

Proyecto de Fin de Grado

Manaject

Un sistema de recomendación basado en ontologías que empareja personas con proyectos.

Autor

Aitor Corrales Rodríguez

Director

Jesús Bermúdez de Andrés



Resumen

Los sistemas de recomendación son un recurso muy frecuente empleado para recomendar a una persona aquellos productos o servicios más adecuados o que mejor satisfagan sus necesidades. En este proyecto se pretende construir un componente de software que permita a los usuarios aportar descripciones de proyectos así como determinar personas adecuadas para participar en los mismos, empleando un sistema de recomendación basado en las necesidades de un proyecto y en las competencias de las personas, haciendo uso de tecnologías de web semántica.

En **Manaject** se prioriza el grado de interés que solicita el proyecto en determinadas competencias, de forma que la recomendación del personal sea lo más ajustada posible a las necesidades del mismo. La persona que esté describiendo el proyecto que está interesada en realizar, seleccionará una serie de aptitudes que se requieren para ese proyecto, y en cada una de ellas, el grado de interés que tiene en la misma.

Palabras clave: Sistemas de recomendación, Tecnologías de la web semántica, Gestión de recursos humanos.

Índice de contenido

Capítulo 1. Introducción.....	1
1.1 Proyecto.....	1
1.2 Organización del documento.....	3
1.3 Análisis de viabilidad.....	3
Capítulo 2. Alcance del proyecto.....	5
2.1 Objetivo.....	5
2.2 Alcance definido.....	5
2.3 Funcionalidades futuras.....	6
2.3.1 Interfaz web.....	6
2.3.2 Recogida de datos.....	7
2.3.3 Seguridad.....	7
2.3.4 Implantación de ontologías.....	7
2.3.5 Sistema de recomendación.....	8
2.4 Tiempo dedicado a las distintas actividades.....	9
2.5 Riesgos del proyecto.....	10
Capítulo 3. Análisis de antecedentes.....	13
3.1 Contexto tecnológico.....	13
3.1.1 La Web Semántica.....	13
3.1.1.1 RDF.....	13
3.1.1.2 OWL.....	14
3.1.1.3 SPARQL.....	14
3.1.2 Ontologías.....	15
3.1.2.1 Modelo conceptual.....	16
3.1.2.1.1 Ontologías de aplicación.....	18
3.1.2.1.2 Ontologías de dominio.....	20
3.2 Trabajos relacionados.....	21
3.2.1 The Impact of Semantic Web Technologies on Job Recruitment Processes.....	22
3.2.2 Ontology-based Skills Management.....	23
3.2.3 Ontology Development For Human Resource Management.....	24
3.2.4 An ontology-based intelligent system for recruitment.....	25
3.2.5 Matching People and Jobs: A Bilateral Recommendation.....	26
3.2.6 A Recommender System for Job Seeking and Recruiting Website.....	27
3.3 Relación de trabajos con Manaject.....	28
Capítulo 4. Funcionalidades y características del sistema.....	31
4.1 Registro de usuarios.....	31
4.1.1 Captura de perfiles.....	31
4.1.2 Iniciar sesión.....	32
4.1.3 Inicio.....	34
4.1.3.1 Perfil Personal.....	34
4.1.3.2 Generar proyecto.....	37
4.2 Núcleo del sistema.....	38
4.2.1 Procesamiento de datos.....	38
4.2.1.1 Paso 1, estudio y selección de ontología.....	38
4.2.1.2 Paso 2, formato y envío de los datos.....	39
4.2.1.3 Paso 3, procesamiento de los datos.....	40
4.2.1.3.1 Algoritmos empleados para el sistema de recomendación.....	43
Sumatorio.....	45
Coeficiente de similitud de Jaccard.....	45

Similitud del coseno.....	46
4.3 Tecnologías empleadas.....	48
4.3.1 Cliente.....	48
4.3.2 Servidor.....	48
4.3.3 Base de datos.....	49
4.3.4 Núcleo del sistema.....	49
Capítulo 5.Conclusiones.....	51
5.1 Consideraciones.....	52
Capítulo 6.Bibliografía.....	55
Anexo	57

Índice de tablas

Tabla 1: Tiempo dedicado al aprendizaje.....	9
Tabla 2: Tiempo dedicada a la gestión del proyecto.....	9
Tabla 3: Tiempo dedicado a la aplicación Manaject.....	10
Tabla 4: Situaciones de riesgo del proyecto.....	11
Tabla 5: Competencias del perfil de prueba1.....	36
Tabla 6: Valores literales enviados al servidor del proyecto de prueba.....	40
Tabla 7: Competencias de proyecto de prueba.....	44
Tabla 8: competencias del usuario manaject@gmail.com.....	57
Tabla 9: competencias del usuario prueba1@gmail.com.....	58
Tabla 10: competencias del usuario natalie@gmail.com.....	58
Tabla 11: competencias del usuario user1@gmail.com.....	58
Tabla 12: competencias del usuario antonio@gmail.com.....	59
Tabla 13: competencias del usuario joseteRodriguez@gmail.com.....	59
Tabla 14: competencias del usuario josu@gmail.com.....	60
Tabla 15: competencias del usuario ibanm@gmail.com.....	60
Tabla 16: competencias del usuario diego@gmail.com.....	61
Tabla 17: competencias del usuario john@gmail.com.....	61

Índice de figuras

Figura 1: Recomendaciones basadas en contenido.....	8
Figura 2: Grafo RDF.....	14
Figura 3: Consulta SPARQL que devuelve el recurso con el nombre del literal.....	15
Figura 4: Red de ontologías.....	17
Figura 5: Conceptos principales y relaciones.....	18
Figura 6: Diagrama de clases de Job Offer Ontology.....	19
Figura 7: Diagrama de clases de Job Seeker Ontology.....	20
Figura 8: Diagrama de ejemplo de la ontología de dominio Education Ontology.....	21
Figura 9: Registro.....	31
Figura 10: Inicio de sesión.....	32
Figura 11: Tripleta para insertar identificador de sesión.....	33
Figura 12: Proceso de inicio de sesión.....	33
Figura 13: Página de inicio.....	34
Figura 14: Perfil personal.....	34
Figura 15: Usuario prueba1 antes de introducir sus competencias.....	35
Figura 16: Usuario prueba1 tras introducir sus competencias.....	36
Figura 17: Perfil del proyecto.....	37
Figura 18: Apartado numérico en las competencias.....	38
Figura 19: Resultado de Project Seekers con alguna de las competencias del proyecto.....	43
Figura 20: Personas recomendadas por la técnica de suma.....	45
Figura 21: Personas recomendadas por la técnica del índice de Jaccard.....	46
Figura 22: Personas recomendadas por la técnica de similitud del coseno.....	47

Capítulo 1. Introducción

1.1 Proyecto

Con el paso del tiempo la tecnología ha ido evolucionando, y con esta, se han ido generando nuevos productos. Hoy en día, nos encontramos con que existen millones y millones de productos de entre los cuales, la gran mayoría no conocemos o, de conocerlos, no somos conscientes si son o no de nuestro interés ya que desconocemos sus características. Con el fin de cubrir esta necesidad, que es cada vez más evidente debido a la ingente cantidad de *items* nuevos que salen a diario, nacieron los sistemas de recomendación.

Los sistemas de recomendación son una subclase de sistemas de filtrado de la información, orientados a proporcionar una serie de *items* personalizados para el usuario. Estos sistemas han tenido una gran aceptación y hoy en día se utilizan en gran parte de las páginas web enfocadas al e-commerce (Amazon y eBay son un buen ejemplo de estas), así como en redes sociales (Facebook, Google+, ...), *video-sharing websites* o páginas para compartir vídeos (YouTube), páginas con listas ordenadas de películas y/o series, etc.

El primer sistema de recomendación con fines comerciales, Tapestry [1], fue diseñado para recomendar documentos de grupos de noticias a una colección de usuarios, con el fin de prevenir que los usuarios quedasen inundados por una enorme cantidad de documentos que no son de su interés. Aquel introdujo el término *collaborative filtering* o filtro colaborativo que es una técnica de filtrado de los sistemas de recomendación la cual analiza las interacciones de los usuarios con los distintos *items* del sistema.

Con esta, hacen tres técnicas de filtrado empleadas en los sistemas de recomendación: *content-based filtering* o filtrado basado en contenido (basado en los atributos del perfil del usuario en cuestión), *collaborative filtering* y *hybrid filtering* o filtrado híbrido (una mezcla de las dos anteriores).

La Web Semántica es una extensión de la web actual en la cual la información tiene significado no solo para el ser humano, sino también para los agentes software, permitiendo que las personas y los agentes inteligentes trabajen de forma cooperativa.

En este proyecto se pretende desarrollar un sistema de recomendación empleando

la técnica de filtrado basado en contenido, para personas interesadas en realizar proyectos de cualquier ámbito. Para ello se hace uso de tecnologías de la Web Semántica. Una vez se ha realizado la descripción de un proyecto, el sistema devuelve a la persona que ha realizado el proyecto una lista de las i personas, donde i es el número de personas solicitadas, cuyas competencias son más afines a las competencias exigidas por el mismo.

Las ventajas de que el sistema de recomendación se base en tecnologías de Web Semántica se harán más evidentes con el paso del tiempo. Hoy en día, si una persona quiere enviar su curriculum a diferentes portales de ofertas de empleo, como cada uno tiene su forma de registrar la información, esa persona tendrá que repetirlo todas las veces que quiera realizar dichas solicitudes; mientras que, si los portales web fuesen semánticos, con el vocabulario adecuado, una persona solamente tendría que tener su curriculum hecho una vez y este serviría para todos los portales que compartiesen dicho vocabulario. De la misma manera, si lo que se buscan son personas para realizar un proyecto, con hacer la descripción de un proyecto una vez, sería suficiente para todas las aplicaciones que compartiesen un mismo vocabulario.

Para que este componente de software sea efectivo, es necesario establecer las especificaciones del proyecto correctamente. Antes de realizar la descripción del proyecto, la persona que va a realizar este cometido debería definir claramente cuales son las aptitudes, conocimientos y habilidades necesarias, así como el grado de importancia de estas, para poder realizar las tareas que hagan posible la realización del proyecto. Una descripción realista del proyecto conduce a un filtrado más eficaz y con ello, a un resultado de mayor calidad y de personas mejor cualificadas para los requisitos impuestos.

Como ya se ha mencionado, este proyecto ha tenido una base semántica sobre la cual se han llevado a cabo todas las operaciones, tanto de consulta como de actualización. La decisión de realizarlo bajo el amparo de las tecnologías semánticas fue tomada debido a las ventajas que esto proporciona, las cuales se mencionan más adelante en el apartado de *Análisis de antecedentes*.

Se hace distinción entre dos roles de usuarios en el sistema, *Project Head* y *Project Seeker*, entendiendo por usuario a cualquier persona que se haya registrado en la aplicación. *Project Head* es el rol que toma la persona que introduce la descripción de un nuevo proyecto sobre dicho proyecto; y *Project Seeker* es un rol que poseen todos los usuarios. Bajo este rol, el usuario será recomendado a otros *Project Head* que hayan introducido la descripción de un proyecto afín a aquel.

1.2 Organización del documento

Este documento es la memoria del proyecto **Manaject** que se ha llevado a cabo por Aitor Corrales Rodríguez con la dirección de Jesús Bermúdez de Andrés y va a seguir la siguiente estructura:

1. Introducción y contextualización del proyecto donde se explica que son los sistemas de recomendación.
2. Alcance del proyecto, donde se define el objetivo del mismo, el alcance definido y las ampliaciones futuras para el proyecto.
3. Análisis de antecedentes, donde se muestra el estudio realizado para asentar unas bases de conocimiento necesarias para la realización del proyecto.
4. Funcionalidades y características del sistema donde se recoge el diseño y la implementación llevadas a cabo en el proyecto.
5. Conclusiones del proyecto y un apartado de consideraciones personales del autor del proyecto.

1.3 Análisis de viabilidad

El proyecto que se ha planteado llevar a cabo tiene un tiempo de realización estimado de 300 horas. Este proyecto pretende ser realizado en el cuatrimestre que abarca de septiembre a enero, y será el foco principal de trabajo del autor. **Manaject** va a hacer uso de diversas tecnologías que no han sido impartidas por ninguna asignatura del grado y entre las cuales, algunas aún están en estado de desarrollo, por lo que la curva de aprendizaje será muy elevada.

Aun y todo, se considerará que la realización del proyecto es viable ya que se va a disponer del tiempo necesario para llevarlo a cabo, así como de los recursos de información necesarios para posibilitar el aprendizaje, tanto de las tecnologías como de las técnicas necesarias para llegar a la fase de cierre del proyecto de forma satisfactoria.

Capítulo 2. Alcance del proyecto

En este capítulo aparece detallado el alcance del proyecto, de entre los objetivos iniciales, cuáles se han podido realizar y cuáles no, así como el objetivo principal de la aplicación **Manaject**.

2.1 Objetivo

Los sistemas de recomendación son un recurso muy frecuente empleado para recomendar a una persona aquellos productos o servicios más adecuados o que mejor satisfagan sus necesidades. En este proyecto se pretende construir un componente de software que permita a los usuarios describir las competencias necesarias de un proyecto que pretendan realizar, así como incluir en su perfil personal las competencias que poseen, empleando un sistema de recomendación basado en contenido que toma como información para realizar el filtrado las necesidades de un proyecto y las competencias de las personas. Las personas serán recomendadas por el sistema para formar parte del proyecto en base a ciertos parámetros impuestos por el jefe de proyecto (la persona que lo haya propuesto) utilizando tecnologías semánticas.

