Universidad Politécnica de Valencia

Mejora de un portal de Patrimonio Inmaterial mediante etiquetado semántico

Tesina presentada por Cédric Marco Detchart
para obtener el grado de Máster en Inteligencia Artificial,
Reconocimiento de Formas e Imagen Digital

Dirigida por Miguel Rebollo Pedruelo

2015

Resumen

En este trabajo nos hemos centrado en el análisis de diferentes herramientas de web semántica para la creación de un portal de datos sobre patrimonio inmaterial. La información semántica permite el etiquetado de los datos de manera que una máquina pueda extraer información relevante, facilitando posteriormente la búsqueda de contenido y permitiendo su disponibilidad para otras máquinas.

La aportación principal de este trabajo es la creación de un nuevo sistema de gestión de información patrimonial y el etiquetado de los datos disponible paracfacilitar el acceso y el manejo de grandes cantidades de datos.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En primer lugar exponemos el marco y la motivación que nos ha llevado a la realización de este proyecto, además de indicar los objetivos del mismo. También presentamos la estructura de la memoria, plasmando en ella tanto la base teórica como la parte práctica y los resultados obtenidos.

1.1. Objetivos

En la actualidad la cantidad de datos que se trata en el mundo es cada vez mayor. Debido a este incremento, la extracción de información relevante así como la relación entre contenidos, su almacenamiento y su gestión se ve dificultada. Concretamente en el contexto del patrimonio inmaterial, en el que se centra este proyecto, la información disponible se puede considerar ilimitada, ya que cada vez se van obteniendo más datos y estos son añadidos a los ya existentes.

Este proyecto tiene su origen en la necesidad en la Cátedra de Patrimonio Inmaterial de Navarra de la Universidad Pública de Navarra (UPNA) de modernizar y mejorar su actual sistema de gestión de información de datos sobre el patrimonio inmaterial de Navarra. En la actualidad el sistema utilizado se basa en un entorno web en el que se insertan contenidos y se posibilita el acceso a estos mediante una página. Dicho sistema está desfasado y no permite ninguna actualización, además de ser costoso de mantener y de que presenta ciertas deficiencias.

Debido a los problemas expuestos anteriormente, se ha decidido plantear un en-

torno nuevo, realizando una migración de los datos existentes e incorporar nuevas herramientas. En el nuevo sistema, la gestión de los contenidos se hará de una manera mucho más sencilla. Debido a la flexibilidad de la herramienta, se pueden incorporar nuevas funcionalidades según las necesidades de los usuarios. Por otra parte, hemos enfocado el proyecto a la facilidad y claridad de navegación por el contenido disponible, así como al consumo de datos desde aplicaciones externas. Relacionado con este último apartado se han integrado herramientas de Web Semántica, orientadas a facilitar la descripción del contenido presente.

El presente trabajo está enfocado a la creación de un sistema de gestión de contenido, que facilite la búsqueda de información a los investigadores, así como la presentación de datos de interés al público en general de la manera mas simple y comprensible posible.

1.2. Motivación

Los motivos que nos han llevado a la realización de este proyecto son de índole educativa y profesional.

Por un lado, este desarrollo nos ha permitido poner en práctica parte de los conocimientos adquiridos a lo largo de los estudios realizados, combinando las diferentes técnicas estudiadas en relación a bases de datos, programación web, así como teoría de agentes inteligentes y web semántica.

El hecho de realizar un proyecto de esta envergadura es enriquecedor para el estudiante ya que permite hacer frente a dificultades que no están presentes en las prácticas aisladas que se pueden encontrar a lo largo de la enseñanza de las diferentes materias.

Por otro lado el hecho de que se trate de un proyecto en colaboración con la UPNA y que se le vaya a dar un uso real en la Cátedra, por parte del grupo de investigacin lingstica de esta universidad, es muy positivo ya que nos permite una introducción en el mundo profesional.

En este contexto, el proyecto que se ha realizado se encuentra en el ámbito del patrimonio inmaterial de Navarra siguiendo la guía del Plan Nacional de Patrimonio Inmaterial [13] que recoge todos los aspectos que deben describir a las manifestaciones

culturales de una comunidad, en este caso Navarra. Los objetivos del Plan Nacional, según se indica en su página web, son:

- Fomento de la investigación sobre los bienes culturales que integran el Patrimonio Inmaterial.
- Desarrollo de criterios comunes para la ejecución de intervenciones en Patrimonio Inmaterial.
- Establecimiento de una metodología consensuada que guíe la actuación de las administraciones e instituciones en la protección del Patrimonio Inmaterial.
- Diseño y desarrollo de estrategias que favorezcan la difusión de los valores culturales propios de las manifestaciones inmateriales de la cultura.

1.3. Estructura de la memoria

En este apartado plasmamos todo el trabajo realizado a lo largo de este proyecto de manera resumida a través de cinco capítulos.

En el primer capítulo explicamos qué es el sistema que vamos a desarrollar y cuáles son sus objetivos, así como las razones que nos han llevado a la realización del mismo.

En el segundo capítulo describimos un análisis de las tecnologías existentes en la actualidad para la realización de proyectos como el expuesto en este trabajo.

En el tercer capítulo nos centramos en las diferentes fases que ha tenido el proyecto realizado además de su desarrollo, exponiendo el análisis del sistema existente, la creación de la nueva herramienta, la migración de datos, así como el apartado de enriquecimiento de los datos con información semántica.

En el cuarto capítulo exponemos las pruebas realizadas para comprobar la funcionalidad añadida al sistema.

En el quinto y último apartado explicamos las conclusiones a las que hemos llegado tras la realización de este proyecto y comentamos las posibles líneas futuras.

Finalmente, en un anexo, detallamos bibliografía utilizada en todo el desarrollo del proyecto, los diferentes recursos web consultados y el listado de las figuras de esta memoria.

Capítulo 2

Estado del arte

2.1. Introducción

En este capítulo realizamos un análisis de las tecnologías web y de almacenamiento existentes para gestión documental y en concreto, para la gestión de patrimonio cultural.

Desde sus inicios, hace escasos años, la Web ha sido una gran revolución que ha permitido el acceso a una inmensa cantidad de información de manera global [28]. Debido a la extensión y a la facilidad para conseguir toda esta información, la realidad actual es que contamos con demasiados datos y la tarea de filtrarlos y obtener lo que el usuario necesita es cada vez más difícil.

A raíz de este problema surgen los buscadores, cuya función es proponer resultados al usuario. Existen principalmente dos tipos, los buscadores por categorías y los basados en palabras clave.

En cuanto a los buscadores por categorías, su funcionamiento consiste en el almacenamiento manual de los datos por parte de una persona, de tal manera que queden disponibles y puedan ser utilizados por los usuario. En este caso la información se almacena y se muestra al usuario en forma de categorías temáticas, estructuradas en forma de árbol, desde los términos más generales a los más concretos. Entre este tipo de buscadores encontramos Yahoo! Directory [1] (cerrado desde el 31 de Diciembre de

[1] https://business.yahoo.com/

2014 [34]) o DMOZ [2].

Los buscadores por palabras clave, basan su funcionamiento en la recolección y análisis automático, mediante robots, de texto presente en las páginas web para crear índices a partir de estas palabras. De esta manera, un usuario introducirá un texto y el buscador analizará las coincidencias de esas palabras en su base de conocimiento para proponer resultados que contengan esos términos. Este tipo de funcionamiento nos lleva al siguiente problema: los resultados mostrados no siempre tienen relación con lo buscado o en algún caso, más raro, no se encuentra ningún resultado. Esto es debido a que los buscadores funcionan de manera sintáctica. En este apartado tenemos buscadores como Lycos [3] o Altavista [4], uno de los primeros en surgir y que acabó desapareciendo tras su adquisición por parte de Yahoo!, más adelante aparecieron los más conocidos actualmente, como Google [5], Yahoo! Search [6] o Bing [7].

Aunque no es un motor de búsqueda propiamente dicho, cabe destacar Wolfram Alpha [8] [11]. Su funcionamiento es radicalmente distinto a los buscadores clásicos ya que es capaz de responder a una pregunta o un cálculo introducido por un usuario mediante el lenguaje natural [31], proporcionando una respuesta directa y no una lista de posibles recursos almacenados que coinciden con las palabras introducidas por el usuario.

En los últimos años la web ha evolucionado hacia dos estrategias de manera separada, pero que son complementarias. Por un lado la Web Semántica, que intenta aportar cierto formalismo a la web original, a través de descripciones únicas de los contenidos, de manera que un sistema inteligente sea capaz de comprender e intercambiar datos. Por otro, tenemos la web clásica en el que la figura central es el usuario como productor de contenidos, así como el intercambio de información entre personas a través de herramientas y servicios de fácil utilización.

En este proyecto nos centramos en la técnica de Web Semántica donde a conti-

[2] http://www.dmoz.org/
[3] http://www.lycos.com/
[4] www.altavista.com/
[5] https://www.google.com/
[6] https://search.yahoo.com/
[7] http://www.bing.com/
[8] http://www.wolframalpha.com/

nuación analizamos tecnologías y herramientas para ponerla en marcha.

2.2. Sistemas de Gestión de Contenidos

En el campo de la construcción de páginas web cada vez es más frecuente, desde hace unos años, la tendencia de separar los datos de su estructura de representación [22]. Debido a este paradigma surgen los Sistemas de Gestión de Contenidos (CMS, *Content Management System*). Esta separación permite no depender únicamente del lenguaje HTML para servir la información a los usuarios o a otros sistemas.

Existen numerosos CMS en el ámbito de la web, entre los que destacan los tres más importantes que son Wordpress [9], Joomla![10] y Drupal [11] [41]. En la Tabla 2.1 se indican los aspectos más destacados.

Estas tres herramientas se caracterizan por ser de código abierto y modulares. Todos cuentan con un núcleo, que contiene la funcionalidad básica y que posteriormente podemos ampliar mediante el uso de extensiones.

Wordpress es la herramienta más popular y utilizada en la actualidad. La ventaja que tiene frente a sus competidores es su facilidad de manejo y administración, que puede resultar perfecto para usuarios sin mucho conocimiento. Posee una gran comunidad de desarrolladores, por lo que la variedad de extensiones es muy amplia. Por otra parte, tiene una cierta limitación en cuanto a la gestión de grandes cantidades de contenido debido a su diseo original como plataforma de blogs. Debido a su gran utilización y popularidad, es el sistema que más ataques sufre.

Joomla es un producto más orientado a crear redes sociales y con una interfaz amigable para el usuario, aunque su curva de aprendizaje es más elevada que Wordpress. Permite la personalización del sistema y cuenta con una fuerte comunidad de desarrolladores. Aunque, no cuenta con una gestión amplia de control de acceso para usuarios (ACL, *Access Control List*), es decir, no se pueden crear permisos específicos para tipos de usuario segn los contenidos del sistema o para ciertas partes del sistema.

Drupal parece en principio la herramienta más compleja de utilizar. Es un CMS

[9] https://wordpress.org/
[10] http://www.joomla.org/
[11] https://www.drupal.org/

	Wordpress	**Joomla!**	**Drupal**
Portales usando esta tecnología	60,3 %	7,5 %	5,1 %
Precio	Gratuito	Gratuito	Gratuito
Usabilidad	Alta	Baja	Media
Instalación	Fácil	Media	Muy fácil
Mantenimiento	Fácil	Fácil	Medio
Flexibilidad	Alta	Baja	Muy alta
Curva de aprendizaje	Baja	Media	Alta
Documentación	Muy buena	Media	Buena
Comunidad	Gran comunidad de soporte	Gran comunidad de desarrolladores	Gran comunidad de desarrolladores
Base de datos	MySQL	MySQL	MySQL, por defecto, aunque ofrece la posibilidad de cambiarla
Dificultad de administración	Baja	Media	Alta
Seguridad	Baja	Media	Alta
Tipos de proyectos	Páginas simples, blogs,...	Portales tipo redes sociales	Portales complejos

Tabla 2.1: Comparativa de los principales CMS

más orientado a desarrolladores y posee la curva de aprendizaje más elevada, pero es la que más posibilidades ofrece a la hora de realizar cambios y personalizaciones. Además es una de las opciones que integra posibilidades de gestión de acceso.

La unidad básica de Drupal es el *nodo*, cuya funcionalidad se puede asimilar a una página, en la que se permite insertar un título y el cuerpo de dicha página. Esta funcionalidad puede ser ampliada haciendo uso de los módulos, en concreto *Field* y *Entity*, que nos permiten hacer que un nodo tenga más campos disponibles, para poder personalizar el contenido que se pueda insertar según las necesidades del usuario.

Aunque el número de usuarios que hacen uso de Drupal y su curva de aprendizaje sea la más elevada, nos hemos decantado por este sistema debido a su gran flexibilidad a la hora de crear páginas web a medida, proporcionando la estructura necesaria para

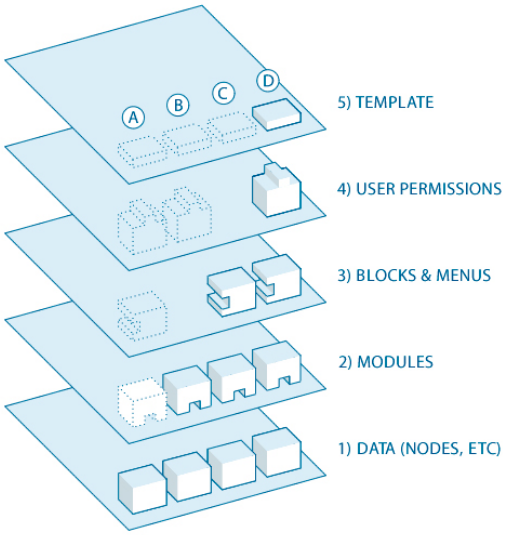

Figura 2.1: Estructura interna de Drupal (https://www.drupal.org)

realizar el mantenimiento, la creación de contenido, el control de acceso y de usuarios, así como el almacenamiento y la gestión de la base de datos (Figura 2.1). Además permite el uso de módulos que completan la funcionalidad básica del sistema.

2.3. La Web Semántica

En la web clásica un usuario es capaz de identificar conceptos dentro de un documento dado, al igual que las relaciones existentes entre diferentes recursos. Este comportamiento, es muy difícil de realizar por parte de un sistema software. Una máquina puede llegar a entender el significado y el contexto de un contenido empleando técnicas de procesado de lenguaje natural, extayendo información, pero no sigue un esquema definido y estándar, comprendido por cualquier sistema y que permita la inferencia de información nueva a partir de la ya existente y establecer relaciones entre los contenidos.

