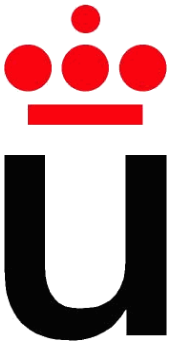

UNIVERSIDAD REY JUAN CARLOS

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2012/2013

Proyecto de Fin de Carrera

Aplicación para la integración de datos cinematográficos enlazados

Autor: Ana María Mateo González
Tutor: Alberto Fernández Gil

Agradecimientos

A mis padres y mis hermanas, que me han apoyado incondicionalmente durante toda la carrera.

A mi tutor Alberto por su comprensión y paciencia y por haber conseguido que encuentre la motivación para terminar.

A Jose, por todo.

Resumen

En los últimos años la rápida evolución de la Web ha cambiado nuestra forma de ver y realizar todo tipo de tareas, ya sean relacionadas con nuestra vida privada como con nuestro trabajo diario y se ha convertido en la principal fuente de conocimiento al que se recurre para realizar búsquedas de cualquier tipo de información.

Sin embargo, esta evolución implica un cambio constante tanto en la representación de la información como en el acceso a la misma, por lo que sería deseable que las nuevas aplicaciones que vayan surgiendo en los próximos años estén basadas en tecnologías capaces de adaptarse a este cambio continuo.

Este Proyecto de Fin de Carrera aborda el extendido problema de la heterogeneidad de los datos contenidos en la web actual carente de semántica que, en su mayoría, es una red caótica de información incoherente. Se trata de establecer una organización para esos datos aplicando la Inteligencia Artificial a las tecnologías web para optimizar el acceso, la combinación y la distribución de la información describiendo de manera explícita los recursos existentes en la Web.

Por este motivo se propone la creación de una aplicación web con tecnologías semánticas que trata de reutilizar varias fuentes de datos publicados en diferentes puntos de acceso, extraer información relevante y combinar esa información de forma que la representación resultante sea coherente. Esta información susceptible de ser recombinada estará compuesta por datos relacionados con el mundo del cine.

En resumen, se propone una solución mediante la que se pretende demostrar la flexibilidad y versatilidad que supone el uso de las tecnologías semánticas para la construcción de nuevos contenidos web creados combinando información preexistente.

Índice general

1. Introducción 15
1.1 Presentación del problema 15
1.2 Objetivos 17
1.3 Método de trabajo 18
2. Estado del arte 19
2.1 De la web de documentos a la web semántica 19
2.1.1 Posicionamiento Web (SEO) 19
2.2.2 Microformatos 20
2.2.3 APIs Web 20
2.2 La Web Semántica 21
2.2.1 Estructura de la Web Semántica 22
2.2.2 RDF 24
2.2.3 SPARQL 26
2.2.4 Plataformas de desarrollo 27
2.3 La Web de datos 28
3. Fuentes de datos 31
3.1 Cómo obtener información de las fuentes de datos 31
3.2 Linked Movie DataBase 33
3.3 DBpedia 38
3.4 Revyu 41

3.5 Sindice 44
3.6 Flickr wrappr 44
3.7 Freebase 46
3.8 Enlazando las fuentes de datos 48
4. Descripción informática 49
4.1 Especificación de requisitos: 49
4.2 Análisis 51
4.3 Diseño 56
4.3.1 Diseño de la interfaz gráfica 56
4.3.2 Diseño de la base de datos 59
4.3.3 Arquitectura del software 59
4.4 Implementación 64
4.4.1 Herramientas utilizadas 64
4.4.2 Decisiones de implementación 65
4.4.3 Modelo de implementación 67
4.5 Pruebas 69
4.5.1 Pruebas funcionales de acceso a la aplicación 69
4.5.2 Pruebas funcionales de búsqueda de películas 70
4.5.3 Pruebas no funcionales 71
5. Conclusiones 73
5.1 Logros principales 73
5.2 Problemas principales encontrados 73
5.3 Líneas futuras 74
6. Bibliografía 75

Apéndice I 81
Manual de instalación y despliegue de la aplicación web 81
Instalación del software necesario 81
Despliegue de la aplicación 85

Índice de figuras

Figura 1: Recuperación de información 16

Figura 2: Estructura de la web semántica (2006) 22

Figura 3: Tripla RDF 24

Figura 4: Ejemplo de grafo RDF 25

Figura 5: Grafo de ejemplo SPARQL 26

Figura 6: Grafo de datos enlazados 30

Figura 7: Gráfico de disponibilidad del punto de acceso SPARQL de LMDB 33

Figura 8: Grafo de enlaces de Linked Movie DataBase 34

Figura 9: Estructura de la consulta a LMDB 37

Figura 10: Arquitectura de DBpedia 40

Figura 11: Estructura de la consulta a DBpedia 41

Figura 12: Fuentes de datos de Revyu 42

Figura 13: Arquitectura de Revyu 43

Figura 14: Estructura de la consulta a Revyu 44

Figura 15: Estructura de la consulta a flickr wrappr 46

Figura 16: Estructura de la consulta a Freebase 48

Figura 17: Grafo de enlazado de las fuentes de datos 48

Figura 18: Diagrama de casos de uso 52

Figura 19: Diagrama de clases de análisis 52

Figura 20: Interfaz gráfica de la aplicación web al realizar una primera búsqueda 56

Figura 21: Distribución de los resultado de la búsqueda según la fuente de datos 57

Figura 22: Distribución de los resultado de la búsqueda según la fuente de datos 58

Figura 23: Modelo Vista Controlador 60

Figura 24: Diagrama de clases y paquetes 62

Figura 25: Diagrama de secuencia de la búsqueda de una película 63

Figura 26: Diagrama de despliegue 68

Figura 27: Diagrama de comunicación con los puntos de acceso 69

Figura 28:Variables de entorno del servidor 82

Figura 29: Configuración de la base de datos 1 83

Figura 30: Configuración de la base de datos 2 83

Figura 31: Configuración de la base de datos 3 84

Figura 32: Importar esquema de la base de datos 85

Índice de tablas

Tabla 1: Estadísticas de Linked Movie DataBase 35

Tabla 2: Número de enlaces por propiedad en Linked Movie DataBase 36

Tabla 3: Información de DBpedia en the Data Hub 38

Tabla 4: Información de Revyu en the Data Hub 41

Tabla 5: Información de flickr wrappr en the Data Hub 45

Tabla 6: Información de Freebase en the Data Hub 46

Tabla 7: Requisitos funcionales de la aplicación web 49

Tabla 8: Requisitos no funcionales de la aplicación web 50

Tabla 9: Requisitos de implementación de la aplicación web 51

Tabla 10: Esquema de la tabla Usuario 59

Tabla 11: Pruebas funcionales de acceso a la aplicación 69

Tabla 12: Pruebas funcionales de búsqueda de películas 70

Tabla 13: Pruebas no funcionales 71

1. Introducción

1.1 Presentación del problema

A medida que pasa el tiempo el perfil de los usuarios que utilizan aplicaciones web va siendo cada vez más especializado y por tanto cada vez se busca información más precisa y aplicaciones más concretas y especializadas. Esta búsqueda de aplicaciones concretas y adecuadas a nuestras necesidades está dando fuerza al desarrollo de la web semántica, puesto que permite una mayor eficiencia en la circulación de la información.

Es importante destacar que la falta de precisión en la recuperación de información es un hecho habitual que puede hacer perder tiempo al usuario, incluso aunque se tenga una idea clara de la información que se necesita y dónde se puede encontrar dicha información. De esta manera se puede comprobar que los problemas más susceptibles de producirse en el uso de la web actual se centran en:

- la recuperación de documentos relevantes
- la extracción de datos relevantes de dichos documentos
- la combinación de la información a partir de distintas fuentes

Por estos motivos se plantea el desarrollo de una aplicación adecuada a las necesidades de un público específico, como son los aficionados al mundo del cine.

En Internet se pueden encontrar muchas páginas web relacionadas con el cine que aglutinan mucha información útil para el usuario como pueden ser: las opiniones sobre las películas (ya sea por expertos o por simples aficionados), música relacionada, fotografías, etc. Sin embargo, si un usuario observa mejor calidad en la información sobre cine proporcionada por un determinado sitio web (que contenga por ejemplo un elenco de actores más amplio), elige otro distinto para recuperar los datos sobre la música (con otras bandas sonoras relacionadas) de una determinada película y en otro observa que las opiniones sobre dichas películas son más veraces, no

tendrá más remedio que recopilar la información por sí mismo accediendo a los tres sitios web de manera aislada.

Otro problema relacionado con las búsquedas específicas de información tiene que ver con la representación de información. La combinación de información realizada por el usuario anteriormente descrito puede convertirse en una tarea tediosa y a veces difícil si la presentación de la información no es especialmente intuitiva. Por ello, con este proyecto se pretende representar el conjunto de datos requeridos de manera homogénea.

Se ha tratado de buscar una solución existente a este problema realizando dos tipos de búsquedas en la fase previa del proyecto:

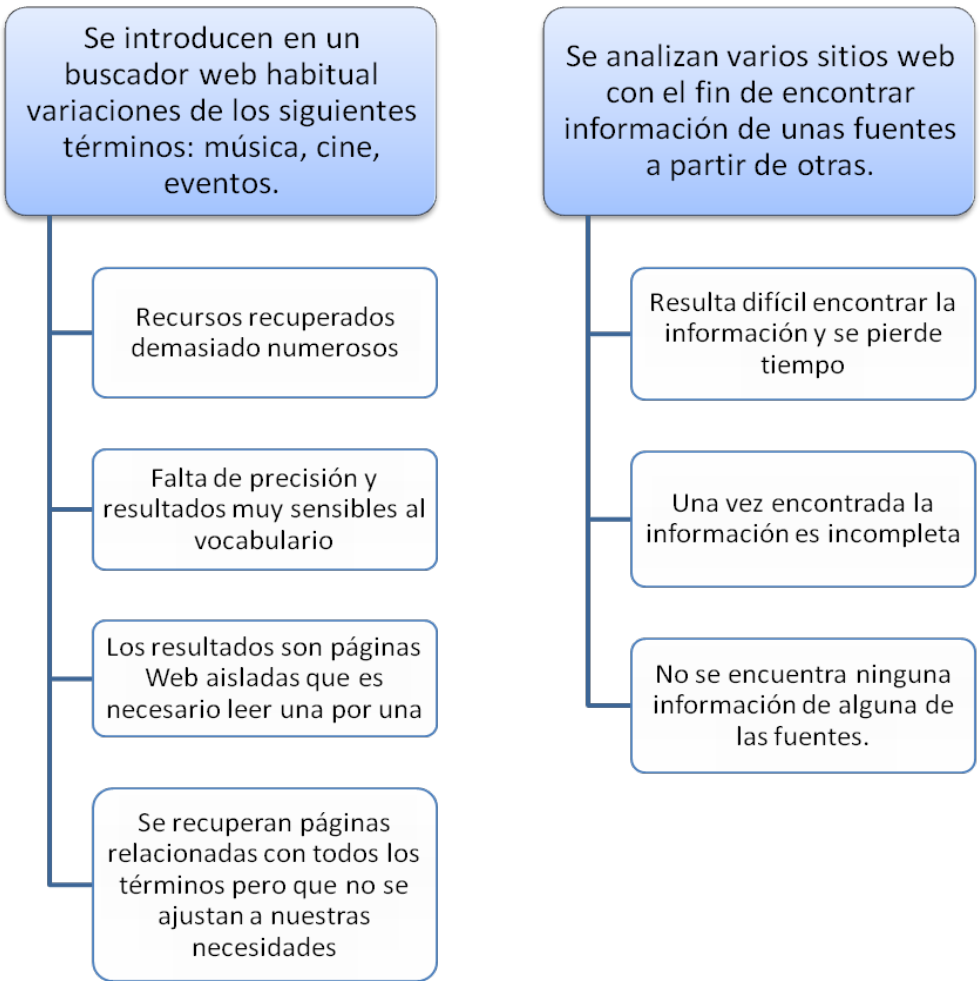

Figura 1: Recuperación de información

Dados los pobres resultados obtenidos con estos dos tipos de búsquedas se decide que la realización del proyecto puede resultar interesante, necesaria y útil para la evolución global de la web semántica, puesto que es necesaria tanto la publicación de datos semánticos como la aparición de consumidores de dichos datos. Estos consumidores deberían tener las siguientes características:

- Utilización de múltiples fuentes de datos independientes y heterogéneos pudiendo utilizar un modelo de representación común para integrarlos.
- Reutilización de datos originales en contextos no siempre esperados por los proveedores.
- Bajo coste de agregación gracias a tecnologías como RDF que facilita la integración de sistemas de forma global.

En definitiva, este proyecto surge de esta necesidad de componer e integrar varias fuentes de datos concretas de libre acceso mediante enlaces que relacionen estas fuentes.

1.2 Objetivos

Lo objetivos principales de este proyecto, teniendo en cuenta los problemas comentados anteriormente, son:

- ✓ Analizar la evolución de la web tendente hacia la web semántica.
- ✓ Analizar la situación actual y las tecnologías de la web semántica, incluyendo problemas y ventajas.
- ✓ Desarrollar una aplicación web capaz de integrar datos de sistemas heterogéneos de manera interoperable.
- ✓ Crear un valor agregado de datos coherentes, unificando contenidos mediante el estudio de las fuentes de datos.
- ✓ Desarrollar *software* escalable y que facilite el posterior mantenimiento con una interfaz gráfica usable e intuitiva.

1.3 Método de trabajo

A continuación se define la metodología a seguir durante la realización del proyecto para garantizar la coherencia y la estructura del mismo.

En lo relativo al modelo de desarrollo *software* se hará uso de *RUP* (*Rational Unified Process*, Proceso Racional Unificado) por lo que la metodología será iterativa e incremental y utilizará *UML* (*Unified Modeling Language*, Lenguaje Unificado de Modelado) para la documentación de la arquitectura del sistema, el cual puede dividirse en las siguientes fases de trabajo:

Estudio previo: Se recopilará información sobre las tecnologías más utilizadas en la actualidad en el ámbito de la web semántica así como información relativa a la interacción persona-computador para facilitar la usabilidad de la aplicación.

Fase de especificación de requisitos: Se determinarán las principales funcionalidades de la aplicación.

Fase de análisis: Se analizarán las fuentes de datos de las que se va a obtener la información necesaria, así como el modo de acceso a esos datos.

Fase de diseño: Se describirán las diferentes clases que van a ser necesarias y se utilizará un patrón arquitectural en tres capas que permite el desacoplamiento del modelo de datos y la lógica de negocio de la interfaz de usuario facilitando de esa manera tanto su desarrollo como su posterior mantenimiento.

Fase de implementación: Se implementará la aplicación web utilizando tecnologías de la web semántica que se describirán más adelante.

Fase de pruebas: Finalmente, se realizarán las pruebas necesarias para la validación de la aplicación, su adecuación a los requisitos propuestos y la verificación de su calidad con respecto a las necesidades y preferencias de los usuarios finales utilizando un conjunto de datos lo suficientemente representativo para legitimar la muestra de datos.

2. Estado del arte

2.1 De la web de documentos a la web semántica

Como se ha comentado anteriormente la web de documentos posee muchos inconvenientes a la hora de recuperar datos relevantes y/o reutilizar estos datos. Esto es debido a que el lenguaje en el que se basan las páginas de la web de documentos, HTML, está orientado al contenido textual en lugar de al contenido estructurado/semántico.

Un factor clave de la evolución de la web de documentos a la web actual es la preocupación de la reutilización de los datos existentes en la web. Para permitir dicha reutilización es necesario añadir datos estructurados a los documentos para permitir a los consumidores de datos utilizar herramientas que procesen estos datos estructurados. Una gran cantidad de soluciones están siendo utilizadas con bastante éxito a la hora de organizar el contenido caótico de la web de documentos. Algunas de estas soluciones se detallan a continuación:

2.1.1 Posicionamiento Web (SEO)

El posicionamiento Web es el conjunto de tareas dedicadas a la mejora de la visibilidad de un sitio web en los resultados de los diferentes buscadores. Empezó a utilizarse en 1990 cuando los motores de búsqueda comenzado a indexar y catalogar Internet.

Al principio estas tareas simplemente se basaban en indexar y extraer información de las páginas web, como las palabras que contenían, dónde estaban localizadas y su relevancia dentro del documento, además de todos los vínculos. Esta información sobre las palabras era provista por los administradores web en forma de metatags[1], lo cual provocó que las pocas restricciones en la utilización de estas

[1] http://es.wikipedia.org/wiki/Metatag

metatags, fueran utilizadas por los administradores web para obtener un buen posicionamiento incluso aunque las búsquedas no estuvieran ni siquiera relacionadas con el contenido de las páginas.

Más tarde surgieron soluciones a este problema, como algoritmos más complejos para la extracción de metatags como el PageRank[2] que también fueron manipulados por los administradores de páginas web.

En la historia del posicionamiento web siempre ha habido un conflicto de intereses entre el interés de los buscadores por ofrecer contenido relevante y el interés de los administradores por posicionarse, que hasta la fecha no ha sido realmente solucionado.

2.2.2 Microformatos

Son una tecnología similar a los datos enlazados, el objetivo común de ambos es extender la Web actual mediante datos estructurados. Los microformatos definen una serie de formatos de datos que se encuentran embebidos dentro de los documentos HTML por medio de atributos de clase. El punto débil de los microformatos es que se restringe la representación de los datos a un pequeño conjunto de entidades y atributos de estas entidades. Por este motivo los microformatos no son la mejor manera de compartir datos en la Web.

2.2.3 APIs Web

Las APIs Web representan otra solución para el acceso a datos estructurados en la Web actual. Muchas de las mayores fuentes de datos como Amazon[3], eBay[4], Yahoo![5] y Google[6] proporcionan acceso a sus datos vía APIs Web haciendo uso del protocolo HTTP.

El número de aplicaciones especializadas que combinan datos de diferentes fuentes ha crecido increíblemente ya que ofrece beneficios indudables a los

[2] http://es.wikipedia.org/wiki/PageRank
[3] http://www.amazon.es/
[4] http://www.ebay.es/
[5] http://es.yahoo.com/
[6] https://www.google.es/

programadores. Sin embargo, el problema principal de estas APIs es que sus datos pueden ser accedidos de maneras muy heterogéneas y los datos obtenidos pueden tener múltiples formatos dependiendo de la fuente de datos que se utilice, lo cual supone un gran esfuerzo a la hora de integrar cada nuevo conjunto de datos en la aplicación que se esté desarrollando.

2.2 La Web Semántica

"La Web Semántica no es una web separada sino una extensión de la web actual, en la que la información es ofrecida con un significado bien definido, facilitando el trabajo cooperativo de computadoras y personas." Tim Berners-Lee, 2001

La Web Semántica nace como resultado al problema de la falta de descripción de contenido en la información, falta de significado y relación entre los datos existentes en la Web. Se pretende crear una base de conocimiento común que proporcione a los agentes inteligentes un modo legible para poder procesar esta información.

Con la web semántica se evitan los problemas derivados de la web actual en los que la recuperación y selección de información relevante no es en muchos casos tarea fácil, ya sea porque la cantidad de resultados obtenidos en una búsqueda es demasiado grande o por que los resultados son demasiado sensibles al vocabulario utilizado. Algunas formas de organización del conocimiento planteadas para la web semántica son:

- Taxonomías: modelo de organización jerárquico.
- Tesauros: catálogo de términos agrupados por un mismo significado.
- Ontología: Como señala Tom Gruber *"una ontología es una especificación de una conceptualización, es decir, una representación conceptual compartida que proporciona una comprensión común de un dominio".* Así se define un vocabulario de clases y atributos que proporciona el modo de compartir conocimiento del mismo dominio de forma consensuada entre varias entidades.