El sistema, independientemente de la seleccionada para la aplicación, cargará la ontología sobre la interfaz web automáticamente, de forma que aunque se decida cambiar de ontología, no suponga ningún coste adicional. Gracias a esto, se podrán realizar cambios sobre la ontología directamente ya que la ontología en sí constituirá un módulo aparte del propio núcleo de la aplicación y de la interfaz web.

Los datos del perfil personal de los usuarios que utilizará el sistema de recomendación se recogerán de diversas fuentes. En primera instancia, la información se buscará en el perfil de la persona en la aplicación, y después, se buscarán en diversas fuentes de información como redes profesionales (LinkedIn) o redes sociales (Facebook).

2.2 Alcance definido

El alcance del proyecto está orientado a ofrecer al usuario la funcionalidad básica para poder ofertar un proyecto siendo el jefe de proyecto (*Project Head*), así como poder ser recomendado a ciertos proyectos con los que sea compatible (*Project Seeker*).

Manaject cuenta con una interfaz web para que cualquier persona interesada pueda registrarse e iniciar sesión. Una vez iniciada la sesión, esa persona podrá rellenar su perfil personal así como incluir la descripción de un nuevo proyecto para que el sistema busque el emparejamiento óptimo con aquellas personas con características afines. Tanto en el perfil personal del usuario como en la descripción de los proyectos, aparecen grupos de competencias a seleccionar y serán mediante las cuales el sistema hará el filtrado. Los elementos que componen el alcance de este proyecto son los siguientes:

1. Una interfaz web para la interacción con el usuario.
2. La posibilidad de incluir datos personales en la base de datos mediante la interfaz así como las competencias profesionales, las habilidades, la formación académica y el conocimiento de ciertos idiomas.
3. La posibilidad de incluir la descripción de un nuevo proyecto. Las competencias que se desplegarán a la hora de rellenar el perfil personal del usuario y a la hora de introducir un nuevo perfil de proyecto serán las mismas.
4. Dos funciones de valoración de la recomendación reconocidas, para emparejar los proyectos ofertados con las personas con características afines al mismo.
5. Núcleo del sistema escalable.

Lo que se espera conseguir con esta aplicación es una herramienta que ayude a encontrar personal adecuado para un determinado proyecto. Es un componente software con el que se pretende dar una facilidad para llevar a cabo ideas de proyectos y facilitar la obtención de un recurso primordial en los mismos: el personal.

2.3 Funcionalidades futuras

Debido a las limitaciones de recursos que un contexto académico nos supone, ciertas funcionalidades del proyecto (las cuales se van a describir a continuación) no se han podido llevar a cabo, la mayoría de ellas son aquellas enfocadas a mejorar la experiencia del usuario. Se ha hecho una distinción entre las distintas líneas de mejora por las que se puede enfocar el futuro desarrollo de la aplicación.

2.3.1 Interfaz web

La interfaz web es la forma que tiene el usuario de conectar con el núcleo del

sistema, pudiendo depositar sus características personales para poder ser emparejados con proyectos afines, o pudiendo introducir la descripción de un nuevo proyecto.

El núcleo del sistema ofrece métodos con los que se podrían añadir ciertas funcionalidades, accesibles mediante la interfaz:

1. Visualización del perfil del usuario que haya iniciado sesión.
2. Actualización de las características del usuario.
3. Diseño de la interfaz *user-friendly*.

2.3.2 Recogida de datos

En la primera iteración planteada del alcance del proyecto, se incluyeron ciertas funcionalidades con respecto a la recogida de datos que, por falta de recursos, no se han podido incluir en la primera versión del sistema:

1. Recogida de datos de personas en distintas redes sociales (Facebook, Twitter, ...) y profesionales (LinkedIn ...), de forma que, tras un procesamiento de esos datos, se puedan incluir en la base de datos basada en la ontología correspondiente, y así se pueda rellenar de forma automática el curriculum del usuario en **Manaject**.
2. Incluir nuevos campos de características mediante la interfaz web. Al ser la primera versión del sistema, es posible que los usuarios detecten cierta escasez de opciones a la hora de personalizar sus currículos, por lo que sería de interés darles la posibilidad de ampliar el abanico de posibilidades.

2.3.3 Seguridad

La seguridad del sistema no ha sido uno de los principales focos de la aplicación, por lo que esta constituye una línea interesante de progreso para **Manaject**.

Al no estar en el alcance inicial del proyecto, es una línea recomendable a seguir de cara a la siguiente versión de la misma, ya que la actual constituye una base sobre la que construir un proyecto orientado a un uso fuera del contexto académico en el que se ha movido.

2.3.4 Implantación de ontologías

Las ontologías son uno de los aspectos más importantes de la aplicación y sobre ellas se construye la base del sistema de recomendación. Actualmente, las ontologías se despliegan de forma estática en la interfaz web (los valores de los

distintos *input* en la interfaz web representan las propiedades y clases de las ontologías) de manera que, a pesar de cambiar de ontología, en la interfaz web se desplegarían las mismas opciones. Resultaría interesante implementar la función que cause que la ontología se despliegue automáticamente en la misma. En la versión actual de la aplicación, las ontologías se han representado de forma manual, por lo que solamente se han podido mostrar parte de las mismas y no se ha podido aprovechar todo el potencial que estas contienen. La red de ontologías [2] incluidas en la aplicación ofrece una cantidad de posibilidades inmensa, por lo que se vería muy enriquecido el número de opciones disponibles para el usuario, automatizando este proceso de representación.

2.3.5 Sistema de recomendación

Como se ha especificado en el apartado de **Introducción** de este documento, el sistema de recomendación implementado actualmente en la aplicación, emplea la técnica de filtrado basada en contenido. Esta técnica, como indica su nombre, hace uso del contenido que hay en las distintas muestras del conjunto de muestras y recomienda aquellas más similares o más cercanas a la que se quiere evaluar.

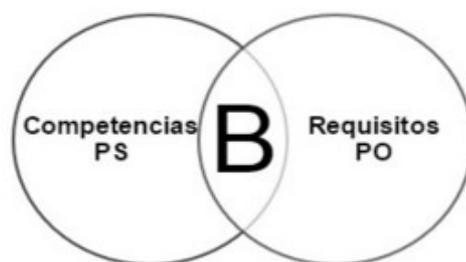


Figura 1: Recomendaciones basadas en contenido

Por una parte, se tienen las competencias que ha señalado poseer un *Project Seeker* en su perfil personal; y por otra, las competencias que ha introducido en la descripción del proyecto el *Project Head*. La intersección de ambos conjuntos (B) tiene aquellas competencias que se comparten en ambos, y es esto lo que utiliza esta técnica, aplicada al sistema de recomendación de **Manaject**.

Lo ideal para que un sistema de recomendación sea óptimo es que utilice una técnica de filtrado híbrida, ya que, además del contenido, también tiene en cuenta los gustos de los usuarios (en este caso, se tendrían en cuenta los gustos de los *Project Seeker*). Por lo que sería interesante en un futuro desarrollar un segundo sistema de recomendación, dentro de la aplicación, que se base en un filtrado colaborativo, y hacer una intersección de los resultados de ambos sistemas, de

forma que la aplicación asegure que las personas recomendadas no solo son las más adecuadas con respecto a las exigencias del proyecto, sino que también asegure que esas personas tienen interés en las características descritas para el proyecto a llevar a cabo.

2.4 Tiempo dedicado a las distintas actividades

A continuación se presentan las tablas en la que se reflejan la dedicación contabilizada en horas de las distintas tareas que se han llevado a cabo durante el ciclo de vida del proyecto.

Tareas	Dedicación (horas)
Web Semántica	20
Sparql	10
Jena	10
Base de datos Stardog	10
Sistemas de recomendación	25
JSP	10
Tecnologías semánticas	10

Tabla 1: Tiempo dedicado al aprendizaje

Tareas	Dedicación (horas)
Planificación	3
Alcance	10
Seguimiento y control	5
Reuniones	10

Tabla 2: Tiempo dedicada a la gestión del proyecto

Tareas	Dedicación (horas)
Parte cliente de la aplicación	
Desarrollo y contenido de la página web	25
Integración de las tecnologías necesarias	10
Parte servidor de la aplicación	
Integración y conexión con Tomcat	10
Configuración e implementación de JSP	20
Núcleo del sistema	
Búsqueda y selección de la ontología	10
Métodos de interacción con la base de datos empleando Jena	30
Pruebas y solución de errores	15
Implementación de los procesos de transformación de datos	30
Sistema de recomendación	10
Base de datos	
Integración y conexión con Stardog	5
Documentación	
Memoria	40

*Tabla 3: Tiempo dedicado a la aplicación **Manaject***

En total, el tiempo invertido para el desarrollo del proyecto ha sido de 328 horas.

2.5 Riesgos del proyecto

Como en todas las decisiones que tomamos en la vida, hay riesgos que hay que tener en cuenta para poder gestionar su solución de manera proactiva. Si bien es cierto que muchas de las situaciones que suponen un riesgo para el proyecto no pueden ser previstas y, por lo tanto paliadas, también es cierto que muchas otras si pueden serlo. A continuación se muestra la tabla que presenta aquellos riesgos que han sido previstos, junto con la probabilidad (en símbolos) de que ocurra esta situación, el impacto que tendría sobre el proyecto en cuestión y la medida

adoptada para evitar, en la medida de lo posible, que esta se produzca.

Situación	Probabilidad	Impacto	Medida
Pérdida de datos	Baja	Alto	Se han hecho copias de seguridad de as distintas partes del proyecto en diversos repositorios: 1. Google Drive 2. GitHub 3. Copia local
Decisiones de diseño equivocadas	Media	Variable	Organizar reuniones con el director del proyecto para plantearle determinadas decisiones de diseño
Disparidad de disponibilidad de horarios	Media	Bajo	Organizar las reuniones con antelación por medio de un soporte de comunicación electrónico

Tabla 4: Situaciones de riesgo del proyecto

Capítulo 3. Análisis de antecedentes

3.1 Contexto tecnológico

3.1.1 La Web Semántica

Lo que se pretende conseguir con la Web Semántica es proveer de un significado y una estructura a la información contenida en las distintas páginas web, permitiendo así que agentes software lleven a cabo tareas más sofisticadas que de otra manera serían imposibles de conseguir o mucho más costosas. Gracias a la web semántica, los agentes inteligentes van a poder entender las relaciones implícitas entre la distinta información contenida en la página, mediante ciertas palabras del vocabulario seleccionado contenidas en las propias relaciones. La web actual también recibe el nombre de “web de documentos” ya que hoy en día casi toda la información que está esparcida por la web está en diferentes documentos y las relaciones entre ellos no son legibles para un agente inteligente. La Web Semántica busca conseguir una web orientada a datos, en la cual las relaciones entre la información sean claras y sigan una semántica acorde con la ontología en la que se basan, de forma que un agente software pueda comprender las distintas relaciones entre la información y procesarla como sea conveniente. Mediante esta infraestructura en común, es posible compartir y procesar la información de manera relativamente simple, ofreciendo así una mejor solución que la actual al problema de encontrar determinada información por la web.

La Web Semántica está basada en unos estándares para que la representación de los datos sea universal y así facilitar la manipulación de datos. Los estándares básicos son los siguientes: RDF, OWL y SPARQL.

3.1.1.1 *RDF*

RDF es un modelo de datos que proporciona información descriptiva simple sobre los recursos que se encuentran en la web y que se utiliza, por ejemplo, en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etc [3]. En este modelo, cada recurso tiene un URI (Universal Resource Identifier) asignado. Estos URIs son únicos y sirven para identificar el propio recurso. La Figura 2 [4] es un ejemplo de un grafo RDF, que representa a una persona llamada Eric Miller.

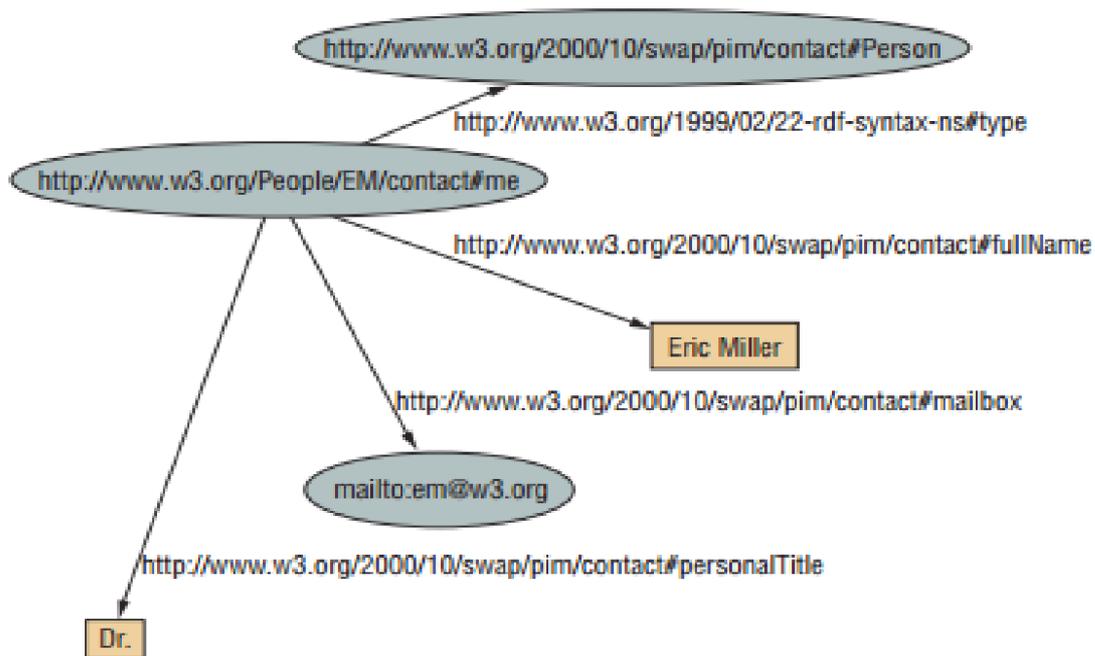


Figura 2: Grafo RDF

Como se puede ver en la Figura 2, las relaciones que se establecen entre dos *items* se representan con tres elementos y es por esto que, a las expresiones que representan este tipo de relaciones se las conoce como triples. Los triples constan de sujeto-predicado-objeto donde el sujeto indica el recurso en cuestión, (`http://www.w3.org/People/EM/contact#me`), el predicado denota un determinado rasgo del recurso (`...#fullName`) y el objeto es el valor de ese predicado para ese sujeto.