La Web Semántica intenta hacer que la infomación y los contenidos de las páginas web sean capaces de ser interpretadas por las máquinas [36], dejando de lado el modo en que son representados. La incorporación de contenido semántico no se asimila

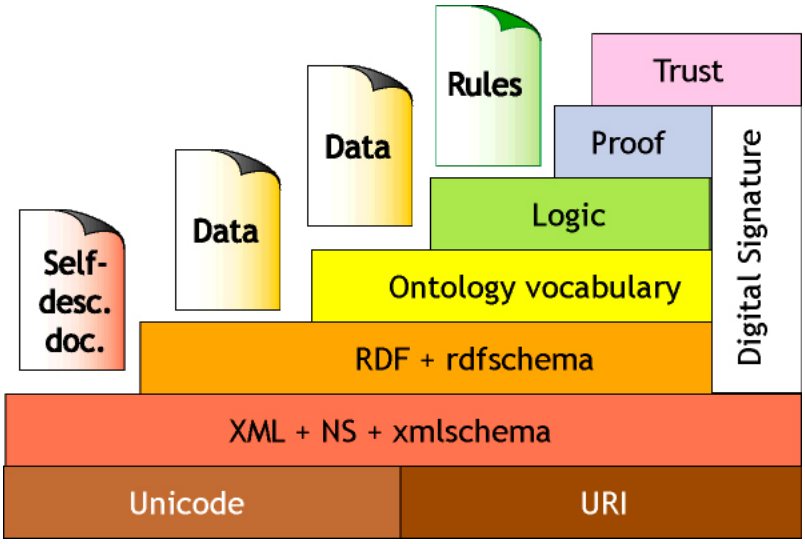

Figura 2.2: Arquitectura de la Web Semántica (http://www.w3.org)

a un nuevo tipo de web, sino que es una herramienta complementaria a la web existente. Esta nueva herramienta forma una arquitectura (Figura 2.2) en la que cada componente sigue unas especificaciones concretas. Para comprender mejor el concepto de semántica es necesario no perder de vista que existen numerosos niveles.

Las dos primera capas engloban los estándares sobre los que se basa la web semántica. Unicode hace posible la codificación de los textos para que se puedan interpretar independientemente de su idioma, sin que aparezcan caracteres extraos. URI se encarga de identificar de manera única un recurso en la Web para que pueda ser localizado.

La capa XML+NS+xmlschema agrupa una serie de tecnologías que son las que permiten la comunicación entre sistemas. XML (Extensible Markup Language) es un lenguaje que permite describir el contenido de una página web y actúa como fomato común para el intercambio de documentos. NS (Namespaces) permite identificar elementos mediante el uso de espacios de nombres a través de URIs. XML Schema permite realizar la descripción de la estructura y las restriccionese un documento XML, obteniendo de esta manera documentos con un esquema uniforme.

No obstante, aunque este lenguaje permita describir en cierto modo el contenido de una página web, tiene ciertas limitaciones; en concreto no permite la descripción explícita de un tipo de recurso. Una forma de salvar estos obstáculos y poder rea-

lizar esta tarea, consiste en incorporar al contenido de las páginas web anotaciones formales identificando de manera única cada concepto. Este tipo de anotaciones se pueden realizar de cualquier modo, pero para que todas las páginas puedan entenderse se ha formalizado un modelo de datos recomendado, y basado en XML, por parte del *World Wide Web Consortium* (W3C) llamado *Ressource Description Framework* (RDF).

La capa de RDF+rdfschema engloba el modelo universal que permite la descripción de los datos en la Web y el vocabulario con el que se puede describir. RDF es el lenguaje con el que realizamos las descripciones de los elementos de un documento a través de URIs y RDF Schema es el encargado de proveer un vocabulario definido sobre RDF que nos permite utilizar una semántica definida. Esta capa es la encargada de describir los datos y una parte de la información semántica.

La capa de Ontologías aporta la descripción de los objetos y sus relaciones con otros objetos. Permite tener una conceptualización de los recursos disponibles y aporta nuevas clases y propiedades para definir los recursos.

La capa lógica aporta la parte de reglas de inferencia, con las que una máquina es capaz de razonar para extraer conocimiento y tomar decisiones, relacionando conceptos. De esta manera se consiguie que una máquina manipule los datos de una manera más eficiente.

La capa de prueba permite verificar que la información encotrada por un sistema inteligente es correcta. Aporta información sobre como se ha obtenido dicha información, a partir de que datos y desde que fuentes. Esta capa es en esencia la que permite conocer el origen del conocimiento obtenido.

La capa de confianza es la que permite comprobar que la información obtenida proviente de las fuentes de información correctas.

Finalmente, la capa de firma digital trata de mejorar la seguridad de la web a través del uso de sistemas de encriptación y firma digital para identificar los cambios en los contenidos

2.3.1. RDF

El modelo RDF permite descomponer el conocimiento contenido en una página en pequeñas partes que reciben el nombre de tripletes. Estos tripletes son de la forma

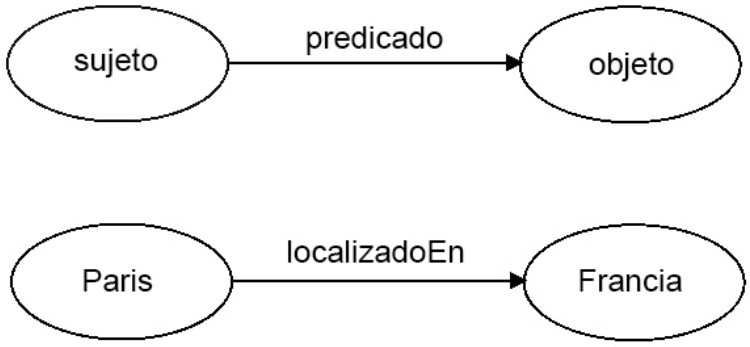

Figura 2.3: Tripletes RDF

sujeto-predicado-objeto (Figura 2.3). El recurso correspondiente queda representado por el sujeto, el predicado describe sus rasgos o aspectos, especificando la relación entre el sujeto y el objeto y por último, el objeto es el valor de la propiedad que se está representando. Este modelo permite añadir significado a las páginas de manera que un sistema inteligente pueda entenderlo, independientemente del idioma. Por otro lado, es un recurso idóneo para entornos distribuidos, en los que se necesita tener una interrelación de contenidos presentes en diferentes páginas web, sin que sea necesario actualizar, en el caso de producirse un cambio externo, los datos de nuestro sistema.

Para identificar estos recursos RDF, tanto sujetos como predicados, de manera única surgen lo que se llaman *Univeral Resource Identifiers* (URIs). Estos identificadores suelen tener la misma forma que una dirección web, siendo accesibles y permitiendo de esta manera la disponibilidad de cierto tipo de información acerca del recurso, de modo que un cliente RDF pueda manejar más información y sepa de lo que trata un contenido. El problema que surge con el formato de las direcciones web es que son bastante extensas. Esto se puede solucionar si se adopta el mecanismo de espacios de nombre, que permite abreviar las URIs, asociando una dirección web a un alias que se declarará al inicio de un documento y se podrá usar a lo largo del mismo.

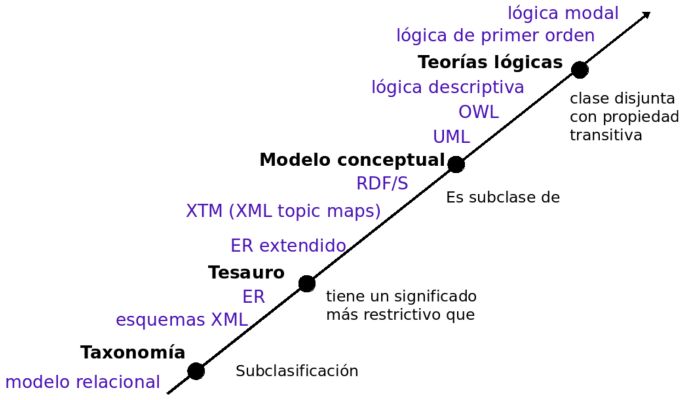

Figura 2.4: Espectro de las ontologías (adapatado de [32])

2.3.2. Ontologías

Para poder ser de utilidad en las aplicaciones web, estas anotaciones semánticas no pueden funcionar de manera independiente. Para que el sistema sea completo y funcione de manera que una máquina sepa lo que está buscando relacionando conceptos, se debe definir un dominio en el que englobar todos los términos semánticos necesarios. Para ello es imprescindible incorporar los vocabularios y propiedades que los definan, además de cierto tipo de reglas que permitan indicar las relaciones entre los diferentes conceptos de dicho dominio. Este modelo de dominio es lo que se llama *ontología*. La verdadera utilidad de una ontología es que sea compartida por el mayor número de personas u organizaciones.

Las ontologías se engloban dentro de lo que se denomina el espectro semántico (Figura 2.4), en el que vamos desde una especificación sencilla, como son las taxonomías, hasta la más complejas, como las teorías lógicas.

Como herramienta más completa que RDF contamos con *Web Ontology Language* (OWL). Permite identificar de manera precisa los recursos de un documento y sus relaciones. Se basa en RDF y XML y está orientado especialmente para describir información en la Web. Se diferencia de RDF por su mayor capacidad de comprensión por parte de un sistema informático y por tener un vocabulario mas extenso y una sintaxis más estricta. Existen tres especificaciones de OWL:

- OWL Lite: Es la variante más sencilla de OWL. Permite describir taxonomias

y restricciones sencillas. Debido a esto se garantiza que un motor de inferencia puede obtener siempre una respuesta a lo que la máquina esté buscando.

- OWL DL: Proporciona máxima expresividad manteniendo completitud computacional y decibilidad. Esta variente está orientada a proporcionar capacidades de lógica descriptiva.

- OWL Full: Esta última variante combina la semántica de las dos anteriores, manteniendo la compatibilidad con RDF Schema.

Existen diferentes métodos de representación del modelo RDF, como pueden ser *Notation3* (N3) [37], *Terse RDF Triple Language* (Turtle) [25], *RDF in attributes* (RDFa) [30] o RDF/XML.

N3 es un lenguaje lógico, que no sigue un representación XML y que pretende la facilidad de lectura de la información. Sus objetivos son:

- Optimizar la representación de los datos y su lógica en un mismo lenguaje.

- Permitir el uso de RDF.

- Permitir la integración de reglas con RDF.

- Permitir las citas de manera que se puedan realizar declaraciones de declaraciones.

- Ser lo más natural, legible y simétrico posible.

A continuación vemos un ejemplo de su uso:

```
@prefix content: <http://purl.org/rss/1.0/modules/content/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<http://localhost/~chivo/navarchivoDP/?q=es/node/5577> a schema:City
    ;
    schema:geo "42.6630587,-2.149935599999935" ;
    schema:name "Abáigar"^^xsd:string ;
    geo:alt "498 m."^^xsd:float ;
    geo:lat_long "42.6630587,-2.149935599999935" .
```

Turtle es un subconjunto del lenguaje N3 que permite expresar las tripletas RDF, de manera textual sin seguir un formato XML. Contrariamente a N3, este lenguaje solamente permite expresar información de un grafo RDF válido. Su utilización es muy parecida al anterior formato:

```
@prefix schema: <http://schema.org/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://localhost/~chivo/navarchivoDP/?q=es/node/5577>
  a schema:City ;
  geo:alt "498 m."^^xsd:float ;
  geo:lat_long "42.6630587,-2.149935599999935" ;
  schema:geo "42.6630587,-2.149935599999935" ;
  schema:name "Abáigar"^^xsd:string .
```

RDFa es una recomendación de la W3C que permite añadir extensiones a documentos HTML, XHTML y basados en XML a nivel de atributos para incorporar metadatos a estos documentos utilizando la sintaxis de RDF. Los atributos que se incorporan son:

- **about:** URI especificando información sobre los metadatos.
- **rel, rev:** Se encarga de especificar la relacion o relación inversa con otros recursos.
- **src, href, resource:** Especifica la fuente del recurso.
- **property:** Indica la propiedad del contenido de un elemento.
- **content:** Sobrescribe el contenido de un elemento cuando se usa el atributo *property*.

- **datatype:** Especifica el tipo de datos que se está expresando en el atributo del tipo *property*.
- **typeof:** Especifica el tipo RDF del recurso descrito.

RDF/XML es una sintaxis definida por la W3C para expresar información de un grafo RDF en formato XML.