2.2.1 Estructura de la Web Semántica

A continuación se muestra un esquema propuesto en el año 2006 que describe la estructura en capas de la web semántica desde el punto de vista de las tecnologías que se han desarrollado:

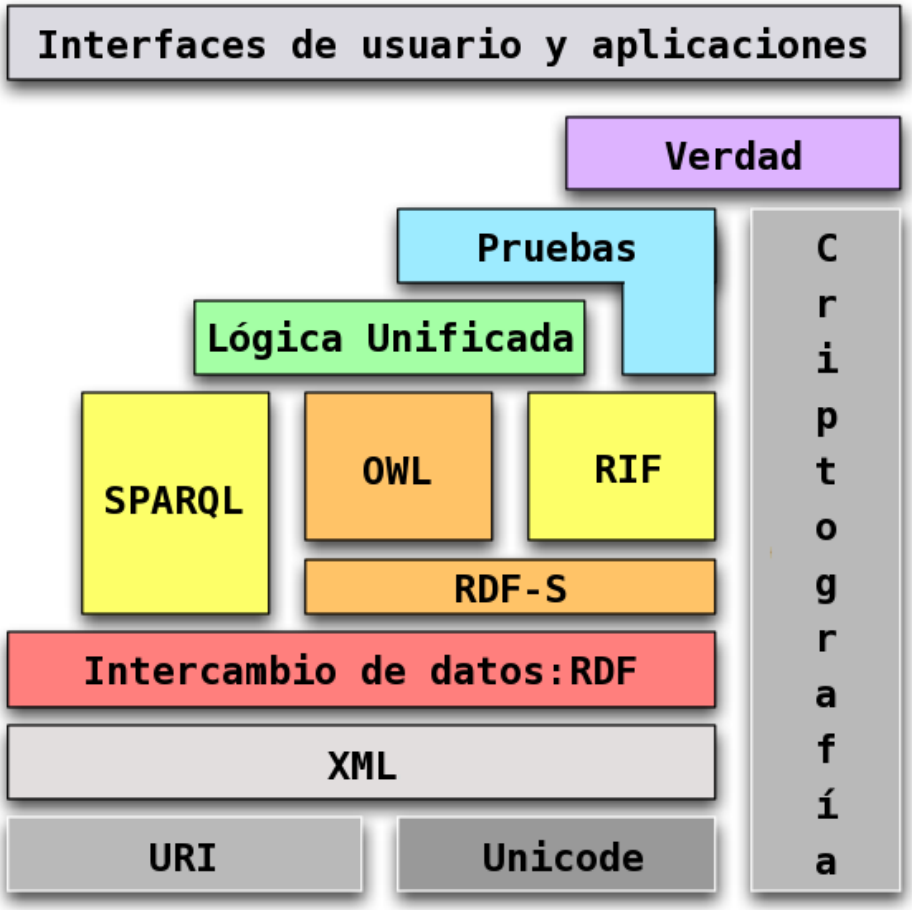

Figura 2: Estructura de la web semántica (2006)

- **Unicode[7]:** estándar de representación y codificación de caracteres en cualquier lenguaje.
- **URI[8]:** (Uniform Resource Identifier) identifica de forma unívoca un recurso publicado en Internet mediante URL (descripción de la ubicación) + URN (descripción del espacio de nombres).

[7] http://es.wikipedia.org/wiki/Unicode
[8] http://es.wikipedia.org/wiki/Uniform_Resource_Identifier

- **XML[9]:** (eXtensible Markup Language) formato de marcado común y no propietario que permite definir estructuras de datos y documentos en forma de árboles de etiquetas con atributos pero sin restricciones de significado. XMLSchema[10] establece la estructura del documento XML, manejando datos primitivos y derivados.
- **RDF[11]:** (Resource Description Framework) es una especificación del W3C diseñada como modelo de metadatos y actualmente es el estándar más popular y extendido en la comunidad de la web semántica para describir recursos.
- **RDFShema[12]:** es una forma de definir ontologías en RDF mediante jerarquías de clases de recursos. RDFS permite definir vocabularios de términos y relaciones entre esos términos, así como restricciones de tipos de datos para sujetos y objetos.
- **OWL[13]:** (Web Ontology language) es la recomendación oficial del W3C para definir y publicar ontologías en la web. Es un lenguaje derivado de DAML (DARPA Agent Markup Language) y OIL (Ontology Inference Layer) que incluye la capacidad expresiva de RDFS y la extiende con la posibilidad de utilizar expresiones lógicas y otras propiedades de las relaciones como la cardinalidad, simetría, transitividad, etc. Existen tres tipos de OWl dependiendo de los requerimientos de expresividad: OWL Full, OWL DL y OWL Lite.
- **SPARQL[14]:** (Simple Protocol and RDF Query Language) es la recomendación oficial del W3C como lenguaje de consultas y protocolo de acceso a datos sobre la base de conocimiento en OWL/RDF.
- **Reglas RIF[15]:** Infraestructura que permite la definición de reglas de forma interoperable entre distintos sistemas.
- **Lógica:** en esta capa se encuentran los mecanismos de inferencia sobre los datos, que le darían un valor agregado permitiendo crear nuevo conocimiento.

[9] http://www.w3.org/XML/
[10] http://www.w3schools.com/schema/
[11] http://www.w3.org/RDF/
[12] http://www.w3.org/TR/rdf-schema/
[13] http://www.w3.org/TR/owl-features/
[14] http://www.w3.org/TR/sparql11-overview/
[15] http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/

- **Pruebas / Verdad:** capas mediante las que sería posible validar los datos inferidos de una recomendación o conclusión y así obtener un buen nivel de confianza sobre la calidad de los datos obtenidos.

2.2.2 RDF

RDF es un lenguaje de descripción universal que permite a los usuarios describir recursos mediante el uso de sus propios vocabularios. Se basa en un modelo de tripletas del tipo (sujeto, predicado, objeto) en el que el sujeto y el objeto representan los recursos y el predicado la relación entre los mismos.

Los recursos (sujeto y objeto) son las *'cosas'* que se quieren describir en el documento RDF y cada uno de ellos debe tener una URI asociada que constituirá un identificador único aunque también pueden ser literales. Las propiedades (predicado) describen las relaciones entre los recursos y el conjunto formado por los tres nodos forma una tripla o sentencia RDF.

Esta relación se puede representar en forma de grafo dirigido con nodos y arcos etiquetados:

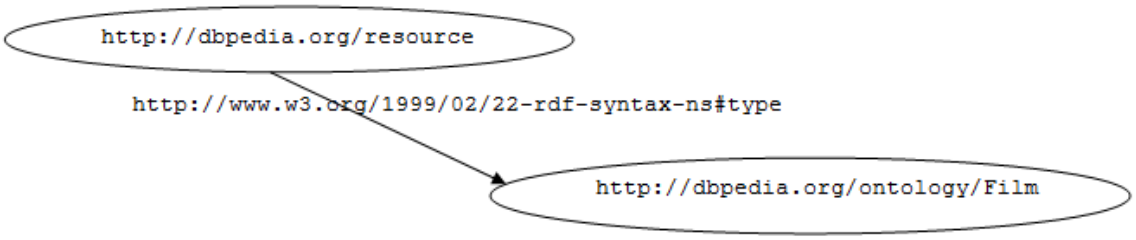

Figura 3: Tripla RDF

Este modelo en grafo facilita la integración de datos heterogéneos de varios grafos distintos, así como la representación de datos tabulares de forma sencilla. La Web de Datos sería una red semántica formada por el grafo resultante de la unión de todos los grafos publicados en la web.

Existen varias formas para la formulación escrita de RDF (XML, N3, RDFa, Turtle, etc.) siendo la más extendida la basada en XML. A continuación se muestra un ejemplo de la estructura de un documento simple RDF/XML extraído de Revyu:

```
<rdf:RDF
   xml:base="http://revyu.com/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:dcterms="http://purl.org/dc/terms/"
   xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:rev="http://purl.org/stuff/rev#"
   xmlns:tag="http://www.holygoat.co.uk/owl/redwood/0.1/tags/">

<rdf:Description rdf:about="things/forrest-gump-1254607504">
   <rev:hasReview
 rdf:resource="reviews/cbe1fd43cf7de69ee0530fe65593d6d77d03daed"/>
   <rev:hasReview
 rdf:resource="reviews/436f699d347d433315507923664cf567fe872a59"/>
   <tag:tag
 rdf:resource="taggings/cbe1fd43cf7de69ee0530fe65593d6d77d03daed"/>
   <tag:tag
 rdf:resource="taggings/436f699d347d433315507923664cf567fe872a59"/>
</rdf:Description>

<owl:Thing rdf:about="things/forrest-gump-1254607504">
   <rdfs:label>Forrest Gump</rdfs:label>
</owl:Thing>
</rdf:RDF>
```

Como se puede ver en este ejemplo al principio del documento RDF se define un espacio de nombres que será utilizado en el resto del documento para facilitar la lectura. El atributo rdf:about del elemento rdf:Description es básicamente la definición de un identificador por lo que una sentencia del ejemplo anterior sería:

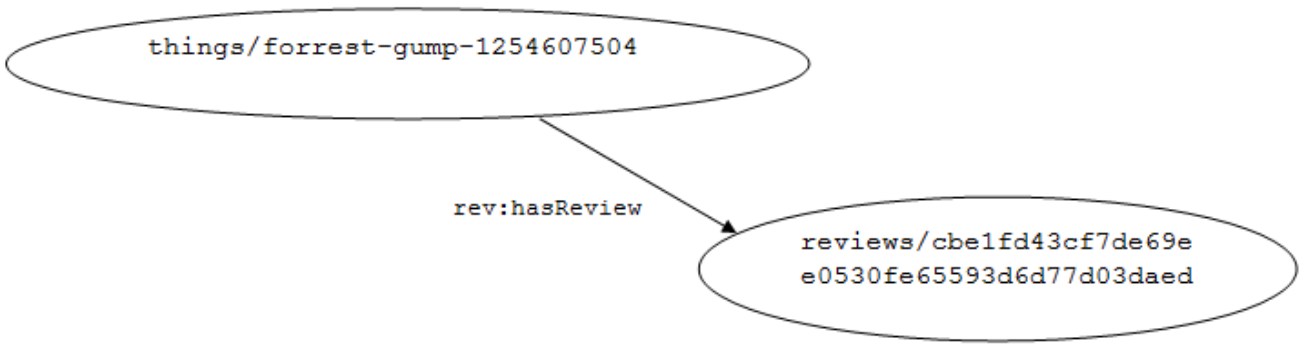

Figura 4: Ejemplo de grafo RDF

2.2.3 SPARQL

SPARQL Es el estándar de consulta de bases de conocimiento descritas en RDF. Es un lenguaje parecido a SQL que posee además funciones avanzadas para la creación de consultas complejas como pueden ser OPTIONA, UNION Y FILTER.

Fundamentalmente se basa en consultas que especifican un patrón y tratan de hacer coincidir este patrón dentro del modelo de datos RDF sobre el que se realiza la consulta. SPARQL también permite definir prefijos para los espacios de nombres utilizados en la consulta, haciéndola más legible. A continuación se muestra un ejemplo de consulta básica relacionada con el ejemplo descrito para RDF.

```
PREFIX rev: <http://purl.org/stuff/rev#>
SELECT DISTINCT ?text ?rating
       WHERE {
       "things/forrest-gump-1254607504" rev:hasReview ?review .
             OPTIONAL{ ?review rev:rating ?rating .}
             OPTIONAL{ ?review rev:text ?text .}
       }
```

Mediante esta consulta recuperaríamos el texto y la puntuación de las críticas escritas para la película Forrest Gump (things/forrest-gump-1254607504). El grafo que esta consulta buscaría dentro del modelo de datos sería el siguiente:

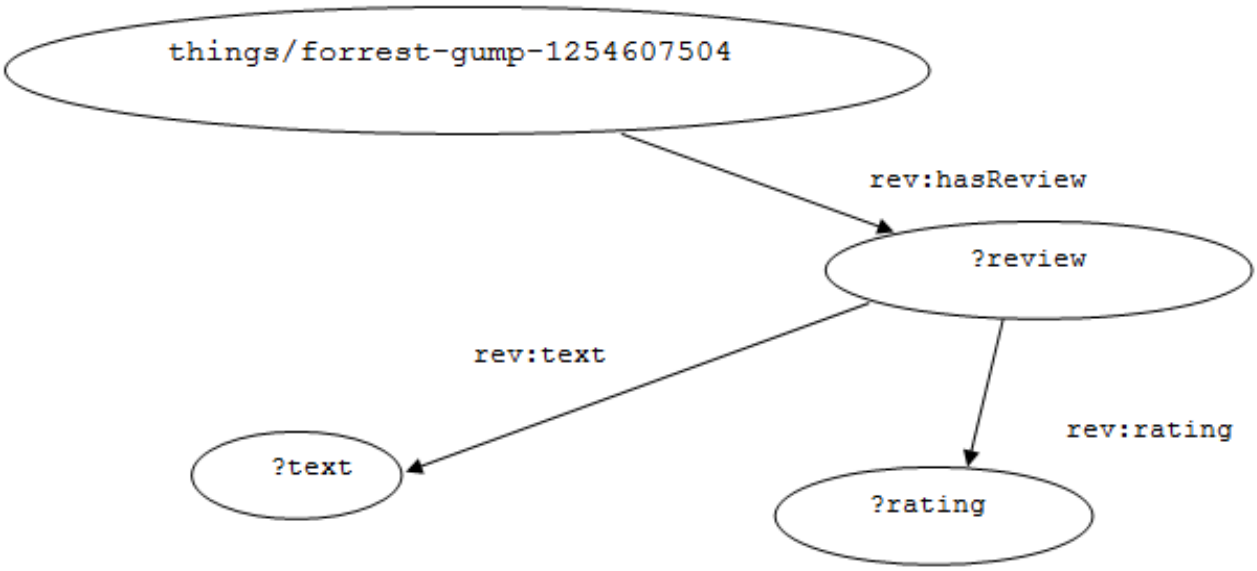

Figura 5: Grafo de ejemplo SPARQL

2.2.4 Plataformas de desarrollo

Como hemos visto anteriormente la web semántica es un esfuerzo por dar significado a la Web de Documentos, por lo que se hace necesario el uso de herramientas que sean capaces de procesar los daros estructurados. A continuación se describen dos de los marcos de desarrollo más importantes que ofrecen la posibilidad de explotar el potencial de la Web Semántica.

Jena[16]

Plataforma que proporciona una colección de herramientas y librerías Java para el desarrollo de aplicaciones web semánticas y de datos enlazados.

- Permite leer, procesar y escribir RDF en XML, N-triples y Turtle.
- API para la creación de ontologías en OWL y RDFS, así como razonadores sobre estas.
- Permite guardar RDF textual y en formato de base de datos
- Motor de consultas SPARQL.
- Servidores para la publicación de datos en RDF

Sesame[17]

Desarrollado por el proyecto europeo Ontoknowledge, Sesame es un marco de trabajo que proporciona todo tipo de herramientas para consultar, analizar, inferir y guardar datos RDF.

Sesame ofrece soporte para consultas SPARQL y SeRQL[18] así como acceso a repositorios remotos de RDF que utilizan la misma API que para el acceso local. También soporta los tipos principales de formatos de RDF incluyendo RDF/XML, Turtle, N-Triples, TriG y Trix.

[16] http://jena.apache.org/
[17] http://www.openrdf.org/doc/sesame/users/userguide.html
[18] http://es.wikipedia.org/wiki/SeRQL

Posee una API de nombre Alibaba que permite generare clases Java a partir de ontologías y a la inversa. Esto posibilita el uso de ontologías específicas como RSS[19], FOAF[20] y Dublin Core[21] directamente desde el código Java.

2.3 La Web de datos

Como se ha comentado anteriormente, la web tradicional contiene datos publicados en muy diversos formatos que no permiten establecer relaciones entre unos documentos y otros de manera sencilla. En los últimos años la web ha comenzado su evolución hacia un espacio global de documentos enlazados denominado *Linked Data* y que cuenta además con una serie de 'buenas prácticas' para la publicación y el enlazado de los datos estructurados.

Según Tim Berners-Lee (creador de la Web) hay cuatro principios que caracterizan los datos vinculados:

1. Utilizar URIs para identificar los recursos publicados en la Web
2. Aprovechar el HTTP de la URI para que la gente pueda localizar y consultar estos recursos.
3. Proporcionar información útil acerca del recurso.
4. Incluir enlaces a otras URI relacionadas con los datos contenidos en el recurso, de forma que se potencie el descubrimiento de información en la Web.

El *Consorcio World Wide Web*[22] (W3C), comenzó en 2007 un nuevo proyecto llamado "*Linking Open Data"* cuyo objetivo es crear y publicar bases de datos en la Web en formato RDF así como establecer relaciones entre esas fuentes de datos de la misma forma que lo hacen los hiperenlaces en la web como la conocemos hasta el momento. Así la información puede ser accedida desde un navegador siguiendo enlaces entre documentos RDF en lugar de páginas en HTML.

[19] http://es.wikipedia.org/wiki/RSS
[20] http://www.foaf-project.org/
[21] http://dublincore.org/
[22] http://www.w3.org/

Si los datos en la web se encuentran enlazados, es decir, se ha establecido una serie de relaciones lógicas entre diferentes conceptos, se hace posible llegar a información referenciada partiendo de otros datos iniciales.

Esta Web centrada en los enlaces entre datos permite la aparición de nuevos tipos de aplicaciones como son los navegadores de *Linked Data* entre los que se encuentra Tabulator[23], Disco[24], Zitgist, Openlink[25].

Además de publicar y enlazar contenido, el proyecto *Linked Data* también incluye el desarrollo de motores de búsqueda para la Web de datos, así como rastreadores que utilizan los datos vinculados como son son Falcons, Sindice, Swoogle y Watson.

Un aspecto importante sobre los datos enlazados es la necesidad de mantenimiento de los enlaces, puesto que constantemente se añaden nuevas entidades y datos desactualizados son eliminados lo cual puede llevar a la existencia de demasiados enlaces no válidos. En esto se están centrando los esfuerzos actuales de la comunidad de *Linked Data* que ha creado herramientas como *Ping the Semantic Web* que mantiene un índice de los datos cambiados o nuevos.

Otros valores a tener en cuenta en el desarrollo de los datos enlazados son la confianza, calidad y relevancia de los contenidos puestos a disposición de los usuarios, por lo que serán necesarios algoritmos del tipo de *PageRank* adaptados a los patrones de enlazado de la Web de Datos.

