Universitat Oberta de Catalunya

Desarrollo de un Gestor (buscador) de documentación intrahospitalaria aplicando técnicas de Web Semántica

Autor: Jaime Buera Agraz

Tutor: Felipe Geva Urbano.

Profesor: Jordi Conesa Caralt

Grado de ingeniería Informática

Itinerario de sistemas de información

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desarrollo de un gestor de documentación con web semántica*
Nombre del autor:	*Jaime Buera Agraz*
Nombre del colaborador/a docente :	*Felipe Geva Urbano*
Nombre del PRA:	*Jordi Conesa Caralt*
Fecha de entrega (mm/aaaa):	*01/2018*
Titulación o programa:	*Grado informática*
Área del Trabajo Final:	*TFG Web semántica*
Idioma del trabajo:	*Español*
Palabras clave	*Gestor documental semántica*

Resumen del Trabajo (máximo 250 palabras):

El trabajo final ha sido creado para resolver la gestión de documentos intrahospitalarios y servirlos en cualquier lugar en el entorno de trabajo.
La aplicación consta de un buscador semántico el cual ofrece de forma rápida y sencilla las búsquedas orientadas a servicios, categorías y a los propios documentos. Para conseguir el objetivo final se ha realizado en distintas etapas. La metodología utilizada para crear la aplicación ha sido la realización de las etapas de conceptualización, adquisición del conocimiento, integración, implantación, evaluación y documentación.
Los resultados obtenidos han sido satisfactorios ya que el producto final es una herramienta muy intuitiva y efectiva sobre la búsqueda de documentos por parte del usuario.
Como conclusión podemos decir que la experiencia que puedo sacar es que las tecnologías actuales hacen más fácil el trabajo y sobre todo, dependiendo del campo en que se trabaja, puede llevar a cabo una misión tan delicada como la de incluso salvar vidas en determinados casos. El seguimiento de un paciente realmente puede ser efectivo si el especialista reúne toda la información necesaria en cualquier momento bajo una ordenación y sencillez.

Abstract (in English, 250 words or less):

The final work has been created to resolve the management of intrahospital documents and serve them anywhere in the work environment.
The application consists of a semantic search engine which offers quickly and easily the searches oriented to services, categories and the documents themselves. To achieve the final objective, it has been carried out in different stages. The methodology used to create the application has been the realization of the stages of conceptualization, knowledge acquisition, integration, implementation, evaluation and documentation.
The results obtained have been satisfactory since the final product is a very intuitive and effective tool on the search of documents by the user.
In conclusion we can say that the experience I can get is that current technologies make work easier and above all, depending on the field in which you work, you can carry out a mission as delicate as saving even lives in certain cases. The monitoring of a patient can really be effective if the specialist gathers all the necessary information at any time under an order and simplicity.

04/10/2017

Palabras clave

Herramienta, documento, buscador, parámetro, servicios, organización, web, semántica, protocolo, hospitalaria, plantilla, guía, consentimiento

KeyWords

Tool, document, browser, parameter, services, organization, web, protocol, hospital, template, guide,assent

Resumen de la propuesta

El trabajo final consiste en desarrollar un gestor documental basado en la tecnología java con incorporación de técnicas de web semántica [1,2] para organizar y acceder a la información, siendo interrogado por un gestor de BB.DD capaz de interpretar el código semántico [4] y como es capaz de realizar búsquedas sobre material documental. Es importante destacar que el código será estructurado mediante XML y RDF [3] para que sea entendible sobre el motor de búsqueda semántico. También se utilizará el Lenguaje de Ontologías Web (OWL) [5,6] que está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. Por lo tanto, podremos combinar RDF y OWL como estándares de la Web Semántica que proporcionan una infraestructura para la gestión e integración de datos en la Web. Estos formatos estándar para la compartición de datos amplían las fronteras de aplicaciones, así tipos diferentes de "usuario" pueden compartir la misma información sobre los documentos a buscar.

El material que incorporará nuestra herramienta, es decir, el buscador, será material que constará de documentos como plantillas, protocolos, guías y consentimientos…
Estos documentos son imprescindibles para que el personal de la organización puedan cumplir con los requisitos de la empresa, ya que dichos documentos forman parte de un proceso tanto clínico como administrativo.
Su uso cotidiano hace necesario que la organización sea escrupulosa a la hora de poder ofrecer los documentos para cualquier profesional que los necesite, siendo así una herramienta imprescindible.

Por lo tanto, el buscador será una herramienta potente, capaz de realizar búsquedas exactas de documentos de cualquier servicio de la organización diferenciando el tipo de documento (guías, protocolos, plantillas y consentimientos). Si bien es verdad, que la información sobre que tipo de documento es necesario para el usuario será mediante la búsqueda, pero también será posible identificar mediante el propio registro del usuario en la aplicación el que pueda ofrecer información de su naturaleza, pudiendo obtener información acerca del servicio que proviene, del ámbito en que ejerce, etc…
El buscador permitirá el uso de varios parámetros para mejorar el resultado, siendo estos parámetros los siguientes:

- Por nombre de documento.
- Por contenido en el documento.
- Por servicio.
- Por tipo de documento.

El interface principal para el usuario contendrá un formulario para poder realizar las búsquedas con los parámetros que se han mencionado. Estos resultados serán ofrecidos sobre una tabla indicando información del documento que además contendrá un enlace para poder acceder al él. Cuando el documento sea encontrado y abierto, el usuario podrá utilizarlo de forma que siempre estará limitado a lectura para que los datos nunca se puedan modificar bajo ningún concepto de forma local.

Justificación y motivación

Actualmente trabajo en un Hospital como informático, desempeñando todo tipo de tareas como puede ser sobre sistemas, programación, ofimática, redes, etc…
El proyecto que actualmente se va a elaborar nace partiendo del impedimento de enfermería a la hora de poder utilizar protocolos y consentimientos de forma eficaz, y lo que es más importante, documentos que sean “oficiales”, es decir, que presenten un formato elaborado por un órgano regulador como puede ser una comisión. A raíz de esta necesidad surge la idea de poder elaborar una herramienta capaz de servir cualquier tipo de documento con un buscador potente y especializado.

En el ámbito en el que trabajo existe una desorganización sobre el tema de documentos como son las plantillas, protocolos, guías y consentimientos. Todos los servicios tienen este

tipo de documentos, de los cuales pueden ser comunes o propios, por lo tanto, existen muchos documentos que deben estar disponibles en cualquier momento para el personal y además accesibles de manera que su búsqueda no resulte una pérdida de tiempo o incluso inútil por la desorganización.
Por lo tanto, la motivación que me lleva a realizar este proyecto es la de poder ofrecer un servicio o herramienta capaz de realizar búsquedas simples y rápidas, otorgando un documento al instante con una simple búsqueda coherente. Esto es importante ya que los profesionales pierden mucho tiempo en poder buscar documentos en carpetas organizadas por servicios que además están divididas por tipo de documento y que no ofrecen una organización por nombre documental, si no que el nombre en muchas ocasiones no sigue un “standard” y es difícil su búsqueda.
A esta dificultad existe un añadido y es que en el ámbito hospitalario existen muchos movimientos de personal lo que hace que una persona nueva esté completamente perdida ante la ineficacia de encontrar el tipo de documento que necesita, con la consiguiente pérdida de tiempo y retraso en sus labores.

También hay que tener en cuenta que los documentos deben ser supervisados por una comisión para poder ser utilizados, algo que actualmente en la organización no se está realizando ya que son documentos que se quedan alojados de forma local por los Pcs y llevan a quedarse obsoletos e incluso mal configurados por cambios de los usuarios.
Así pues, podemos cumplir una doble misión por la organización y por la estructuración de los documentos hacia el usuario final.

Y por último, una motivación extra es el campo de la semántica. Sobre todo en buscadores cualificados, es un tema que me interesa para empezar a conocer y que mejor que realizarlo en mi entorno de trabajo para poder iniciarme.
Si bien es verdad que la web semántica puede abarcar infinidad de proyectos, la iniciación sobre el tema puede abrir un gran abanico de posibilidades para abarcar en el futuro más inmediato en proyectos más ambiciosos y que pueden repercutir en la organización con mayor calaje. Es por eso que la motivación sobre algo que nunca se ha trabajado es un extra a la hora de realizarlo y ponerlo en marcha.

Agradecimientos

Realizar este proyecto me ha supuesto un gran esfuerzo, pero no solo realizar en si el proyecto, si no el conseguir llegar hasta aquí. Es difícil cuantificar el esfuerzo de tantos años de trabajo, pero la verdad es que el sacar adelante unos estudios mientras uno trabaja se hace realmente duro.

Por eso quiero agradecer a todos los amigos que han estado ahí en los momentos difíciles dando su apoyo incondicional, su tiempo para escuchar y sus ánimos para levantar la moral.

Gracias también a toda mi familia, en especial a mi madre Josefina y a mi novia Laura, que son las personas que más fuerza me han dado para seguir hacia delante en los momentos más difíciles.

Y por último agradecer a mi tutor Felipe Geva por ayudarme y guiarme durante todo el proyecto.

A todos Gracias.

Índice

1. Estado del Arte 10
1.1 Introducción al Gestor documental basado en Web semántica 10
1.2 Web semántica. 10
1.3 Estructura Gestor Documental 11
1.4 Composición del Gestor Documental 12
1.5 Funcionalidad 13
2. Objetivos y alcance 14
2.1 Objetivos 14
2.2 Objetivos del proyecto 14
2.3 Fases de realización 15
2.4 Estructura del documento 15
3. Metodología. Ontologías 16
3.1 Introducción 16
3.2 Características de las ontologías 17
3.3 Clasificación de las ontologías 18
3.4 Uso de las ontologías 19
3.5 Utilización de las ontologías 20
3.6 Ventajas de las ontologías 20
3.7 OWL (Web Ontology Language) 21
3.8 SPARQL (Lenguaje de consulta para RDF) 21
3.9 Estructura de consultas SPARQL 22
3.10 Formas de consultas 23
4. Planificación 23
4.1 Productos obtenidos. 25
4.2 Descripción de capítulos 25
5. Definición de las necesidades 26
5.1 Estudio inicial 26
5.2 Estructura global 27
5.3 Funcionamiento aplicación 27

5.4 Funcionamiento de la ontología sobre la aplicación 29
6. Estructura del hardware 30
6.1 Componentes 30
7. Sistema operativo 31
8. Software del nucleo base 31
8.1 Tomcat 7 31
8.2 MySQL 5 32
8.3 Apache 2.0 32
8.4 Java JDK 1.7 32
9. Análisis 33
9.1 Actores 33
9.2 Diagrama de casos de uso 33
9.3 Descripción funcionamiento del diagrama de casos de uso 33
10. Entorno visual de la aplicación 34
11. Diseño 36
11.1 Arquitectura aplicación 36
11.2 RichFaces 38
11.3 Rutas desarrollo aplicación 40
11.4 Información de las clases 40
11.5 Acceso a datos 40
11.6 Construcción de la ontología en la aplicación 41
12. Diagrama de clases 44
13. Herramientas de programación 44
14. Construcción aplicación 45
15. Puesta en marcha 45
16. Valoración económica 46
17. Conclusiones 47
18. Glosario 48
19. Bibliografía 50
20. Anexos 51

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las cuales aparecen. Para actualizar cada uno de los índices, hay que hacer botón derecho con el ratón y escoger la opción “Actualizar campos”.

Índice de figuras

Ilustración 1: Web Semántica 10
Ilustración 2: Web 3.0 ..... **¡Error! Marcador no definido.**11
Ilustración 3: Diagrama RDF ..... **¡Error! Marcador no definido.**12
Ilustración 4: Buscador ..... **¡Error! Marcador no definido.**12
Ilustración 5: Resultados ..... **¡Error! Marcador no definido.**13
Ilustración 6: Java/JSP ..... **¡Error! Marcador no definido.**25
No se encuentran elementos de tabla de ilustraciones. Ilustración 8: Buscador Gestor Documental 28
Ilustración 9: Resultados Gestor Documental 28
Ilustración 10: Información detallada resultado 28
Ilustración 11: Diagrama casos de uso 33
Ilustración 12: Entorno visual 1 34
Ilustración 13: Entorno visual 3 35
Ilustración 14: Entorno visual 3 35
Ilustración 15: Entorno visual 4 36
Ilustración 16: FrameWork RichFaces 36
Ilustración 17: ecnología richfaces ..... **¡Error! Marcador no definido.**39
Ilustración 18: Sparql ..... **¡Error! Marcador no definido.**41
Ilustración 19: Clases ..... **¡Error! Marcador no definido.**42
Ilustración 20: Instancia clase Documento ..... **¡Error! Marcador no definido.**43

No se encuentran elementos de tabla de ilustraciones. Ilustración 22: Aplicación 45

Índice de tablas

Tabla 1: Estructura actividades ..... **¡Error! Marcador no definido.**23
Tabla 2: Duración actividades ..... **¡Error! Marcador no definido.**24
Tabla 3: Diagrama de Gantt 25
Tabla 4: Valoración económica 46

1. Estado del Arte

1.1 Introducción al Gestor documental basado en Web semántica

El gestor documental de documentos intrahospitalarios basados en la técnica de web semántica, es un buscador avanzado el cual permitirá localizar de forma ágil y sencilla documentos necesarios para el usuario.
Una de las partes clave del buscador será el uso de técnicas de web semántica, que permitirá estructurar la información para enriquecer las búsquedas.

1.2 Web semántica.

La **Web Semántica** *(Semantic Web)* [7] es la web de los "metadatos". Se basa en la idea de añadir metadatos *(datos sobre los datos)* en las páginas web. Esa información adicional describe el contenido, el significado y la relación de los datos.

