

Universidad de las Ciencias Informáticas

Facultad 6



**Indexa: buscador para la web del Centro Nacional
de Información de Ciencias Médicas.**

Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas

Autora: Danay Leyva Mayedo

Tutora: MsC. Frank Alain Castro Sierra

La Habana, Mayo de 2015

“Año 57de la Revolución”

“La disciplina es la parte más importante del éxito”.

Truman Capote

Yo, Danay Leyva Mayedo, declaro ser la única autora de este trabajo y autorizo al Centro Nacional de Información de Ciencias Médicas (Infomed) a hacer uso del mismo en su beneficio. Para que así conste firmo el presente a los _____ días del mes de _____ del año _____.

Autor

Danay Leyva Mayedo

Firma del autor

Tutor

Frank Alain Castro Sierra

Firma del tutor

Tutor: MsC. Frank Alain Castro

Categoría Docente: Asistente.

Síntesis: Graduado en la Universidad de las Ciencias Informáticas. Profesor de Álgebra Lineal y Matemática Discreta.

Correo electrónico: fcastro@uci.cu

Teléfono: 837 3140

Danay Leyva Mayedo

Dmayedo@estudiantes.uci.cu

Agradezco a mi familia toda, sin ellos, este sueño no hubiese sido posible. Los amo mucho.

Agradezco al departamento de desarrollo de Infomed. Gracias por haberme apoyado cuando tomé una decisión importante en mi vida.

Agradezco a mis mejores profesores, a todos los que me enseñaron todo el conocimiento que hoy poseo.

A todos los que me apoyaron durante la realización de esta tesis.

A mi madre, sin su apoyo no hubiera sido posible lograr este sueño. Gracias por hacer de mí la persona que soy hoy.

A mi esposo, sin él, cumplir este sueño no hubiera sido posible.

A mi niña Rachel, te quiero con todo mi corazón.

A mis tías, las quiero mucho.

A mi familia toda.

Resumen

En el Centro Nacional de Información de Ciencias Médicas el uso de las tecnologías de la información y las comunicaciones orientadas a la actividad productiva, permitió el surgimiento de una intranet y diversos sitios de difusión de información, por este motivo existe un gran volumen de información que debe ser localizada y organizada. Actualmente existe un metabuscador en funcionamiento que se encarga de gestionar las búsquedas. Como toda aplicación de su tipo no cuenta con una base de datos propia, sino que utiliza la de los buscadores de internet, por lo que no le da prioridad a la información que genera el CNICM y no tiene el control de la información que se muestra a los usuarios.

Producto a lo expuesto anteriormente surge la idea de crear un sistema de búsqueda y recuperación de información, específicamente un motor de búsqueda, que permita llevar a cabo la búsqueda y recuperación de la información existente en la web de la red telemática del centro. Para ello se realizó un estudio del estado del arte referente a los principales motores de búsqueda existentes en internet. Siguiendo la política del departamento de desarrollo del centro se decidió utilizar las siguientes herramientas, tecnologías y metodologías para la implementación del motor de búsqueda, *Apache Solr* como plataforma de búsqueda, *php* como lenguaje de programación, *MySQL* como gestor de base de datos, *Cakephp* como *framework* de desarrollo y para guiar el proceso de desarrollo del software se escogió la metodología para el desarrollo ágil Programación Extrema (en lo adelante XP).

Se realizó el diseño, la implementación y las pruebas de la solución propuesta obteniéndose un producto de software con los resultados esperados. Con la realización de este trabajo se obtuvo un motor de búsqueda para cumplir las necesidades actuales de la búsqueda de información en la red telemática del centro.

Abstract

Today at the National Information Center of Medical Sciences the use of information technology and communications-oriented productive activity, has allowed the emergence of an intranet and various information dissemination sites for this reason there is great volume of information that must be located and organized.

Product to the foregoing arises the idea of creating a system of information retrieval, specifically a search engine that allows to conduct search and retrieval of existing information on the website of the telematic network Infomed.

To do a study of the state of the art concerning the main drivers of existing Internet search was conducted, all with free software licenses, mostly GPL. It was further investigated, on the main tools, technologies and methodologies that best suit the solution, allowing choose Apache Solr as search engine, php programming language, MySQL as manager database, Cakephp as development framework to guide the process and methodology for agile development XP was chosen.

Design, implementation and testing of the proposed solution to obtain a software product quality expected results and thus complying with the requirements set by the client initially performed. The completion of this work has made it possible to obtain a product to help develop systems more efficient searches that would meet the current needs of Infomed.

Introducción	1
Capítulo #1: Fundamentación teórica.....	5
Introducción	5
1.1 Los sistemas de búsqueda y recuperación de información.....	5
1.2 Clasificación de los Sistemas de Recuperación de Información	5
1.2.1 Directorios	6
1.2.2 Metabuscadores	6
1.2.3 Motores de búsquedas	6
1.2.3.1 Arquitectura general de un motor de búsqueda.....	7
1.2.3.2 Los componentes de los motores de búsqueda	8
1.2.3.3 Motores de búsqueda verticales.....	9
1.3 La Indexación.....	9
1.4 Tecnología y herramientas	12
1.4.1 Apache Solr	12
1.4.2 Nutch	13
1.4.3 Cakephp	14
1.4.4 PHP	15
1.4.5 XAMPP	16
1.4.6 Metodología de desarrollo de software	18
1.4.6.1 Comparación entre metodologías ágiles y tradicionales.....	18
1.4.6.2 Metodologías ágiles	18
Fundamentación de la elección:.....	21
1.5 Sistema de gestión de base de datos.....	22
1.5.1 MySQL	22
Conclusiones parciales	22
Capítulo# 2: Planeación y diseño del sistema	24
Introducción	24
2.1 Propuesta del motor de búsqueda.....	24
2.2 Planeación	26
2.2.1 Personal relacionado con el motor de búsqueda	26
2.2.2 Historias de Usuario.....	27

2.2.3	Estimación de esfuerzos por HU.....	31
2.2.4	Plan de iteraciones	31
2.2.5	Plan de duración de iteraciones.....	32
2.2.6	Plan de entregas.....	32
2.3	Lista de Reserva del Producto.....	33
2.4	Diseño del sistema	34
2.4.1	Modelo de dominio	34
2.4.2	Tarjetas CRC.....	36
2.4.3	Patrones de arquitectura.....	37
2.4.3.1	Patrón Modelo Vista Controlador.....	37
2.4.4	Patrones de diseño.....	38
2.4.4.1	GRASP	39
2.4.5	Tratamiento de errores	41
2.4.6	Interfaz	41
2.5	Diseño de la base de datos	44
	Conclusiones parciales	44
	Capítulo # 3 Codificación y prueba	45
	Introducción	45
3.1	Codificación	45
3.2	Prueba	48
3.2.1	Pruebas Unitarias	48
3.2.2	Pruebas de aceptación	49
3.2.2.1	Iteración 1	49
3.2.2.2	Iteración 2	51
3.2.2.3	Iteración 3	53
	Conclusiones parciales	55
	Conclusiones	56
	Recomendaciones	57
	Referencias bibliográficas	58
	Glosario de términos.....	61
	Anexos.....	63

Fig. 1 Arquitectura de un motor de búsqueda	8
Fig. 2 Elementos de Apache Solr	13
Fig. 3 Fases de desarrollo de XP.	21
Fig. 4 Principales elementos del motor de búsquedas. Elaboración propia.	24
Fig. 5 Componentes del motor de búsqueda. Elaboración propia.	25
Fig. 6 Tipo de servicio web. Elaboración propia.	26
Fig. 7 Modelo de dominio. Elaboración propia	35
Fig. 8 Ciclo de vida de Modelo-Vista-Controlador. Elaboración propia.	38
Fig. 9 Clase User. Elaboración propia.	39
Fig. 10 Relación entre las clases User y AppControler. Elaboración propia.	40
Fig. 11 Clase AppControler. Elaboración propia.	40
Fig. 12 Interfaz de Indexa. Elaboración propia.	42
Fig. 13 Interfaz de Indexa. Elaboración propia.	42
Fig. 14 Modelo entidad -relación. Elaboración propia.	44
Fig. 15 Gráfico del resultado de las pruebas de aceptación. Elaboración propia.	54
Fig. 16 Autenticar usuario.	63
Fig. 17 Búsqueda de información.	63
Fig. 18 Añadir FIS.	64
Fig. 19 Indexar FIS.	64
Fig. 20 Autenticar usuario.	65

Tabla 1. Personal relacionado con el motor de búsqueda.....	27
Tabla 2: HU. Añadir FIS.....	27
Tabla 3: HU. Recuperar FIS.....	28
Tabla 4: HU. Actualizar FIS.....	28
Tabla 5: HU. Eliminar FIS.....	28
Tabla 6: HU. Indexar FIS.....	29
Tabla 7: HU. Búsqueda de información.....	29
Tabla 8: HU. Añadir perfil de usuario.....	29
Tabla 9: HU. Actualizar perfil de usuario.....	29
Tabla 10: HU. Eliminar perfil de usuario.....	30
Tabla 11: HU. Listar perfiles de usuarios.....	30
Tabla 12: HU. Autenticar usuario.....	30
Tabla 13. Estimación del esfuerzo por HU.....	31
Tabla 14. Plan de duración de iteraciones.....	32
Tabla 15. Plan de entregas.....	33
Tabla 16. Lista de Reserva del Producto.....	33
Tabla 17. Descripción de la tarjeta CRC user.....	36
Tabla 18. Descripción de la tarjeta CRC AppModel.....	36
Tabla 19. Descripción de la tarjeta CRC log.....	36
Tabla 20. Descripción de la tarjeta CRC sources.....	36
Tabla 21. Descripción de la tarjeta CRC search.....	36
Tabla 22. Descripción de la tarjeta CRC services.....	37
Tabla 23. Descripción de errores.....	41
Tabla 24. Tarea #1.....	45
Tabla 25. Tarea #2.....	45
Tabla 26. Tarea #3.....	46

Tabla 27. Tarea #4.....	46
Tabla 28. Tarea #5.....	46
Tabla 29. Tarea #6.....	46
Tabla 30. Tarea #7.....	46
Tabla 31. Tarea #8.....	47
Tabla 32. Tarea #9.....	47
Tabla 33. Tarea #10.....	47
Tabla 34. Tarea #11.....	47
Tabla 35: Descripción de la Prueba de Aceptación 1 HU Añadir FIS	49
Tabla 36: Descripción de la Prueba de Aceptación 2 HU Añadir FIS	49
Tabla 37: Descripción de la Prueba de Aceptación 3 HU Recuperar FIS	50
Tabla 38: Descripción de la Prueba de Aceptación 4 HU Actualizar FIS	50
Tabla 39: Descripción de la Prueba de Aceptación 5 HU Actualizar FIS	50
Tabla 40: Descripción de la Prueba de Aceptación 6 HU Eliminar FIS.....	50
Tabla 41: Descripción de la Prueba de Aceptación 7 HU Eliminar FIS.....	51
Tabla 42. Descripción de la Prueba de Aceptación 8 HU Indexar FIS.....	51
Tabla 43: Descripción de la Prueba de Aceptación 9 HU Búsqueda de información	51
Tabla 44: Descripción de la Prueba de Aceptación 10 HU Crear perfil de usuario	52
Tabla 45: Descripción de la Prueba de Aceptación 11 HU Crear perfil de usuario	52
Tabla 46: Descripción de la Prueba de Aceptación 12 HU Actualizar perfil de usuario	52
Tabla 47: Descripción de la Prueba de Aceptación 13 HU Eliminar perfil de usuario	52
Tabla 48: Descripción de la Prueba de Aceptación 14 HU Eliminar perfil de usuario	53
Tabla 49: Descripción de la Prueba de Aceptación 15 HU Autenticar usuario.....	53
Tabla 50: Descripción de la Prueba de Aceptación 16 HU Autenticar usuario.....	53

Introducción

En el actual desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC's) internet constituye una revolución sin precedentes, que ofrece infinitas oportunidades de comunicación e intercambio entre las personas. La información en internet presenta un índice de crecimiento muy elevado, en muchos casos se encuentra dispersa y poco estructurada lo que hace engorroso el proceso de encontrar información útil en la red; es por esta razón que surgen los sistemas de recuperación de información.

Cuba en su afán de informatizar la sociedad ha destinado cuantiosos recursos para lograr este objetivo. Uno de los sectores en los que se aprecia el resultado es en la salud, ya que cuenta con una red telemática, que se encarga de informatizar al Centro Nacional de Información de Ciencias Médicas (en lo adelante CNICM), el cual dentro de sus objetivos tiene mejorar la búsqueda de información en todos sus sitios web.

Actualmente existe en cada uno de los sitios web del CNICM un metabuscador en funcionamiento que soporta las búsquedas en su totalidad. Como toda aplicación de su tipo, el metabuscador, no cuenta con una base de datos propia sino que emplea otros buscadores como Google y Yahoo para encontrar la información solicitada por el usuario. Es común que estos buscadores publiquen servicios web para que terceros empleen su motor de búsqueda de forma remota, pero esto es diferente para Cuba. Como muchos servicios de internet, numerosos servicios webs de búsqueda de grandes compañías, están prohibidos para Cuba amparados bajo las leyes gubernamentales de los Estados Unidos de América que integran el bloqueo económico, comercial y financiero impuesto a la isla.

Las búsquedas que realizan los buscadores de los sitios del CNICM no le dan prioridad a la información que se genera en el mismo, por otra parte no tienen el control de la información que se muestra a los usuarios. También utilizan la Interfaz para el Acceso de Información en Salud, una herramienta desarrollada por el Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud. Esta tiene como inconveniente que los períodos de lanzamiento de sus versiones son muy prolongados y no responden al ritmo y a las exigencias de trabajo que necesita el CNICM y por lo tanto se hace difícil satisfacer las actuales necesidades funcionales para la búsqueda de información. Su código fuente no es

muy legible y carece de documentación para desarrollar dentro de la herramienta en aras de extender sus funcionalidades.

De la situación problemática anteriormente descrita se deriva como **problema científico** a resolver:

¿Cómo contribuir a la búsqueda y recuperación de información en los sitios web de la red telemática del CNICM?

