia el que debemos dirigirnos.

Se han estudiado y analizado muchas más fuentes de datos aunque las tres seleccionadas fueron, como hemos visto, la de DBpedia, la British National Bibliography y la Biblioteca Nacional Española.

Una de las dificultades encontradas ha sido la falta de documentación en algunas fuentes de datos. Hay un largo camino por recorrer, pero las fuentes están ahí y su potencial aportación al conjunto del sistema de conocimiento, es inmenso. En ese sentido es todo un reto encontrar una forma de documentar fuentes de datos tan grandes como DBpedia. El panorama es muy irregular, con fuentes de datos muy bien documentadas, como la de la BNB y otras sin documentar como la de la BNE. Otra dificultad encontrada es la que la información disponible en la BNE está pensada para la gestión bibliotecaria, como vimos al analizar esta fuente.

La aplicación aquí expuesta es sólo una muestra de lo que se podría hacer con Linked Data y fue abordada con la intención de dar una muestra clara de su utilidad y del enorme potencial de este método de publicación de estructuras de datos relacionables.

Y estos dos objetivos han quedado completamente demostrados a lo largo del desarrollo práctico de la aplicación, su utilidad es evidente pues de una forma sencilla y rápida se pueden conjuntar datos de tres grandes fuentes y respecto al potencial queda demostrado en las mejoras y posibles trabajos de futuro que se enumeran en el siguiente apartado.

En cuanto al resultado concreto de la aplicación, tras haberla realizado, podemos afirmar que reúne las siguientes características:

- Es entendible ya que cuenta con un proceso bien definido
- Es confiable porque el margen de error de la aplicación es escaso.
- Es mantenible porque su evolución es factible y fácil, gracias a esta documentación.
- Es rápida por almacenar en local el listado de autores.

4.1. POSIBLES MEJORAS Y TRABAJOS FUTUROS

Las inmensas posibilidades de la utilización de Linked Data hacen difícil enumerar las mejoras que podrían implementarse. Las siguientes son sólo unos ejemplos:

- Implementar un demonio para que la actualización de los autores sea un proceso automático y no manual. En la aplicación desarrollada es necesario hacer click sobre un botón. La mejora podría ser, por ejemplo, un proceso automático que se ejecutase cada noche.
- Añadir todas las bibliotecas nacionales que tienen Linked Data como por ejemplo la Deutsche Nationalbibliothek [59] u otras bibliotecas de famosas universidades, como la de Harvard [60]. De esta forma los bibliófilos podrían con una sencilla búsqueda saber dónde encontrar casi cualquier texto o realizar estudios sobre traducción de las obras a los distintos idiomas, etc.
- Realizar un buscador por título o tipo de libro o por cualquiera de los datos de un autor (país de nacimiento, fecha de nacimiento…).
- Incorporar a la aplicación más información de los autores y de su bibliografía.
- Añadir el ISBN de los libros lo que nos permitiría realizar un catálogo de búsqueda en bibliotecas o un catálogo de compra online.
- Mostrar avisos de que las fuentes de datos están caídas. Mayor detalle en los errores que pueden surgir.
- Enlazar DBpedia con otras fuentes de datos a través de VIAF ID. Así la aplicación no sería dependiente del estado de DBpedia. En este momento la mayoría de autores no tienen su identificador de VIAF en la DBpedia.

5. BIBLIOGRAFÍA

[1] **Eco Umberto**, *Cómo se hace una tésis. Técnicas y procedimientos de investigación, estudio y escritura*, 1ª ed., Barcelona, Gedisa, S.A., 1982 (6ª ed., México, Editorial Gedisa Mexicana, S.A.), p. 230.