ILUSTRACION 1

Esos metadatos se deben facilitar con unos ciertos criterios establecidos para que se puedan leer desde cualquier sistema de proceso de información, ya sea otra página web, un buscador, o una aplicación, de terceros. El objetivo es mejorar la calidad y la organización de la información de Internet.

La Web Semántica [8] se basa en metalenguajes y estándares de representación. Los más conocidos son XML y RDF:

- **XML**: Aporta sintaxis superficial para los documentos estructurados, sin dotarles de significado.
- **RDF**: Modelo de datos para los recursos y las relaciones que se puedan establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.

La principal ventaja es que la usabilidad y aprovechamiento de la Web puede mejorar sensiblemente con la web semántica gracias a que los documentos están etiquetados con información semántica. Se pretende que esta información sea interpretada por sistemas automáticos con una capacidad comparable a la del lector humano.

Por lo tanto, la técnica de web semántica nos dará una extensión de la Web actual dotada de significado, teniendo un espacio donde la información tiene un significado bien definido.
Nos permite que la información pueda ser interpretada tanto por agentes humanos como por agentes computerizados, consiguiendo resultados más avanzados y más exactos. Esto se consigue realizando un marcado descriptivo de los documentos y de los datos que éstos contienen (el marcado), mediante información estructurada y legible automáticamente.
Y por último, podemos decir que la web semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y a las personas trabajar en cooperación. Está basada en la idea de

proporcionar en la Web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas localicen, integren, razonen y reutilicen la información presente en la Web.

1.3 Estructura Gestor Documental

La estructura está basada en la idea de [9] web 3.0, lo que significa que está relacionada a lo que se conoce como web semántica. Los usuarios y los equipos, en este marco, pueden interactuar con la red mediante un lenguaje natural, interpretado por el software. De esta manera, acceder a la información resulta más sencillo. Por lo tanto, todos los datos alojados en la web 3.0 deberían ser "entendidos" por las máquinas, que podrían procesarlos con rapidez.
Esta estructura nos permite utilizar lenguajes ontológicos, lo que nos proporciona eficacia en las búsquedas.

ILUSTRACION2

La infraestructura de tecnologías y lenguajes necesaria para la implementación de la Web Semántica se puede esquematizar en varias capas. Estas capas son:

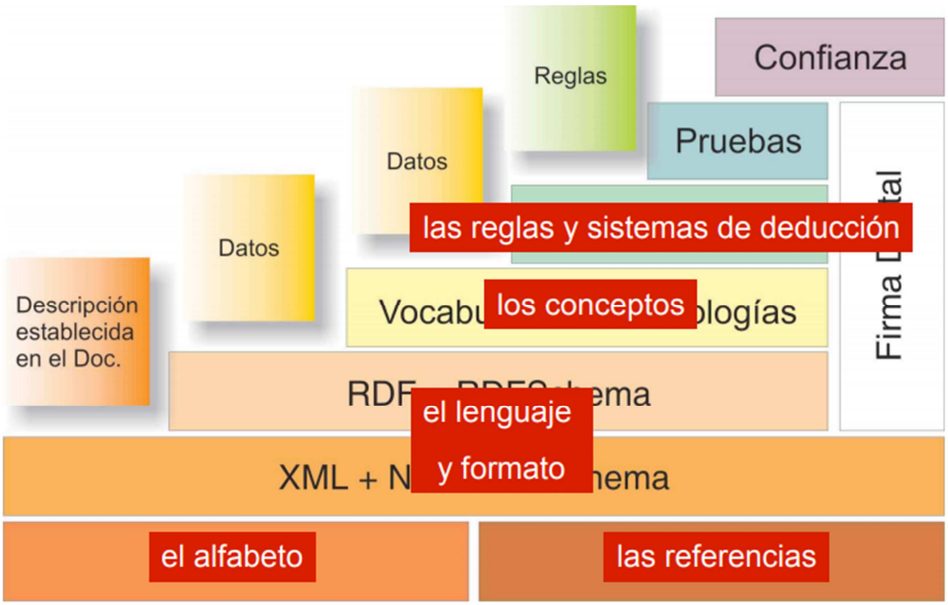
- Un modelo básico para establecer propiedades sobre los recursos, para el que se empleará RDF – Resource Description Framework.
- Un modelo para definir relaciones entre los recursos, a través de clases y objetos, expresado mediante RDF Schema.
- Una capa lógica que permita realizar consultas e inferir conocimiento, donde entrarían en juego las ontologías y los agentes software.
- Una capa de seguridad que permita asignar niveles de fiabilidad a determinados recursos.

Así pues, teniendo en cuenta las capas que definen el modelo de Web Semántica, pasamos a descomponer estas capas en una estructura de siete capas para su implementación. Cada capa necesita ser compatible con la previa.

Las capas son:

1. **Unicode-URI**: codificación básica e identificación de recursos.
2. **XML-NS-XML Schema (XML-NS-Esquema XML)**: sintaxis y su relación con los espacios del nombre, además de expresar la estructura de los XML.
3. **RDF+RDFSchema (RDF+Esquema RDF)**: RDF/RDFS para expresar primitivas en un modelo de datos que represente el conocimiento.
4. **Ontology Vocabulary (Vocabulario de Ontologías)**: Estructurando y clasificando conceptualmente mediante una ontología.
5. **Logic (Lógica)**: expresadas mediante axiomas y reglas monotónicas.
6. **Proof (Prueba)**: para validar las sentencias lógicas (aserciones).
7. **Trust (Confianza)**: Fiabilidad de los datos, normalmente mediante firma digital

En el siguiente diagrama [3] de la ilustración 3, se puede observar el conjunto de tecnologías que forman el esquema de capas mencionado.



ILUSTRACION 3

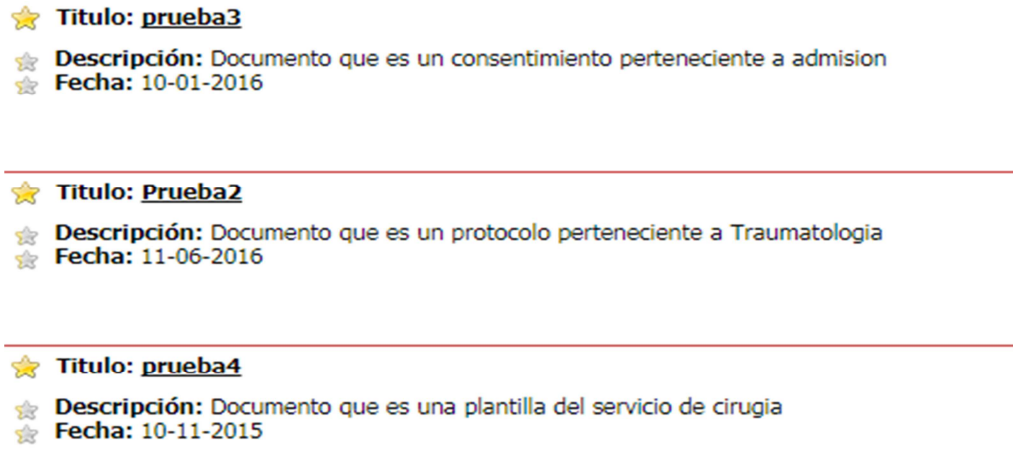
Una parte importante de estas capas será RDF (Resource Description Framework), que es el formato base de la web semántica, sobre el cual se basan los demás formatos o sobre el que se apoyan.

1.4 Composición del Gestor Documental

El gestor documental de documentos intrahospitalarios basados en la técnica de web semántica estará compuesta por un textbox para poder escribir las palabras que servirán de término para la búsqueda. En la ilustración 4 podemos ver la herramienta definida.

ILUSTRACION 4

Cuando se realiza las búsquedas se obtiene los resultados serán ofrecido en una tabla mostrando el nombre del documento, la descripción y la fecha del mismo. En la siguiente ilustración 5 se puede observar el contenido de una búsqueda.



ILUSTRACION 5

Una vez obtenido el resultado se puede acceder al documento por medio del enlace en el detalle del documento.

1.5 Funcionalidad

El hecho de que actualmente se crean ontologías para describir ciertos dominios del conocimiento, y que esto se haga empleando distintas tecnologías, fomenta precisamente el desarrollo y la evolución de estos métodos de representación del conocimiento. Y esto último por supuesto, desemboca en una mejora de la web semántica. La Web Semántica debe ser capaz de:

- Procesar contenido
- Razonarlo
- Hacer deducciones lógicas a partir de éste y de forma automática

Esto se puede conseguir realizando una estandarización de:

- El alfabeto
- Las referencias
- El lenguaje
- El formato
- Las anotaciones sobre significados
- Los conceptos generales
- Las reglas y sistemas de deducción

La apliación de estos estándares no permitirán que nuestro gestor documental cumpla los siguientes requisitos:

- Buena construcción de las consultas
- Evita ambigüedades por Polisemia
- Disminuye resultados presentados
- Soluciona problemas de interoperabilidad

2. Objetivos y alcance

Cuando hablamos de objetivos siempre viene motivado por la acción de poner en marcha un proyecto con la finalidad de ofrecer un servicio que siempre va ligado a la intención de mejorar, por el cual se encomienda. En nuestro caso será el poder tratar de organizar y ofrecer accesibilidad a los documentos intrahospitalarios mediante un gestor documental.

2.1 Objetivos

El desarrollo de una aplicación con la técnica de web semántica viene motivado por la inminente necesidad que tiene la organización de poder organizar el caos sobre documentos de uso diario, como son protocolos, guías, plantillas, consentimientos, etc… y sobre todo dar accesibilidad de forma ágil y rápida.

Atendiendo a las ventajas que tiene el buscador, la puesta en marcha proporcionará al usuario siempre un acceso a los documentos en cualquier punto de la organización, siendo un punto importante por la movilidad de los usuarios por toda la organización.

Se han realizado consultas a distintos servicios para comprobar como trabajan con los documentos, y prácticamente en su totalidad los documentos son alojados de forma local en los PCs, haciendo que siempre exista un discordancia entre el documento que utiliza el profesional de cara al paciente y el documento oficial que es realizado por la comisión y la dirección para su utilización.

También se ha detectado que el constante movimientos del profesional por distintas áreas de la organización hacen que los documentos siempre sean copiados o incluso llevado en unidades extraíbles con el consiguiente problema de desorganización.

Por lo tanto, tenemos un objetivo mayúsculo de desorganización el cual será solventado con el gestor documental.

2.2 Objetivos del proyecto

El objetivo principal de este Trabajo de Fin de Grado es desarrollar un gestor documental intrahospitalario con técnica web semántica para poder gestionar y dar accesibilidad a los documentos.

Otro de los principales objetivos es tener siempre los documentos en un repositorio donde el documento siempre se encuentra vivo, es decir, la comisión o la dirección si necesita actualizar, modificar o crear un documento, este estará accesible y organizado en su correspondiente destino listo para ser utilizado.

La agilidad y sencillez es otro de los objetivos, ya que los usuarios mediante el registro en la aplicación podrán acceder al buscador desde cualquier equipo dentro de la organización.

Y por último, destacar el objetivo de la puesta en marcha de un sistema sobre la técnica de web semántica en una organización que ha descuidado totalmente la organización tan costosa como es la gran cantidad de documentación.

2.3 Fases de realización

En la fase de la realización del proyecto se lleva a cabo todos las fases [10] necesarias para afrontar la puesta en marcha. Estas fases se dividen en:

- **Estudio:** Una vez se tienen los conocimientos sobre que se necesita, se realiza un estudio sobre la tecnología que se va a utilizar, ya que existen muchos factores que pueden cambiar el rumbo sobre el producto final.

- **Análisis:** Partiendo de un estudio previo es necesario definir con exactitud los objetivos para poder construir el producto final.

- **Diseño:** Una vez finalizadas las fases de estudio y análisis, comienza la fase de diseño que será la tarea programar y construir nuestra aplicación. En este apartado podemos ir creando lo que será el gestor documental, utilizando una capa visual, capa de datos, capa de gestión, base de datos y la seguridad.
- **Pruebas:** Llegados a esta fase se pone en marcha la aplicación poniendo a prueba las fases anteriores, donde se puede comprobar la efectividad del gestor.

- **Manual de usuario:** Con el fin de conseguir que la aplicación sea lo menos traumática posible para el usuario, sobre todo en su inicio, se documenta con detalle el funcionamiento del buscador

- **Soporte:** Por último, la herramienta dispondrá de soporte para cualquier incidencia que se pueda producir en el transcurso de su vida.

2.4 Estructura del documento

El documento describe todas las fases y procesos que el proyecto va a llevar a cabo para su puesta en marcha. Toda la información está detallada en los puntos anteriores así como en los siguientes, donde detallará toda la información acerca del análisis y el diseño.

A parte de la estructura y de cómo se va a realizar paso por paso, también se detallará que tipo de tecnologías y lenguajes son utilizados en el desarrollo, todo ello para aclarar aspectos básicos del proyecto

Hemos visto en el capítulo 1 como se realiza un estudio del estado del arte, que contiene una breve descripción del sistema junto a sus aspectos más relevantes, y una breve descripción de lo que es la técnica de la web semántica.

En los siguientes capítulos se describirá las fases de análisis, definición del proyecto, metodologías de desarrollo, descripción de características y listado de requisitos. Todo ello será ejecutado con una planificación previa para cumplir unos plazos o informar de la duración.

También veremos cómo se crea el diseño de la aplicación, diseño de interfaces, diseño de base de datos, lenguaje y seguridad.

Y por último se realizarán una serie de pruebas con su implementación llegando a realizar una serie de conclusiones además de la puesta en marcha de un soporte de usuario.

3. Metodología. Ontologías

3.1 Introducción

Los esfuerzos por normar la forma de hacer ontologías [11] y principalmente la necesidad de desarrollo de algunos proyectos, han derivado en la creación de diferentes metodologías y métodos para el diseño de ontologías.