Queda definido como **objeto de estudio** de la investigación: los sistemas de búsqueda y recuperación de información, delimitado como **campo de acción**: los motores de búsqueda.

Para dar solución al problema planteado anteriormente se precisó como **objetivo general** de la investigación: implementar un motor de búsqueda capaz de satisfacer la búsqueda y la recuperación de información en los sitios web de la red del CNICM.

Para dar cumplimiento al objetivo general se hace necesaria la realización de las siguientes **tareas de la investigación**:

1. Sistematización de los sistemas de búsqueda y recuperación de la información en la web y las principales herramientas, tecnologías y metodologías a emplear en el desarrollo del motor de búsqueda.
2. Descripción de las historias de usuario a implementar.
3. Codificación del motor de búsqueda.
4. Validación del motor de búsqueda a través de las pruebas unitarias y de aceptación.

La investigación se sustenta en la siguiente **Idea a defender**: al aplicarse a los sitios web de la red telemática del CNICM la propuesta de solución: se obtendrán sitios web capaces de realizar búsquedas de documentos enriquecidos, imágenes y videos, utilizando su propia base de datos.

Métodos de la investigación

Un método de investigación provee estrategias elementales para ahorrar esfuerzo y tiempo en una investigación científica. Para guiar las tareas de la investigación se utilizaron los siguientes métodos científicos:

▪ Histórico – Lógico

Se realizó un análisis de los antecedentes y el estado actual de los sistemas de búsqueda y recuperación de información, principalmente los motores de búsqueda o buscadores, que sirvió de base fundamental de conocimiento para la realización de la propuesta de solución. Además para el análisis del comportamiento de los datos históricos que han caracterizado los buscadores desde sus inicios, y permitió llegar a conclusiones acerca de su futura evolución y trayectoria.

▪ Analítico – Sintético

Se analizó la bibliografía encontrada referente a los buscadores y mediante la síntesis del conocimiento, se determinaron de ella los elementos más importantes para la investigación.

▪ Inductivo – Deductivo

Se obtuvieron conocimientos generales de los sistemas de búsqueda y recuperación de información, partiendo de lo particular a lo general.

▪ Entrevista

Se realizaron entrevistas al cliente, con el fin de crear las HU y de hacer las pruebas de aceptación. El cliente fue entrevistado a lo largo del ciclo de vida de la propuesta de solución, ya que se integra al equipo de desarrollo.

Este trabajo ha sido organizado en tres capítulos que recogen todo lo abordado en la investigación.

Capítulo# 1 “Fundamentación teórica”: Se abordan todos los conceptos relacionados con los sistemas de búsqueda y recuperación de información, específicamente los motores de búsqueda o buscadores

como también se les conoce. También se aborda sobre los elementos teóricos de la investigación tales como las herramientas, tecnologías y metodologías de desarrollo de software.

Capítulo# 2 “Planeación y diseño del sistema”: Se describirá de forma sencilla y explícita el sistema como propuesta de solución al objetivo general planteado. Se describen los requisitos de software y de hardware. En este capítulo también se crearán las historias de usuario que son la columna vertebral para el desarrollo del sistema. Se hará una estimación de esfuerzo por HU, un plan de iteraciones, un plan de duración de cada iteración y un plan de entregas.

Se diseñará el sistema de forma sencilla según la metodología XP, se crearán las tarjetas CRC. También se tendrán en cuenta los patrones de arquitectura y diseño empleados en la solución. Se tendrá en cuenta el tratamiento de errores, la seguridad, el diseño de la interfaz y de la base de datos.

Capitulo #3 “Codificación y prueba”: Se implementan las funcionalidades identificadas a través de las tareas de la ingeniería describiéndose cada una de ellas. Se detallan las pruebas realizadas al *software* con el objetivo de asegurar la calidad y eficiencia de la solución.

Capítulo #1: Fundamentación teórica

Introducción

En este capítulo se abordan los conceptos relacionados a los sistemas de búsqueda y recuperación de información, específicamente los motores de búsqueda o buscadores como se les conoce. Se hará referencia a los elementos teóricos de la investigación tales como las herramientas, tecnologías y metodologías de desarrollo de software.

1.1 Los sistemas de búsqueda y recuperación de información

Muchos son los autores que han intentado dar una definición exacta de lo que es recuperación de información aunque todos observan el término desde diferentes aristas, todos concluyen en que la recuperación de información difiere sustancialmente de la recuperación de los datos. En este trabajo, se describirá el concepto dado por Ricardo Baeza-Yates [1] el cual sí hace una distinción más concisa entre recuperación de la información y recuperación de datos. Según Baeza-Yates *“los datos se pueden estructurar en tablas, árboles, para recuperar exactamente lo que se quiere”*.

Por otra parte, este autor considera que la recuperación de la información, se puede expresar como una *necesidad de información (consulta) y un conjunto de documentos ordenados de mayor a menor relevancia para esa necesidad y presentar un conjunto de aquellos con mayor relevancia* [2].

Un Sistema de Recuperación de Información se define como el *“proceso donde se accede a una información previamente almacenada, mediante herramientas informáticas que permiten establecer ecuaciones de búsquedas específicas. Dicha información ha debido ser estructurada previamente a su almacenamiento”* [3].

1.2 Clasificación de los Sistemas de Recuperación de Información

Los principales sistemas de búsqueda y recuperación de información existentes son: los motores de búsqueda, los directorios y los metabuscadores, a continuación se hace una breve descripción de cada uno de ellos.

1.2.1 Directorios

Son herramientas que organizan las páginas web jerárquicamente, o sea, permiten organizar la web por temas, lo que facilita la búsqueda de la información existente en un área determinada del conocimiento. Los resultados son recorridos en profundidad, lo que garantiza que al final de la jerarquía, exista una alta probabilidad de encontrar lo que realmente necesita el usuario. Además estos sistemas no poseen una araña u otro mecanismo automático que recorra la web en busca de nueva información como suele suceder con los motores de búsqueda, sino que es operado por humanos [5].

1.2.2 Metabuscadores

Un metabuscador no posee una base de datos propia sino que utiliza la base de datos de estos para encontrar la información solicitada por los usuarios. Su único trabajo consiste en combinar las mejores páginas que ha devuelto cada buscador, logrando así un mayor abanico de resultados con más calidad [6].

Hay que tener en cuenta que cada buscador utiliza su propia estrategia a la hora de recoger información de una página y a la hora de ordenar los resultados de las búsquedas, esto repercute en las páginas de mayor relevancia, ya que los resultados de un buscador no tienen por qué coincidir con los del resto. Aportando puntos de vistas distintos [6].

1.2.3 Motores de búsquedas

Permiten dado un criterio de búsqueda introducido por el usuario, obtener un subconjunto de aquellos documentos de mayor relevancia para dicho criterio de búsqueda, mediante la realización de ciertas operaciones sobre una base de datos.

Los motores de búsqueda son, programas informáticos formados por complejos algoritmos de búsqueda e indexación y enormes bases de datos que abarcan gran parte del contenido de la inmensa red que nos rodea, internet, y que son utilizados para que todo usuario pueda acceder a una información global [5]. Poseen una araña o mecanismo automático que recorre la web en busca de nueva información [5].

1.2.3.1 Arquitectura general de un motor de búsqueda

La mayoría de los motores de búsqueda emplean una arquitectura araña-indexador centralizada, es decir, la araña y el componente de indexación se encuentran unidos. La araña es la encargada de realizar peticiones a servidores distantes en busca de la información contenida en los mismos. Estas han evolucionado a un punto tal que permiten realizar peticiones por distintos protocolos de la familia TCP/IP (Protocolo de Control de Trasmisión y protocolo de internet) tales como: HTTP (Protocolo de Transferencia de Hipertextos), FTP (Protocolo de Trasmisión de Archivos), NNTP (Protocolo de Transferencias de Noticias en red), entre otros.

Por su parte, el componente encargado de la indexación, recibe las páginas recuperadas por la araña, extrae una representación interna de la misma y la almacena en forma de índices en una base de datos [7].

Muchos indexadores emplean técnicas avanzadas para la extracción de vocabulario tales como:

- Lista de palabras de parada o *stopwords*: son listas de palabras muy habituales que no aportan significado a la información. Por ejemplo, las preposiciones, los artículos, etc.
- Extracción de raíces o *stemming*: consiste en extraer la raíz de las palabras con significado parecido, por ejemplo, plurales, tiempos verbales y otras. Todas estas palabras obtenidas por el indexador, son almacenadas en una base de datos o archivo invertido en forma de índices, lo que facilita la recuperación de la información por el motor de búsqueda.

El motor de búsqueda, por su parte, recibe la consulta de un usuario, que consiste en la introducción de un grupo de palabras claves sobre la información deseada. Estas palabras claves son convertidas por el sistema de formulación de consultas en un conjunto de incógnitas entendibles por el sistema y las que serán utilizadas por el subsistema de evaluación para devolver los documentos existentes en la base de datos, otorgando un orden de relevancia a dichos documentos en correspondencia con la consulta originalmente introducida por el usuario [4].

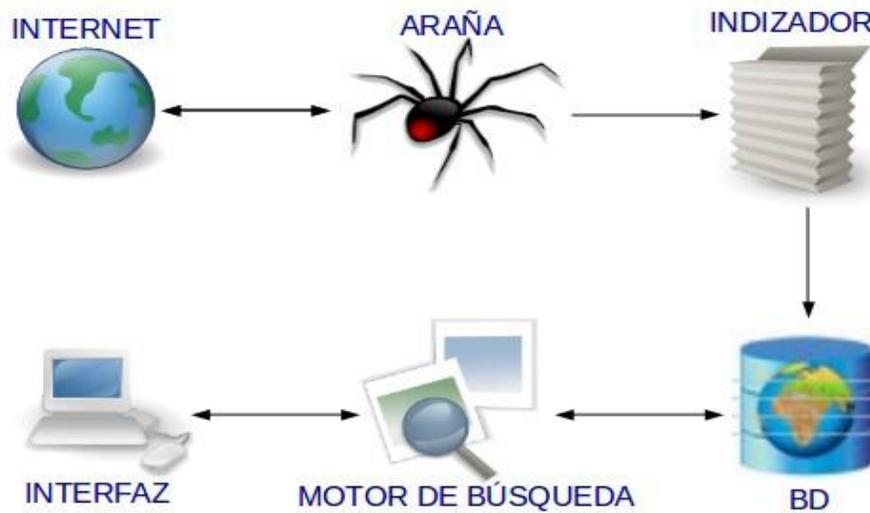


Fig. 1 Arquitectura de un motor de búsqueda

1.2.3.2 Los componentes de los motores de búsqueda

Están formados por cinco componentes:

- **Interfaz:** Es el componente en el que el usuario podrá introducir las palabras a buscar y donde posteriormente se mostrarán los resultados.
- **Programa Buscador:** Implementa métodos de búsqueda de información.
- **Programa Rastreador:** Implementa métodos de exploración de archivos almacenados en servidores web.
- **Programa Indexador:** Implementa métodos de indexación de información.
- **Base de Datos:** Contenedor de datos escalable que posibilita tratar con gran cantidad de información. Debe presentar una función básica, y es la existencia de una interfaz que todos los componentes que vayan a hacer uso de ella sepan interpretar.

1.2.3.3 Motores de búsqueda verticales

Son motores de búsquedas especializados en un sector, lo que le permite analizar la información con mayor profundidad que en los buscadores genéricos, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas [8].

Los motores de búsqueda verticales envían sus robots a un número limitado de páginas web sobre un tema concreto, lo que permite que tanto la obtención de la información como la creación del índice sean más especializadas en el sector del que se trata. Al tratarse de un número de fuentes más reducido que en un motor de búsqueda genérico los motores verticales pueden también actualizar su información con mayor frecuencia. Adicionalmente, un motor de búsqueda vertical ofrece a los usuarios mecanismos de búsqueda avanzada específicamente diseñados para el sector [9].

Los motores de búsqueda verticales o temáticos suponen una nueva generación después de los modelos de búsqueda horizontal como Google y Yahoo. La revista *Business Week* ha clasificado a los motores de búsqueda verticales como parte de un “renacimiento tecnológico”.

Se escogió este tipo de motor de búsqueda para aplicárselo a la propuesta solución porque es el que más se adecua a las necesidades del CNICM, ya que este desea centrar su búsqueda en el sector de la salud.

1.3 La Indexación

La indexación es la operación destinada a representar los resultados del análisis del contenido de un documento o de una parte del mismo, mediante elementos (términos de indexación) de un lenguaje documental o natural, orientados a facilitar la posterior recuperación de los documentos indexados.

El problema fundamental que se encuentra al intentar indexar la web, es la necesaria representación de los gráficos de la red con millones de nodos y, en consecuencia, crear índices a partir de millones de páginas. Por lo tanto, se deberá tener mucho cuidado a la hora de crear un índice a gran escala para conseguir su eficiencia. Se presenta a continuación una breve descripción de cada tipo de índice:

- **Índice basado en los hipervínculos:** Para construir un índice basado en los *links* una porción de la web es tratada como a un esquema con nodos y vectores. Cada nodo en esa estructura es una página y un vector que desde una página *A* apunta a otra página *B* representa un hipervínculo en

la página *A* que señala a la página *B*. Normalmente, la estructura más utilizada por los algoritmos de búsqueda es la información del vecino; por ejemplo, dada una página *P*, ordena la colección de páginas señaladas por *P* o la colección de páginas que señalan a *P*. En la siguiente figura se muestra la web basada en hipervínculos.