A continuación se muestra el último formato mencionado:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:schema="http://schema.org/"
  xmlns:content="http://purl.org/rss/1.0/modules/content/">
  <rdf:Description rdf:about="http://localhost/~chivo/navarchivoDP/?
     q=es/node/5577">
    <rdf:type rdf:resource="http://schema.org/City"/>
    <geo:alt rdf:datatype="http://www.w3.org/2001/XMLSchema#float
       ">498 m.</geo:alt>
    <geo:lat_long>42.6630587,-2.149935599999935</geo:lat_long>
    <schema:geo>42.6630587,-2.149935599999935</schema:geo>
    <schema:name rdf:datatype="http://www.w3.org/2001/XMLSchema#
       string">Abáigar</schema:name>
  </rdf:Description>

</rdf:RDF>
```

El etiquetado RDF también puede estar integrado dentro del código HTML de una página web del siguiente modo:

```
<html lang="es" dir="ltr"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">

<div>Geolocalización:</div>
<div property="geo:lat_long">42.6630587,-2.149935599999935</div>
```

En cuanto al consumo de recursos basados en RDF, existen herramientas capaces de realizar consultas y obtener información de diferentes fuentes. En este caso la

W3C recomienda el uso de SPARQL. Aunque existen numerosas implementaciones de este lenguaje, que permiten operaciones de creación, modificación y borrado, su funcionalidad básica es la de realizar consultas sobre grafos RDF.

2.4. Bases de datos no relacionales

En cuanto al apartado de almacenamiento de la información propiamente dicho, los CMS utilizan lo que se llaman Sistemas Gestores de Bases de Datos (SGBD). Estos SGBD pueden ser de diferentes tipos, entre los que destacan los que siguen un esquema relacional o SQL y los no relacionales o NoSQL.

Un SGBD relacional permite extraer información presente en diferentes tablas de una base de datos siguiendo las interconexiones existentes entre la información. Existen modelos alternativos como el que sigue NoSQL, en el que las mencionadas relaciones no existen y los datos no se almacenan en tablas como en el modelo anterior [29]. Una de las grandes diferencias entre estos dos sistemas es el tiempo que tardan en ejecutar las diferentes tareas requeridas. En general los sistemas NoSQL se han optimizado para ser rápidos y escalables en el manejo de grandes cantidades de datos.

Una base de datos relacional almacena la información en tablas, definiendo columnas que representan un atributo de un tipo. El contenido de las tablas se aade en forma de filas en las tablas, en la que cada elemento insertado ir a parar a una columna. Este modo de funcionamiento hace que no se pueda insertar en una tabla información diferente a la que se haya diseado originalmente, se insertan todos los valores aunque sean nulos.

Existen alternativas no relacionales para almacenar datos, son las llamadas bases e datos NoSQL. Dentro de esta familia de sistemas podemos encontrar cuatro tipos: orientadas a columnas, de clave-valor, basadas en grafos y orientadas a documentos.

Las bases de datos orientadas a columnas funcionan de un modo parecido a las relacionales salvo que almacenan la información en columnas. Están estudiadas para poder ejecutar consultas sobre grandes cantidades de datos.

Los sistemas de tipo clave-valor almacenan los datos en forma de tupla, funcionando como un array asociativo. La lectura y escritura de datos se realiza mediante la clave. Se basan en tres operaciones sencillas, insertar un elemento identificado con

su clave, consultarlo a través de la clave y eliminarlo a partir de la clve.

Las bases de datos basadas en grafos representan la información mediante nodos y las relaciones existentes entre los datos mediante aristas entre los mismos.

Una base de datos NoSQL documental ofrece la posibilidad de guardar registros utilizando una estructura tipo JSON (*JavaScript Object Notation*), en la que se asocia un nombre con un valor. Estos nombres no se especifican con anterioridad a cualquier inserción siguiendo un modelo. Aunque en principio se pueden almacenar registros que no sigan un esquema, generalmente las aplicaciones que utilizan este tipo de bases de datos requieren un modelo. La principal ventaja de este sistema es su flexibilidad. Permite aadir o eliminar campos en un registro sin que afecte al resto, ni a la integridad del documento, siempre que que el sistema que lo use no necesite dichos valores. Cada contenido que se guarda en la base de datos queda asociado a un único documento, en vez de separarlo en diversas tablas.

Existen numerosos sistemas de bases de datos NoSQL, entre los que destacan MongoDB, CouchDB, Cassandra y Redis.

MongoDB es una base de datos para gestionar grandes cantidades de datos, que ofrece facilidad a la hora de trabajar con ellos y un bajo nivel de mantenimiento. Es una base de datos documental que almacena los datos en formato BSON (*Binary JSON*), permitiendo guardar información en atributos anidados. Además de permitir al almacenamiento de información como documentos, MongoDB puede ser utilizado para almacenar ficheros de gran tamaño, como fotos o vídeos [35]. Por otra parte, cabe mencionar que este sistema permite el uso de índices georeferenciados, es decir, un índice en dos dimensiones. Esta funcionalidad es especialmente útil ya que permite posicionar los documentos mediante coordenadas, proporcionando una serie de funciones de búsqueda y cálculo de información aplicada a información geográfica, como la búsqueda por distancia o la obtención de documentos dentro de un radio determinado. Esta última funcionalidad nos lleva a la necesidad de introducir el formato GeoJSON, que no es otra cosa que una estructura JSON adaptada para almacenar información geográfica. Permite guardar elementos como puntos (direcciones), líneas (calles, fronteras, etc.) o polígonos (provincias, países, etc.).

CouchDB es un sistema de código abierto hecho especialmente para la Web [3]. Es similar al anterior, orientado a documentos pero se diferencia por su forma de

gestionar las consultas y el escalado de datos. Por defecto no gestiona de manera tan óptima el escalado de datos como MongoDB. almacena la información utilizando el formato JSON y permite el acceso a sus datos a través de HTTP. Uno de sus puntos fuertes es la consistencia de la base de datos y su facilidad de uso.

Cassandra es un sistema de bases de datos de código abierto creado originalmente por Facebook. Es una base de datos de tipo clave/valor que sigue un esquema para los datos, simmilar a los sistemas relacionales. La estructura que sigue este sistema está optimizada para tareas de escritura, siendo en ocasiones lenta para operaciones de lectura, aunque este problema se puede solucionar debido a la funcionalidad que permite la configuración del nivel de consistencia, para poder obtener un equilibrio entre consistencia y rapidez.

Redis se caracteriza por su rapidez, ya que almacena toda la base de datos en la memoria RAM, y la posibilidad de almacenar estructuras de datos complejas. La rapidez que ofrece es una gran ventaja frente a otras opciones, pero dependiendo de la cantidad de datos que se almacenen el rendimiento puede reducirse considerablemente, si se llega a exceder el tamaño de la memoria.

2.5. Integración

Una vez descritas cada una de las tecnologías pasamos a analizar la existencia de alguna herramienta que haga uso de todas ellas. Estas herramientas no son muy numerosas y no hay ninguna que integre todos los elementos que estamos buscando (CMS, semántica y NoSQL), pero hemos enecontrado alguna que por lo menos tienen dos de las mencionadas características. Tenemos disponibles tecnologías que combinan los CMS con bases de datos no relacionales o CMS que integran semántica.

Por un lado, en cuanto a las tecnologías que combinan los gestores de contenidos con las bases de datos NoSQL hemos encontrado principalmente dos, RubedoCMS [6] y Mongopress [5]. Ambas herramientas son de código abierto y se pueden utilizar de manera gratuita.

RubedoCMS es una plataforma especialmente pensada para BigData ya que se apoya sobre MongoDB. Está principalmente orientada a comercio electrónico. Ofrece soluciones a la escalabilidad de datos a través de una arquitectura distribuida.

Mongopress es un CMS que utiliza MongoDB y PHP un entorno flexible orientado a objetos. Está inspirado Wordpress, y aporta la flexibilidad de los sistemas NoSQL de cara a escalabilidad de contenido, característica que no permite Wordpress debido a su funcionamiento con MySQL. En su funcionalidad básica ofrece herramientas de copias de seguridad, balanceo de carga y gestión de réplicas.

Por otro lado, contamos con herramientas en los que la semántica está integrada en el CMS. En este caso, mencionaremos Webnodes [7] y Flowli [4]. El principal problema de estas dos herrameintas es que no son de código abierto ni gratuitas.

Webnodes es un CMS contruído en torno a la semántica. Cuenta con un gestor de contenido semántico que da la posibilidad al usuario de crear ontologías personalizadas, a partir de las relaciones entre los diferentes contenidos. Esta herramienta se centra en los datos y no en los documentos, en forma de páginas web. De esta manera se consigue una navegación por contenidos, donde son las relaciones entre ellos las que determinan su posición a la hora de mostrarlos en un página web y no el autor que los publica.

Flowli es una plataforma de publicación de contenidos que no se basa en carpetas ni en páginas, es capaz de organizar el contenido introducido a través de su significado. Mediante el uso de algoritmos lingüísticos, resume, etiqueta y clasifica por categorías textos automáticamnete. Esta técnica permite reducir el tiempo de publicación de los contenidos ya que el usuario no se tiene que preocupar de relacionar los contenidos. Debido a este flujo de trabajo, Flowli es capaz de construir de manera dinámica la información que se va a mostrar a un usuario, basándose, por ejemplo, en sus preferencias.

Estas herramientas mencionadas, aunque presentan ciertas ventajas, no nos parecen las adecuadas para realizar este proyecto. Por un lado, no tienen la comunidad de usuarios suficiente como para que pueda ser una herramienta relevante como es el caso de las primeras, y por otro, porque son herramientas de pago y esto es una característica que no se puede tener en cuenta en este trabajo. Por este motivo en el proyecto se han utilizado herramientas bastante extendidas en la comunidad, como son Drupal, RDF y MongoDB.

En primer lugar, hemos elegido Drupal debido a que es uno de los CMS de código abierto más utilizados en la actualidad. Y aunque no es el que más cuota

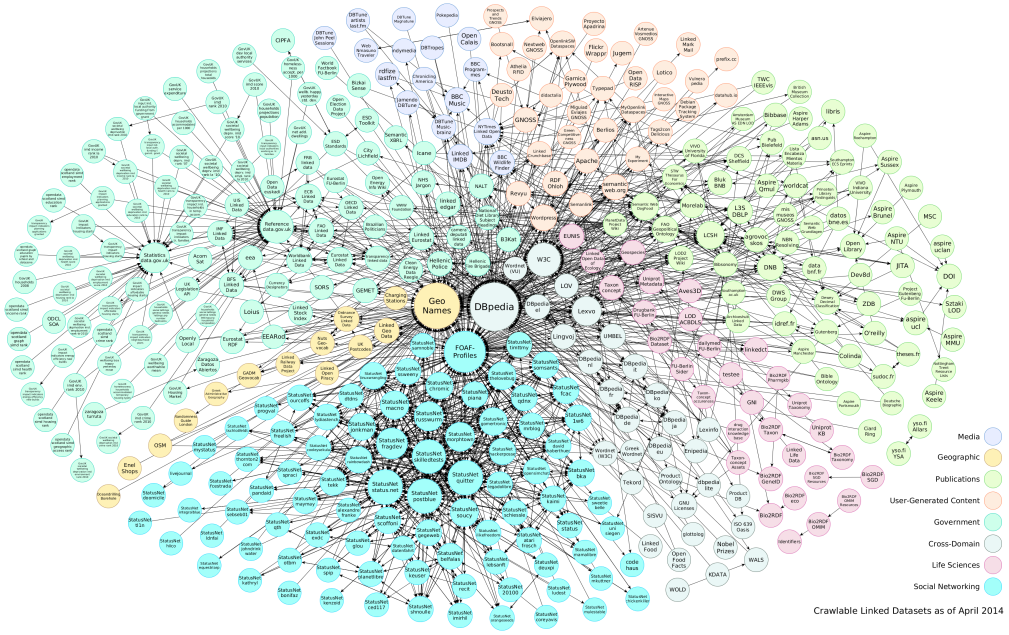

Figura 2.5: Grafo de datos entrelazados (http://lod-cloud.net/)

de mercado abarca, es el que más flexibilidad ofrece de cara a personalización y contrucción de portales de gran tamaño. Además cabe mencionar que Drupal ofrece en sus funcionalidades por defecto una integración básica de RDF, que puede ser ampliada mediante módulos.

Aprovechando el modo de funcionamiento de Drupal, se puede añadir al contenido propio de una página todo el abanico de posibilidades que ofrece RDF y así acceder y producir contenido dentro de lo que se denomina *Linked Data* [18] (Figura 2.5). Este termino se refiere a un método para publicar contenido en la Web para que pueda ser consumido no por personas, sino por máquinas. Los cuatro pilares sobre los que se apoya son:

1. Utilizar URIs para nombrar recursos.
2. Hacer uso del formato HTTP de las URIs para permitir la descripción y localización de los recursos.
3. Proporcionar información de utilidad a través de la URI.
4. Incluir otras URIs de modo que se pueda acceder a otra información.

Aunque Drupal utiliza MySQL como SGBD por defecto, y se caracteriza por ser muy rápido en operaciones de lectura, puede provocar problemas en un entorno donde haya una gran cantidad de datos. Para solucionar este problema de escalabilidad Drupal permite la posibilidad de utilizar MongoDB, aunque es una opción en fase de pruebas y su integración no es completa del todo.

Una vez analizados los diferentes aspectos de la web y las herramientas disponibles para la construcción de contenido semántico, en el siguiente capítulo vamos a explicar las fases que hemos seguido a la hora de realizar este proyecto.

Capítulo 3

Fases y desarrollo del proyecto

En el presente capítulo exponemos las diferentes fases llevadas a cabo durante la realización del proyecto, así como el desarrollo del mismo.

La fases que hemos realizado son: análisis de la estructura del sistema anterior, creación del nuevo sistema, migración de datos, mapeo de datos con etiquetas RDF, puesta en marcha de servicios web para consumo de datos, uso de datos externos, pruebas de de etiquetado y exportación de datos a MongoDB.

3.1. Situación actual

En primer lugar, cabe destacar que el punto de partida de este trabajo es una plataforma web existente [12], por lo que la fase inicial de este trabajo ha sido realizar un análisis exhaustivo de la estructura ya existente y de todos los datos disponibles, para poder tomar decisiones de cara al desarrollo del nuevo sistema, enfocado a la facilidad de uso y mantenimiento.

El sistema actual sobre el que trabajan los investigadores es una plataforma web obsoleta, de difícil mantenimiento y sin soporte por la empresa instaladora. Una parte clave para facilitar el trabajo de investigación es contar con un buen buscador de contenidos y poder acceder a la información de manera clara y rápida. El sistema de búsqueda existente funciona por medio de índices introducidos manualmente y no existe ningún etiquetado semántico del contenido existente, ni ningún mecanimo que permita su incorporación. Por estas razones hemos acordado con el grupo de

investigadores de la Cátedra del Patrimonio Inmaterial de Navarra [10] la creación de una nueva plataforma.

3.2. Diseño de la base de datos

En primer lugar se pretende modelar la base de datos en la que se van a almacenar todos los datos del nuevo sistema. Para ello analizaremos las necesidades de la citada Cátedra.