Se trata de un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo del buscador. Existe una gran cantidad de métodos diferenciados por sus fortalezas y debilidades.

Para hacer ontologías debe considerarse que ya existen lenguajes específicos con ese fin. Estos lenguajes permiten el manejo de taxonomías complejas y de relaciones lógicas, además se estructuran mediante una semántica definida.

Dentro de estos lenguajes, entre los más utilizados, actualmente se encuentra OWL (Lacy, 2005). La programación con dichos lenguajes requiere de experiencia en el desarrollo de sistemas por lo que también puede hacerse uso de editores, que mediante interfaces gráficas ayudan a la construcción de la ontología.

Su función es tomar el conocimiento de cierto dominio y representarlo en un formato amigable para una máquina. El conocimiento va traduciéndose al lenguaje RDF. Desde el punto de vista técnico es un espacio de nombres de XML, es decir, si queremos escribir un documento con formato RDF, declaramos en el documento la versión de XML, y después introducimos como espacio de nombres el de la definición de RDF.
Por lo tanto, con RDF podemos representar conocimiento, pero a veces necesitamos extender ese conocimiento, para crear nosotros nuestros propios vocabularios, completamente personalizados para un determinado uso. No hay problema en ese caso, puesto que RDF sirve como base para dos lenguajes que extienden su utilidad, como son RDFS y OWL.

Y por último, OWL (Web Ontology Language), es el lenguaje que mayor significado semántico nos permitirá añadir, así como el que mayor libertad otorga a la hora de implementar una ontología.
Se construye sobre una base en RDF. Se escribe usando una sintaxis XML. Lo que hace OWL es proveer de una serie de axiomas que dotan de ciertas restricciones a unas clases que hemos definido y las relaciones permitidas entre estos elementos. Además OWL es capaz de usar lógica descriptiva.

Para conseguir una estimación y una planificación ajustada utilizaremos el ciclo de vida en cascada para desarrollar el buscador. La utilización del ciclo de vida en cascada nos proporcionará la capacidad de análisis para el desarrollo de la herramienta.
En el llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Las diferentes etapas por la que pasa un software construido con este tipo de metodología son:

- Especificación
- Conceptualización
- Adquisición del conocimiento
- Integración
- Implantación
- Evaluación
- Documentación

La Web Semántica, desde el punto de vista tecnológico, está ya lista para utilizarse, sin embargo hace falta el desarrollo de aplicaciones que exploten las bondades de esta tecnología, que permitirá contar con contenidos y sistemas educativos más eficientes en su desempeño tecnológico.
La creación de una ontología no es una actividad trivial, se requiere de un amplio entendimiento del tema, contar con capacidades analíticas y conocer el dominio que se describe.

A fin de crear una metodología el desarrollo evolutivo debe ser construido en poco tiempo y sin utilizar muchos recursos. El diseño rápido se centra en la representación de aspectos visibles para el usuario. Este rápido diseño conduce a la creación de un primer prototipo que será evaluado y retroalimentado por el cliente, gracias al cual el equipo de desarrollo podrá entender mejor lo que se debe hacer y permitiendo ver al cliente resultados progresivos a corto plazo.

3.2 Características de las ontologías

Una ontología [15] contiene definiciones que proveen del vocabulario para referirse a un dominio y éstas dependen del lenguaje que se usa para describirlas.

Por lo tanto, podemos describir sus características típicas:

- Pueden existir ontologías múltiples:

  El propósito de una ontología es hacer explícito algún punto de vista. Por ello, a veces, se necesita combinar dos o más ontologías. Cada una de ellas introduce conceptualizaciones específicas.

- Se puede identificar niveles de abstracción de las ontologías:

  Estos niveles de generalización proporcionan una topología de ontologías. La idea es caracterizar una red de ontologías usando multiplicidad y abstracción. Como no se puede aspirar a tener una descripción completa del mundo, se puede pensar en una estrategia de construcción gradual de abajo hacia arriba.

- Multiplicidad de la representación:

Un concepto puede ser representado de muchas formas por lo que pueden coexistir múltiples representaciones de un mismo concepto.

- Mapeo de ontologías:

  Permite crear relaciones entre los elementos de una o más ontologías para establecer conexiones, especializaciones, generalizaciones, etc.

3.3 Clasificación de las ontologías

Las ontologías se clasifican dependiendo de su dependencia y relación con una tarea específica desde un punto de vista:

- Ontologías de Alto Nivel o Genéricas:

  Describen conceptos más generales. En relación con los SI, estas ontologías describirían conceptos básicos. Por ejemplo: una teoría describiría partes y todos, y sus relaciones con la topología.

- Ontologías de Dominio:

  Describen un vocabulario relacionado con un dominio genérico. Por ejemplo, podría ser una descripción de datos y entidades relacionados con la sensorización remota con un ambiente urbano.

- Ontologías de Tareas o de Técnicas básicas:

  Describen una tarea, actividad o artefacto. Por ejemplo la evaluación de la contaminación sonora en ambientes urbanos o la descripción de características generales de componentes, procesos o funciones.

- Ontologías de Aplicación:

  Describen conceptos que dependen tanto de un dominio específico como de una tarea específica y, generalmente son una especialización de ambas.

Otra distinción y diferencia ontologías refinadas y no-refinadas, o lo que es lo mismo, off-line y on-line.

- No refinada:

  Tiene un número mínimo de axiomas y su objetivo es ser compartida por usuarios que concurren sobre una determinada visión del mundo.

  Tienen una mayor capacidad de ser compartidas y deberían ser utilizadas on-line para dar soporte en funcionalidad de sistemas de información.

- Refinada:

Precisa de un lenguaje de alta expresividad y tiene un gran número de axiomas. Deberían ser usadas off-line y solamente para referencia.

También se clasifican las ontologías de acuerdo a la cantidad y tipo de estructura de la conceptualización:

- Ontologías terminológicas:

  Especifican los términos que son usados para representar el conocimiento en el universo del discurso. Suelen ser usadas para unificar vocabulario en un campo determinado.

- Ontologías de información:

  Especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para el almacenamiento estandarizado de información.

- Ontologías de modelado de conocimiento:

  Especifican conceptualizaciones del conocimiento. Contienen una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen.

3.4 Uso de las ontologías

El uso de las ontologías puede aplicarse a tres ámbitos:

1. Comunicación:

   En modelos normativos, crea la semántica de un sistema y el modelo para extenderlo y transformarlo entre diferentes contextos.

   Establece una red de relaciones.

   Ofrece consistencia y falta de ambigüedad.

   Integra diferentes perspectivas de usuarios.

2. Interoperabilidad: usa ontologías como una inter-lingua.

   La interoperabilidad puede ser:

   - Interna: bajo el control de una unidad organizacional.
   - Externa: aislada del resto, integrada entre dominios e integrada entre herramientas.

3. Ingeniería de sistemas:

   Especificación, confiabilidad y reutilización.

3.5 Utilización de las ontologías

Actualmente, las ontologías facilitan aspectos como comunicación, interoperabilidad y razonamiento automático.

Comunicación

Las ontologías reducen las confusiones de tipo conceptual y terminológico; de esta forma, habilitan el entendimiento compartido y la comunicación entre personas, organizaciones y aplicaciones con diferentes necesidades y puntos de vista sobre un contexto particular. Esto se logra ya que a través de las ontologías se pueden crear modelos normativos los cuales permiten agregar semántica a un sistema determinado y crear un modelo extensible que puede ser refinado posteriormente y que permite transformaciones semánticas entre diferentes contextos .

Interoperabilidad

La interoperabilidad es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos. Para el tratamiento de la interoperabilidad, las ontologías se utilizan para soportar traducciones entre diferentes lenguajes y representaciones. En términos más prácticos, una ontología puede funcionar como traductora permitiendo la interoperabilidad entre dos aplicaciones de software.

Razonamiento Automático

Las ontologías resultan muy útiles para facilitar el razonamiento automático, es decir, sin necesidad de la intervención humana. Partiendo de unas reglas de inferencia, un motor de razonamiento puede usar los datos de las ontologías para inferir conclusiones de los mismos.

Una de las aplicaciones más importante del razonamiento automático es la validación de datos. También se usa para establecer relaciones entre ontologías, para descubrir relaciones ocultas o inesperadas entre los datos y para integrar esquemas de bases de datos.

Por otro lado, el uso de ontologías, aunado a sus bondades en cuanto a comunicación, interoperabilidad y razonamiento automático, permite que los usuarios organicen y estructuren la información de manera que los agentes de software puedan interpretar el significado y, por tanto, puedan buscar e integrar datos mucho mejor que como se hace actualmente. Gracias al conocimiento almacenado en las ontologías, las aplicaciones podrán extraer automáticamente datos de las páginas web, procesarlos y sacar conclusiones de ellos, así como tomar decisiones y negociar con otros agentes o personas.

3.6 Ventajas de las ontologías

Una de las ventajas de las ontologías es la posibilidad de ser utilizadas como una herramienta para obtener información, y de esta manera poder elaborar sistemas basados en el conocimiento, en este contexto las ontologías aportan autenticidad, consistencia y fiabilidad cuando se necesita recobrar la información.

Otra ventaja de las ontologías es la ayuda que proporcionan para compartir el conocimiento y la capacidad de realizar un trabajo colaborativo entre organizaciones por

medio de un conocimiento común. Por lo tanto, aplicar una lógica descriptiva que pueda generar un procesamiento semántico a través de herramientas y tecnologías.

Una de las principales ventajas visto desde una BB.DD es la simultaneidad para recuperar información buscada. Además la información puede ser de cualquier contenido, estructurada o no, documentación geográfica, urbanística, etc…

Otra ventaja significativa es la posibilidad de utilizar información persistente en dichas BB.DD a través de la propia ontología sin tener que renunciar a la BB.DD original, obteniendo una convivencia mutua. Las ontologías pueden ser reutilizadas para nuevos sistemas. De esta manera se consigue la interoperabilidad entre varios sistemas, en otras palabras el procedimiento para intercambiar información y usarla en cada sistema.

3.7 OWL (Web Ontology Language)

OWL es el acrónimo del inglés Web Ontology Language, un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML.
OWL es un lenguaje de Ontologías Web. Proporciona un lenguaje que utiliza la conexión por RDF para añadir las siguientes capacidades a las ontologías:

- Capacidad de ser distribuidas a través de varios sistemas
- Escalable a las necesidades de la Web Compatible con los estándares Web de accesibilidad e internacionalización
- Abierto y extensible

OWL extiende RDFS para permitir la expresión de relaciones complejas entre diferentes clases RDFS, y mayor precisión en las restricciones de clases y propiedades específicas.

OWL se trata de un lenguaje de marcado recomendado por W3C que está diseñado para representar el contenido de la web de manera que es interpretable por agentes inteligentes, aplicaciones web, apps, etc. Es decir, se utiliza OWL cuando se requiere que la información sea procesada por el software y no para representarla.

En la actualidad, OWL posee sub-lenguajes que aumentan la expresión del mismo, ya que incorporan diferentes funcionalidades: OWL Lite, OWL DL y OWL Full. Al igual que el lenguaje RDF, OWL emplea las tripletas de RDF pero el mecanismo de OWL es mejor, debido a que permite interpretar mejor los contenidos de la web, puesto que proporciona un mecanismo adicional además de una semántica formal. En conclusión, OWL es un lenguaje que tiene más poder expresivo que RDF y XML.

3.8 SPARQL (Lenguaje de consulta para RDF)

Como se ha descrito en apartados anteriores, RDF es un formato de datos para grafos dirigidos y etiquetados para representar la información en la Web. Esta especificación define la sintaxis y la semántica del lenguaje de consulta SPARQL para RDF.
SPARQL se puede utilizar para expresar consultas que permiten interrogar diversas fuentes de datos, si los datos se almacenan de forma nativa como RDF o son definidos mediante vistas RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con sus conjunciones y disyunciones.

SPARQL también soporta la ampliación o restricciones del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF.

3.9 Estructura de consultas SPARQL

La estructura que pueden tomar las consultas SPARQL se puede observar en la siguiente estructura.

```
#Directiva base
BASE <URI>
#Declaraciones de los prefijos
PREFIX pref: <URI>
#Definición del conjunto de datos
FROM . . .
#Resultado de la cláusula
SELECT . . .
#Patrón de consulta
WHERE {
OPTIONAL { . . . }
FILTER ( . . . )
}
#Modificadores de la consulta
ORDER BY . . .
```

La estructura empieza con dos partes que sirven para abreviar URIs, una directiva BASE y la definición de uno o varios prefijos. A continuación, se definen los conjuntos de datos que se va a realizar la consulta, a través de un grafo RDF.

La cláusula FROM también es opcional, por lo que si se omite se toma por defecto el conjunto de datos que se haya creado previamente a la consulta.

Seguidamente tenemos la cláusula SELECT se especifica que información (variables y valores) debe devolver como resultado la consulta.

La cláusula WHERE especifica el patrón de la consulta, es decir, establece que conjunto de datos se va a obtener a partir de las indicaciones de la cláusula. Esta cláusula no es opcional pero la palabra WHERE sí lo es.

La palabra clave OPTIONAL sirve para especificar los patrones de la consulta que pueden hacer que la misma falle si no se encuentran. Esta cláusula es opcional.

Cuando se quiere añadir restricciones para filtrar la solución se utiliza la palabra reservada opcional FILTER. Esta palabra reservada utiliza condiciones que devuelven verdadero o falso para filtrar los resultados de la consulta deseados. Se pueden utilizar los operadores lógicos && y ||.

