

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4



Módulo recomendador de recursos de información para la intranet del Centro de Tecnologías para la Formación.

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

Autores:

Massiel Agüero Cabreja

Aliosky Martínez Hernández

Tutores:

Ing. Ana Delia González Ricardo

Ing. Yuneikys Recio Miranda

Co-Tutora:

Ing. Liliana Martínez Sanabria

La Habana, junio de 2014

“Año del 56 Aniversario de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 17 días del mes de junio del año 2014.

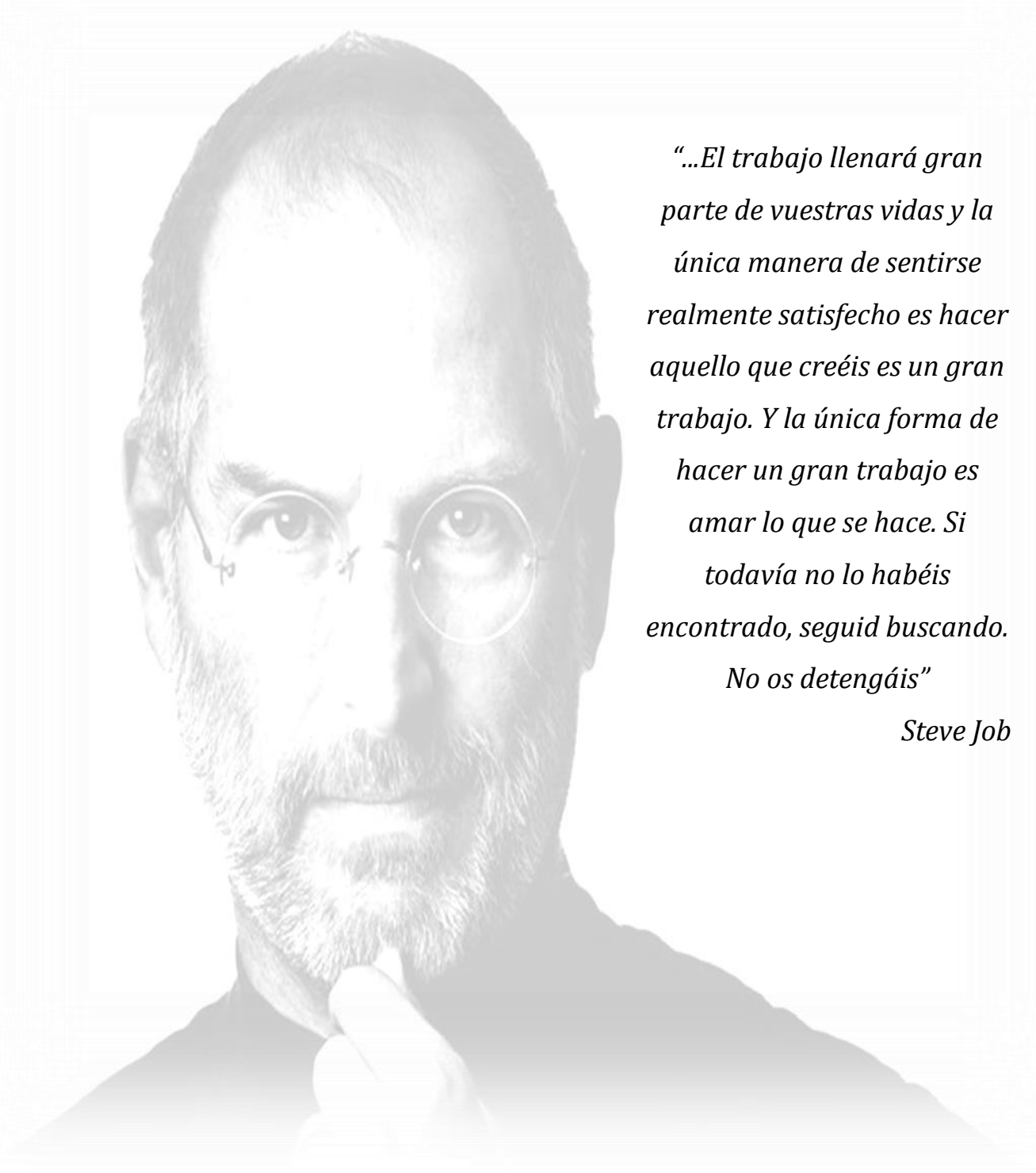
Firma de la tesista
Massiel Agüero Cabreja

Firma del tesista
Aliosky Martínez Hernández

Firma de la tutora
Ing. Ana Delia González Ricardo

Firma de la tutora
Ing. Yuneikys Recio Miranda.

Firma de la cotutora
Ing. Liliana Martínez Sanabria



“...El trabajo llenará gran parte de vuestras vidas y la única manera de sentirse realmente satisfecho es hacer aquello que creéis es un gran trabajo. Y la única forma de hacer un gran trabajo es amar lo que se hace. Si todavía no lo habéis encontrado, seguid buscando. No os detengáis”

Steve Job

Dedicatoria

De Massiel:

*A mis padres **Juan y Lissette** por ser las personas que me motivan a seguir adelante.*

*A mis hermanos **Gabriel, Osbiel y Aciel** que representan un gran ejemplo para mí y una fuente de inspiración.*

*A mi abuelito **Walfredo** y a mi abuelita **Élida** en donde quiera que esté.*

*Al resto de mi **familia** por su apoyo incondicional y por siempre estar pendientes de mí.*

De Aliosky:

*A mi mamá **Laura Elena** por ser mi fuente de inspiración, mi mayor motivación y mi ejemplo a seguir.*

*A mis hermanitas **Lisset y Lisdaimys** esperando que esto les sirva de motivación y nunca dejen de luchar por lo que deseen alcanzar.*

*A mi abuelo **Tomás** por su amor incondicional.*

Agradecimientos

De Massiel Agüero Cabreja:

En primer lugar le doy gracias a Dios y a todos aquellos que me dieron las fuerzas necesarias para llegar a cumplir este gran sueño.

A mis padres Juan y Lisette que con mucho trabajo y esfuerzo y sin importar la necesidad, dieron todo porque tanto yo como mis hermanos llegáramos a ser alguien en la vida. Gracias por enseñarme que con sacrificio y dedicación y enfrentándose a las cosas se pueden cumplir nuestras metas. Espero que se sientan orgullosos por ser los grandes padres que son, por eso y más los quiero mucho y les agradezco de todo corazón-discúlpeme por mis momentos de malcriadez-.

A mis hermanos Aciel, Osbiel y Gabriel porque a pesar de la distancia siguen brindándome mucho apoyo y siempre piensan en mí en los momentos importantes –excepto en la comida...ggg-, espero sigamos así siempre. Gracias a los tres por imponerme esta meta y por motivarme a no quedarme atrás. Muchas gracias a los tres, los quiero mucho, incluso cuando... ¡ya ustedes saben! ggg.

A mis tutoras: Ana Delia y Yuneikys y a mi cotutora Liliana, por su inmensa paciencia, comprensión y dedicación. Gracias por ayudarnos, por guiarnos, por sus críticas, sus aportes y sus buenos consejos durante el desarrollo de este trabajo. Les agradezco de todo corazón por todo su apoyo –disculpen por el tiempo que les robamos–ggg.

A mi abuelito Walfredo y a mi abuelita Éliada (que en paz descanse), por brindarme su amor, por cuidarme y por todos los momentos que pasamos juntos. Muchas gracias, los quiero-pronto espero cocinar mejor ggg-.

A mis tías Lázara, Lidia, Lérida, Martha, Yeniséi, Mari y La negra y a mis tíos pipe, Julito, lázaro, Diosnel, Pedro, Fuácata y El negro por el apoyo que me han brindado y más en estos años de mi carrera. Gracias por tratar de complacerme cuando voy a sus casas, por las comidas ricas y por hacerme sentir muy bien-aunque también hay quién me ha hecho llorar ggg-. Los quiero mucho.

A todos mis primos (as), en especial a mis primas Yadvís y Selky porque siempre me han brindado su amor, su cariño y su amistad, porque han sido mis confidentes y porque son las mejores primas que podía tener y a

Agradecimientos

mis primos Oriel, Orbe Luis, Adalberto, Yasmany... porque siempre han estado presentes y han formado parte en mi vida, además de compartir muchísimas cosas.

A mis buenos vecinos del barrio en especial Arely y Roberto, Elvia, Marisol, Elena y Jelber, Moya y Mari, Mima, Eva, Rosa y Orlando, Esmeralda y Tata, Yasmiany y Pastorcito, Blanca, Liudmila, María y Pacolo, Emma y Enmo... porque casi todos ustedes forman parte de casi toda mi vida, porque siempre han estado pendientes de mí deseando que las cosas me salgan bien y porque siempre me han brindado su apoyo. Nadie puede tener tan buenos vecinos, gracias.

A las personas que han representado una nueva familia para mí Nidia y Pedri quienes desde que los conozco han estado siempre ahí para mí. Gracias por su apoyo por su dedicación, por su amor, los quiero mucho.

A todos mis compañeros de la UCI en especial a Jeiser, Deyrin, Malidia, Yadiel, Lisandra, Osvaldo y Leonardo quienes han estado junto a mí durante mis años en la universidad y hemos compartido grandes cosas. Aunque hay algunos que ya no están aquí, espero que en donde están estén bien.

A mis compañeros de aula con quienes he convivido muchas cosas, ya sean malas o buenas, en especial a Claudia, Yasmin y Yaneisi con quienes compartí además del aula la casa y por ayudarme en los momentos en que las necesité. A mi novio Ivan por estar a mi lado y ayudarme cuando lo necesito, por su amor y su comprensión. Gracias, te quiero.

A todos los profesores con los que compartí durante mis años en la universidad y al compañero Fidel por permitirme alcanzar este sueño.

Quiero agradecer a dos personas muy importantes sin las que no se hubiese podido materializar este sueño, la primera es mi gran amigo, mi compañero de tesis y otro hermano Aliosky con quien compartí grandes aventuras y mis mejores momentos en la Universidad. Gracias por ser una persona en la que pude confiar siempre, por tu dedicación y apoyo, por hacerme reír en los malos momentos y por poner todo tu esfuerzo para hacer de este, nuestro gran sueño. La segunda persona "soy yo" ggg.

Agradecimientos

De Aliosky Martínez Hernández:

Tras un largo camino de alegrías, nuevas experiencias, sacrificios y en ocasiones momentos de tristeza, hoy es el día en que materializo un gran sueño, por ello no debo pasar por alto, un agradecimiento infinito a todas aquellas personas que de una forma u otra colaboraron en mi formación.

*Primeramente quiero agradecerle a **Dios** por darme la fuerza y ayuda para continuar en los momentos difíciles que pensaba que no lo podía lograr.*

*A mi mamá **Laura Elena** un millón de gracias, por ser la mejor de todas, por todo el amor recibido, por ser mi mayor motivación y motor impulsor en este proyecto, este es nuestro triunfo. A mis hermanitas **Lisset** y **Lisdaimys** por su amor, confianza y por ser parte de mi motivación, a ustedes muchísimas gracias.*

*A mi hermana y compañera de tesis **Massiel**, gracias por compartir conmigo los mejores momentos que he vivido durante estos años en la universidad, por tu capacidad de esfuerzo, dedicación y simpatía, gracias además por confiar en mí para acompañarte en la realización de este, nuestro sueño.*

*Gracias de corazón a mis tutoras **Ana Delia**, **Yuneykis** y a mi amiga y cotutora **Liliana**, por ser personas de una paciencia y dedicación inmensa. Gracias por su presencia incondicional, por sus apreciados y relevantes aportes, críticas y comentarios. Les agradezco por su motivación y aliento y por hacer fácil lo difícil, para mí ha sido un privilegio poder contar con su guía y ayuda en este largo camino.*

*A mi abuelo **Tomás** le agradezco por ser un ejemplo y guía a seguir, gracias abuelo, ya soy ingeniero.*

*De igual manera le agradezco a mi padrastro, por la ayuda recibida y por cuidar de mi mamá mientras me superaba, quiero agradecerle a mis tías y tíos, por la confianza y apoyo recibido durante los años de mi carrera, gracias **Barbi**, **Mami**, **Ania**, **Lasi**, **Mary**, **Marsen**, **Yoel** y **Papo**, a mis primos y primas por siempre estar presente y ser parte de mi vida, en especial a **Lisi**, **Mileydis**, **Tito**, **Aliesky**, **Daniel**, **Mayrelis** y los demás.*

Agradecimientos

*No puedo pasar por alto, dos personas que me han apoyado inmensamente durante estos maravillosos años, ellos son mis tíos **Cheo y Lido**, este triunfo también es de ustedes, gracias por toda su ayuda y buenos consejos.*

*A mis amigos que a pesar de estar lejos en esta etapa siempre me han brindado su apoyo incondicional y siempre han estado al tanto de mi progreso, lleguen a **Lisandra, Laura, Diane (La Rubia), Luz, Yandilier, Daniel y al Eduard** mis más sinceros agradecimientos, a **Dayni y Yornue** gracias por compartir conmigo su amistad, los buenos momentos, las buenas series y la buena música jejej.....*

A todos mis otros amigos de la UCI, a los que he robado horas de compañía gracias por hacer que la vida lejos de casa sea mejor, mencionarlos a todos sería extenso y podría cometer algún olvido injusto.

*También debo agradecer a **Alejandro y Gustavo**, por todos los momentos en que me brindaron su ayuda incondicional durante el desarrollo de este trabajo, por supuesto gracias a mis compañeros de apartamento y a mis compañeros(as) del grupo 4503.*

Quiero agradecer además a todos los profesores que marcaron mi paso por la universidad, a todos muchísimas gracias.

Las Tecnologías de la Información y las Comunicaciones (TIC) se han empleado como alternativas para la gestión de la información, siendo utilizadas como soporte para ayudar a las personas a obtener recursos de información que contribuyan en su superación. En el Centro de Tecnologías para la Formación (FORTES) se encuentra en desarrollo la elaboración de una intranet. Esta herramienta va a contar con un gran cúmulo de información, razón por la cual el usuario se puede ver desbordado de tanta información y no saber cuál elegir. Actualmente para ayudar en este proceso, en instituciones y universidades cubanas, se han identificado diferentes herramientas: Sistemas de Recomendación (SR), Diseminación Selectiva de la Información (DSI) y sistemas de Sindicación Realmente Simple (RSS, por sus siglas en inglés). Los autores deciden la implementación de un módulo recomendador híbrido, pues este permite a través de un seguimiento del usuario determinar cuáles son sus intereses y responder a ellos. Los SR son herramientas de software y/o técnicas encargadas de realizar determinadas sugerencias, ayudando a obtener la información que se necesita de acuerdo a las preferencias del usuario. El mismo permite que los trabajadores del Centro FORTES, obtengan los recursos de información publicados en la intranet del centro que puedan ser de su interés, sin necesidad de dedicar mucho tiempo en esta función. Para el desarrollo de la investigación se hizo uso de los métodos científicos: análisis documental, histórico-lógico, análisis sintético y entrevista.