[2] http://linkeddata.org/

[3] **Berners-Lee, Tim** *The next Web of open, linked data*, TED 2009 http://www.youtube.com/watch?v=OM6XIICm_qo

[4] **Berners-Lee; Hendler; Lassila.** (2001). *The Semantic Web*. Scientific American, Vol. 284, num. 5, pp. 34-43 http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[5] **Berners-Lee, Tim.** *Semantic Web -XML2000. Architecture* http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

[6] http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica

[7] *Frequently Asked Questions on W3C's Web Ontology Language (OWL)* **Traducido al español por Carlos Iglesias Moro** http://www.w3c.es/Traducciones/es/SW/2005/owlfaq

[8] **Gruber, Tom R.** (1993). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Technical Report KSL*-93-04, Knowledge Systems Laboratory, Stanford University, CA, 1993.

[9] http://protege.stanford.edu/

[10] **W3C.** (World Wide Web Consortium). http://www.w3.org/

[11] http://www.w3.org/XML/Schema

[12] **W3C**. Resource Description Framework (RDF). http://www.w3.org/RDF/

[13] Recogido de **Lamarca Lapuente, M. J.** (2006) *Hipertexto: El nuevo concepto de documento en la cultura de la imagen. http://www.hipertexto.info*

[14] http://www.w3.org/TR/rdf-primer/

[15] http://www.w3.org/TR/rdf-concepts/

[16] http://www.w3.org/TR/rdf-syntax-grammar/

[17] http://www.w3.org/TR/rdf-mt/

[18] http://www.w3.org/TR/rdf-schema/

[19] http://www.w3.org/TR/rdf-testcases/

[20] http://www.w3.org/TR/2013/CR-turtle-20130219/

[21] **García García**, **A**. (2012). *Datos abiertos enlazados Linked Open Data (LOD) en documentación científica.* http://riunet.upv.es/bitstream/handle/10251/18272/Alicia%20Garc%C3%ADa%20Garc%C3%ADa.pdf?sequence=1

[22] http://www.w3.org/TR/owl-guide/

[23] http://www.w3.org/TR/2008/REC-rdf-sparql-XMLres-20080115/

[24] http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/

[25] http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

[26] http://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/

[27] http://www.w3.org/TR/2013/REC-rdf-sparql-XMLres-20130321/

[28] ***GeoLinkedData.es Consultas SPARQL***
http://geo.linkeddata.es/web/guest/endpoints

[29] **Berners-Lee, Tim.** (2006). *Linked Data - Design Issues. W3C.*
http://www.w3.org/DesignIssues/LinkedData.html

[30] Recogido de **Míguez Pérez, R.; Santos Gago, J. M.; Alonso Rorís, V. M.; Álvarez Sabucedo, L. M. y Mikic Fonte, *F. A.*** (2012) *Linked Data como herramienta en el ámbito de la nutrición.* Versión impresa Nutrición Hospitalaria vol.27 no.2 Madrid mar.-abr. 2012
http://scielo.isciii.es/scielo.php?pid=S0212-16112012000200001&script=sci_arttext&tlng=en

[31] http://dbpedia.org/

[32] http://datos.bne.es/

[33] http://www.bl.uk/bibliographic/datafree.html

[34] http://openbiblio.net/2010/10/05/jisc-openbibliography-cul-data-release

[35] http://cern.ch/bookdata

[36] http://dblp.org/db/

[37] **Vilches Blázquez, L. M.; Villazón-Terrazas, B.; Corcho, O.; Gómez Pérez, A.** (2010). *GeoLinked Data: An application case/ Un caso de aplicación.* I Jornadas Ibéricas de Infra-estruturas de dados espaciais.
http://oa.upm.es/6166/1/GeoLinkedData_LMVilches_Art%C3%ADculo_JIDEE2010_v2.pdf

[38] http://datahub.io

[39] http://viaf.org/

[40] **W3C Library Linked Data Incubator Group** (2011). *Draft report with transclusion.* <*http://www.w3.org/2005/Incubator/lld/wiki/DraftReportWithTransclusion*>
Informe Final del Grupo Incubador de Datos Vinculados de Bibliotecas. 2011. Traducido por Agenjo X. http://www.larramendi.es/LAM/Incubator/lld/XGR-lld-20111025.html

[41] http://www.bne.es/es/Inicio/Perfiles/Bibliotecarios/DatosEnlazados/DescargaFicheros/