Y por último se definen de manera opcional los modificadores de la consulta que proporcionan la manera de organizar los resultados de la misma. Esta cláusula es opcional. La cláusula ORDER BY ordena de manera ascendente o descendente los resultados de la consulta..

3.10 Formas de consultas

Cláusula SELECT

Como se mencionó anteriormente, SELECT devuelve un conjunto de valores en forma de tabla en concordancia con el patrón establecido de búsqueda.

Cláusula DESCRIBE

Cuando no se conoce el grafo RDF se utiliza la cláusula DESCRIBE para pedir al procesador de consultas SPARQL que describa el recurso

Cláusula CONSTRUCT

La cláusula CONSTRUCT devuelve un nuevo grafo RDF. Se trata de una alternativa a la cláusula SELECT. Es decir, en lugar de devolver una tabla de valores con los resultados obtenidos, devuelve un grafo RDF.

Cláusula ASK

La cláusula ASK devuelve verdadero o falso dependiendo de si se cumple el patrón de la consulta y encuentra alguna coincidencia en el conjunto de datos. Es decir, según la pregunta realizada devuelve la respuesta en forma de booleano.

```
PREFIX prop: <http://dbpedia.org/property/>
ASK{
<http://dbpedia.org/resource/Amazon_River> prop:length ?amazon .
<http://dbpedia.org/resource/Nile> prop:length ?nile .
}
```

4. Planificación

Para la planificación utilizaremos la herramienta Microsoft Project planificaremos la estimación del tiempo con las tareas a realizar desde el inicio. El desarrollo será en función de los 3 meses teniendo en cuenta el esfuerzo en cada etapa del desarrollo.

En caso de que la planificación sea con un desarrollo complicado se utilizará la herramienta COCOMO II para estimar el esfuerzo y poder distribuir así los recursos de forma óptima.

Consideramos que las actividades necesarias para llevar a cabo el proyecto son las siguientes:

Descomposición estructural de actividades		
Código de actividad	*Actividades nivel 1*	*Actividades nivel 2*
01	Inicio del proyecto	
02	Gestión del proyecto	

03	Construcción del software	
03.01		Análisis
03.02		Diseño
03.03		Programación y Pruebas unitarias
03.04		Pruebas
04	Formación de usuarios	
05	Puesta en producción	
06	Final del proyecto	

TABLA 1

La actividad 03.03 se agrupa mediante dos actividades y que más adelante se explicará su desarrollo para facilitar el desarrollo de las mismas.
Teniendo definidas las actividades que se van a realizar en el proyecto, vamos a desglosar el tiempo estimado para cada actividad en jornadas de trabajo.

La estimación máxima es de 3 meses (aprox) desde el inicio, por lo tanto, procedemos a desglosar las etapas de la siguiente manera:

Duración estimada de las actividades		
Código de actividad	*Actividad*	*Jornadas*
01	Inicio del proyecto	--
02	Gestión del proyecto	5
03	Construcción del software	--
03.01	Análisis	10
03.02	Diseño	15
03.03	Programación y Pruebas unitarias	35
03.04	Pruebas	9
04	Formación de usuarios	1
05	Puesta en producción	1
06	Final del proyecto	--
TOTAL		**76**

TABLA 2

Utilizaremos Microsoft Project para hacer la planificación temporal del proyecto teniendo en cuenta las precedencias entre las actividades establecidas.
Por lo tanto, los plazos vienen definidos en jornadas de trabajo las cuales forman el calendario de la puesta en marcha y el final del proyecto.

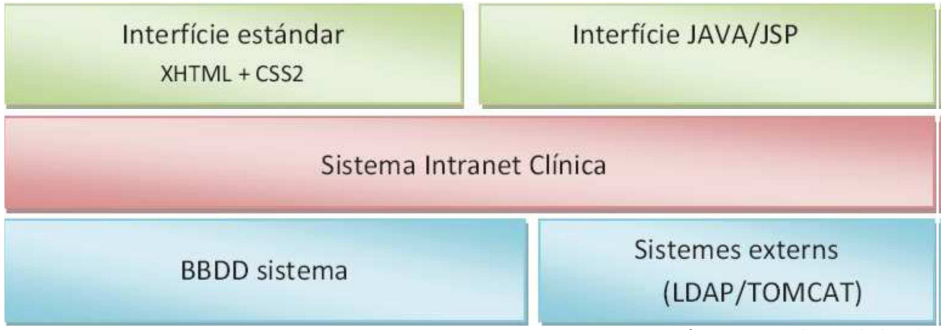
Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
Inicio proyecto	1 día	jue 04/01/18	jue 04/01/18		
Gestión proyecto	**6 días**	**vie 05/01/18**	**vie 12/01/18**		**gestión**
Administración de aplicación	3 días	vie 05/01/18	mar 09/01/18		gestión
Consulta necesidades	2 días	mié 10/01/18	jue 11/01/18	3	gestión
Seguimiento	1 día	vie 12/01/18	vie 12/01/18	4	gestión
Construcción del software	**68 días**	**lun 15/01/18**	**mié 18/04/18**		
Análisis	10 días	lun 15/01/18	vie 26/01/18		analista
Diseño	15 días	lun 29/01/18	vie 16/02/18	7	analista programador
Programación y pruebas	35 días	lun 19/02/18	vie 06/04/18	8	analista programador
Pruebas	8 días	lun 09/04/18	mié 18/04/1:	9	analista
Formación usuario	**2 días**	**jue 19/04/18**	**vie 20/04/18**		
Usuarios	2 días	jue 19/04/18	vie 20/04/18	6	analista
Puesta producción	1 día	lun 23/04/18	lun 23/04/18		técnico de sistemas
Fin proyecto	1 día	mar 24/04/18	mar 24/04/1:		

TABLA 3

4.1 Productos obtenidos.

Para tener una visión de conjunto clara respecto de qué hacer para desarrollar el sistema, a continuación presentamos algunos elementos que nos permitirán tomar contacto con la magnitud del proyecto. Para ello, utilizaremos herramientas propias de la ingeniería del software, y mostraremos cómo será la base de datos que utilizará el sistema, cuál será el diagrama de contexto y como habrá que configurar los perfiles de seguridad del nuevo sistema. La estructura de la aplicación será en base a la tecnología JAVA/JSP.

El esquema que define la simplicidad del sistema respecto al usuario final es:



Esquema sistema. ILUSTRACION 6

4.2 Descripción de capítulos

Los capítulos que se han realizado forman parte de todo el proceso que ha ocasionado el desarrollo de la aplicación, es decir, se ha detallado cualquier configuración o instalación que forma que se pueda seguir en cualquier punto en que se encuentre de forma técnica o presencial.
La descripción de los capítulos es la siguiente:

Capítulo1: La introducción forma parte en todos los proyectos para describir brevemente en que consiste.

Capítulo2: En todo proyecto es necesario identificar las necesidades y que solución o cómo podemos obtener en definitiva un desarrollo. Las necesidades dan lugar a obtener todo la estructura necesaria para poder desarrollar y obtener el producto final.

Capítulo3: La estructura del hardware será un paso importante en la aplicación. Si una aplicación no tiene una base correcta y bien configurada, los problemas de rendimiento pueden ser graves y ocasionar un abandono de la aplicación. Por lo tanto, en este capítulo definiremos una estructura de hardware y una configuración para que el rendimiento sea el correcto.

Capítulo4: Todo hardware da una base a un sistema operativo para llevar o desplegar un entorno de trabajo. En este capítulo describimos brevemente el S.O seleccionado para la aplicación.

Capítulo5: Las aplicaciones necesarias para poder desplegar la aplicación y que funcione sin problemas, vendrán definidas en este capítulo. Es importante tener claro desde un principio cuales son las necesidades básicas, y a raíz de obtener una necesidad seleccionamos que software puede ser el adecuado para su implantación.

Capítulo6: Todo hace referencia al anterior capítulo, ya que trata sobre la configuración del software seleccionado para el desarrollo y puesta en producción de la aplicación.

Capítulo7: En este capítulo pasamos a definir la estructura interna, es decir, ver la composición de las capas, configuración de archivos de sistema, rutas e información de las clases. Con esto conseguimos una descripción adecuada de cómo funciona internamente la aplicación a nivel de configuración.

Capitulo8: La aplicación lleva toda una serie de tablas en MySQL donde irá alojado el archivo OWL, ya que será desde MySQL el interrogatorio por medio de SPAQL para ofrecer la información de la ontología. Estas tablas serán las que alimenten a la aplicación para mostrar los datos de consulta.

Capitulo10: Muestra en entorno de trabajo para el desarrollo de la aplicación, en definitiva, las herramientas usadas para realizar el trabajo.

5. Definición de las necesidades

5.1 Estudio inicial

El gestor documental representa una herramienta de consulta, es decir, servirá para dar apoyo al facultativo, enfermera y personal no sanitario a la hora de poder obtener documentos de interés para el desarrollo de la actividad.

La necesidad de tener un gestor documental surge principalmente para que todo el personal del centro pueda tener a su disposición desde cualquier punto con soporte informático todos los documentos que precisen.

Es importante recalcar que los documentos son de vital importancia en cualquier lugar del centro, ya que pueden ser de carácter informativo como un consentimiento necesario para

cualquier operación, por lo tanto es imprescindible que el o los profesionales tengan disponibles estos documentos.

5.2 Estructura global

El desarrollo de la herramienta irá siempre enfocado a ser un gestor documental con un aspecto sencillo y amigable. La base del gestor es siempre un buscador el cual se moverá en función de las necesidades del usuario y de las ontologías creadas. Esta estructura será flexible ya que podrá ser alterada o modificada dependiendo de las necesidades del usuario según vaya siendo utilizada en el futuro, es decir, en una herramienta abierta a modificaciones o actualizaciones.

En la estructura inicial podremos encontrar 3 pantallas:

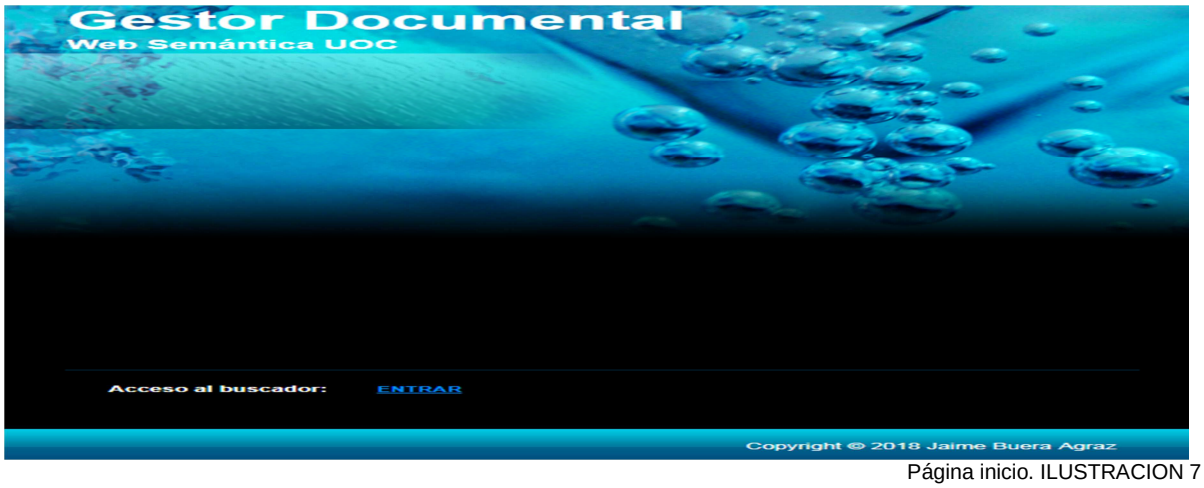
- La primera pantalla será la página de inicio la cual nos llevará al gestor documental.

- La segunda pantalla será donde aparecerá el buscador. En ella hay un textbox para introducir los elementos de búsqueda y un botón para lanzar la acción de búsqueda. En esta pantalla también aparecerán los datos buscados en una tabla con detalle.

- La tercera pantalla será la que muestre la información del documento seleccionado con información y el enlace al propio documento.

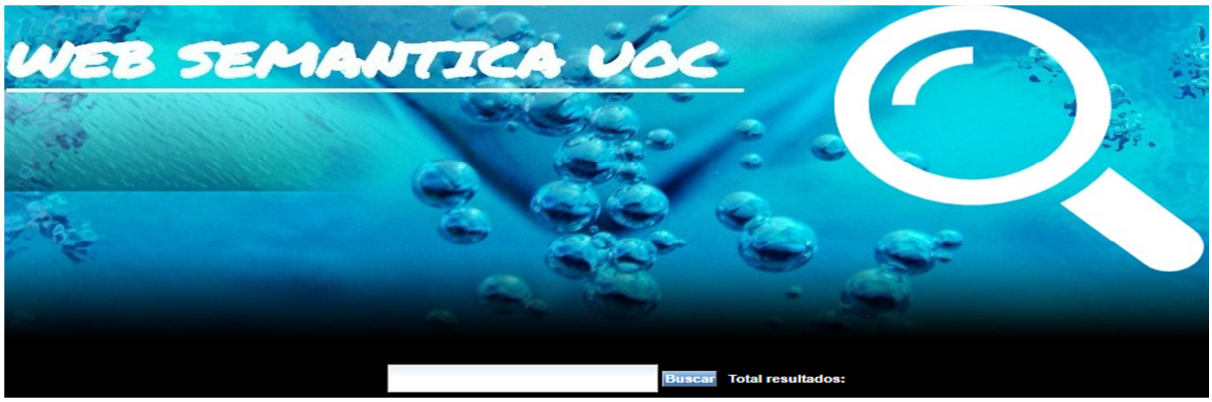
5.3 Funcionamiento aplicación

La aplicación se basa en 3 pantallas básicas. La página principal es la pantalla de inicio. En ella aparecerá un enlace que nos llevará al buscador.