Palabras clave: gestión de la información, módulo recomendador, sistema de recomendación

Introducción	1
Capítulo 1: Fundamentación teórica.....	5
1.1. Generalidades de la gestión de la información	5
1.2. Tecnologías utilizadas en la divulgación de la información de manera personalizada	6
1.2.1. Diseminación Selectiva de Información	7
1.2.2. Sindicación Realmente Simple	8
1.3. Generalidades de los sistemas de recomendación	9
1.3.1. Estructura de un sistema de recomendación	10
1.4. Clasificación de los sistemas de recomendación	11
1.4.1. Sistemas de recomendación basados en contenido.....	11
1.4.2. Sistemas de recomendación basados en el filtrado colaborativo	11
1.4.3. Sistemas de recomendación basados en datos demográficos	13
1.4.4. Sistemas de recomendación híbridos	13
1.5. Técnicas algorítmicas utilizadas en los sistemas de recomendación	13
1.5.1. Técnicas algorítmicas basadas en memoria	14
1.5.2. Técnicas algorítmicas basadas en modelo.....	15
1.6. Aplicación de los sistemas de recomendación	16
1.7. Análisis de soluciones similares existentes.....	17
1.8. Metodología de desarrollo de software	18
1.9. Herramientas y tecnologías para la creación de la propuesta de solución	21
1.9.1. Sistema gestor de contenidos.....	21
1.9.2. Lenguaje de programación.....	23
1.9.3. Sistemas gestores de base de datos.....	25
1.9.4. Entorno Integrado de Desarrollo	26
1.9.5. Herramientas de ingeniería de software asistida por ordenador.....	26
1.9.6. Servidor web.....	27
Conclusiones parciales	28
Capítulo 2: Concepción y características del sistema.	29

2.1.	Propuesta del sistema.....	29
2.2.	Personal relacionado con el sistema.....	33
2.3.	Fase de Planificación.....	33
2.3.1.	Descripción de las Historias de Usuario.....	33
2.3.2.	Estimación de esfuerzos por HU.....	36
2.3.3.	Plan de iteraciones.....	37
2.3.4.	Plan de duración de iteraciones.....	37
2.3.5.	Plan de Entrega.....	38
2.4.	Fase de Diseño.....	38
2.4.1.	Tarjetas CRC.....	39
	Conclusiones parciales.....	41
	Capítulo 3: Implementación y prueba del sistema.....	42
3.1.	Fase de Implementación.....	42
3.1.1.	Iteraciones.....	42
3.2.	Estructura de Drupal.....	44
3.3.	Creación de las tablas adicionales en la base de datos.....	45
3.4.	Módulo desarrollado para la recomendación.....	46
3.5.	Aplicación de patrones de diseño.....	47
3.6.	Estándares de codificación.....	49
3.7.	Instalación de módulos.....	50
3.8.	Fase de Prueba.....	51
3.8.1.	Pruebas unitarias.....	51
3.8.2.	Prueba de aceptación.....	52
	Conclusiones parciales.....	57
	Conclusiones generales.....	58
	Recomendaciones.....	59
	Referencias bibliográficas.....	60
	Bibliografía.....	63

Introducción

En la actualidad las personas representan el activo principal en una institución. Sus ideas constituyen un aporte relevante para alcanzar el éxito en las organizaciones. Además, sus opiniones y acciones apoyan la toma de decisiones y el logro de los objetivos trazados por la entidad (1).

En las organizaciones deben gestionarse procesos que potencien el desarrollo de nuevas habilidades y conocimientos en los trabajadores para lograr un mejor desempeño de los mismos en sus puestos de trabajo. En este sentido, la información juega un papel importante, pues contribuye a que las personas adquieran nuevos conocimientos.

Las Tecnologías de la Información y las Comunicaciones (TIC) se emplean como alternativas para facilitar el acceso a la información. Por esta razón, han sido utilizadas como soporte para ayudar a los trabajadores a obtener los recursos de información que contribuyan en su superación, cultura u obtención de nuevas competencias. Asumiendo los autores de la presente investigación como competencias: características personales que incluyen habilidades y conocimientos, que determinan el éxito en un puesto de trabajo (2).

La Universidad de las Ciencias Informáticas (UCI) tiene implantada un conjunto de tecnologías que sirven de apoyo a los subprocesos de gestión de la información¹: adquisición, transmisión, procesamiento, almacenamiento, diseminación y uso de la información. Entre las herramientas que se pueden mencionar se encuentran: bibliotecas virtuales, intranet y correo electrónico.

La UCI en su estructura cuenta con varias facultades, las cuales incluyen centros de desarrollo de software. Entre ellos se encuentra el Centro de Tecnologías para la Formación (FORTES) perteneciente a la Facultad 4. El mismo posee un área de Desarrollo Humano (DH) que tiene la misión de desarrollar un ambiente de trabajo que influya en el rendimiento y satisfacción de los trabajadores de la organización.

El área de DH sostiene como parte de su misión apoyar los procesos de gestión de la información y el conocimiento, así como contribuir en el clima y cultura organizacional fortaleciendo el uso de las TIC como soporte a los procesos. Para cumplir con los objetivos de esta misión, se comenzó a desarrollar una intranet. El propósito fundamental de esta aplicación informática radica en que la información relevante y

¹ Término que sirve para designar un conjunto de procesos por los que puede atravesar la información durante su ciclo de vida, provenientes por diferentes fuentes que gestionan el acceso sobre la misma (5).

de interés para los trabajadores se disemine, almacene y comparta. Dándole acceso además, a sistemas y servicios existentes en la universidad.

Entre los recursos de información a publicar en la intranet se identifican: convocatorias a eventos, cursos, noticias, diplomados, maestrías, doctorados, listado de revista a publicar, entre otros que pueden contribuir con la adquisición de competencias en los trabajadores. De la misma manera, se tendrá disponible toda la información que debe conocer el trabajador para que obtenga una cultura organizacional (actitudes, valores, deberes y derechos, entre otros).

Como se puede apreciar, la intranet manejará gran cúmulo de información, la cual se encontrará dispersa. El alto flujo de información, unido a la alta frecuencia de publicaciones puede provocar que los usuarios en ocasiones no consulten los contenidos de su preferencia. Navegar por el sistema para encontrar lo que les interesa, aunque la aplicación posea una buena estructura que facilite el acceso a la información, requiere de un tiempo que en ocasiones no tienen. La deficiencia mencionada ocasiona que los trabajadores no conozcan los recursos que puedan ser de su interés o que lo identifiquen de manera tardía.

Por lo anteriormente expresado, se plantea el siguiente **problema a resolver**: ¿Cómo facilitar recursos de información a los usuarios de la intranet del Centro FORTES según sus áreas de interés?

Se tiene como **objeto de estudio** los sistemas de recomendación. Enmarcándose en el **campo de acción** los sistemas de recomendación de recursos de información. Teniendo en cuenta el problema planteado se define el siguiente **objetivo general**: desarrollar un módulo recomendador de recursos de información para los usuarios de la intranet del Centro FORTES.

A partir del objetivo general definido se derivan los siguientes **objetivos específicos**:

- Construir el marco teórico referencial relacionado con el objeto de estudio mediante la consulta, extracción y recopilación de la información relevante sobre el problema a investigar.
- Diseñar un módulo recomendador de recursos de información para los usuarios de la intranet del Centro FORTES.
- Desarrollar la implementación y pruebas al módulo recomendador de recursos de información para los usuarios de la intranet del Centro FORTES.

Se tiene como **idea a defender**: el desarrollo de un módulo recomendador facilitará recursos de

información a los usuarios de la intranet del Centro FORTES según sus áreas de interés.

Para dar cumplimiento a los objetivos específicos se definen las siguientes **tareas de investigación**:

1. Elaboración del estado del arte relacionado con los temas asociados al problema a investigar, fundamentalmente sobre los sistemas de recomendación.
2. Selección del tipo de sistema de recomendación a desarrollar, así como técnicas algorítmicas, la metodología de desarrollo de software, tecnologías y herramientas a utilizar en su diseño e implementación.
3. Identificación de los artefactos a utilizar en el diseño de la propuesta de solución.
4. Elaboración de los artefactos que permitan el diseño de la propuesta de solución.
5. Descripción del proceso de implementación de la propuesta de solución.
6. Selección del tipo de prueba a aplicar para validar el funcionamiento del sistema informático desarrollado.
7. Aplicación del tipo de prueba seleccionada para validar el sistema informático desarrollado.

Desde el punto de vista metodológico fueron empleados durante la investigación los siguientes **métodos científicos**:

Métodos teóricos

Analítico-sintético: se utilizó en el estudio y análisis de la teoría e información, con el objetivo de seleccionar los elementos fundamentales enmarcados en el objeto de estudio y campo de acción.

Histórico-lógico: utilizado para conocer la evolución de los sistemas informáticos que se han utilizado para facilitar el acceso a recursos de información que satisfagan los intereses de las personas.

Análisis documental: se aplicó durante la revisión de la bibliografía consultada para llevar a cabo la propuesta de solución. Permitió el acceso a las siguientes fuentes de información: libros, artículos, revistas y publicaciones de gran utilidad para elaborar la base teórica de la presente investigación.

Métodos empíricos

Entrevista: se aplicó para identificar el problema existente y conocer si existen alternativas que faciliten el

acceso a recursos de información acorde a los intereses de las personas.

El documento posee la siguiente estructura capitular:

Capítulo 1: Fundamentación teórica. Se realiza un análisis de los principales elementos teóricos y conceptos que guiarán el desarrollo de la presente investigación. Se lleva a cabo un estudio de los sistemas de recomendación y de algunos sistemas similares existentes. Además, se describe el análisis de las distintas herramientas, algoritmos, tecnologías y metodología a utilizar en el desarrollo de la propuesta de solución.

Capítulo 2: Concepción y características del sistema. Se realiza una descripción de la propuesta de solución y sus características principales. Se exponen los artefactos generados durante la elaboración del módulo cumpliendo con los establecidos en la metodología seleccionada.

Capítulo 3: Implementación y pruebas. Se describen las tareas a realizar para llevar a cabo la implementación de las principales funcionalidades definidas para el módulo recomendador. Se muestra el análisis realizado sobre los tipos de pruebas que se pueden aplicar para validar el funcionamiento de la propuesta de solución. Se describe el proceso de aplicación del tipo de prueba seleccionada y se muestran los resultados obtenidos.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación teórica

Introducción

En el presente capítulo se desarrolla el marco teórico referencial de la investigación. Se abordan aspectos relacionados con la gestión de la información, indicando los conceptos que regirán el estudio del presente trabajo de diploma. Se describe el análisis realizado sobre los sistemas de recomendación consultando criterios de autores que tratan la temática. Se expone el proceso de selección de las herramientas y tecnologías a utilizar en la implementación de la aplicación, así como la metodología que guiará su proceso de desarrollo.

1.1. Generalidades de la gestión de la información

Para un mejor entendimiento de lo que representa la Gestión de Información (GI) y sus funcionalidades, se necesitan conocer los siguientes conceptos.

Gestión

La gestión es un proceso definido como la acción que hay que llevar a cabo en función de conseguir o resolver determinado elemento (3).

Datos

Los datos son hechos discretos sobre acontecimientos que no suministran explicaciones ni dan razones. Representan la materia prima fundamental para la creación de información (2).

Información

Según Idalberto Chiavenato, información es *"...un conjunto de datos que poseen un significado, de modo tal que reducen la incertidumbre y aumentan el conocimiento de quien se acerca a contemplarlos"* (4).

A partir de los conceptos anteriores, autores como Josep Curto definen la **"gestión de la información"** como un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma (5).

Ponjuán Dante define la gestión de información como: *proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información en y para la sociedad a la que se sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización* (6).

Los autores coinciden con las definiciones anteriores, no obstante, deciden regir la presente investigación a partir de la definición que propone Ponjuán. Los mismos consideran que la GI permite el manejo de recursos de información y su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios. Por otro lado agregan que este proceso implica acciones como: determinar la información que se necesita, su acceso y su divulgación.

Como apoyo a la gestión de la información se han creado varias herramientas. Entre estas se encuentran: intranet, bibliotecas virtuales y el correo electrónico, que permiten la divulgación de la información. Además los repositorios, que almacenan diferentes recursos informativos. También los blog y chat que son usados para compartir información e intercambiar ideas; así como los portales de comunidades en los cuales se tratan temas específicos donde las personas pueden compartir sus conocimientos.

En la actualidad, los niveles de información han aumentado considerablemente. Por esta razón cada vez se hace más difícil gestionar la excesiva cantidad de información a la que las personas se enfrentan diariamente. Este problema suele aumentar cuando se disponen a realizar una búsqueda de la misma.

Decía Cronin que existe la necesidad de comprender cómo el comportamiento de los individuos, en la búsqueda de la información, puede ser mejor apoyado por sistemas para generar, capturar, recuperar y manipular datos e informaciones (7). En correspondencia con lo anterior, se han identificado en instituciones cubanas la implantación de aplicaciones informáticas que apoyan el subproceso de divulgación de la información.

1.2. Tecnologías utilizadas en la divulgación de la información de manera personalizada

Mediante la aplicación del método científico entrevista, los autores identificaron alternativas informáticas vigentes en entidades con el propósito de facilitar el acceso a recursos de información. Las instituciones en las que se realizó dicha entrevista, además de la UCI, fueron el Centro de Desarrollo de Software (DESOFT) y la Universidad de la Habana (UH).

Como resultado se conoció que la información se divulga a través de aplicaciones informáticas como: intranet, bibliotecas virtuales y correo electrónico. Además, se pudo identificar el uso de herramientas tecnológicas que permiten perfilar el envío de la información de manera personalizada. De manera general, los entrevistados en cada contexto ofrecieron datos que ayudaron a saber lo que a continuación se describe.

En el caso del Centro DESOFT se divulga la información a través de una intranet o por correo electrónico. La segunda alternativa provoca inconformidades en los usuarios, pues en ocasiones reciben correos que no se ajustan a sus intereses. Por otra parte, en la intranet, la información se encuentra dispersa y suele tomar mucho tiempo realizar una búsqueda de los contenidos que desea.

En la UH, con el propósito de solventar el problema descrito en el caso anterior, se insertó en la biblioteca virtual un servicio de Diseminación Selectiva de Información (DSI). De esta manera, el usuario puede recibir un correo con el recurso de información deseado adjunto en un mensaje de notificación. Para ello, debe registrar sus intereses informacionales durante el proceso de inscripción.

De la misma manera, en la UCI se tiene implementado un servicio de DSI en la biblioteca virtual. Además, muchas de las aplicaciones informáticas existentes en la universidad (portales, intranet, entre otros) tienen implantado un servicio de Sindicación Realmente Simple (por sus siglas en inglés, RSS). Una vez que el usuario está suscrito en la fuente de contenidos, la herramienta lo mantiene informado automáticamente que se produzca cualquier novedad en el sistema al que está inscrito.

Otra alternativa identificada en la UCI fue el uso de Sistemas de Recomendación (SR) o recomendadores, los cuales le sugieren al usuario información de interés. Por ejemplo, los jurados online de programación incluyen un SR para sugerir al usuario aquellos problemas que a pesar de haber sido resueltos por usuarios semejantes a él, todavía no han sido resueltos por este. Realizando la misma función, pero con el propósito de guiar el aprendizaje interactivo del estudiante se encuentra implantado uno en el módulo Ejercicios de la colección El Navegante.