- **Índice de texto:** Aunque los índices basados en los hipervínculos entre páginas son utilizados mucho para mejorar la calidad y la relevancia de los resultados de búsqueda, los índices basados en largas colecciones de texto (que sirven por ejemplo para poder identificar ciertas palabras con páginas almacenadas) continúan siendo primordiales a la hora de definir la relevancia de los resultados. Pueden ser implementados utilizando varias estructuras, y hay dos muy importantes: *signature files*, que son índices estructurados implementando un modelo que asigna claves a bloques de texto, e índices invertidos.
- **Índices útiles:** El número y el tipo de índices útiles construidos depende de las características del algoritmo de búsqueda y de lo que ofrezca cada motor de búsqueda en particular. Por ejemplo, un buscador que otorgue la posibilidad de restringir las búsquedas a un dominio concreto o a un sitio específico, se beneficiará de un índice que relacione el nombre de tal dominio o sitio con la lista de páginas que le pertenecen. Del mismo modo, utilizando la *neighborhood information*, un algoritmo iterativo (que se repite bajo unas condiciones) puede computar y almacenar el valor de la relevancia que se le otorga a cierta página (por ejemplo, el algoritmo *PageRank*). Un índice de este tipo será utilizado en el momento en el que se necesiten ordenar los resultados obtenidos por una cierta consulta.
- **Índice invertido:** Los índices invertidos son los más utilizados hoy en día por los motores de búsqueda. Un índice invertido de una colección de páginas web consiste en un conjunto de listas invertidas, una para cada palabra (o término del índice). Una lista invertida para un término consiste en una lista de localizaciones de la colección donde el término aparece. En el más simple de los casos, una localización será el identificador de la página y la posición del término en la página. Sin embargo, los algoritmos de búsqueda se sirven usualmente de información adicional sobre la ocurrencia de términos en una página. Por ejemplo, términos que aparezcan en el cuerpo de una página, en los encabezamientos o en los hipervínculos serán tratados de forma distinta a la hora de evaluar el *ranking* de la página para ése término. Para conseguir que esta distribución sea

más eficiente, se crean campos donde la información adicional de un término se almacena. Por ejemplo, la mayoría de los índices de texto mantienen lo que se llama *lexicon*, que lista todos los términos que aparecen en el índice junto con estadísticas sobre cada uno de ellos (por ejemplo, número total de documentos en los que el término aparece) que son usados por los algoritmos de catalogación de relevancia para los sitios *web*.

- **Índice invertido local:** En este tipo de organización, cada nodo es responsable de una desunión de listas de páginas en una colección. De alguna manera, el algoritmo de búsqueda será en este caso el que se difunda a través de los nodos, que devolverán listas de páginas que contengan las palabras buscadas. A modo de ejemplo, consideremos la estructura E1, formada por las páginas {P1, P2,..., Pn}, una posible distribución siguiendo este método sería admitiendo una o varias páginas iniciales (Pi, Pj,..., Pk), y que de cada una de estas páginas se deriva una o más de una, y así sucesivamente.
- **Índice invertido global:** Esta organización esquematiza los términos indexados de manera que cada servidor almacena listas de sólo ciertos términos de la colección. Por ejemplo, en un sistema en el que hay dos servidores S1 y S2, el primero podrá almacenar una lista de términos que empiecen por las letras [a,b,...,q] y el segundo las restantes, de manera que el algoritmo de búsqueda accederá a S1 cuando esté buscando la palabra „proceso“.
- **Arreglo de Sufijos:** Son una implementación eficiente en espacio de los árboles de sufijos. Este tipo de índice permite responder eficientemente a consultas más complejas. Sus principales desventajas son que el texto debe estar disponible en el momento de la consulta y que los resultados no se recuperan en el mismo orden que su posición en el texto. Esta estructura puede utilizarse para indexar palabras (con o sin *stopwords*) o caracteres, lo que la hace adecuada para un amplio espectro de aplicaciones como las BD genéticas.

El arreglo de sufijos ve el texto como una cadena larga. Cada posición se considera como un sufijo de texto (i.e., una cadena que va desde dicha posición hasta el final). Dos sufijos que comienzan en posiciones diferentes serán lexicográficamente diferentes (se asume que existe una marca de final de texto). De este modo, cada sufijo se identifica de forma única por su posición en el texto.

1.4 Tecnología y herramientas

Siguiendo la política del departamento de programación del CNICM se usará para la implementación del motor de búsqueda las siguientes herramientas y tecnologías que se describen a continuación:

1.4.1 Apache Solr

Se usará *Apache Solr* como plataforma de búsquedas ya que el volumen de información que se manejará es muy alto y se hace necesario un mecanismo de búsqueda rápido y eficiente. *Apache Solr* funciona como un "servidor de búsquedas" y entre sus principales características se incluyen:

- Búsquedas de texto completo.
- Resaltado de resultados.
- Clustering dinámico.
- Manejo de documentos ricos (como Word y PDF).
- Realiza búsquedas distribuidas.
- Replicación de índices. [12].
- API estilo REST.

Ventajas:

- Es agnóstico del lenguaje porque usa XML y JSON, que hoy en día pueden ser interpretados por casi cualquier herramienta.
- Es agnóstico de los tipos de datos, ya que HTTP sólo transmite textos. Los lenguajes dinámicos como PHP tienen éxito porque su protocolo básico no tiene tipos estrictos.
- Se dice que es un protocolo estándar (aunque su representación de datos no lo es).

Solr se divide en dos partes:

1. **Índice:** Sistema de ficheros que almacenan la información. Contiene la configuración de *Solr* y la definición de la estructura de datos.

2. **Servidor:** Proporciona el acceso a los índices y las características adicionales. Admite plugins para añadir funcionalidades.

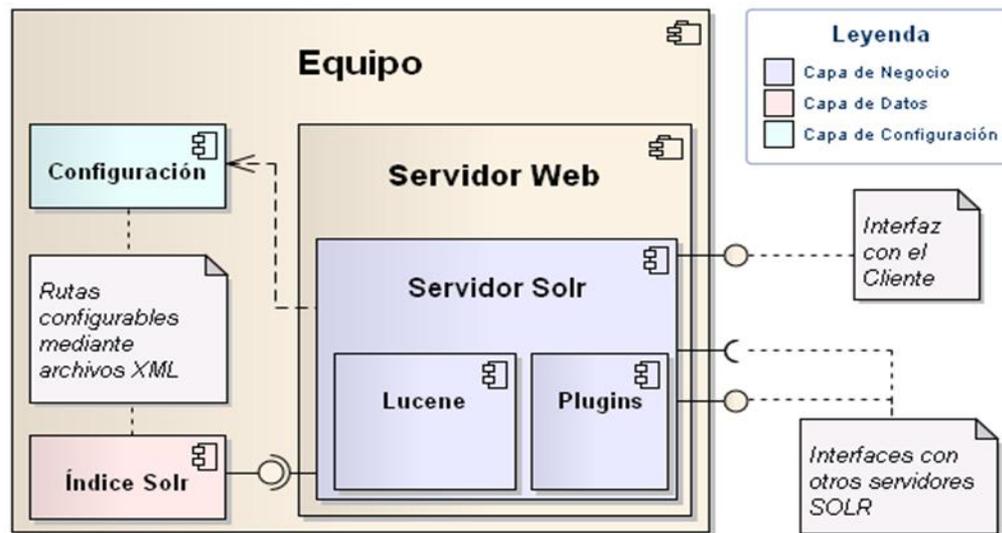


Fig. 2 Elementos de Apache Solr

1.4.2 Nutch

Se usará *Nutch* para llevar a cabo el proceso de indexación ya que actúa como araña para la recopilación de información. También permite haciendo uso de analizadores de varios formatos de archivos, extraer información desde archivos PDF, XML, HTML, ZIP, ODF, JS, RSS, entre otros. [14]

Posee una amplia comunidad de desarrolladores y usuarios. Su desarrollo está patrocinado por la fundación *Apache* y *Lucene Imagination*.

Entre sus principales características se encuentran:

- Captura, *parser* e indexación en modo paralelo y distribuido.
- Extensibles mediante *plugins*.
- Sistema de fichero distribuido.
- Soporte para autenticación.

1.4.3 Cakephp

Se usó *Cakephp* como *framework* de desarrollo porque puede manejar cualquier aspecto, desde la solicitud inicial del usuario hasta el renderizado final de la página *web*. Además, sigue los principios Modelo Vista Controlador (en adelante MVC), permitiendo personalizar y extender fácilmente muchos aspectos de esta. El *framework* también suministra una estructura de organización básica, desde los nombres de los archivos hasta los de las tablas de la base de datos, manteniendo la aplicación consistente y lógica [16].

Está caracterizado por:

- Compatible con PHP4 y PHP5.
- CRUD de la base de datos integrado.
- URLs amigables.
- Sistema de plantillas rápido y flexible.
- Ayudas para AJAX, *Java script*, HTML, *forms*.
- Trabaja en cualquier subdirectorio del sitio.
- *Scaffolding* (andamiaje) de las aplicaciones.
- Listas de Control de Acceso.
- Componentes de seguridad y sesión.

Ventajas:

- Licencia flexible, *Cakephp* está distribuido bajo la MIT *License*.
- Desarrollo rápido.
- Buenas prácticas, *Cakephp* es muy fácil de entender y cumple los estándares en seguridad y autenticación, manejo de sesiones y muchas otras características.

1.4.4 PHP

Como lenguaje de programación se usará PHP denominado preprocesador de hipertexto, ya que es un lenguaje libre y multiplataforma. Se sustenta en la actualidad bajo el paradigma más difundido actualmente en el mundo que es programación orientado a objeto, aunque incluye también la programación estructurada y servicios web. Presenta excelente integración con todos los sistemas gestores de base de datos. Cuenta con una biblioteca que trae un conjunto de funciones para realizar cualquier operación (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF, entre otros). PHP, es un lenguaje de programación usado normalmente para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. [17].

Principales ventajas de PHP [18]

- **Sintaxis cómoda:** PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Es decir, cuando se crea una variable no se indica de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse mucho las posibilidades del intérprete para detectar incompatibilidades entre variables.
- **Soporta objetos y herencia:** Tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores, etc. Además soporta herencia, aunque no múltiple.
- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página web con el código HTML resultante de la ejecución del código PHP compatible con todos los navegadores.
- Se puede incrustar código PHP con etiquetas HTML.
- **Compatibilidad con bases de datos:** Amplio soporte para una gran cantidad de bases de datos. Tiene acceso a un gran número de gestores de bases de datos: *Adabas D*, *dBase*, *Empress*,

Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc.

- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.
- **Multiplataforma:** Funciona tanto en sistemas *Unix* o *Linux* con servidor *web Apache* como en sistemas *Windows* con *Microsoft Internet Information Server*, de forma que el código generado por cualquiera de éstas plataformas no debe ser modificado al pasar a la otra.
- **Licencia de software libre:** Es un lenguaje basado en herramientas con licencia de *software* libre, es decir, no hay que pagar licencias, ni existen límites en su distribución y, es posible ampliarlo con nuevas funcionalidades.
- **Extensa librería de funciones:** Cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.

1.4.5 XAMPP

XAMPP traduce un conjunto de aplicaciones que permite establecer una plataforma web bien sea para desarrollo o para producción, y todas las aplicaciones de libre disposición en la red.

- La "X" cubre los sistemas operativos *Linux, Windows, Mac Os X* o *Solaris*. En la actualidad el *Linux* se ha posicionado fuertemente en el sector de los servidores y está siendo utilizado cada día más para soluciones de misión crítica.
- La "A" denota *Apache* como servidor *web*. El mismo simplemente se encarga de servir páginas estilo *web* hacia quien lo solicita, que por lo general es un PC o dispositivo con un visor de *web*. El *Apache* es el servidor de mayor número de instalaciones en la red. Inició como el reemplazo al servidor de la NSF (*National Science Foundation*) y su curva de crecimiento es exactamente opuesta a la de decrecimiento del mismo. Este servidor está inclusive incorporado en muchas de las soluciones que brindan las diferentes casas comerciales como parte de su oferta y como manejador de servicios *web*, dándole un respaldo total al servidor.
- Hoy en día, con el desarrollo de las tecnologías, no se percibe que un sitio que pretenda efectuar transacciones comerciales, brindar personalización y creación de comunidad, funcione sin una

base de datos como esquema de almacenamiento. La plataforma XAMPP incluye la "M" de *MySQL*, una base de datos de muy fácil manejo y de relativamente buen desempeño.

- El último componente está para los lenguajes de programación. Las ofertas que propone XAMP, todas iniciando por la "P" son *PHP* y *Perl*. De estos tres, el que hasta hace poco tenía la delantera y todavía presenta buena acogida entre programadores es el *Perl*. Normalmente permite la elaboración de programas de una manera sencilla, y permite la interacción y la manipulación de información. Recientemente, con el avance que ha tenido el *PHP* en su versión 4 y la reciente liberación de la versión 5 (actualmente se trabaja en la versión 6) que incorpora un nuevo modelo de POO (Programación Orientada a Objetos) y extensiones para el trabajo con *MySQL* entre otras funciones; es muy utilizado para soluciones web en la plataforma abierta. Tiene también la posibilidad de ser instalado en múltiples plataformas, lo que lo convierte en una buena elección para empresas que quieran desarrollar a bajo costo. El *PHP* se caracteriza en términos generales por proveer funciones de conexión con las bases de datos y funciones de presentación para que el resultado sea presentado al usuario en formato de página web. La cantidad de aplicaciones ya disponibles en *PHP* es amplia.

Las soluciones que ofrecen las casas comerciales se encargan de hacer que los cuatro componentes que hemos analizado: sistema operativo, servidor web, base de datos y lenguaje de programación; funcionen sincronizada y adecuadamente dentro de un conjunto de equipos. La propuesta XAMPP pretende ser la respuesta a las casas comerciales con software que hasta ahora ha dado buenos resultados en forma separada y que promete siempre y cuando se logre consolidar el soporte para cada una de las herramientas (base de datos y lenguajes en especial) y garantizar la facilidad de instalación de cada uno de los componentes para permitir una integración transparente. XAMPP provee un robusto y flexible ambiente para el desarrollo de aplicaciones web, es un servidor independiente de plataforma, software libre, fácil de usar y capaz de interpretar páginas dinámicas.

XAMPP es regularmente actualizado para incorporar las últimas versiones de *Apache/MySQL/PHP* y *Perl*. También incluye otros módulos como *OpenSSL* y *phpMyAdmin*, viene con un administrador de servicios integrado que se instala como ícono en la bandeja del sistema, que permite administrarlo fácilmente y acceder a todos los servicios. El programa está liberado bajo la licencia GNU. Las aplicaciones web debido a las exigencias de sus propósitos funcionales imponen la necesidad de usar para su desarrollo

lenguajes de programación con características que faciliten su óptimo desempeño. La incorporación de bibliotecas o paquetes de clases, la disponibilidad de documentación son factores fundamentales en la selección de una plataforma de desarrollo determinada.