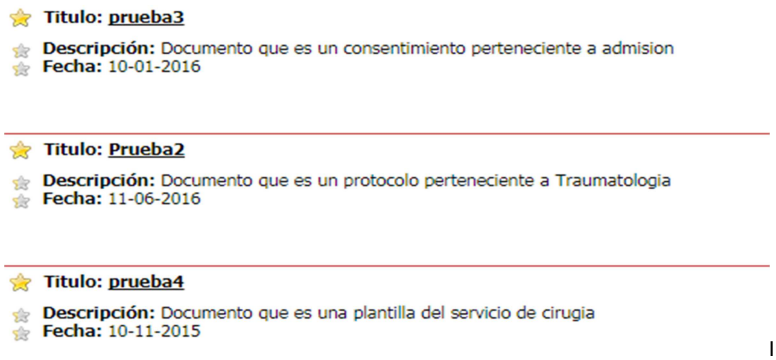
Página inicio. ILUSTRACION 7

Una vez llegados al buscador podremos realizar búsquedas por servicio, categoría y documento. Además se han añadido búsquedas por título, descripción del documento, url del documento y autor para enriquecer las búsquedas. En la siguiente ilustración se puede ver la pantalla donde podemos introducir palabras y realizar la búsqueda.



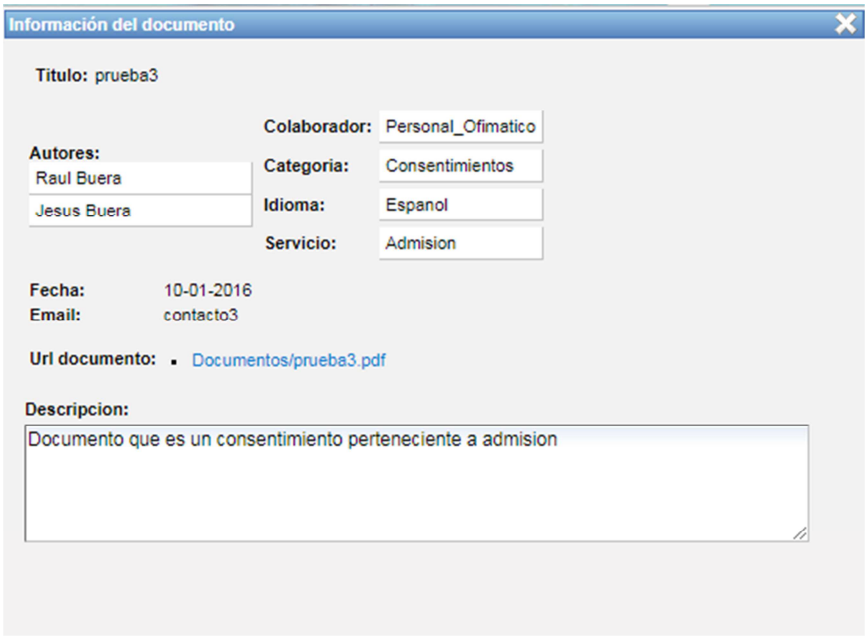
ILUSTRACION 8

Cuando introducimos un término a buscar se obtienen los resultados en una tabla mostrando el nombre del documento, la descripción y la fecha del mismo. En la siguiente ilustración 9 se puede observar el contenido de una búsqueda.



ILUSTRACION 9

Una vez obtenido el resultado se puede acceder al documento por medio del enlace que aparece en el título. Una vez que hemos accedido al documento podemos ver el detalle del mismo como se puede observar en la siguiente ilustración.



ILUSTRACION 10

5.4 Funcionamiento de la ontología sobre la aplicación

La ontología es la base sobre la cual está montado el buscador. El buscador base directamente su funcionamiento sobre las reglas, propiedades, tipo, relaciones, etc… que se generan sobre las ontologías para obtener la información que necesita el usuario.
Estas reglas han sido creadas bajo el lenguaje de OWL, con la herramienta de Protege.

Su funcionamiento es el siguiente, utilizamos las clases para tener definidos los aspectos básicos de la aplicación, siendo estas clases las siguientes:

- **Autor**: Clase que hace referencia al autor del documento.
- **Categoría**: Clase que describe o se obtiene la naturaleza del documento
- **Colaborador**: Persona o servicio que ha intervenido en el desarrollo del documento.
- **Documento**: Clase que muestra el documento como tal.
- **Hospital**: Clase sobre la ubicación del documento.
- **Idioma**: Clase que muestra su idioma
- **Servicios**: Clase referenciada al servicio del Hospital

Una vez definidas las clases se crean las propiedades de los objetos así como las de tipo de datos. Esta parte nos ayudará a poder asignarlas a las clases y obtener una referencia de que contiene cada una y como se puede relacionar.

Se crearan las instancias donde tendremos las clases con sus propiedades y se definirán los datos propios sobre los que buscar directamente en el buscador. Cada clase tendrá sus propiedades y sus instancias con los datos.

Todas las ontologías serán almacenadas a MySQL para que puedan ser leídas desde consultas SPARQL cuando el buscador tenga que realizar una búsqueda.
La búsqueda será interrogada a través de la siguiente consulta:

```
Query="?x table:titulo ?titulo. " +
        "OPTIONAL {?x table:autor ?autor. }"+
        "OPTIONAL {?x table:nombreServicio ?nombreServicio. }"+
        "OPTIONAL {?x table:categoria ?categoria. }"+
        "OPTIONAL {?x table:apellido ?apellido. }"+
        "OPTIONAL {?x table:url ?url. }"+
```

```
"OPTIONAL {?x table:fecha ?fecha. }"+
"OPTIONAL {?x table:descripcion ?descripcion. }"+
"FILTER (regex(?titulo, '^"+this.getTextoBuscar()+"') || "+
"regex(?url, '^"+this.getTextoBuscar()+"') || "+
"regex(?categoria, '^"+this.getTextoBuscar()+"') || "+
"regex(?descripcion, '^"+this.getTextoBuscar()+"') || "+
"regex(?apellido, '^"+this.getTextoBuscar()+"') || "+
"regex(?nombreServicio, '^"+this.getTextoBuscar()+"'))}";
```

Siempre y cuando la búsqueda tenga relación con el contenido almacenado y con el campo que se pueden realizar las búsquedas, se obtendrán los resultados para poder abrir el documento que se necesita.

Por lo tanto, construir un OWL con todo detalle sobre el contenido nos asegura una mayor precisión a la hora de buscar un contenido específico. Es por ello, que la principal importancia de un buscador semántico reside sobre cómo está estructurado y como está relacionado desde su origen el fichero OWL, es decir, las ontologías definidas.

6. Estructura del hardware

El servidor es una parte importante del proyecto, ya que en su estructura y componente se aloja el software que dará servicio a toda la estructura. Si bien inicialmente la carga de usuarios será baja, con la implementación y desarrollo de nuevas aplicaciones web esta demanda irá creciendo.

Por lo tanto, necesitamos una máquina la cual pueda soportar cargas puntuales bastantes elevadas. En el siguiente punto describimos un servidor tipo que se ajusta a las necesidades y que garantiza un servicio óptimo.

6.1 Componentes

El desglose del servidor se realiza teniendo en cuenta que el retardo en mostrar la información puede suponer para el usuario que abandone la herramienta y el proyecto sea un fracaso por culpa del hardware.

Principalmente, el software que más necesidad tiene de utilizar hardware óptimo es Tomcat 7, Java JDK 1.7 y MySQL 5. Partiendo de estas necesidades construimos el servidor con los siguientes componentes:

- **Procesador**: 2x AMD Opteron 6128, 8C, 2.0GHz, 8x512K L2/12M L3 Cache, 80W ACP, DDR3-1333MHz.

- **Placa Base**: PowerEdge R715 Rack Chassis for Up to 6x 2.5" HDDs

- **Memoria**: 8GB Memory for 2 CPUs, DDR3, 1333MHz (8x1GB Single Ranked RDIMMs)

- **Raid:** C4 8/12HD - R0 for PERC H700, 1-8 or 1-12 SAS/SATA/SSD HDDs based on chassis

- **Controladora RAID:** PERC H200 Integrated RAID Controller
- **Disco**: 3 discos de 600GB, SAS 6Gbps, 2.5-in, 10K RPM Hard Drive (Hot Plug)

- **Tarjeta red:** Intel® Gigabit ET Dual Port Server Adapter

- **Fuente alimentación:** Power Supply, Redundant (2 PSU), 750W Los demás componentes no son nombrados ya que carecen de valor en el rendimiento del servidor.

7. Sistema operativo

El sistema operativo influye sensiblemente a la hora de proporcionar un servicio de calidad. Es importante que el sistema operativo no se degrade con facilidad y que además su rendimiento sea bueno en comunión con el software que utilicemos.

El sistema operativo que se utilizará será Debian, ya que aparte de ser un software libre tiene una ventaja y es que al no instalar un entorno gráfico el rendimiento del S.O es bastante elevado y el consumo de recursos es infinitamente más bajo que cualquier S.O con entorno gráfico. A parte de todo esto podemos encontrar casi todo el software de aplicaciones para este sistema operativo en modo stable, por lo que da tranquilidad a la hora de efectuar cualquier instalación.

El único requisito imprescindible es tener actualizado el repositorio de librerías para poder actualizar en cualquier momento cualquier paquete necesario para el correcto funcionamiento de servidor.

8. Software del nucleo base

El software que utilizaremos para que nuestra aplicación funcione correctamente y el rendimiento sea óptimo, será el que describimos a continuación:

8.1 Tomcat 7

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es un servidor Web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el Servidor Apache Web Tomcat puede funcionar como servidor Web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de

transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Resumiendo, Tomcat será el encargado de servirnos todas las páginas dinámicas provenientes de los JSPs.

8.2 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Por un lado se ofrece bajo la GNU para cualquier uso compatible con esta licencia. Está desarrollado en su mayor parte en ANSI C Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. MySQL será el encargado de almacenar toda la información que necesitemos que sea almacenada para posteriormente ser mostrada desde la Web.

8.3 Apache 2.0

El servidor HTTP Apache es un servidor Web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Por lo tanto, Apache será bastante útil para utilizarlo como servidor http o HTTPS si queremos implementar el protocolo seguro bajo certificado y obtener así una seguridad casi absoluta con el cifrado del HTTPS.

8.4 Java JDK 1.7

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java.

Con el JDK obtenemos las herramientas necesarias para poder desarrollar servlets, JSPs, etc… y poder así mostrar el contenido de la información que extraeremos de las bases de datos.

9. Análisis

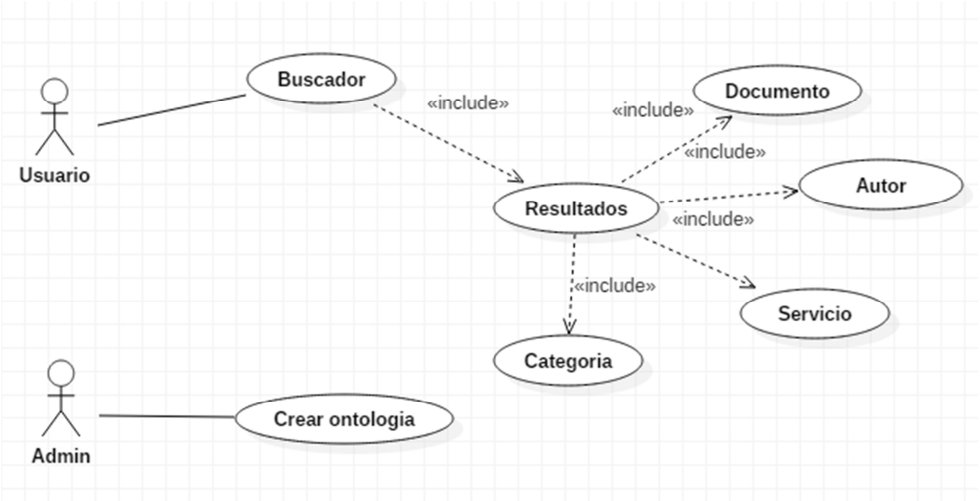
Vamos a analizar la aplicación desde el punto de vista del esquema de actores y sobre el diagrama de casos de uso. También aportaremos una breve descripción del esquema para detallar su funcionamiento.

9.1 Actores

El esquema principal de la aplicación viene dada por el usuario, ya que como es una herramienta abierta para cualquier persona del centro no vendrá determinado ningún tipo de rol para su acceso. Será un acceso público con libertad para utilizar cualquier documento que pueda ofrecer la herramienta.

9.2 Diagrama de casos de uso

El diagrama de casos de uso es el siguiente:



Casos de uso. ILUSTRACION 11

9.3 Descripción funcionamiento del diagrama de casos de uso

Admin

Será la persona encargada de administrar la aplicación. En concreto será el que cargue la ontología en el sistema. La ontología puede variar por lo que habrá una gestión de la misma.

Usuario

Será la persona o el cliente que utilice la aplicación

Buscador

Consiste en el formulario de búsqueda que realizará mediante las ontologías cargadas y ofrecerá una serie de resultados siempre dependiendo de que ontolgias se han utilizado.

Resultados

Cuando en el buscador se introduzcan los términos a buscar será aquí donde se mostrarán los resultados obtenidos sobre las ontologías cargadas. Las principales son las que se citan a continuación.

Documento

Es el propio documento al que hace referencia la búsqueda, siendo también una opción en el cribado inicial sobre la ontología.

Servicio

La estructura de la empresa ejerce una estructura sobre servicios los cuales son cargados ontológicamente para ofrecer un desglose más estructurado.