En la Figura 6 se muestra el gráfico del estado de la nube de Linked Data en el año 2010. Actualmente aunque siguen creándose nuevas fuentes de datos, han dejado de añadirse a este grafo por el tamaño que este ha adquirido:

[23] http://www.w3.org/2005/ajar/tab
[24] http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/disco/
[25] http://www.openlinksw.com/rdfbrowser/

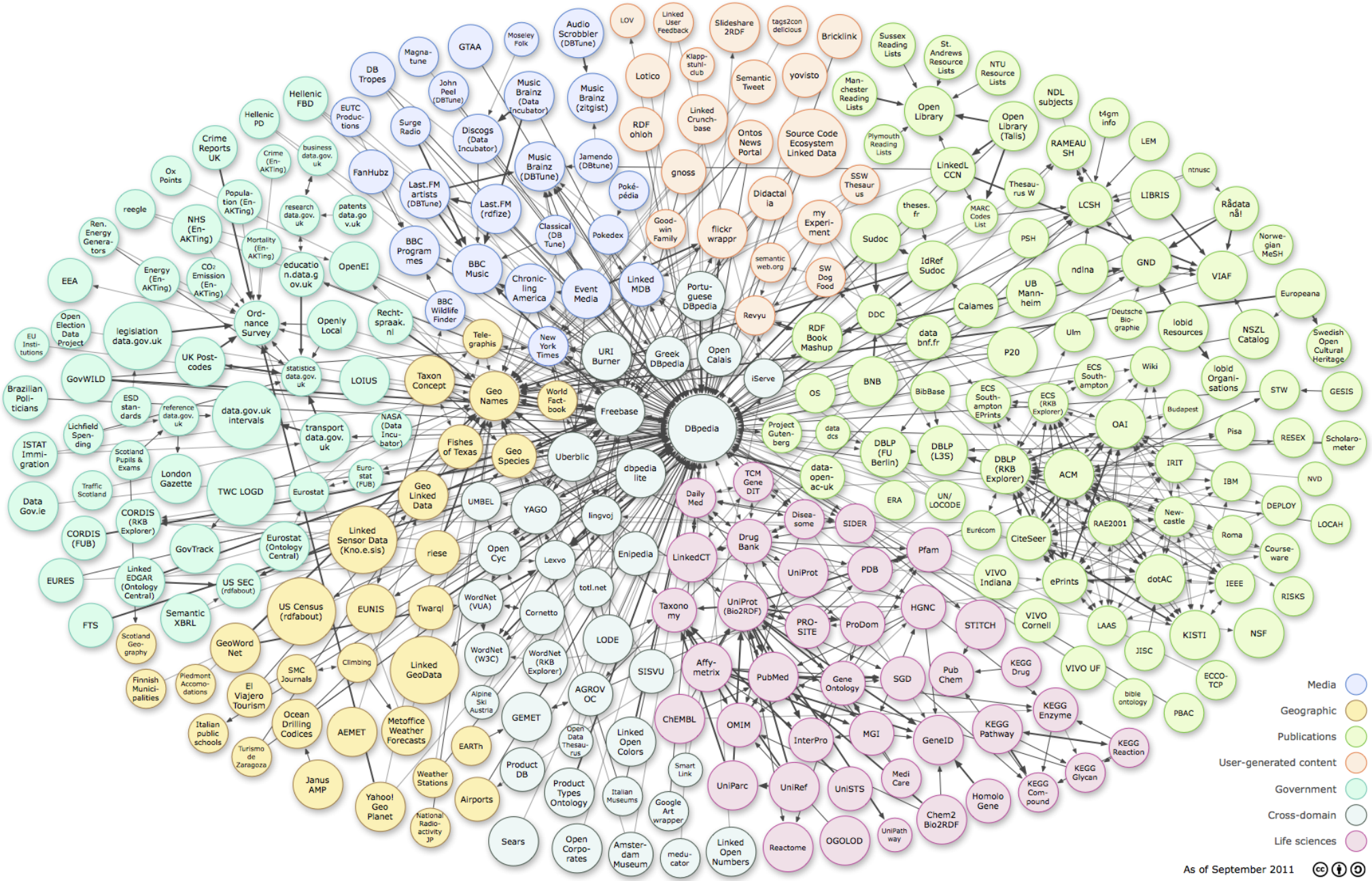

Figura 6: Grafo de datos enlazados

3. Fuentes de datos

Como se ha descrito anteriormente, el principal objetivo de este proyecto es reutilizar información relativa al mundo del cine y poder combinar y completar esa información accediendo a varias fuentes de datos. Para poder manipular esa información correctamente ha sido necesario realizar un profundo análisis de las fuentes datos y de la forma de acceso a estas, teniendo como precondición que fueran fuentes de acceso libre. El resultado de la investigación es el siguiente:

3.1 Cómo obtener información de las fuentes de datos

Para poder determinar las fuentes de datos que se iban a utilizar en este proyecto, ha sido necesario encontrar puntos de información generalizada sobre todas las fuentes de datos de la Web de Datos. Uno de los puntos de información sobre la Web de Datos más importantes actualmente es *The Data Hub*.

The Data Hub[26] es un catálogo de conjuntos de datos en el que cualquiera puede publicar contenido semántico y que constituye ahora mismo la mayor unión de bases de conocimiento de la Web de Datos. En *The Data Hub* cada conjunto de datos referenciado o guardado tiene una descripción de los datos contenidos y otras informaciones útiles, como pueden ser los formatos en los que los datos están disponibles, a quien pertenecen dichos datos o si son de uso libre. Cuenta con un índice que permite realizar búsquedas sobre los datos siendo la mayor parte de ellos de libre uso, lo cual implica que al tener la opción de reutilizarse los datos pueden crecer en gran medida por la colaboración de los usuarios del mismo modo que crecen las entradas contenidas en Wikipedia[27].

[26] http://datahub.io/
[27] http://www.wikipedia.org/

The Data Hub está desarrollado mediante la herramienta de gestión de datos de uso libre CKAN[28] (Comprehensive Knowledge Archive Network) creada por la Open Knowledge Foundation[29] y actualmente contiene 6535 conjuntos de datos de los cuales 339 son de *Linked Data*.

Otro punto de información necesario para la realización del presente proyecto y para que los usuarios puedan obtener el máximo beneficio del uso de la aplicación es el conocimiento sobre la disponibilidad de los puntos de acceso. Es necesario que los puntos de acceso tengan alta disponibilidad para poder obtener información completa durante la mayor parte del tiempo. Para obtener esta información ha resultado muy útil el proyecto *SPARQL Endpoints Status*[30] desarrollado por Pierre-Yves Vandenbussche[31]. Este proyecto actualiza la lista y el estado de los puntos de acceso SPARQL indexados en *The Data Hub* de manera dinámica haciendo consultas al punto de acceso SPARQL de *The Data Hub* y generando a su vez un punto de acceso SPARQL con los datos de disponibilidad generados por la herramienta.

Esta información se ha tenido muy en cuenta en la realización de las pruebas del proyecto para determinar si algunos problemas surgidos eran debido a las consultas realizadas a los puntos de acceso o debido a una caída momentánea del servicio. A continuación se muestra un gráfico con la disponibilidad del punto de acceso SPARQL de Linked Movie DataBase:

[28] http://ckan.org/
[29] http://okfn.org/
[30] http://labs.mondeca.com/sparqlEndpointsStatus/
[31] https://sites.google.com/site/pierreyvesvandenbussche/

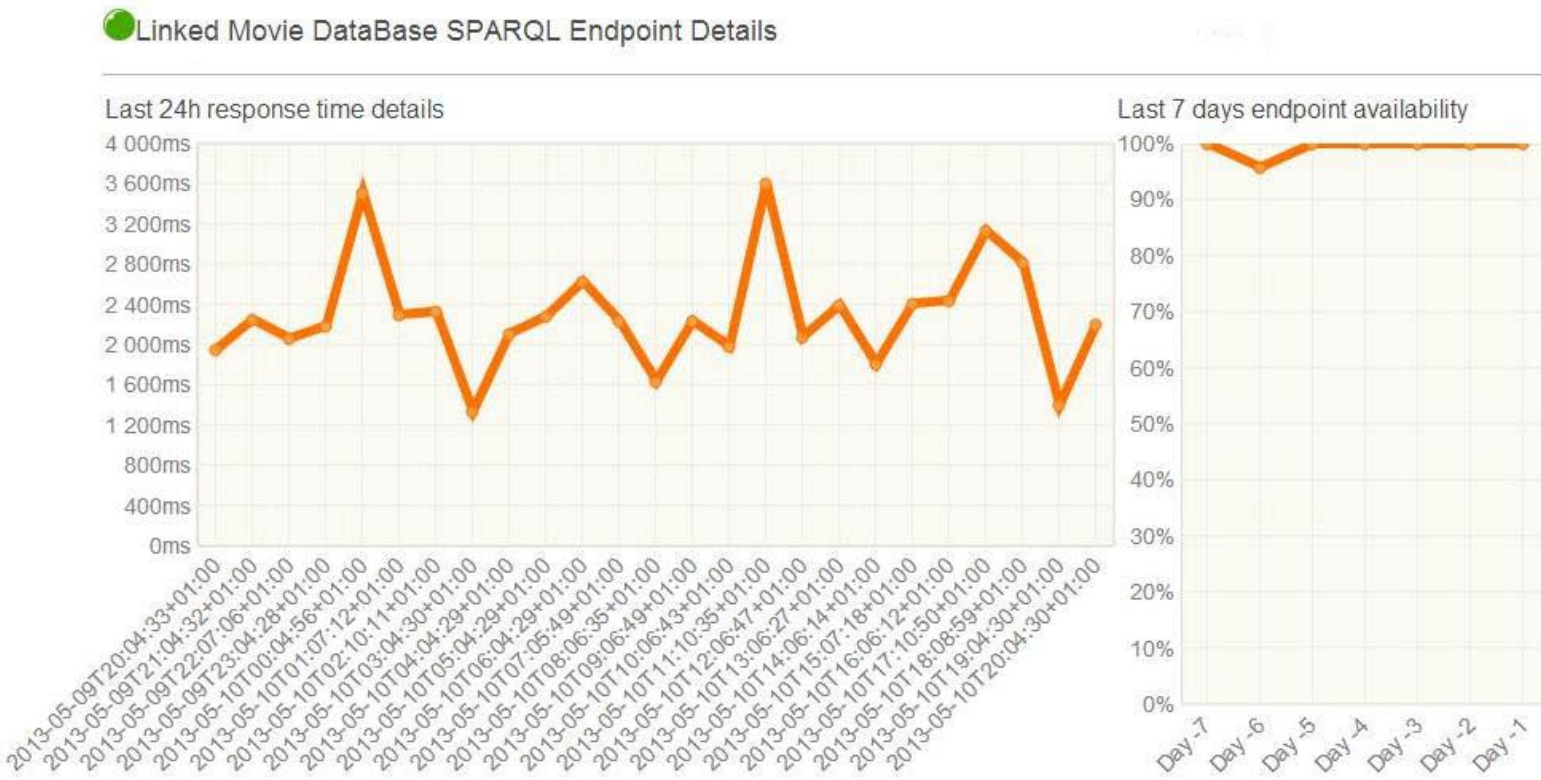

Figura 7: Gráfico de disponibilidad del punto de acceso SPARQL de LMDB

3.2 Linked Movie DataBase

Linked Movie Data Base[32] (LMDB) es la principal fuente de datos que se utiliza en este proyecto y que sirve como punto de partida de los vínculos entre todos los datos mostrados al usuario.

El proyecto LMDB creado por Oktie Hassanzadeh y Mariano P. Consens es el primer proyecto de datos semánticos de libre acceso relacionados con el cine y ganó el primer premio del *LOD Triplification Challenge*[33]. Los enlaces entre datos han sido creados mediante la herramienta *ODDLinker* que hace uso de medio millón de atributos sobre alrededor de cuarenta mil películas y otras entidades relacionadas con las películas. Los datos enlazados de LMDB están publicados mediante un servidor D2R.

[32] http://linkedmdb.org/
[33] http://triplify.org/Challenge

En la siguiente figura pueden verse los enlaces de LMDB a otros conjuntos de datos pertenecientes al *Linking Open Data Proyect* [34]entre los que se encuentran:

- Geo-names[35]
- DBpedia[36]
- MusicBrainz[37]
- flickrWrappr[38]
- lingvoj[39]
- RDF Book Mashup[40]

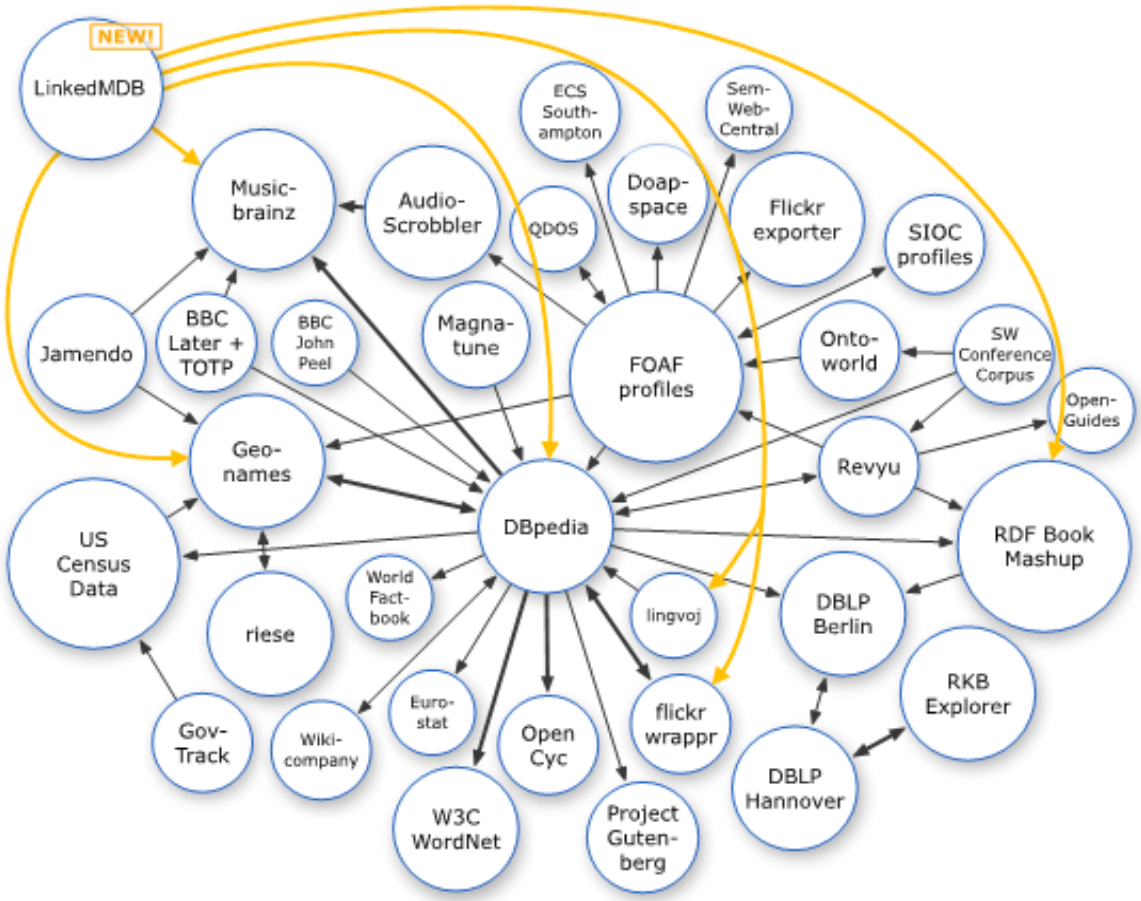

Figura 8: Grafo de enlaces de Linked Movie DataBase

[34] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[35] http://www.geonames.org/
[36] http://dbpedia.org
[37] http://musicbrainz.org/
[38] http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/
[39] http://www.lingvoj.org/
[40] http://wifo5-03.informatik.uni-mannheim.de/bizer/bookmashup/

Además también posee enlaces a importantes páginas web relacionadas con el cine tales como:

- IMDB (Internet Movie Data Base)[41]
- Rotten Tomatoes[42]
- FreeBase[43]

En LMDB el acceso a los datos se puede realizar de varias maneras, siendo la elegida para la realización de este proyecto la consulta al punto de acceso mediante un cliente SPARQL:

- Mediante un navegador web (Google Chrome, Mozilla Firefox, etc.)
- Mediante un navegador web semántico (Marbles, DISCO, Tabulator) accediendo a http://data.linkedmdb.org/all
- Mediante un Cliente SPARQL tomando como punto de acceso la siguiente dirección http://data.linkedmdb.org/sparql

Es importante tener en cuenta la cantidad de enlaces existentes en una fuente de datos que se vaya a utilizar, para poder determinar la cantidad de resultados que se va a obtener al realizar búsquedas de películas. En las siguientes tablas se muestran las estadísticas totales de enlaces totales contenidas en LMDB y las estadísticas de enlaces en función de la propiedad:

Tabla 1: Estadísticas de Linked Movie DataBase

Número de triplas publicadas por LMDB.	6 148 121
Número de enlaces a otras fuentes de datos de LOD	162 199
Número de referencias a sitios web de películas	541 810
Número de entidades en LMDB	503 242

[41] http://www.imdb.com/
[42] http://www.rottentomatoes.com/
[43] http://www.freebase.com/

Tabla 2: Número de enlaces por propiedad en Linked Movie DataBase

Destino	**Propiedad**	**Número de enlaces**
DBPedia	owl:sameAs	30 354
YAGO	owl:sameAs	30 354
flickr™ wrappr	dbpedia:hasPhotoCollection	30 354
RDF Book Mashup (Books)	movie:relatedBook	700
RDF Book Mashup (Authors)	rdfs:SeeAlso	12 990
MusicBrainz	owl:sameAs	2 207
GeoNames	foaf:based_near	27 272
GeoNames	owl:sameAs	272
lingvoj	movie:language	28 253
IMDb, Rotten Tomatoes, Freebase.com	foaf:page	541 810

Teniendo los datos estadísticos anteriores, se escogerán para el proyecto los enlaces generados a través de DBpedia, MusicBrainz, Geonames y los enlaces a sitios web relacionados. Mediante la consulta al punto de acceso SPARQL de LMDB se obtiene información de las siguientes propiedades de una película con sus respectivos identificadores de recurso (URI):

- Titulo <http://purl.org/dc/terms/title>
- Fecha <http://purl.org/dc/terms/date>
- Duración <http://data.linkedmdb.org/resource/movie/runtime>
- Actor <http://data.linkedmdb.org/resource/movie/actor>
- Director <http://data.linkedmdb.org/resource/movie/director>
- Productor <http://data.linkedmdb.org/resource/movie/producer>
- Escritor <http://data.linkedmdb.org/resource/movie/writer>
- Editor <http://data.linkedmdb.org/resource/movie/editor>
- Genero <http://data.linkedmdb.org/resource/movie/genre>
- Localizaciones del rodaje <http://data.linkedmdb.org/resource/movie/featured_film_location>

- Autores de la banda sonora
  <http://data.linkedmdb.org/resource/movie/music_contributor>
- Distribuidora
  <http://data.linkedmdb.org/resource/movie/film_distributor>
- Enlace a sitios web relacionados <http://xmlns.com/foaf/0.1/page>

A continuación se muestra un grafo con la estructura de la consulta que se debe realizar a LMDB en el que se incluyen las propiedades que se utilizan para el enlazado con otras fuentes de datos:

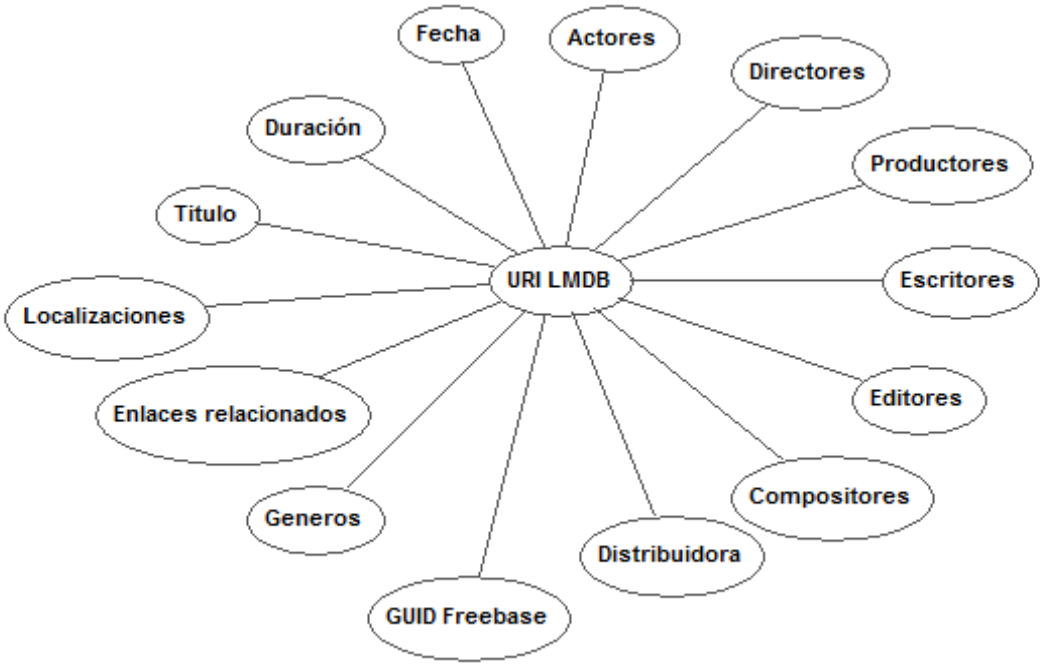

Figura 9: Estructura de la consulta a LMDB

3.3 DBpedia

DBpedia[44] es un proyecto que comenzó en 2007 desarrollado en colaboración conjunta por la Universidad de Leipzig[45], la Freie Universität Berlin[46] y la organización OpenLink Software[47]. El objetivo principal de DBpedia es la extracción automática de información estructurada de Wikipedia[48] para hacer accesible esta información en la Web de Datos.