1.4.6 Metodología de desarrollo de software

Las metodologías son una parte importante en el desarrollo de un software, son las que definen dentro de cualquier proyecto ¿Quién debe hacer?, ¿Qué?, ¿Cuándo y Cómo debe hacerlo? [19]. Usar una metodología ayuda a que el trabajo sea organizado y estructurado.

Metodologías tradicionales: están definidas para procesos de desarrollo a largo plazo. Los métodos usados no son flexibles ante la posibilidad de nuevos requerimientos, estos cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto [20].

1.4.6.1 Comparación entre metodologías ágiles y tradicionales

Las metodologías ágiles están basadas en heurísticas provenientes de prácticas de producción de código, están especialmente preparadas para cambios durante el proyecto ya que es un proceso menos controlado y con pocos principios. No existe contrato tradicional o al menos es bastante flexible, el cliente es parte del equipo de desarrollo. Los grupos de trabajos son pequeños y trabajando en el mismo sitio, estas metodologías generan pocos artefactos y cuenta con pocos roles. La metodología ágil no hace énfasis en la arquitectura del *software*. Mientras que las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, muestran resistencia a los cambios. Estas metodologías tradicionales cuentan con un proceso mucho más controlado con numerosas políticas y normas, existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones. Los equipos de trabajo de las tradicionales son grandes y distribuidos; esta metodología genera gran cantidad de artefactos y cuenta con mucho más roles que las metodologías ágiles. En las metodologías tradicionales la arquitectura del software es esencial y se expresa mediante modelos.

1.4.6.2 Metodologías ágiles

El trabajo con las metodologías ágiles se caracteriza por tener un desarrollo iterativo y adaptable. Permite la integración de nuevas funcionalidades a lo largo del desarrollo del proyecto. El cliente y los

desarrolladores trabajan juntos constantemente y existe buena comunicación. Se caracterizan por generar poca documentación, simplicidad y pruebas diarias [21]. A continuación se hace una breve síntesis de las metodologías ágiles.

- **SCRUM:** Desarrollada por *Ken Schwaber, Jeff Sutherland y Mike Beedle*. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos:

El desarrollo de *software* se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.

La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [22]

- **Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de *software* se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo *Crystal Clear* (3 a 8 miembros) y *Crystal Orange* (25 a 50 miembros). [23]
- **Dynamic Systems Development Method6 (DSDM):** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases. [24]

- **Adaptive Software Development⁷ (ASD):** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. [25]
- **Feature-Driven Development (FDD):** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad. [26]
- **Lean Development (LD):** Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios. [27]
- **XP (Extreme Programming):** Esta metodología está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. La XP por sus siglas en inglés *Extreme Programming* se basa en la realimentación continua entre el cliente y el equipo de desarrollo. Se recomienda para proyectos con requisitos imprecisos y muy cambiantes donde existe un alto riesgo técnico [28]. El ciclo de vida de XP está formado por 4 fases (planeación, diseño, codificación y prueba).

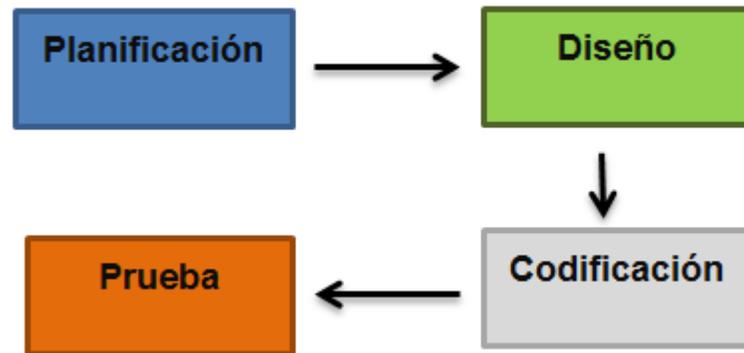


Fig. 3 Fases de desarrollo de XP.

Planificación: Es la primera fase de la metodología donde se definen las Historias de usuario, Plan de entrega y se definen las iteraciones que tendrá el proyecto.

Diseño: Se propone un diseño sencillo para lograr una mejor comprensión. Se propone el uso de las tarjetas Clases, Responsabilidades, Colaboradores.

Codificación: La codificación debe hacerse contando con el cliente como parte del equipo de desarrollo, atendiendo a estándares de codificación ya creados. La programación debe ser en pareja, pues permite obtener códigos más eficiente y de calidad.

Prueba: Se realizan las pruebas unitarias y de aceptación, el propósito es asegurar el funcionamiento final de una determinada historia de usuario.

Fundamentación de la elección:

Luego de un análisis de algunas de las metodologías ágiles se decide utilizar XP ya que propone una estructura de roles adaptada al proyecto y permite la realimentación continua entre el cliente y el equipo de desarrollo. Esta metodología ofrece simplicidad en las soluciones implementadas y a la hora de enfrentar los cambios. La metodología XP es la adecuada para guiar al desarrollo del motor de búsqueda ya que los requerimientos del mismo son imprecisos y muy cambiantes.

1.5 Sistema de gestión de base de datos

Un Sistema Gestor de Bases de Datos (SGBD) que permite definir bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación.

1.5.1 MySQL

MySQL es un SGBD Relacional, licenciado bajo la GPL de la GNU. Su diseño multihilos le permite soportar una gran carga de forma muy eficiente. Actualmente existen infinidad de librerías y otras herramientas que posibilitan su uso, a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Las principales características de este gestor de bases de datos son las siguientes [31] y [32]:

- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos.
- Es multiplataforma ya que trabaja tanto con sistemas operativos basados en *Unix* como con el sistema operativo de *Microsoft*.

Conclusiones parciales

El estudio de los principales SRI posibilitó escoger a los motores de búsqueda como la solución que más se adecua a las necesidades del CNICM. Se estudiaron motores de búsqueda existentes en internet como Google y Yahoo, con el fin de encontrar características que se pudieran tener en cuenta a la hora de la implementación del motor de búsqueda. El estudio de la arquitectura de los motores de búsqueda sirvió para crear la arquitectura del motor de búsqueda a implementar.

Teniendo en cuenta que el departamento de programación del CNICM tiene como política de trabajo la utilización de las herramientas y metodologías descritas en este capítulo se concluye que se utilizará en el

desarrollo del motor de búsqueda: Apache Solr versión 4.0 como único servidor de búsquedas, Nutch para la indexación de las fuentes de información de salud, ya que funciona como araña, como lenguaje de programación, PHP versión 5.2.8, como framework de desarrollo a Cakephp versión 2.3.6 y como SGBD a MySQL versión 5.1. El desarrollo estará guiado por la metodología para el desarrollo ágil XP.

Capítulo# 2: Planeación y diseño del sistema

Introducción

Se describirá de forma sencilla y explícita el motor de búsqueda como propuesta de solución al objetivo general planteado. Se crearán las historias de usuario que son la columna vertebral para el desarrollo del sistema. Se hará una estimación de esfuerzo por Historia de Usuario (en lo adelante HU), un plan de iteraciones, un plan de duración de cada iteración y un plan de entregas. Se diseñará el sistema según la metodología XP, se crearán las tarjetas Clases Responsabilidad y Colaboración (en lo adelante CRC). También se tendrán en cuenta los patrones de arquitectura y diseño empleados en la solución. Se tendrá en cuenta el tratamiento de errores, el diseño de la interfaz y de la base de datos.

2.1 Propuesta del motor de búsqueda

El motor de búsqueda se compone por cinco elementos fundamentales: Una interfaz, servidor de búsqueda *Apache Solr* y una base de datos MySQL.

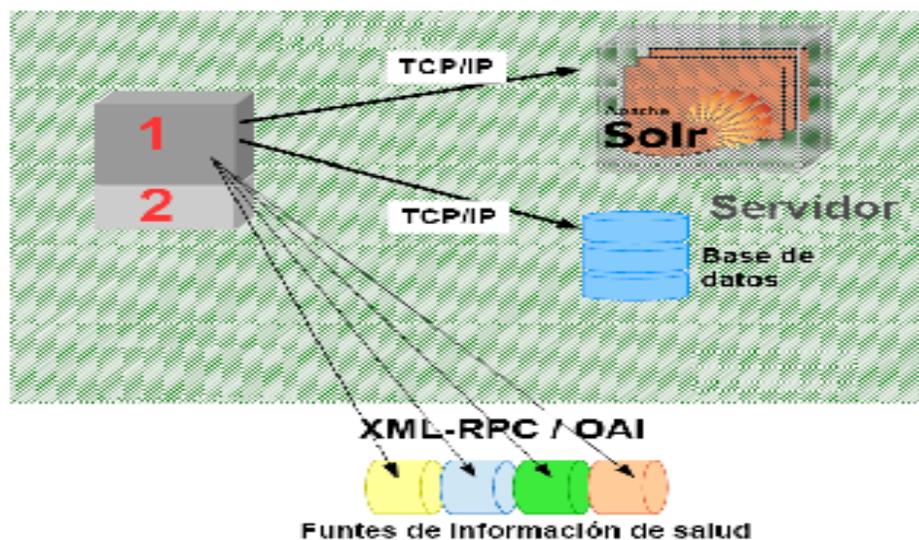


Fig. 4 Principales elementos del motor de búsquedas. Elaboración propia.

La aplicación se divide en dos partes, una para los administradores del sistema otra que brinda como interfaz un servicio de búsqueda y contiene el controlador de búsquedas en Solr.

Administración: es la parte de la aplicación gestora contenido, las fuentes, en el buscador. Dispone de las funcionalidades añadir, actualizar, guardar y eliminar para el manejo de las Fuentes de Información de Salud (en lo adelante FIS), manejando de estas toda la información demandada para conectarse con dichos servidores y obtener los datos a indexar.

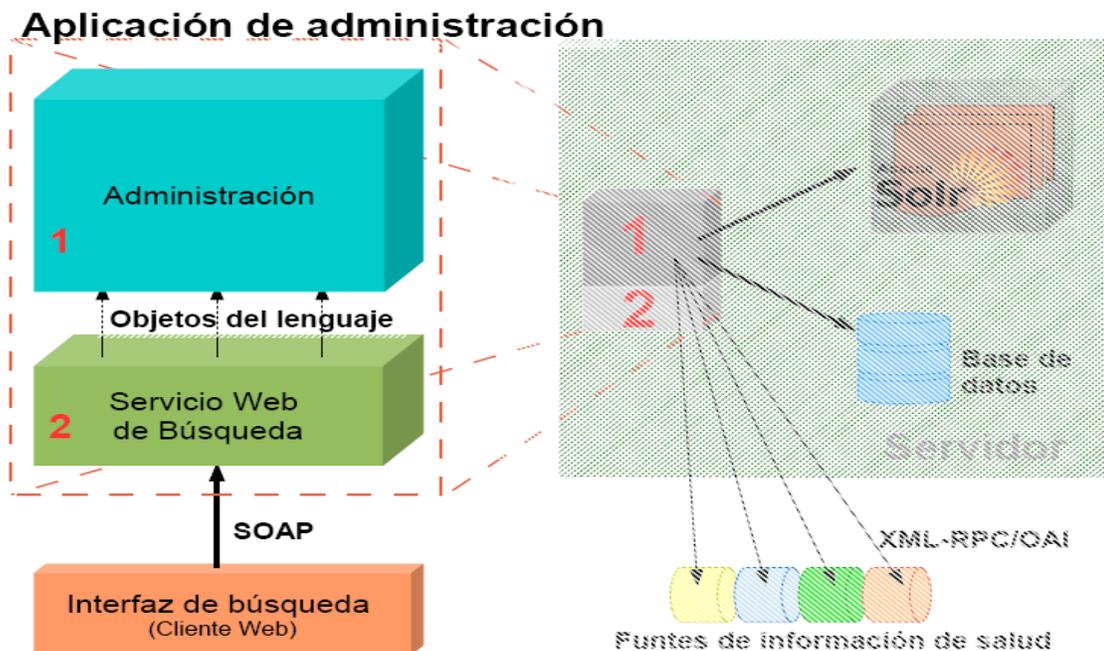


Fig. 5 Componentes del motor de búsqueda. Elaboración propia.

Interfaz de Búsqueda: puede ser un cliente web cualquiera que haga uso de las funciones de búsqueda a través de un servicio web de la aplicación de *Administración*. Esta interfaz es para uso de los administradores del sistema. Es objetivo lograr una interfaz sencilla, pero funcional que les permita administrar el motor de búsqueda.

Servicio web de búsqueda: se concibe como un servicio independiente que mediante una interfaz de servicio web brinda todas las funcionalidades necesarias para realizar búsquedas en recursos de información.

Base de datos: con un diseño simple, es para uso exclusivo del motor de búsqueda, en ella se albergarán todos los datos referentes a FIS, como tipo de servicio web, host, estado de indexación, disponibilidad de búsqueda, programación de indexado y demás.

Servidor Solr: En sus índices, estará guardada toda la información disponible para las consultas de los usuarios extraída de las FIS.

Cada fuente de información a indexar para el motor de búsqueda debe poseer una interfaz estándar XML-RPC u OAI, para la conexión a ella.

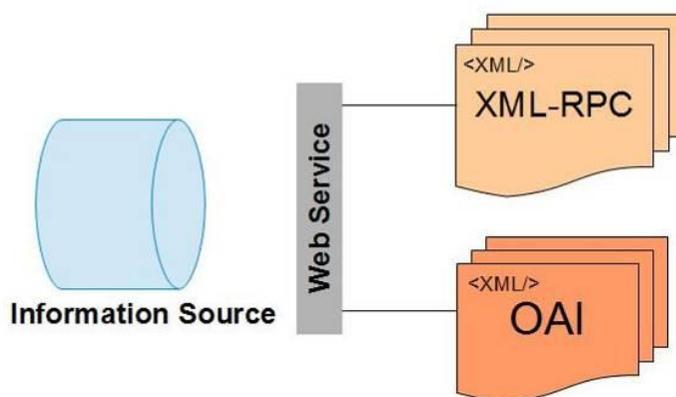


Fig. 6 Tipo de servicio web. Elaboración propia.