Autor

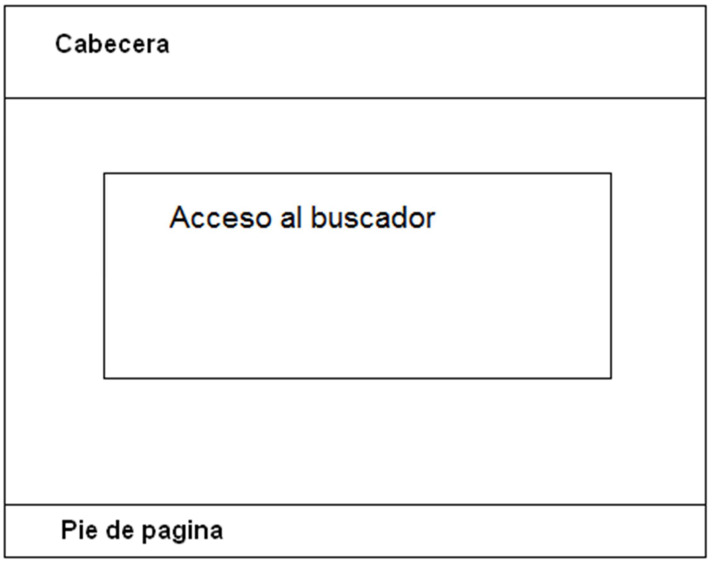
Es importante muchas veces que sabiendo quien es el autor podemos ofrecer información más detallado sobre el documento, sobre todo con personas muy cercanas sobre el entorno de trabajo

Categoría

Simplemente es una segregación sobre la naturaleza del documento.

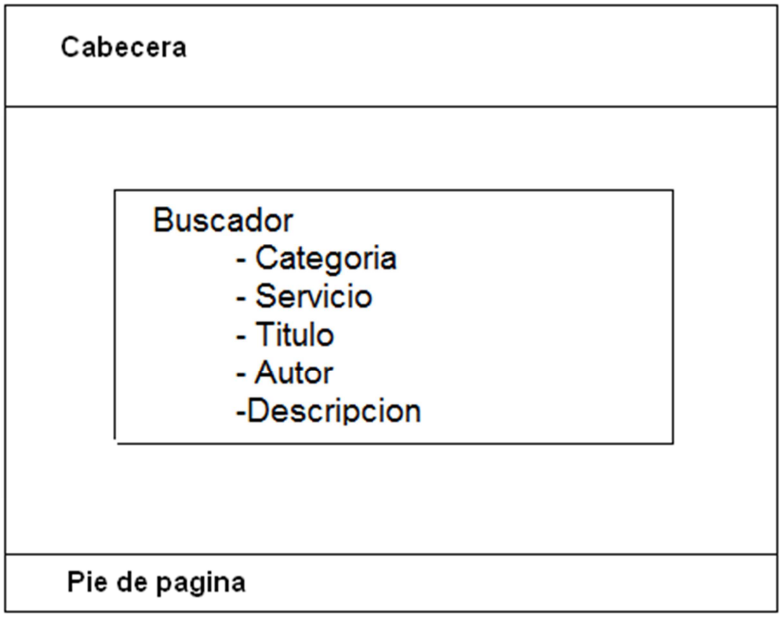
10. Entorno visual de la aplicación

La primera pantalla es la presentación y acceso a la aplicación:



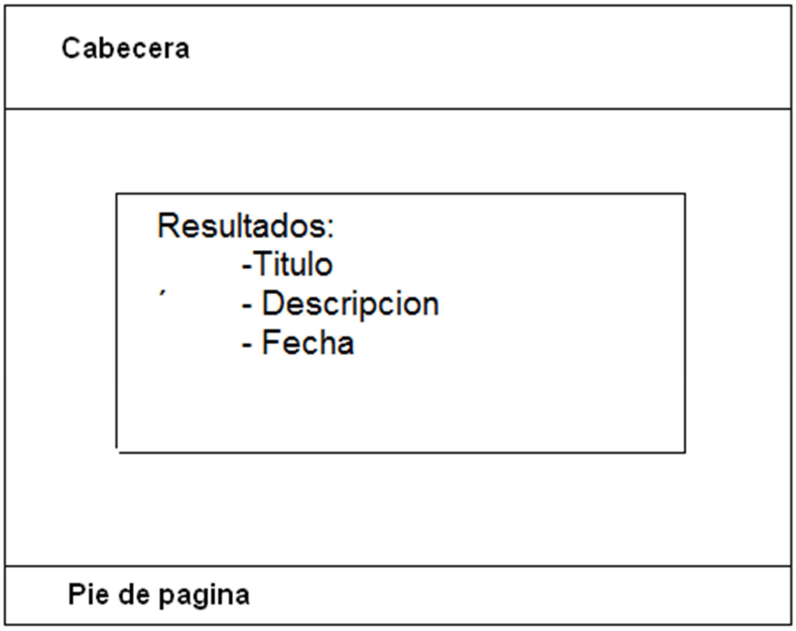
ILUSTRACION 12

Una vez accedemos al enlace de “ENTRAR” accedemos al buscador que nos permitirá realizar búsquedas sobre los documentos alojados.



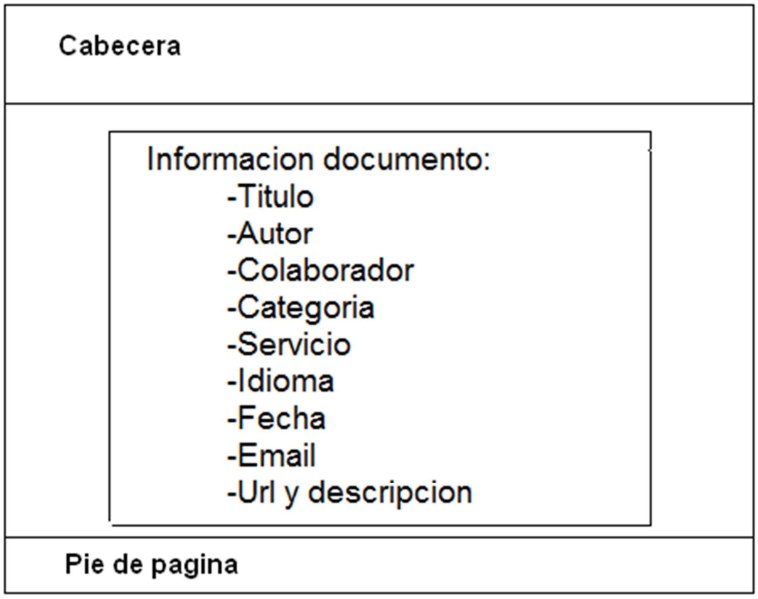
ILUSTRACION 13

En la búsqueda aparecerá la siguiente información:



ILUSTRACION 14

Una vez obtenido los resultados podemos acceder a la información del documento:



ILUSTRACION 15

11. Diseño

11.1 Arquitectura aplicación

La aplicación será creada con tecnología J2EE. Esta tecnología nos permite utilizar un modelo como el que detallamos a continuación:

Programación eficiente

Para conseguir productividad es importante que los equipos de desarrollo tengan una forma estándar de construir múltiples aplicaciones en diversas capas (cliente, servidor web, etc.).

En cada capa necesitaremos diversas herramientas, por ejemplo en la capa cliente tenemos applets, aplicaciones Java, etc…En la capa web tenemos servlets, páginas JSP, etc. Con JEE tenemos una tecnología estándar, un único modelo de aplicaciones, que incluye diversas herramientas; en contraposición al desarrollo tradicional con HTML, Javascript, CGI, servidor web, etc. que implicaba numerosos modelos para la creación de contenidos dinámicos, con los lógicos inconvenientes para la integración.

Extensibilidad

En un contexto de crecimiento de número de usuarios es precisa la gestión de recursos, como conexiones a bases de datos, transacciones o balanceo de carga. Además los equipos de desarrollo deben aplicar un estándar que les permita abstraerse de la implementación del servidor, con aplicaciones que puedan ejecutarse en múltiples servidores, desde un simple servidor hasta una arquitectura de alta disponibilidad y balanceo de carga entre diversas máquinas.

Integración

Los equipos de ingeniera precisan estándares que favorezcan la integración entre diversas capas de software. La plataforma JEE implica una forma de implementar y desplegar aplicaciones empresariales. La plataforma se ha abierto a numerosos fabricantes de software para conseguir satisfacer una amplia variedad de requisitos empresariales.

La arquitectura JEE implica un modelo de aplicaciones distribuidas en diversas capas o niveles. La capa cliente admite diversas tipos de clientes (HTML, Applet, aplicaciones Java, etc.). La capa intermedia contiene subcapas (el contenedor web y el contenedor EJB). La tercera capa dentro de esta visión sintética es la de aplicaciones como ERP, EIS, bases de datos, etc.

Como se puede ver un concepto clave de la arquitectura es el de contenedor, que dicho de forma genérica no es más que un entorno de ejecución estandarizado que ofrece unos servicios por medio de componentes. Los componentes externos al contenedor tienen una forma estándar de acceder a los servicios de dicho contenedor, con independencia del fabricante.
Utilizaremos un tipo de arquitectura de diseño MODELO – VISTA – CONTROLADOR también denominado MVC.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Pasamos a describir la arquitectura:

Modelo

Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador

Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.
Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo.

La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Para facilitar el desarrollo de la capa de Vista utilizaremos RICHFACES la cual detallamos a continuación.

11.2 RichFaces

RichFaces [12] es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.
Son características de RichFaces las siguientes:

- Se integra perfectamente en el ciclo de vida de JSF,
- Incluye funcionalidades Ajax, de modo que nunca vemos el JavaScript y tiene un contenedor Ajax propio,
- Contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas nuestras necesidades,
- Soporta facelets,
- Soporta css themes o skins,
- Es un proyecto open source, activo y con una comunidad también activa.

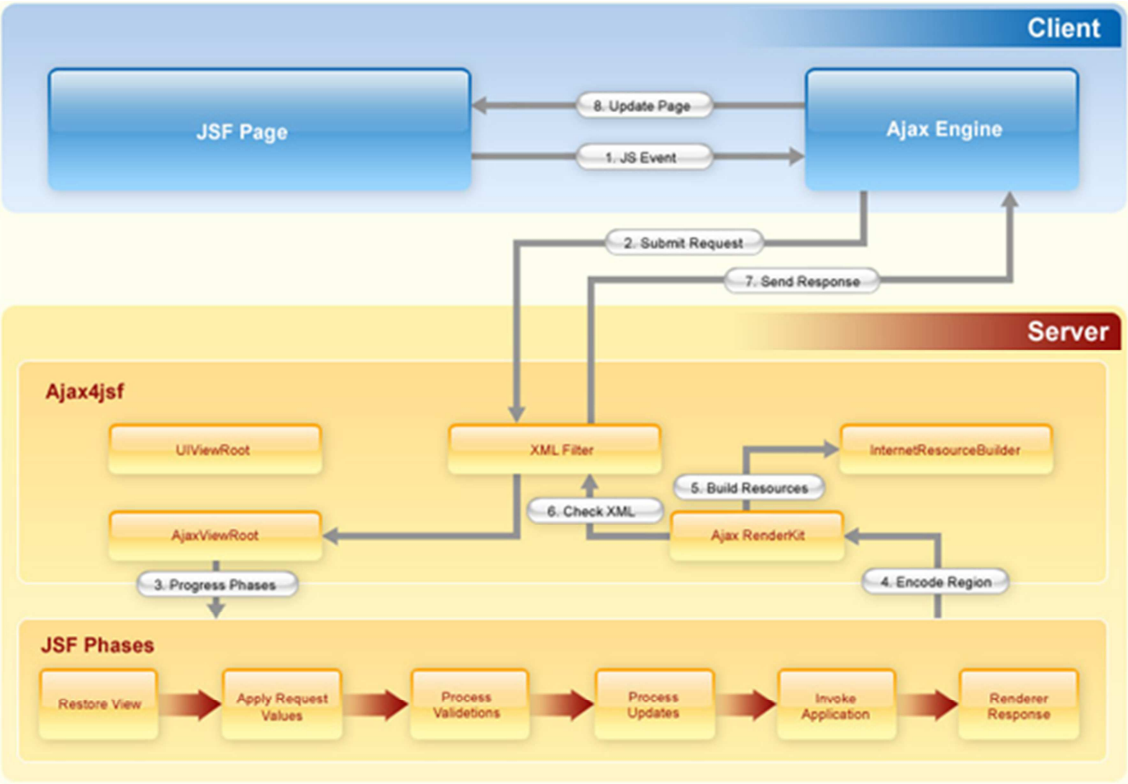
Aunque así enumeradas prodrían ser características aplicables también a ICEFaces.

Como inconvenientes, podríamos decir que:

- Usando Ajax4JSF tenemos que indicar qué parte de la pantalla tiene que repintarse. No es tan simple como ICEfaces, pero implica tener más control sobre los eventos que se producen en la interfaz de usuario.