A partir de lo descrito anteriormente, los autores determinan profundizar en los sistemas informáticos que posibilitan al usuario obtener información de manera personalizada. El objetivo de este estudio es identificar cuál de ellos pueden ayudar al usuario a agilizar su proceso de búsqueda de información en las aplicaciones informáticas existentes, teniendo en cuenta sus áreas de interés.

1.2.1. Diseminación Selectiva de Información

El término Diseminación Selectiva de Información o DSI fue acuñado en 1958 por Hans Luhn, pero no fue hasta 1961 que creó el servicio. Este servicio brinda periódicamente a cada usuario o grupo de usuarios las referencias de los documentos que corresponden a sus intereses cognoscitivos, a través del correo electrónico (8).

El servicio puede proporcionarse a un usuario basándose en su perfil individual. Para lograr su eficacia se requiere de una adecuada definición del perfil del usuario mediante la selección de un conjunto de palabras clave que reflejen la temática en cuestión (9).

Según el estudio realizado por los autores, el servicio suele usarse con fines educativos o de trabajo. En las entidades es empleado para mantener informados y actualizados a los usuarios sobre documentos que entran a la institución o sobre aquellos que se publican en el propio centro y que son de su interés. Además, se pudo conocer que el servicio mayormente es utilizado en bibliotecas y centros de información.

1.2.2. Sindicación Realmente Simple

El RSS, constituye un formato para la sindicación de contenidos de páginas web. Existen opiniones divergentes sobre su origen, pero sus antecedentes más documentados son el Channel Definition Format, creado por Microsoft para el Internet Explorer 4.0 en 1997. El servicio permite que las personas puedan recibir de forma sencilla, directamente en su ordenador o en una página web en línea, información actualizada sobre sus páginas web favoritas, sin tener que visitarlas (10). Los usuarios no tienen necesidad de actualizar la información, pues el proceso se realiza automáticamente.

Para recibir las noticias RSS la página debe tener disponible el servicio. Los documentos que contienen las noticias (generalmente llamados canales) son leídos mediante lectores RSS denominados agregadores. Los agregadores son programas que se instalan en el ordenador, sitios web especializados en la lectura de RSS o se encuentran en los programas utilizados normalmente para recibir información de Internet (navegadores web y clientes de correo electrónico) (11).

El servicio RSS posee una extensión del tipo .rss o .xml, pero en realidad es un archivo de texto donde aparecen referencias a contenidos publicados. Al igual que HTML sirve para escribir páginas en un formato entendible por los navegadores, este sirve para enumerar artículos o páginas dentro de un sitio, en un formato que pueda entender el lector RSS. Actualmente, las últimas versiones de navegadores como: Mozilla Firefox, Safari y Opera, permiten leer los RSS sin necesidad de un software adicional.

1.2.3. Sistemas de recomendación

Los SR surgen como respuesta a la existente sobrecarga de información. En un inicio se conocían como filtros colaborativos y los primeros trabajos datan de los años 90. Uno de los primeros grupos de investigación en su desarrollo fue el proyecto GroupLens perteneciente a la Universidad de Minnesota, el cual ha proporcionado una gran parte de la base algorítmica de estos sistemas. Su uso se está

incrementando cada vez más debido a que son muy útiles para evaluar y filtrar la gran cantidad de información disponible en la web, ayudando a las personas en su búsqueda (12). Son herramientas que generan recomendaciones sobre un determinado objeto a partir de las preferencias y opiniones dadas por los usuarios.

Aunque los sistemas anteriormente descritos han sido creados para proporcionarle al usuario contenidos que puedan ser de su interés, sus mecanismos de funcionamiento son diferentes. El servicio DSI envía al usuario las novedades de acuerdo a su perfil, mientras que en el servicio RSS el usuario se suscribe a un recurso de información y mediante un lector RSS se conecta automáticamente y va extrayendo y mostrándole al usuario las novedades aparecidas.

Por otra parte, los SR son más amplios, utilizan técnicas para identificar aquellos elementos que mejor satisfacen las preferencias del usuario. Tienen el objetivo de aprender más sobre los mismos y tratar cada contenido disponible para anticiparse a sus necesidades, por lo cual pueden ser usados como herramientas de decisión. Para sus recomendaciones al igual que el servicio DSI se pueden apoyar de un perfil del usuario. Además tiene en cuenta las evidencias de navegación del mismo y las opiniones de otros usuarios para realizar su recomendación. Por esta razón, se decide llevar a cabo su implementación para darle solución a la situación problemática identificada.

1.3. Generalidades de los sistemas de recomendación

Para identificar la conceptualización de un sistema de recomendación a asumir en la presente investigación, se determina analizar algunas definiciones abordadas por estudiosos de la temática.

Según Yager *“son herramientas cuyo objetivo es asistir a los usuarios en sus procesos de búsqueda de información, ayudando a filtrar los ítems de información recuperados, usando recomendaciones propuestas sobre esos ítems². Dichas recomendaciones se generan a partir de las opiniones proporcionadas por otros usuarios sobre ciertos ítems, tales como documentos, libros e informes en búsquedas previas o bien a partir de las preferencias del usuario objeto de la recomendación”* (13).

Según Josep Konstan *“los sistemas de recomendación utilizan la opinión de los miembros de una comunidad para ayudar a los individuos a identificar la información o los productos más relevantes o interesantes según sus necesidades actuales”* (14).

² Objeto que el sistema debe recomendar, como pueden ser, canciones, documentales o documentos (13).

Los autores consideran que el primer concepto es bastante amplio por lo cual deciden regir la presente investigación a través del mismo. De la misma manera, deciden agregar que los SR en general, se encargan de realizar recomendaciones al usuario que puedan ser de interés, de forma automática y personalizada. Para entender su funcionamiento, en función de utilizarlo con la finalidad para la cual fueron creados, es necesario profundizar en su estructura.

1.3.1. Estructura de un sistema de recomendación

La estructura de un SR permite que cuando un usuario indique la información que necesita, el sistema, a partir de la retroalimentación de la información dada por el mismo, realice el proceso de recomendación. La misma se compone por los siguientes componentes (15):

Las entradas y salidas del proceso de generación de la recomendación: las entradas representan la información que el sistema ingresa del usuario (puede ser recogida de forma implícita o explícita) y las salidas son las recomendaciones finales.

El método usado para generar las recomendaciones (proceso de recomendación): es la manera en que el proceso de recomendación se va a desenvolver.

El grado de personalización:

- No personalizados: brindan las mismas recomendaciones a todos los usuarios.
- Personalización efímera: las recomendaciones son dadas tomando en cuenta el comportamiento y acciones del usuario en su sesión actual de navegación.
- Personalización persistente: estos sistemas están basados en el perfil de los usuarios, por lo que hacen uso de métodos de filtrado colaborativo, filtrado basado en contenidos o correlaciones entre ítems. Brindan recomendaciones con mayor grado de personalización.

Los datos que utilizan los sistemas de recomendación pueden separarse en tres grupos: ítems, usuarios y transacciones (16).

Ítems: son los elementos que se les recomienda a las personas. Puede tomar diferentes valores: positivo si resulta útil para el usuario o negativo si este considera que no fue una buena decisión. Además, poseen características que pueden indicar cuán necesario resultan para determinado individuo.

Usuarios: son las personas a las cuales se les va a proveer recomendaciones. Estas pueden tener

diversidad de preferencias cuando acceden a un software. Para poder ofrecer las recomendaciones adecuadas, el sistema debe buscar toda la información disponible acerca de dicho usuario, ya sea la que inicialmente se haya solicitado o la que se va recopilando a lo largo de la interacción de la persona con el sistema.

Transacciones: así se le denomina a la interacción de los usuarios con el SR. De esta forma se puede almacenar información acerca de las preferencias de los usuarios sobre los ítems.

1.4. Clasificación de los sistemas de recomendación

Los SR tienen diferentes clasificaciones dependiendo del tipo de información que utilicen para realizar las recomendaciones. Pueden ser basados en: contenido, filtrado colaborativo, datos demográficos e híbridos.

1.4.1. Sistemas de recomendación basados en contenido

Los SR basados en contenido recomiendan los ítems a los usuarios teniendo en cuenta su descripción y el perfil con los intereses del usuario. Este tipo de SR sigue la filosofía “muéstrame más de lo que ya me ha gustado”, su técnica consiste en recomendar a un determinado usuario ítems que poseen similitud con los que a él le han gustado (17).

Para llevar a cabo la recomendación utiliza las preferencias del usuario frente a los nuevos ítems. La creación del perfil del usuario puede realizarse de manera explícita, preguntándole directamente al usuario por sus preferencias. También, el sistema puede recoger la información de manera implícita, obteniendo indirectamente elementos que muestren las preferencias del usuario.

Entre las principales deficiencias de este tipo de recomendación se pueden encontrar que suele recomendar ítems muy parecidos a los que el usuario activo ya conoce. Esta desventaja provoca que el usuario pierda la confianza que va adquiriendo sobre el sistema. Otra insuficiencia es que los usuarios que sean nuevos utilizando el sistema tienen que rankear un número suficiente de ítems para que el sistema pueda realmente entender sus preferencias (18).

El estudio realizado por los autores evidenció que aunque esta técnica de recomendación es muy utilizada, existen otras alternativas más eficientes.

1.4.2. Sistemas de recomendación basados en el filtrado colaborativo

Los sistemas basados en filtrado colaborativo tienen la particularidad de crear sus recomendaciones basadas en personas con similitudes en sus preferencias. Estos sistemas sugieren nuevos ítems o

predicen la utilidad de cierto ítem para un usuario determinado basado en los gustos previos del mismo y en las opiniones de otros usuarios con iguales intereses (19).

En este tipo de recomendación, se les presentan algunos ítems a los usuarios y a partir de estos ellos expresan sus intereses. Luego de obtener la información, el sistema crea un perfil y lo compara con el de los demás construyendo una lista de personas con intereses similares, la cual se puede denominar como “usuarios más cercanos”. Se combina la lista de intereses y se devuelve otra con las recomendaciones, conteniendo los ítems que mejor evaluaron estos usuarios y que todavía el nuevo individuo no ha evaluado (20).

Existen dos tipos de filtros colaborativos, los que son basados en el usuario y basados en el ítem:

Filtrado basado en usuarios

Esta técnica sugiere al usuario que solicita la recomendación aquellos elementos que han causado interés a los usuarios que guarden cierta similitud con él (vecinos). Otra alternativa es realizar recomendaciones basándose más en la calidad en sí de los productos que en las propiedades concretas de los mismos. Este tipo de filtrado es muy útil, pues le brinda directamente una opción al usuario de lo que el sistema evalúa como ítems de posible interés para el mismo, basándose en usuarios con un comportamiento similar para realizar las predicciones (21).

Filtrado colaborativo basado en ítem

Mediante este método, un producto es recomendado a un usuario activo si es similar a los definidos en su perfil personal. En este caso, se considera que dos productos son similares (o vecinos) si los usuarios que han seleccionado uno de ellos tienden a seleccionar el otro, asignándole índices de interés parecidos. Esta técnica da mejores resultados que la variante basada en usuario cuando el número de productos disponibles en el SR es mucho menor que el número de usuarios (19).

Una desventaja que posee este tipo de recomendación es que cuando un usuario llega al sistema no es posible hacerle recomendaciones hasta que su perfil sea lo suficientemente completo para encontrarle a su grupo de vecinos cercanos (22). Además, si los gustos del usuario son poco comunes, encontrarle un conjunto de vecinos cercanos será una tarea complicada.

Según el estudio realizado por los autores se identificó la presencia de mayor cantidad de SR basados en el filtrado colaborativo que SR basados en contenido. Además se pudo conocer que los mismos hacen

uso de técnicas algorítmicas para realizar sus recomendaciones.

1.4.3. Sistemas de recomendación basados en datos demográficos

Los sistemas basados en datos demográficos se basan en el perfil demográfico de la persona. Estos clasifican a los usuarios por grupos y recomiendan de acuerdo al grupo al que este pertenezca. Varios sitios web basados en esta información adoptan personalizaciones simples y efectivas (23).

Estos pueden ser utilizados en aplicaciones que tengan en cuenta la edad, país, nivel escolar u otras características para ofrecer determinado ítem. Sin embargo, no tiene en cuenta las preferencias específicas del usuario para realizar la recomendación.

1.4.4. Sistemas de recomendación híbridos

Como se puede observar los sistemas de recomendación expuestos anteriormente poseen algunas limitantes. Se pueden mencionar: la sobreespecialización en el sistema basado en contenido y en el caso del colaborativo la cantidad de usuarios que deben estar registrados en el sistema para que este funcione correctamente. Para mitigar dichas desventajas surgen los sistemas híbridos, los cuales combinan los tipos anteriores, explotando las características de ambos (22).

Para crear un sistema híbrido colaborativo basado en contenido, la construcción de preferencias se basa tanto en los intereses del usuario, como en los intereses de los usuarios similares. El autor Burke³ define un SR híbrido de la siguiente manera “...combinan dos o más técnicas de recomendación para obtener mayor eficacia sin los inconvenientes de cada sistema hibridado” (24).

Los sistemas híbridos, según la bibliografía consultada, ofrecen recomendaciones mucho más eficientes que los demás SR. Por lo anteriormente expuesto se decide implementar un módulo recomendador guiado por los SR híbrido, pues combina el basado en contenido con el filtrado colaborativo, mitigando así las desventajas que poseen los mismos. Además, permite que se tengan en cuenta las acciones previas del usuario, basándose en la descripción del objeto y las preferencias del usuario, característica que se tiene en cuenta en el filtrado colaborativo y el de contenido.

1.5. Técnicas algorítmicas utilizadas en los sistemas de recomendación

Las técnicas algorítmicas pueden clasificarse en dos grandes grupos, las basadas en memorias y las basadas en modelo.

³ Profesor de facultad de informática en la Universidad DePaul en Chicago (24).

1.5.1. Técnicas algorítmicas basadas en memoria

Las técnicas algorítmicas basadas en memoria utilizan la base de datos completa de usuarios-ítems para generar las predicciones. Un SR usa una técnica estadística para buscar un conjunto de usuarios que tengan una historia de concordancia con el usuario objetivo (vecinos), una vez es formada una vecindad se utiliza un algoritmo que combine preferencias de los vecinos para producir una predicción y un *ranking* de los “N⁴ principales” para el usuario objeto (21).

La definición anterior se refleja en la siguiente relación: sea R una matriz de relación usuario-ítems la cual contiene la evaluación (*ratings*) de cada ítem por el usuario. Se toman dos usuarios u_i y u_j que pertenecen a R y se aplican algunos de los siguientes métodos:

Algoritmos de vecinos más cercanos o *k-nearest neighbor*:

Este fue uno de los primeros algoritmos de filtrado colaborativo en implementarse. Se basa fundamentalmente en tres pasos: la representación de los datos, formación de vecinos y generación de recomendaciones (25).