2.2 Planeación

2.2.1 Personal relacionado con el motor de búsqueda

Para determinar al personal relacionado con el sistema se hace referencia, a todas las personas que de una forma u otra estén vinculadas con el mismo, ya sea en el proceso de desarrollo o interactuando con él.

Tabla 1. Personal relacionado con el motor de búsqueda

Personal relacionado con el motor de búsqueda	Justificación
Rol de administrador	Son todas aquellas personas que interactúan con el motor de búsqueda.
Desarrolladores	Son las personas que se van a encargar de implementar las características definidas por el cliente para lograr el desarrollo del motor de búsqueda.

Planificación es la primera fase de la metodología XP donde el cliente establece la prioridad de cada HU y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Además define el Plan de entrega y se proponen las reuniones diarias donde el programador expone sus problemas.

2.2.2 Historias de Usuario

Las historias de usuario son utilizadas como herramientas para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que realizará el sistema; la redacción de las mismas se realiza bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles.

Tabla 2: HU. Añadir FIS

NÚMERO HISTORIA: 01
NOMBRE LA HISTORIA: Añadir Fuente de Información de Salud.
FECHA: 15 de marzo de 2014.
ENTREVISTADO: Cliente.
TIEMPO ESTIMADO: 1 Hora.
DESCRIPCIÓN: El motor de búsqueda debe permitir añadir una fuente de información de salud.

Tabla 3: HU. Recuperar FIS

NÚMERO HISTORIA: 02
NOMBRE LA HISTORIA: Recuperar Fuente de Información de Salud
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitir recuperar una fuente de información de salud.

Tabla 4: HU. Actualizar FIS

NÚMERO HISTORIA: 03
NOMBRE LA HISTORIA: Actualizar Fuente de Información de Salud
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitir actualizar una fuente de información de salud.

Tabla 5: HU. Eliminar FIS

NÚMERO HISTORIA: 04
NOMBRE LA HISTORIA: Eliminar Fuente de Información de Salud
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitir eliminar una fuente de información de salud.

Tabla 6: HU. Indexar FIS

NÚMERO HISTORIA: 05
NOMBRE LA HISTORIA: Indexar Fuente de Información de Salud
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitir almacenar las fuentes de información de salud.

Tabla 7: HU. Búsqueda de información

NÚMERO HISTORIA: 06
NOMBRE LA HISTORIA: Búsqueda de información
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: Debe permitir buscar documentos tales como pdf, Word, además de videos, partiendo de la petición que haga el usuario.

Tabla 8: HU. Añadir perfil de usuario

NÚMERO HISTORIA: 07
NOMBRE LA HISTORIA: Añadir perfil de usuario
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitirle al usuario crear su propio perfil de usuario.

Tabla 9: HU. Actualizar perfil de usuario

NÚMERO HISTORIA: 08
NOMBRE LA HISTORIA: Actualizar perfil de usuario
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente

TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitirle al usuario actualizar su propio perfil.

Tabla 10: HU. Eliminar perfil de usuario

NÚMERO HISTORIA: 09
NOMBRE LA HISTORIA: Eliminar perfil de usuario
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitirle al usuario eliminar su propio perfil.

Tabla 11: HU. Listar perfiles de usuarios

NÚMERO HISTORIA: 10
NOMBRE LA HISTORIA: Listar perfiles de usuario
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: El motor de búsqueda debe permitirle al usuario listar los perfiles que han sido creados.

Tabla 12: HU. Autenticar usuario

NÚMERO HISTORIA: 11
NOMBRE LA HISTORIA: Autenticar usuario
FECHA: 15 de marzo de 2014
ENTREVISTADO: Cliente
TIEMPO ESTIMADO: 1 Hora
DESCRIPCIÓN: La autenticación a implementar debe ser sencilla. Que todos los usuarios del sistema tengan los mismos derechos como rol administrativo.

2.2.3 Estimación de esfuerzos por HU

Las HU deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogan directamente con el cliente para obtener todos los detalles necesarios. Inicialmente, el equipo de desarrolladores estima el esfuerzo necesario para implementar las HU y los clientes aprueban los objetivos y tiempos de entrega, la estimación temporal se basa en un cálculo estimado por parte de los desarrolladores de cada una de las HU.

Las estimaciones realizadas en esta fase son primarias y podrían variar cuando se analicen más en detalle en cada iteración. Para la realización del motor de búsqueda se utilizarán las semanas como medida para la estimación de esfuerzos establecidas por los programadores. A continuación se muestra la tabla Estimación de esfuerzos:

Tabla 13. Estimación del esfuerzo por HU

No	Historias de Usuario	Estimación (por semanas)
1	Añadir FIS	0.2
2	Recuperar FIS	0.2
3	Actualizar FIS	0.2
4	Eliminar FIS	0.2
5	Indexar FIS	0.2
6	Búsqueda de información	0.2
7	Crear perfil de usuarios	0.2
8	Actualizar perfil de usuarios	0.2
9	Eliminar perfil de usuarios	0.2
10	Listar perfil de usuarios	0.2
11	Autenticar usuarios	0.2

2.2.4 Plan de iteraciones

Todo proyecto que emplea metodología XP debe dividirse en iteraciones. Las iteraciones son fases o etapas de la implementación donde se obtienen resultados en un tiempo estimado. En el plan de iteraciones se especifican detalladamente el orden de desarrollo de las HU dentro de cada iteración conjuntamente con la duración de las mismas.

Iteración 1: En esta iteración se implementan las HU que tienen prioridad alta en el negocio, las principales son las HU 1, 2, 3, 4 y 5 las cuales son de vital importancia para el sistema ya que conforman la base de la estructura del mismo. Con esta iteración se obtiene la primera versión de la aplicación la cual se utilizará para mostrar al cliente y permitirá al grupo de trabajo tener una retroalimentación.

Iteración 2: En esta iteración se realiza la implementación de las HU 6, 7, 8, 9 y 10. Con esta iteración se corrigen errores o inconformidades del cliente con las HU implementadas en la iteración anterior. De esta forma se obtiene la segunda versión del sistema. Esta segunda iteración es también mostrada al cliente con el objetivo de evaluar las aceptaciones por parte del cliente con el *software*.

Iteración 3: En esta iteración se realiza la implementación de la HU 11 corrigiéndose los errores de las iteraciones anteriores.

2.2.5 Plan de duración de iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en días que demora implementar cada historia de usuario. El objetivo del plan de duración de iteraciones es especificar detalladamente el orden de desarrollo de las HU dentro de cada iteración y la duración de las iteraciones.

Tabla 14. Plan de duración de iteraciones

Iteraciones	Orden de las HU a implementar	Duración (semanas)
Iteración 1	Añadir FIS Recuperar FIS Actualizar FIS Eliminar FIS Indexar fuente de información de salud	1.5
Iteración 2	Búsqueda de información Crear perfil de usuario Actualizar perfil de usuario Eliminar perfil de usuario Listar perfil de usuario	1.5
Iteración 3	Autenticar usuario	0.2

2.2.6 Plan de entregas

En el plan de entregas se establecen qué HU son agrupadas para conformar una entrega y el orden de las mismas.

Tabla 15. Plan de entregas

Historias de usuarios	Primera iteración	Segunda iteración	Tercera iteración
Añadir FIS	V 1.0	Finalizado	
Recuperar FIS	V 1.0	Finalizado	
Actualizar FIS	V 1.0	Finalizado	
Eliminar FIS	V 1.0	Finalizado	
Indexar fuente de información de salud	V 1.0	Finalizado	
Indexar fuente de información de salud	V 1.0	Finalizado	
Búsqueda de información		V 1.0	Finalizado
Crear perfil de usuario		V 1.0	Finalizado
Actualizar perfil de usuario		V 1.0	Finalizado
Eliminar perfil de usuario		V 1.0	Finalizado
Listar perfil de usuario		V 1.0	Finalizado
Autenticar usuario			V 1.0 / Finalizado

2.3 Lista de Reserva del Producto

La Lista de Reserva del Producto (LRP) es uno de los artefactos generados por la metodología XP, que contiene el listado de las HU representadas anteriormente, ordenadas por prioridad de implementación: Muy Alta, Alta, Media y Baja. Los requisitos no funcionales siempre estarán clasificados con prioridad baja.

Tabla 16. Lista de Reserva del Producto

Item*	Descripción	Estimación	Estimado por
Prioridad: Muy Alta			
1	Añadir FIS	0.2	Analista
2	Recuperar FIS	0.2	Analista
3	Actualizar FIS	0.2	Analista
4	Eliminar FIS	0.2	Analista
5	Indexar FIS	0.2	Analista
Prioridad: Alta			
6	Búsqueda Información	0.2	Analista
7	Crear perfil de usuario	0.2	Analista
8	Actualizar perfil de usuario	0.2	Analista
9	Eliminar perfil de usuario	0.2	Analista
Prioridad: Media			
10	Listar perfil de usuario	0.2	Analista
11	Autenticar usuario	0.2	Analista
Prioridad: Baja			
12	Disponibilidad		

	El sistema debe permanecer disponible las 24 horas del día. En caso de la ocurrencia de una falla técnica o eventualidad no prevista, el sistema no estará disponible hasta tanto no sea solucionada la falla.		
13	Hardware 1 GB de memoria RAM. Conexión de red Ethernet.		
14	Software Servidor Apache Solr 4.0 Servidor de Base de Datos: MySQL 5.1 o superior. Lenguaje de Programación PHP versión 5.2.8 o superior.		
15	Interfaz Interfaz simple, intuitiva y agradable al usuario. No hacer uso de más de 3 tonalidades de colores. Textos visibles y con buena tipografía.		

2.4 Diseño del sistema

En lo que respecta a la simplicidad del diseño, se acogió la recomendación de XP. Solo invirtiendo el tiempo exclusivamente necesario para la elaboración de diagramas y diseño de interfaces gráficas. Al no haber hecho muchos diagramas, la orientación a objeto no fue tan completa, sacrificando la escalabilidad, versatilidad y elegancia del diseño, lo que fue considerado un precio justo a cambio del cumplimiento de los plazos. Desde el punto de vista de las interfaces, tampoco se invirtió mucho tiempo en su diseño, sin embargo se prestó mucha atención al ubicar los elementos tal como el cliente los había solicitado.

Es importante aclarar que los sacrificios en ningún momento representaron una baja calidad de la aplicación en cuanto a funcionalidad se refiere.

2.4.1 Modelo de dominio

Para un mejor entendimiento del dominio del problema, se representan las relaciones entre los principales conceptos asociados a la problemática. A continuación se describen cada uno de los conceptos:

- **Usuario:** Toda persona que interactúa con el metabuscador del CNICM.
- **Metabuscador:** Aplicación de búsqueda de información del CNICM.
- **Buscadores de internet:** Son motores de búsqueda que publican servicios para responder a la solicitud realizada por el metabuscador.
- **Google:** Buscador de Internet.
- **Yahoo:** Buscador de Internet.
- **Bing:** Buscador de Internet.

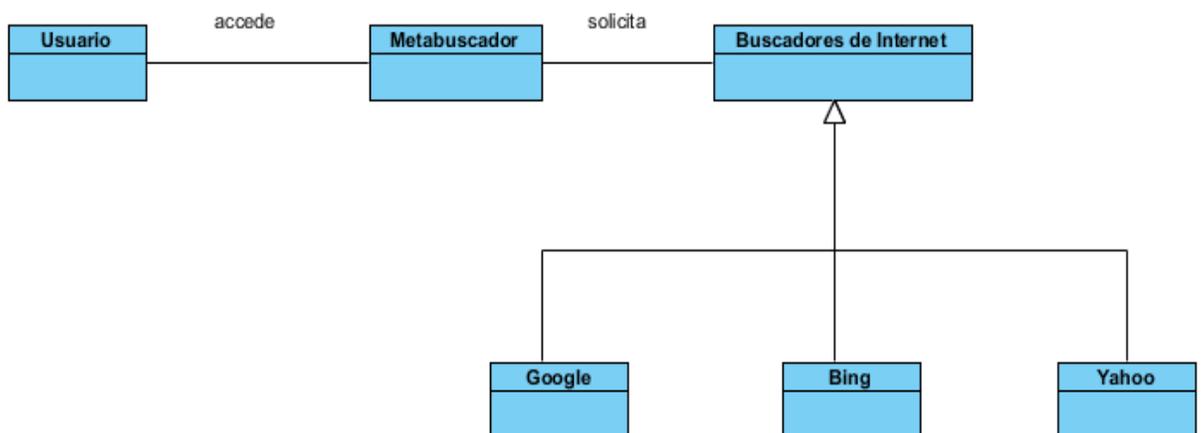


Fig. 7 Modelo de dominio. Elaboración propia

2.4.2 Tarjetas CRC

Una de las principales piezas de diseño empleadas en el proyecto fueron las tarjetas CRC, que no solo sirvieron como columna vertebral de este, sino que también fueron la base para la realización del modelo entidad-relación, elaborado para modelar la base de datos.

Tabla 17. Descripción de la tarjeta CRC user

Clase: user	
Responsabilidad	Colaboración
Crear usuario Actualizar usuario Eliminar usuario Autenticar Usuario	AppModel

Tabla 18. Descripción de la tarjeta CRC AppModel

Clase: AppModel	
Responsabilidad	Colaboración
	Model

Tabla 19. Descripción de la tarjeta CRC log

Clase: log	
Responsabilidad	Colaboración
Indexar FIS	AppModel

Tabla 20. Descripción de la tarjeta CRC sources

Clase: sources	
Responsabilidad	Colaboración
Crear FIS Recuperar FIS Actualizar FIS Eliminar FIS	AppModel

Tabla 21. Descripción de la tarjeta CRC search

Clase: search	
Responsabilidad	Colaboración

Búsqueda	AppModel
----------	----------

Tabla 22. Descripción de la tarjeta CRC services

Clase: services	
Responsabilidad	Colaboración
Contiene los servicios que se consumen.	