En este contexto los contenidos que podemos gestionar son:

- Usuarios, que pueden ser administradores, investigadores, traductores o personas de la comunidad, externas a la Cátedra.
- Recursos. Son elementos que forman parte de los testimonios y pueden ser de diferentes tipos como, vídeos, audios, imágenes, textos o partituras musicales.
- Informantes. Son personas que aportan información sobre un acontecimiento cultural.
- Agentes culturales. Son colectivos o asociaciones que incorporan conocimiento cultural.
- Localidades. Son los diferentes municipios que hay en la comunidad.
- Paisajes culturales. Son zonas
- Agrupamientos. Es una entidad que se encarga de combinar localidades o paisajes en conjuntos que representan zonas, rutas, itinerarios, etc.
- Testimonios o manifestaciones culturales. Contienen información relacionada de localidades, informantes y recursos.

usuarios (administradores,), recursos (vídeo, audio, imágenes, texto, partituras musicales), informantes, agentes culturales, localidades, paisajes culturales, agrupamientos (localidades o paisajes), rutas y testimonios o manifestaciones culturales.

De los usuarios nos interesa su nombre, fecha de creación en el sistema, tipo de usuario o rol, de cara a poder asignarle diferentes permisos.

En cuanto al apartado de los recursos, tendremos una parte de los datos comunes. Para identificar cada recurso tendremos un conjunto de atributos que servirán para identificarlo como son un código, generado automáticamente, su título, la fecha de inserción del recurso, el formato original, su uso y el idioma en el que se ha recogido.

Para el apartado de captación del recurso tendremos la fecha y el lugar, su localización (paisaje, localidad, agrupamiento) y su contexto. En cuanto al recurso propiamente dicho se deben guardar datos de la digitalización, el enlace del archivo informático, una breve descripción, anotaciones científicas, el estudio sobre el recurso y diferentes palabras clave que representen al recurso.

En cuanto a los datos técnicos del recurso se analizarán posteriormente de acuerdo a cada uno de los tipos. Para el apartado de la ubicación se necesitan almacenar las referencias bibliográficas, la entidad, el municipio, la ubicación en el almacén y la persona de contacto.

También es necesario poder guardar datos sobre el estado de conservación del recurso y sobre los derechos de cesión del mismo, como los derechos de autor, la cesión de imagen, la cesión de explotación y el acta de cesión. Además se debe permitir un apartado en el que anotar observaciones y relaciones con otros recursos.

Como hemos dicho antes disponemos de diferentes tipos de recursos. Cada uno de ellos tendrá unos campos adicionales según su tipo.

Los vídeos contarán con una tipología (testimonio, manifestación, difusión, presentación, divulgación científica, didáctica), datos técnicos del vídeo (minutos, tc offset, tanda, relación aspecto, color y calidad), además se contará con un apartado en el que guardar una transcripción del vídeo.

De los recursos audio necesitamos los mismos campos que en el tipo vídeo, salvo la relación de aspecto y el color. Además loas tipos de audio son diferentes (testimonio, manifestación, difusión, audioguía).

En cuanto a las imágenes, cada una de ellas puede ser de un tipo (testimonio, manifestación, presentación), tendrá un formato determinado, un tamaño y una resolución. Además dependiendo de la foto pueden aparecer personajes.

Para los recursos de tipo texto, tenemos una tipologá definida (testimonio, manifestación, difusión, didáctico o científico). Estos textos tienen una transcripción, un tamaño y una descripción del soporte en el que están escritos.

Por último contamos con partituras musicales que se describen mediante un tipo de representación sonora, un tipo de partitura y una notación musical. Además se caracterizan por tener una versión reproducirle.

También disponemos de información sobre las localidades donde se encuentra el patrimonio. Cada localidad tiene una ficha básica con la que se identifica. Estos datos son la codificación oficial de la localidad, su denominación oficial, su denominación extraoficial (en euskera o en castellano), el tipo de entidad que representa, pudiendo ser un municipio simple, complejo, concejo, lugar habitado, etc., el escudo oficial, el censo actual de población, su extensión en metros cuadrados, su altitud en metros sobre el nivel del mar, la distancia en kilómetros hasta Pamplona y los datos oficiales de la localidad (Dirección del ayuntamiento, email, página web y entidades administrativas).

Además de la ficha identificativa disponemos de información adicional para geoposicionar la localidad (latitud y longitud) y una galería fotográfica de la misma. Además contamos con información específica sobre el patrimonio, como son su situación geográfica, información sobre las lenguas y sus hablantes, patrimonio natural, una breve introducción de su hostoria, datos de patrimonio cultural sobre su arte, las diferentes posibilidades de oferta turística así como enlaces de interés y referencias bibliográficas sobre el patrimonio inmaterial.

También tendremos datos de la relación con otras localidades, es decir, municipios compuestos que dependen de municipio mayor, de los diferentes paisajes culturales asociados a una localidad y agrupaciones (rutas, itinerarios, zonas, vales, etc.) a las que pertenece la localidad. Por último, una localidad puede tener la posibilidad de disponer de patrocinadores.

Vinculado con las localidades, existen los paisajes culturales que también disponen de una ficha identificativa con datos como la codificación oficial, obtenida a partir de la localidad correspondiente, el nombre del paisaje, su tipo, que puede ser por un lado, rural, urbano, industrial o arqueológico y por otro, vivo o relicto. Además cada paisaje debe depender de una localidad y de una imagen que lo identifique. También se tienen datos sobre geolocalización del paisaje. Como información adicional contamos con una breve descipción del paisaje cultural y unos enlaces de interés.

Estos dos últimos elementos, tanto localidadades como paisajes, pueden estar

agrupados, formando parte de una zona, una ruta, un itinerario, un valle, etc. A estas agrupaciones se les asigna un código, construido a partir del de la localidad, un nombre que describa dicho conjunto de entidades, su tipo y el conjunto de localidades o paisajes que la forman. Las agrupaciones tienen además fotografías características, provenientes de las localidades y los paisajes, así como una geolocalización precisa, un texto con la descripción de la agrupación y enlaces de interés.

Otro de los elementos del patrimonio son los informantes y los agentes culturales, que son personas o colectivos que aportan información. En cuanto a los informantes, de manera a poder identificarlos nos interesa su nombre, apellidos, sobrenombre, el nombre de la casa donde vive, su fecha de nacimiento, la localidad en la que nació y en la que vive actualmente, su sexo, la profesión a la que se dedica junto con el cargo que desempeña. Además un informante puede ser individual o colectivo y puede aportar una fotografía suya que le identifique. Por otro lado, la información sobre cada informante queda recogida por un investigador en un año determinado. Si nos referimos a los agentes culturales, estos tendrán un nombre,una fecha de constitución, una bibliografía, una foto y un vídeo que identifiquen al colectivo. También tendrán un lugar de residencia, que será alguna de las localidades del sistema, un CIF que los identifique, el tipo de asociación o agente cultural y su estructura de funcionamiento. Una asociación puede tener varios miembros y se dedica a una tarea determinada. Por otro lado puede contar con una página web con su información y un idioma oficial. Un agente cultural cuenta con un repertorio sobre patrimonio aprendido de otros agentes o informantes, de un contexto determinado, que ejerce influencia en una zona geográfica y dirigido a un público. Dentro de este apartado se cuenta además con referencias bibliográficas. Estos agentes tienen información de contacto como puede ser su dirección, teléfono, e-mail y la persona de contacto. Para poder utilizar los datos proporcionados por los agentes culturales, existe una cesión de imagen y se proporciona el acta de cesión.

Finalmente contamos con los testimonios o manifestaciones culturales, que que son una agrupación de elementos, como pueden ser las localidades, los informantes, los recursos y los investigadores.

Al ser un proyecto que cuenta con una comunidad externa, además de los investigadores del proyecto, que colabora con la aportación de información, los elementos

del patrimonio serán introducidos de manera provisional en el sistema y deberán ser validados por los investigadores para su publicación.

Tras el análisis de las necesidades de almacenamiento de información, la base de datos del sistema que hemos construido queda como se indica en la Figura 3.1. Podemos observar información detallada de los tipos localidad e informantes que podemos ver en la Tabla 3.1 y en la Tabla 3.2 respectivamente.

Figura 3.1: Modelo relacional de la base de datos

Nombre	**Descripción**
Codificación	Codificación oficial de la localidad
Denominación oficial	Nomenclátor oficial
Denominación en euskera / castellano	Nombre de la localidad no oficial
Tipo de entidad	Municipio simple, complejo, concejo, lugar habitado...
Fotografía de la localidad	Escudo oficial de la localidad
Censo de población	Datos de población actualizados
Extensión	Metros cuadrados de la localidad
Altitud	Metros sobre el nivel del mar
Distancia a Pamplona	Kilómetros de distancia desde Pamplona
Entidades inferiores dependientes	Los municipios compuestos tienen concejos y lugares habitados dependientes. Organización local oficial
Paisajes culturales dependientes	Todos los paisajes están asociados a una localidad
Agrupaciones	Rutas, itinerarios, zonas, valles, etc. a las que pertenece la localidad
Datos lingüísticos	Descripción de las lenguas y hablantes
Situación geográfica	Descripción de su situación geográfica
Patrimonio natural	Elementos paisajísticos y naturales destacados
Historia	Resumen de la historia de la localidad
Patrimonio cultural	Elementos patrimoniales destacados: arte, documentos...
Galería	Imágenes de la localidad
Datos oficiales	Dirección del ayuntamiento, email, página web, entidades administrativas a las que pertenece
Informantes	Relación de informantes y/o agentes culturales de una localidad
Oferta turística	Oferta de establecimientos, rutas, actividades...
Enlaces de interés	Otras páginas webs, blogs, etc. sobre la localidad
Patrocinadores	Logo, lema, etc.del patrocinador de la página de la localidad
Geolocalización	Latitud y longitud de la localidad
Bibliografía	Referencias bibliográficas sobre el patrimonio inmaterial

Tabla 3.1: Datos de las localidades

Nombre	**Descripción**
Código	Códificación numérica de cada informante
Nombre	Nombre de pila del informante
Apellidos	Apellido(s) del informante
Sobrenombre	Sobrenombre que se da a una persona por una cualidad o condición suya
Nombre de la casa	Denominacin de la casa natal o de residencia del informante
Fecha de nacimiento (Edad)	Fecha de nacimiento en formato día, mes y año
Lugar de nacimiento	Localidad en la que nació el informante
Sexo	Sexo
Lugar de residencia	Localidad en la que vive el informante
Profesión	Principal trabajo realizado por el informante
Categoría laboral / Cargo	Puesto que desempeña/ desempeñaba el informante en su trabajo
Comunidad o grupo al que pertenece	Nombre del grupo, comunidad o asociación a la que pertenece
Tipología	Amateur, semiprofesional o profesional
Idioma	Idioma utilizado por el informante en las grabaciones
Foto informante	Foto identificativa del informante
Año de recopilación	Fecha en la que se realizó la grabación
Investigador	Nombre y apellidos de la persona, o denominación del agente, que realizó la grabación
Aprendido de	Nombre de la persona o situación en la que aprendió lo grabado
Contextos principales	Situaciones y lugares en los que se realiza el evento grabado
Dirección	Dirección postal del informante
Teléfono	Número de telfono del informante
E-mail	Dirección electrónica del informante
Persona de contacto	Nombre y apellidos de la persona de enlace con el informante
Cesión de imagen	Informacin SI/NO sobre si el informante permite el uso de su imagen
Acta de cesión	Fecha de la firma del documento de cesin de imagen
Archivos y recursos	Todos los recursos que provienen del mismo informante
Observaciones	Otros aspectos relevantes relacionados con la grabación

Tabla 3.2: Datos de los informantes

3.3. Diagramas de modelado

Para describir las diferentes funciones de la aplicacioón realizada hemos creído conveniente utilizar los diagramas de casos de uso, ya que nos darán una representación gráfica de todo lo necesario para poder desarrollar nuestro proyecto. Además nos permite ver como puede interactuar el usuario final, sin profundizar en aspectos como la implementación.

3.3.1. Identificación de actores

Para el proyecto que nos ocupa tenemos un conjunto de actores que interactúan con el sistema. Los diferentes actores disponen de distintos permisos a la hora de utilizar las funcionalidades del sistema.

En primer lugar tenemos el administrador del sistema, que tiene acceso al conjunto del sistema y puede controlar todas las funciones. Es el encargado de gestionar a los demás usuarios, asignar permisos y controlar el funcionamiento del sistema, además de poder crear cualquier tipo de contenido.

Contamos con la figura del investigador, que puede generar contenido y es el encargado de realizar las validaciones del contenido que aporten usuarios externos. Es una figura parecida al administrador, pero no tiene la posibilidad de control total sobre el sistema, solo sobre el contenido.