3.1.1.2 OWL

El Lenguaje Web de Ontologías (OWL) es un lenguaje semántico de marcas para crear y compartir ontologías, y está basado en lógicas de descripciones. Se usa para definir ontologías cuando el contenido necesita ser procesado por agentes o aplicaciones, en vez de solamente ser presentado a la gente [5]. OWL aporta una mayor expresividad que RDF o que RDF schema.

3.1.1.3 SPARQL

SPARQL es el lenguaje de consulta recomendado por W3C para RDF [6]. Es un lenguaje diseñado para hacer consultas sobre recursos que estén en formato RDF.

Este lenguaje también está compuesto de triples para hacer las consultas y/o las actualizaciones, pero permite la inclusión de variables que sustituyan cualquier parte del triple (sujeto, predicado u objeto). Haciendo esto, el motor de consulta buscará en la base de datos y devolverá todos los casos que satisfagan el triple de la consulta. También es posible hacer consultas sin incluir ninguna variable en ella, cosa que podría resultar útil si lo que se busca es determinar la existencia de un triple en concreto. A continuación se plantea un ejemplo de consulta SPARQL:

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2
3 SELECT ?x WHERE {
4     ?x foaf:name "Mikel López Rodríguez"
5 }
```

Figura 3: Consulta SPARQL que devuelve el recurso con el nombre del literal

Como se puede ver en la la Figura 3, SPARQL permite incluir prefijos, que sirven para especificar los *namespaces* que se van a utilizar en las consultas. Después de la declaración del prefijo en cuestión, viene la palabra clave SELECT, tras la cual se ponen las variables que se desea que la consulta devuelva. En este caso concreto, se le pide a la consulta que devuelva todos los sujetos cuyo nombre sea “Mikel López Rodríguez”.

SPARQL permite hacer consultas mucho más complejas que la que está aquí expuesta. En [6] se puede encontrar información más explícita acerca de esta tecnología.

3.1.2 Ontologías

En la web semántica, los vocabularios definen una serie de conceptos y relaciones (también conocidos como “términos”) empleados para describir y representar el área a la que hace referencia. Los vocabularios se usan para clasificar los términos que pueden tener una aplicación en particular, caracterizar posibles relaciones, y definir posibles restricciones del empleo de los términos contenidos.

Los vocabularios pueden ser muy complejos (con varios miles de términos) o muy simples (describiendo uno o dos conceptos solamente).

No hay una clara diferencia entre lo que es un “vocabulario” y una “ontología”. La tendencia es emplear la palabra “ontología” para una colección de términos

más formal y más compleja, mientras que “vocabulario” se usa cuando no es necesario tanto formalismo. Los vocabularios son los bloques de construcción básicos para las técnicas de inferencia en la Web Semántica.

En este proyecto, se ha escogido la red de ontologías [2] como ya se ha mencionado con anterioridad. Antes de tomar esta decisión, se realizó un estudio de las distintas opciones de ontologías disponibles orientadas a la gestión de recursos humanos. Esta, una red de ontologías en la cual cada una de ellas representan un área distinta relacionada con la gestión de recursos humanos, es una red de ontologías muy completa y está disponible en los formatos *.wsml* y *.ttl* (este último es el formato escogido, ya que la base de datos STARDOG puede leerlo). A continuación se presenta el modelo conceptual de la misma, explicando cada ontología y las relaciones que estas describen entre ellas.

3.1.2.1 Modelo conceptual

En esta sección se va a presentar el modelo conceptual de la red de ontologías que componen la ontología de gestión de recursos humanos que se va a emplear en el proyecto¹. Para ello, se va a describir cada una de las ontologías que la componen, que son las siguientes: *Education Ontology*, *Competence Ontology*, *Project Offer Ontology*, *Project Seeker Ontology*, *Language Ontology*, *Occupation Ontology*, *Skill Ontology* y *Time Ontology*. Cada una de estas ontologías se va a ocupar de una parte en concreto de la gestión de recursos humanos que se va a llevar a cabo en el proyecto. Algunas de estas ontologías solamente serán un apoyo de otras. Esta red de ontologías se puede descomponer en tres tipos de ontologías diferentes: ontologías de aplicación (*Job Seeker Ontology*, *Job Offer Ontology*), ontologías de dominio (*Competence Ontology*, *Education Ontology*, *Language Ontology*, *Occupation Ontology*, *Skill Ontology*) y ontologías generales (*Time Ontology*). La red de ontologías en general tiene el siguiente aspecto:

1 http://oa.upm.es/10301/1/A_network.pdf

y *Job Vacancy*, si surge una vacante nueva, tanto *Project Vacancy* como *Vacancy* actualizarán valores. La mayoría de axiomas definen este tipo de relaciones, no solo en la ontología de la oferta de proyectos, sino en todas las demás también. A este tipo de axiomas se les denomina “axiomas para las relaciones inversas”.

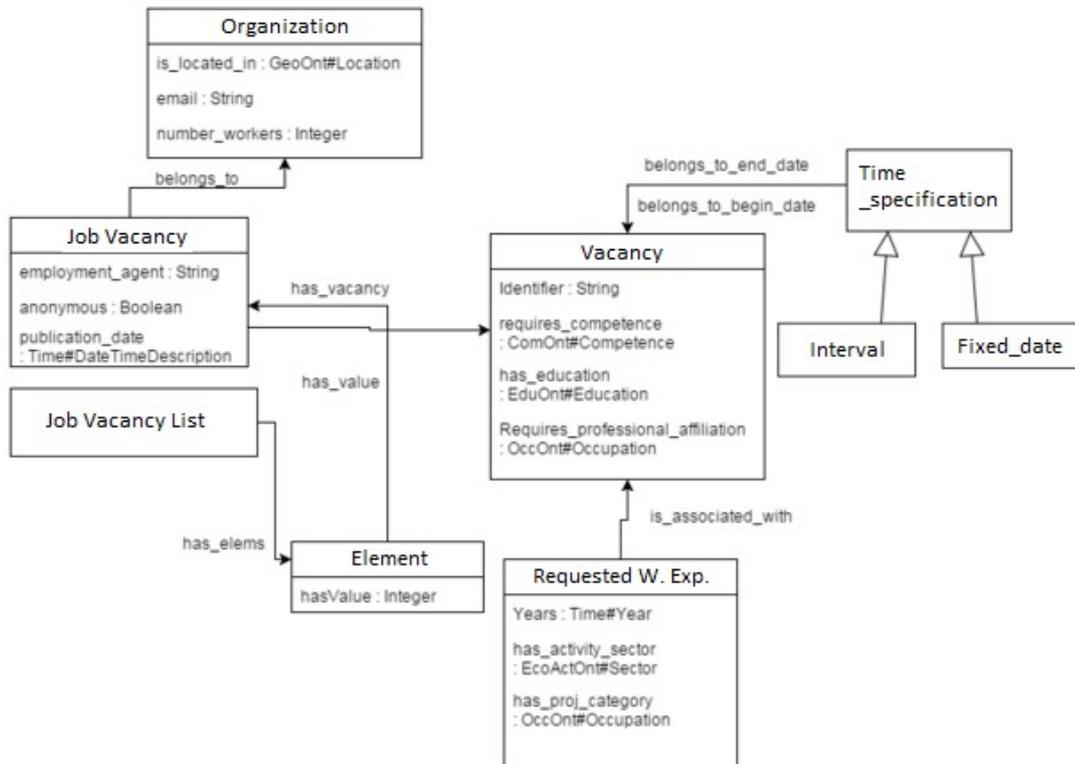


Figura 6: Diagrama de clases de Job Offer Ontology

Job Seeker, sin embargo, guarda los datos de la persona que busca proyectos que realizar y, dependiendo de sus aptitudes, tendrá acceso a un tipo de proyectos o a otros. Contiene datos tanto académicos, como profesionales y personales y, a partir de estos, será como el sistema de recomendación hará el emparejamiento con los proyectos. Por lo tanto, la ontología de *Job Seeker* debe definir un concepto de CV e incluir atributos relacionados con la información personal, entre ellos, un atributo *has_nationality_from* de tipo *Country*, donde *Country* es un concepto importado de otra ontología llamada *Geography*. Esta ontología define un concepto de alto nivel llamado *Job_Seeker*, el cual es el que almacena los atributos relacionados con la información personal y un atributo de tipo *Candidacy*. El concepto *Candidacy* reúne información relacionada con las competencias, las lenguas habladas, educación de la persona, experiencia laboral, etc.

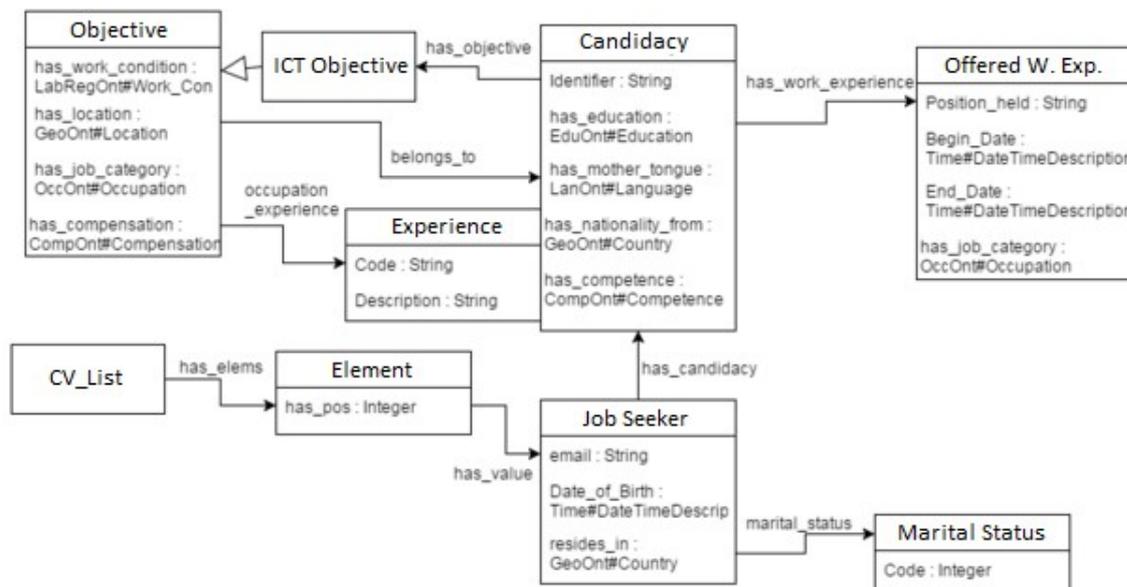


Figura 7: Diagrama de clases de Job Seeker Ontology

3.1.2.1.2 Ontologías de dominio

El desarrollo de las ontologías de dominio se ha basado en emplear estándares y códigos internacionales siempre que ha sido posible. En algunas áreas de conocimiento como geografía, niveles de educación, lenguajes y niveles de lenguaje solamente hay un estándar internacional/europeo aceptado.

Education contiene las distintas competencias académicas que puede tener una persona y las distintas áreas en las que puede haberse especializado, así como sus títulos, el nivel de esas competencias, el tiempo que estuvo en ellos etc. Estos datos los empleará el sistema para emparejar a las personas con los estudios necesarios, con aquellos proyectos que los exigen.

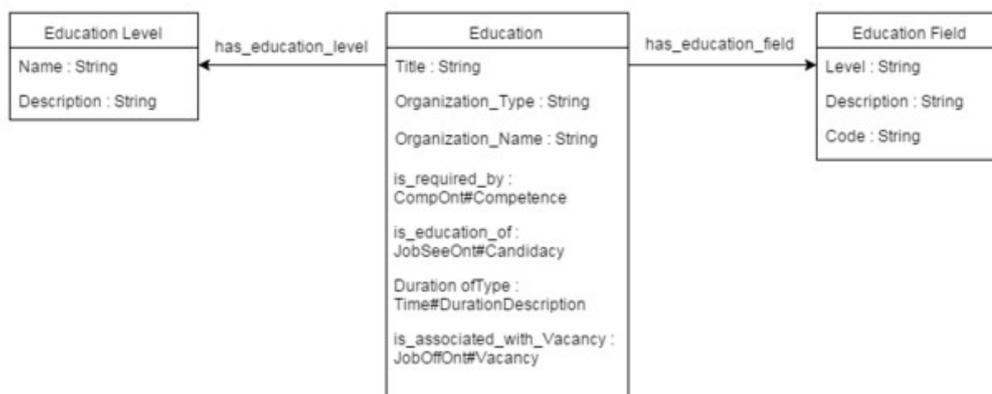


Figura 8: Diagrama de ejemplo de la ontología de dominio Education Ontology

Cada una de las distintas áreas en las que puede estar especializada una persona está organizada como un concepto, de forma que cada una de ellas tenga su código para darles un identificador propio.

En *Language*, la información relevante de contexto contiene los distintos lenguajes que están disponibles en la ontología. Esta descansa sobre tres conceptos: *Language*, *Language_Level* y *Language_Proficiency*. Con estos se determinará el nivel de una persona en las distintas competencias de un lenguaje, como son la lectura y la escritura por ejemplo.

Occupation contiene las diversas ocupaciones de los distintos campos que puede tener una persona. Los axiomas definidos para esta ontología se ocupan de darles un código, un nombre y una descripción diferente a cada labor.

Skill es otra de las ontologías principales del proyecto, cuya información de contexto almacena las distintas habilidades existentes, distinguidas por tipos de habilidad como son habilidades sociales, artísticas, relacionadas con bases de datos, etc.

Todas las ontologías de dominio siguen una estructura similar a la que se puede ver en la Figura 8. Cada una de ellas tiene un número de axiomas diferente y ciertos axiomas propios de la ontología en cuestión dentro de la red de ontologías, pero la estructura de estas ontologías es parecida a la descrita para la ontología *education ontology*.

3.2 Trabajos relacionados

Se pretende realizar un estudio acerca de algunos proyectos ya existentes relacionados con el mundo de la gestión de recursos humanos mediante tecnologías semánticas. Este estudio se hará con el fin de poner en conocimiento

ciertos trabajos realizados en el área de las tecnologías de Web Semántica en los sistemas de recomendación y, más concretamente, en aquellos destinados a la gestión de recursos humanos. También se pretende aprender las distintas técnicas empleadas para realizar los emparejamientos óptimos en los sistemas de recomendación.

3.2.1 The Impact of Semantic Web Technologies on Job Recruitment Processes

En [7] se describe el impacto de las tecnologías semánticas en los procesos de reclutamiento de personal para las distintas ofertas de empleo. Existen muchos portales de trabajo en los distintos países que permiten a las personas interesadas en encontrar empleo hacer una solicitud al mismo, pero son demasiados portales los que existen y las personas interesadas en encontrar trabajo solamente pueden hacer la solicitud en unos pocos portales. Además, los empleadores tienen que preseleccionar a los solicitantes manualmente, lo cual es un trabajo muy costoso. Empleando tecnologías semánticas, se pueden solucionar ambos problemas.

Un proceso de reclutamiento normal se divide en cuatro fases principales:

- Análisis de requisitos
- Publicitar la oferta de empleo
- Recibir y preseleccionar candidatos
- Decisión final de reclutamiento

Estas fases pueden mejorarse haciendo uso de la web semántica publicando la oferta de empleo con un vocabulario en concreto, de forma que se facilite un emparejamiento semántico entre el perfil de la persona y la oferta de empleo.