El paso más crítico es la formación de vecinos en el cual se ve inmerso el proceso de encontrar la relación de similitud que existe entre dos usuarios. Para la formación de relación de similitud entre todos los usuarios y el usuario activo se definen algunos métodos:

Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson es una métrica típica de similitud entre funciones de preferencias de usuarios o (menos frecuentemente) distancias de vectores o productos puntos. La fórmula da una aproximación de qué tan bien los vectores comparados (perfiles, ítems) coinciden en la escala desde cero [0] (no similares) a uno [1] (total coincidencia) o menos uno [-1] (total diferencia) (19).

$$corr_{i,k} = \frac{\sum_{j=1}^l (r_{ij} - \bar{r}_i)(r_{kj} - \bar{r}_k)}{\sqrt{\sum_{j=1}^l (r_{ij} - \bar{r}_i)^2 \sum_{j=1}^l (r_{kj} - \bar{r}_k)^2}}$$

Figura 1: Cálculo del coeficiente de correlación de Pearson (25).

Donde:

i : Usuario activo.

⁴ Define los usuarios que mayor similitud poseen con el usuario objeto.

k : Usuario al que se le desea calcular la similitud.

j : Ítem.

\bar{r} : Media aritmética del índice de preferencia del ítem.

$r_{i,j}$: Índice de preferencia del usuario i con el ítem j .

$r_{k,j}$: Índice de preferencia del usuario k con el ítem.

Similitud basada en coseno

Se considera cada elemento como un vector dentro de un espacio vectorial de m^5 dimensiones y se calcula la similitud como el coseno del ángulo que forman. Es decir, si se tienen dos vectores x_1 y x_2 consistentes en un arreglo, cuyos elementos son las votaciones recibidas de cada usuario su similitud estará dada por (15):

$$sim(x_1, x_2) = \frac{\sum_{u \in U} (R_{u,x_1} - \bar{R}_{x_1})(R_{u,x_2} - \bar{R}_{x_2})}{\sqrt{\sum_{u \in U} (R_{u,x_1} - \bar{R}_{x_1})^2} \sqrt{\sum_{u \in U} (R_{u,x_2} - \bar{R}_{x_2})^2}}$$

Figura 2: Cálculo de la similitud basada en coseno (20).

Donde:

\bar{R} : Media aritmética del nivel de interacción de los contenidos.

x_i : Elementos a los que se le calcula la similitud.

u : Usuario que evalúa los elementos.

U : Conjunto de usuarios.

1.5.2. Técnicas algorítmicas basadas en modelo

Estos algoritmos primero modelan los votos de los usuarios. Tratan el problema como una dificultad de predicción estadística y calculan el valor esperado para cada ítem en función de los votos anteriores. Para ello se utilizan distintos algoritmos de aprendizaje: Clustering, Redes Neuronales como las Redes de Funciones de Base Radial (RBFN), Horting, Encasillamiento y Redes Bayesianas. En general, ante las consultas responden más rápido que los basados en memoria, pero necesitan de un proceso de

⁵ Cantidad de dimensiones

aprendizaje intensivo (26).

Horting

Es una técnica basada en grafos en la cual los nodos son los usuarios y las aristas entre nodos son indicadores de los grados de similitud entre dos usuarios. Las predicciones se producen al recorrer el grafo entre nodos cercanos y combinando las opiniones entre usuarios cercanos. Esta técnica difiere de los algoritmos de vecindad más cercana, en la forma como el grafo puede ser recorrido por otros usuarios que no han valorado los ítems. Luego, esta técnica explora las relaciones transitivas que los algoritmos de vecindad más cercana no tienen en cuenta (17).

Predictores “Slope-One”

Es un algoritmo para recomendadores con filtrado colaborativo basado en ítems. Básicamente este toma dos ítems y los compara con relación a la cantidad de usuarios que los han evaluado para buscar una similitud entre estos. También, predice cuando un usuario puede preferir un ítem dadas las valoraciones del propio usuario hacia otro ítem. Este algoritmo tiene en cuenta tanto los datos de los usuarios que tienen en común un elemento determinado como la información de los demás elementos que ya han sido evaluados (27).

Los autores deciden emplear la técnica algorítmica basada en memoria: vecinos más cercanos tomando como base el coeficiente de correlación de Pearson, pues es la más empleada y recomendada en trabajos análogos para medir la similitud entre varios usuarios. La misma permite elaborar un perfil algorítmico que se basa en la filosofía de recomendar información al usuario apoyándose en las características de usuarios semejantes a él.

Los SR son muy usados en diferentes sitios web para brindar ayuda a sus usuarios sobre los recursos de información que les podrían interesar y se encuentren en el mismo. Por esta razón, se utilizan en varias áreas como el comercio y la educación.

1.6. Aplicación de los sistemas de recomendación

Los sistemas de recomendación pueden ser aplicados en diferentes áreas de la vida cotidiana. En la aplicación de estos sistemas hay que destacar su amplia y estratégica utilización en el marco del comercio y la educación.

La aplicación de estos sistemas en el comercio está evidenciada a través de la página web Amazon, la

cual es muy popular en Internet. En esta se pueden buscar diferentes materiales como películas, libros y productos electrónicos. Para todas sus áreas la aplicación inicialmente propone una lista de productos de oferta con los más vendidos, lo más novedoso, entre otros. A partir de lo que la persona seleccione de esa lista, el sistema utiliza su SR para guiar al usuario hacia el producto que este busca. Además, Amazon permite que los usuarios se registren llenando un formulario con sus preferencias y que emitan una votación sobre los productos que visita, haciendo que el proceso sea mucho mejor. Otros ejemplos del empleo de los SR se pueden encontrar en: Amie Street, Genius y Pandora, páginas creadas con el objetivo de brindar servicios de música.

En el ámbito educativo se identifican universidades que tienen implementado este sistema en algunas de sus aplicaciones web. En la Universidad Simón Bolívar de Venezuela, se realizó un SR con el objetivo de ofrecer a los estudiantes aspirantes a ingresar a instituciones universitarias algunas opciones de carreras en sintonía con sus preferencias, competencias y recursos.

Para llevar a cabo la realización de un nuevo SR es importante realizar un estudio de otros sistemas de este tipo que se hayan construido anteriormente. De esta manera se pueden identificar similitudes entre los mismos que puedan ayudar a la elaboración del sistema que se desea.

1.7. Análisis de soluciones similares existentes

Para ayudar en la elaboración del SR se realiza un estudio de soluciones similares existentes, para determinar elementos que presentan dichas herramientas que pueden ser útiles en el desarrollo de la solución propuesta. Los sistemas analizados son los siguientes:

- Un SR para jurados *online* de programación que utiliza la recomendación basada en el filtrado colaborativo. Su técnica consiste en sugerir al usuario aquellos problemas que a pesar de haber sido resueltos por usuarios semejantes a él, todavía no han sido resueltos por este.
- Un sistema de recomendación de contenido que tiene la misión de recomendar noticias a los usuarios del Portal Octavitos⁶. Este sistema permite a los usuarios de dicho portal satisfacer intereses informativos específicos, proporcionando un grupo de noticias. Esta tarea se realiza a partir de un perfil de usuario donde el mismo selecciona las temáticas que desea que se le

⁶ Portal Octavitos: Portal empleado para la publicación de información relacionada con: noticias, eventos a desarrollar en la universidad, entre otras informaciones.

recomiende y también a través de las evidencias de navegación dejadas en el portal.

- Un SR de ejercicios físicos que emplea la recomendación híbrida y tiene la tarea de determinar los ejercicios que debe realizar un estudiante. Para que el mismo lleve a cabo la recomendación el usuario debe crear un perfil con sus características y especificar sus preferencias acerca de los ejercicios físicos que desea realizar.
- La red social Facebook realiza recomendaciones de grupos a las personas en función del perfil de las mismas y teniendo en cuenta el perfil del grupo. El perfil del grupo se obtiene de los perfiles de las personas que son miembros de dicho grupo (28). Además utiliza las recomendaciones para sugerir amigos empleando para ellos datos de su perfil como lugar de procedencia, donde estudió y otros aspectos.

El estudio de estos sistemas fue provechoso, pues se identificaron características y funcionalidades comunes (creación del perfil de usuario, técnicas algorítmicas, tipo de sistema) que sirven como base para el desarrollo del módulo recomendador propuesto. Además se obtuvo una visión de cómo manejar algunos de los procesos que se encuentran inmersos dentro de la estructura de los SR, facilitando el trabajo a la hora de tomar ideas para la construcción del mismo.

Para llevar a cabo el desarrollo de cualquier software se requiere utilizar una metodología que guíe su proceso de elaboración.

1.8. Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y documentación que ayuda a los desarrolladores a realizar un nuevo producto (29). Las metodologías de desarrollo se clasifican en dos grupos: ágiles o ligeras y robustas o tradicionales. A continuación se realiza una comparación de las mismas.

Tabla 1: Comparación entre la metodología ágil y la tradicional (30).

Metodología Ágil	Metodología Tradicional
Pocos artefactos	Más artefactos
Pocos roles, más genéricos y flexibles.	Más roles, más específico.
El cliente es parte del equipo de desarrollo (además, en su sitio).	El cliente interactúa con el equipo de desarrollo mediante reuniones.

Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< diez [10] integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.

A partir de la comparación realizada los autores determinan utilizar una metodología ágil para guiar el proceso de desarrollo de la propuesta de solución. Se define este tipo de metodología porque el cliente forma parte del equipo de desarrollo, el equipo es pequeño y es un proyecto de corta duración; además los requisitos pueden ser cambiantes. A solicitud del cliente se decide aplicar Extreme Programming (XP).

Metodología Extreme Programming

La metodología XP fue creada por Kent Beck en el año 1996. Tiene sus antecedentes desde mediados de los años 80 cuando Kent Beck y Ward Cunningham probaron formas de desarrollar software diferente a las que se habían utilizado hasta el momento, trabajando en un grupo de investigación de Tektronix. Posteriormente, en los años 90, Beck comenzó un proyecto en DaimlerChrysler, conocido como C3 (Chrysler Comprehensive Compensation) que dio paso al surgimiento de esta metodología (31).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Es adecuada para proyectos con equipos de desarrollo pequeños y con requisitos imprecisos y muy cambiantes (32).

La metodología XP fomenta los siguientes valores (30):

- **Comunicación:** la comunicación con el cliente es muy importante, por ende forma parte del equipo de trabajo. De esta manera, cualquier duda sobre los requisitos del sistema puede ser evacuada de manera inmediata.

- **Simplicidad:** consiste en desarrollar solo el sistema que realmente se necesita resolviendo en cada momento las necesidades actuales.
- **Retroalimentación:** brindar una retroalimentación correcta y precisa hace que se pueda mantener una buena comunicación y conocer el estado en que se encuentra el proyecto actualmente.
- **Coraje:** implica poder realizar cambios cuando algo no funciona bien del todo, diseñar e implementar solo lo necesario para el presente, así como pedir ayuda o reducir el alcance de una entrega si el tiempo no alcanza.

Toda metodología, para su desarrollo de software, propone varios roles. Donde cada uno acarrea consigo ciertas responsabilidades. A continuación se definen los roles que plantea XP para su desarrollo (30).

Programador (Programmer): es el responsable de tomar las decisiones técnicas y de construir el sistema. En XP, los programadores diseñan, programan y realizan las pruebas.

Jefe de Proyecto (Manager): es el encargado de organizar y guiar las reuniones. Además, asegura condiciones adecuadas para el proyecto.

Cliente (Customer): es parte del equipo y determina qué construir y cuándo. Además es el encargado de establecer las pruebas funcionales.

Encargado de Pruebas (Tester): ayuda al cliente con las pruebas funcionales y se asegura de que las pruebas funcionales se superan.

Rastreador (Tracker): observa sin molestar y conserva datos históricos.

Entrenador (Coach): es quien advierte si ocurre una desviación en el proceso. Además, guía al equipo de desarrollo para que se siga el proceso XP correctamente.

Las fases de desarrollo de XP consisten en (33):

Fase I: Planificación. En esta fase el cliente establece la prioridad de cada Historia de Usuario (HU) y los programadores realizan una estimación del esfuerzo necesario para llevar a cabo la implementación de cada una.

Fase II: Diseño. Sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacer todo lo menos complicado posible para conseguir un diseño fácil, entendible e implementable que en el futuro

costará menos tiempo y esfuerzo desarrollar.

Fase III: Implementación. Esta fase implica el desarrollo de las HU, la cual debe realizarse con la presencia del cliente. La codificación debe hacerse atendiendo a estándares de codificación ya creados, lo cual mantiene el código consistente y facilita su comprensión y escalabilidad.

Fase IV: Prueba. En esta fase se realizan pruebas para comprobar el funcionamiento del sistema implementando.

Luego de seleccionada la metodología se deben escoger las herramientas y tecnologías a utilizar en el desarrollo, cumpliendo lo establecido en cada fase. Las mismas permitirán generar los artefactos que se indican en la metodología.

1.9. Herramientas y tecnologías para la creación de la propuesta de solución

El vigor y dinamismo de las aplicaciones web está determinada fundamentalmente por el uso de tecnologías utilizadas en su implementación. Para la elaboración de este sistema se hace uso de las siguientes herramientas y tecnologías, seleccionadas por el cliente, quien especifica las que a continuación se describen:

1.9.1. Sistema gestor de contenidos

La evolución de las herramientas para la creación de páginas web ha permitido que se construyan sitios web con mayor funcionalidad y presentación visual entre las que se encuentran los Sistemas de Gestión de Contenidos, del inglés Content Management System (CMS).

Un CMS, es una aplicación web que permite la creación de software desde un navegador de Internet y sin necesidad de tener grandes conocimientos informáticos. Según Fran Gil Rodríguez un CMS: *“...es un software que permite crear una estructura base para la creación y administración de contenidos, principalmente de páginas web. Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan los contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web”* (34).

Los CMS brindan ventajas tales como (35):

- **Inclusión de nuevas funcionalidades en la web:** mediante la inclusión de módulos realizados por terceros pueden solucionarse necesidades concretas, por lo que el sistema puede crecer y adaptarse a las necesidades requeridas.
- **Reutilización de objetos o componentes:** un CMS permite la recuperación y reutilización de páginas, documentos y de cualquier otro objeto publicado o almacenado.
- **Cambios del aspecto de la web:** si no hay una correcta separación entre el contenido y la presentación, un cambio en el diseño puede conllevar a la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo de estándar CSS (Cascading Style Sheets u hojas de estilo en cascada).
- **Control de acceso:** permite gestionar los diferentes permisos a cada área de un sitio web, aplicados a grupos o individuos.

Dentro de los CMS que existen se encuentran Joomla, Wordpress y Drupal. Por indicaciones del cliente se decide utilizar Drupal en su versión 7.23 para el desarrollo del sistema.

Sistema gestor de contenido Drupal

El CMS Drupal, es una solución *Open-Source*⁷ distribuida bajo la licencia General Public License (GPL⁸), escrito en Preprocessed Hipertext Pages (PHP) y desarrollado y mantenido por una activa comunidad de usuarios. Reconocido por la calidad de su código y de las páginas generadas, así como el énfasis especial en la usabilidad y consistencia de todo el sistema (36).