2.4.3 Patrones de arquitectura

2.4.3.1 Patrón Modelo Vista Controlador

El framework seleccionado para el desarrollo del sistema a implementar, está desarrollado bajo las restricciones, suposiciones y recomendaciones de la arquitectura Modelo-Vista- Controlador. MVC sugiere la separación del software en tres componentes: Modelo, Vista y controlador, los cuales serán explicados brevemente:

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí lo constituyen los datos puros y la lógica de los propios datos que puestos en el contexto del sistema proveen de información al usuario y en algunos casos a la propia aplicación.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción del administrador con el motor de búsqueda.

Controlador: Es la parte encargada de manejar y responder las solicitudes del usuario, procesando toda la información necesaria y modificando el Modelo en caso de ser necesario [27].

Ciclo de vida del MVC

El ciclo de vida empieza cuando el usuario hace una solicitud al controlador con información sobre lo que él desea realizar. Entonces el controlador decide a quién debe delegar la tarea y es aquí donde el modelo empieza su trabajo. En esta etapa, el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Luego de terminada su labor, le regresa al controlador la información resultante de sus operaciones, el cual a su vez redirige a la vista la cual se encarga de transformar los datos en información visualmente entendible para el usuario. Para un mejor entendimiento de este ciclo el siguiente diagrama lo representa de manera sencilla [27]:

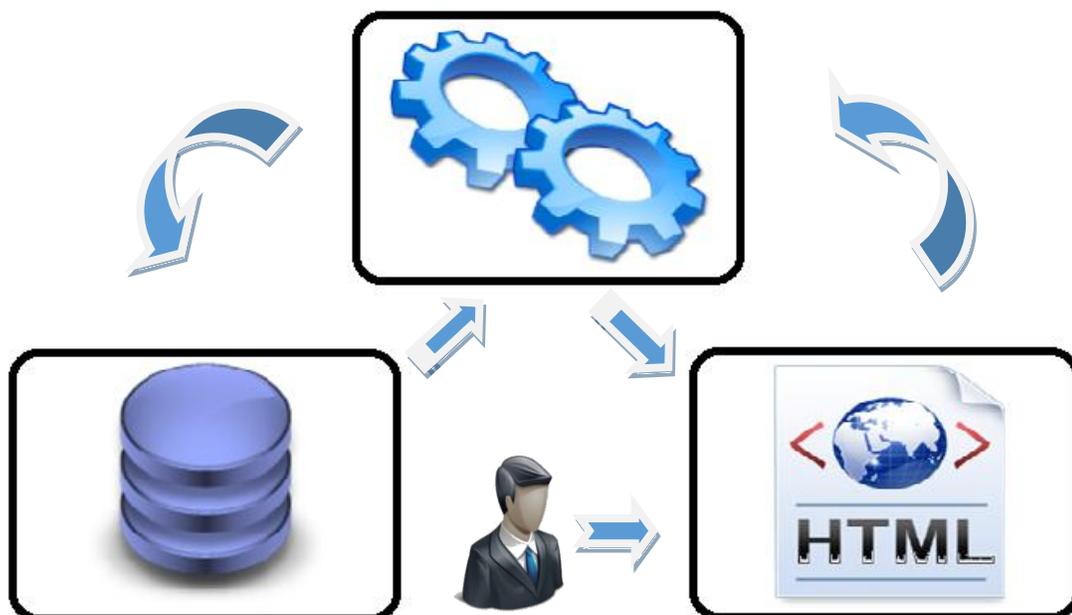


Fig. 8 Ciclo de vida de Modelo-Vista-Controlador. Elaboración propia.

2.4.4 Patrones de diseño

Con el desarrollo del modelo orientado a objeto, OOP (del inglés: *Object Oriented Programming*) el uso de los patrones de diseño ha tomado un auge importante, hoy en día son muy populares y están presentes en casi todos los sistemas informáticos a nivel mundial.

Un patrón de diseño constituye la descripción de clases y objetos involucrados en la solución adaptada de un problema de diseño general en un contexto particular, hace especial énfasis en la forma de comunicarse de los objetos. Identifica las clases, instancias, roles, colaboraciones, que contribuyen a lograr que el código obtenido sea flexible y que satisfaga los criterios especificados. Un patrón de diseño es una manera más práctica de describir aspectos organizacionales de la organización de un problema.

En el desarrollo de la solución planteada, se usaron varios patrones de diseño con el objetivo de lograr un software más acabado, flexible y de mayor calidad. Entre los distintos patrones se pueden citar los encargados de describir los principios fundamentales durante la asignación de responsabilidades a objetos GRASP (del inglés: *General Responsibility Assignment Software Patterns*). A continuación se describen las principales características de los patrones usados.

2.4.4.1 GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones [28].

Experto: Al implementar un sistema es importante saber asignarle a cada clase el grupo de operaciones que debe llevar a cabo, el patrón experto recomienda asignarle a una clase solo aquellas responsabilidades para las que ella contenga la información necesaria para su cumplimiento. Durante el desarrollo de la aplicación se llevó a cabo el uso de este patrón de diseño con el propósito de disminuir el tiempo de respuesta a cada petición realizada por el cliente así como la organización del contenido.

Este patrón se evidencia en la clase User ya que la misma cuenta con toda la información para poder realizar las operaciones relacionadas con los usuarios.

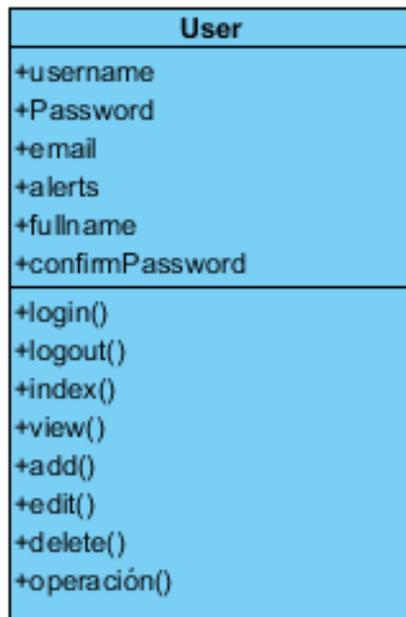


Fig. 9 Clase User. Elaboración propia.

Creador: Las peculiaridades del sistema obligan a hacer un uso óptimo de la memoria del sistema, es determinante saber asignar la responsabilidad de crear una instancia de una clase solo aquella clase que la requiera y así evitar que se creen objetos repetidos.

Este patrón se evidencia entre la clase `User` y la clase `AppControler`, ya que `AppControler` es la encargada de crear las instancias de las clases. En este caso se muestra la relación entre la clase `User` y la clase `AppControler`.

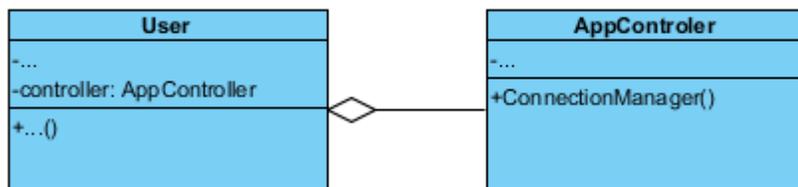


Fig. 10 Relación entre las clases `User` y `AppControler`. Elaboración propia.

Bajo Acoplamiento: Con este patrón se persigue lograr poca dependencia entre las clases, evitando así que cualquier cambio que pueda suceder durante el desarrollo del motor de búsqueda no sea de gran repercusión. La clase `AppControler` no depende de ninguna clase para realizar sus funciones.

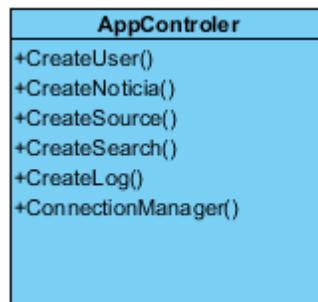


Fig. 11 Clase `AppControler`. Elaboración propia.

2.4.5 Tratamiento de errores

El sistema posibilita un control interno para el tratamiento de los posibles errores que puedan ocasionarse durante la ejecución de alguna acción en el servidor o en el proceso de comunicación con el mismo, con el fin de garantizar la integridad y confiabilidad de la información que se maneja. Se lleva a cabo un análisis del error ocurrido detectando la gravedad del mismo iniciando una serie de acciones y medidas en pos de mantener la integridad y funcionalidad del servidor.

En el desarrollo del sistema se han detectado una serie de acciones que pueden provocar un mal funcionamiento del servidor, especificando para cada una de ellas un mecanismo de recuperación. Los mensajes de errores son traducidos a un lenguaje de fácil comprensión para el cliente. Algunos de los mensajes de error que pueden llegar a la aplicación cliente son los siguientes:

Tabla 23. Descripción de errores

Error	Causa	Respuesta del Sistema
ERROR_USUARIO_DESCONOCIDO	El cliente trata de acceder al sistema con un usuario desconocido, que no ha sido registrado en el sistema.	Notificar al usuario que el nombre de usuario y/o contraseña no son los correctos.
ERROR_ANADIR_USUARIO_CAMPOS_VACÍOS	El cliente trata de añadir un nuevo usuario y deja campos sin especificar.	Notifica al usuario que no se puede añadir, por favor reinténtelo.
ERROR_AÑADIR_FIS_CAMPOS_VACÍOS	El cliente trata de añadir una FIS y deja campos sin especificar.	Notifica al usuario que no se puede añadir, por favor reinténtelo.
ERROR_CONTRASEÑA_DESCONOCIDA	El cliente trata de acceder al sistema con una contraseña desconocida, que no ha sido registrada en el sistema.	Notificar al usuario que el nombre de usuario y/o contraseña no son los correctos.

2.4.6 Interfaz

Para el desarrollo del sistema se creó una interfaz sencilla, agradable y útil para el usuario. La interfaz del sistema cumple con las exigencias de navegación para sistemas informáticos web. A continuación se muestran imágenes para más adelante abordar cada elemento, que debe conocer para hacer uso de la aplicación.

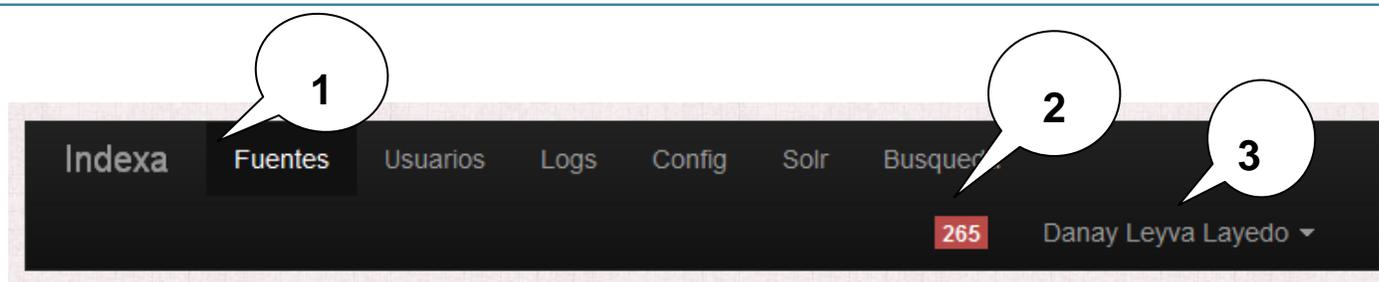


Fig. 12 Interfaz de Indexa. Elaboración propia.



Fig. 13 Interfaz de Indexa. Elaboración propia.

1. Menú de navegación principal.

- a. Fuentes – acceso a las FIS
- b. Usuarios – acceso a los usuarios del sistema.
- c. Logs – acceso a los logs generados por el sistema.

d. Config – acceso de solo lectura a la configuración del sistema.

e. Solr – acceso a la aplicación web de Solr.

2. Notificador de problemas en la indexación.

3. Menú del usuario.

a. Perfil del usuario

b. Salir

4. Área de notificación visual.

5. Banda de navegación rápida para las fuentes de información.

a. Agregar nueva fuente

b. Todas las fuentes

6. Área de contenido principal.

2.5 Diseño de la base de datos

La base de datos tiene un diseño simple, cuenta con 4 tablas en las que se guarda toda la información que poseen las FIS.

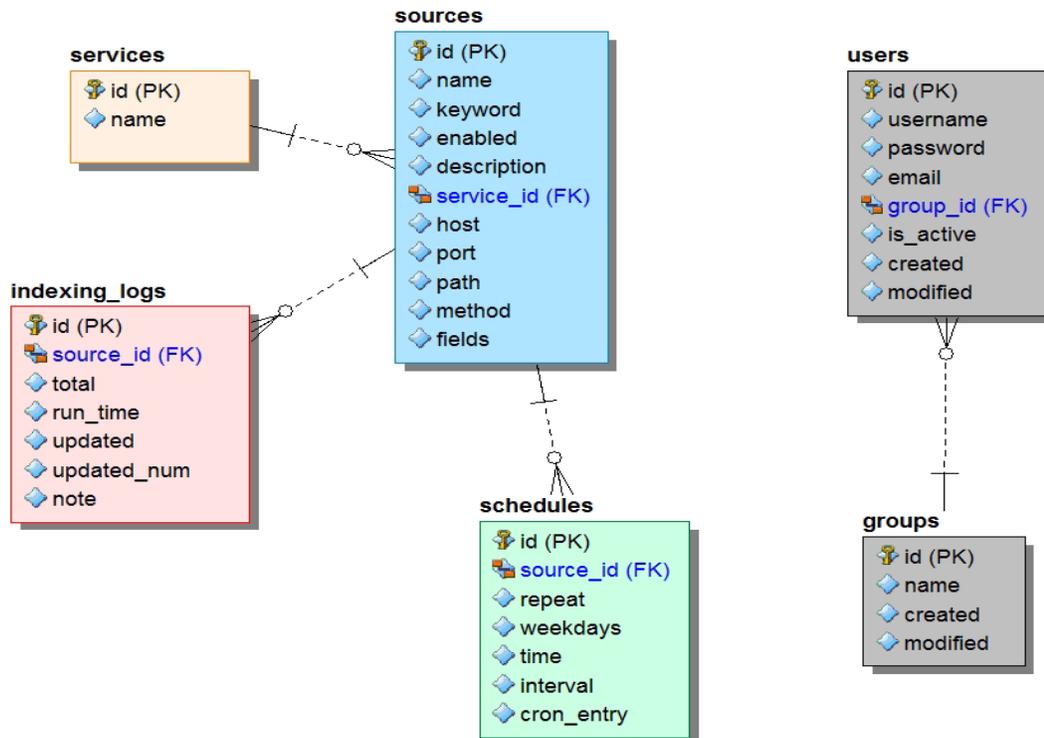


Fig. 14 Modelo entidad -relación. Elaboración propia.