La base de conocimiento de DBpedia tiene actualmente 3.77 millones de entidades de las cuales 2.35 millones están clasificadas mediante una ontología y que en conjunto suponen 1.89 billones de triplas RDF. Entre otros muchos datos relevantes DBpedia contiene por ejemplo etiquetas y descripciones de cada una de estas entidades en 111 idiomas distintos.

Tabla 3: Información de DBpedia en the Data Hub

Fuente	http://dbpedia.org/
Autor	DBpedia Team - http://wiki.dbpedia.org/Imprint
Mantenedor	DBpedia Team - http://wiki.dbpedia.org/Imprint
Versión	2010-09-02 (3.7)
links:2000-us-census-rdf	12529
links:dbtune-musicbrainz	22981
links:education-data-gov-uk	1697
links:eunis	3600
links:flickr-wrappr	8800000
links:freebase	3400000
links:fu-berlin-dailymed	43
links:fu-berlin-dblp	196
links:fu-berlin-diseasome	1943
links:fu-berlin-drugbank	729
links:fu-berlin-eurostat	137
links:fu-berlin-project-gutenberg	2510

[44] http://dbpedia.org
[45] http://www.zv.uni-leipzig.de/
[46] http://www.fu-berlin.de/en/
[47] http://www.openlinksw.com/
[48] http://www.wikipedia.org/

links:fu-berlin-sider	751
links:geonames-semantic-web	86547
links:geospecies	11400
links:italian-public-schools-linkedopendata-it	5822
links:linkedgeodata	53024
links:linkedmdb	13800
links:nytimes-linked-open-data	10359
links:opencyc	20362
links:rdf-book-mashup	9078
links:reference-data-gov-uk	22
links:revyu	6
links:tcmgenedit__dataset	904
links:transport-data-gov-uk	3768
links:uk-legislation-api	33
links:w3c-wordnet	467101
links:wikicompany	8348
links:world-factbook-fu-berlin	233
links:yago	18100000
namespace	http://dbpedia.org/resource/
triples	1200000000

Actualmente hay una gran cantidad de publicaciones de datos y de contribuciones a la Web de Datos enlazados que resultan en un conjunto enorme de temas enlazados. Dado que DBpedia define URIs para millones de conceptos muchas de las publicaciones de datos contienen enlaces a DBpedia, lo cual le confiere la posición de núcleo central de la Web de Datos.

DBpedia ofrece varias ventajas sobre las bases de conocimiento existentes entre las que se encuentra la posibilidad de realizar consultas complejas sobre DBpedia y obtener así información inferida que Wikipedia no podría ofrecernos.

En cuanto a la arquitectura, los datos de DBpedia están publicados con OpenLink Virtuoso[49] y pueden obtenerse mediante consultas a su punto de acceso SPARQL, mediante la descarga de los datos en RDF[50] o mediante peticiones HTTP desde navegadores habituales.

[49] http://virtuoso.openlinksw.com/

[50] http://wiki.dbpedia.org/Downloads32

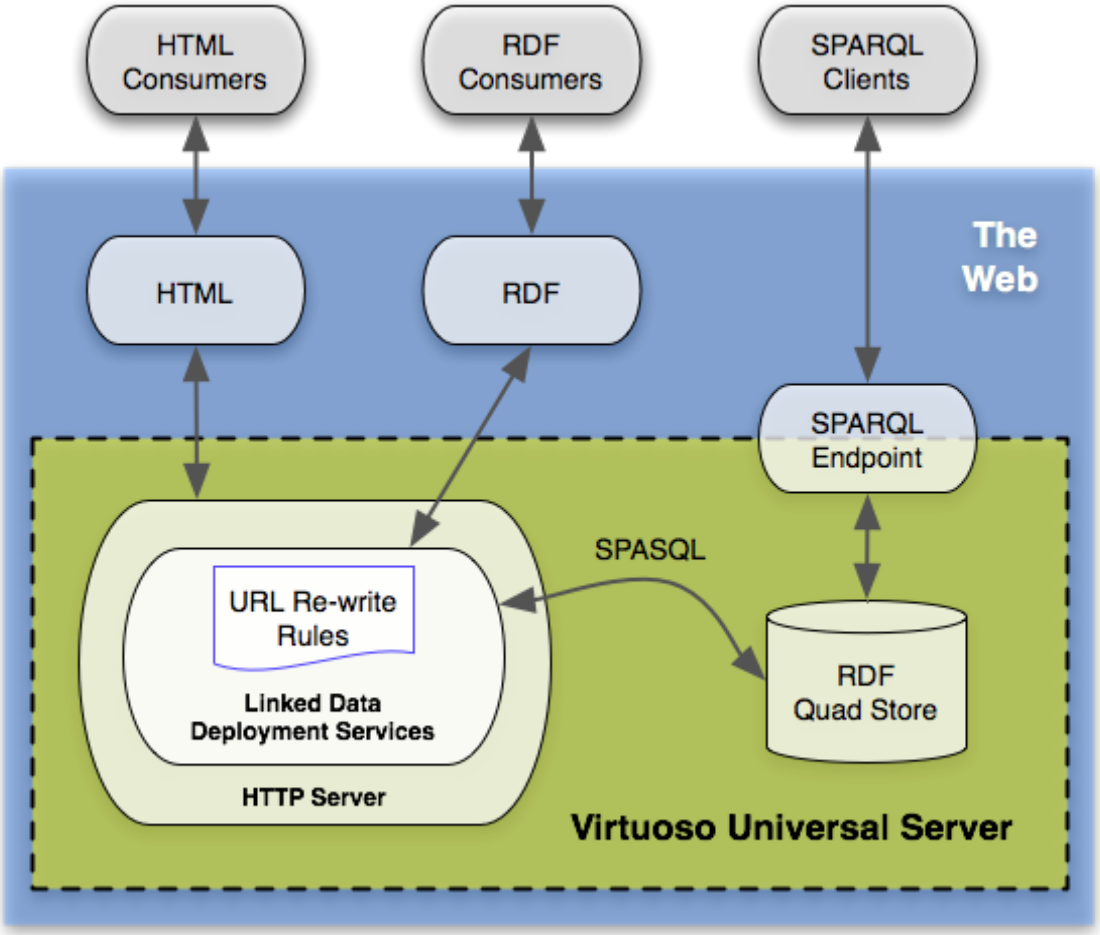

Figura 10: Arquitectura de DBpedia

En este proyecto se realizan consultas al punto de acceso SPARQL de DBpedia para obtener información de las siguientes propiedades de una película con sus respectivos identificadores de recurso (URI):

- Descripción < http://dbpedia.org/ontology/abstract>
- Colección de fotos de usuarios <http://dbpedia.org/property/hasPhotoCollection>
- Enlace a Wikipedia <http://xmlns.com/foaf/0.1/isPrimaryTopicOf>
- País <http://dbpedia.org/property/country>

A continuación se muestra un grafo con la estructura de la consulta que se debe realizar a DBpedia en el que se incluyen las propiedades que se utilizan para el enlazado con otras fuentes de datos:

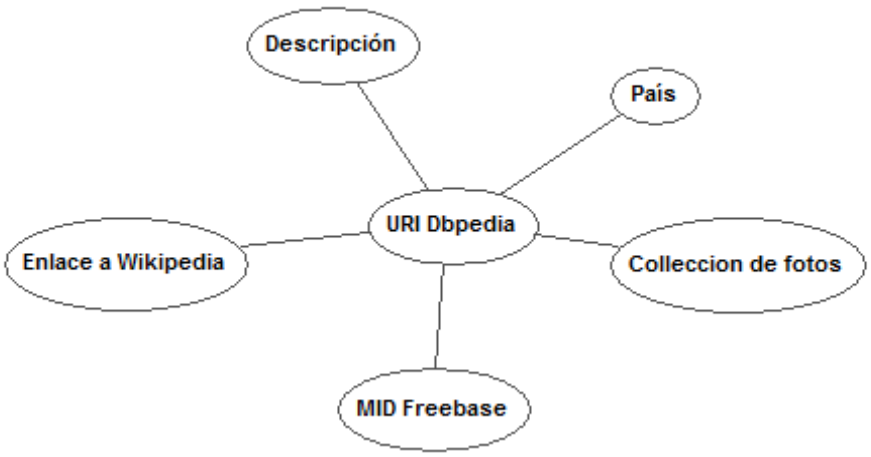

Figura 11: Estructura de la consulta a DBpedia

3.4 Revyu

Revyu Anything es un sitio web libremente accesible en la que cualquier usuario registrado puede realizar cualquier crítica y valoración de cualquier cosa que desee. Desarrollado por Tom Heath y Enrico Motta nace con el objetivo de permitir la reutilización de los sistemas de opiniones existentes actualmente en la web tradicional y que han adquirido tanto valor para el usuario durante el desarrollo de la Web 2.0.

Tabla 4: Información de Revyu en the Data Hub

Fuente	http://revyu.com/
Autor	Tom Heath
Mantenedor	Mantenedor no especificado
links:dbpedia	29
namespace	http://revyu.com/reviews/
Triplas	20000

Revyu utiliza el sistema de etiquetado que otorga al usuario que realiza la crítica la flexibilidad de etiquetarla con los términos que considere adecuados. Además de esta funcionalidad ofrecida a los usuarios, Revyu genera datos en RDF de las críticas realizadas por los usuarios y de manera transparente a estos, de modo que los datos son reusables por otras aplicaciones además de contar con una interfaz que

no requiere conocimientos técnicos por parte del usuario. Si un elemento ha sido etiquetado como 'film' o 'movie' Revyu hace una consulta al punto de acceso SPARQL de DBpedia para buscar películas con el mismo nombre que el de la crítica, si se encuentra algún resultado entonces se determina que el elemento es una película y se añade la propiedad <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> a los datos RDF de la crítica.

Como puede verse en la siguiente figura Revyu consume datos de FOAF, DBpedia, Open Guide y RDF Book Mashup para poder hacer inferencias semánticas a partir de las etiquetas establecidas por los usuarios:

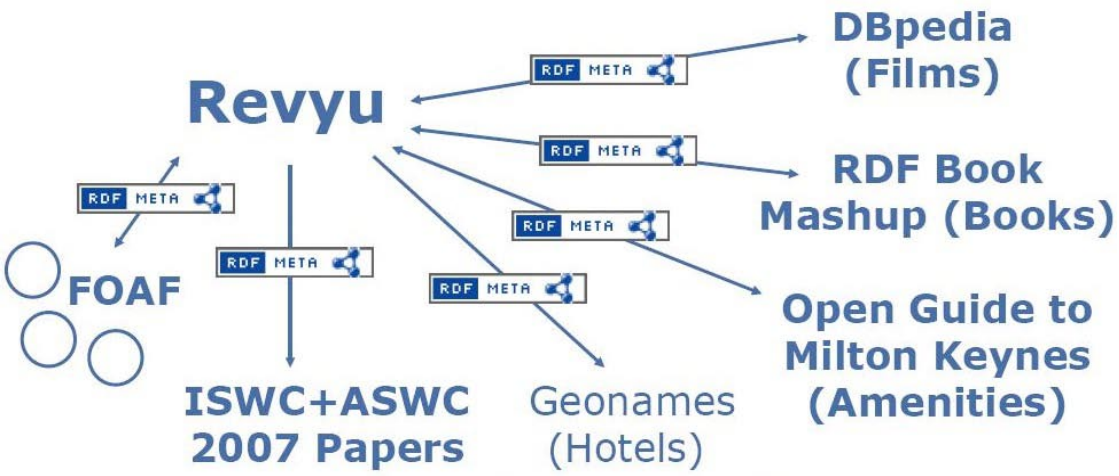

Figura 12: Fuentes de datos de Revyu

Revyu está construido con tecnologías de la web tradicional (Apache, MySQL y PHP) pero también utiliza una API de RDF para PHP llamada RAP[51] para tratar los datos RDF, cuyas tripletas se persisten en una base de datos MySQL de normalizada. Entre las formas de acceso a los datos de Revyu se encuentran las peticiones HTTP de los recursos RDF contenidos en la base de datos MySQL y un punto de acceso SPARQL.

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

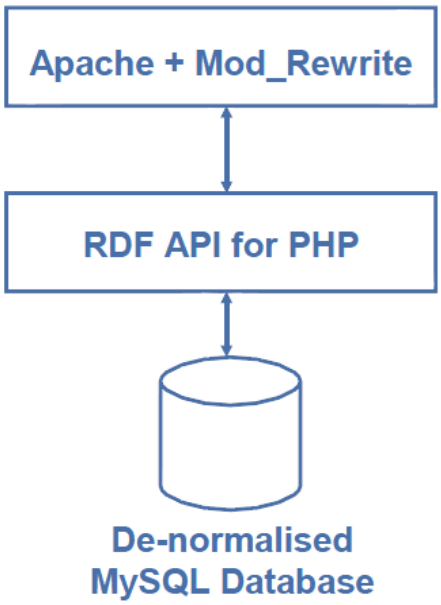

Figura 13: Arquitectura de Revyu

En este proyecto Revyu es la fuente de datos de la que se obtienen las críticas de los usuarios sobre las películas buscadas. Para ello se utilizan las siguientes propiedades:

- Tiene_crítica <http://purl.org/stuff/rev#hasReview>
- Texto <http://purl.org/stuff/rev#text>
- Puntuación <http://purl.org/stuff/rev#rating>

Actualmente Revyu ha dejado de permitir registros nuevos debido a la necesidad de controlar la edición de contenido, para evitar contenido inapropiado, puesto que una de las bases de la reutilización de contenido que promueve la web de datos es la confianza que debe tenerse en ese contenido.

A continuación se muestra un grafo con la estructura de la consulta que se debe realizar a Revyu en el que se incluyen las propiedades que se utilizan para el enlazado con otras fuentes de datos:

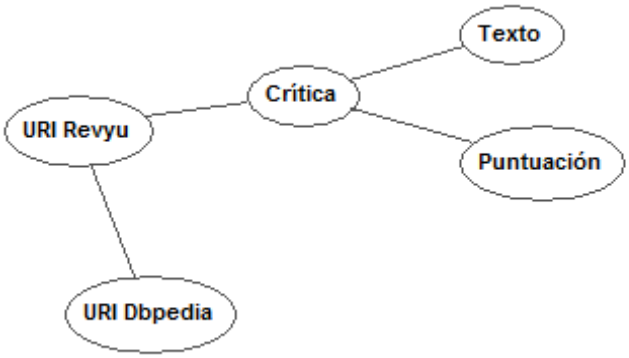

Figura 14: Estructura de la consulta a Revyu

3.5 Sindice

Sindice es un índice de documentos semánticos que se utiliza en este proyecto como alternativa para la recuperación de los datos de Revyu. Ha sido creado conjuntamente por DERI, Fondazione Bruno Kessler y OpenLink software y fue concebido como un servicio para que clientes web semánticos pudieran encontrar fuentes de datos relevantes a partir de una URI del recurso, una propiedad o la búsqueda de una palabra clave.

Se puede acceder a los datos indexados por Sindice mediante una Web API, con un navegador web o a través de un punto de acceso SPARQL. Los resultados obtenidos de Sindice son datos RDF con enlaces a los recursos buscados y que siguen los principios de los datos enlazados.

3.6 Flickr wrappr

Flickr wrappr es un proyecto de la web de datos desarrollado por Chris Bizer y Christian Becker de la Freie Universität Berlin que extiende DBpedia con enlaces a fotografías que se encuentran en el sitio web Flickr[52], uno de los mayores repositorios de fotografías del mundo. Se utiliza en este proyecto para mostrar fotografías

[52] http://www.flickr.com/

relacionadas con la película que se busca a fin de obtener un valor agregado creado por los usuarios de Flickr. Estos disponen de un sistema de etiquetado sin reglas específicas y sin restricciones de idioma a partir del cual se obtiene la relación película-fotografía.

Tabla 5: Información de flickr wrappr en the Data Hub

Fuente	http://www4.wiwiss.fu-berlin.de/flickrwrappr
Autor	Christian Becker
Mantenedor	Maintainer not given
links:dbpedia	3400000
namespace	http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/
triplas	56100000

Flickr wrappr esta implementado como un script de PHP que sigue las especificaciones de negociación de contenido recomendadas por el Linking Open Data Project. Cada vez que se recibe la petición de una entrada, consulta el punto de acceso SPARQL de DBpedia para obtener las etiquetas relacionadas con la entrada que se ha pedido y le pasa la información sobre estas etiquetas a la API de búsquedas de Flickr. El código fuente de flickr wrappr se encuentra disponible en el repositorio SVN de DBpedia:

https://dbpedia.svn.sourceforge.net/svnroot/dbpedia/related_apps/flickrwrappr

El agregado de datos generados por flickr wrappr puede accederse mediante peticiones HTTP a la dirección URL http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/***Entrada*** siendo 'Entrada' el identificador de un artículo de la versión inglesa de Wikipedia *http://en.wikipedia.org/wiki/**Entrada.*** Los resultados que se obtienen de flickr wrappr son un conjunto de datos en RDF que utilizan la propiedad *foaf:depiction* <http://xmlns.com/foaf/0.1/depiction>

http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/photos/Forrest_Gump

A continuación se muestra un grafo con la estructura de la consulta que se debe realizar a flickr wrappr en el que se incluyen las propiedades que se utilizan para el enlazado con otras fuentes de datos:

Figura 15: Estructura de la consulta a flickr wrappr

3.7 Freebase

Freebase es una base de conocimiento colaborativa que recopila datos de varias fuentes como Wikipedia, ChefMoz, NNDB y MusicBrainz además de incluir las contribuciones de miembros de su comunidad para lo cual proporciona una interfaz de usuario para añadir metadatos y clasificaciones o enlaces semánticos . Fue creado por la empresa estadounidense Metaweb[53] en el año 2007 y adquirida por Google[54] en 2010. Tim O'Reilly describió Freebase como 'el puente entre la visión de abajo hacia arriba de la inteligencia colectiva de la Web 2.0 y el mundo más estructurado de la web semántica'.

Tabla 6: Información de Freebase en the Data Hub

Fuente	http://freebase.com/
Autor	Google
Mantenedor	Shawn Simister
links:bbc-music	350110
links:dbpedia	3348530
links:geospecies	100000
links:nytimes-linked-open-data	9930
links:sec-rdfabout	120626
namespace	http://rdf.freebase.com/
triplas	337203427

[53] http://es.wikipedia.org/wiki/Metaweb_Technologies

[54] http://es.wikipedia.org/wiki/Google

Para el almacenamiento de tal cantidad de datos Freebase utiliza un modelo de grafos cuya estructura es un conjunto de nodos y un conjunto de relaciones entre dichos nodos. De este modo se permite el modelado de relaciones mucho más complejas que mediante el uso de una base de datos relacional. Las consultas a este grafo se realizan en el MQL[55] (Metaweb Query Language), lenguaje de consulta análogo a SPARQL pero que utiliza objetos JSON en peticiones y respuestas HTTP.