Entre las características que destacan a este CMS se identifican las siguientes (37):

- Ofrece la posibilidad de gestionar las taxonomías y la estructuración de contenidos de forma personalizable, algo indispensable para sitios de complejidad media-alta.
- Las funcionalidades que brinda el CMS se pueden ampliar instalándoles nuevos módulos que implementen dichas funcionalidades.

⁷ Así se conoce al software distribuido y desarrollado libremente.

⁸ Licencia orientada para proteger la libre modificación, distribución y uso de un software.

- El contenido creado en Drupal es funcionalmente un objeto (nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios, o no, sobre cada objeto.
- Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una “capa de abstracción de base de datos” que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.
- Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.
- Drupal puede mostrar en las páginas web de administración, informes sobre popularidad del contenido, o de cómo los usuarios navegan por el sitio.

El CMS Drupal requiere de un conjunto de herramientas y lenguajes para llevar a cabo la implementación de una aplicación informática.

1.9.2. Lenguaje de programación

Según la bibliografía consultada *“un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existente”* (38).

En una arquitectura cliente-servidor, los lenguajes de programación pueden dividirse en dos grupos: lenguajes del lado del cliente y del lado del servidor.

1.9.2.1. Lenguajes de programación del lado del cliente

Un lenguaje del lado cliente basa su procesamiento en el cliente web; se ejecuta en el navegador del usuario (38). Existen varios lenguajes de programación del lado del cliente, como son: VBScript, JavaScript, HTML, XHTML, DHTML, CSS. Sin embargo, en la presente investigación se trabajará con

XHTML por las características que posee y por recomendación del cliente.

Lenguaje XHTML

El lenguaje eXtensible Hypertext Markup Language (XHTML) es la versión eXtensible Markup Language (XML⁹) de HyperText Markup Language (HTML¹⁰). Se utiliza en la creación de páginas web, que al ser un extensible facilita el desarrollo y la integración de nuevos elementos (39).

El lenguaje XHTML incorpora una nueva concepción o filosofía de modelación de las páginas web que busca la creación de una web semántica. Suprime todas las etiquetas y atributos que sirven para definir el aspecto y solo se dejan las etiquetas que sirven para definir el significado de cada elemento de la página, dejando para JavaScript su aspecto y diseño en distintos medios tales como: ordenadores, teléfonos móviles, impresoras, entre otros (40). Para el desarrollo de la propuesta de solución se determina utilizar XHTML en su versión 5.

1.9.2.2. Lenguajes de programación del lado del servidor

Los lenguajes del lado del servidor son reconocidos, ejecutados e interpretados por el propio servidor (41). Dentro de estos lenguajes se encuentran PHP, Perl, Microsoft ActiveServer Pages (ASP), Java Server Pages (JSP) y ColdFusion. En la presente investigación se analiza el lenguaje PHP, pues es el empleado en Drupal. El mismo será utilizado en su versión 5.4.

Preprocessed Hypertext Pages

El lenguaje de código abierto Preprocessed Hipertext Pages (PHP) es adecuado para el desarrollo web y que puede ser incrustado en HTML. Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de sistemas de bases de datos como MySQL y PostgreSQL (42).

Consume pocos recursos y es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Otras ventajas que aporta el uso de PHP es su sencilla implementación y puede adaptarse fácilmente a un servidor como Apache. No obstante, posee la desventaja de requerir un alto nivel de experiencia para la realización de proyectos complejos (43).

Para desarrollar aplicaciones web en PHP se hace uso de un sistema que permita conservar información de forma permanente para recuperarla posteriormente. Para ello, son utilizados los sistemas gestores de

⁹ Es un lenguaje similar a HTML pero su función principal es describir datos y no mostrarlos como HTML.

¹⁰ Lenguaje para la elaboración de páginas web.

base de datos.

1.9.3. Sistemas gestores de base de datos

Un Sistema Gestor de Base de Datos (SGBD) permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado (44).

Los sistemas gestores de base de datos presentan características tales como:

- Permite crear y gestionar base de datos de forma fácil, cómoda y rápida.
- Ofrece una gran flexibilidad para el trabajo con base de datos relacionales.
- Ofrece un ambiente agradable dado por su interfaz gráfica.

Para la construcción de aplicaciones de gestión se destacan por su eficiencia gestores como: Oracle, MySQL, SQL Server y PostgreSQL. Para la elaboración de la propuesta de solución, por solicitud del cliente, debido a las características que presenta, se hará uso del SGBD MySQL en su versión 5.5.34.

Sistema gestor de base de datos MySQL

Según la bibliografía consultada, MySQL es un SGBD muy conocido y ampliamente usado por su simplicidad y notable rendimiento. El SGBD MySQL es multihilo (permite dividir un programa en dos o más tareas que corren simultáneamente), es multiusuario y se caracteriza por tener licenciamiento dual. Por un lado se ofrece bajo la licencia GNU GPL para cualquier uso compatible y por otro, las empresas que quieran incorporarlo en productos privativos pueden comprar una licencia específica que les permita este uso (45).

Entre las ventajas que suelen destacarse de este sistema, se podrían citar las siguientes (46):

- **Facilidad de uso:** es un sistema de base de datos de alto rendimiento pero relativamente simple y menos complejo de configurar y administrar.
- **Capacidad:** pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias bases de datos al mismo tiempo.

- **Conectividad y seguridad:** está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar de Internet.

Existen herramientas orientadas a la implementación de aplicaciones dinámicas que otorgan mayor potencialidad e interactividad con el usuario como es el caso del Entorno Integrado de Desarrollo.

1.9.4. Entorno Integrado de Desarrollo

Un Entorno Integrado de Desarrollo (del inglés, Integrated Development Environment IDE) es una herramienta que permite a los desarrolladores de software escribir sus programas en uno o más lenguajes. Consiste básicamente en una plataforma en la que se integran un editor de código, un compilador, un depurador y una interfaz gráfica de usuario (47). Entre los IDE que existen se encuentran: Netbeans, Eclipse, Zend Studio, entre otros.

Los autores deciden utilizar el IDE NetBeans 7.4, pues permite crear aplicaciones web con PHP 5. Además, ofrece formato de código acorde a los estándares de código de Drupal y reconocimiento a los principales tipos de archivo específicos del CMS (.info, .module .install, entre otros).

Entorno de desarrollo integrado Netbeans

La herramienta está pensada para escribir, compilar, depurar y ejecutar programas. Aunque está escrita en Java permite programar en cualquier lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso. La nueva versión 7.4 se destaca por ampliar el soporte avanzado de desarrollo HTML, a la vez que ofrece un nuevo soporte para el desarrollo móvil y continuas mejoras en PHP y otros lenguajes de programación (48).

1.9.5. Herramientas de ingeniería de software asistida por ordenador

Las Herramientas de Ingeniería de Software Asistida por Ordenador (CASE, por sus siglas en inglés) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un producto (49). Para la realización de este sistema se utiliza la herramienta Visual Paradigm for UML en su versión 8.0.

Lenguaje unificado de modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender,

diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo y en todas las etapas del ciclo de vida (50).

Visual Paradigm for UML

Visual Paradigm for UML es una herramienta CASE que utiliza como lenguaje de modelado UML y que soporta el ciclo de vida completo del desarrollo de software. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra con varios IDE como Eclipse, .Net, NetBeans, entre otros (51). En el presente trabajo se hace útil su uso para representar el modo de funcionamiento del sistema a implementar.

Para atender y responder a las diferentes peticiones del navegador y asegurar que se muestren y representen todos los elementos necesarios, se utilizan los servidores web.

1.9.6. Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Se encarga de responder a estas peticiones de forma adecuada, entregando como resultado una página web o información de acuerdo a los comandos solicitados (52). Entre los servidores web que existen, se encuentran Internet Information Server (IIS) y Apache, utilizándose este último en su versión 2.2.22 para el desarrollo de la propuesta de solución por solicitud del cliente.

Servidor Apache

El servidor Apache es una tecnología gratuita y de código abierto. Posee una licencia descendiente de las licencias BSD (Distribución de Software Berkeley), la cual permite actualizar, modificar y reutilizar el código fuente (53).

Es un servidor web que se distingue por las siguientes características (54):

- Puede ser ejecutado en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Altamente configurable de diseño modular. Es muy sencillo ampliar sus capacidades, actualmente existen muchos módulos para Apache que son adaptables, con facilidades de instalación según se necesiten.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

El servidor Apache en su versión 2.2.22 permitirá instalar fácilmente PHP, MySQL y otros componentes necesarios para el desarrollo del módulo recomendador.

Conclusiones parciales

El estudio realizado apoyado en los métodos de la investigación científica, permitió elaborar el marco teórico conceptual que soporta la investigación, siendo así definidas las características generales de los SR híbridos, las cuales se tendrán en cuenta para el desarrollo de la propuesta de solución. La solución será desarrollada utilizando la metodología Extreme Programming para guiar el proceso de desarrollo. Las tecnologías seleccionadas para llevar a cabo la implementación de la propuesta de solución fueron: PHP como lenguaje de programación del lado del servidor, XHTML para la elaboración de páginas web. Se selecciona como servidor web Apache y sistema gestor de contenido Drupal. Además, como IDE se elige NetBeans y como sistema gestor de base de datos MySQL.

Capítulo 2: Concepción y características del sistema

Capítulo 2: Concepción y características del sistema.

Introducción

El presente capítulo tiene el objetivo de describir el proceso de construcción de la solución y puntualizar las principales funciones que tendrá el módulo recomendador a través de funcionalidades y características del sistema siguiendo la metodología de desarrollo XP. Se muestran las historias de usuarios que fueron escritas por el cliente, el plan de iteraciones y otros artefactos generados por la metodología durante el desarrollo de la solución.

2.1. Propuesta del sistema

Para solucionar el problema identificado se determina elaborar un módulo recomendador que facilite el acceso a los recursos de información disponibles en la intranet del Centro FORTES que puedan ser de interés de los usuarios. Las preferencias del usuario se obtendrán tanto de forma explícita como implícita.

El módulo cuenta con dos posibles estados para realizar la recomendación. El primero, es cuando el usuario no ha llenado su perfil, mientras que el segundo ocurre en caso contrario. El perfil estará compuesto por los diferentes roles que puede desempeñar una persona. Podrán hacer uso del módulo recomendador todos los usuarios que accedan al sitio, siempre y cuando se hayan autenticado previamente.

Cuando el usuario se encuentra en el primer caso, para realizar las recomendaciones se deben tener en cuenta las evidencias dejadas en su navegación por el sitio, lo cual permitirá determinar sus preferencias y poder recomendarle a partir de sus gustos. En este caso se recomendarían aquellos ítems similares a los que el usuario ya conoce. Se realizarán hasta un máximo de diez recomendaciones y el peso de las mismas estará dado teniendo como criterio de selección aquellos ítems que reciben mayor cantidad de visitas por parte de otros usuarios registrados en el sistema.

Si no existen recomendaciones para realizarle al usuario (esto sucede cuando ya ha valorado todas las recomendaciones realizadas o cuando es nuevo en el sistema) se procede a recomendar los cinco ítems que mayor cantidad de visitas tienen por parte de otros usuarios. Se decide recomendar estos ítems teniendo como premisa que al ser contenidos tan relevantes para otras personas, también podrían ser de interés para el usuario en cuestión y un comodín para comenzar la navegación. En caso que el usuario ya haya visitado estos ítems no se realiza la recomendación, pues no se cuenta con otro criterio para efectuar la misma.

Cuando el usuario se encuentra en el segundo caso, lo primero que se realiza es determinar el vecindario. Para ello se emplea la fórmula del coeficiente de Correlación de Pearson, la cual da una medida de qué tan similares son dos usuarios con respecto a la percepción que estos tienen de los ítems. Para establecer el tamaño del vecindario se seleccionan solo aquellos usuarios que superan cierto umbral definido, para la elección del mismo se tomó en cuenta la tabla de clasificación de correlaciones que se utiliza en la Correlación de Pearson. Determinándose como vecinos, aquellos usuarios cuyo umbral está entre 0.4 y 1.

Tabla 2: clasificación de correlaciones (55).

Umbral	Significado
0	Correlación Nula
0,01 a 0,19	Correlación positiva muy baja
0,2 a 0,39	Correlación positiva baja
0,4 a 0,69	Correlación positiva moderada
0,7 a 0,89	Correlación positiva alta
0,9 a 0,99	Correlación positiva muy alta
1	Correlación positiva perfecta

En la fórmula siguiente la correlación ($corr_{ik}$) que se asigna al usuario (k) para predecir al usuario actual (i) viene dado por: (r_{ij}) que corresponde a las visitas del usuario (i) al elemento (j).

$$corr_{i,k} = \frac{\sum_{j=1}^l (r_{ij} - \bar{r}_i)(r_{kj} - \bar{r}_k)}{\sqrt{\sum_{j=1}^l (r_{ij} - \bar{r}_i)^2 \sum_{j=1}^l (r_{kj} - \bar{r}_k)^2}}$$

Donde:

i : Usuario activo.

k : Usuario al que se le desea calcular la similitud.

j : Ítem.

\bar{r} : Media aritmética del índice de preferencia del ítem.

$r_{i,j}$: Índice de preferencia del usuario i con el ítem j .

$r_{k,j}$: Índice de preferencia del usuario k con el ítem.

Una vez que se obtienen los usuarios que podrían formar parte del vecindario se procede a formar el

mismo. El paso que sigue, es determinar las posibles recomendaciones, para ello se tienen en cuenta los ítems visitados por los usuarios vecinos que no han sido visitados por el usuario al que se le va a efectuar la recomendación. De estas posibles recomendaciones se valoran las que podrían ser de mayor interés para el usuario. La cantidad de visitas que tiene un ítem y la cantidad de vecinos que lo hayan visitado, van a definir su peso o qué tan recomendable es para el usuario autenticado.

Para determinar el peso de la recomendación se emplea la siguiente fórmula (ver Figura 3):

$$P_{a,i} = \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) * corr_{i,k}}{\sum_{u=1}^n corr_{i,k}}$$

Figura 3: Fórmula para calcular el peso de las recomendaciones.

Donde:

$P_{a,i}$: Peso de las recomendaciones.

$r_{u,i}$: Visitas de un vecino u al mismo ítem i .

\bar{r}_u : Promedio de visitas del vecino u .

$corr_{i,k}$: Peso asignado al usuario actual.

En el caso que no se pudiera determinar un vecindario aplicando el coeficiente de Correlación de Pearson (esto sucede si no hay usuarios que tienen ítems en común con el usuario al que se desea brindar la recomendación), el mismo estaría compuesto por aquellos usuarios que tienen algún rol en común con el usuario activo. El peso del vecindario estaría dado por la cantidad de roles en común con el usuario autenticado, pues mientras más similitud haya entre el usuario actual y sus vecinos, más exacta puede ser la recomendación. El peso de la recomendación en este caso estaría dado por la cantidad de visitas que tenga el ítem por los usuarios del vecindario.