También existen colaboradores que son actores externos del proyecto, que pueden generar contenido, pero con un menor nivel de permisos, no pudiendo validar información. Dentro de loa colaboradores tenemos una figura especial, que son los traductores, que solamente pueden generar contenido a partir de uno ya existente, a diferentes idiomas, es decir, no pueden crear contenido nuevo.

Finalmente, contamos con la comunidad de usuarios a la que se le permite el envío de información al sistema, sin que esta quede reflejada directamente, sino que tiene que pasar con la corrección y validadción de los investigadores.

3.3.2. Identificación de los casos de uso iniciales

La Figura 3.2 muestra la iteración inicial o de borrador de los casos de uso del sistema. Este diagrama es el más general, de manera que iremos definiendo el sistema

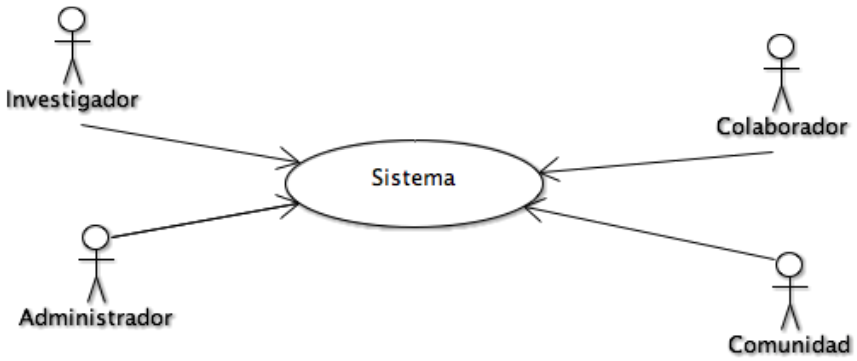

Figura 3.2: Representación del sistema en la iteración draft (borrador)

a desarrollar con más precisión a medida que vayamos avanzando en los sucesivos refinamientos.

Una primera iteración nos permite ver las funcionalidades más generales necesarias en el sistema, como se ve en la Figura 3.3, y los actores que interactúan con los diferentes módulos.

En la segunda iteración profundizamos en las funcionalidades necesarias para llevar a cabo las tareas requeridas en el proyecto.

3.3.3. Gestión de edición

En la Figura 3.4 mostramos, de manera general, aunque su funcionamiento sea diferente para cada tipo de contenido, una serie de módulos que representan las diferentes funciones disponibles para los contenidos del sistema:

- **Creación:** Se encarga de mostrar al usuario una serie de campos que tendrá que rellenar y los guardará en la base de datos.
- **Edición:** Recupera la información de un contenido y se la muestra al usuario, para poder aplicar modificaciones.
- **Borrado:** Muestra una lista de los elementos de un tipo de contenido, para que el usuario pueda elegir aquellos que desea eliminar.

Figura 3.3: Representación del sistema en la primera iteración

- **Traducción:** Lista los contenidos que están disponibles para realizar traducciones.

- **Consultar:** Se encarga de mostrar al usuario el contenido disponible y toda su información.

Podemos ver como cada uno de los actores solo puede acceder a una funcionalidad determinada. Y el Investigador tiene la posibilidad de publicar o no los contenidos introducidos por el resto de actores. Esta funcionalidad utiliza el módulo de validadción por el que se acepta o rechaza dicho contenido.

3.3.4. Gestión de recursos

En el caso de la iteración de los recursos, el módulo de creación queda reflejado en la Figura 3.5. Se pueden crear recursos de varios tipos, que tienen todos ellos una parte de la información en común.

Los módulos que se encargan de la creación de los tipos *Vídeo* y *Audio* permiten la transcripción del audio proporcionado al sistema.

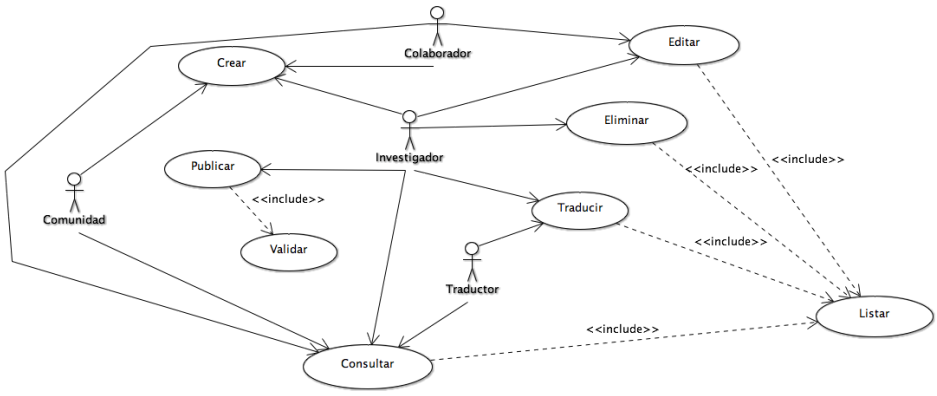

Figura 3.4: Representación general de los módulos de gestión en la segunda iteración

El módulo de creación de partituras musicales incorpora la funcionalidad de reconocimiento de partituras, para posteriormente poder reproducirlas.

3.3.5. Gestión de agrupamientos

Para la creación de agrupamientos la estructura del caso de uso queda reflejada en la Figura 3.6.

Para crear agrupamientos necesitamos tener localidades existentes o paisajes que se listarán para poder elegirlos. Por el contrario,si no hay ninguno de ellos disponible o si una localidad o paisaje no existe todava se puede crear.

3.4. Sistema propuesto

Siguiendo los objetivos marcados, y para facilitar la labor investigadora, en una primera fase de desarrollo, este nuevo sistema permite la creación de localidades, informantes y testimonios en forma de vídeos. Para llevar a cabo la creación de diferentes entidades para cada uno de estos contenidos el funcionamiento básico de Drupal no es suficiente, ya que solo permite crear nodos en los que la información

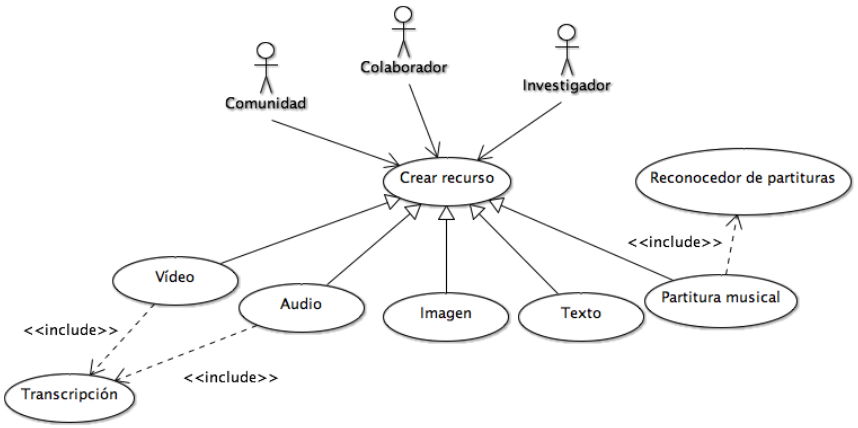

Figura 3.5: Representación de los casos de uso referentes a la creación de recursos en la segunda iteración

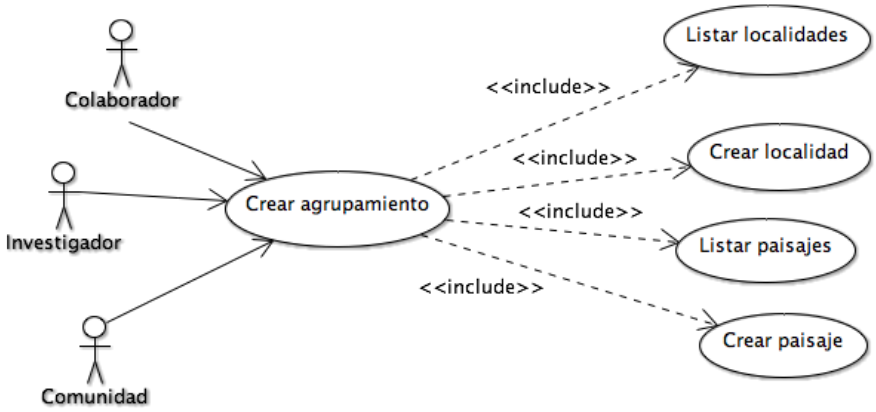

Figura 3.6: Representación de los casos de uso referentes a la creación de agrupamientos en la segunda iteración

permitida se limita a un ti'tulo y un texto, por lo que hemos tenido que extender sus posibilidades mediante el uso de módulos disponibles, detallados a continuación.

Por un lado, hemos utilizado el módulo *Field* para incrementar los diferentes campos que pueda tener un contenido concreto. Este módulo hace uso de la *Field API*, que es la que permite aadir nuevos campos a un nodo, encargándose del alamacenamiento, la carga y la edicin de los datos. La mencionada API sobre la que funciona este módulo, crea en la base de datos dos estructuras, el campo nuevo que se crea y la instancia del mismo. La primera se encarga de almacenar la información que define este nuevo campo, como su nombre, su formato o el tipo de datos. La segunda nos permite saber en que nodo se esta utilizando ese campo. Por otro lado, también crea una tabla para guardar la información que el usuario escriba en ese campo.

Dentro del funcionamiento de Drupal cada uno de estos recursos es, en un nivel más abstracto, un nodo. Para distinguir cada uno de ellos y poder personalizar su funcionamiento, hemos creado un tipo de contenido diferente para cada caso al que tenemos asociados una serie de campos, necesarios para almacenar los diferentes datos marcados por la guía del Plan Nacional.

Una vez creada toda la estructura del nuevo sistema la siguiente fase que hemos realizado ha sido la incorporación de contenido. Debido a que algunos de los datos ya existían en el sistema de gestión anterior, hemos procedido a hacer una migración de datos en la que hemos tenido que mapear la información del sistema antiguo al nuevo. El mapeo ha sido directo en algunos casos, pero en otros, hemos tenido que reubicar los datos ya que algunos de los campos existentes anteriormente han desaparecido en el nuevo sistema, han cambiado de nombre o se han unido varios campos en uno solo.

3.5. Etiquetado semántico

Una vez que ya tenemos la estructura del sistema y todo el contenido, podemos pasar a la siguiente fase, en la que hemos incorporado información semántica a la estructura construida.

Para la realización de esta tarea, también nos hemos apoyado en los módulos

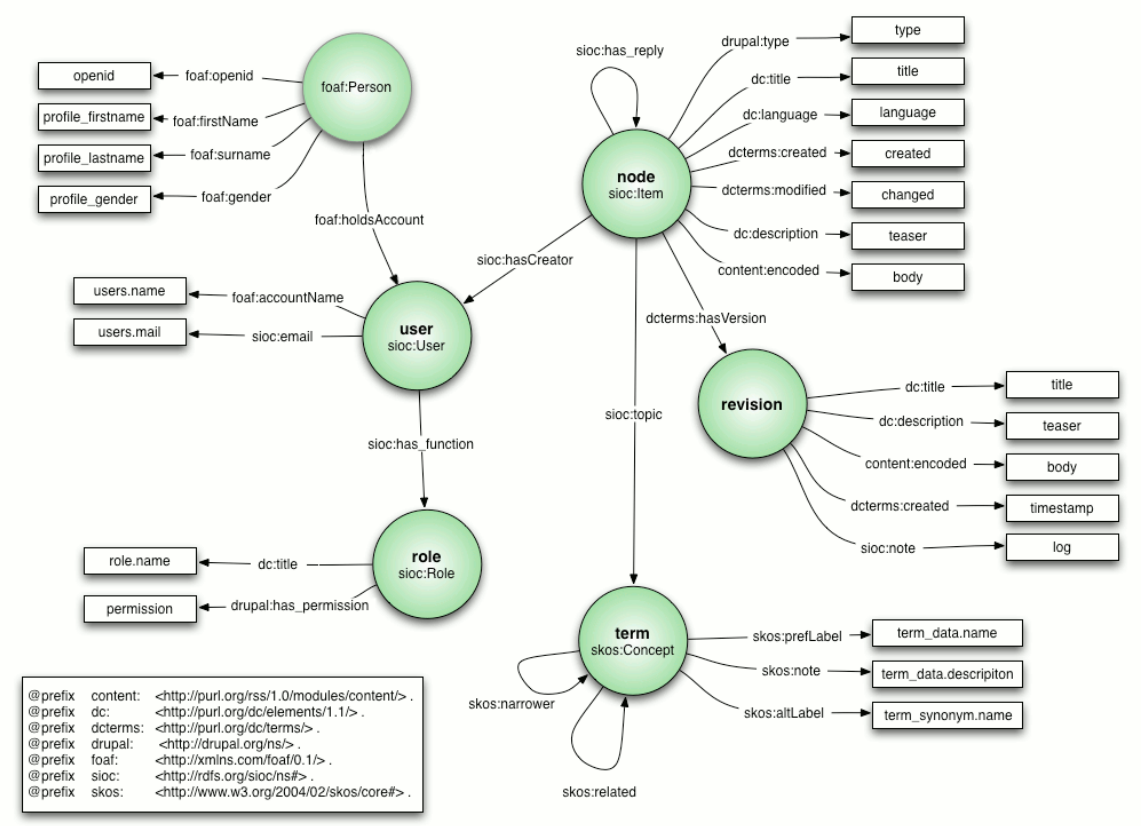

Figura 3.7: Esquema RDF propuesto por defecto de Drupal para la version 7 (http://www.groups.drupal.org)

citados anteriormente y hemos aprovechado la funcionalidad que brinda Drupal para incorporar datos semnticos a través del API de mapeo RDF.

La funcionalidad por defecto que propone Drupal para la versión 7, en cuanto la incorporación de RDF de manera global en el sistema es la que se indica en la Figura 3.7. Los datos contenidos describen de manera muy genérica el contenido y no son suficientes para expresar toda la información incorporada sobre Patrimonio Inmaterial.

Esta funcionalidad, por defecto, solo incorpora información semántica para describir aspectos básicos sobre recursos web. Las anotaciones comunes a todos los tipos de contenido son las indicadas en la Tabla 3.3. Estas etiquetas semánticas utilizan los vocabularios *Dublin Core* (DC), *content*, *Semantically-Interlinked Online Com-*

munities (SIOC) y *Friend of a Friend* (FOAF), explicados más adelante junto con las demás ontologías.

Campo	Predicado RDF	Mapeo	Tipo
title	dc:title	property	
created	dc:date, dc:created	property	xsd:dateTime
changed	dc:modified	property	xsd:dateTime
body	content:encoded	property	
uid	sioc:has_creator	rel	
name	foaf:name	property	
comment_count	sioc:num_replies	property	xsd:integer
last_activity	sioc:last_activity_date	property	xsd:dateTime

Tabla 3.3: Etiquetas semánticas por defecto

Para la versión 8 de Drupal, que se encuentra en fase de desarrollo, el grafo RDF por defecto evoluciona y aade una mayor cantidad de etiquetas al sistema (Figura 3.8).

Debido a que no podemos utilizar la última versión de Drupal, por no disponer de una versión estable, hemos añadido al esquema original de la versión 7 de Drupal las etiquetas que mejor describen la información que contienen.

En la Figura 3.9 podemos ver el grafo RDF propuesto para todo el contenido del sistema, en el que se aprecian las relaciones entre los diferentes componentes.

Nos hemos adaptado a la funcionalidad disponible en Drupal 7 pero enfocando el diseo RDF al esquema de la nueva versión de manera que en un futuro y ante una posible actualización del sistema el etiquetado sea válido.

En cuanto a todos los campos adicionales que hemos añadido, contamos con vocabularios y ontologías disponibles que son comunes con otras fuentes de información:

- **content**. Describe los contenidos la página web [9].
- **cv**. Ontología diseñada para aportar información referente a una persona o a un Curriculum Vitae (CV) [20].
- **dc**. Vocabulario para la descripción de recursos web, así como elementos físicos [16].

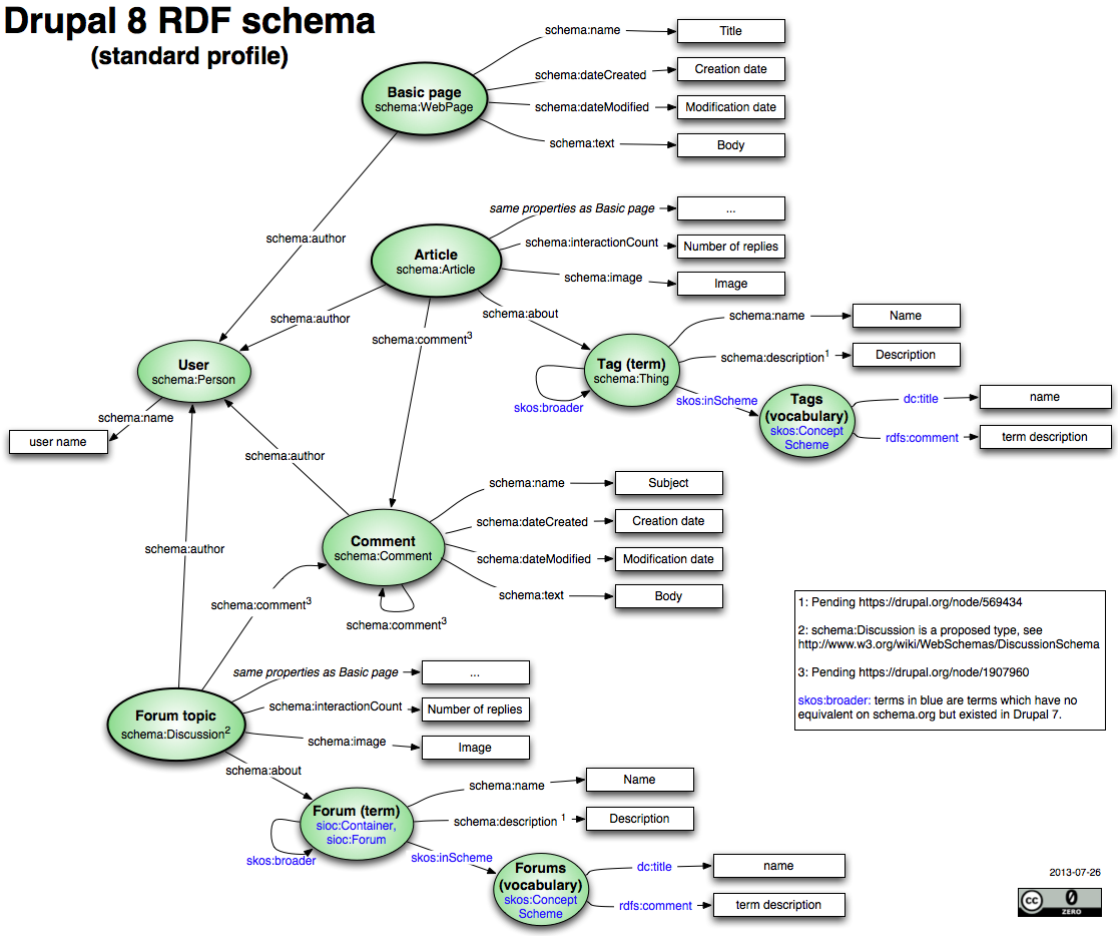

Figura 3.8: Esquema RDF propuesto por defecto de Drupal para la versión 8 (http://www.groups.drupal.org)

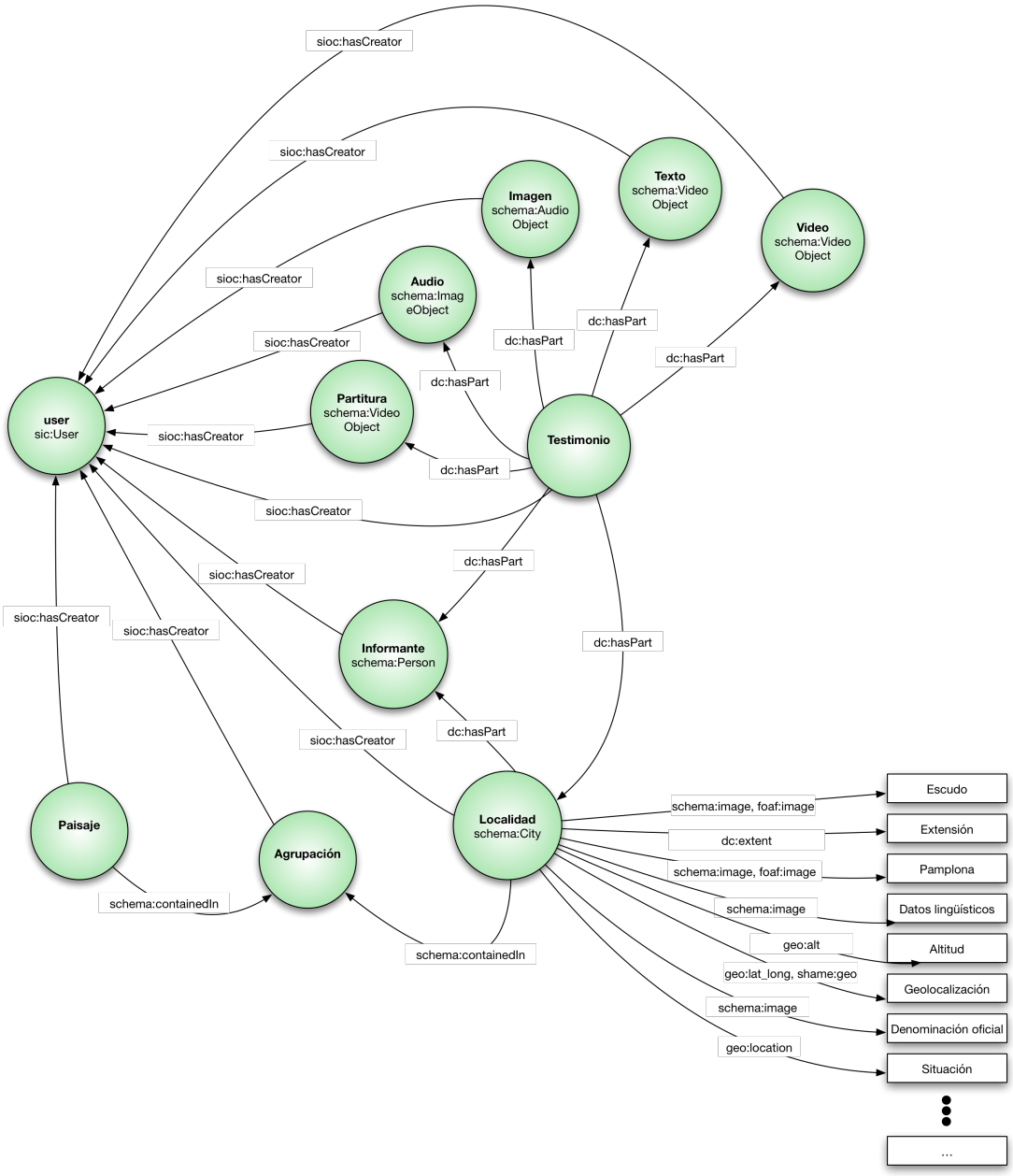

Figura 3.9: Esquema RDF propuesto

- **foaf**. Ontología centrada en la descripción de personas, sus actividades y su relación con otras personas y objetos [24].

- **geo**. Vocabulario para representar información de la latitud, longitud y altitud [21].

- **mads**. Vocabulario para la representación de datos sobre bibliotecas y ciencias de la información [23]. incluye museos, archivos e instituciones culturales. También permite el etiquetado de recursos bibliográficos.

- **og**. Etiquetas del protocolo OpenGraph, utilizado por Facebook. Permite aadir información sobre la página, así como objetos multimedia. [27].

- **rdfs**. Vocabulario con un conjunto de clases que permiten la desripción de ontologías [40].

- **sioc**. Ontología que proporciona conceptos y propiedades para la descripción de información sobre comunidades online [26].

- **sioct**. Extensión que define subcalses y subpropiedades de la ontología SIOC [15].

- **skos**. Siglas de Simple Knowledge Organization System. Proporciona un modelo para la representación de tesauros, taxonomías, esquemas de clasificación o cualquier tipo de vocabulario controlado. [38].

- **xsd**. Es una especificación de cómo se deben describir formalmente elementos en un documento XML [17].

- **owl**. Es un lenguaje de representación del conocimiento que permite definir ontologías para la web [39].

- **rdf**. Representa información general sobre la web [33].

- **schema**. Proporciona un esquema de marcado de datos común para los principales buscadores existentes [19].

Cada una de las ontologías anteriores cuenta con términos que describen alguna propiedad o relación. Para los tipos de contenido de este sistema se han utilizado algunos de los términos que mejor describen el contenido que hemos creado en el sistema. Como podemos observar en la Tabla 3.4 tenemos el mapeo correspondiente a las localidades y en la Tabla 3.5 el de los informantes.

Campo	**Predicado RDF**	**Mapeo**	**Tipo**
Imagen de la localidad	schema:image, foaf:img	rel	
Extensión	dc:extent	property	xsd:float
Pamplona (distancia)	dc:extent	property	xsd:float
Datos lingüísticos	dc:language	property	xsd:string
Altitud	geo:alt	property	xsd:float
Geolocalización	geo:lat_long, schema:geo	property	
Tipo de localidad	dc:type	property	xsd:int
Paisajes culturales a los que pertenece	dc:isPartOf	rel	
Informantes	dc:hasPart	property	
Galería	foaf:img, schema:photo	rel	
Enlaces de inters	sioc:links_to	rel	
Título	dc:title	property	xsd:string
Denominación oficial	foaf:name, schema:name	property	xsd:string
Situación	geo:location	property	xsd:string
Bibliografía	dc:bibliographicCitation	property	
Historia	mads:historyNote, skos:historyNote	property	xsd:string
Código	mads:code	property	
Agrupaciones	schema:containedIn	property	xsd:string
Datos oficiales de la localidad	schema:description	property	xsd:string
Otras denominaciones	schema:alternateName	property	xsd:string
url	schema:url	rel	

Tabla 3.4: Etiquetas semánticas para las localidades

Campo	Predicado RDF	Mapeo	Tipo
Nombre	foaf:surname, sioc:first_name	property	xsd:string
Apellidos	foaf:family_name, sioc:last_name	property	xsd:string
Fecha	schema:birthDate	property	xsd:date
Sexo	foaf:gender, schema:gender	property	xsd:string
Tipo de informante	dc:type property	xsd:string	
Idioma	dc:language	property	xsd:language
Telfono	foaf:phone	property	xsd:integer
Derechos de cesin de imagen	dc:rights	property	xsd:string
Foto	foaf:img	property	
Profesin	cv:jobDescription	property	xsd:string
Cargo que ocupa en su profesin	cv:jobTitle	property	xsd:string
Lugar de nacimiento	schema:birthPlace	property	xsd:string
Sobrenombre	schema:additionalName	property	
Comunidad a la que pertenece	schema:memberOf	property	xsd:string
Direccin	schema:address	property	xsd:string
Correo electrnico	schema:email	property	xsd:string
Relaciones	schema:relatedTo	property	xsd:string
Lugar de residencia	schema:homeLocation	property	xsd:string

Tabla 3.5: Etiquetas semánticas para los informantes

3.6. Diseño de BBDD NoSQL

En cuanto al apartado de almacenamiento de la información, hemos realizado pruebas con una parte de la base de datos original del sistema. Concretamente hemos exportado todos los nodos a una base de datos sobre MongoDB, de manera a tener la información preparada para trabajos de *BigData*. La exportación directa de los nodos de Drupal consiste en construir un array con todo el contenido y transformarlo al formato JSON para insertarlos en MongoDB. Esta técnica hace que los campos estén anidados y se crea una estructura compleja que dificulta bastante las consultas sobre MongoDB. Además el sistema crea una sola colección con todos los nodos exportados, sin importar el tipo de contenido.

En el caso de las localidades, para los campos que no son traducibles la exportación se realiza del siguiente modo:


```
"field_localidad_extension" : {
  "und" : [
    {
      "value" : "4.00"
    }
  ]
}
```


En el caso de la galería de imágenes en el que tenemos los títulos de las imágenes traducibles la exportación resulta de un modo similar:


```
"field_localidad_galeria" : {
  "es" : [
    {
      "fid" : "1738",
      "uid" : "1",
      "filename" : "",
      "uri" : "public://galerias_localidades/abaigar/1.jpg",
      "filemime" : "image/jpeg",
      "filesize" : "1596193",
      "status" : "1",
      "timestamp" : "1414436787",
      "type" : "image",
      "image_dimensions" : {
        "width" : "2304",
        "height" : "1536"
      },
      "alt" : "Vista de Abáigar",
      "title" : "Vista de Abáigar",
      "width" : "2304",
      "height" : "1536"
    },
        {
      "fid" : "1739",
      "uid" : "1",
```

```
      "filename" : "",
      "uri" : "public://galerias_localidades/abaigar/2.jpg",
      "filemime" : "image/jpeg",
      "filesize" : "1828341",
      "status" : "1",
      "timestamp" : "1414436787",
      "type" : "image",
      "image_dimensions" : {
        "width" : "2304",
        "height" : "1728"
      },
      "alt" : "Vista de la Iglesia de Abáigar",
      "title" : "Vista de la Iglesia de Abáigar",
      "width" : "2304",
      "height" : "1728"
    }
  ]
}
```

Para el campo en el que se almacenan los datos administrativos de una localidad el mapeado queda de la siguiente manera:

```
"field_localidad_datos" : {
    "es" : [
        {
            "value" : "<p>Información sobre el municipio de Abáigar
               ...",
            "summary" : "",
            "format" : "full_html",
            "safe_value" : "<p>Información sobre el municipio de Abá
               igar...",
            "safe_summary" : ""
        }
    ],
    "eu" : [
        {
            "value" : "<p>Abaigarko udalari buruzko informazioa...",
            "summary" : "",
            "format" : "full_html",
```

```
            "safe_value" : "<p>Abaigarko udalari buruzko informazioa
                ....",
            "safe_summary" : ""
        }
    ]
}
```

Para que los datos se puedan manejar más fácilmente proponemos una estructura alternativa, más acorde con la filosofía de MongoDB. Consideramos cada uno de los tipos de contenido como un documento diferente y creamos una colección para cada uno con tosos sus campos agrupados, eliminando la información interna de Drupal y los anidamientos de información innecesarios. De esta manera los únicos campos con información anidada son aquellos a los que se les permite una traducción.