[42] http://datos.bne.es/sparql

[43] http://authorities.loc.gov/

[44] http://www.lexvo.org/

[45] http://www.bl.uk/bibliographic/pdfs/bldatamodelbook.pdf

[46] http://dbpedia.org/sparql

[47] http://datos.bne.es/sparql

[48] http://bnb.data.bl.uk/sparql

[49] http://datahub.io/dataset/bluk-bnb

[50] http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/

[51] http://git-scm.com/

[52] https://github.com/

[53] http://api.jqueryui.com/autocomplete/

[54] http://php-java-bridge.sourceforge.net/pjb/

[55] http://api.jquery.com/jQuery.get/

[56] http://www.os-templates.com/

[57] http://httpd.apache.org/

[58] http://tomcat.apache.org/

[59] http://www.dnb.de/EN/datendienste/linkedData

[60] http://openmetadata.lib.harvard.edu/bibdata

5.1 OTRA BIBLIOGRAFÍA CONSULTADA

1. Ejemplos consultas BNB.- https://github.com/ldodds/bnb-queries
2. Disponibilidad SPARQL endpoints.- http://labs.mondeca.com/sparqlEndpointsStatus/
3. Guía para instalar Apache y PHP.- http://www.jeremymorgan.com/tutorials/linux/how-to-install-apache-php-and-mysql-on-ubuntu-12-dot-10-quantal-quetzal/
4. Guía para instalar memcaché.- http://www.php.net/manual/es/memcache.installation.php
5. Instalar myeclipse.- http://www.myeclipseide.com/
6. Tutorial crear un servicio web.- http://www.myeclipseide.com/documentation/quickstarts/webservices_jaxws/

7. Tutorial autocompletar con jQuery.- http://www.jqueryautocomplete.com/jquery-autocomplete-json-example.html

8. Tutorial cómo hacer una aplicación usando THe New York Times dataset.- http://open.blogs.nytimes.com/2010/03/30/build-your-own-nyt-linked-data-application/

9. **Tramullas Saz J.** (1999) *Agentes y ontologías para el tratamiento de la información: clasificación y recuperación en Internet* IV Congreso ISKO-España EOCONSID'99, Granada. 1999. http://dialnet.unirioja.es/servlet/articulo?codigo=1300520

10. **González Pérez, Y.** (2006) *Las ontologías en la representación y organización de la información***.** 2006 http://bvs.sld.cu/revistas/aci/vol14_4_06/aci08406.htm

11. **Méndez Rodríguez, E.M.** (1999) *RDF: Un modelo de metadatos flexible para las bibliotecas digitales del próximo milenio.*1999. http://www.cobdc.org/jornades/7JCD/1.pdf

12. **Giraldo, G.; Marín, J.C.; Urrego Giraldo, G.** (1999) *Extracción de elementos de una ontología del dominio a partir de documentos tipo esquema.* Revista Avances en Sistemas e Informática Vol.6 Nº 2. Medellín 1999. http://www.redalyc.org/articulo.oa?id=133113598003

13. **Cáceres Tello, J.** (2010) *La Web Semántica y el lenguaje RDF* https://portal.uah.es/portal/page/portal/GP_EPD/PG-MA-PROF/OLD_PG-PROF-138886%202008-07-14%2010-07-31/TAB4348465/TAB4348469/TAB4348477/Articulo_WebSemantica_Jesus_Caceres_CISTI_06.pdf

6. ANEXO

En este apartado daremos las instrucciones para la instalación y utilización de la aplicación, además de describir su configuración.

6.1 CONFIGURACIÓN DE LA APLICACIÓN

La aplicación cuenta con dos ficheros de configuración: uno para el servicio web y otro para la aplicación.

El fichero de configuración del servicio web, localizado dentro de este, en la ruta *downloader/WEB-INF/classes/com/pfc/downloader/CONF*, contiene una única variable de configuración.

- data = Esta variable tendrá como clave *data* y su valor será el directorio donde se quiere guardar los ficheros con las listas de autores. Por defecto tendrá la siguiente ruta */var/www/PFC/data/*. Esta carpeta deberá tener permisos de escritura.