El módulo recomendador propuesto es un híbrido que combina las recomendaciones basadas en contenido con las de filtrado colaborativo, por lo tanto se debe contar con los dos criterios para realizar la recomendación al usuario. Por esta razón, se seleccionan las recomendaciones que se podrían realizar teniendo en cuenta las evidencias de navegación del usuario y las recomendaciones de acuerdo a su perfil. Luego de tener estas recomendaciones se seleccionan las diez a recomendar.

Para realizar esta acción, primero se van a recomendar aquellos ítems que coinciden en ambos casos. Posteriormente, se recomiendan los ítems elegidos de acuerdo a su perfil y que no hayan coincidido con los seleccionados a recomendar por la navegación. Finalmente, completarían estas recomendaciones los ítems listos a recomendar a partir de la navegación del usuario en el sitio. Siguiendo el orden descrito anteriormente, se plasman las recomendaciones en un bloque del sitio donde esté vigente el módulo recomendador, siempre teniendo presente que se harían un máximo de diez recomendaciones, lo cual fue definido por el cliente.

Para un mejor entendimiento del funcionamiento del módulo recomendador, se muestra el siguiente diagrama (ver Figura 4):

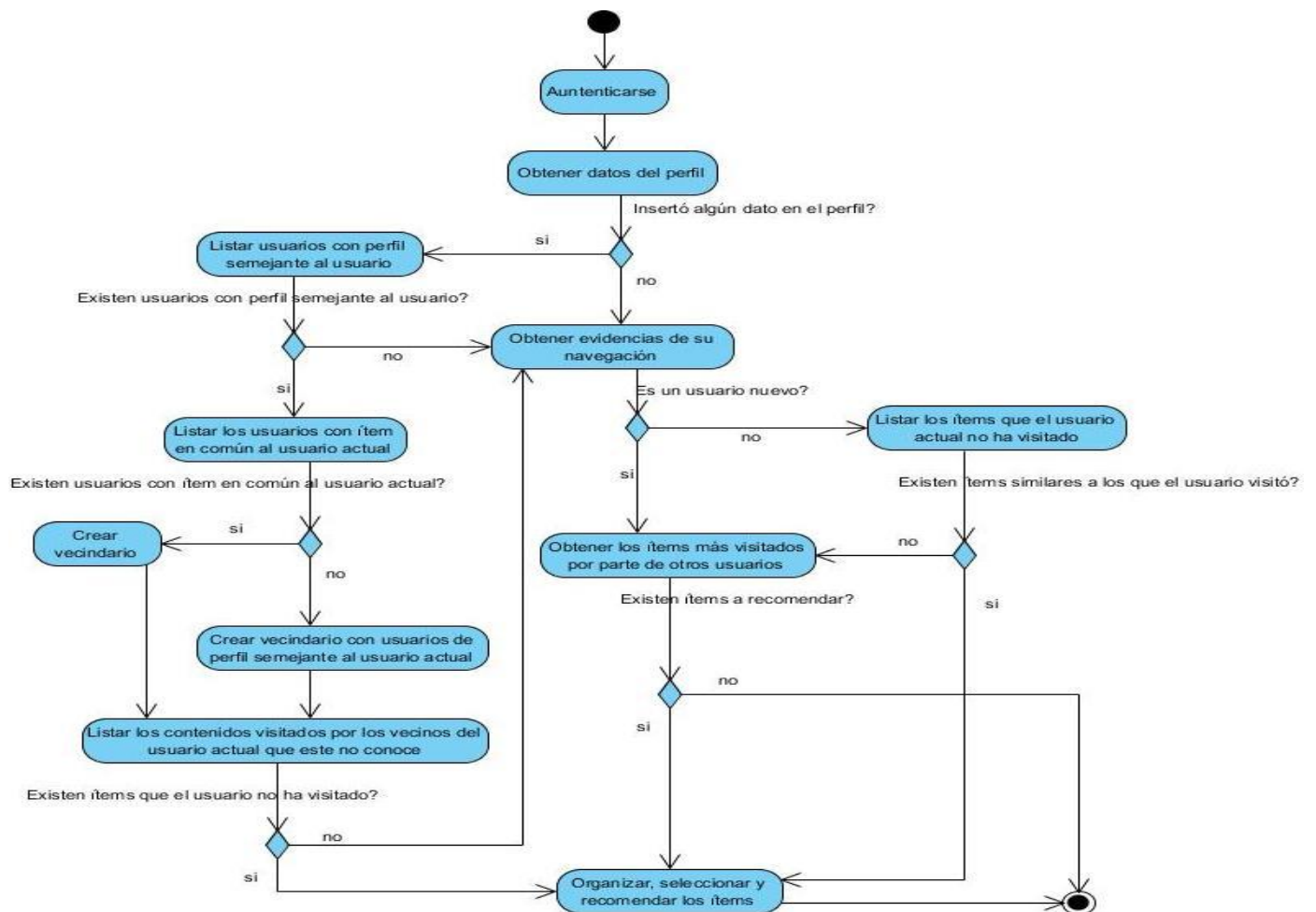


Figura 4: Funcionamiento del módulo recomendador.

2.2. Personal relacionado con el sistema

Todo personal que intervenga en la administración de la aplicación o haga uso de esta, se considera personal relacionado con la aplicación.

Tabla 3. Personal relacionado con el sistema.

Personal relacionado con el sistema	Descripción
Usuario autenticado	El usuario debe estar autenticado para poder obtener el servicio de recomendación. Para poder autenticarse depende del sitio web donde se utilice si debe registrarse primeramente o no.
Administrador	Persona encargada de administrar las acciones a realizar por el módulo recomendador y los datos de los usuarios que se registran en la aplicación donde sea empleado el mismo.

Para hacer un correcto uso de la metodología XP es necesario transitar por las fases que propone y generar los artefactos correspondientes.

2.3. Fase de Planificación

En la fase de Planificación se describen detalladamente las funcionalidades del sistema empleando Historias de Usuarios (HU). Además, se establecen las prioridades para implementarlas y se estima el tiempo de realización de cada una. La Planificación es una fase que se realiza en pocos días, en ella el cliente establece la prioridad de cada HU y los programadores se encargan de realizar una estimación del esfuerzo necesario de cada una de ellas utilizando como medida el punto. El resultado de esta fase es un Plan de Entrega (33).

2.3.1. Descripción de las Historias de Usuario

Las HU son unos de los artefactos más importantes que genera la metodología de desarrollo ágil XP. Son escritas por el cliente, tal y como ven ellos las necesidades del sistema. Incluyen descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Se elabora una por cada funcionalidad del módulo. El tiempo de desarrollo ideal para una HU es entre una y tres semanas y permiten responder rápidamente a los requerimientos cambiantes (56).

Los aspectos que recoge cada HU según Kent Beck son (57):

- **Número:** posee el número asignado a la HU.

- **Nombre de HU:** atributo que contiene el nombre de la HU.
- **Usuario:** el usuario del sistema que utiliza o protagoniza la HU.
- **Modificación:** indica si la HU fue modificada.
- **Iteración asignada:** número de la iteración donde va a desarrollarse la HU.
- **Prioridad en negocio:** evidencia el nivel de prioridad de la HU en el negocio.
- **Riesgo en desarrollo:** evidencia el nivel de riesgo en caso de no realizarse la HU.
- **Puntos estimados:** es el tiempo estimado en semanas que se demorará el desarrollo de la HU.
- **Descripción:** posee una breve descripción de lo que realizará la HU.
- **Observaciones:** señalamiento o advertencia del sistema.

A continuación se muestran las HU asociadas a las funcionalidades del módulo.

Tabla 4. HU Elaboración del perfil de usuario.

HU		
Número: 1	Nombre de la HU: Elaboración del perfil del usuario	
Usuario: usuario registrado	Iteración asignada: 1	Puntos estimados: 2.0
Prioridad en negocio: alta	Riesgo en desarrollo: alta	Modificación: si
Descripción: brinda la posibilidad a todos los usuarios de llenar un formulario con el rol o los roles que le interesan. Los datos que inserten serán guardados para ser utilizados en la recomendación que el módulo les va a brindar a cada uno. Si un usuario no desea insertar ningún dato después de estar en el mismo, puede cancelar la acción.		
Observaciones: el usuario debe estar autenticado.		

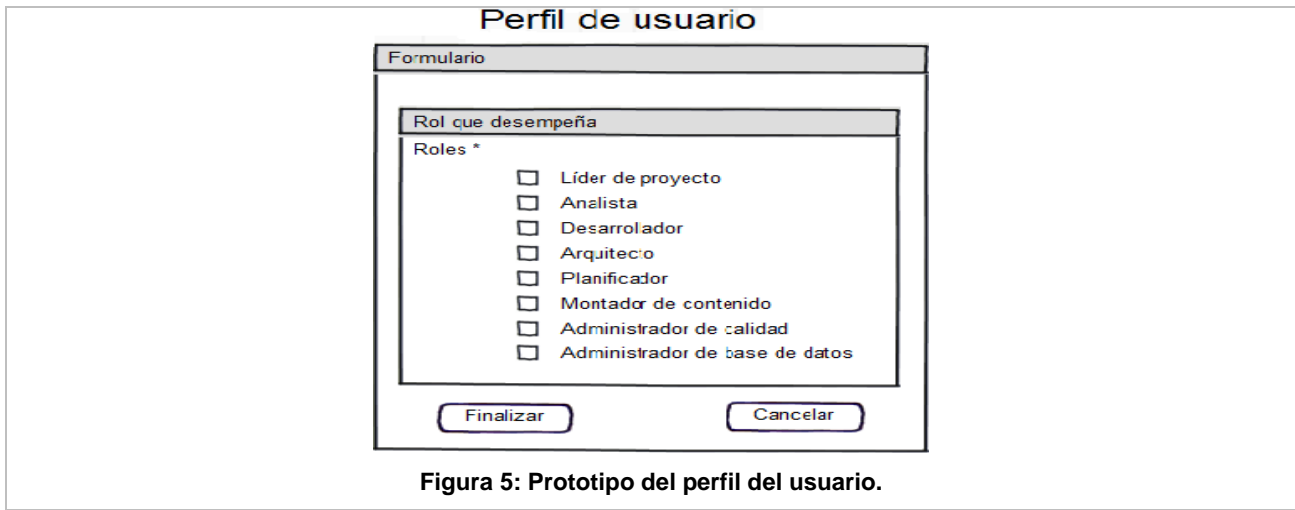


Tabla 5: HU Generar recomendaciones según la navegación del usuario.

HU		
Número: 2	Nombre de la HU: Generar recomendaciones según la navegación del usuario.	
Usuario: Administrador	Iteración asignada: 1	Puntos estimados: 2.5
Prioridad en negocio: alta	Riesgo en desarrollo: alta	
Descripción: permite recomendar a todos los usuarios del sistema que estén autenticados los contenidos que le puedan interesar teniendo en cuenta los lugares visitados dentro del sitio.		
Observaciones: el usuario debe estar autenticado.		

Tabla 6: HU Generar recomendación según el perfil del usuario.

HU		
Número: 3	Nombre de la HU: Generar recomendaciones según el perfil de usuario.	
Usuario: Administrador	Iteración asignada: 2	Puntos estimados: 3
Prioridad en negocio: alta	Riesgo en desarrollo: alta	
Descripción: permite recomendar información a los usuarios teniendo en cuenta su perfil. Para realizar la recomendación se deben tener en cuenta los usuarios que tengan relación con el usuario autenticado. Recomienda los contenidos que prefieren estos usuarios. Se debe tener en cuenta la cantidad de roles comunes entre el usuario actual y los demás usuarios.		
Observaciones: el usuario debe estar autenticado y haber creado su perfil.		

Tabla 7: HU Mostrar listado de recomendaciones.

HU		
Número: 4	Nombre de la HU: Generar recomendaciones de acuerdo al perfil y a la navegación del usuario	
Usuario: Administrador	Iteración asignada: 2	Puntos estimados: 3
Prioridad en Negocio: alta	Riesgo en Desarrollo: alta	
Descripción: para realizar esta acción se deben tener en cuenta la relación que existe entre los usuarios a partir de su perfil y los lugares que han visitado en común dentro del sitio. Se deben vincular los dos criterios para realizar la recomendación.		
Observaciones: el usuario debe estar autenticado y haber creado su perfil.		

Tabla 8: HU Mostrar listado de recomendaciones.

HU		
Número: 5	Nombre de la HU: Mostrar listado de recomendaciones	
Usuario: usuario registrado	Iteración asignada: 3	Puntos estimados: 1.5
Prioridad en Negocio: media	Riesgo en Desarrollo: alta	
Descripción: muestra al usuario un listado con la información recomendada. Para mostrar el listado con las recomendaciones se apoya de las HU 2, 3 y 4. En dependencia del resultado que se obtenga en la implementación de las HU, será la recomendación que se brinde. Solo se efectuarán hasta un máximo de diez recomendaciones.		
Observaciones: el usuario debe estar autenticado.		

Tabla 9: HU Mostrar listado de contenidos más visitados.

HU		
Número: 5	Nombre de la HU: Mostrar listado de contenidos más visitados.	
Usuario: usuario registrado	Iteración asignada: 3	Puntos estimados: 1.0
Prioridad en Negocio: media	Riesgo en Desarrollo: media	
Descripción: muestra al usuario actual un listado con los cinco contenidos más visitados por parte de otros usuarios, que el mismo no ha visitado. En caso que ya los haya visitado no se le recomiendan.		
Observaciones: el usuario debe estar autenticado.		

2.3.2. Estimación de esfuerzos por HU

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de uno a tres [1-3] puntos.

Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a HU que fueron

terminadas en la última iteración (56). La estimación de esfuerzo para el desarrollo del módulo se puede observar en la siguiente tabla.

Tabla 10: Estimación de esfuerzos.

No	HU	Puntos estimados
1	Elaboración perfil de usuario	2.0
2	Generar recomendación según la navegación del usuario	2.5
3	Generar recomendación según el perfil del usuario	3.0
4	Generar recomendaciones de acuerdo al perfil y a la navegación del usuario	3.0
5	Mostrar listado de recomendaciones	1.5
6	Mostrar listado de contenidos más visitados	1.0
Total	13 Semanas	

2.3.3. Plan de iteraciones

En la metodología XP, la creación del sistema se divide en iteraciones. Para cada iteración el cliente define las HU que serán implementadas. Al terminar cada iteración, la aplicación tendrá implementadas las funcionalidades asociadas a cada HU que se realiza en la iteración.

Iteración 1: En la primera iteración se entregarán dos de las funcionalidades que tienen prioridad alta para el cliente correspondiendo a las HU 1 y 2: Elaboración del perfil de usuario y Generar recomendación según la navegación del usuario.

Iteración 2: en la segunda iteración se realizarán las restantes HU que tiene prioridad alta para el cliente correspondiendo a las HU 3 y 4: Generar recomendación según el perfil del usuario y Generar recomendaciones de acuerdo al perfil y a la navegación del usuario.

Iteración 3: en esta iteración se implementan las HU de menor prioridad para el cliente (aunque no dejan de ser importantes) correspondientes a las HU 5 y 6: Mostrar listado de recomendaciones y Mostrar listado de contenidos más visitados.

2.3.4. 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 HU. Se tendrá en cuenta además la prioridad que el cliente le asigna a cada HU y el

nivel de complejidad que estas poseen.