Conclusiones parciales

La metodología seleccionada para la elaboración del motor de búsqueda, permite realizar el análisis y diseño de la aplicación. Se definieron un total de 11 HU que permitieron describir los aspectos principales a tener en cuenta para el desarrollo de la solución. Se precisó la prioridad de cada una de las HU que describen las funcionales del sistema, puntualizando el orden de su implementación y las iteraciones en que serán implementadas. A partir de las HU se construyó el plan de entregas y 6 tarjetas CRC que traducen las HU a funcionalidades a implementar.

Capítulo # 3 Codificación y prueba

Introducción

En el presente capítulo se describen las tareas a implementar en la fase de codificación y lo referente a las pruebas. En este caso al motor de búsqueda se le practicaron dos tipos de pruebas, las unitarias y las de aceptación. Las de aceptación se dividieron en 3 iteraciones y se tuvo en cuenta los escenarios que planteó el cliente.

3.1 Codificación

Codificación es la fase de la metodología XP donde el cliente es parte del equipo de desarrollo. A la hora de codificar una HU su presencia es aún más necesaria debido a que los clientes son los que se encargan de crearlas y negocian los tiempos en los que serán implementadas. Además, la fase propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas. Como parte de la codificación se traducen las HU a tareas que luego son implementadas, como se muestra en las tablas 22, 23, 24,... y 32.

A continuación se muestran las tareas más importantes efectuadas para la implementación del motor de búsqueda:

Tabla 24. Tarea #1

Tarea #1
Número de la HU (No.01): Crear FIS
Nombre de la tarea: Desarrollar la creación de una FIS.
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar la funcionalidad de crear una fuente de información de salud.

Tabla 25. Tarea #2

Tarea #2
Número de la HU (No.02): Recuperar FIS
Nombre de la tarea: Desarrollar la recuperación de una FIS
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo

Descripción: Implementar la funcionalidad de recuperar una fuente de información de salud.
--

Tabla 26. Tarea #3

Tarea #3
Número de la HU (No.03): Actualizar FIS
Nombre de la tarea: Desarrollar la actualización de una FIS
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar la funcionalidad de actualizar una fuente de información de salud.

Tabla 27. Tarea #4

Tarea #4
Número de la HU (No.04): Eliminar FIS
Nombre de la tarea: Desarrollar la eliminación de una FIS
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar la funcionalidad de eliminar una fuente de información de salud.

Tabla 28. Tarea #5

Tarea #5
Número de la HU (No.05): Indexar FIS
Nombre de la tarea: Desarrollar la indexación de las fuentes de información de salud
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar la indexación de las fuentes de información de salud en los índices de Apache Solr

Tabla 29. Tarea #6

Tarea #6
Número de la HU (No.06): Búsqueda de información
Nombre de la tarea: Desarrollar la búsqueda
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar la de búsqueda de videos, documentos enriquecidos o cualquier otro tipo de información que este en las FIS.

Tabla 30. Tarea #7

Tarea #7
Número de la HU (No.07): Crear perfil de usuario
Nombre de la tarea: Desarrollar la creación de un perfil de usuario

Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar las funcionalidad de crear un perfil de usuario.

Tabla 31.Tarea #8

Tarea #8
Número de la HU (No.08): Actualizar perfil de usuario
Nombre de la tarea: Desarrollar la actualización de un perfil de usuario
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar las funcionalidad de actualizar un perfil de usuario.

Tabla 32. Tarea #9

Tarea #9
Número de la HU (No.09): Eliminar perfil de usuario
Nombre de la tarea: Desarrollar la eliminación de un perfil de usuario
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar las funcionalidad de crear un perfil de usuario.

Tabla 33. Tarea #10

Tarea #10
Número de la HU (No.10): Listar perfiles de usuarios
Nombre de la tarea: Listar los perfiles de usuarios
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar las funcionalidad de listar un perfil de usuario.

Tabla 34.Tarea #11

Tarea #11
Número de la HU (No.11): Autenticar usuario
Nombre de la tarea: Desarrollar la autenticación
Tipo de tarea: Desarrollo
Programador responsable: Danay Leyva Mayedo
Descripción: Implementar autenticación de los usuarios al sistema

Durante el desarrollo del motor de búsqueda, se implementó y reutilizó los componentes de *software* que se describen a continuación:

- **Solarium 3:** constituye las librerías de Apache Solr para el lenguaje de programación PHP 5: dicha librería contiene las principales funciones utilizadas en la interacción con las opciones que brinda Apache Solr.
- **Cakephp:** constituyen las librerías utilizadas del *framework Cakephp 2.3.6*.
- **Plugins:** paquete de plugins implementados por terceros que sirvieron de base para la implementación de algunas funcionalidades necesarias para el sistema.

3.2 Prueba

Esta fase es muy importante ya que garantiza el buen funcionamiento de la aplicación. La metodología XP plantea dos pruebas, las unitarias y las de aceptación. A pesar de que el desarrollador es quien debe encargarse de programar y realizar las pruebas unitarias, en este caso fue el cliente quien asumió esta responsabilidad. Las pruebas de aceptación se le realizaron al motor de búsqueda, arrojando resultados satisfactorios. El cliente tuvo un papel importante en el desarrollo de las mismas ya que se encargó de especificar los escenarios para verificar el correcto funcionamiento de las mismas.

3.2.1 Pruebas Unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, revisar el código, que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas deben ser guardados junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo [45].

Las pruebas unitarias fueron llevadas a cabo por el cliente, que se encargaron de verificar el código de forma automática, se aplicaron a todas las funcionalidades del sistema comprobando el correcto funcionamiento de cada una de las mismas y obteniendo resultados satisfactorios.

3.2.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las HU, en cada iteración. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información [45]. A continuación se exponen las pruebas de aceptación utilizadas para probar el sistema divididas en las iteraciones correspondientes a cada historia de usuario:

3.2.2.1 Iteración 1

Tabla 35: Descripción de la Prueba de Aceptación 1 HU Añadir FIS

Caso de Prueba de Aceptación	
Código: HU01_p1	Historia de Usuario (No.01): Añadir FIS
Nombre: Añadir una fuente de información con datos erróneos.	
Descripción: Se desea probar que el sistema no añade la fuente con datos erróneos.	
Condiciones de ejecución: Juego de datos incorrectos o incompletos.	
Entrada/ Pasos de ejecución: Datos incorrectos. El sistema comprueba que los datos estén correctos para añadir la fuente de información de salud.	
Resultado esperado: Mostrar un mensaje de error.	
Evaluación de la prueba: Satisfactoria.	

Tabla 36: Descripción de la Prueba de Aceptación 2 HU Añadir FIS

Caso de Prueba de Aceptación	
Código: HU01_p1	Historia de Usuario (No.01): Añadir FIS
Nombre: Añadir una fuente de información con datos correctos.	
Descripción: Se desea probar que el sistema añade la fuente de información de salud.	
Condiciones de ejecución: Juego de datos correctos.	
Entrada/ Pasos de ejecución: Datos correctos. El sistema comprueba que los datos estén correctos para añadir la fuente de información de salud.	
Resultado esperado: Añade la fuente de información de salud.	
Evaluación de la prueba: Satisfactoria.	

Tabla 37: Descripción de la Prueba de Aceptación 3 HU Recuperar FIS

Caso de Prueba de Aceptación	
Código: HU02_p1.	Historia de Usuario (No.02): Recuperar FIS
Nombre: Recuperar una fuente de información.	
Descripción: Se desea probar que el sistema recupera la fuente de información de salud.	
Condiciones de ejecución: Juego de datos correctos.	
Entrada/ Pasos de ejecución: Datos correctos. El sistema comprueba que los datos estén correctos para recuperar la fuente de información de salud.	
Resultado esperado: Reupera la fuente de información de salud.	
Evaluación de la prueba: Satisfactoria.	

Tabla 38: Descripción de la Prueba de Aceptación 4 HU Actualizar FIS

Caso de Prueba de Aceptación	
Código: HU03_p1	Historia de Usuario (No.03): Actualizar FIS.
Nombre: Actualizar una fuente de información con datos correctos.	
Descripción: Se desea probar que el sistema actualiza la fuente de información de salud.	
Condiciones de ejecución: Juego de datos correctos.	
Entrada/ Pasos de ejecución: Datos correctos. El sistema comprueba los datos para actualizar la fuente de información de salud.	
Resultado esperado: Mostrar un mensaje.	
Evaluación de la prueba: Satisfactoria.	

Tabla 39: Descripción de la Prueba de Aceptación 5 HU Actualizar FIS

Caso de Prueba de Aceptación	
Código: HU03_p1	Historia de Usuario (No.03): Actualizar FIS.
Nombre: Actualizar una fuente de información con datos erróneos.	
Descripción: Se desea probar que el sistema no actualiza la fuente de información de salud.	
Condiciones de ejecución: Juego de datos erróneos.	
Entrada/ Pasos de ejecución: Datos erróneos. El sistema comprueba los datos para actualizar la fuente de información de salud.	
Resultado esperado: Mostrar un mensaje de error.	
Evaluación de la prueba: Satisfactoria.	

Tabla 40: Descripción de la Prueba de Aceptación 6 HU Eliminar FIS

Caso de Prueba de Aceptación	
Código: HU04_p1	Historia de Usuario (No.04): Eliminar FIS

Nombre: Eliminar una fuente de información con confirmación válida.
Descripción: Se desea probar que el sistema elimina la fuente de información de salud.
Condiciones de ejecución: Señalar con clic la FIS a eliminar.
Entrada/ Pasos de ejecución: FIS señalada. El sistema pide confirmación para eliminar la FIS.
Resultado esperado: Mostrar un mensaje
Evaluación de la prueba: Satisfactoria.

Tabla 41: Descripción de la Prueba de Aceptación 7 HU Eliminar FIS

Caso de Prueba de Aceptación	
Código: HU04_p1	Historia de Usuario (No.04): Eliminar FIS
Nombre: Eliminar una fuente de información con confirmación no válida.	
Descripción: Se desea probar que el sistema no elimina la fuente de información de salud.	
Condiciones de ejecución: Señalar con clic la FIS a eliminar.	
Entrada/ Pasos de ejecución: FIS señalada. El sistema pide confirmación para eliminar la FIS.	
Resultado esperado: No se elimina la fuente de información de salud.	
Evaluación de la prueba: Satisfactoria.	

Tabla 42. Descripción de la Prueba de Aceptación 8 HU Indexar FIS

Caso de Prueba de Aceptación	
Código: HU05_p1	Historia de Usuario (No.05): Indexar FIS
Nombre: Indexar una fuente de información de salud.	
Descripción: Se desea probar que el sistema indexa la fuente de información de salud.	
Condiciones de ejecución: Señalar con clic la FIS a indexar.	
Entrada/ Pasos de ejecución: FIS señalada.	
Resultado esperado: El sistema indexa la fuente de información de salud.	
Evaluación de la prueba: Satisfactoria.	

3.2.2.2 Iteración 2

Tabla 43: Descripción de la Prueba de Aceptación 9 HU Búsqueda de información

Caso de Prueba de Aceptación	
Código: HU06_p1	Historia de Usuario (No.06): Búsqueda de información.
Nombre: Búsqueda de información	
Descripción: Se desea probar que el sistema es capaz de devolver la información correspondiente.	

Condiciones de ejecución: Palabra ofrace que decea buscar.
Entrada/ Pasos de ejecución: El sistema busca lo relacionado con el parámetro.
Resultado esperado: Mostrar los resultados de la búsqueda.
Evaluación de la prueba: Satisfactoria.

Tabla 44: Descripción de la Prueba de Aceptación 10 HU Crear perfil de usuario

Caso de Prueba de Aceptación	
Código: HU07_p1	Historia de Usuario (No.07): Crear perfil de usuario.
Nombre: Crear un perfil de usuario con datos incorrectos.	
Descripción: Se desea probar que el sistema no crea el usuario.	
Condiciones de ejecución: Datos incorrectos.	
Entrada/ Pasos de ejecución: El sistema confirma los datos.	
Resultado esperado: No se crea el usuario	
Evaluación de la prueba: Satisfactoria.	

Tabla 45: Descripción de la Prueba de Aceptación 11 HU Crear perfil de usuario

Caso de Prueba de Aceptación	
Código: HU07_p1	Historia de Usuario (No.07): Crear perfil de usuario.
Nombre: Crear un perfil de usuario con datos correctos.	
Descripción: Se desea probar que el sistema crea el usuario.	
Condiciones de ejecución: Juego de datos correctos.	
Entrada/ Pasos de ejecución: El sistema confirma los datos.	
Evaluación de la prueba: Satisfactoria.	

Tabla 46: Descripción de la Prueba de Aceptación 12 HU Actualizar perfil de usuario

Caso de Prueba de Aceptación	
Código: HU08_p1	Historia de Usuario (No.08): Actualizar perfil de usuario.
Nombre: Actualizar un perfil de usuario.	
Descripción: Se desea probar que el sistema actualiza los datos del usuario.	
Condiciones de ejecución: Juego de datos correctos.	
Entrada/ Pasos de ejecución: El sistema confirma los datos.	
Resultado esperado: Se actualiza el usuario.	
Evaluación de la prueba: Satisfactoria.	