Se puede acceder a los datos de Freebase mediante peticiones HTTP concatenadas con consultas escritas en MQL o mediante la Topic API, que es un servicio web que devuelve todos los hechos para un tema determinado. También se ofrece la posibilidad de descarga[56] de los datos RDF para la creación de un punto de acceso local. La URL de una petición HTTP a Freebase tendría el siguiente aspecto:

https://www.googleapis.com/freebase/v1/mqlread?query=[{"type":"/music/album","name":null,"artist":{"id":"/en/bob_dylan"}}]

En este proyecto se utiliza Freebase para la obtención de la correspondencia entre la información de la película buscada en LMDB y en DBpedia. Además también se utiliza para la obtención de la imagen principal de la película cuya URL se obtiene mediante el identificador de la imagen con una petición HTTP a la siguiente URL:

https://usercontent.googleapis.com/freebase/v1/image/id_imagen

A continuación se muestra un grafo con la estructura de la consulta que se debe realizar a Freebase en el que se incluyen las propiedades que se utilizan para el enlazado con otras fuentes de datos:

[55] http://wiki.freebase.com/wiki/MQL

[56] http://download.freebase.com/

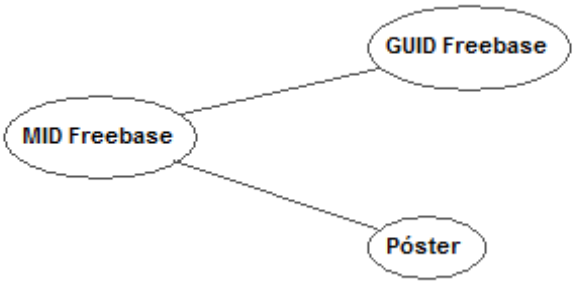

Figura 16: Estructura de la consulta a Freebase

3.8 Enlazando las fuentes de datos

Para poder enlazar los datos y estar seguros de que los contenidos que pedimos a cada punto de acceso remoto son coherentes con la petición del usuario es necesario encontrar las relaciones de unas fuentes de datos con otras. Ya se han descrito anteriormente las consultas que se realizan a cada fuente de datos, por lo que a continuación se muestra el grafo de conexión entre todas las fuentes utilizadas:

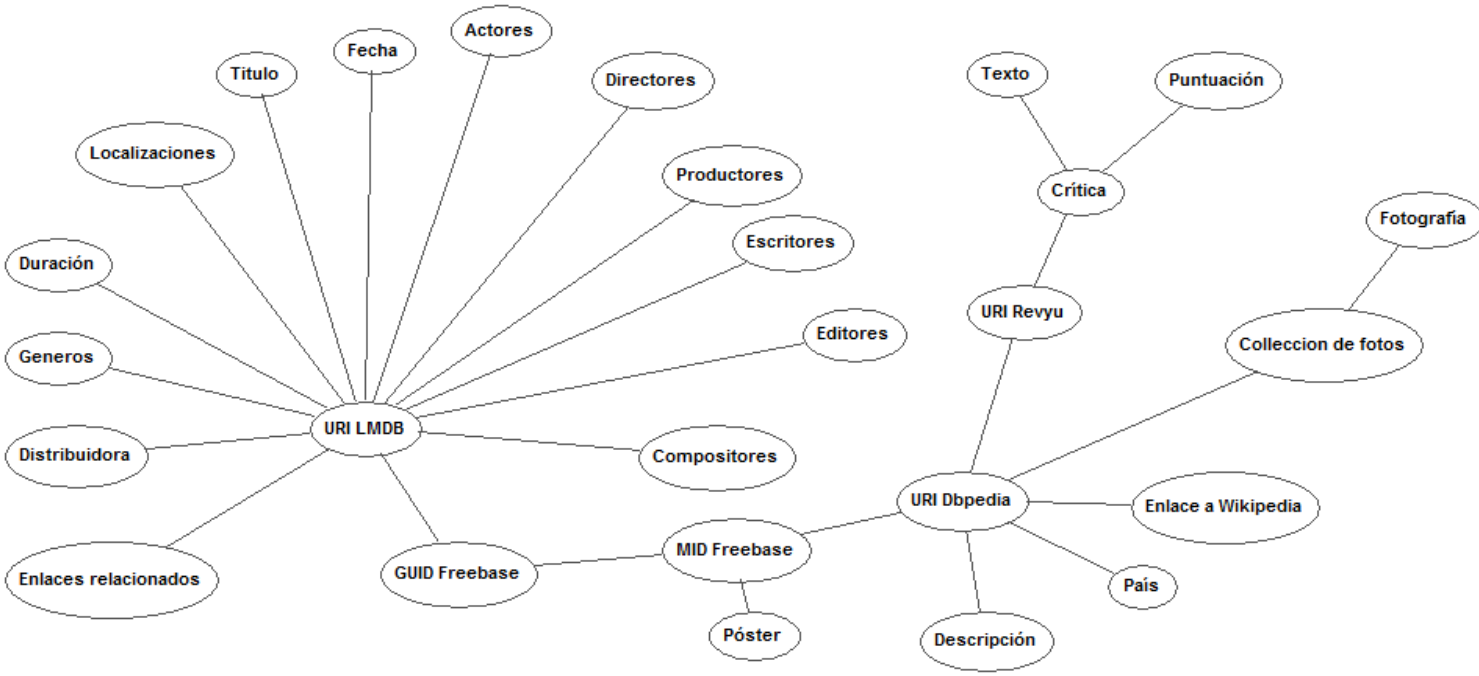

Figura 17: Grafo de enlazado de las fuentes de datos

4. Descripción informática

4.1 Especificación de requisitos:

A continuación se describen los requisitos que debe satisfacer la aplicación para cumplir con los objetivos del proyecto y que serán verificados durante la fase de pruebas:

Tabla 7: Requisitos funcionales de la aplicación web

Requisito	**Descripción**
R1	La aplicación deberá permitir a los usuarios registrados autenticarse mediante su usuario y contraseña.
R2	La aplicación deberá permitir a los usuarios no registrados crear una nueva cuenta de usuario.
R3	La aplicación deberá permitir cerrar sesión a los usuarios previamente autenticados.
R4	La aplicación deberá impedir el acceso a los recursos a usuarios no autenticados.
R5	La aplicación deberá permitir realizar búsquedas de películas dado el nombre en inglés de las mismas.
R6	La aplicación permitirá búsquedas que ignoren mayúsculas y minúsculas para proporcionar al usuario mayor flexibilidad en las búsquedas.
R7	En el caso de encontrarse más de un resultado para una determinada búsqueda, la aplicación mostrara todos los resultados al usuario para permitirle elegir entre ellos.
R8	En caso de no encontrarse información en alguna de las fuentes de datos, la información del resto de las fuentes será presentada al usuario de manera ordenada y coherente.

R9	En caso de no poder establecer conexión con alguna de las fuentes de datos, la información del resto de las fuentes será presentada al usuario de manera ordenada y coherente.
R10	En el caso de no encontrarse resultados será notificado al usuario.

Tabla 8: Requisitos no funcionales de la aplicación web

Requisito	**Descripción**
R11	La aplicación deberá disponer de una interfaz gráfica amigable, intuitiva y adecuada a los tipos más habituales de usuarios finales.
R12	Se deberá ofrecer al usuario una navegación sencilla y cómoda.
R13	La aplicación deberá ser visualizada correctamente en las últimas versiones estables de los siguientes navegadores Mozilla Firefox, Internet Explorer y Google Chrome, instalados en un equipo con sistema operativo Windows XP y versiones posteriores hasta Windows 7.
R14	La aplicación será capaz de realizar las operaciones requeridas con un tiempo de espera corto.
R15	La funcionalidad de la aplicación será fácilmente extensible en la medida necesaria.
R16	La aplicación mostrará toda la información recopilada en un formato consistente y accesible para el usuario.
R17	La aplicación mostrara las atribuciones correspondientes a las diferentes fuentes de datos utilizadas, de acuerdo con las licencias de uso de las mismas.
R18	El buscador estará presente durante toda la navegación para permitir al usuario acceder en cualquier momento a la funcionalidad principal.

Tabla 9: Requisitos de implementación de la aplicación web

Requisito	**Descripción**
R19	La aplicación deberá recopilar información relevante para el usuario de varias fuentes distintas.
R20	Se deberán utilizar al menos tres fuentes de datos distintas para la obtención de información relevante.
R21	Las fuentes de datos deberán contener información en formato de descripción semántica RDF
R22	Algunas de las fuentes de datos deberán ser consultadas mediante el lenguaje SPARQL
R23	Deberá recuperarse la información de las fuentes de datos realizando una correspondencia entre enlaces de unas a otras, siguiendo así los principios de los datos enlazados.
R24	Las consultas realizadas a las fuentes de datos deberán cumplir con los requisitos de uso de los puntos de acceso.
R25	De cada fuente de datos se extraerá en la medida de lo posible información exclusiva con respecto al resto de las fuentes.
R26	Se utilizará el patrón de diseño modelo-vista-controlador

4.2 Análisis

Para poder determinar las clases necesarias según los requisitos definidos anteriormente es necesario identificar los casos de uso y los actores que interactúan en estos casos de uso. Para ello se muestra a continuación un diagrama de casos de uso que proporciona una visión clara de las funcionalidades ofrecidas al usuario. En el caso de esta aplicación en diagrama de casos de uso es bastante sencillo ya que los requisitos más importantes de esta aplicación tienen que ver con la implementación de la funcionalidad y no con la funcionalidad en sí.

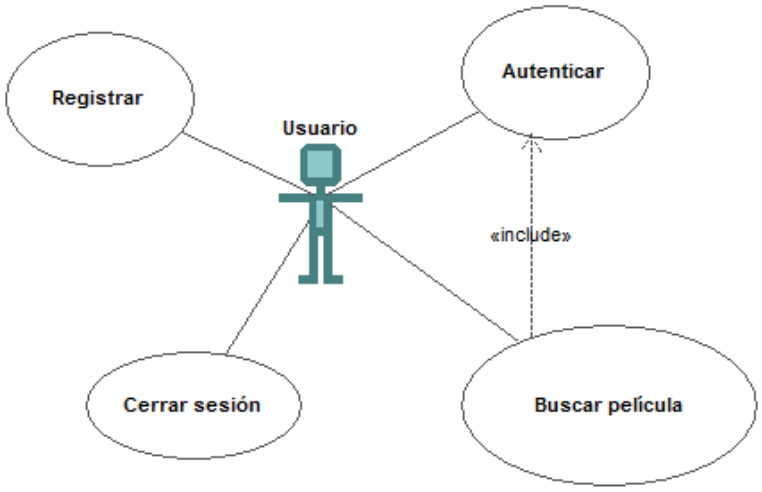

Figura 18: Diagrama de casos de uso

Como puede verse en el diagrama, en esta aplicación solo existe un tipo de actor que sería un usuario interesado en el mundo del cine. Para poder implementar correctamente estos casos de uso se ha realizado un análisis de las clases necesarias, incluyendo los principales atributos de cada clase, así como las principales responsabilidades, en función del análisis de las fuentes de datos realizado previamente. A continuación (Figura 19) se muestra el conjunto de clases analizadas y sus relaciones:

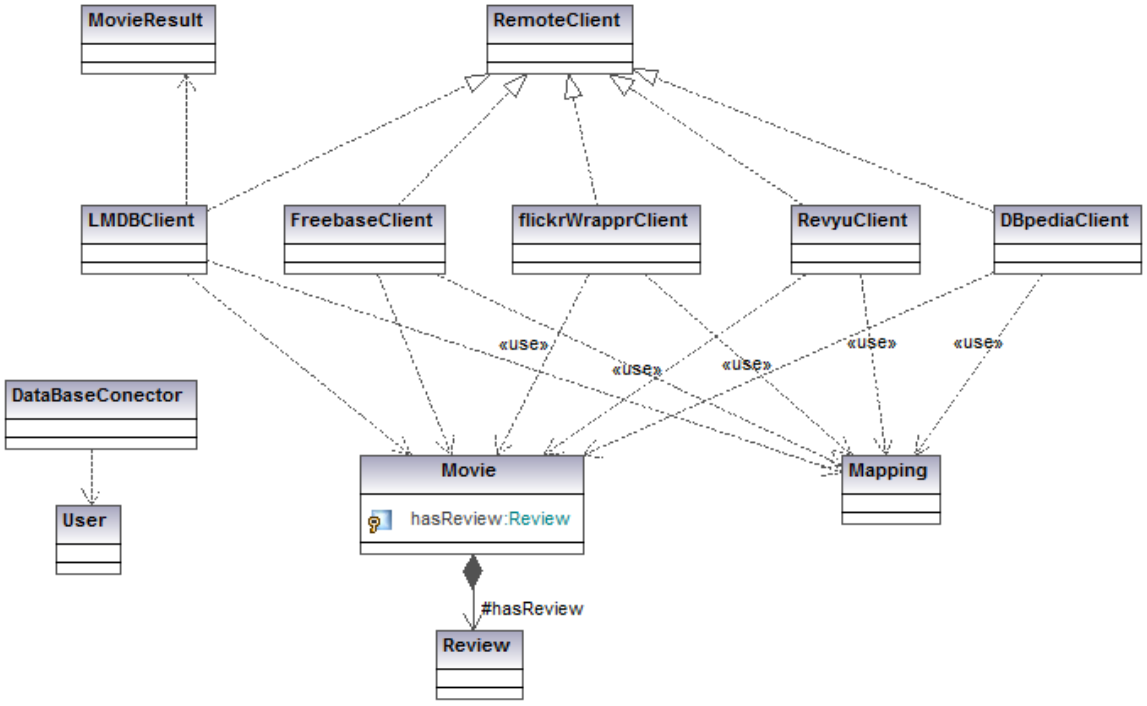

Figura 19: Diagrama de clases de análisis

Clase PELÍCULA

- Esta clase contiene la combinación de toda la información que se presenta al usuario al realizar la búsqueda de la película. También debe ser capaz de añadir y devolver la información recuperada de cada una de las fuentes de datos.
- Atributos:

Titulo	Productores
País	Editores
Fecha	Géneros
Imagen principal	Empresa distribuidora
Duración	Compositores.
Descripción	Localizaciones del rodaje
Actores	Críticas de la película
Directores	Enlaces a sitios web de la película
Escritores	Álbum de fotos de los usuarios

Clase CRÍTICA

- Es la clase que encapsula la información recuperada por la fuente de datos Revyu y que se relaciona con la clase PELÍCULA mediante composición (Una PELÍCULA tiene una serie de **CRÍTICAS** asociadas).
- Atributos:
  - Comentario
  - Valoración

Clase USUARIO

- Esta clase encapsula la información que se almacena de manera persistente de cada usuario que se registra en la aplicación.
- Atributos:

Nombre	Email
Apellidos	Contraseña

Clase BASE DE DATOS

- Esta clase gestiona la conexión del modelo de datos con la base de datos en la que se almacena la información de los usuarios. Deberá permitir añadir usuarios y realizará la autenticación de los mismos.
- Atributos:

  Credenciales de la base de datos

  Identificador de la base de datos de la aplicación.

Clase CLIENTE LMDB

- Esta clase gestiona la comunicación con el punto de acceso remoto de Linked Movie Data Base[57]. Es la encargada de realizar la consulta SPARQL al punto de acceso y recuperar la información para añadirla a la clase PELÍCULA.
- Atributos:

  URL del punto de acceso remoto

Clase CLIENTE DBPEDIA

- Esta clase gestiona la comunicación con el punto de acceso remoto de DBpedia[58]. Es la encargada de realizar la consulta SPARQL al punto de acceso y recuperar la información para añadirla a la clase PELÍCULA.
- Atributos:

  URL del punto de acceso remoto

Clase CLIENTE FREEBASE

- Esta clase gestiona la comunicación con el punto de acceso remoto de Freebase[59]. Es la encargada de realizar la consulta MQL mediante una petición HTTP al punto de acceso y recuperar la información para añadirla a la clase PELÍCULA. Esta clase también es la encargada de realizar la correspondencia entre la información de Linked Movie Database y DBpedia.
- Atributos

  URL del punto de acceso remoto

[57] http://data.linkedmdb.org/sparql

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

[59] https://www.googleapis.com/freebase/v1/mqlread

Clase CLIENTE REVYU

- Esta clase gestiona la comunicación con el punto de acceso remoto de Revyu[60]. Es la encargada de realizar la consulta SPARQL al punto de acceso y recuperar la información para añadirla a la clase PELÍCULA.
- Atributos:

  URL del punto de acceso remoto

Clase CLIENTE FLICKR WRAPPR

- Esta clase gestiona la comunicación con el punto de acceso remoto de flickr wrappr[61]. Es la encargada de realizar la consulta al modelo de datos en RDF mediante una petición HTTP al punto de acceso y recuperar la información para añadirla a la clase PELÍCULA.
- Atributos

  URL del punto de acceso remoto.

Clase RESULTADO-PELÍCULA

- Esta clase contiene el contenido de una película después de la primera búsqueda del usuario en la que únicamente se contempla la fecha y el titulo de la película.
- Atributos:

  URI de la película en LMDB.

  Titulo

  Fecha

Clase MAPPING

- Esta clase contiene la correspondencia entre los enlaces de cada una de las fuentes de datos para una determinada película.
- Atributos:

  LMDBUri

  FreebaseURILMDB

  FreebaseURIDBpedia

  flickrWrapprURI

[60] http://revyu.com/sparql

[61] http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/

4.3 Diseño

4.3.1 Diseño de la interfaz gráfica

Para el diseño de la interfaz gráfica de la aplicación web, se utilizará una plantilla CSS predefinida y se adaptará a las necesidades de la funcionalidad de la aplicación.

Existirá una única página que cambiará de apariencia según las acciones realizadas por el usuario, teniendo así un único nivel de navegación, cosa que permite maximizar la usabilidad de la aplicación. Además en cualquier momento el usuario tendrá la posibilidad de realizar una búsqueda, mediante un campo de búsqueda que estará siempre presente para usuarios autenticados en una zona lo suficientemente visible de la página.

A continuación se muestra la apariencia de la aplicación después de realizar la búsqueda de la película 'Forrest Gump'.