```
{
  "field_localidad_extension" : "4.00",
  "field_localidad_galeria" : [
    {
      "uri" : "public://galerias_localidades/abaigar/1.jpg",
      "alt" : [
        {"es": "Vista de Abáigar"},
        {"eu": "..."},
        {"fr": "..."},
        {"en": "..."}
      ],
      "title" : [
        {"es": "Vista de Abáigar"},
        {"eu": "..."},
        {"fr": "..."},
        {"en": "..."}
      ]
    },
    {
      "uri" : "public://galerias_localidades/abaigar/2.jpg",
      "alt" : [
        {"es": "Vista de la Iglesia de Abáigar"},
        {"eu": "..."},
        {"fr": "..."},
```

```
        {"en": "..."}
      ],
      "title" : [
        {"es": "Vista de la Iglesia de Abáigar"},
        {"eu": "..."},
        {"fr": "..."},
        {"en": "..."}
      ]
    }
  ],
    "field_localidad_datos" : [
        {"es" : "<p>Información sobre el municipio de Abáigar: C/
           San Vicente, s/n. C. P.: 31280, Abáigar. Tfnol.: 948 534
           257..."},
        {"eu" : "<p>Abaigarko udalari buruzko informazioa: San
           Vicente, z/g. P. K.: 31280, Abaigar. Tfnoa.: 948 534257
           ..."}
    ]
}
```

De esta manera podemos realizar un manejo de datos a gran escala y facilitar la consulta de los datos, así como ciertas funcionalidades, como por ejemplo la geolocalizacin, que es una parte muy importante y que se quiere potenciar por parte de la Cátedra. Para este aspecto existe en MongoDB una funcionalidad que se llama índice 2D, que consiste en asociar los documentos almacenados con una localización en el espacio de las dos dimensiones, es decir, un punto en un mapa. Este índice nos permite realizar consultas sobre los documentos basadas en su proximidad o en su posición en un área determinada. Esta característica se puede aplicar en este nuevo sistema para relacionar las localidades, con los diferentes testimonios de cada una de ellas o para poder crear zonas para clasificar los conocimientos almacenados en el archivo.

3.7. Geolocalización

Una parte importante de este proyecto es la geolocalización de contenidos y por consiguiente su visualización en mapas, en los que podemos aprovechar los datos de

los contenidos añadidos al sistema, teniendo una visión global de los mismos. Así, a través de los mapas veremos todas las localidades situadas, junto con su censo. Además para dotar de más funcionalidad y aprovechar iniciativas Open Data, hemos aadido contenido externo proveniente de Open Data Navarra [8]. En concreto, hemos elegido datos sobre el Camino de Santiago y la zonificación lingüística de Navarra.

La tecnología que hemos utilizado para generar los mapas ha sido CartoDB [14]. Es una plataforma en la nube que proporciona un Sistema de Información Geográfico (GIS, del inglés *Geographic Information System*) y herramientas de creación de mapas para poder mostrarlos en un navegador web. CartoDB nos ha permitido incorporar datos de diversas fuentes en distintos formatos para crear mapas interactivos, mostrando las diferentes informaciones por capas Figura 3.10, concretamente una capa por cada tipo de dato importado. Al utilizar la versión gratuita de este servicio, hemos tenido que importar los datos manualmente, lo que no nos ha permitido utilizar la funcionalidad de actualizar automáticamente los datos cada cierto tiempo, proporcionando una dirección web de origen.

Con esta integración de datos a modo de mapa, proporcionamos a los usuarios una manera rápida y visual de acceder a los contenidos del sistema y relacionarlos con otros de forma automática.

3.8. Disponibilidad de datos mediante Servicios Web

Tras la incorporación de etiquetas semánticas al contenido del sistema, no solo queremos que estén disponibles directamente incorporadas en las diferentes páginas que genera el sistema, para proporcionar así a los buscadores una mayor cantidad de información que entiendan, sino que hemos hecho que los contenidos se puedan consultar de forma individual a través de servicios web, en diferentes formatos. Un Servicio Web es un método de comunicación que permite a dos máquinas el intercambio de datos a través de una red. Existen principalmente dos tipos de tecnología para poner en funcionamiento esta tecnología:

- Los Servicios Web de tipo *Representational state transfer* (REST), que son capaces de servir la información como un conjunto de recursos (URI) identificados

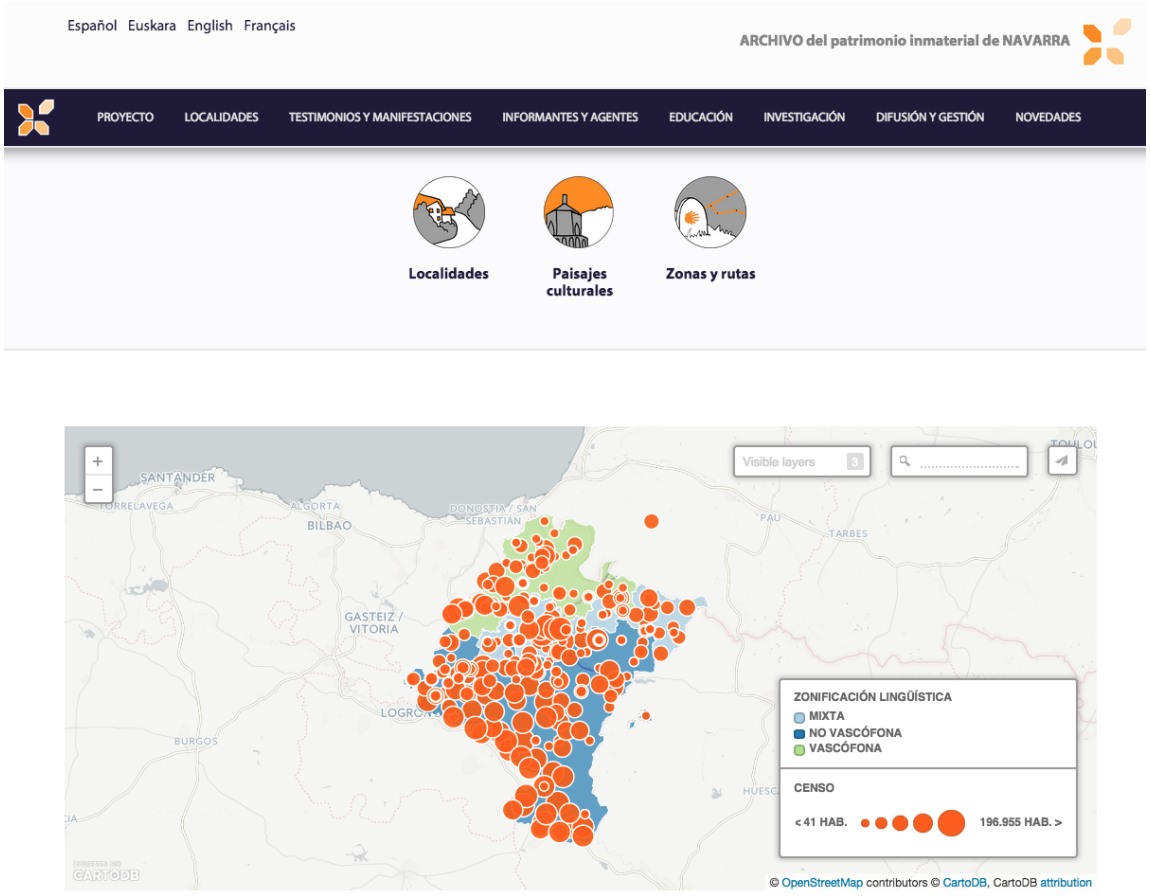

Figura 3.10: Mapa con cartoDB

y accesibles a través del protocolo HTTP.

- Los Servicios Web *WS-**, que sirven la misma información, pero haciendo uso de los protocolos SOAP y WSDL, que se apoyan en XML para describir el contenido del servicio.

La tecnología que hemos elegido para este proyecto ha sido la primera, por su facilidad de puesta en marcha. Para ello hemos utilizado la herramienta *RESTful Web Services* que nos ha permitido añadir al sistema la posibilidad de servir la información disponible en diversos formatos. Esta información es accesible desde cada una de las páginas generadas por el sistema, añadiendo el formato en el que el usuario quiera obtener la información:

- **XML:** En este formato mostramos el contenido de los diferentes campos de un contenido concreto (Figura 3.11).
- **JSON:** En este formato mostramos todos los datos de un nodo, incluyendo toda la información, como por ejemplo el tipo de imagen o su tamaño (Figura 3.12).
- **RDF:** Generamos un documento con las etiquetas del contenido que hemos mapeado, en formato RDF-XML (Figura 3.13).
- **JSON-LD:** Este tipo de formato nos permite exportar las mismas etiquetas que el RDF, pero utilizando JSON adaptado a Linked Data (Figura 3.14).

```
<node>
  <title_field>Abáigar</title_field>
  <field_localidad_extension>4.00</field_localidad_extension>
  <field_localidad_altitud>498 m.</field_localidad_altitud>
  <field_localidad_pamplona>58.00</field_localidad_pamplona>
  <field_localidad_paisajes/>
  <field_localidad_agrupaciones/>
  <field_localidad_galeria/>
  <field_localidad_interes/>
  <field_localidad_geolocalizacion>42.6630587,-2.149935599999935</field_localidad_geolocalizacion>
  <field_localidad_codigo>310010000</field_localidad_codigo>
  <field_localidad_censo>98</field_localidad_censo>
  <field_localidad_denom_oficial>Abáigar</field_localidad_denom_oficial>
  <field_localidad_informantes/>
  <field_localidad_tipologia>1</field_localidad_tipologia>
  <nid>5577</nid>
  <vid>5577</vid>
  <is_new/>
  <type>localidad</type>
  <title>Abáigar</title>
  <language>es</language>
  <url>...</url>
  <edit_url>...</edit_url>
  <status>1</status>
  <promote>0</promote>
  <sticky>0</sticky>
  <created>1414674612</created>
  <changed>1418660060</changed>
  <author resource="user" id="1">http://localhost/~chivo/navarchivoDP/?q=es/user/1</author>
  <log/>
  <views>0</views>
  <day_views>0</day_views>
</node>
```


Figura 3.11: Contenidos servidos en formato XML

```
{
  "title_field": "Abáigar",
  "field_localidad_imagen": {
    "fid": "4545",
    "uid": "1",
    "filename": "",
    "uri": "public://import/fotos/escudos/abaigar.escudo.jpg",
    "filemime": "image/jpeg",
    "filesize": "12763",
    "status": "1",
    "timestamp": "1414519476",
    "type": "image",
    "media_title": [],
    "media_description": [],
    "field_tags": [],
    "field_license": [],
    "rdf_mapping": [],
    "image_dimensions": {
      "width": "137",
      "height": "173"
    },
    "alt": "",
    "title": "",
    "width": "137",
    "height": "173"
  },
  "field_localidad_extension": "4.00",
  "field_localidad_altitud": "498 m.",
  "field_localidad_pamplona": "58.00",
  "field_localidad_paisajes": [],
  "field_localidad_agrupaciones": [],
```


Figura 3.12: Contenidos servidos en formato JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/terms/"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:mads="http://www.loc.gov/mads/rdf/v1#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:schema="http://schema.org/"
  xmlns:sioc="http://rdfs.org/sioc/ns#"
  xmlns:content="http://purl.org/rss/1.0/modules/content/">

  <rdf:Description rdf:about="http://localhost/~chivo/navarchivoDP/?q=es/node/5577">
    <rdf:type rdf:resource="http://schema.org/City"/>
    <rdf:type rdf:resource="http://rdfs.org/sioc/ns#Item"/>
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Abáigar</dc:title>
    <dc:extent rdf:datatype="http://www.w3.org/2001/XMLSchema#float">4.00</dc:extent>
    <dc:extent rdf:datatype="http://www.w3.org/2001/XMLSchema#float">58.00</dc:extent>
    <geo:alt rdf:datatype="http://www.w3.org/2001/XMLSchema#float">498 m.</geo:alt>
    <geo:location rdf:datatype="http://www.w3.org/2001/XMLSchema#string">&lt;p&gt;Limita
al N con Murieta y al E con Igúzquiza y Villamayor de Monjardín, al S con Olejua y al O
con Oco. El término municipal se extiende entre el Ega y el Alto de los Encinos, al SO
de Monjardín, y su altitud se halla comprendida entre los 460 m. del río. La mitad S es
la más accidentada y se corresponde con los terrenos detríticos (conglomerados,
areniscas y arcillas) del Oligoceno-Mioceno: la mitad N, llana, con arcillas miocénicas
de la cubeta de Murieta: entre ambas formaciones está la falla que une Codés con
Monjardín.&lt;/p&gt;
&lt;p&gt;Comunicaciones: carretera local con enlace a la comarcal Estella-Vitoria NA
132-A.&lt;/p&gt;
</geo:location>
```

Figura 3.13: Contenidos servidos en formato RDF

```
{
  "@context": {
     "content": "http://purl.org/rss/1.0/modules/content/",
     "cv": "http://rdfs.org/resume-rdf/cv.rdfs#",
     "dbp": "http://dbpedia.org/property/",
     "dc": "http://purl.org/dc/terms/",
     "foaf": "http://xmlns.com/foaf/0.1/",
     "geo": "http://www.w3.org/2003/01/geo/wgs84_pos#",
     "mads": "http://www.loc.gov/mads/rdf/v1#",
     "og": "http://ogp.me/ns#",
     "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
     "sioc": "http://rdfs.org/sioc/ns#",
     "sioct": "http://rdfs.org/sioc/types#",
     "skos": "http://www.w3.org/2004/02/skos/core#",
     "xsd": "http://www.w3.org/2001/XMLSchema#",
     "owl": "http://www.w3.org/2002/07/owl#",
     "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
     "rss": "http://purl.org/rss/1.0/",
     "site": "http://localhost/~chivo/navarchivoDP/?q=es/ns#",
     "schema": "http://schema.org/"
  },
  "@id": "http://localhost/~chivo/navarchivoDP/?q=es/node/5577",
  "@type": [
     "http://schema.org/City",
     "http://rdfs.org/sioc/ns#Item",
     "http://xmlns.com/foaf/0.1/Document"
  ],
  "http://purl.org/dc/terms/title": [
     {
        "@value": "Abáigar",
        "@type": "xsd:string"
     }
  ],
  "http://purl.org/dc/terms/extent": [
     {
        "@value": "4.00",
        "@type": "xsd:float"
     },
     {
        "@value": "58.00",
        "@type": "xsd:float"
     }
  ],
  "http://www.w3.org/2003/01/geo/wgs84_pos#alt": [
     {
        "@value": "498 m.",
        "@type": "xsd:float"
     }
  ],
```


Figura 3.14: Contenidos servidos en formato JSONLD

Capítulo 4

Pruebas

4.1. Pruebas de etiquetado

Una vez que hemos realizado todo el etiquetado semántico de los datos del sistema, podemos comparar el número de etiquetas reconocidas por diferentes sistemas disponibles y así comprobar la funcionalidad que aporta la semántica. Hemos realizado pruebas con las herramientas *RDFa Test Suite* [1] que nos permite ver las etiquetas que detecta en forma de grafo (Figura 4.1).