Tabla 11: Plan de duración de iteraciones.

Iteraciones	Orden de las HU a implementar	Duración
Iteración 1	Elaboración perfil de usuario.	4 semanas y 3 días
	Generar recomendación según la navegación del usuario.	
Iteración 2	Generar recomendación según el perfil del usuario.	6 semanas
	Generar recomendaciones de acuerdo al perfil y a la navegación del usuario	
Iteración 3	Mostrar listado de recomendaciones	2 semanas y 3 días
	Mostrar listado de contenidos más visitados	

2.3.5. Plan de Entrega

En el Plan de Entrega se realiza un cronograma de entregas donde el cliente establece las HU que serán agrupadas para conformar una entrega. A continuación se muestra el Plan de Entrega para este trabajo.

Tabla 12: Plan de entrega.

HU	1 ^{ra} iteración	2 ^{da} iteración	3 ^{ra} iteración
Elaboración perfil de usuario.	V 1.0	Finalizado	Finalizado
Generar recomendación según la navegación del usuario.	V 1.0	Finalizado	Finalizado
Generar recomendación según el perfil del usuario.	--	V 1.0	Finalizado
Generar recomendaciones de acuerdo al perfil y a la navegación del usuario	--	V 1.0	Finalizado
Mostrar listado de recomendaciones.	--	--	V 1.0
Mostrar listado de contenidos más visitados	--	--	V 1.0

2.4. Fase de Diseño

La metodología XP propone que el diseño de la aplicación ha de ser lo más simple posible, siempre que cumpla con las funcionalidades especificadas por el cliente. Además, recomienda que se utilicen tarjetas CRC (Contenido, Responsabilidad y Colaboración) para la representación de las clases.

2.4.1. Tarjetas CRC

La metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Clase, Responsabilidad y Colaboración). Cada tarjeta CRC representa una clase, objeto, módulo, paquete, entre otros, con su nombre en la parte superior. En la parte inferior izquierda se describen las responsabilidades y a la derecha las clases que le sirven de soporte (58).

La filosofía del CMS Drupal no es basada en la Programación Orientada a Objeto (POO), sin embargo, utiliza en su diseño varios conceptos o técnicas del paradigma orientado a objeto (“objetos”, “abstracción”, “polimorfismo”, “encapsulación” y “herencia”). Con su diseño adapta la técnica de las tarjetas CRC a estas características, representando cada tarjeta CRC como una HU, las responsabilidades definidas como las funciones que realiza y los colaboradores serían los métodos que utiliza cada funcionalidad (34). A continuación se muestran las tarjetas CRC de la propuesta de solución:

Tabla 13: Tarjeta CRC Elaboración perfil de usuario.

Tarjeta CRC: Elaboración perfil de usuario	
Responsabilidades	Colaboraciones
Crear un perfil del usuario autenticado, donde se guarde los roles que le interesan.	recomendador_form recomendador_roles_list recomendador_form_submit

Tabla 14: Tarjeta CRC Generar recomendación según la navegación del usuario.

Tarjeta CRC: Generar recomendación según la navegación del usuario	
Responsabilidades	Colaboraciones
Recomendar al usuario hasta un máximo de diez contenidos que él no haya visitado, teniendo en cuenta su navegación en el sitio.	recomendador_listado_terminos recomendador_nodos_terminos recomendador_nodos_visitados recomendador_items_contenido

Tabla 15: Tarjeta CRC Generar recomendación según el perfil del usuario.

Tarjeta CRC: Generar recomendación según el perfil del usuario	
Responsabilidades	Colaboraciones
Recomendar al usuario hasta un máximo de diez contenidos que él no haya visitado, teniendo en cuenta su perfil.	recomendador_rol_user_log recomendador_visitas_user_a_nodo recomendador_users_con_rolles_comunes_log recomendador_determinar_vecindario recomendador_posibles_recomendaciones recomendador_peso_item recomendador_listos_recomendar

Tabla 16: Tarjeta CRC Generar recomendaciones de acuerdo al perfil y a la navegación del usuario.

Tarjeta CRC: Generar recomendaciones de acuerdo al perfil y a la navegación del usuario	
Responsabilidades	Colaboraciones
Recomendar al usuario hasta un máximo de diez contenidos que él no haya visitado, teniendo en cuenta su perfil y su navegación en el sitio.	recomendador_visitas_user_a_nodo recomendador_cantidad_visitas_nodos recomendador_algoritmo_pearson recomendador_calculo_media recomendador_users_con_rolles_comunes_log recomendador_users_con_nodos_comunes_log_segun_listado recomendador_determinar_vecindario recomendador_posibles_recomendaciones recomendador_peso_item recomendador_items_colaborativo recomendador_mezclar_arreglos_conservando_llave recomendador_listos_recomendar

Tabla 17: Tarjeta CRC Mostrar listado de recomendaciones.

Tarjeta CRC: Mostrar listado de recomendaciones.	
Responsabilidades	Colaboraciones
Mostrar en un bloque hasta un máximo de diez recomendaciones realizadas al usuario autenticado.	recomendador_block_view recomendador_theme recomendador_listos_recomendar

	recomendador_recomenUsers
--	---------------------------

Tabla 18: Tarjeta CRC Mostrar listado de contenidos más visitados.

Tarjeta CRC: Mostrar listado de contenidos más visitados.	
Responsabilidades	Colaboraciones
Mostrar en un bloque los cinco contenidos más visitados por otros usuarios, que el usuario autenticado no ha visitado.	recomendador_mas_visitados recomendador_recomenUsers recomendador_block_view recomendador_theme

Conclusiones parciales

Las funcionalidades y características del módulo permitieron definir las opciones que tendrá el mismo teniendo en cuenta las necesidades del cliente. Las HU permitieron delimitar las tareas a realizar por los desarrolladores para llevar a cabo la implementación de cada funcionalidad, así como establecer el tiempo aproximado que se dedicará para el cumplimiento de las mismas. Las tarjetas CRC confeccionadas ayudaron a establecer la relación entre las funcionalidades a implementar y determinar la responsabilidad y colaboración entre ellas. La metodología XP aportó los artefactos necesarios que darán paso a la fase de implementación del módulo.

Capítulo 3: Implementación y prueba del sistema

Capítulo 3: Implementación y prueba del sistema

Introducción

El presente capítulo se centra en las fases de Implementación y Prueba de la metodología seleccionada, tomando como base el proceso de Planificación y Diseño descrito en el capítulo anterior, lo que dio paso al desarrollo de entregas funcionales que dan cumplimiento a las necesidades del cliente. En el mismo se muestran las tareas generadas por cada HU, así como algunas características y observaciones que se tuvieron en cuenta para el desarrollo del módulo. Se realizan las pruebas propuestas por la metodología seleccionada, documentando los resultados arrojados.

3.1. Fase de Implementación

La parte más importante en el proceso XP es el desarrollo. Esta metodología propone comenzar la implementación de la solución partiendo de una arquitectura lo más flexible posible, con el propósito de que los desarrolladores puedan reestructurar el sistema sin cambiar su comportamiento y así remover duplicaciones de código, mejorar la comunicación, simplificar el código o agregar flexibilidad. Debido a ello, la solución tiene una arquitectura simple y muy bien definida.

La programación extrema propone una forma iterativa para la implementación de un software junto a las prácticas. Al terminar cada iteración se obtiene una versión del producto funcional que debe ser probado y mostrado al cliente.

3.1.1. Iteraciones

Para el desarrollo de la aplicación se definieron tres iteraciones donde se describen las HU que van a estar implicadas en cada iteración. Para llevar a cabo la correcta implementación de las HU se deben definir por los desarrolladores las tareas de ingeniería que se realizarán en cada iteración.

A continuación se muestra algunas de las tareas de ingeniería pertenecientes a la iteración 1. Las tareas correspondientes a las demás iteraciones pueden ser consultadas en los anexos (ver [Anexo 1](#)).

Tabla 19: Tareas de Ingeniería para la primera iteración.

Tarea de ingeniería	
No. de tarea: 1	No. de HU: 1
Nombre de la tarea: Configuración del perfil de usuario	
Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	

Descripción: se especificarán los datos que mostrará el módulo recomendador. Se crearán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las funcionalidades.	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 1
Nombre de la tarea: Creación de la tabla en la base de datos correspondiente a la funcionalidad perfil de usuario.	
Tipo de tarea: desarrollo	Estimación: 0.6
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se creará un archivo de tipo .install donde se crea la tabla “perfil_user” para posibilitar la persistencia de los datos.	
Tarea de ingeniería	
No. de tarea: 3	No. de HU: 1
Nombre de la tarea: Construcción de la funcionalidad perfil de usuario.	
Tipo de tarea: desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se modificará el código creando la funcionalidad que le permitirá al usuario crearse su perfil. Se utiliza la integración de los ganchos ¹¹ : recomendador_menu, recomendador_form y recomendador_form_submit para posibilitar el resultado esperado.	
Tarea de ingeniería	
No. de tarea: 4	No. de HU: 1
Nombre de la tarea: Configuración 2 de la funcionalidad perfil de usuario.	
Tipo de tarea: desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se configurará la interfaz gráfica, permitiendo el acceso a la base de datos para actualizar los campos correspondientes.	
Tarea de ingeniería	
No. de tarea: 1	No. de HU: 2
Nombre de la tarea: Configuración del generar recomendaciones según la navegación del usuario	

¹¹Palabra usada en la comunidad de Drupal para nombrar los métodos de la interfaz que deben implementar los módulos desarrollados para este CMS.

Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se actualizarán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las nuevas funcionalidades.	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 2
Nombre de la tarea: Construcción de la tabla recomendaciones.	
Tipo de tarea: desarrollo	Estimación: 1.7
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se modificará el archivo de tipo .install donde se creará la tabla “recomendaciones” para posibilitar la persistencia de los datos.	
Tarea de ingeniería	
No. de tarea: 3	No. de HU: 2
Nombre de la tarea: Construcción de funcionalidades para generar recomendaciones según la navegación del usuario.	
Tipo de tarea: desarrollo	Estimación: 1.7
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se modificará el código creando las funcionalidades para permiten generar recomendaciones que puedan ser de interés del usuario, a partir de su navegación.	

3.2. Estructura de Drupal

La estructura de capas y datos del CMS Drupal está conformada por varios elementos base que mantienen los componentes organizados y flexibles (59).

El subsistema más importante de Drupal es el *node* (nodo), el cual representa los elementos básicos en que Drupal almacena la información o los contenidos. Un nodo puede ser cualquier tipo de contenido que se quiera mostrar en un sitio web. Tienen una serie de campos de información asociados tales como: autor, fecha, título y cuerpo del contenido.

Otro subsistema importante en Drupal es el *modules* (módulos). Los módulos operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sistema. Son extensiones que se instalan en el sitio web proporcionándoles nuevas funcionalidades.

Los subsistemas de *blocks* (bloques) y *menus* (menús) de Drupal se encargan de la configuración y la visualización de los contenidos. Los bloques pueden ser configurados en dependencia de la necesidad, es decir, pueden ser usados para que se muestren en ciertas páginas definidas o solo para cierto tipo de usuarios. Los menús de Drupal son una colección de enlaces a diferentes contenidos de un sitio web y que se muestran dentro de un bloque en una zona concreta.

La seguridad y control de los usuarios se garantiza con la capa *user permissions* (permisos de usuario), subsistema que es clave para manejar la integridad de la información almacenada en un sitio web. En Drupal se pueden crear todos los roles de usuarios que se necesiten y asignarles los permisos que tendrán sobre las funcionalidades que se estimen convenientes.

Por último, el subsistema *themes* (temas) es un conjunto estructurado de código (como un módulo) que proporciona herramientas para la transformación de los datos con un formato de salida. Los sitios utilizan al menos un tema para que todas las páginas apliquen un aspecto coherente y personalizado.

3.3. Creación de las tablas adicionales en la base de datos

Por defecto, la instalación de Drupal crea 74 tablas que son utilizadas por el CMS para almacenar y manejar datos de nodos, usuarios, bloques, menús, contenidos, entre otros. El diseño de la estructura de las tablas que se agregaron a la base de datos de Drupal es el primer paso para comenzar la codificación de la solución informática. Para completar el desarrollo del módulo recomendador y teniendo en cuenta que las nuevas funcionalidades incluyen nuevos contenidos, fue necesario crear dos tablas en el módulo desarrollado (ver Figura 6).

De acuerdo a los resultados arrojados por la investigación para resolver el problema de la recomendación se debe establecer una conexión entre los usuarios y los contenidos (módulo *taxonomy*) demostrando que esta relación son las transacciones que se establecen entre el usuario y los ítems. Para el desarrollo de la aplicación se establece una relación de mucho a mucho entre la tabla *users* (contiene información referente a los usuarios) y la tabla *node* (contiene los contenidos presentes en el sitio web) provocando el surgimiento de una nueva tabla recomendaciones (esta contiene la relación de los usuarios con los contenidos).

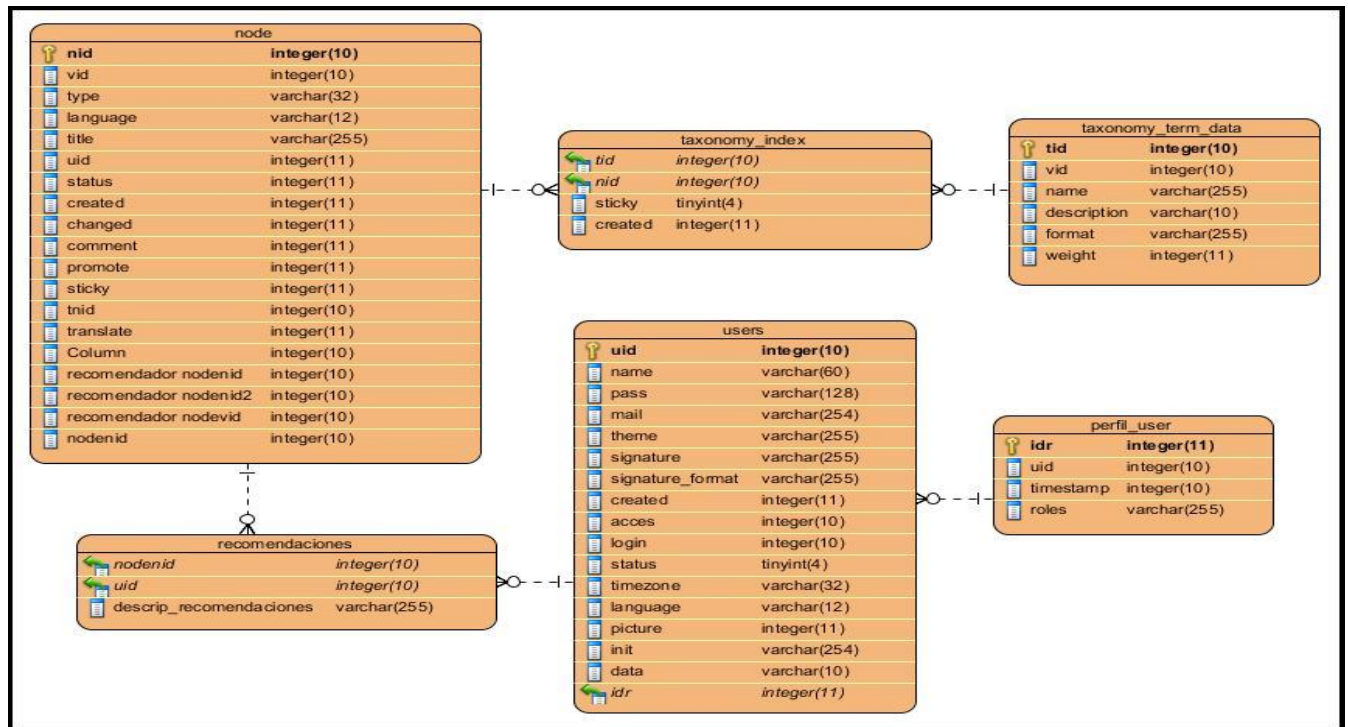


Figura 6. Tablas adicionadas a la base de datos de Drupal.