Análisis de requisitos: En vez de escribir los requisitos en texto plano, describiendo libremente cada requisito, se emplea un lenguaje común de un cierto grupo de vocabularios para facilitar la comunicación de todas las partes “interesadas” y además poder automatizar ciertos procesos.

Publicitar la oferta de empleo: Hay muchos y diferentes portales de empleo que compiten entre ellos y se especializan en diferentes regiones geográficas. Los empleadores publican sus ofertas de trabajo en unos pocos de esos portales, no en todos, con lo que, muchos potenciales candidatos nunca llegan a saber de dicha oferta.

Para solucionar este problema, algunas empresas comenzaron ciertos proyectos para poder centralizar dichas ofertas de trabajo pero eso trae como problema un

cuello de botella.

Con el empleo de tecnologías de web semántica se puede evitar dicho problema además de que los portales de trabajo podrían ofrecer un servicio de emparejamiento semántico que calcularía la similitud entre un perfil de trabajo y una oferta de empleo, basándose en un conocimiento previo acerca de cómo determinados términos están relacionados.

Recibir y preseleccionar candidatos: Empleando tecnologías semánticas, una persona candidata para un trabajo podrá reutilizar su perfil RDF de forma que pueda enviarlo a todas las ofertas de empleo en las que esté interesado y los empleadores solo tendrán que poner los requisitos mínimos con el vocabulario adecuado para hacer su preselección, ya que se hará de forma automática y no será necesario hacerlo manualmente.

3.2.2 Ontology-based Skills Management

En [8] se habla acerca de un proyecto basado en ontologías para la gestión de competencias de las personas dentro de una empresa, llamado KOWIEN.

Un sistema de gestión de habilidades debe conseguir tres objetivos principales:

1. Debe ser capaz de adquirir conocimiento acerca de las habilidades que poseen los miembros de la empresa.
2. Debe dar a conocer las competencias y las personas que las poseen.
3. Debe emplear ese conocimiento para un fin.

Para poder alcanzar estos objetivos con mayor eficacia, el sistema de gestión de habilidades va a emplear ontologías. El usuario de un sistema de gestión de habilidades basado en ontologías obtiene respuestas más valiosas que las que obtendría mediante una query de una base de datos relacional. El mecanismo de inferencia se beneficia en calidad, capacidad de actualización y aceptación.

El modelo de proceso KOWIEN está orientado a aquellos que quieren conceptualizar, implementar y evaluar un sistema de gestión de habilidades basado en ontologías. Dicho modelo consta de cinco fases:

Investigación de requisitos: Define el área de la aplicación. Los requisitos deben recogerse sistemáticamente para poder llevar a cabo la fase de evaluación más adelante. Si se identifican demasiados requisitos, es aconsejable emplear una herramienta de gestión de los mismos.

Adquisición del conocimiento: Recoger toda la información necesaria para poder

conceptualizar la ontología. Las fases de adquisición y conceptualización deben hacerse de forma iterativa ya que las actividades de adquisición y de estructurar el conocimiento van acompañadas.

Conceptualización: Por una parte contiene la terminología y por otra parte las reglas para la interpretación y el uso de los conceptos. No solo conceptualizan los miembros del proyecto, también lo hacen los usuarios que hayan sido interrogados a la hora de adquirir el conocimiento.

Implementación: Representación formal de la conceptualización y la integración de la aplicación basada en ontologías en el entorno del sistema.

Evaluación, documentación y gestión del proyecto

Para implementar y modificar la ontología que emplean.

3.2.3 Ontology Development For Human Resource Management

En [9] se crea una ontología para apoyar dos proyectos distintos:

- 1- Un proyecto basado en meta-búsqueda en portales de trabajo
- 2- Un sistema de gestión de competencias en la universidad

Esta ontología soporta distintos niveles de competencias así como distintos grados de satisfacción para cada nivel.

Para que una ontología sea adecuada para la gestión de recursos humanos debe cumplir ciertas características. Los perfiles de las personas con sus competencias tienen que cumplir tres importantes requisitos:

Comparabilidad: Es necesario utilizar mismo vocabulario.

Validez: Los valores dados a los niveles de competencias deben ser ciertos. Esto se comprueba mediante pruebas o certificados.

Privacidad de datos: Como los niveles de una competencia que una persona pueda tener no son estáticos (ya que puede ir aprendiendo), es necesario que estos siempre sean modificables. El problema de representar los niveles de las competencias es saber qué condiciones deben cumplirse para alcanzar la plenitud.

Estos son los requisitos finales que definieron para la ontología generada:

1. Capacidad de emparejamiento entre los requisitos del trabajo y los perfiles de competencias de las personas.
2. Sinónimos de ocupaciones y competencias.

3. Conceptos para atributos personales.
4. Relaciones entre ocupaciones y competencias requeridas.
5. Escalas y conceptos de medición para las competencias.
6. Dependencias entre competencias.

Consideran dos tipos de competencias: funcionales y de comportamiento.

Competencias funcionales: Conocimiento y habilidades (también conocidas como competencias duras).

Competencias de comportamiento: Comportamiento y aptitudes personales (competencias blandas).

3.2.4 An ontology-based intelligent system for recruitment

El objetivo de [10] es la construcción de un portal web que ofrece guía para buscar trabajo a los usuarios, e instrumentos para que sea posible que contacten estos usuarios. Este portal va a servir como punto de encuentro entre empresas, oportunidades de formación y gente que quiera encontrar trabajo. Además, emplea tecnologías semánticas con el fin de hacer emparejamientos más inteligentes entre los curriculums de la gente y las ofertas de trabajo, empleando un motor de búsqueda basado en ontologías.

La información que contiene el portal debe ser actualizada dinámicamente. Las etiquetas principales de información en dicho portal son las siguientes: Empleos públicos y privados (público y privado hacen referencia al sector), Formación, Noticias y Solicitudes de Empleo. Los descriptores para las ofertas de trabajo y para las demandas de los mismos son comunes, de forma que el emparejamiento entre el candidato y la oferta de empleo es mejor.

La ontología que se emplea tiene 4 conceptos principales sobre los que se va a construir el trabajo: Solicitante (de trabajo), Empleador, Oferta y Perfil. Cada uno de estos conceptos tiene sus propios atributos, gracias a los cuales se va a poder llevar a cabo los emparejamientos entre las ofertas de trabajo y los solicitantes de las mismas. Los atributos de cada concepto son los siguientes:

Solicitante: E-mail y contraseña; Teléfonos de contacto; País, provincia y ciudad; Fecha de nacimiento; Salario mensual (deseado); Disponibilidad; Carnet de conducir, fecha del carnet de conducir y vehículo propio; Sexo y foto.

Empleador: E-mail y contraseña; Nombre y CIF; Actividad (sector en el que trabaja la compañía); Número de teléfono; Provincia, ciudad, dirección y código postal; Número de empleados; Información de contacto.

Oferta: La oferta se descompone en dos tipos: Profesional y No-Profesional.

Profesional (oferta de trabajo, anuncio de cursos de formación...).

No-Profesional (publicación de noticias...)

La lista de atributos de Oferta es la siguiente: Nombre (heredado de Oferta, es una pequeña traducción de la oferta); Area; Departamento; Fecha de contratación; Provincia y ciudad; Tipo de contrato; Calendario de trabajo; Descripción; Salario mínimo y máximo; Carnet de conducir, fecha del carnet de conducir y vehículo propio; Número de vacantes; Fecha de expiración.

Perfil: Es el concepto más importante de la aplicación ya que es en el que se basan para poder realizar el emparejamiento. Este concepto no tiene atributos propios pero está compuesto por la unión de las siguientes habilidades: Estudios; Conocimiento de idiomas; Habilidades sociales; Experiencia profesional.

Se emplean dos tipos de relaciones interconceptuales, llamadas 'Part-of' y 'IS-A':

Part-of: un concepto forma parte de otro concepto (por ejemplo, cualquiera de las habilidades descritas en el concepto Perfil con respecto a Perfil).

IS-A: es una relación taxonómica (Oferta de Trabajo con respecto a Oferta de tipo Profesional).

3.2.5 Matching People and Jobs: A Bilateral Recommendation

En [11] crean un sistema de recomendación para poder emparejar personas con trabajos. En dicho trabajo, defienden que para poder hacer un buen emparejamiento, se tienen que tener en cuenta tanto las preferencias del trabajo para las personas que se requieren, como las preferencias de las personas. Para poder construir una aproximación estadística para escoger los mejores candidatos, hay que tener en cuenta dos aspectos importantes:

1- El individuo debe tener las habilidades necesarias para poder llevar a cabo la tarea encomendada.

2- El emparejamiento de individuos con otros individuos con los que pueda trabajar sin problema.

Para hacer una primera búsqueda del personal, se centran en el primer punto, ya que si una persona no cumple los requisitos, no tiene sentido que sea recomendada.

Otra de las características importantes para poder recomendar personas consiste en la relación existente entre la persona y el entorno (P-E). *“No son la persona o el entorno por separado los que determinan los cambios en las variables de comportamiento y actitudinales, sino la interacción entre las características de ambos”*. Un nivel positivo de interacción, da resultados positivos tanto para el individuo, como para el equipo para el que trabaja y, por lo tanto, para la empresa también.