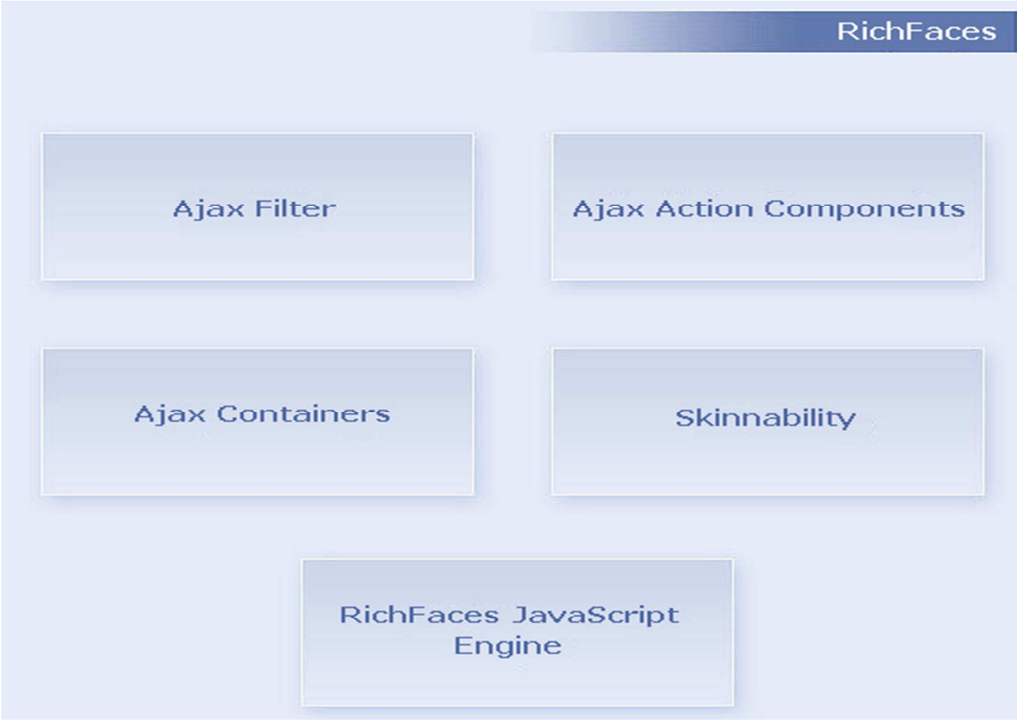
- En las últimas versiones siempre se les cuela alguna “peora”, que merma la funcionalidad de algún componente y donde, por ejemplo, funcionaba la subida de ficheros mediante un componente JSF con barra de progreso en Internet Explorer, ahora solo funciona en Firefox. Aunque también es cierto que se detecta y soluciona en la siguiente versión.

En la siguiente ilustración se puede observar cómo funciona dicho framework:



ILUSTRACION 16

Los principales elementos del framework son los que se muestran en la siguiente ilustración:



ILUSTRACION 17

11.3 Rutas desarrollo aplicación

La aplicación será desplegada en el webapps de Tomcat y se llamará Intranet. Será un war que se desplegará automáticamente para ser utilizado sin más. En el war irán las clases con las librerías, imágenes, scripts, estilos y plantillas necesarias para el correcto funcionamiento. Por lo tanto, partiendo del directorio de Tomcat la ruta principal será webapps/Buscador.

Partiendo de la ruta principal tendremos las siguientes rutas:

- Plantillas: Buscador/plantillas
- Imágenes: Buscador /imágenes, img
- Scripts, estilos: Buscador /estilos, stylesheet
- Librerías: Buscador /WEB-INF/lib
- Buscador resultados: Buscador /layout
- Documentos: Buscador /Documentos
- Página inicial: Buscador /index.jsp

11.4 Información de las clases

Las clases irán alojadas en el contenedor de la Intranet en:

- Buscador/WEB-INF/clases

La estructura será la siguiente:

- ***classes/accion***: Son las clases donde se implementan las búsquedas

- ***classes/modelo***: Son las clases que contienen la información y presentan los datos

- ***classes/utilidades*****:** Directorio donde están las clases con las utilidades sobre insertar ontologías, constantes, sql…

11.5 Acceso a datos

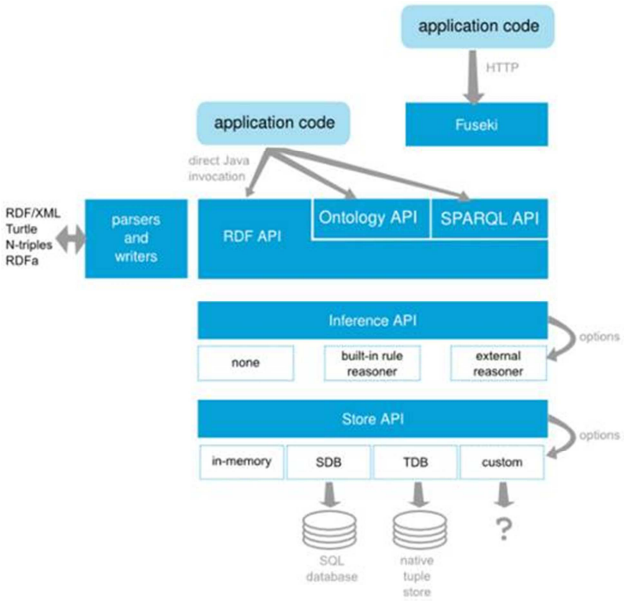
La base de datos será desarrollada bajo MySQL. Todas las tablas serán creadas bajo OntModel model del framework Jena [13] de Java para construir aplicaciones basadas en ontologías.

Jena se desarrolló en HP Labs en el 2000, en 2009 HP cedió el proyecto a la fundación Apache que decidió adoptarlo en noviembre de 2010.

Su Arquitectura incluye:

- API para trabajar (leer, procesar, escribir) ontologías RDF y OWL
- Motor de inferencia para razonar sobre ontologías RDF y OWL
- Estrategias de almacenamiento flexible para almacenar tripletas RDF en memoria o fichero
- Motor de queries compatible con especificación SPARQL

En la siguiente ilustración se puede observar su arquitectura:



ILUSTRACION 18

Para acceder a las ontologías insertas en MySQL mediante el framework de Jena se utilizará el lenguaje SPARQL. Este lenguaje es un formato de datos para grafos dirigidos y tiquetados para representar la información en la Web. Esta especificación define la sintaxis y la semántica del lenguaje de consulta SPARQL para RDF.

11.6 Construcción de la ontología en la aplicación

La construcción de ontologías no responde a una única aproximación lógica sino que depende, en gran manera, del contexto en el que se construyen. Independientemente del contexto, se han establecido un conjunto de principios que deben respetar las ontologías.

Los principales principios son:

- La representación de objetos del mundo real siempre depende del contexto en el que los objetos son usados. Este contexto puede verse como un punto de vista tomado del objeto. Es generalmente, imposible enumerar de antemano todos los posibles puntos de vista útiles de (o clases de) objetos.

- La reutilización de alguna parte de conocimiento requiere una descripción explícita de los puntos de vista que están inherentemente presentes en el conocimiento. De acuerdo a estos principios, una ontología especifica una conceptualización, una forma de ver el mundo. Por lo cual cada ontología incorpora un punto de vista.

Una ontología pues, contiene definiciones que se proveen del vocabulario para referirse a un dominio y éstas dependen del lenguaje que se usa para describirlas.

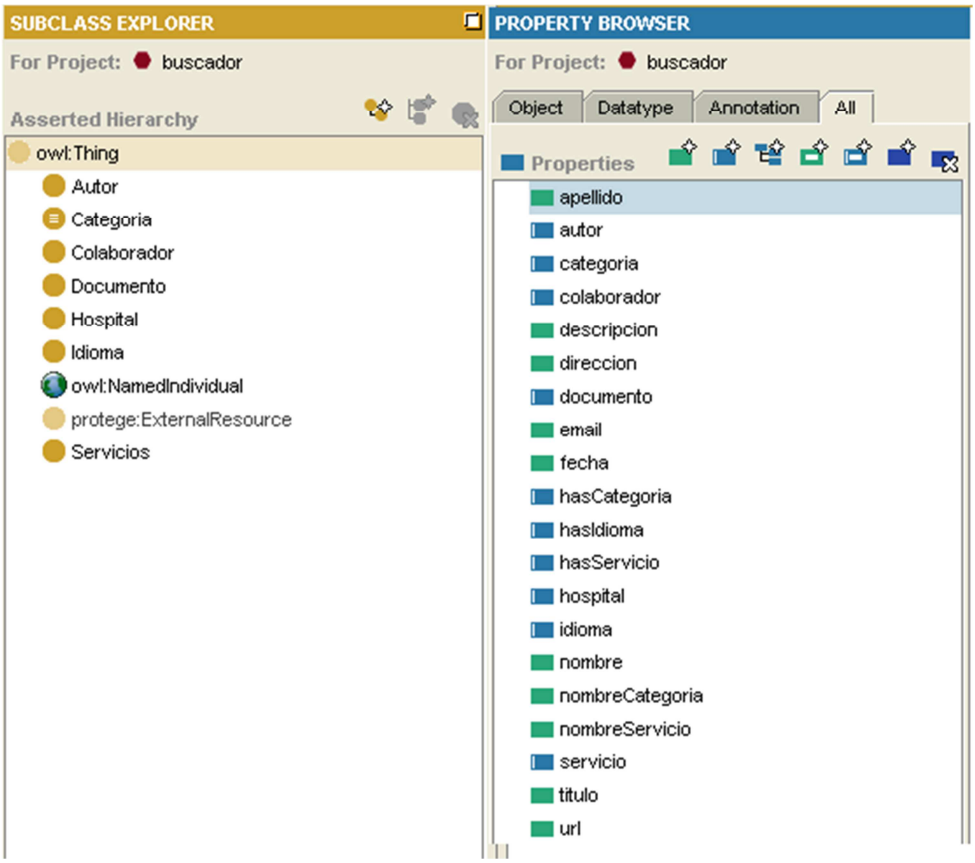
Una ontología se toman decisiones de diseño tales como:

- **Claridad**: una ontología debe de poder comunicar de manera efectiva el significado de sus términos. Las definiciones deben ser objetivas y comentadas en lenguaje natural.
- **Coherencia**: debe permitir hacer inferencias que sean consistentes con las definiciones.
- **Extendible**: debe anticipar usos y permitir extensiones y especializaciones monotónicas.
- **Sesgo de codificación mínimo (Minimal encoding bias)**: debe especificar al nivel de conocimiento sin depender de una codificación particular al nivel de símbolo.
- **Mínimo compromiso ontológico**: debe hacer la menor cantidad posible de "pretensiones" acerca del mundo modelado.

Por lo tanto, para la creación de la ontología OWL en la aplicación realizaremos mediante protege. Con esta herramienta crearemos de forma visual los individuos, propiedades y clases. Los individuos representan los objetos del dominio que creamos para nuestro proyecto, también son llamados como instancias. Las propiedades son relaciones binarias sobre los individuos anteriormente citados y por últimos tenemos las clases que se entienden por conjuntos que contienen individuos y pueden ser organizados dentro de otras clases y subclases.

En la siguiente imagen podemos ver los individuos, las propiedades y las clases:

Clases y propiedades:



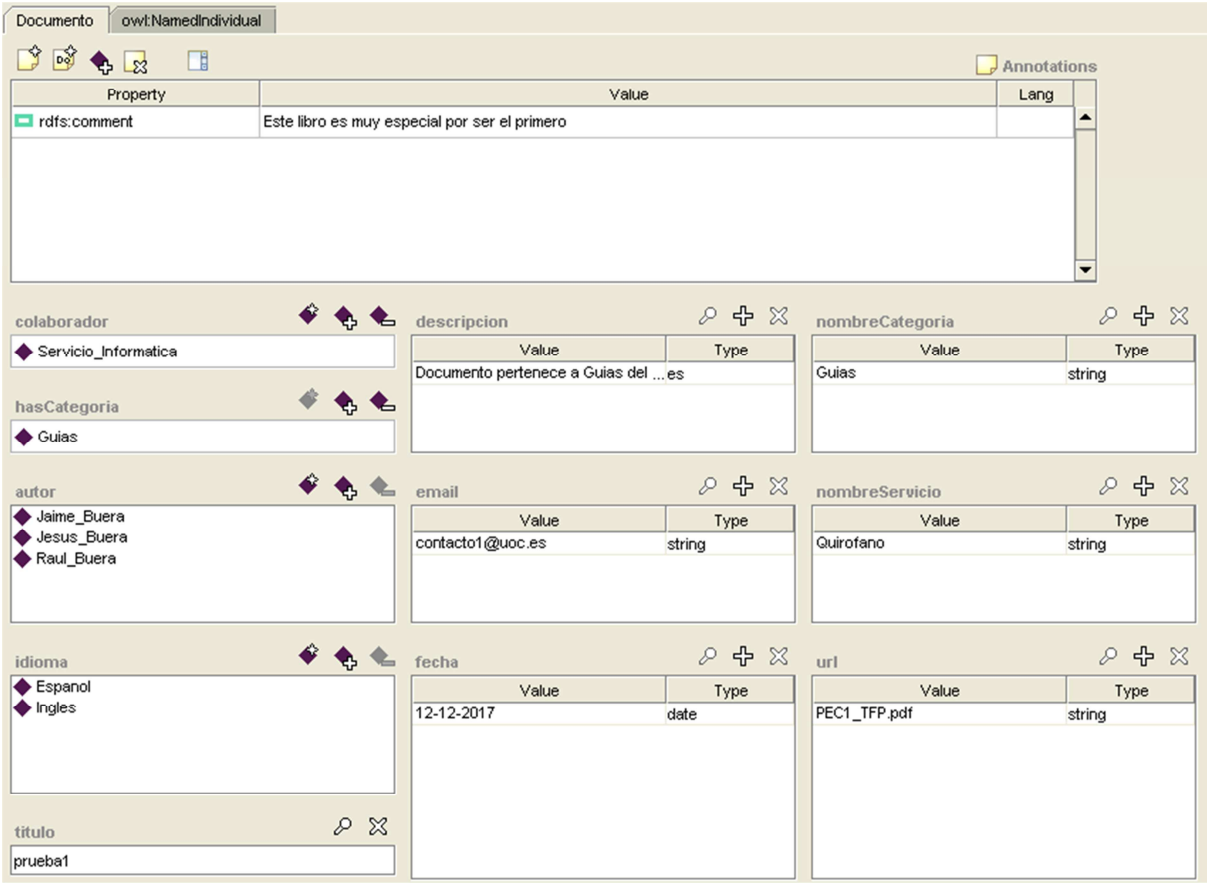
ILUSTRACION 19

Las clases principales con sus instancias son:

- **Categoria**
  - Consentimientos
  - Guías
  - PLantillas
  - Protocolos

- **Documento**
  - Prueba1
  - Prueba2
  - Prueba3
  - Prueba4
- **Servicios**
  - Admisión
  - Cirugía
  - Enfermería
  - Quirofano
  - Traumatología

La clase que contiene la información más detallada ya que corresponde al documento final es la clase Documento. Está clase contiene propiedades diversas que son las que utilizaremos como principales en las búsquedas. En la siguiente ilustración se muestra el formulario con sus propiedades sobre la instancia prueba1 de la clase documento:

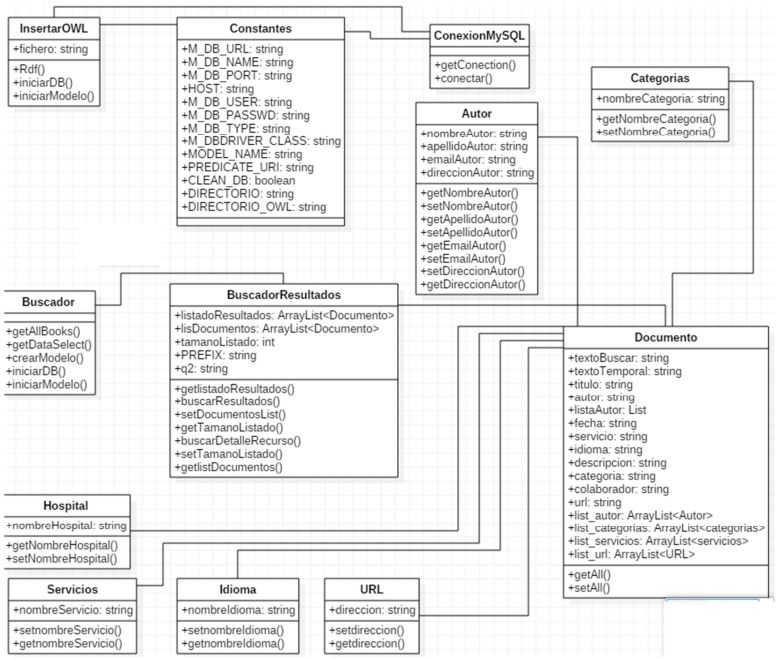


ILUSTRACION 20

Las clases restantes son informativas que simplemente muestran información más detallada sobre las clases principales.

12. Diagrama de clases

El diagrama de clases es el siguiente:



ILUSTRACION 21

13. Herramientas de programación

Las herramientas de trabajo para realizar el proyecto son:

- Entorno de desarrollo utilizaré Netbeans 7
- Navegador para la visualización Chrome
- Capa de presentación utilizaré JSP, HTML, CSS
- La comunicación con la capa de negocio será mediante Framework RichFaces.
- Como gestor de base de datos utilizaré MySQL.
- Para generar documentos utilizaré:
    - Microsoft Word 2003
    - Power point 2003
    - UML Draw
    - Project
    - Protege

Y por último, necesitaremos el API de Java para poder consultar las clases y packages necesarios para el desarrollo:

http://download.oracle.com/javase/1.5.0/docs/api/

14. Construcción aplicación

La aplicación tiene una serie de características que hacen que pueda ser operativa. El proceso comienza creando la ontología OWL mediante la aplicación de Portege. Una vez creada la ontología la cargamos en la BB.DD de MySQL mediante una clase capaz de interpretar el código OWL e insertarla en MySQL. Está clase es creada mediante el frame work de JENA.

Teniendo nuestra ontología en la BB.DD lo siguiente es construir la aplicación que será la encargada de ofrecer al usuario un entorno visual para realizar búsquedas. Para ello utilizamos el framework RichFaces, el cual utilizará el modelo MCV.

Con la aplicación ya creada, será la hora de poder interrogar a la BB.DD mediante el lenguaje de SPARQL, que será el que traduzca nuestras búsquedas sobre la BB.DD donde tenemos la ontología. Por lo tanto, desde la aplicación al realizar una búsqueda, obtendremos una consulta que será la que nos diga si tiene o no resultados.

En la siguiente ilustración podemos observar a modo resumen su funcionamiento.

ILUSTRACION 22

15. Puesta en marcha

La puesta en marcha del proyecto ha llegado a ser algo confusa. Desde un primer momento el enfoque inicial del proyecto era utilizar la potencia que tiene una BB.DD indexada como es SOLR, ya que todos los documentos son indexados y almacenadas siendo sus búsquedas muy rápidas y muy concisas ya que tiene referenciados cualquier texto en sus índices. A partir

de aquí, lo siguiente era utilizar además las ontologías, creando así un buscador rápido, eficaz y sobre todo aportar inteligencia a las búsquedas.

Se realizó en primer lugar la aplicación con el buscador utilizando Solr. Este buscador es muy eficiente y eficaz pero no consigue profundizar como lo hace un buscador semántico, por lo tanto, se dejó realizado pero apartado para desarrollar el buscador semántico.

El Buscador semántico fue algo confuso porque hubo una mal interpretación sobre cómo se conseguía la información que almacena el OWL haciendo pensar que esa información estaba fuera del propio OWL y que simplemente hacía de filtro. Por lo tanto, el desarrollo se hizo largo hasta llegar a entender su correcto funcionamiento.

Una vez terminado el buscador semántico, la idea inicial era el tener un buscador completo utilizando solr con sus indexados y las ontologías para poder ofrecer inteligencia a las búsquedas. El plazo sobre el proyecto ha sido algo justo y si a eso le sumamos que podríamos estar hablando de dos proyectos (el buscador con indexado y el buscador semántico) va a ser imposible el poder fusionarlos por falta de tiempo.

Por lo tanto, y a modo de conclusión, podemos observar como en muchas ocasiones es tan importante no salirse del guión y realizar lo que en un principio se pide como organizarse y presentar un producto el cual va más allá de lo expuesto construyendo un modelo más completo, otorgando así mayor satisfacción al usuario final.

16. Valoración económica

En este capítulo se hace una valoración básica haciendo referencia, a grandes rasgos, las operaciones, software y hardware más importantes del proyecto en general. Se detalla en la siguiente tabla la valoración económica:

Duración estimada de las actividades		
Código de actividad	*Actividad*	*Jornadas*
01	Inicio del proyecto	--
02	Gestión del proyecto	1500
03	Construcción del software	2100
03.01	Análisis	600
03.02	Diseño	2400
03.03	Programación y Pruebas unitarias	2500
03.04	Pruebas	200
04	Formación de usuarios	300
05	Puesta en producción	200
06	Final del proyecto	--
TOTAL		**10.100€**

TABLA 4

La valoración económica solo muestra el aspecto desde el inicio del proyecto hasta el final, sin tener en cuenta aspectos como el hardware o el software que se pudiera utilizar en el caso de mantener propiamente la herramienta.

17. Conclusiones

Con la realización de este proyecto se ha comprobado la eficacia de la herramienta a la hora de obtener información sobre documentos de forma centralizada.

Se ha sustituido el papel por la agilidad y ordenación de los datos para llevar a cabo un recurso o simplemente para aportar información al usuario final. Existen muchos casos en los cuales existe un desorden sobre la gestión de los documentos. Un paciente puede asistir a una consulta y necesitar un consentimiento, una enfermera utilizar un protocolo de actuación etc… y es difícil ordenar y agrupar cuando toda esa información está en papel. No solo eso, si no que puede causar un gran retraso en cualquier consulta o incluso en urgencias cuando hay que ordenar y leer toda esa información para que el especialista pueda saber con exactitud que cuadro clínico puede estar presentando el paciente.

Por eso, el conseguir mostrar en una sola pantalla, ordenado por servicios, categorías, etc… conseguimos una mayor rapidez a la hora de tomar un documento y una mejora a la hora de generar una actuación.

A medida que vas trabajando sobre el proyecto te vas dando cuenta sobre las mejoras que puedes realizar sobre la planificación inicial. A medida que desarrollas el trabajo puedes observar que la información se puede mostrar con el mismo contenido pero con un desarrollo distinto para ser más eficaz. Por eso mismo, después de haber recorrido todo el camino, la verdad es que podría cambiar bastantes aspectos del proyecto haciendo que la herramienta fuera más interactiva con el usuario.

La experiencia o conclusión que puedo sacar es que las tecnologías actuales hacen más fácil el trabajo y sobre todo, dependiendo del campo en que se trabaja, puede llevar a cabo una misión tan delicada como la de incluso salvar vidas en determinados casos. El seguimiento de un paciente realmente puede ser efectivo si el especialista reúne toda la información necesaria en cualquier momento bajo una ordenación y sencillez.

Para concluir, solo quiero hacer referencia a la planificación inicial. He visto y comprobado que una buena planificación puede ser la clave del éxito de cualquier proyecto ya que es la que realmente organiza todas las fases del proyecto haciendo que un proyecto llegue a buen puerto y sobre todo con el cumplimiento de las fechas que es un tema importante en el ámbito empresarial

18. Glosario

Buscadores: Son servidores Web que tienen acceso a una extensa base de datos sobre recursos disponibles en la propia Web. El usuario conecta con un buscador e indica unas pocas palabras representativas del tema sobre el que está buscando información y que se utilizan como clave de búsqueda. Como resultado de la búsqueda se muestra al usuario una lista con enlaces a páginas Web en cuya descripción o contenidos aparecen las palabras claves suministradas.

Autentificación: Acción de verificar la identidad de una persona o proceso.

Cabecera: Información acerca de un documento Web o un mensaje de correo que se encuentra al principio del documento o mensaje. La información que contiene una cabecera puede hacer referencia al autor, o el generador del texto.

Conexión: Circuito virtual de transporte que se establece entre dos programas de aplicación con fines comunicativos.

Framework: Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

CPU: Central Process Unit. Unidad Central de Proceso. El "cerebro" funcional del ordenador. Ejecuta instrucciones. Elemento que realiza la suma y sustracción real de los ceros y unos esencial para el cálculo.

Formulario: Fragmento de código HTML que permite la interactividad del usuario con una página Web mediante el sistema denominado CGI.

Navegador: Aplicado normalmente a programas usados para conectarse al servicio WWW.

Servidor: Cualquier ordenador de una red que proporciona servicios a otras estaciones de la red.

URL: Uniform Resource Locator. "Localizador Uniforme de Recursos". Denominación que no solo representa una dirección de Internet sino que apunta a un recurso concreto dentro de esa dirección.

S.O: Sistema operativo.

Base de datos: Sistema de almacenamiento de datos muy flexible que te permite utilizar la información en función de diversos criterios.

Hardware: (soporte físico): Nombre con el que se designa a los componentes físicos de los sistemas informáticos: unidad central del ordenador, periféricos, cables, conectores...

Java: Lenguaje de programación, desarrollado por Sun Microsystem, que también se utiliza para incluir pequeños programas (applets) en las páginas Web, que se activan al cargar la página. Permiten introducir más interacción en las páginas Web. El sistema ActiveX de Microsoft funciona de forma parecida.

Sistema operativo (S.O): Es el conjunto de programas que nos permiten comunicarnos con el ordenador y ordenarle la ejecución de determinadas tareas: ver lo que hay en un disco, copiar y transferir datos, ejecutar programas...

WWW *(World Wide Web)*: La telaraña mundial. Gran sistema de información en Internet formado por todas las páginas web alojadas en los servidores de la Red y relacionadas entre si mediante enlaces o hipervínculos. Cada página tiene una dirección a través de la cual se puede acceder a ella con un programa navegador (Netscape, Internet Explorer...) desde cualquier ordenador conectado a Internet.

Ontología: Parte de la metafísica que estudia el ser en general y sus propiedades

SPARQL: Es una tecnología clave en el desarrollo de la web semántica que se constituyó como recomendación oficial del W3C

OWL: Es el acrónimo del inglés **Web Ontology Language**, un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. **OWL** tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML.

RDF: Es una familia de especificaciones de la World Wide Web Consortium (W3C) originalmente diseñado como un modelo de datos para metadatos.

XML: Es un subconjunto de SGML(Estándar Generalised Mark-up Language),simplificado y adaptado a Internet

19. Bibliografía

En la bibliografía añado como parte de información relacionada a web semántica, RDF, SPARQL y OWL.

[1] https://www.w3c.es/Divulgacion/GuiasBreves/WebSemantica

[2]https://www.researchgate.net/publication/236658693_La_Web_Semantica_y_el_lenguaje_RDF

[3] https://es.slideshare.net/manzamb/xml-y-rdf-en-web-semntica-2113225

[4] http://skos.um.es/TR/rdf-sparql-query/

[5] https://e-archivo.uc3m.es/handle/10016/11423

[6]http://www.bcn.cl/obtienearchivo?id=documentos/10221.1/22771/1/LenguajeOntolog(OWL)W3C.pdf.

[7] https://boluda.com/tutorial/web-semantica/

[8]http://www.tlm.unavarra.es/research/seminars/slides/20080606_Juanra_Web_Semantica.pdf

[9]http://glossarium.bitrum.unileon.es/Home/web-semantica

[10]http://www.it.uc3m.es/jvillena/irc/practicas/08-09/05.pdf

[11]http://slideplayer.es/slide/1586066/

[12]https://docs.jboss.org/richfaces/latest_3_3_X/en/devguide/html/ArchitectureOverview.html

[13] https://unpocodejava.com/2012/07/27/que-es-apache-jena/

[14] http://skos.um.es/TR/rdf-sparql-query/

[15] http://personales.upv.es/ccarrasc/doc/2000-2001/Ontolog%C3%ADas/Index.htm

20. Anexos

No existen anexos al respecto