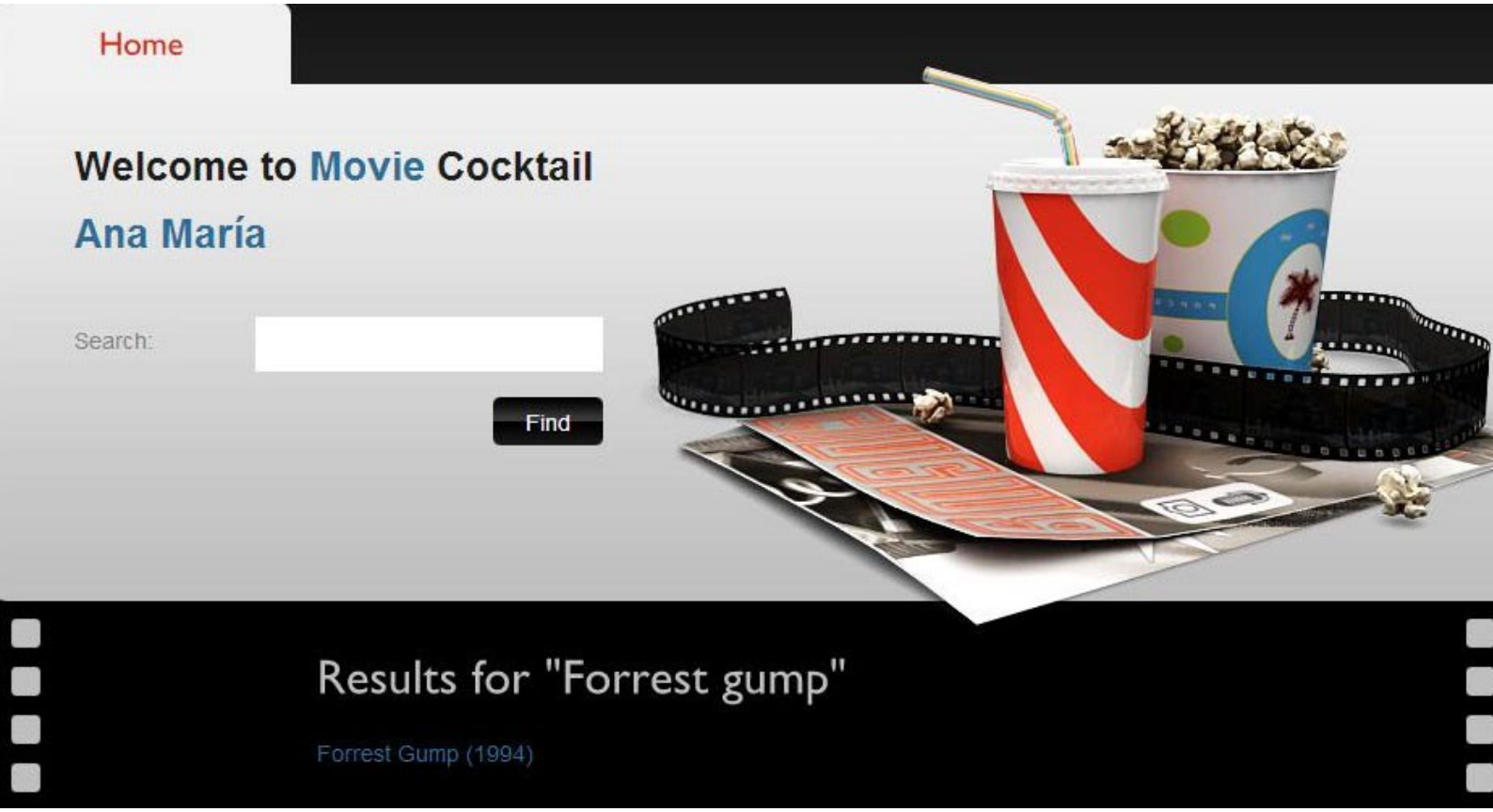

Figura 20: Interfaz gráfica de la aplicación web al realizar una primera búsqueda

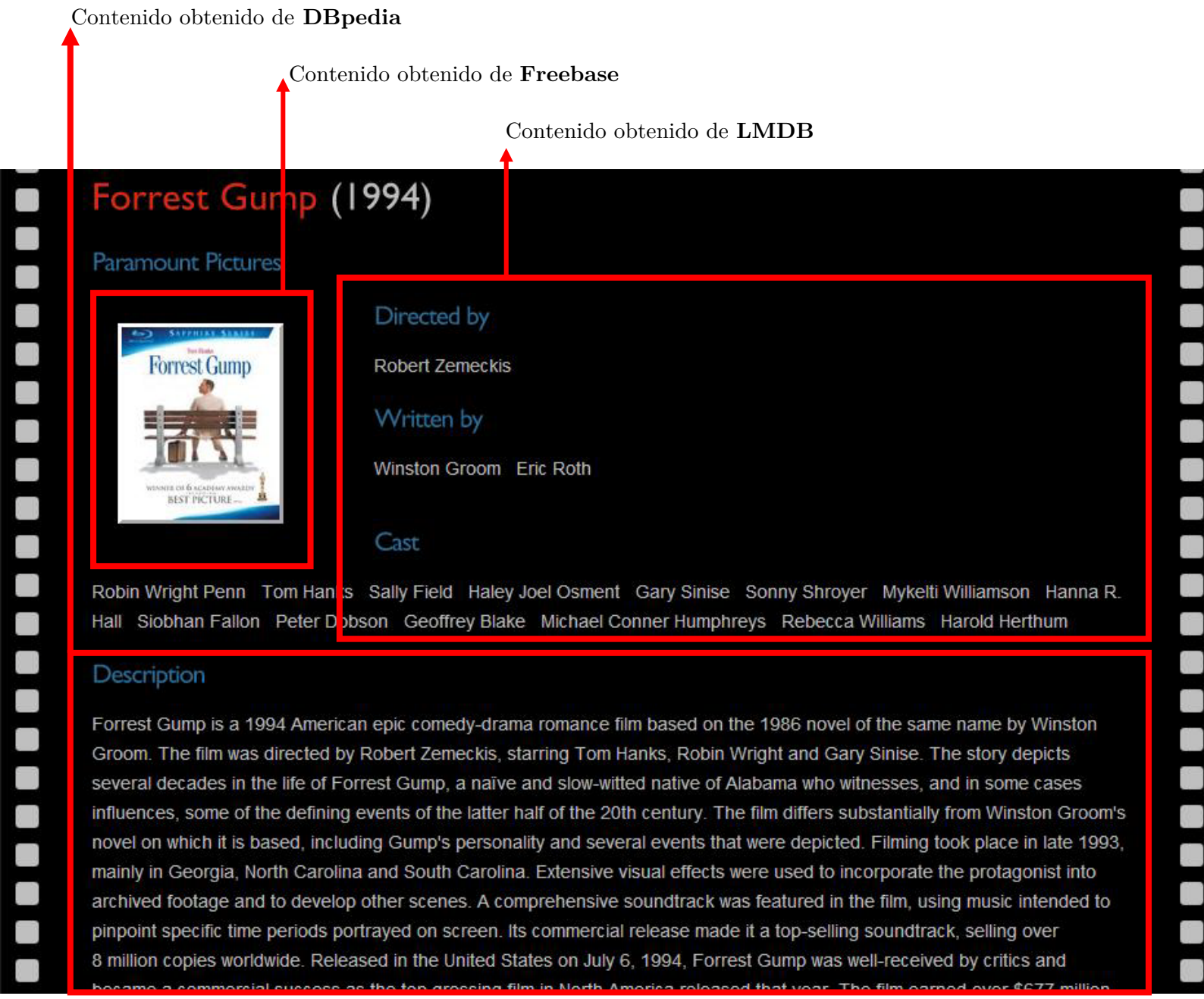

Figura 21: Distribución de los resultado de la búsqueda según la fuente de datos

Las figuras Figura 21 y Figura 22 muestran el contenido completo de una película mostrada al usuario para la que existe información en cada una de las fuentes de datos. En estas figuras se encuentra señalada la procedencia de la información siendo la combinación de la información coherente y con un formato legible para el usuario.

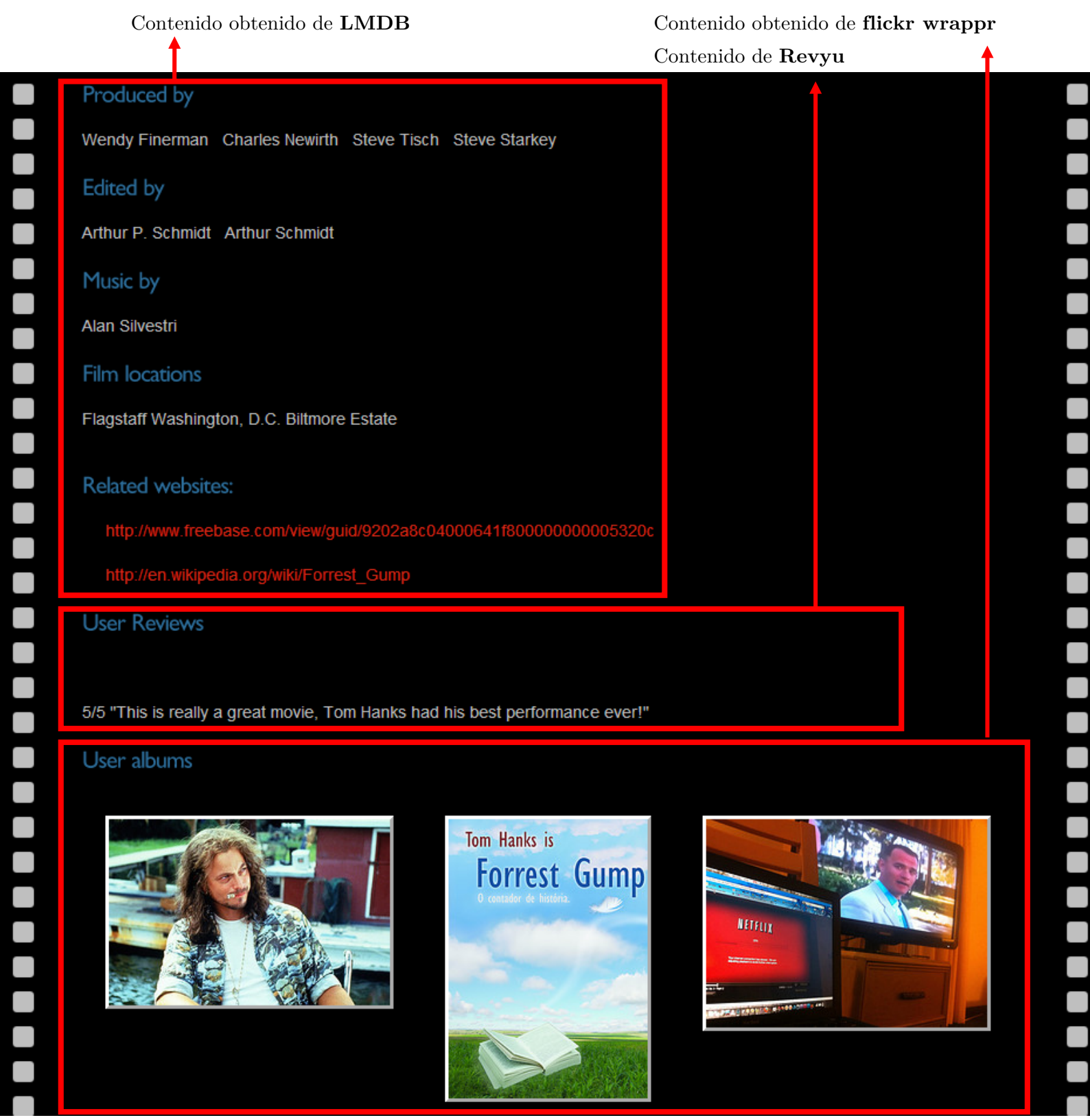

Figura 22: Distribución de los resultado de la búsqueda según la fuente de datos

4.3.2 Diseño de la base de datos

Para el funcionamiento de la aplicación web es necesaria la creación de una tabla en la base de datos que almacenará información sobre los usuarios registrados. A esta tabla se realizaran las consultas pertinentes tanto para la autenticación de los usuarios como para el registro de los mismos. Cada usuario tendrá un identificador único autogenerado que será la clave primaria de la tabla y un identificador 'email' que deberá ser único por lo que deberá gestionarse correctamente el registro nuevo de usuarios con un 'email' existente en la base de datos. El esquema de dicha tabla se detalla en la siguiente figura:

Tabla 10: Esquema de la tabla Usuario

Campo	**Tipo**	**Valor**
idusuario	INT(11)	UNIQUE, PRIMARY KEY
email	VARCHAR(45)	UNIQUE
password	VARCHAR(45)	NOT NULL
nombre	VARCHAR(45)	NOT NULL
apellidos	VARCHAR(45)	NOT NULL

4.3.3 Arquitectura del software

Para cumplir los requisitos de escalabilidad y reusabilidad de la aplicación es necesario utilizar un patrón de diseño adecuado a las necesidades de la aplicación. Por este motivo se decide utilizar el patrón *Modelo – Vista – Controlador* que permite separar la lógica de negocio de la interfaz de usuario.

El flujo de control de esta arquitectura sería el siguiente:

1. El usuario realiza una acción con la interfaz.
2. La vista manda la petición de ejecución de la acción al controlador.
3. El controlador interpreta esta petición y se comunica con el modelo de datos para ejecutar las acciones asociadas a dicha petición.

4. Una nueva vista se muestra al usuario con el estado del modelo después de ejecutar la acción.

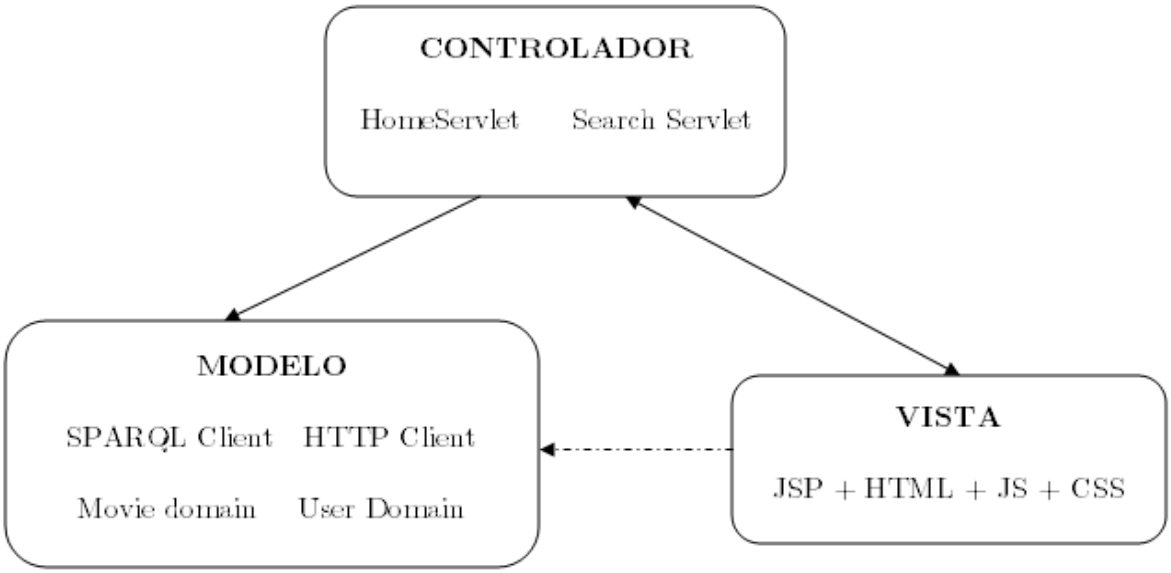

Figura 23: Modelo Vista Controlador

Con este diseño tendremos una serie de páginas JSP que constituirán la vista del usuario (index.jsp, login.jsp, movie.jsp, search.jsp). La index.jsp se encargará de esperar las acciones relacionadas con la autenticación y el registro de nuevos usuarios y se comunicará con HomeServlet para que este ejecute sobre el modelo de datos las acciones pertinentes. Login.jsp representará la vista mostrada al usuario cuando este se acaba de autenticar y se comunicará con SearchServlet para gestionar la acción de la búsqueda de una película. Search.jsp se comunicará con SearchServlet para mostrar al usuario los resultados obtenidos tras la primera búsqueda y permitirá a este seleccionar una de las opciones representadas. Por último movie.jsp mostrará al usuario el conjunto de la información obtenida para una la película seleccionada.

Una vez definida la arquitectura de comunicación entre el controlador y la vista, es necesario el diseño del modelo de datos que contendrá la lógica de negocio y las conexiones con los puntos de acceso remotos para la obtención de información relevante.

A continuación en la Figura 24 se muestra el diagrama de clases y paquetes que constituyen la vista estática de la aplicación con las conexiones entre las tres capas de la arquitectura.

Asimismo para ilustrar la comunicación en el tiempo de los elementos principales de la aplicación se propone el siguiente diagrama de secuencia en el que se muestran todos los elementos involucrados en el proceso de búsqueda de una película (Figura 25).