Como se ha comentado anteriormente, no se podría considerar un emparejamiento adecuado entre un candidato y una oferta de trabajo si los requisitos para generar dicho emparejamiento solamente consideran a uno de los afectados. Para que el emparejamiento sea idóneo, se tiene que tener en cuenta los requisitos de ambas partes, no solo los de la oferta de trabajo para escoger a los candidatos, también los de los candidatos para buscar aquellas ofertas de trabajo en las que estén interesados. Teniendo esto en cuenta, generaron dos sistemas de recomendación, uno que tuviese en cuenta las necesidades del trabajo y que buscase aquellas personas cuyo CV satisficiera dichas necesidades, y otro que tuviese en cuenta las preferencias de las personas y de esa forma encontrase aquellos trabajos que pudiesen interesarle.

3.2.6 A Recommender System for Job Seeking and Recruiting Website

En [12] desarrolla un sistema de recomendación pensado para encontrar trabajo y reclutamiento de personas. En dicho trabajo se defiende que para que un sistema de recomendación basado en reclutamiento de personas y emparejamiento de las mismas para encontrar trabajo sea efectivo, no solo hay que emparejar por aptitudes sino también por gustos. El sistema de recomendación híbrido tiene en cuenta el trabajo y el perfil del usuario para hacer el emparejamiento, pero también tiene en cuenta las acciones que lleva a cabo dicho usuario con el fin de poder personalizar en mayor medida las recomendaciones de los candidatos del empleo.

La recomendación se genera sobre un grafo dirigido donde las relaciones basadas en contenido y las relaciones basadas en interacción son las aristas y las distintas entidades del sistema (empleadores, trabajos y candidatos) son los vértices del grafo. Después, las entidades se ordenan en base a la relevancia para el usuario, considerando las preferencias de los empleadores y de los candidatos.

Como se ha mencionado anteriormente, este sistema de recomendación emplea dos tipos de relaciones: las relaciones basadas en contenido y las basadas en interacción. Las relaciones basadas en contenido se dividen en dos tipos:

coincidencias de perfil y similitudes de perfil.

Coincidencias de perfil: Si el CV de un candidato contiene coincidencias con la descripción de un trabajo, es probable que esté interesado en él. Relación bidireccional.

Similitudes de perfil: Si dos entidades del mismo grupo tienen perfiles similares, presumiblemente tendrán intereses y oportunidades en común.

Las relaciones basadas en interacción son un tanto más complejas y se desgranán en más tipos.

Postear: Si una persona muestra interés en un empleador, habrá mostrado interés también en los trabajos que emplea, y viceversa. Relación bidireccional.

Referenciar: Si el candidato hace referencia a un trabajo, está interesado en el y puede que esté interesado en trabajos similares. Relación unidireccional.

Favorito: Si el usuario añade una entidad en su lista de favoritos, está interesado en esa entidad. Relación unidireccional.

Gustar: Muestra un interés menos fuerte que postear o referenciar.

Visitar: El hecho de haber visitado la página implica que muestra algo de interés en ella pero puede que, después de haber leído la descripción, no lo haga. Interés débil.

La similitud de dos entidades del mismo tipo está definida por una combinación lineal de los valores de similitud de las distintas áreas.

Para poder calcular las similitudes de perfil entre un trabajo y los candidatos, los ficheros se convierten en texto plano y la similitud se calcula empleando las herramientas de “Latent Semantic Analysis (LSA)”.

3.3 Relación de trabajos con Manaject

En esta sección se hace referencia a los aspectos que más se han tenido en cuenta de los trabajos estudiados y que se han adoptado para que formen parte del sistema que se está llevando a cabo en este proyecto. El trabajo denominado “The Impact of Semantic Web Technologies on Job Recruitment Processes” se puso en cabeza de la sección de *Trabajos relacionados* ya que es un trabajo que tiene muchos aspectos en común con lo que se ha llevado a cabo en este proyecto. La forma en la que en **Manaject** se realiza el *Análisis de requisitos* es como se describe en aquel trabajo; poniendo un lenguaje que sea común para todos los usuarios interesados en realizar la descripción de un proyecto. Además, en un futuro, con un vocabulario en común, las descripciones de proyectos publicitadas

en la aplicación de **Manaject**, serán exportables a aplicaciones de esta misma índole.

En el segundo trabajo se habla de KOWIEN, un sistema de gestión de habilidades basado en ontologías. Esto es, a grandes rasgos, lo que pretende construir **Manaject**. Este trabajo llamado “Ontology-based Skills Management”, se ha recogido en los análisis de antecedentes debido a su parecido con el proyecto **Manaject**.

“Ontology development for human resource management” habla acerca de un modelo de ontología para la gestión de recursos humanos y de ciertos requisitos que debería cumplir. En dicho trabajo, solo se presenta el modelo de la ontología, pero no la propia ontología, por lo que hubo que buscar una ontología de gestión de recursos humanos alternativa. Este trabajo sirvió para saber el modelo de ontología que debía seguir la ontología que se fuese a seleccionar.

En el trabajo de “Matching People and Jobs: A Bilateral Recommendation”, se habla de la importancia de que el sistema de recomendación de la aplicación sea híbrido. Esta era la idea inicial de la aplicación de **Manaject**, pero por falta de tiempo se ha tenido que tomar la decisión de implementar un sistema de recomendación basado en contenido y construir, en una versión futura de la aplicación, el sistema de recomendación colaborativo. Una vez se haya hecho eso, haciendo la intersección de los resultados de ambos sistemas, obtendremos un sistema de recomendación de filtrado híbrido.

“A Recommender System for Job Seeking and Recruiting Website” describe que relaciones utiliza su sistema de recomendación para determinar la compatibilidad entre una oferta de empleo y sus candidatos. Este sistema de recomendación también es híbrido, y como se acaba de señalar, el sistema de recomendación de **Manaject** en el estado actual es de filtrado basado en contenido. Además, el trabajo estudiado hace distintos grupos de personas, por lo que emplea un filtrado basado en contenido más complejo.

Capítulo 4. Funcionalidades y características del sistema

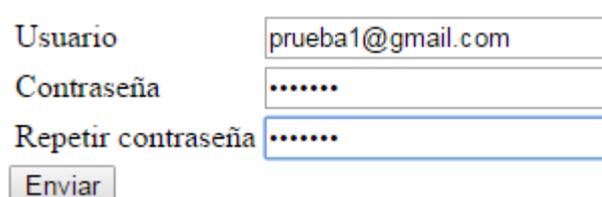
En este capítulo se pueden ver las distintas funcionalidades que ofrece la aplicación en su estado actual de desarrollo. También se presentan las distintas características del sistema y las tecnologías que se han empleado para poder llevar a cabo el desarrollo del proyecto.

4.1 Registro de usuarios

Cualquier usuario de la aplicación puede ejercer los dos roles principales ofertados: *Project Head* y *Project Seeker*. El rol de *Project Head* permite al usuario generar un proyecto con una serie de preferencias sobre las cuales trabajará el sistema para emparejarlo con aquellos *Project Seeker* que considere más competentes para el mismo. El usuario, al rellenar las aptitudes de su perfil, se convierte automáticamente en *Project Seeker* y el sistema va a utilizar sus datos para recomendar su emparejamiento con cualquier proyecto con el que sea afín.

4.1.1 Captura de perfiles

Para que un usuario pueda hacer uso de la aplicación de **Manaject**, primero deberá darse de alta en un sencillo formulario que recogerá solamente un correo electrónico y una contraseña con la que poder, más adelante, iniciar la sesión de su cuenta. A continuación se hará el registro del usuario *prueba1@gmail.com*, y será este el usuario que se utilice para los ejemplos de toda la sección de *Captura de perfiles*.



The image shows a registration form with the following elements:

- A label "Usuario" followed by a text input field containing the email address "prueba1@gmail.com".
- A label "Contraseña" followed by a password input field with seven dots representing masked characters.
- A label "Repetir contraseña" followed by another password input field with seven dots.
- A button labeled "Enviar" located below the input fields.

Figura 9: Registro

Una vez se han rellenado los datos pertinentes y se pulsa el botón “Enviar”, pueden darse las siguientes situaciones de las cuales nos avisaría el sistema, de

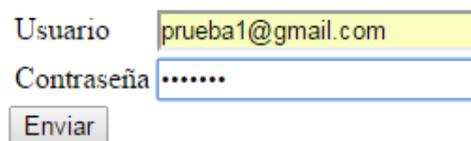
forma que el intento de registrarse fuese insatisfactorio:

1. El usuario ya está registrado en la base de datos.
2. Las contraseñas son distintas.
3. Alguno de los campos está vacío.

Todas las comprobaciones de estas posibles situaciones de error se realizan en el servidor y, parte de estas también en el cliente. En caso de que todo sea correcto, aparecerá un enlace que reenviará al usuario a la página web donde se le permite iniciar sesión para poder entrar en la aplicación con el email con el que se haya registrado.

4.1.2 Iniciar sesión

A la hora de iniciar sesión, al usuario se le pedirán los datos con los que se registró: Usuario y Contraseña.



El formulario muestra dos campos de entrada y un botón. El campo 'Usuario' contiene el texto 'prueba1@gmail.com'. El campo 'Contraseña' contiene siete puntos para ocultar el texto. El botón 'Enviar' está situado debajo de los campos.

Figura 10: Inicio de sesión

De manera análoga a lo que sucedía en el apartado de Registro, a la hora de iniciar sesión el usuario también podrá encontrarse con determinados impedimentos, a saber:

1. El usuario no está registrado en la base de datos.
2. El usuario y la contraseña no coinciden.
3. Alguno de los dos campos se encuentra vacío.

Una vez el usuario inicia sesión satisfactoriamente, se le asigna un *token* de sesión [14]. Los *token* de sesión están constituidos por numerosos elementos, de entre los cuales, uno de ellos es un identificador de sesión que hace que ese *token* sea único. El sistema, tras asignarle al nuevo usuario conectado un *token* nuevo, almacena en la base de datos STARDOG un triple de la siguiente forma:

```

1 insert data {
2   <http://people/prueba1@gmail.com> jobSeeker:id_session "session"
3 }

```

Figura 11: Triple para insertar identificador de sesión

En este triple, se está almacenando en el recurso de “prueba1@gmail.com” del grafo “people”, mediante la propiedad “id_session”, el identificador de sesión “session” obtenido del *token* que se le ha asignado al usuario a la hora de iniciar sesión.

De esta forma, para cualquier consulta o modificación que se vaya a realizar después del inicio de sesión, se podrá relacionar a la persona conectada mediante el identificador de su *token* de sesión actual con el identificador que se ha almacenado en la base de datos. A continuación, en la Figura 12 se presenta un ejemplo gráfico en el que se visualiza el proceso desde que un usuario hace la solicitud al servidor para iniciar sesión, hasta que el identificador de sesión se guarda en la base de datos.

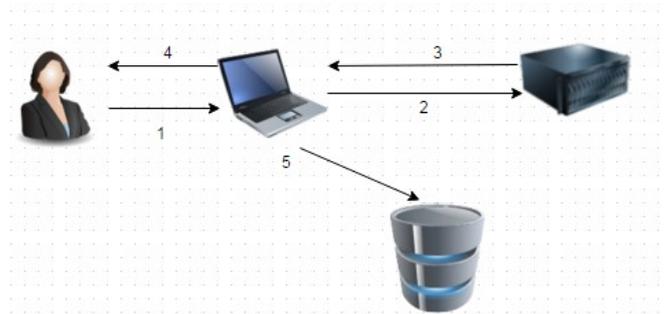


Figura 12: Proceso de inicio de sesión

El proceso que se representa en la Figura 12 es el siguiente:

1. El usuario p1 solicita iniciar sesión mediante la interfaz web ofrecida por la aplicación.
2. La aplicación envía al servidor la solicitud de iniciar sesión de p1.
3. El servidor, tras hacer las comprobaciones pertinentes, genera un nuevo *token* de sesión y se lo devuelve a la aplicación.
4. La aplicación guarda el nuevo *token* de sesión en una cookie.
5. La aplicación extrae el identificador de sesión del *token* y lo asocia en la base de datos con la clave primaria, en este caso con el email, de p1 para que las próximas operaciones se hagan en base a la coincidencia del identificador de sesión.

Una vez ha sucedido todo este proceso, se redirige al usuario a la página de inicio de la aplicación web.

4.1.3 Inicio

La página de inicio ofrece una interfaz simple en la que se dan a escoger entre las dos funcionalidades principales del sistema, que ya se han descrito con anterioridad: *Project Seeker* y *Project Head*.

Inicio

[Perfil Personal](#) [Generar Proyecto](#)

Figura 13: Página de inicio

En la Figura 13 se muestran dos opciones: Perfil Personal y Generar Proyecto. Como ya se ha mencionado anteriormente en este documento, en el momento que el usuario rellena su Perfil Personal, se convierte automáticamente en *Project Seeker* y, si el usuario genera un proyecto, se convierte en el *Project Head* del mismo.

4.1.3.1 Perfil Personal

Una vez se entra en el apartado del perfil personal del usuario, este se va a encontrar con una interfaz con el siguiente aspecto:

Perfil personal

- Formación académica
- Lenguas conocidas
- Lengua materna
- Competencias profesionales
- Habilidades personales

Enviar

Figura 14: Perfil personal

En la Figura 14 se muestran las competencias sobre las que el usuario puede desarrollarse para rellenar su perfil. Como se puede ver, en un principio

solamente se muestran las competencias de forma genérica. En el momento que se pincha sobre una de estas con el ratón, la seleccionada despliega una serie de elementos que están contenidos en el conjunto de la competencia seleccionada. Para ver más acerca de la ontología que representa estos datos, ver [2].

De esta forma, el usuario podrá ir escogiendo entre los distintos campos de competencias, y una vez dentro de ellos ir seleccionando aquellos elementos que representen mejor sus conocimientos y habilidades.

A continuación se presentará un ejemplo del proceso completo del usuario *prueba1*, desde que ya ha iniciado sesión para introducir sus competencias personales en la base de datos, hasta que todos los datos seleccionados han llegado a la base de datos.

El usuario *prueba1*, tras registrarse e iniciar sesión tiene los siguientes datos en la base de datos

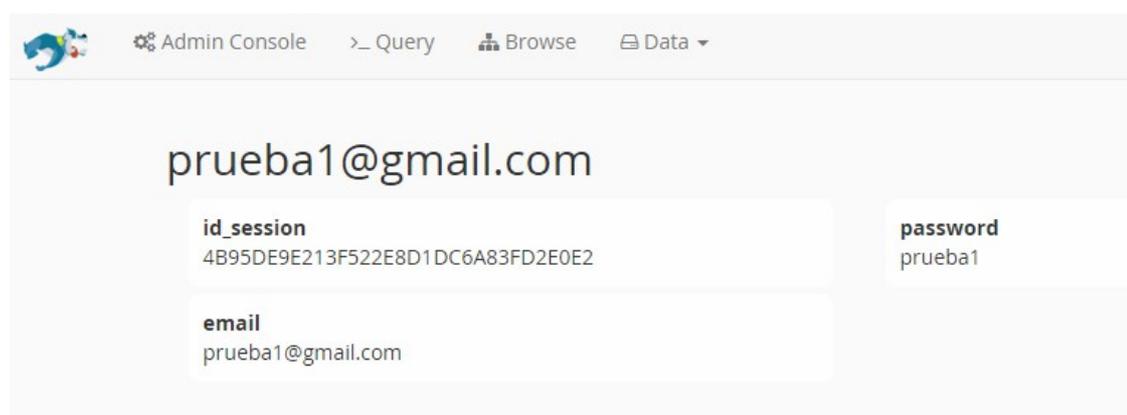


Figura 15: Usuario *prueba1* antes de introducir sus competencias

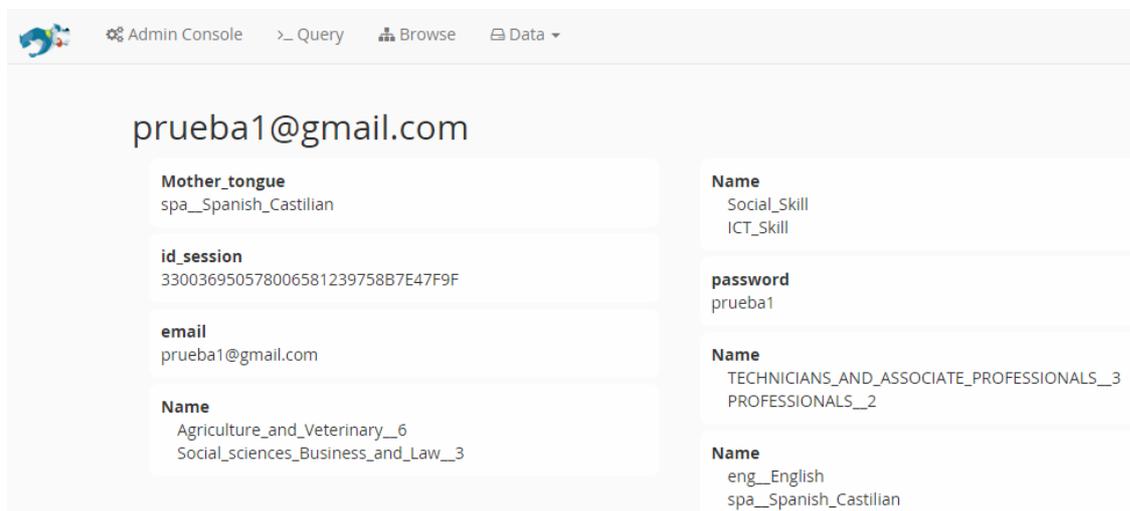
Como se puede observar, el usuario *prueba1* no tiene ninguna competencia hasta la fecha, ya que aún no ha introducido sus datos en el sistema. Por ahora, solo se han almacenado en la base de datos el email y la contraseña con las que se ha registrado y el identificador de sesión que se le ha asignado.

Prueba1 ha decidido añadir las siguientes competencias a su perfil:

Globales	Específicas
Formación académica	Social sciences, business and law
	Agriculture and veterinary
Lenguas conocidas	Inglés
	Castellano
Lengua materna	Castellano
Competencias profesionales	Professionals
	Technicians and associate professionals
Habilidades personales	Social skills
	ICT skills

Tabla 5: Competencias del perfil de prueba1

Una vez que las competencias a añadir han sido seleccionadas y se pulsa el botón “Enviar”, estas se mandan en forma de triples a la base de datos, basándose en la coincidencia entre el identificador de sesión del *token* y de la propiedad *id_session* ya almacenada. El sistema advierte de la correcta inserción de los datos, permitiendo al usuario volver a la página principal. Tras añadir estos elementos, la base de datos de *prueba1* tendría la siguiente forma:



prueba1@gmail.com	
Mother_tongue	spa__Spanish_Castilian
id_session	330036950578006581239758B7E47F9F
email	prueba1@gmail.com
Name	Agriculture_and_Veterinary__6 Social_sciences_Business_and_Law__3
Name	Social_Skill ICT_Skill
password	prueba1
Name	TECHNICIANS_AND_ASSOCIATE_PROFESSIONALS__3 PROFESSIONALS__2
Name	eng__English spa__Spanish_Castilian

Figura 16: Usuario prueba1 tras introducir sus competencias

De esta forma, el sistema, apoyándose en las competencias que encontrará en este recurso, lo emparejará con proyectos que requieran de estas competencias.

4.1.3.2 Generar proyecto

Para poder crear un nuevo proyecto, el usuario deberá pinchar en el enlace de “Generar Proyecto”, desde el cual será reenviado a una página que tiene la siguiente forma:

Perfil del proyecto

Nombre del proyecto:

Número de participantes solicitados:

- Formación académica
- Lenguas conocidas
- Lengua materna
- Competencias profesionales
- Habilidades personales

Figura 17: Perfil del proyecto

Es importante mantener el formato de los datos que se van a enviar a la base de datos para permitir que el sistema de recomendación haga el emparejamiento de forma efectiva. En la Figura 17 se muestra la estructura que sigue el perfil para generar un nuevo proyecto. Los cambios más significativos con respecto al perfil del usuario son los siguientes:

1. Un apartado para incluir el nombre que se le quiere dar al proyecto, este nombre se utilizará como parte de la URI del recurso.
2. Permite incluir el número de personas que se desea que recomiende el sistema, siempre en orden descendiente de personas más adecuadas.
3. En cada una de las competencias a incluir en el proyecto, una sección que permite la inclusión de un número del 1 al 100 que simboliza la importancia que el *Project Head* le da a esa competencia en concreto (El sistema lo transformará en un valor decimal, dividiendo entre 100 dicho valor). Este apartado numérico estará inhabilitado siempre que esa competencia no esté seleccionada, en el momento que una competencia se marca, se habilita ese apartado.

- Social sciences, business and law
- Agriculture and veterinary

Figura 18: Apartado numérico en las competencias

4.2 Núcleo del sistema

Una vez que el usuario ha introducido todos los datos y ha pulsado el botón de “Enviar”, esta información es recogida por la parte servidor de la aplicación, y será la que luego se envíe al núcleo de la aplicación.

Como se mencionará más adelante en el documento, se ha utilizado Apache Tomcat como servidor web, ya que implementa ciertas especificaciones de JavaEE, entre las cuales se encuentra Java Servlet, que será la clase de Java que se va a emplear para realizar, tanto la recogida de datos del cliente, como las respuestas a las solicitudes enviadas por el mismo. La recogida de datos se va a realizar de manera similar para los dos roles disponibles de la aplicación, la diferencia principal reside en que, para el rol de *Project Seeker*, los datos remitidos por el cliente se recogen en el servidor y estos, tras darles un formato adecuado para la ontología empleada y convertirlos en triples, se envían directamente a la base de datos, al grafo de personas; mientras que, para el rol de *Project Head*, además de hacer lo mismo que lo que se realiza en el rol de *Project Seeker*, esta misma información se recoge también en una estructura de datos para procesarla de forma paralela y realizar la consulta pertinente. Este proceso está explicado más detalladamente en la sección 4.2.1.

4.2.1 Procesamiento de datos

En esta sección se van a detallar los distintos métodos que se han empleado para poder procesar los datos paso a paso, desde que el usuario selecciona mediante la interfaz web las competencias que desea, hasta que el sistema, con esos datos, recomienda las personas más adecuadas para el proyecto. El procesamiento de los datos se hace de forma análoga a la hora de recomendar proyectos a una persona teniendo en cuenta sus aptitudes. Dicho desarrollo se lleva a cabo en tres pasos.

4.2.1.1 Paso 1, estudio y selección de ontología

El primer paso, antes de poder procesar los datos a un formato en concreto, fue saber que formato tenían que adoptar; y para esto, se tenía que escoger primero la ontología en la que se iba a basar el proyecto. Como ya se ha comentado

anteriormente en este documento, la ontología seleccionada fue la descrita en el trabajo [2], orientada específicamente a la gestión de recursos humanos. Los valores y las clases descritas en esta, fueron la base del formato adoptado para los datos, de forma que coincidiesen con aquellos, ya que, de otra forma, sería imposible realizar las consultas en SPARQL, así como las actualizaciones.

Por lo tanto, el paso 1 fue un análisis de alternativas de las diferentes ontologías existentes que estuviesen orientadas a la gestión de recursos humanos en un entorno laboral. Este constituye el módulo más importante de todos los que forman el apartado de “procesamiento de datos”, ya que es la base de todo el proceso.

Los términos de la ontología serán los metadatos sobre los que se harán el envío de la diferente información del sistema a la base de datos en forma de triples. De esta manera, se sabe donde se almacena toda la información y se pueden hacer consultas por medio de triples, haciendo uso de esos metadatos.

4.2.1.2 Paso 2, formato y envío de los datos

Una vez habiendo seleccionado la ontología que se va a emplear a lo largo de todo el ciclo de vida del proyecto, ya se tiene claro el formato que deberían adoptar los diferentes datos para que puedan darse coincidencias entre los valores de la información que el usuario envía al servidor y las clases de la ontología en cuestión.

Por lo tanto, para enviar al servidor los datos en el formato correcto, se ha decidido poner en el cliente los valores de la información a enviar con el mismo formato que la información seleccionada tiene en la propia ontología. A continuación, siguiendo el proyecto de prueba descrito en la Tabla 6, se va a mostrar el recorrido de los datos desde la parte cliente hasta el servidor.

Globales	Específicas	Valor literal
Nombre del proyecto	Proyecto de prueba	Proyecto de prueba
Número de participantes	4	4
Formación académica	Social sciences, business and law	Social_sciences_Business_and_Law__3
	Agriculture and veterinary	Agriculture_and_Veterinary__6
Lenguas conocidas	Inglés	eng__English
	Castellano	spa__Spanish_Castilian
Lengua materna	Inglés	eng__English
Competencias profesionales	Professionals	PROFESSIONALS__2
	Technicians and associate professionals	TECHNICIANS_AND_ASSOCIATE_PROFESSIONALS__3
Habilidades personales	Social skills	Social_Skill
	ICT skills	ICT_Skill

Tabla 6: Valores literales enviados al servidor del proyecto de prueba

Todos estos datos se envían en formato `HttpServletRequest` y se almacenan en un *array* de *Strings*, del cual se pueden ir obteniendo mediante el *tag name* que se le haya impuesto a dicho atributo.

Todos estos datos, como se puede ver en la Tabla 6, están divididos en distintos apartados, dependiendo a la competencia global a la que hacen referencia. Como se va a explicar más adelante, esta distinción se realiza dentro del propio núcleo del sistema, además de en la propia interfaz. Esto es así para posibilitar el procesamiento de los datos dentro de la aplicación, ya que de otra forma, realizar una consulta sobre dicho elemento resultaría imposible.

4.2.1.3 Paso 3, procesamiento de los datos

Una vez los datos han sido recogidos por la parte servidor de la aplicación, empieza el proceso que hace que el sistema de recomendación funcione. Este es un proceso complejo y que se divide en varios módulos, ya que tiene que pasar por varios niveles de procesamiento para que esta información sea realmente útil

para el sistema.

La estructura de datos seleccionada para realizar todo el procesamiento de los datos ha sido `Vector<E>` (`java.util.Vector<E>`). De cara a la implementación actual del sistema, no era de gran relevancia la estructura de datos a emplear ya que, lo que se hace es rellenar la estructura pertinente con todos los elementos que se van a utilizar y por lo tanto, hay que iterar sobre todos ellos con lo que, independientemente de la estructura de datos que se emplease, el orden va a ser $O(n)$. Además, actualmente no se manejan una gran cantidad de datos dentro de la aplicación por lo que, actualmente, la cantidad de memoria que ocupan es muy pequeña como para que sea señalable, pero en un futuro es posible que haya que manejar cantidades de datos mucho mayores y, la clase `Vector` permite especificar el tamaño del crecimiento del vector cada vez que este llega a su capacidad máxima. Los datos que van a llegar al servidor son limitados y se puede conocer el máximo de los mismos que pueden llegar a enviarse desde el cliente, con lo que se puede calcular el factor de crecimiento adecuado para la estructura de datos dependiendo de lo que interese (memoria o velocidad).

Dependiendo de si lo que se está generando es el perfil de un usuario nuevo o el perfil de un nuevo proyecto, el proceso es diferente. Esto se debe a que la información del usuario se guarda directamente, sin ningún paso intermedio; mientras que la de los proyectos, antes de almacenarse en la base de datos, se almacena en un vector dentro del núcleo del sistema con la información remitida por el cliente. La información se almacena en los vectores de la siguiente manera: (*String* competencia, *double* puntuaciónCompetencia); y la competencia sigue el siguiente formato: *competenciaGlobal* + '&' + *competenciaEspecífica*. De esta forma, se puede determinar a que competencia global pertenece cada uno de los datos que se reciben en el servidor, y esto es necesario para poder hacer las consultas SPARQL a la base de datos.

La decisión de realizar el almacenamiento de los datos de esta forma en los vectores se tomó con el fin de agilizar el proceso de consulta que se realiza inmediatamente después. De esta forma, al almacenar los datos recogidos del cliente en las estructuras de datos utilizadas para realizar el procesamiento de estos a medida que se reciben en el servidor y a la par que se almacenan en la base de datos, el sistema se ahorra el paso de buscar este proyecto en la base de datos para realizar la consulta pertinente.

Una vez se tienen todos los datos enviados por el cliente, con el formato señalado previamente, en la estructura de datos almacenados, se procede a realizar la consulta sobre la base de datos. La consulta en SPARQL tiene la siguiente forma:

```

1 SELECT * WHERE {
2   {
3     SELECT ?x ?y WHERE {
4       ?j jobSeeker:email ?x .
5       ?j education:Name ?y
6       FILTER (?y="Social_sciences_Business_and_Law_3"^^xsd:string || ?y="Agriculture_and_Veterinary_6"^^xsd:string )
7     }
8   }
9   UNION
10  {
11    SELECT ?x ?y WHERE {
12      ?j jobSeeker:email ?x .
13      ?j language:Name ?y
14      FILTER (?y="spa_Spanish_Castilian"^^xsd:string )
15    }
16  }
17  UNION
18  {
19    SELECT ?x ?y WHERE {
20      ?j jobSeeker:email ?x .
21      ?j occupation:Name ?y
22      FILTER (?y="PROFESSIONALS_2"^^xsd:string || ?y="TECHNICIANS_AND_ASSOCIATE_PROFESSIONALS_3"^^xsd:string )
23    }
24  }
25  UNION
26  {
27    SELECT ?x ?y WHERE {
28      ?j jobSeeker:email ?x .
29      ?j skill:Name ?y
30      FILTER (?y="Social_Skill"^^xsd:string || ?y="ICT_Skill"^^xsd:string )
31    }
32  }
33 }

```

donde *competenciaGlobal* (education, language, occupation y skill) representa el prefijo del predicado que es, a su vez, la cadena de caracteres previa al símbolo '&' de toda la cadena almacenada en el vector de competencias, y la competencia global a la que está haciendo referencia (en este caso, a las habilidades, a las competencias profesionales, a la formación académica o a los lenguajes conocidos); y donde *competenciaEspecífica* hace referencia a la competencia específica dentro de las globales (por ejemplo, en *language*, *competenciaEspecífica* tendría como valor *spa_Spanish_Castilian*) y es la cadena de caracteres que se encuentra después del símbolo '&'. Esta consulta devolverá toda la lista de personas que posean, al menos, alguna de las competencias que se hayan solicitado en la descripción del proyecto, junto con la competencia que se estuviese consultando en ese momento.

x	y
manaject@gmail.com	Agriculture_and_Veterinary_6
manaject@gmail.com	Social_sciences_Business_and_Law_3
antonio@gmail.com	Agriculture_and_Veterinary_6
antonio@gmail.com	Social_sciences_Business_and_Law_3
natalie@gmail.com	Agriculture_and_Veterinary_6
prueba1@gmail.com	Agriculture_and_Veterinary_6
prueba1@gmail.com	Social_sciences_Business_and_Law_3
prueba1@gmail.com	spa_Spanish_Castilian
antonio@gmail.com	TECHNICIANS_AND_ASSOCIATE_PROFESSIONALS_3
antonio@gmail.com	PROFESSIONALS_2
joseteRodriguez@gmail.com	PROFESSIONALS_2
prueba1@gmail.com	TECHNICIANS_AND_ASSOCIATE_PROFESSIONALS_3

Figura 19: Resultado de Project Seekers con alguna de las competencias del proyecto

Pero eso no es suficiente, ya que lo que se busca es la lista de los i candidatos más competentes para dicho proyecto.

Con estos resultados, obtenemos una lista de *Strings* con un formato que sigue la siguiente estructura: [x="emailDeLaPersona";y="competencias "]. Esta lista de cadenas de caracteres recién obtenida, es procesada para obtener una lista con elementos de la clase *Person* (que tiene un identificador en forma de *String* y una lista de competencias). De esta forma, se agrupan las competencias de cada persona utilizando sus emails como identificadores, y resulta más fácil obtener la puntuación de cada persona empleando cualquiera de las técnicas que se van a describir más adelante. Habiendo procesado los datos de esta forma, ahora es posible obtener las listas que se van a emplear para asignar la puntuación a los *Project Seeker*; que son, la lista de las personas con las competencias que posee cada una de ellas y la lista de competencias con la puntuación asignada por el *Project Head* a cada una de ellas.

A continuación se procede a explicar las distintas funciones que se han empleado para calcular la puntuación de cada *Project Seeker* en los diferentes proyectos.

4.2.1.3.1 Algoritmos empleados para el sistema de recomendación

Para determinar la fiabilidad de un sistema de recomendación, es importante conocer las distintas técnicas en los que se basa para realizar los emparejamientos, y los credenciales de las mismas para saber que dichos algoritmos son efectivos. Este sistema de recomendación ha utilizado dos de las técnicas de recomendación más empleadas: similitud del coseno y coeficiente de similitud de Jaccard [13], con las que se determina que dos muestras son

similares si sus vectores son cercanos el uno del otro.

Para hacer una comparación entre las distintas técnicas que se van a presentar a continuación, se va a generar un proyecto con las competencias que figuran en la Tabla 7, y en cada caso, se va a mostrar los resultados que ofrece el sistema de recomendación.

Globales	Específicas	Valor
Nombre del proyecto	Proyecto de prueba	-
Número de participantes	4	-
Formación académica	Social sciences, business and law	50
	Agriculture and veterinary	85
Lenguas conocidas	Inglés	69
	Castellano	21
Lengua materna	Inglés	-
Competencias profesionales	Professionals	12
	Technicians and associate professionals	57
Habilidades personales	Social skills	25
	ICT skills	53

Tabla 7: Competencias de proyecto de prueba

Para poder explicar detalladamente como se han aplicado las siguientes técnicas al sistema, hay que tener en cuenta ciertos detalles de implementación. Estas técnicas en la aplicación de **Manaject** han recibido como parámetros dos listas de datos. Una de ellas contiene los nombres de las competencias que posee la persona que se está evaluando en el momento, y la otra contiene los nombres de las competencias del proyecto, junto con el valor que se les ha asignado a cada una de ellas por el *Project Head*. Todas las operaciones de unión e intersección de entre las competencias que poseen las dos listas se realizan mediante el método *contains(Object elem)* de la clase *Vector*. Como esta persona ya ha sido seleccionada anteriormente por el sistema como candidata, por el hecho de que tenía alguna de las competencias señaladas, se aplica este método sobre el *Vector*

de los nombres de las competencias que posee la persona, pasándole como parámetro uno a uno los nombres de las competencias que están en el Vector continente de las competencias requeridas por el proyecto.

Sumatorio

La técnica del sumatorio no es una de las técnicas sugeridas para realizar recomendaciones. Se ha utilizado a lo largo del ciclo de vida del proyecto ya que los resultados que iba a devolver son predecibles y eso ha ayudado a determinar que la inserción de datos, tanto en la base de datos, como en el sistema, era correcta. De forma que, si una persona posee una competencia en concreto y en el proyecto se exige esa competencia, a la puntuación de esa persona solo habría que sumarle el valor que le ha dado la persona encargada de crear dicho proyecto (*Project head*).

Sea $p \in P$ donde P es el conjunto de todas las personas y sea $q \in Q$ donde Q es el conjunto de todos los proyectos

Sea $p = (p_1, p_2, \dots, p_n)$ donde $\forall p_i \in \mathbb{N}_0: 0 \leq p_i \leq 1$; $p_i=0$ si la persona no posee la competencia i de las n competencias que se piden en la descripción del proyecto, y $p_i=1$ si la posee. Y, sea $q = (q_1, q_2, \dots, q_n)$, $\forall q_i \in \mathbb{R}: 0 \leq q_i \leq 1$ donde q_i es el valor que le ha dado el *Project Head* a la competencia i dividido entre 100, entonces:

$$r = \sum_{i=1}^n q_i p_i \text{ donde } r \text{ es la puntuación obtenida por } p \text{ para el proyecto } q.$$

El resultado obtenido por el sistema de recomendación se refleja en la figura.

Personas recomendadas

prueba1@gmail.com: 3.7199999999999998

antonio@gmail.com: 3.51

ibanm@gmail.com: 3.07

manaject@gmail.com: 2.82

Figura 20: Personas recomendadas por la técnica de suma

Coefficiente de similitud de Jaccard

La técnica del coeficiente de similitud de Jaccard (también conocido como índice Jaccard) es una de las técnicas más empleadas en los distintos sistemas de

recomendación que no recogen muchos detalles para calcular la cercanía de los vectores. Es una técnica efectiva cuando no hay muchos detalles implícitos, ya que si no pierde información, como señala [15]. El índice Jaccard mide la similitud entre dos conjuntos de muestras finitos, y está definido como el tamaño de la intersección dividido por el tamaño de la unión de los conjuntos de muestra.

Sea $p \in P$ donde P es el conjunto de todas las personas y sea $q \in Q$ donde Q es el conjunto de todos los proyectos.

Entonces: $J = \frac{|\vec{p} \cap \vec{q}|}{|\vec{p}| + |\vec{q}| - |\vec{p} \cap \vec{q}|}$ donde J es el coeficiente de similitud de Jaccard.

Cuanto más se aproxime el valor de J a 1, más cercanos serán los vectores.

El resultado que se obtiene aplicando esta técnica es el siguiente.

Personas recomendadas

antonio@gmail.com: 0.5830404577936115

prueba1@gmail.com: 0.5830404577936115

ibanm@gmail.com: 0.4739722951928812

manaject@gmail.com: 0.4736902200757034

Figura 21: Personas recomendadas por la técnica del índice de Jaccard

Similitud del coseno

La técnica de la similitud del coseno también es una de las técnicas que más se utiliza para medir la similitud entre un conjunto finito de muestras. Esta, ha demostrado ser un buen método para calcular la similitud entre dos vectores, calculando el coseno del ángulo que forman. Si se calcula el coseno de 0° , el resultado es 1, y es el único caso en el que esto es así, ya que para cualquier otro ángulo el resultado siempre será menor que 1. Por lo que, la similitud de los dos vectores que están en el conjunto de muestra se basa en la orientación de los mismos.

Sea $p \in P$ donde P es el conjunto de todas las personas y sea $q \in Q$ donde Q es el conjunto de todos los proyectos

Sea $p = (p_1, p_2, \dots, p_n)$ donde $\forall p_i \in \mathbb{N}_0: 0 \leq p_i \leq 1$; $p_i = 0$ si la persona no posee la competencia i de las n competencias que se piden en la descripción del proyecto, y $p_i = 1$ si la posee. Y, sea $q = (q_1, q_2, \dots, q_n)$, $\forall q_i \in \mathbb{R}: 0 \leq q_i \leq 1$ donde q_i es el valor que le ha dado el *Project Head* a la competencia i dividido

entre 100, entonces:

Entonces:
$$s = \frac{\sum_{i=1}^n p_i q_i}{\sqrt{\sum_{i=1}^n p_i^2} \sqrt{\sum_{i=1}^n q_i^2}}$$
 donde s es el valor de la similitud del coseno.

El resultado que se obtiene aplicando esta técnica es el siguiente:

Personas recomendadas

antonio@gmail.com: 1.819782999012336

prueba1@gmail.com: 1.783439494628622

ibanm@gmail.com: 1.6346541951920446

manaject@gmail.com: 1.4889837379118658

Figura 22: Personas recomendadas por la técnica de similitud del coseno

En esta técnica, a diferencia de la del coeficiente de similitud de Jaccard, se ha trabajado sobre el conjunto de muestra de las competencias que posee la descripción del proyecto, en vez de sobre la unión de ambos conjuntos. Esto se debe a que, en esta última técnica, si se hubiese utilizado la unión de ambos conjuntos, las personas con más competencias en sus perfiles (de las que no están en la descripción del proyecto sobre el que se está realizando la evaluación), tendrían un resultado inferior a una persona que poseyese las mismas competencias descritas en el proyecto que la primera persona, pero menos competencias fuera de este marco.

La técnica que se ha seleccionado como base del sistema para calcular las puntuaciones de los *Project Seeker* ha sido la técnica de similitud del coseno, ya que, como se ha mencionado en el apartado correspondiente a la técnica del coeficiente de similitud de Jaccard, es una técnica efectiva si no hay muchos detalles implícitos.

Las personas que aparecen en las Figuras 20, 21 y 22 son un subconjunto del conjunto de personas que hay introducidas en la base de datos. Estas personas fueron introducidas en la base de datos con una serie de competencias, elegidas de forma aleatoria, y son las personas que han sido seleccionadas por el sistema de recomendación aplicando las distintas técnicas descritas anteriormente, para un proyecto con las competencias y los valores de las mismas que se muestran en la Tabla 7. Para ver las competencias que poseen cada una de las personas

seleccionadas por la aplicación, ver la sección de Anexo.

4.3 Tecnologías empleadas

4.3.1 Cliente

Para llevar a cabo la implementación de la parte cliente de la aplicación, se han empleado diversas tecnologías con el fin de mejorar, en la medida de lo posible en el marco de la aplicación, la experiencia del usuario. La parte cliente de la aplicación está constituida en su plenitud por la interfaz web, como ya se ha visto anteriormente en este documento.

Dicho esto, es evidente que todas las tecnologías que se han empleado para construir la parte cliente de la aplicación son tecnologías web. A continuación se presenta una lista con las que se han utilizado y el rol que han ejercido cada una de ellas:

1. HTML5

Hay múltiples razones por las que se eligió esta tecnología como la base para crear la interfaz web con la que el usuario puede interactuar con el sistema, pero las razones principales fueron la sencillez del lenguaje y su adaptabilidad. HTML5, al ser un lenguaje muy extendido y muy popular a la hora de generar páginas web, se puede encontrar fácilmente ayuda en la web. Además de eso, todos los navegadores modernos y populares (Chrome, Firefox, Safari, IE9 y Opera) pueden utilizarlo.

2. JQuery

JQuery es una librería de JavaScript que permite manipular el árbol DOM de la página html. Esto también puede hacerse mediante JavaScript puro pero resulta mucho más sencillo hacerlo mediante esta librería. En esta aplicación, se utiliza tanto para desplegar las distintas opciones que ofrece una competencia global al pinchar con el ratón sobre ella, como para asegurar que el *Project Head* no pueda introducir el valor a una competencia que no ha solicitado para la búsqueda de personal.

4.3.2 Servidor

El servidor es la parte de la aplicación que se encarga de recoger la información enviada por el cliente y pasar esta información al núcleo del sistema para que este la procese y le devuelva un resultado; así como de dar respuesta a las solicitudes del cliente. Para implementar la parte servidor de la aplicación se ha utilizado el lenguaje de programación Java y como servidor web, se ha utilizado Apache

Tomcat.

1. Java