El segundo fichero de configuración, localizado dentro del código fuente de la aplicación, ruta *PFC/config* contiene cuatro variables de configuración:

- data = Tendrá como valor, el directorio donde se encuentran los ficheros con las listas de autores. Por defecto tendrá la siguiente ruta */var/www/PFC/data/*. Debería coincidir con la variable *data* del fichero de configuración del servicio web.
- downloader_path = contiene el endpoint donde estará desplegado el servicio web.
- memcache_path = Path a Memcaché. Por defecto será localhost.
- memcache_port = Puerto a Memcaché. Por defecto será 112211.

6.2 MANUAL DE INSTALACIÓN

Este manual de instalación funciona en Linux, aunque es posible instalarlo en otros sistemas operativos.

1. Instalar Apache: Introducir en línea de comandos:

```
sudo apt-get install apache2
```

Para comprobar que se ha instalado correctamente cargar la siguiente URL en un navegador: **http://localhost.** Deberá aparecer la siguiente página:

FIG 30.- PÁGINA POR DEFECTO DE APACHE

2. Instalar PHP: Introducir en línea de comandos:

```
sudo apt-get install libapache2-mod-php5 php5
```

Para comprobar que se ha instalado correctamente crear un fichero llamado *test.php* en */var/www* e introducir el siguiente contenido:

```
<?php phpinfo(); ?>
```

A continuación abrir el navegador para acceder a http://localhost/test.php. Si está correctamente instalado veremos la siguiente página:

FIG 31.- OPCIONES DE CONFIGURACIÓN DE PHP

3. Instalar Memcaché: Introducir en línea de comandos:

```
sudo apt-get install php5-memcache
```

Una vez instalado Memcaché reiniciaremos Apache introduciendo en línea de comandos:

```
sudo /etc/init.d/apache2 restart
```

4. Tomcat: Para desplegar el servicio web necesitaremos instalar Tomcat. Para ello introducir en línea de comandos:

```
sudo apt-get install tomcat7
```

Para que funcione la aplicación, Apache y Tomcat no pueden estar escuchando en el mismo puerto, por tanto deberemos configurar Tomcat en otro puerto, por ejemplo el 8082. Para ello modificaremos el siguiente archivo: */var/lib/tomcat7/conf/server.xml* en el que sustituiremos el puerto 8080 por el 8082:

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="UTF-8"
           redirectPort="8443" />
```

Una vez cambiado este fichero reiniciaremos Tomcat, introduciendo en línea de comandos:

```
sudo /etc/init.d/tomcat7 restart
```

El siguiente paso para desplegar el servicio web será copiar el archivo *downloader.war* contenido en la carpeta *lib* del proyecto, al siguiente directorio: */var/lib/tomcat7/webapps/*. Finalmente reiniciaremos Tomcat de nuevo.

Es importante tener en cuenta que para que el servicio web pueda descargarse la lista de autores, debe tener permisos de escritura en la carpeta seleccionada en los ficheros de configuración.

Para que la aplicación se ejecute correctamente, tendremos que iniciar Memcaché, para lo que hemos de introducir en la línea de comandos:

```
memcached -d -m 1024 -u root -l 127.0.0.1 -p 11211
```

Con esto ya tendremos la aplicación lista para su uso.

6.3 MANUAL DE USO

Para utilizar la aplicación, el usuario deberá seguir los siguientes pasos:

- Ejecutar la aplicación: Acceder a la siguiente URL desde el navegador: http://localhost/PFC/html/index.php

- Buscar un autor concreto: Introducir su nombre en la caja de texto. Pasados unos segundos nos aparecerá una lista de los posibles autores a los que nos podemos referir, al elegir uno y pulsar enter o hacer click en la lupa, la aplicación buscará la información del autor, mostrándola cuando finaliza la búsqueda.

- Actualizar la lista de autores: Pinchar en el texto *Update authors list*.

- Acceder al código fuente de la aplicación en Github: Hacer click en el título *Aplicación bibliográfica.*