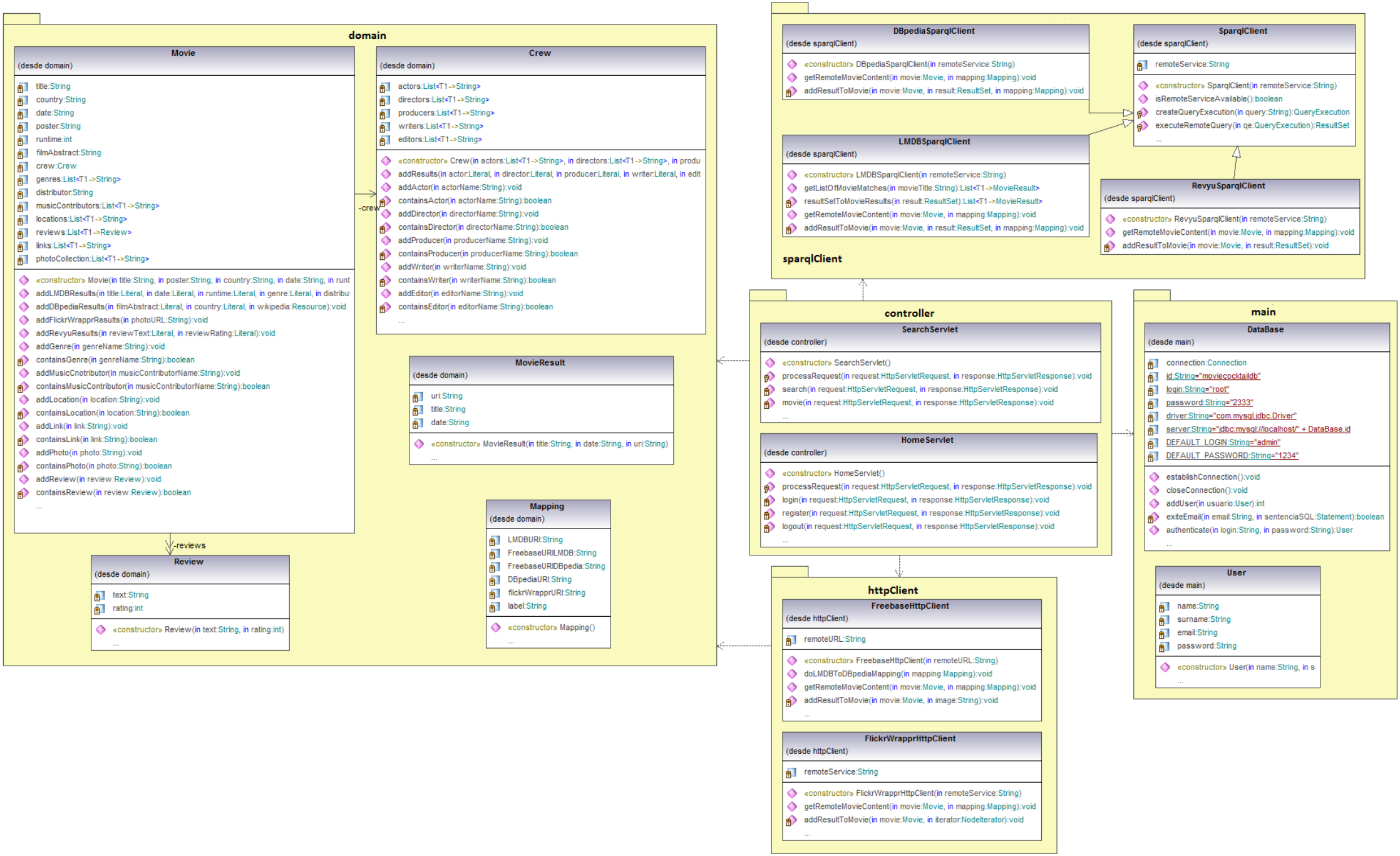

Figura 24: Diagrama de clases y paquetes

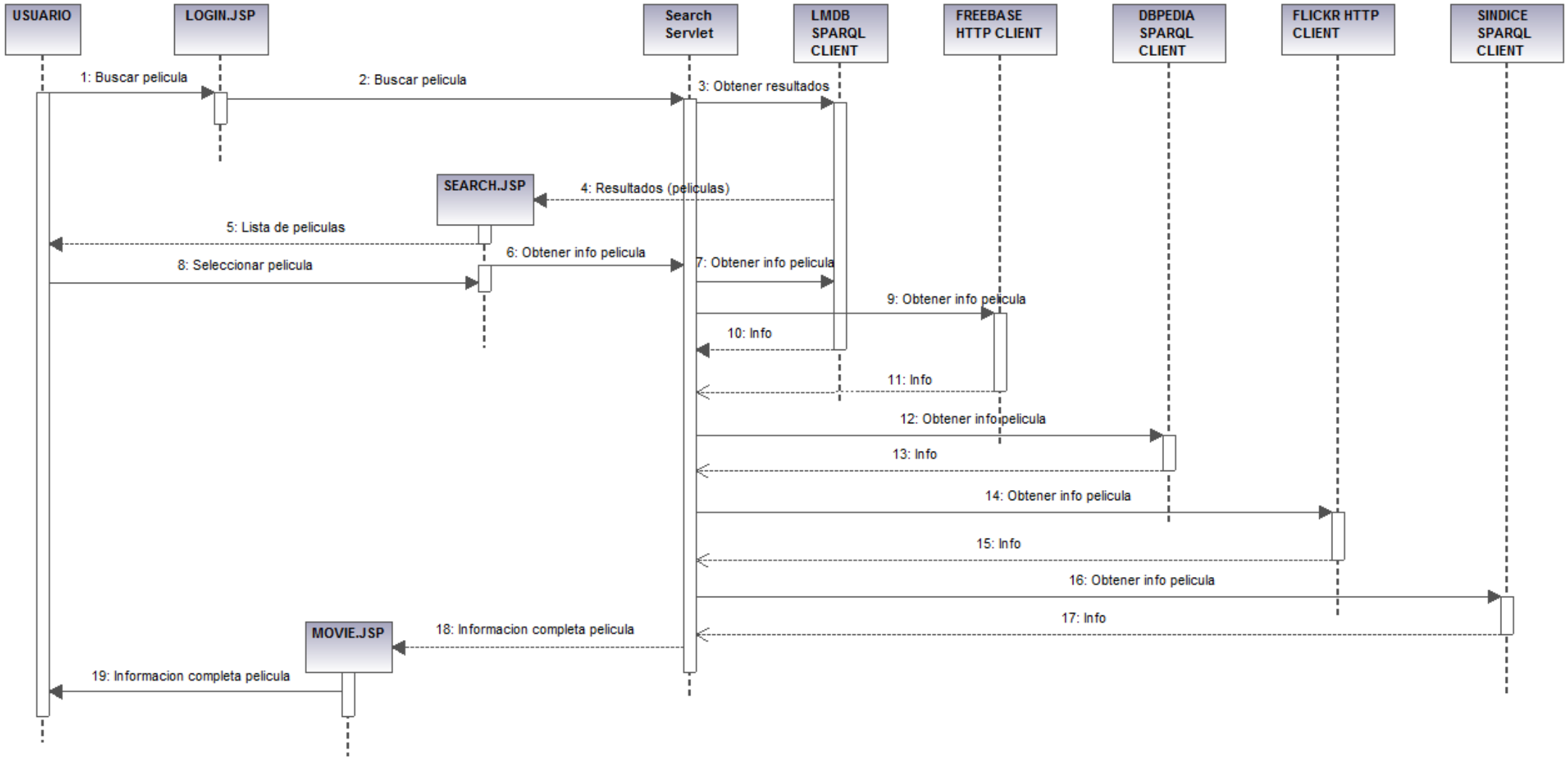

Figura 25: Diagrama de secuencia de la búsqueda de una película

4.4 Implementación

En esta fase de la descripción informática se describirán aspectos específicos del desarrollo de la aplicación, como son las herramientas utilizadas para el desarrollo, las decisiones tomadas durante la fase de implementación de la aplicación y los componentes de despliegue de la aplicación.

4.4.1 Herramientas utilizadas

Esta aplicación web utiliza varias tecnologías que están habitualmente relacionadas con el desarrollo de aplicaciones web y con el desarrollo de aplicaciones de la web de datos. A continuación se detallan el conjunto de herramientas utilizadas en la aplicación:

- El modelo de datos ha sido desarrollado en el lenguaje de programación orientado a objetos Java[62] utilizando como entorno de programación Eclipse IDE Juno [63]instalado con la maquina virtual de Java Development Kit 1.6

- Para la interfaz de usuario se ha utilizado principalmente la tecnología JSP[64] que permite crear páginas web dinámicas basadas en HTML[65] y en las que se pueden incluir fragmentos de código Java, lo cual proporciona una gran flexibilidad a la hora de desarrollar aplicaciones web. A esta tecnología se suman JQuery[66] como librería de Javascript[67], que permite la interacción con el usuario sin necesidad de hacer recargas de página completas, y hojas de estilo CSS[68] para el diseño grafico de la interfaz.

- Apache Tomcat[69] 7.0 es el servidor de aplicaciones seleccionado puesto que es de uso libre y cuenta con soporte de Servlets[70] y paginas JSP.

---

[62] http://www.java.com/
[63] http://www.eclipse.org/
[64] http://www.oracle.com/technetwork/java/javaee/jsp/index.html
[65] http://es.wikipedia.org/wiki/HTML
[66] http://jquery.com/
[67] http://es.wikipedia.org/wiki/JavaScript
[68] http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada
[69] http://tomcat.apache.org/
[70] http://es.wikipedia.org/wiki/Java_Servlet

- Para la comunicación con los puntos de acceso remotos SPARQL se ha utilizado el conjunto de herramientas proporcionadas por el marco de trabajo Apache Jena[71].

- La instancia de la base de datos se ha creado mediante la herramienta MySQL Workbench[72] 5.2 utilizando MySQL Connector/J 5.1.24 como librería que permite la comunicación entre el servidor de aplicaciones y la base de datos.

4.4.2 Decisiones de implementación

Durante la implementación de cualquier aplicación siempre es necesario tomar decisiones ajenas al diseño de la arquitectura. Estas son las decisiones de implementación más relevantes del proyecto:

- Enlace DBpedia - LMDB: Una de las decisiones más importantes tomadas durante la implementación está relacionada con la forma de enlazar los datos de las dos fuentes principales, DBpedia y LMDB. El problema radica en que para la mayoría de las películas de LMDB no existía un link directo a DBpedia, siendo necesario encontrar otra forma de enlazar las dos fuentes de datos. Analizando los resultados de las dos fuentes de datos se determina que el punto en común de las dos fuentes es un enlace a Freebase. Sin embargo, este enlace tiene diferente morfología en cada fuente. LMDB utiliza un enlace a Freebase mediante un identificador de la película denominado *guid*[73]*(globally unique id)*, este identificador esta deprecado actualmente y ha sido sustituido por el identificador *mid*[74]*(machine id)* que es el utilizado por DBpedia.

  Este sería un ejemplo de enlace a Freebase desde LMDB para la película 'El Resplandor':

  http://www.freebase.com/view/guid/9202a8c04000641f800000000046c3da

[71] http://jena.apache.org/
[72] http://www.mysql.com/products/workbench/
[73] http://wiki.freebase.com/wiki/Guid
[74] http://wiki.freebase.com/wiki/Mid

Este sería el enlace a Freebase de la misma película desde DBpedia:

http://rdf.freebase.com/ns/m.04fjzv

Como puede verse es necesario encontrar una correspondencia entre dichos identificadores para poder enlazar los dos conjuntos de datos. Esto se consigue mediante una petición HTTP a Freebase que contiene una consulta en MQL que nos devuelve el *mid* asociado a un *guid.* Esta sería la URL de la petición para la película 'El Resplandor':

'https://www.googleapis.com/freebase/v1/mqlread?query=

[{'id': '9202a8c04000641f800000000046c3da, 'mid': null}]

- Acceso a datos de Revyu: Uno de los puntos de acceso remotos que había sido seleccionado para la recuperación de información es Revyu. Sin embargo, no ha sido posible hacer peticiones directamente a su punto de acceso, puesto que la respuesta obtenida no estaba en el formato adecuado (*application/rdf+xml).* Se intento sin éxito contactar con los desarrolladores de Revyu y por último se decidió utilizar Sindice, un índice de datos en RDF que tiene indexados los datos de Revyu y cuyo punto de acceso funciona correctamente.

- Acceso a datos de flickr wrappr: Para la obtención de las fotografías de los usuarios relacionadas con la película se utiliza flickr wrappr, fuente de datos que no tiene punto de acceso SPARQL, por lo que en necesario recuperar la información mediante una petición HTTP. Esta petición está compuesta por una URL concatenada con el nombre de la entrada en Wikipedia (este dato se obtiene de un enlace de DBpedia) y a la que se le envía el formato en el que se espera la respuesta de la petición. Esta petición tiene por tanto el siguiente aspecto:

  http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Entrada_Wikipedia?format=rdf

- Usuario por defecto: Para permitir el uso de la aplicación para pruebas y poder utilizarla sin crear la instancia de la base de datos se ha definido un usuario por defecto cuyos credenciales son *usuario = 'admin' y clave = '1234'.*

- Uso de LIMIT 1000 en consultas SPARQL: Para hacer un buen uso de los puntos de acceso SPARQL es necesario definir un número máximo de resultados que se recuperan con una consulta.

- Acceso a datos de Freebase: Del mismo modo Freebase no posee tampoco un punto de acceso SPARQL por lo que sus datos también son accedidos mediante consultas vía HTTP. En este caso lo que se intenta recuperar es la imagen principal de la película para lo que es necesario concatenar a la URL de acceso una consulta en formato MQL[75]. En Freebase el identificador principal de la película es el elemento *mid* de cuya obtención ya he ha hablado anteriormente. La petición sería la siguiente:

  'https://www.googleapis.com/freebase/v1/mqlread?query=

  [{'mid': 'm.1234', 'common/topic/image': [{'id': null}]

  Mediante esta consulta se obtiene el identificador de la imagen principal asociada a la película cuyo *mid* es 'm.1234'. Una vez tenemos el identificador de la película es necesario realizar otra petición HTTP para recuperar la URL de la imagen:

  https://usercontent.googleapis.com/freebase/v1/image/id_imagen

4.4.3 Modelo de implementación

El siguiente diagrama de despliegue muestra los nodos y componentes involucrados en el despliegue de la aplicación. Como puede verse el nodo central es el servidor de aplicaciones que recibe las peticiones HTTP del navegador web utilizado por el usuario, gestiona estas peticiones y establece la comunicación con la base de datos y con los nodos externos a la aplicación web para recuperar información sobre las películas.

[75] http://wiki.freebase.com/wiki/MQL

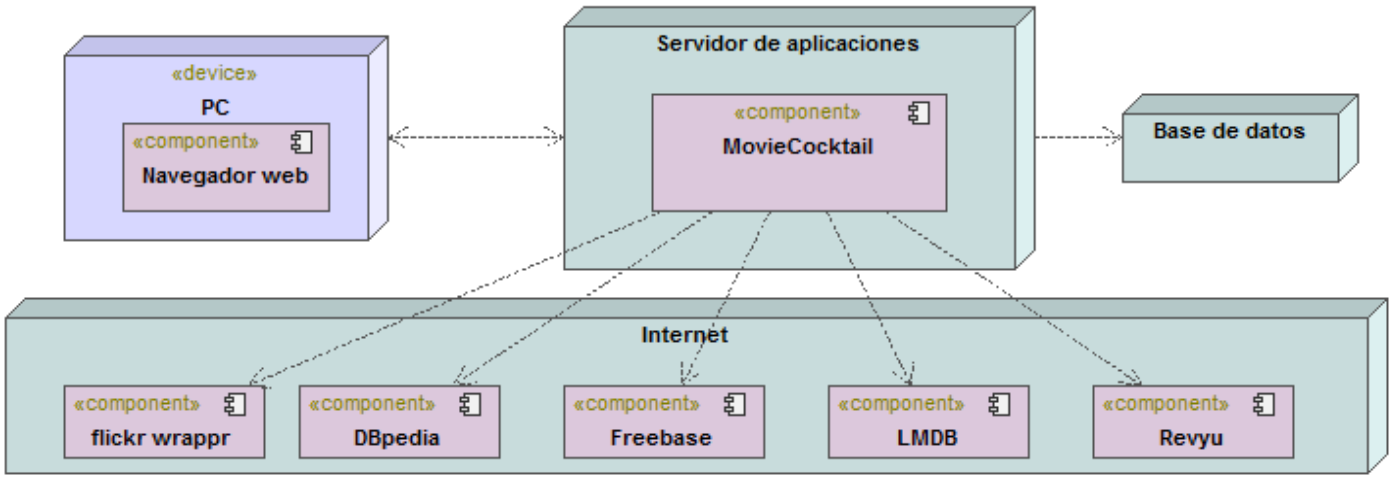

Figura 26: Diagrama de despliegue

En la siguiente figura se muestran todos los componentes externos a los que accede la aplicación y de los que se extrae información de las películas. Cada uno de estos nodos tiene dentro de la aplicación 'MovieCocktail' una clase que gestiona la comunicación con dicho nodo. Como puede verse en el diagrama las consultas a los nodos LMDB, Sindice y DBpedia se realizan mediante consultas SPARQL de la librería Jena. Las consultas a flickr wrappr y a Freebase se realizan mediante peticiones HTTP y además para Freebase se incluye en la petición una consulta en MQL.

El resultado obtenido de cada uno de los nodos no es homogéneo. Para las consultas a LMDB, Sindice y DBpedia se obtiene un objeto *ResultSet* [76]de la librería Jena que contiene los nodos RDF que satisfacen la consulta SPARQL de cada una de las fuentes. Por otro lado, el resultado que se obtiene de Freebase es un objeto JSON[77] de cuyo resultado se hace una lectura mediante la librería json-simple[78]. Por último, el resultado que se obtiene de flickr wrappr es un vertido de datos en RDF, para cuya lectura se utiliza de nuevo la librería Jena para la creación de un Modelo[79] de datos sobre el que se puede iterar.

[76] http://jena.apache.org/documentation/javadoc/arq/com/hp/hpl/jena/query/ResultSet.html
[77] http://es.wikipedia.org/wiki/JSON
[78] https://code.google.com/p/json-simple/
[79] http://jena.apache.org/documentation/javadoc/jena/com/hp/hpl/jena/rdf/model/Model.html

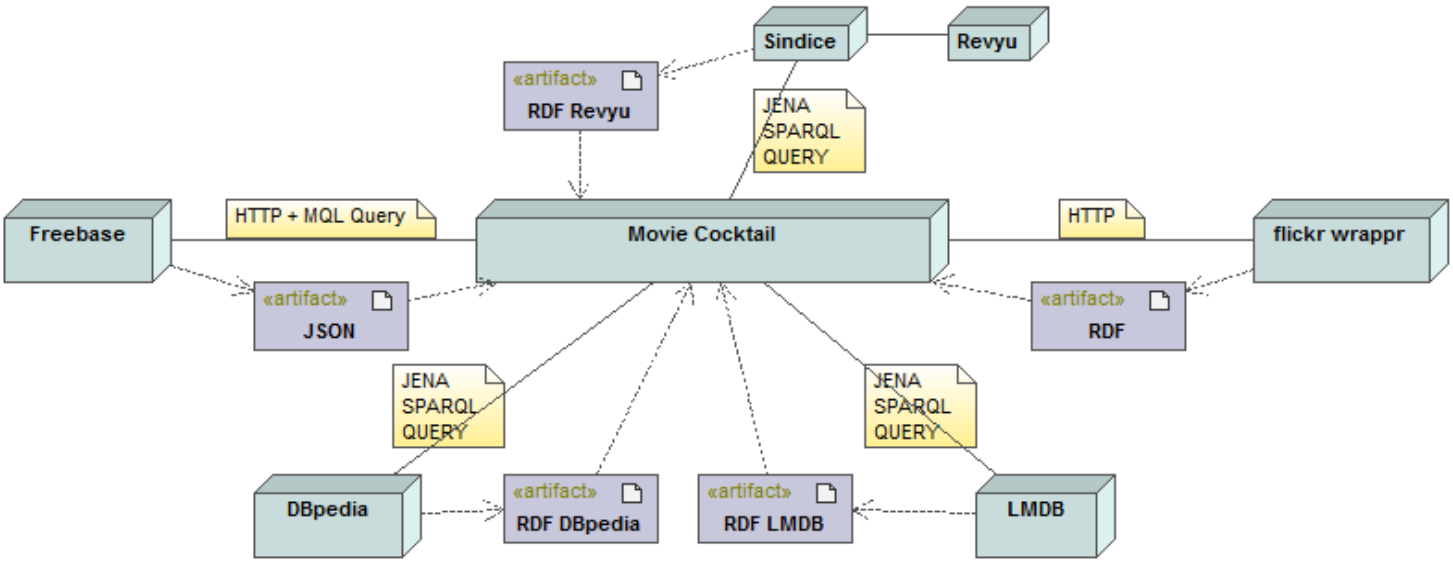

Figura 27: Diagrama de comunicación con los puntos de acceso

4.5 Pruebas

Durante la fase de pruebas se ha seleccionado un conjunto de datos lo suficientemente representativo para la verificación y validación de los requisitos funcionales y no funcionales de la aplicación.

4.5.1 Pruebas funcionales de acceso a la aplicación

Tabla 11: Pruebas funcionales de acceso a la aplicación

Prueba	Requisito que verifica	Descripción	Resultado
P1	R1	Autenticación de un usuario previamente registrado mediante sus credenciales.	✓
P2	R1	Autenticación de un usuario con identificador erróneo.	✓
P3	R1	Autenticación de un usuario con clave errónea.	✓
P4	R2	Registro de un nuevo usuario en la aplicación.	✓
P5	R2	Intento de registro de usuario sin rellenar todos los campos.	✓

P6	R3	Cierre de sesión para un usuario previamente autenticado.	✓
P7	R4	Acceso a una página privada de la aplicación sin estar autenticado.	✓

4.5.2 Pruebas funcionales de búsqueda de películas

Tabla 12: Pruebas funcionales de búsqueda de películas

Prueba	**Requisito que verifica**	**Descripción**	**Resultado**
P8	R5	Búsqueda de 'Forrest Gump' -> Contenido recuperado correctamente de Freebase, LMDB, DBpedia, Revyu y flickr wrappr	✓
P9	R5, R8	Búsqueda de 'Agora' -> Contenido recuperado correctamente de Freebase, LMDB y DBpedia. Contenido en Revyu y flickr wrappr no encontrado	✓
P10	R5, R8	Búsqueda de 'Broken Flowers' -> Contenido recuperado correctamente de LMDB, DBpedia y Revyu. Contenido en Freebase no encontrado y en flickr wrappr no relacionado con el contenido.	×
P11	R6	Búsqueda de 'forrest gump' -> Contenido recuperado correctamente de Freebase, LMDB, DBpedia, Revyu y flickr wrappr	✓
P12	R7	Búsqueda de 'King Kong' -> obtenido más de un resultado	✓
P13	R9	Conexión no establecida con alguna de las fuentes de datos.	✓
P14	R10	Búsqueda de 'Django unchained' -> ningún resultado obtenido	✓

4.5.3 Pruebas no funcionales

Tabla 13: Pruebas no funcionales

Prueba	Requisito que verifica	Descripción	Resultado
P15	R11	Navegación a través de todas las páginas de la aplicación.	✓
P16	R12	Navegación cómoda a través de todas las páginas de la aplicación.	✓
P17	R13	Navegación a través de todas las páginas de la aplicación en la última versión de Mozilla Firefox.	✓
P18	R13	Navegación a través de todas las páginas de la aplicación en la última versión de Google Chrome.	✓
P19	R13	Navegación a través de todas las páginas de la aplicación en la última versión de Internet Explorer.	✓
P20	R14	La aplicación será capaz de realizar las operaciones requeridas con un tiempo de espera corto.	✓
P21	R15	La funcionalidad de la aplicación será fácilmente extensible en la medida necesaria.	✓
P22	R16	La aplicación mostrará toda la información recopilada en un formato consistente y accesible para el usuario.	✓
P23	R17	La aplicación mostrara las atribuciones correspondientes a las diferentes fuentes de datos utilizadas, de acuerdo con las licencias de uso de las mismas.	✓
P24	R18	El buscador estará presente durante toda la navegación para permitir al usuario acceder en cualquier momento a la funcionalidad principal.	✓

Como podemos comprobar por la serie de pruebas realizadas, los problemas encontrados en la aplicación están relacionados con la sensibilidad del contenido encontrado en cada una de las fuentes de datos, es decir, si una fuente de datos contiene resultados no relacionados con la búsqueda se estará mostrando al usuario contenido inconsistente. Este problema se ha encontrado sobre todo en la fuente de datos flickr wrappr, que se basa en contenido añadido por los usuarios. Este modo de generación de contenido debe estar muy asociado a la verificación de la información para poder confiar en la validez del recurso que se consulta.