Java es un lenguaje de programación orientado a objetos y la decisión de emplearlo en la parte servidor de la aplicación fue tomada debido a que el núcleo de la aplicación también está en Java, y resultaba mucho más sencilla la comunicación entre ambos módulos estando en el mismo lenguaje.

2. Apache Tomcat

Apache Tomcat es un servidor web *open source* que implementa varias especificaciones de JavaEE entre las cuales se encuentra Java Servlet. Esta es una clase de Java que permite la interacción con el cliente, permitiendo recoger las solicitudes y enviar respuestas.

4.3.3 Base de datos

La base de datos semántica va a ser donde se almacenen los grafos que contienen los diferentes triples que almacenan toda la información de los usuarios. La base de datos semántica escogida para ello ha sido Stardog por diversas razones, a continuación se va a presentar una lista con las mismas.

1. Esta implementada puramente en java. Hay ciertas librerías necesarias para poder realizar la conexión desde el sistema directamente con la base de datos, así como para hacer consultas. El hecho de que esta esté implementada en java fue uno de los puntos clave por la que fue elegida.
2. Entre los métodos de acceso a la información almacenada en la base de datos se encuentra SPARQL, que es el lenguaje de consulta que se va a emplear para realizar las actualizaciones como las consultas propiamente dichas.
3. Contiene documentación de fácil acceso y explicativa. Gracias a la misma, fue posible hacer la conexión del sistema con la base de datos sin mayor problema.

4.3.4 Núcleo del sistema

El núcleo del sistema va a ser donde se lleven a cabo todas las operaciones del sistema, como el procesamiento de datos, las consultas, las actualizaciones, etc, como ya se ha visto en este documento anteriormente. El lenguaje en el que se ha implementado fue Java ya que es uno de los lenguajes de programación que más apoya las tecnologías de Web Semántica con *frameworks* como Jena y Sesame.

En esta aplicación se tomó la decisión de utilizar Jena por recomendación, ya que la documentación de Sesame era poco explicativa e iba a suponer un coste adicional que no se estaba dispuesto a asumir. Jena es un *framework* que permite hacer las consultas y las actualizaciones pertinentes a la base de datos de manera sencilla y, tomando este como base, se han ido tomando todas las decisiones de las tecnologías a emplear en los demás módulos que constituyen el sistema.

Capítulo 5. Conclusiones

Los sistemas de recomendación son un recurso cada vez más presente en la vida cotidiana. Se pueden encontrar en distintas áreas y sus resultados pueden variar mucho de un sistema de recomendación a otro, pero aún siendo así, el objetivo que persiguen todos ellos es el mismo: ofrecer un *item* o una lista de *items* lo más personalizada posible a cada usuario, teniendo en cuenta diversos parámetros de entrada que será en los que se basará para exponer sus resultados.

Manaject es un componente de software basado en tecnologías semánticas que permite a los usuarios de la aplicación rellenar un perfil personal para que el sistema de recomendación les empareje con proyectos afines a sus aptitudes, así como introducir la descripción de un proyecto para que el sistema de recomendación devuelva una lista con las *i* personas más cualificadas.

Este proyecto perseguía un claro objetivo: facilitar la obtención de recursos humanos cuando se quiere realizar un proyecto. A pesar de que hubo muchos cambios en el alcance del proyecto, esta fue la meta principal del mismo y se mantuvo inamovible durante todo el ciclo de vida del proyecto. El hecho de que se realizase con tecnologías de la Web Semántica fue una decisión tomada más de cara al futuro de la aplicación que al presente, ya que en un futuro, las descripciones de proyectos realizadas para esta aplicación serán aprovechables para otras aplicaciones de la misma índole, utilizando el vocabulario apropiado. Esto trae consigo que se puedan utilizar agentes software para realizar todo tipo de búsquedas sobre estas aplicaciones, ya que toda la información tiene un significado mediante los predicados impuestos y estos son compartidos con aquellas aplicaciones con el mismo vocabulario. Los portales de trabajo que no utilizan la Web Semántica tienen cada uno un formato distinto, de forma que, al no seguir todos un mismo patrón, resulta imposible realizar un componente de software que trabaje sobre todos ellos. Sin embargo, la Web Semántica hace posible que los agentes software hagan búsquedas e interactúen con las aplicaciones que hacen uso de ella.

Desde el principio de la aplicación, se planteó la posibilidad de hacer pruebas con personas dispuestas a incluir sus competencias en la misma, de forma que se pudiesen meter los diferentes perfiles de estas personas para comprobar su funcionamiento en un marco más cercano a la vida real. Esta idea fue desestimada debido a unos tiempos de planificación desacertados, y hubo que utilizar esos recursos temporales en realizar actividades de mayor importancia

para el desarrollo de la aplicación.

Para realizar las pruebas de funcionamiento de la aplicación se tuvieron que incluir perfiles de personas ficticias, de forma que los resultados no resultan tan fiables.

5.1 Consideraciones

El proyecto **Manaject** ha sido una experiencia muy enriquecedora. Durante el ciclo de vida del proyecto, he aprendido diversas cosas. Personalmente, lo que más interesante me ha parecido ha sido todo lo relacionado con la Web Semántica. Es un concepto con mucho potencial implícito y las distintas aportaciones de las personas y las tecnologías que la rodean están haciendo de ello algo real.

Además de esto, a lo largo del ciclo de vida del proyecto, he aprendido las distintas técnicas y funciones que se utilizan hoy en día para realizar sistemas de recomendación *well-tested*, gracias a las personas que han ido haciendo publicaciones de artículos que mostraban la manera en la que han resuelto el problema de realizar recomendaciones con resultados positivos, mostrando los mismos.

Junto con esto y siendo más importante aún, he aprendido a realizar búsquedas de información en artículos con credenciales, guiándome por distintos criterios en la búsqueda y esto es algo que no solo me va a servir a nivel informático, sino en la vida laboral cotidiana, en cualquier ámbito de aplicación también.

Pero no todas las experiencias del proyecto han sido tan buenas, también ha habido diversos problemas con los que se ha tenido que lidiar. De entre todos los problemas que han surgido a lo largo del ciclo de vida del proyecto, cabe destacar uno que hizo que me diese cuenta de la importancia de las copias de seguridad para todos los archivos de cualquier proyecto que se esté llevando a cabo. Este problema de corrupción de metadatos hizo que toda la aplicación de **Manaject** se corrompiese y hubiese que borrarla y empezarla de nuevo; pero gracias a un análisis de riesgos acertado, la aplicación tenía una copia de seguridad en un repositorio en la nube y se dio muy poca pérdida de información.

Gracias a todo esto, me dí cuenta de la importancia de una metodología autodidacta (como la que se practica, en gran medida, en el *Proyecto de Fin de Grado*) ya que, cuando uno más aprende, es cuando se presenta un problema y los recursos de los que dispone son los que él mismo conoce.

“La parte más importante de la educación del hombre es aquella que él mismo se da” - Walter Scott

Capítulo 6. Bibliografía

- [1] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.
- [2] Villazón-Terrazas, B., Ramírez, J., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2011). A network of ontology networks for building e-employment advanced systems. *Expert Systems with Applications*, 38(11), 13612-13624.
- [3] Guía breve de Web Semántica. W3C. Disponible en <http://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica>.
- [4] Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21(3), 96-101.
- [5] Dean, M., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., ... & Stein, L. A. (2004). OWL web ontology language reference. W3C Recommendation February, 10.
- [6] SPARQL query language for RDF. W3C Recommendation, 2008. Disponible en <https://www.w3.org/TR/rdf-sparql-query/>.
- [7] Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., & Eckstein, R. (2005). The impact of semantic web technologies on job recruitment processes. In *Wirtschaftsinformatik 2005* (pp. 1367-1381). Physica-Verlag HD.
- [8] Dittmann, L., & Zelewski, S. (2004, April). Ontology-based skills management. In *Proc. of the 8th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2004)* (Vol. 4, pp. 190-195).
- [9] Dorn, J., Naz, T., & Pichlmair, M. (2007, August). Ontology development for human resource management. In *4th International Conference on Knowledge Management* (pp. 109-120).
- [10] García-Sánchez, F., Martínez-Béjar, R., Contreras, L., Fernández-Breis, J. T., & Castellanos-Nieves, D. (2006). An ontology-based intelligent system for recruitment. *Expert Systems with Applications*, 31(2), 248-263.
- [11] Malinowski, J., Keim, T., Wendt, O., & Weitzel, T. (2006, January). Matching people and jobs: A bilateral recommendation approach. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on* (Vol. 6, pp. 137c-137c). IEEE.

[12] Lu, Y., El Helou, S., & Gillet, D. (2013, May). A recommender system for job seeking and recruiting website. In Proceedings of the 22nd international conference on World Wide Web companion (pp. 963-966). International World Wide Web Conferences Steering Committee.

[13] A. Rajaraman y J. D. Ullman. Mining of Massive Datasets. Cambridge University Press, 2011. Chapter 9.

[14] K. Kunzelma, S. Hutto, "Common session token system and protocol" U.S. Patent 6041357 A, MAR 21, 2000.

ANEXO

En este anexo se van a mostrar las diferentes competencias que poseen los usuarios con el rol de *Project Seeker*, almacenados en la base de datos de STARDOG.

Globales	Específicas
Formación académica	Social sciences, business and law
	Health and welfare
	Engineering, manufacturing and construction
	Education
	Services
	Agriculture and veterinary
Lenguas conocidas	Inglés
	Alemán
Lengua materna	Alemán
Competencias profesionales	Armed forces
	Legislators, senior officials and managers
	Service workers and shop and market sales workers
	Elementary occupations
Habilidades personales	Social skills
	Technical skills
	ICT skills

Tabla 8: competencias del usuario manaject@gmail.com

Globales	Específicas
Formación académica	Social sciences, business and law
	Agriculture and veterinary
Lenguas conocidas	Inglés
	Castellano
Lengua materna	Castellano
Competencias profesionales	Technicians and associate professionals
	Professionals
Habilidades personales	Social skills
	ICT skills

Tabla 9: competencias del usuario prueba1@gmail.com

Globales	Específicas
Formación académica	Agriculture and veterinary
Lenguas conocidas	Francés
Lengua materna	Francés
Competencias profesionales	Skilled agricultural and fishery workers
Habilidades personales	Social skills

Tabla 10: competencias del usuario natalie@gmail.com

Globales	Específicas
Formación académica	Health and welfare
	Education
	Services
Lenguas conocidas	Italiano
Lengua materna	Alemán
	Italiano
Competencias profesionales	Service workers and shop and market sales workers
Habilidades personales	ICT skills

Tabla 11: competencias del usuario user1@gmail.com

Globales	Específicas
Formación académica	Social sciences, business and law
	Science, mathematics and computing
	General programmes
	Services
	Agriculture and veterinary
Lenguas conocidas	Inglés
Lengua materna	Inglés
Competencias profesionales	Technicians and associate professionals
	Legislators, senior officials and managers
	Plant and machine operators and assemblers
	Professionals
	Elementary occupations
Habilidades personales	Social skills
	Technical skills
	ICT skills

Tabla 12: competencias del usuario antonio@gmail.com

Globales	Específicas
Formación académica	Social sciences, business and law
	Science, mathematics and computing
	General programmes
	Services
	Agriculture and veterinary
Lenguas conocidas	Inglés
	Italiano
Lengua materna	Inglés
Competencias profesionales	Legislators, senior officials and managers
	Professionals
	Elementary occupations
Habilidades personales	Organisational skills
	ICT skills

Tabla 13: competencias del usuario joseteRodriguez@gmail.com

Globales	Específicas
Formación académica	Humanities and arts
	General programmes
	Education
	Agriculture and veterinary
Lenguas conocidas	Alemán
	Francés
Lengua materna	Alemán
Competencias profesionales	Craft and related trades workers
	Plant and machine operators and assemblers
	Clerks
Habilidades personales	Organisational skills
	Artistic skills

Tabla 14: competencias del usuario josu@gmail.com

Globales	Específicas
Formación académica	Science, mathematics and computing
	Social sciences, business and law
	Agriculture and veterinary
Lenguas conocidas	Italiano
	Castellano
	Inglés
Lengua materna	Italiano
Competencias profesionales	Legislators, senior officials and managers
	Technicians and associate professionals
	Plant and machine operators and assemblers
	Elementary occupations
Habilidades personales	Social skills
	Technical skills

Tabla 15: competencias del usuario ibanm@gmail.com

Globales	Específicas
Formación académica	Science, mathematics and computing
	Humanities and arts
	Agriculture and veterinary
Lenguas conocidas	Francés
Lengua materna	Francés
Competencias profesionales	Armed forces
	Skilled agricultural and fishery workers
Habilidades personales	Artistic skills

Tabla 16: competencias del usuario diego@gmail.com

Globales	Específicas
Formación académica	Services
	General programmes
	Agriculture and veterinary
Lenguas conocidas	Castellano
Lengua materna	Castellano
Competencias profesionales	Clerks
	Professionals
	Service workers and shop and market sales workers
	Skilled agricultural and fishery workers
Habilidades personales	Artistic skills

Tabla 17: competencias del usuario john@gmail.com