Tabla 47: Descripción de la Prueba de Aceptación 13 HU Eliminar perfil de usuario

Caso de Prueba de Aceptación	
Código: HU09_p1	Historia de Usuario (No.09): Eliminar perfil

	de usuario.
Nombre: Eliminar un perfil de usuario con confirmación positiva.	
Descripción: Se desea probar que el sistema elimine el usuario.	
Condiciones de ejecución: Seleccionar usuario.	
Entrada/ Pasos de ejecución: El sistema devuelve una confirmación.	
Resultado esperado: Confirmación positiva: Se elimina el usuario.	
Evaluación de la prueba: Satisfactoria.	

Tabla 48: Descripción de la Prueba de Aceptación 14 HU Eliminar perfil de usuario

Caso de Prueba de Aceptación	
Código: HU09_p1	Historia de Usuario (No.09): Eliminar perfil de usuario.
Nombre: Eliminar un perfil de usuario con confirmación negativa.	
Descripción: Se desea probar que el sistema no elimina el usuario.	
Condiciones de ejecución: Seleccionar usuario.	
Entrada/ Pasos de ejecución: El sistema devuelve una confirmación.	
Resultado esperado: Confirmación negativa: No se elimina el usuario.	
Evaluación de la prueba: Satisfactoria.	

3.2.2.3 Iteración 3

Tabla 49: Descripción de la Prueba de Aceptación 15 HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU11_p1	Historia de Usuario (No.11): Autenticar usuario.
Nombre: Entrada al sistema con datos erróneos.	
Descripción: Se desea probar que el sistema no permita el acceso a un usuario con datos erróneos.	
Condiciones de ejecución: Juego de datos incorrectos o incompletos.	
Entrada/ Pasos de ejecución: Datos incorrectos. El sistema comprueba que los datos estén correctos para dar acceso al usuario.	
Resultado esperado: Mostrar un mensaje de error.	
Evaluación de la prueba: Satisfactoria.	

Tabla 50: Descripción de la Prueba de Aceptación 16 HU Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU5_p1	Historia de Usuario (No.5): Autenticar usuario.
Nombre: Entrada al sistema con datos correctos.	

Descripción: Se desea probar que el sistema permite el acceso a un usuario con datos correctos.
Condiciones de ejecución: Juego de datos correctos.
Entrada/ Pasos de ejecución: Datos correctos. El sistema comprueba que los datos estén correctos para dar acceso al usuario.
Resultado esperado: El usuario accede al sistema.
Evaluación de la prueba: Satisfactoria.

Luego de aplicarse las pruebas de aceptación se detectaron un total de 15 no conformidades, de ellas 9 significativas y 6 no significativas.

Las pruebas aplicadas contribuyeron a mejorar la calidad y las funcionalidades del sistema donde se arrojaron resultados visibles. Para una mejor comprensión de los resultados obtenidos al aplicar las pruebas de aceptación, se muestran en el siguiente gráfico:

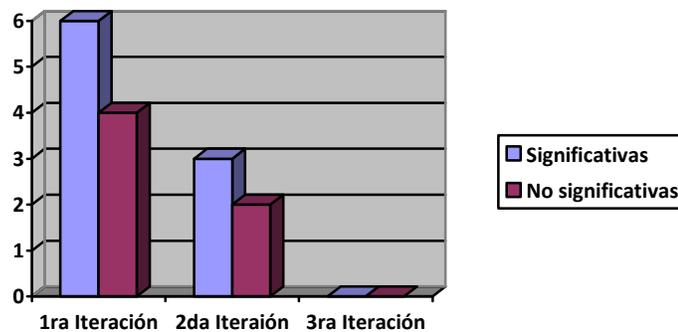


Fig. 15 Gráfico del resultado de las pruebas de aceptación. Elaboración propia.

Una vez concluida cada iteración, se corrigieron cada una de las no conformidades encontradas por lo que luego de la tercera iteración se erradicaron las no conformidades culminándose así las pruebas al motor de búsqueda. Esto permitió entregar al cliente un motor de búsqueda capaz de desarrollar la búsqueda y recuperación de información en la web del CNICM. Tal es así que hoy en día el motor de búsqueda ha sido aplicado a dos de sus sitios web. El resultado alcanzado en las pruebas realizadas al software permitió que el cliente quedara satisfecho con el motor de búsqueda y emitiera una carta de aceptación.

Conclusiones parciales

En el presente capítulo se describieron las tareas de la ingeniería que se utilizaron de base para codificar las HU. Además las pruebas de aceptación que se llevaron a cabo posibilitaron brindar un producto capaz de cumplir con las necesidades del cliente, tal es así que por el resultado positivo de estas pruebas el cliente dio una carta de aceptación. Con la conclusión de este capítulo se considera terminada la propuesta de solución del sistema.

Conclusiones

En el presente trabajo se desarrolló un motor de búsqueda capaz de satisfacer la búsqueda y la recuperación de información en los sitios web de la red del CNICM. Finalmente se arribó a las siguientes conclusiones:

- La sistematización de los SRI permitió escoger a los motores de búsqueda como la solución más acertada para el CNICM. El estudio de los motores de búsqueda existentes permitió encontrar características se te tuvieron en cuenta a la hora de la implementación. El estudio de la arquitectura de los motores de búsqueda sirvió de guía para crear la arquitectura de la propuesta de solución.
- La creación de las HU permitió describir los aspectos necesarios a tener en cuenta en la implementación del motor de búsqueda. La realización de las pruebas permitió brindar un producto de software con la calidad esperada, tal es así que el CNICM emitió una carta de aceptación del producto y aplicó el motor de búsqueda a dos de sus sitios web.
- Sin duda, este proyecto será muy útil si se sigue desarrollando, ya que es una herramienta clave en progreso de la red Infomed, beneficia a más de 70 mil profesionales de la salud y por consecuente al desarrollo científico de Salud Pública.

Recomendaciones

- Se recomienda para futuras versiones que se aumenten las funcionalidades del servicio web de búsqueda, que se incluya un filtrado de la información.

Referencias bibliográficas

[1] Baeza-Yates, Ricardo, Página Web de Ricardo Baeza-Yates, [Disponible en:

<http://www.dcc.uchile.cl/~rbaeza/spanish.html>]

[2] Baeza-Yates, Ricardo, Modern Information Retrieval, 1999

[3] Pinto, María, Búsqueda y Recuperación de Información, 2004 [Disponible en:

http://www.mariapinto.es/e-coms/recu_infor.htm#ri1]

[4] López Herrera, Antonio G., Modelos de Sistemas de Recuperación de Información Lingüística Difusa, 2006

[5] Kiva, Buscadores o Search Engine, 2009 [Disponible en:

<http://www.searchoptimization.es/buscadores-search-engines/buscadores-search-engines.htm>]

[6] Consoft, ¿Qué son los metabuscadores?, 2002 [Disponible en:

http://www.consoft.es/noticias/news_text.asp?id=33219]

[7] Aguirre, Jorge D., Arquitectura de un buscador, 2007 [Disponible en:

http://buscadores.fullblog.com.ar/post/arquitectura_de_un_buscadore_531191953898/

[8] Ricciardi, Agustín, Buscadores Verticales, 2009 [Disponible en:

<http://blog2puntocero.wordpress.com/2009/01/28/buscadores-verticales/>]

[9] Figuerola, Carlos E., Modelos Teóricos de Recuperación de Información, 2008

[10] Colectivo, Historia de Google, 2009 [Disponible en:

http://www.cad.com.mx/historia_de_google.htm]

[11] Comin, Javier, La historia de Yahoo, 2001 [Disponible en:

<http://www.maestrosdelweb.com/editorial/yahoohis/>

[12] **Seta, Leonardo De.** *Apache Solr: una introducción.* 2010.

[13] **Nooijer, Bas de.** Solarium-project. *Solarium-project.* [En línea] 2013.

<http://www.solarium-project.org/>.

[14] Nutch Team, About Nutch, 2009 [Disponible en:

<http://lucene.apache.org/nutch/about.html#Overview>

[15] Nioche, Julien, Features - Nutch Wiki, 2009 [Disponible en:

<http://wiki.apache.org/nutch/Features>

[16] Cookbook. [En línea] 2009. [Citado el: 25 de marzo de 2013.] <http://book.cakephp.org/es/>.

[17] PHP Group, PHP: Hypertext Preprocessor, 2009 [Disponible en: <http://php.net/>]

[18] Rodas Hinostroza, Raul , Características de PHP, 2005 [Disponible en:

<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

[19] Cortez Morales, Roberto. **Introducción Al Análisis de Sistemas Y la Ingeniería de Software. San Jose, Costa Rica: Carlos Fco. Zamora-Murillo, 2006.**

[20] Qualitrain. [En línea] [Citado el: 27 de 03 de 2013.] <http://www.qualitrain.com.mx/Metodologias-Agiles-de-Desarrollo-de-Software-Primera-Parte.html>

[21] Biblioteca virtual. [En línea] [Citado el: 13 de 04 de 2013.]

<http://www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>

[22] Schwaber K., Beedle M., Martin R.C. "Agile Software Development with SCRUM". Prentice Hall. 2001.

- [23] Cockbun, A. "Agile Software Development". Addison-Wesley. 2001.
- [24] Stapleton J. "Dsdm Dynamic Systems Development Method: The Method in Practice". Addison-Wesley. 1997.
- [25] Highsmith J., Orr K. "Adaptive Software Development: A Collaborative Approach to Managing Complex Systems". Dorset House. 2000.
- [26] Coad P., Lefebvre E., De Luca J. "Java Modeling In Color With UML: Enterprise Components and Process". Prentice Hall. 1999.
- [27] Poppendieck M., Poppendieck T. "Lean Software Development: An Agile Toolkit for Software Development Managers". Addison Wesley. 2003.
- [28] **Penadés, Patricio Letelier y M Carmen.** *Métodologías ágiles para el desarrollo de software: Metodología XP.* [En línea] [Citado el: 01 de 04 de 2013.] http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf
- [29] **Castillo Oswaldo, Figueroa Daniel, Sevilla Hector.** Programacion extrema. [En línea] [Citado el: 01 de 04 de 2013.] <http://programacionextrema.tripod.com/fases.htm>
- [30] Highsmith, J. "Agile Software Development Ecosystems". Addison-Wesley. 2002.
- [31] <http://www.slideshare.net/Ing-D-SW-TorresKhano--ME/ventajas-del-por-qu-usar-mysql-como-sistema-gestor-de-bases-de-datos>
- [32] Ecured. Sistema Gestor de Base de Datos. [En línea] [Citado el: 9 de abril de 2013.] [http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos.](http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos)

Glosario de términos

MVC (Modelo-Vista-Controlador): es un estilo de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

PHP: *Hypertext Pre-processor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones *web* dinámicas e interactivas.

MySQL: Sistema de gestión de base de datos relacional orientada a objetos de *software*, publicado bajo la licencia GPL.

FIS: Fuente de información de salud.

HU: Historias de usuario.

XP: Programación extrema. Metodología para el desarrollo ágil.

CRC: Clase, responsabilidad, colaboración.

RF: Requisito funcional.

GPL: Licencia Publica General de GNU o más conocida en ingles por GNU General Public License. Es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

TCP: Protocolo de control de transmisión, es uno de los protocolos fundamentales de internet.

IP: Protocolo de internet, es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red.

HTTP: Protocolo de transferencia de hipertextos. Es el protocolo usado en cada transacción de la World Wide Web (red de informática mundial).

WWW: World Wide Web o red de informática mundial comúnmente conocida como la web, es un sistema de distribución de documentos de hipertextos o hipermedios interconectados y accesibles vía internet.

FTP: Protocolo de transferencia de archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente servidor.

NNTP: Protocolo para la transferencia de noticias en red.

XML: Lenguaje de marcas extensibles, es utilizado para almacenar datos de forma legible y permite definir la gramática de lenguajes específicos.

JSON: Acrónimo de JavaScript Object Notation. Es un formato ligero para el intercambio de datos.

Anexos

Fig. 16 Autenticar usuario.

Fig. 17 Búsqueda de información.

Fuentes

Save Detalles Eliminar

Editar

Descripción

Nombre:

Habilitado:

Descripción: Usa plugin XML-RPC para Wordpress. Desarrollado por por Yazna

Cliente de servicio Web

Servicio:

Host:

Puerto:

Ruta:

Método:

Configuración de escaneado

Clave:

Score:

Solr Fuente

id:

ti:

ur:

ab:

da:

mh:

document:

Para mapear multiples campos, use el caracter '|' como delimitador.

Fig. 18 Añadir FIS.

Todas las fuentes +

Fuentes

Indexar en Solr
Limpiar Solr

- ★ Eventos
- ★ Cumed
- ★ Medicament
- ★ Liscuba
- ★ Dir Institucion
- ★ Al Día
- ★ Legislación
- ★ Obras Audio
- ★ Directorio Ct
- ★ Tramed

Nombre	Eventos	Para indexar
	Descripción	
Habilitado	Sí	Clave: evento
Descripción	Directorio de Eventos en Salud on 23/4/14	Score: 0
Actualizar	Apr 23rd 2014, 11:56	Mapeo de campos
Indexado el	jose luis	Solr Fuente
Indexado por	1536	id: cod_recurso
Cantidad	documentos de 1536 en total	ti: titulo
tiempo de ejecución	19segs	ur: url;
prioridad	0	ab: descripcion
	Cliente de servicio Web	da: fecha_i
		mh: tematica
		document

Editar
Eliminar

Fig. 19 Indexar FIS.

The screenshot displays a web interface for user management. At the top left, there is a button labeled 'Todas las fuentes' with a plus sign. The main heading is 'Usuarios', and below it is the sub-heading 'Editar usuario'. On the right side, there are three buttons: 'Usuarios', 'Enviar', and 'Eliminar'. A vertical sidebar on the left contains a list of categories, each with a star icon: 'Eventos', 'Cumed', 'Medicament', 'Liscuba', 'Dir Instituci', 'Al Día', 'Legislación', 'Obras Audio', 'Directorio Ct', and 'Tramed'. The main form area contains the following fields: 'Nombre de usuario*' with the value 'dmayedo', 'Nombre completo' with the value 'Danay Leyva Layedo', 'Password*', 'Confirm password', and 'Email'.

Fig. 20 Autenticar usuario.