Para el resto de pruebas realizadas los resultados han sido satisfactorios, no teniendo problemas ni en la representación de la información ni en el contenido de los resultados.

5. Conclusiones

5.1 Logros principales

El principal logro conseguido ha sido la creación de una aplicación web a partir de fuentes de datos enlazados Incluso aunque al principio se hayan presentado problemas en los enlaces de unos datos con otros, finalmente se ha conseguido encontrar una correspondencia entre estos datos, permitiendo un uso optimizado de la web de datos.

La complejidad de la aplicación no es muy elevada y requiere una cantidad de recursos mínimos ya que se accede a datos externos y aunque esto presenta algunas desventajas en la manipulación de la información también presenta grandes ventajas de escalabilidad y mantenimiento.

Otro de los objetivos fundamentales conseguido ha sido el encontrar viabilidad en la utilización de fuentes de datos de externas y haber podido encontrar información suficiente sobre las fuentes de datos y el acceso a los mismos.

Por otro lado, se ha conseguido utilizar tecnologías de la web semántica y realizar una integración de datos creando un valor agregado interesante para el usuario final y al haber utilizado el patrón de diseño Modelo – Vista – Controlado se ha conseguido crear software escalable, permitiendo desarrollar además una interfaz grafica usable e intuitiva

5.2 Problemas principales encontrados

Durante la realización de esta fase el problema principal se encuentra en el acceso libre a los datos, que en la mayoría de las web actuales están protegidos. Por ese motivo se decidió utilizar datos enlazados de acceso libre y se encontraron los problemas típicos asociados a los datos de libre acceso o incluso creados por usuario, que son el mantenimiento de los datos y la validez de los datos recuperados.

Por otro lado al consultar fuentes externas, es necesario adecuarse a los requisitos de los puntos de acceso y habría sido útil contar con mas soporte de los equipos de mantenimiento de algunas fuentes de datos.

5.3 Líneas futuras

A continuación se plantea una serie de posibles trabajos futuros para la mejora de la aplicación así como para la extensión de su funcionalidad:

- Permitir a los usuarios crear listas de películas favoritas.
- Permitir a los usuarios crear listas de películas para ver.
- Funcionalidades asociadas a las aplicaciones de la Web 2.0 como puede ser la capacidad de compartir las listas de películas.
- Conexión con una fuente de datos de eventos para mostrar al usuario eventos relacionados con las películas de sus listas.
- Sistema para determinar películas parecidas a una dada determinado por votaciones de usuarios.
- Conexión con redes sociales externas para poder compartir la información deseada por los usuarios.
- Capacidad de realizar búsquedas en varios idiomas y mostrar información en varios idiomas.
- Capacidad de realizar búsquedas que no contengan el nombre completo de la película, siendo necesario para esto crear un punto de acceso local para no entrar en conflicto con las normas de uso responsable de los puntos de acceso remotos.

6. Bibliografía

[1] Grigoris Antoniou, Frank van Harmelen

"A Semantic Web Primer"

Editorial MIT Press 2008

[2] Dean Allemang, Jim Hendler

"Semantic Web for the working ontologist- Effective Modeling in RDFS and OWL"

Editorial Morgan Kaufman Publishers

[3] Jeffrey T. Pollock

"Semantic Web for Dummies"

Editorial Wiley 2009.

[4] Karin K. Breitman, Marco Antonio Casanova, Walter Truszkowski.

"Semantic Web. Concepts, Technologies and Applications"

Editorial Springer-Verlag 2007

[5] John Hebeler, Mathew Fisher, Ryan Blace, Andrew Perez-Lopez

"Semantic Web Programming"

Wiley Publishing

[6] Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph

"Foundations of Semantic Web Technologies"

Chapman & Hall/ CRC

[7] Christian Bizer, Tom Heath, Tim Berners-Lee

"Linked Data - The Story So Far"

edited b: T. Heath, M. Hepp, C. Bizer

http://linkeddata.org/

www.interacciones.com.ar/category/web-semantica

http://www.w3c.es/divulgacion/guiasbreves/websemantica

http://www.w3c.es/divulgacion/guiasbreves/LinkedData

http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients

http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/SemanticWebSearchEngines

http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

http://swa.cefriel.it

http://www.w3.org/DesignIssues/LinkedData.html

http://www.w3.org/wiki/SparqlEndpoints

http://opentox.org/data/documents/development/RDF%20files/JavaOnly/query-reasoning-with-jena-and-sparql

http://jena.apache.org/tutorials/sparql.html

http://topquadrantblog.blogspot.com.es/2010/02/how-to-get-data-from-sparql-endpoints.html

http://linkeddatabook.com/editions/1.0/#htoc1

Apéndices

Apéndice I

Manual de instalación y despliegue de la aplicación web

Para la instalación y despliegue de la aplicación web será necesaria la instalación previa del servidor de aplicaciones y de la base de datos. A continuación se detallan las sencillas instrucciones a seguir:

Instalación del software necesario

Sistema operativo utilizado para la instalación: Windows XP 32bits y Windows 7.

1. Verificar que se tiene instalada la maquina virtual de Java. En el disco de instalación se incluye el *Java Development Kit* (JDK) jdk-7u21-windows-i586 en el directorio 'Java'. También puede descargarse una versión compatible con el equipo de la página web oficial de Oracle[80].

2. Definir como variable de entorno de sistema JAVA_HOME cuyo valor debe ser la ruta del JDK (por ejemplo en nuestro caso es C:\Program Files\Java\jdk1.7.0_21)

[80] http://www.oracle.com/technetwork/es/java/javase/downloads/index.html

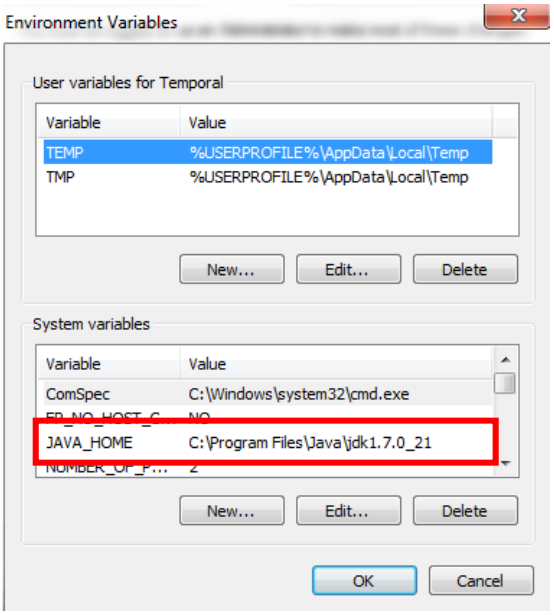

Figura 28:Variables de entorno del servidor

3. Instalar el servidor de aplicaciones *Apache Tomcat*[81]**.** En el disco de instalación se incluye el archivo apache-tomcat-7.0.35-windows-x86 contenido en el directorio 'Servers'. Para la instalación sólo es necesario descomprimir el contenido de este archivo en una carpeta de nuestro equipo.

Con esto ya tenemos disponible todo lo necesario para ejecutar el despliegue de la aplicación salvo la base de datos. Si el objetivo es probar la aplicación no será necesario crear la instancia de la base de datos puesto que se ha definido un usuario y una clave por defecto cuyos credenciales son: Usuario: '*admin*' y Contraseña: '*1234*'.

Si por el contrario se desea crear una instancia de la base de datos, para poder realizar registros de usuario en la aplicación, su instalación se detalla a continuación:

4. Instalar *MySQL Workbench* descargándolo de la página web oficial de *MySQL*[82] o utilizando el ejecutable mysql-installer-community-5.6.10.1 contenido en el disco de instalación en la carpeta 'Data Base'. Ejecutar el archivo y seguir los pasos del asistente de instalación, seleccionando las siguientes opciones:

[81] http://tomcat.apache.org/
[82] http://www.mysql.com/downloads/

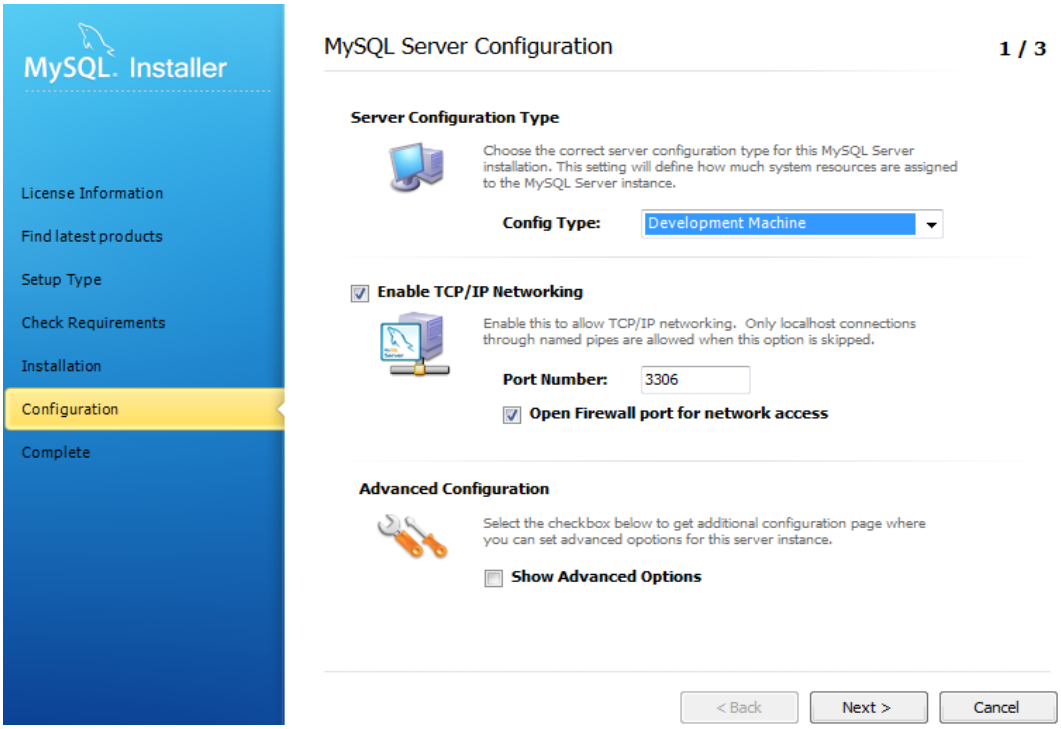

Figura 29: Configuración de la base de datos 1

A continuación seleccionaremos la clave de acceso a la instancia de la base de datos, que en nuestro caso debe ser '1234':

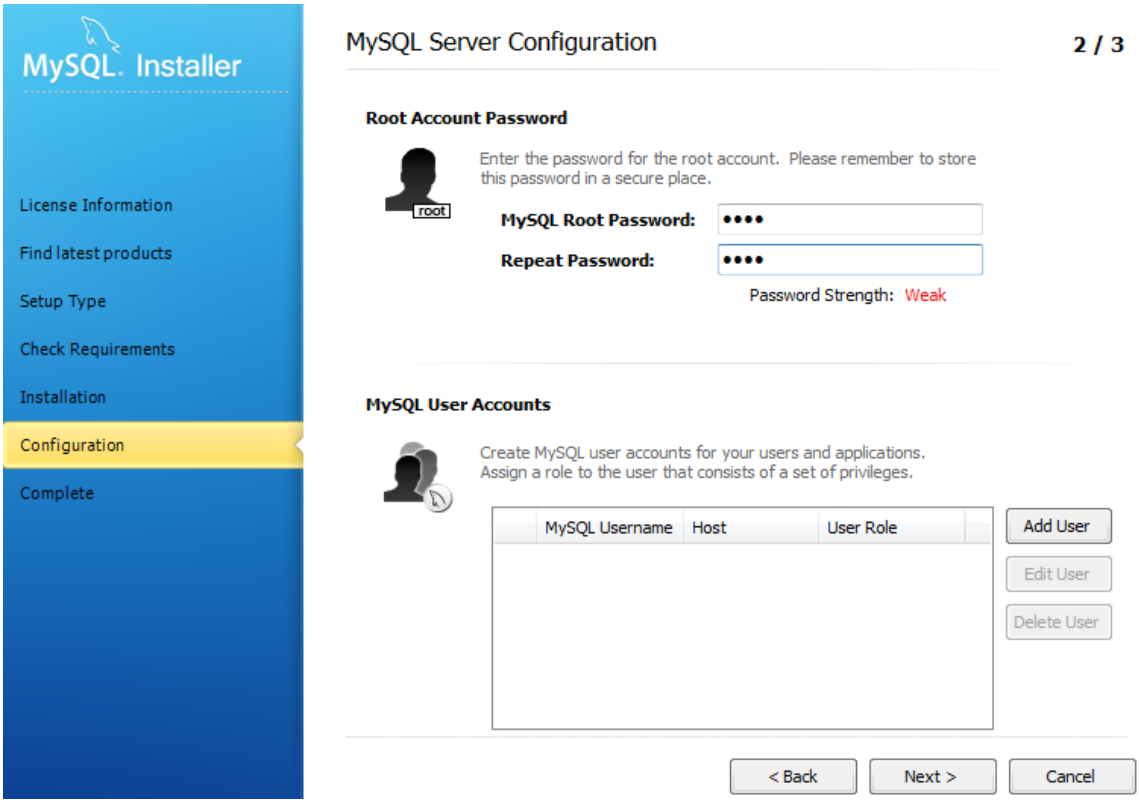

Figura 30: Configuración de la base de datos 2

Para finalizar se dejara la configuración por defecto:

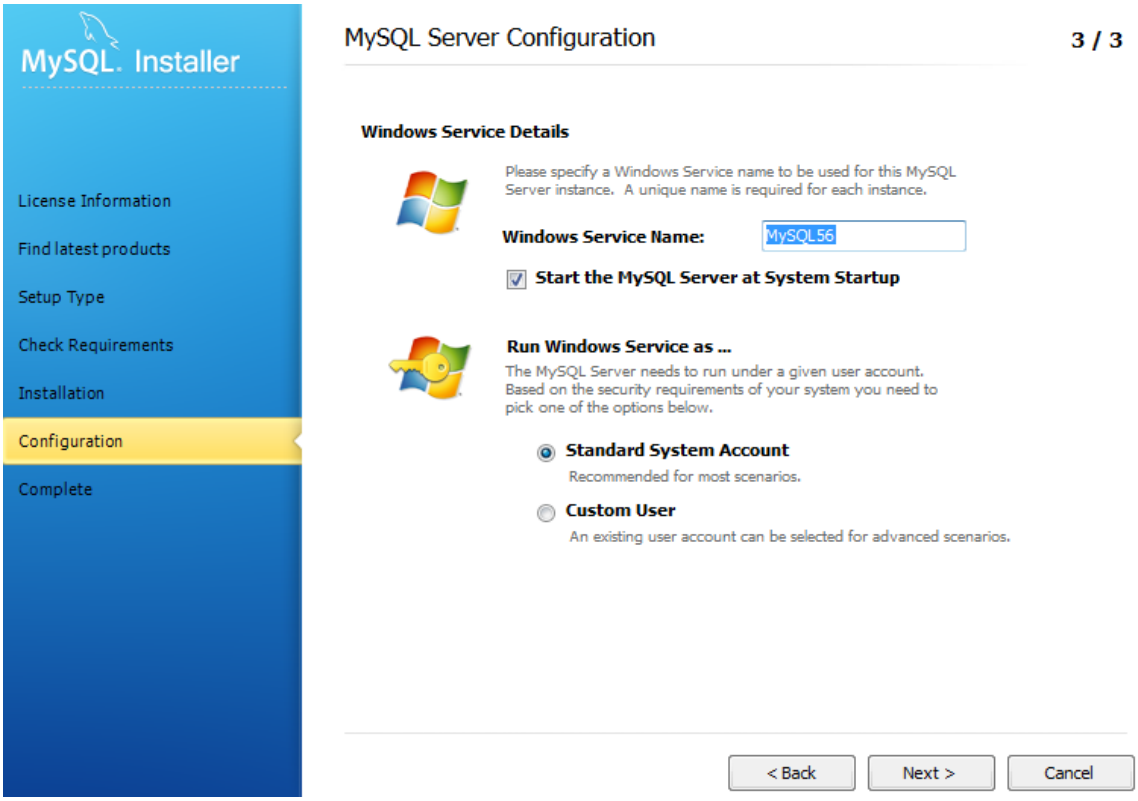

Figura 31: Configuración de la base de datos 3

5. Para conectar el Servidor con la Base de datos será necesario copiar la librería mysql-connector-java-5.1.24-bin contenida en la carpeta 'Data Base' del disco de instalación al directorio lib de Apache (apache-tomcat-7.0.35\lib)

6. A continuación será necesario importar el esquema de la base de datos contenido en la carpeta 'Data Base' moviecocktaildb_usuario.sql en MySQL Workbench:

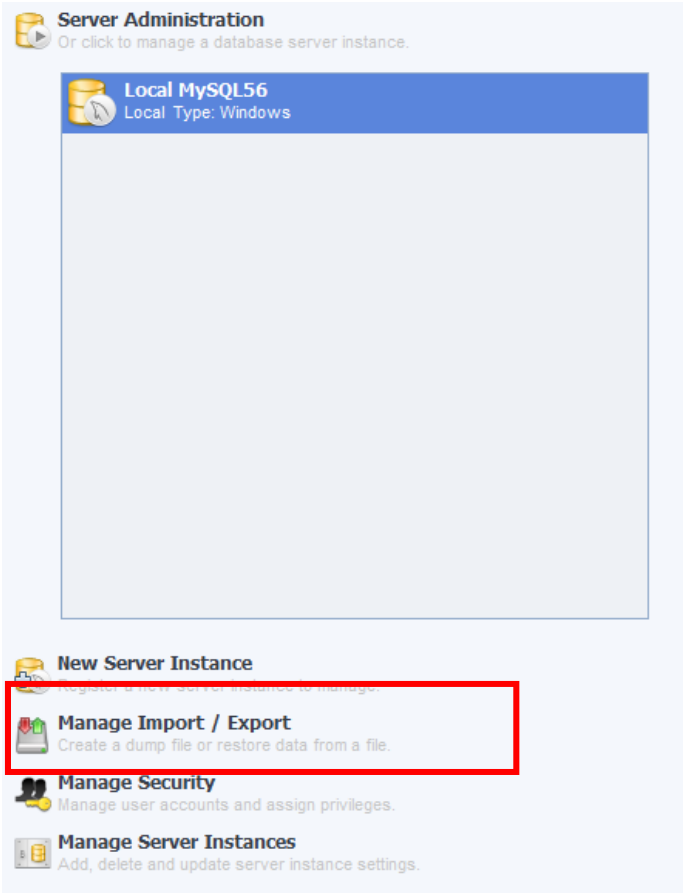

Figura 32: Importar esquema de la base de datos

Despliegue de la aplicación

7. El despliegue de aplicaciones con Apache Tomcat es muy sencillo, sólo es necesario colocar en la carpeta webapps del directorio raíz de Apache (apache-tomcat-7.0.35\webapps) el archivo MovieCocktail.war contenido en el disco de instalación en el directorio 'Web Application'. Con esto el servidor desplegará la aplicación al iniciarse.

8. Para iniciar el servidor Apache será necesario ejecutar el archivo apache-tomcat-7.0.35\bin\startup.bat. Del mismo modo para detener el servidor se ejecutará el archivo apache-tomcat-7.0.35\bin\shutdown.bat.

9. Con esto ya tenemos desplegada la aplicación en nuestro servidor local por lo que podemos acceder a la dirección http://localhost:8080/MovieCocktail/home y comenzar a usar la aplicación.