También hemos hecho un test con la herramienta de detección de datos estructurados de Google [2] en la que a partir de una dirección web o un código HTML nos devuelve los datos encontrados (Figura 4.2).

Para comparar el etiquetado realizado hemos utilizado las mismas herramientas con el sistema actual y tanto la creación del grafo RDF como el analizador de Google, no detectan ninguna etiqueta.

Se puede ver que todas las etiquetas utilizadas para describir los contenidos no se encuentran disponibles en su totalidad en estas pruebas. Esto se debe a que tras la creación del nuevo sistema y la migración de datos desde el anterior, para seguir la guía del Plan Nacional, han sugido nuevos campos de información que no han sido completados por los inverstigadores.

Por otro lado hemos analizado los documentos RDF generados por el sistema, tras el etiquetado, con la herramienta *Ontology Metrics* [1] de la Universidad de Manchester

[1] http://owl.cs.manchester.ac.uk/metrics/

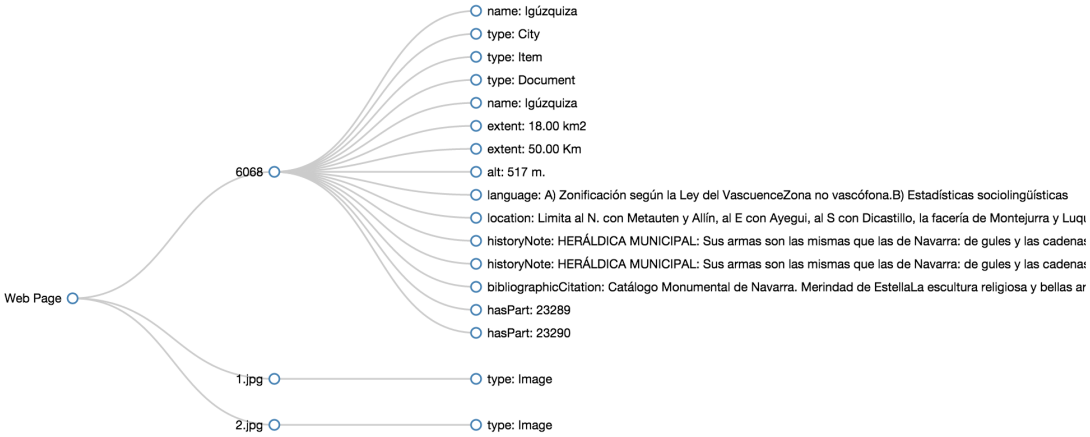

Figura 4.1: Grafo RDF resultante de una localidad

Datos estructurados extraídos

rdfa-node

type:	City
relationship:	
name:	type
value:	Municipio simple
href:	Municipio simple
property:	
name:	Abáigar
name:	Abáigar
extent:	4.00 km2
alt:	498 m.
extent:	58.00 Km
lat_long:	42.6630587,-2.149935599999935
geo:	42.6630587,-2.149935599999935
location:	Limita al N con Murieta y al E con Igúzquiza y Villamayor de Monjardín, al S con Olejua y al O con Oco. El término municipal se extiende entre el Ega y el Alto de los Encinos, al SO de Monjardí...
historyNote:	HERÁLDICA MUNICIPAL. Cuartelado: 1.º y 4.º de gules y una estrella de ocho puntas de oro. 2.º y 3.º de oro y dos lobos de sable andantes, puestos en pal, armados, lampasados y membrados...
historyNote:	HERÁLDICA MUNICIPAL. Cuartelado: 1.º y 4.º de gules y una estrella de ocho puntas de oro. 2.º y 3.º de oro y dos lobos de sable andantes, puestos en pal, armados, lampasados y membrados...

Figura 4.2: Detección de datos estructurados con la herramienta de Google

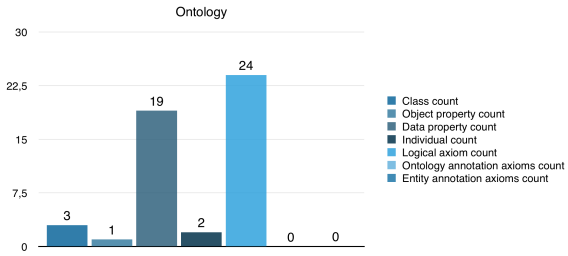

Figura 4.3: Métricas generales presentes en el documento RDF/XML

[2]. Ontology Matrics es una utilidad en la nube para validar y extraer estadísticas sobre una ontología, a partir de un documento OWL o RDF/XML. Podemos medir la cantidad de axiomas y etiquetas presentes en un contenido.

Por un lado se analizan estadísticas (Figura 4.3) generales de la ontología que tiene el documento:

- Número de clases.
- Número de propiedades de objetos
- Número de propiedades de datos
- Número de instancias.
- Número de axiomas lógicos.
- Número de axiomas de anotaciones de la ontología.
- Número de axiomas de anotaciones de las entidades.

Además esta herramienta también nos permite extraer información de los diferentes axiomas perteneciaentes a las instancias presentes en una página del sistema (Figura 4.4). En este caso se analizan los siguientes aspectos:

- Número de axiomas de clase
- Número de propiedades de los objetos

[2] http://www.manchester.ac.uk/

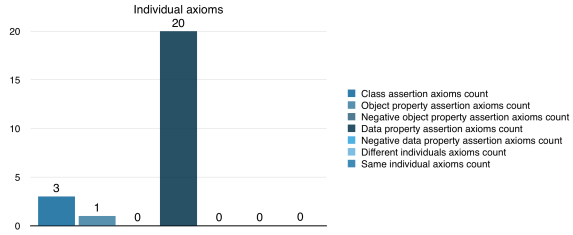

Figura 4.4: Análisis de los axiomas individuales presentes en el documento RDF/XML

- Número de propiedades de negación de los objetos
- Cantidad de propiedades de negación de los datos
- Número de propiedades de negación de datos
- Different individuals axioms count Cantidad de axiomas referentes a instancias diferentes
- Cantidad de axiomas referentes a instancias iguales.

Finalmente, se extrae información de los axiomas referentes a las clases de la ontología de la página (Figura 4.5). En este apartado se tienen en cuenta las siguientes características:

- Cantidad de reglas lógicas encontradas.
- Número de axiomas que hacen referencia a una sublase
- Número de axiomas equivalentes.
- Cantidad de reglas referentes a clases disjuntas.
- Número de axiomas de uniones disjuntas presentes.

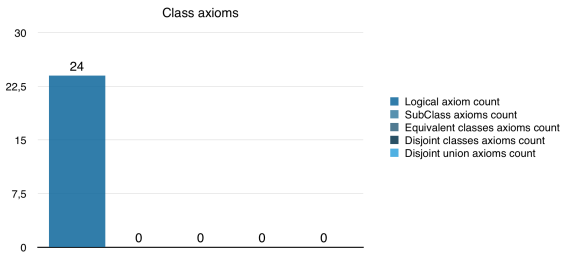

Figura 4.5: Análisis de los axiomas de clase presentes en el documento RDF/XML

Capítulo 5

Conclusiones y líneas futuras

A lo largo de este proyecto hemos realizado diversas tareas que nos han llevado a la creación de la aplicación deseada, cumpliendo los objetivos marcados. Tras haber analizado la evolución de la web actual y el funcionamiento de la web semántica, barajamos diversas posibilidades entre las herramientas más utilizadas para la gestión de contenidos en línea, para la incorporación de semántica a la web convencional y alternativas de almacenamiento para grandes cantidades de datos.

Tras el proceso de análisis de herramientas, pasamos a la siguiente fase en la que pusimos en marcha la estructura que gestionará todo el funcionamiento del sistema de gestión del Patrimonio Inmaterial de Navarra, incorporando etiquetas semánticas al contenido para facilitar la comprensión de los datos por parte de una máquina.

En cuanto a los sistemas bases de datos, podemos decir que aportan funcionalidades que van a facilitar el manejo de los datos que van a ir creciendo a lo largo del tiempo en este proyecto, además de proveer de herramientas para el análisis y el tratamiento de datos a gran escala. En este apartado hemos trabajado con una parte de los datos disponibles mediante la exportación a MongoDB y hemos trabajado con trabajado con un diseo de una sola colección. También cabe destacar las funcionalidades orientadas a geolocalización, que son de gran importancia para los trabajos de investigación de la Cátedra y para la creación de aplicaciones didácticas y de transmisión del conocimiento.

Finalmente, podemos concluir que este proyecto nos ha aportado la posibilidad de aplicar los conocimientos aprendidos a lo largo del máster, ademas de profundizar en

ciertos aspectos y utilizar otros nuevos descubiertos durante el desarrollo del mismo. Además al haber realizado el proyecto dentro del marco de la Cátedra de Patrimonio Inmaterial de Navarra con un grupo de investigación nos ha permitido trabajar en un entorno real en el que hemos podido interactuar de manera activa con los usuarios de la aplicacion, pudiendo hacer evolucionar el proyecto de manera más rápida y acorde a las exigencias de los usuarios.

Aunque se da por finalizado el proyecto debido al cumplimiento de los objetivos inicialmente marcados, este es escalable y siempre puede caber la posibilidad de ciertas ampliaciones o mejoras. Se consideran las siguientes líneas futuras:

- Incorporación de geolocalización a todos los contenidos del sistema.
- Pruebas de rendimiento con diferentes esquemas para el diseño de la base de datos MongoDB.
- Creación de una ontología específica para los datos que no han podido ser etiquetados.
- Mejora y optimización del etiquetado semántico mediante la herramienta *Open Semantic Framework* (OSF). OSF no se ha utilizado en est proyecto debido a incompatibilidades con el entorno de pruebas, en concreto, para su instalación es necesario disponer de una máquina con sistema operativo *Ubuntu* [1].

[1] http://www.ubuntu.com/

Bibliografía

[1] Herramienta de extracción de datos RDF. http://rdfa.info/, .

[2] Herramienta de prueba de datos estructurados de Google. http://www.google.com/webmasters/tools/richsnippets, .

[3] Página oficial de CouchDB. http://couchdb.apache.org/, .

[4] Página oficial de flowli. http://flow.li/, .

[5] Página oficial de Mongopress. http://www.mongopress.org/, .

[6] Página oficial de RubedoCMS. http://www.rubedo-project.org/, .

[7] Página oficial de webnodes. http://www.webnodes.com/, .

[8] Portal de OpenData Navarra. http://www.gobiernoabierto.navarra.es/es/open-data, .

[9] RDF content description. http://purl.org/rss/1.0/modules/content/.

[10] Web de la Cátedra del Patrimonio Inmaterial de Navarra. http://www.unavarra.es/centrosydepartamentos/catedras/catedra-de-patrimonio-inmaterial-de-navarra, .

[11] Web de Wolfram Alpha. http://www.wolframalpha.com/, .

[12] Web del Archivo del Patrimonio Inmaterial de Navarra. http://www.navarchivo.com, .

[13] Web del Plan Nacional de Patrimonio Inmaterial. http://ipce.mcu.es/conservacion/planesnacionales/inmaterial.html, .

[14] Web del servicio CartoDB. http://cartodb.com/, .

[15] Sioc core ontology specification. r/types#, 2010.

[16] Dublin Core vocabulary tags for web resources. 2012. URL http://purl.org/dc/terms/.

[17] Xmlschema. http://www.w3.org/2001/XMLSchema#, 2014.

[18] T. Berners-Lee. Linked data. 2006. URL http://www.w3.org/DesignIssues/LinkedData.html.

[19] Yahoo! Bing, Google. Schema.org. 2001. URL http://schema.org/.

[20] Uldis Bojars. ResumeRDF Ontology Specification. 2007. URL http://rdfs.org/resume-rdf/.

[21] Dan Brickley. Basic Geo (WGS84 lat/long) Vocabulary. 2003. URL http://www.w3.org/2003/01/geo/.

[22] Dave Clark. Content management and the separation of presentation and content. *Technical communication quarterly*, 17(1):35–60, 2007.

[23] Library Of Congress. MADS RDF Primer. 2012. URL http://www.loc.gov/mads/rdf/.

[24] Libby Mille Dan Brickley. FOAF Vocabulary Specification. 2014. URL http://xmlns.com/foaf/0.1/.

[25] Eric Prud'hommeaux Gavin Carothers David Beckett, Tim Berners-Lee. Rdf 1.1 turtle. Febrero 2014. URL http://www.w3.org/TR/turtle/.

[26] Stefan Decker Diego Berrueta, Dan Brickley. Sioc core ontology specification. 2010. URL http://rdfs.org/sioc/spec/.

[27] Facebook. The Open Graph protocol. 2014. URL http://ogp.me/.

[28] Mathias Humbert. Technology and workforce: Comparison between the information revolution and the industrial revolution. Inf. téc., 2007.

[29] MongoDB Inc. Top 5 Considerations When Evaluating NoSQL Databases. 2015.

[30] Manu Sporny Mark Birbeck Ivan Herman, Ben Adida. Rdfa 1.1 primer - second edition. Agosto 2013. URL http://www.w3.org/TR/xhtml-rdfa-primer/.

[31] Bobbie Johnson. British search engine 'could rival google'. 2009. URL http://www.theguardian.com/technology/2009/mar/09/search-engine-google.

[32] Kevin T. Smith Michael C. Daconta, Leo J. Obrst. *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley, 2003.

[33] Markus Lanthaler Richard Cyganiak, David Wood. Rdf 1.1 concepts and abstract syntax. 2014. URL http://www.w3.org/TR/rdf11-concepts/.

[34] Jay Rossiter. Progress report: Continued product focus. Septiembre 2014. URL http://yahoo.tumblr.com/post/98474044364/progress-report-continued-product-focus.

[35] O. Lassila T. Berners-Lee, J. Hendler. Rdbms to nosql: Reviewing some next-generation non-relational database's. *Scientific American*, 2001.

[36] O. Lassila T. Berners-Lee, J. Hendler. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 2001.

[37] Dan Connolly Tim Berners-Lee. Notation3 (n3): A readable rdf syntax. Marzo 2011. URL http://www.w3.org/TeamSubmission/n3/.

[38] W3C. Skos simple knowledge organization system. URL http://www.w3.org/2004/02/skos/.

[39] W3C. Web ontology language. 2007. URL http://www.w3.org/owl.

[40] W3C. Rdf schema 1.1. 2014. URL http://www.w3.org/TR/rdf-schema/.

[41] W3Techs.com. Usage of content management systems for websites. Febrero 2015. URL http://w3techs.com/technologies/overview/content_management/all/.