3.4. Módulo desarrollado para la recomendación

El módulo que se desarrolló lleva como nombre **Recomendador**. El mismo tiene como propósito extraer características de los objetos no conocidos por el usuario autenticado y compararlas con su perfil para predecir sus preferencias sobre tales objetos. Además, le muestra al usuario el listado de recomendaciones de acuerdo a lo que haya definido en su perfil de usuario y a las evidencias registradas en la aplicación.

La programación del módulo recomendador se realizó en pareja. En una primera etapa de la implementación se creó el archivo con la información del módulo (.info) y posteriormente otro archivo (.install) para crear la estructura de las tablas asociadas. La implementación de las funcionalidades se llevó a cabo en un tercer archivo (.module). La estructura del módulo se encuentra en el directorio sites/all/modules/recomendador desde donde se verificó su ejecución (ver Figura 7).

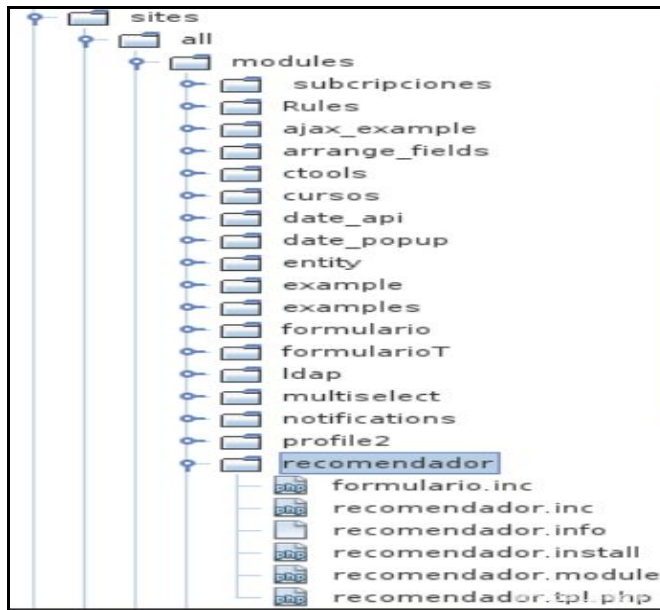


Figura 7: Ubicación del módulo recomendador en la estructura de archivos de Drupal.

Además, es necesario destacar que debido a la manera particular que Drupal establece la escritura de módulos, en la implementación del módulo de la solución están reflejados patrones de diseño creacionales.

3.5. Aplicación de patrones de diseño

Un patrón de diseño es una solución, reutilizable y con una efectividad ya probada, aplicable a diferentes problemas de diseño en distintas situaciones, es decir, es una solución ya creada para un problema que se da comúnmente en el desarrollo de software. Fueron creados con el objetivo de ahorrar tiempo y esfuerzo, así como aumentar la eficiencia y fiabilidad de las soluciones.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular (60). En el desarrollo del módulo se tuvieron en cuenta los siguientes patrones GOF¹² evidenciados en Drupal.

Comando

Muchos de los ganchos definidos por el núcleo de Drupal usan el patrón Comando (*Command*) para reducir el número de funciones que son necesarias implementar, pasando la operación como parámetro

¹² (Ganf of Four) patrones del diseño (60).

conjuntamente con los argumentos. De esta forma, se evidencia en el sistema de ganchos, que cada módulo no necesariamente debe definir cada uno, sino solo los que tienen que ver con su propósito y funcionamiento. En el desarrollo de la solución propuesta se evidencia este patrón, pues los módulos implementados no tienen que definir cada gancho, sino los que se vayan a aplicar. Algunos de los ganchos empleados en la solución fueron el *hook_menu*, *hook_permission* y el *hook_form*.

Instancia única

Este patrón se evidencia si se asocian los módulos y los temas como objetos, cada uno de ellos en sí no almacena datos, sino que representan un conjunto de funciones que extienden el núcleo para añadirle funcionalidades al mismo, cada uno por separado.

A pesar de la falta de clases explícitamente declaradas en Drupal, muchos paradigmas orientados a objetos lo utilizan todavía en su diseño. Un módulo en Drupal es una instancia con funciones únicas, por lo que en la aplicación, el módulo recomendador puede ser pensado como una clase, de ahí se puede decir que sigue el patrón de diseño creacional Instancia única (*Singleton*).

Puente

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón Puente (*Bridge*). Cada módulo debe ser programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar nuevas funcionalidades siguiendo la API¹³ definida, para añadir soporte a nuevos sistemas gestores de base de datos, sin necesidad de modificar el código de los módulos.

En la solución se evidencia el uso de este patrón en diferentes funciones como *recomendador_nodos_visitado* y *recomendador_rol_user_log*, pues las consultas realizadas a la base de datos se realizan para que funcione independiente del SGBD.

Por otra parte, XP promueve la programación basada en estándares, de manera tal que el código sea fácilmente entendible por todo el equipo y facilite la recodificación (61). En ese sentido, se aplicó la utilización de estándares de codificación, lo que fortalece la comprensión de la pareja de programadores y hace que el código sea más legible.

¹³ *Application Programming Interface* o Interfaz de Programación de Aplicaciones.

3.6. Estándares de codificación

Los estándares de codificación son reglas que se utilizan para la escritura del código fuente. Permiten que los desarrolladores puedan interpretar de manera eficiente la escritura del código, asegurando que todos trabajen de forma coordinada y en un vocabulario común.

Para el desarrollo del módulo se tuvieron en cuenta los siguientes estándares de codificación definidos en drupal.org:

Etiquetas de apertura y cierre de php

Cuando se esté escribiendo en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>` y en ningún caso la versión corta `<? y ?>`.

Se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos `.module` y `.inc`. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`) que serían identificados como salida HTML.

La etiqueta de cierre final del archivo (`?>`) es opcional en Drupal. No debe confundirse esto con el uso normal del lenguaje PHP en archivos que también contienen HTML (como los archivos de plantilla `.tpl.php`) donde cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.

Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos a ambos lados del operador, por un espacio. Por ejemplo, `$numero = 3`, en lugar de `$numero=3`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.=`, `+=`, `-=`, entre otros. Los operadores unarios como `++`, `--` no deben tener separación. Por ejemplo, `$numero++`.

Uso de comillas

Se pueden usar tanto las comillas simples (`'cadena'`) como las comillas dobles (`"cadena"`) para delimitar las cadenas de caracteres.

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, `"<h1>${title}</h1>"`. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna simple.

Uso de punto y coma (;) en código php

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo `<?php print $title ?>` en Drupal es siempre obligatorio: `<?php print $title;?>`

Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, entre otros, para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación que tendrán extensión .txt y el nombre es mayúsculas. Por ejemplo, README.txt, INSTALL.txt.

3.7. Instalación de módulos

Para instalar un módulo en el sistema lo primero que se debe realizar es copiar la carpeta contenedora de los archivos al directorio `sites/all/modules` con el nombre definido. Luego en la interfaz administrativa se visita el vínculo “Módulos” en el menú superior del sistema mostrando una lista de los disponibles. Para realizar la activación de un módulo basta con seleccionarlo y guardar los cambios. La interfaz se actualiza y se activan los cambios realizados.

En ocasiones al instalar un módulo es necesario dirigirse hacia la página de permisos para establecer los privilegios referentes al mismo o hacia la página de configuración para cambiar sus opciones. Algunas de las configuraciones básicas que se deben realizar son:

- Actualización del *cron*¹⁴ de Drupal para iniciar las tareas automáticas del sistema.
- Establecer el español como idioma.
- Fijar la disposición de bloques y menús.
- Determinar los formatos de entradas.

¹⁴ Programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o *scripts* a una hora o fecha específica.

Una vez desarrollado un software este puede tener defectos o fallos, por lo que es necesario probarlo para descubrir y corregir la mayor cantidad de errores posibles y así garantizar un producto con calidad.

3.8. Fase de Prueba

Las pruebas constituyen un aspecto importante en el proceso de elaboración del módulo, pues permiten medir el éxito de las funcionalidades del mismo. La metodología XP divide las pruebas en dos grupos.

Pruebas unitarias: son consideradas 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 (33).

Pruebas de aceptación: son creadas en base a las HU, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios, para comprobar que una HU ha sido correctamente implementada. Las pruebas de aceptación son consideradas como pruebas de caja negra. 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 (62).

3.8.1. Pruebas unitarias

El módulo Testing incorporado al núcleo de Drupal 7 permite definir y realizar las pruebas de un sitio de forma automatizada. Se realizan para verificar que un sistema funciona como se espera, de ahí que algunas metodologías de desarrollo de software como XP proponen escribir pruebas antes de escribir el código, lo cual es perfectamente aplicable a módulos de Drupal. Por lo que utilizando el módulo Testing se pueden realizar pruebas unitarias para validar el código.

Una prueba unitaria, es un tipo de algoritmo independiente que se encarga de verificar de manera simple y rápida el comportamiento de una parte mínima de código, de forma individual y sin alterar el funcionamiento de otras partes de la aplicación. En la presente investigación, para el desarrollo de las pruebas automatizadas a través del módulo Testing, se siguió el capítulo 25 del libro Apress Pro Drupal 7 Development. Al finalizar estas pruebas se arrojaron los siguientes resultados (ver Figura 8):

MENSAJE	GRUPO	NOMBRE DE ARCHIVO	LÍNEA	FUNCIÓN	ESTADO
Created permissions: administer blocks	Role	recomendador.test	11	recomendadorTestCase->setUp()	✓
User created with name <i>njpYEwsg</i> and pass <i>x7Fwcs4rDr</i>	User login	recomendador.test	11	recomendadorTestCase->setUp()	✓
User created with name <i>tTDdq7n9</i> and pass <i>mJwdh9BqC2</i>	User login	recomendador.test	12	recomendadorTestCase->setUp()	✓
El modulo Recomendador define 1 bloque	Other	recomendador.test	28	recomendadorTestCase->testBlockInfo()	✓
Enabled modules: <i>recomendador</i>	Other	recomendador.test	10	recomendadorTestCase->setUp()	✓
Created role of name: <i>h1oINJmd</i> , id: 4	Role	recomendador.test	11	recomendadorTestCase->setUp()	✓
Created permissions: administer blocks	Role	recomendador.test	11	recomendadorTestCase->setUp()	✓
User created with name <i>yKyDLRTw</i> and pass <i>GWcdY26cmG</i>	User login	recomendador.test	11	recomendadorTestCase->setUp()	✓
User created with name <i>ckvNsf0s</i> and pass <i>hYW4gvVRB9</i>	User login	recomendador.test	12	recomendadorTestCase->setUp()	✓
GET http://localhost/drupal/?q=user returned 200 (7.21 KB).	Browser	recomendador.test	33	recomendadorTestCase->testrecomendadorBlock()	✓

Figura 8: Resultados de las pruebas unitarias.

Se ejecutaron 32 pruebas en 4 minutos y 22 segundos y todas devolvieron resultados satisfactorios. El intento de hacerlo manualmente sin la automatización de pruebas habría tomado horas, bajo el riesgo de ocurrir algún error humano durante la ejecución de las mismas.

3.8.2. Prueba de aceptación

Las pruebas de aceptación se realizan para asegurar que las funcionalidades cumplen su objetivo y si satisfacen las necesidades del cliente. Los clientes son los responsables de verificar que los resultados sean correctos. Se realizan sobre la interfaz del software y son completamente indiferentes al comportamiento interno y a la estructura del programa (62).

A continuación se exponen las pruebas de aceptación realizadas al módulo, divididas en las iteraciones correspondientes a cada HU:

En la iteración 1 se hacen las pruebas de aceptación para las HU 1 y 2, comprobando junto al cliente que se cumplan los objetivos en cada funcionalidad.

Tabla 20: Caso de prueba de aceptación HU1_P1.

Caso de Prueba de Aceptación	
Código: HU1_P1	No. de HU: 1
Nombre: Perfil de usuario	
Descripción: prueba para la funcionalidad que le permite al usuario llenar un formulario con el rol que desempeña.	
Condiciones de ejecución: el usuario debe estar autenticado.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario se autentica y procede a llenar su perfil mediante la opción “Perfil de usuario”. • Inserta los datos en un formulario y selecciona la opción “Enviar” o “Cancelar”, dependiendo de la acción que desea realizar. • En caso que presione “Enviar” y no haya insertado ningún dato se muestra el siguiente mensaje: “El campo roles es obligatorio”, en caso contrario se almacena la información en la base de datos y se le envía un mensaje al usuario diciendo que sus datos han sido guardados. • En caso que presione la opción “Cancelar” no se guardan los datos y se direcciona a la página principal. 	
Resultado esperado: el perfil de usuario es mostrado satisfactoriamente y el módulo responde de forma correcta a las opciones “Enviar” y “Cancelar”.	
Evaluación de la prueba: satisfactoria	

Tabla 21: Caso de prueba de aceptación HU2_P2.

Caso de Prueba de Aceptación	
Código: HU2_P2	No. de HU: 2
Nombre: Generar recomendaciones según la navegación del usuario.	
Descripción: prueba para la funcionalidad que permite generar recomendaciones según la navegación del usuario.	
Condiciones de ejecución: el usuario debe estar autenticado.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> • El usuario visita un ítem determinado • El módulo genera otros ítems que sean similares al visitado. • En caso que el usuario visite otro ítem, se van agregando recomendaciones atendiendo a este criterio. 	

<ul style="list-style-type: none"> Solo se realizan diez recomendaciones, partiendo de la premisa de que si el usuario ya visitó determinado ítem, el mismo no se le recomienda.
Resultado esperado: los datos son actualizados satisfactoriamente.
Evaluación de la prueba: satisfactoria

En la iteración 2 se hacen las pruebas de aceptación para las HU 3 y 4, comprobando junto al cliente que cada funcionalidad cumpla con su objetivo.

Tabla 22: Caso de prueba de aceptación HU3_P3.

Caso de Prueba de Aceptación	
Código: HU3_P3	No. de HU: 3
Nombre: Generar recomendaciones según el perfil del usuario.	
Descripción: prueba para la funcionalidad que permite generar recomendaciones según el perfil del usuario.	
Condiciones de ejecución: el usuario debe estar autenticado.	
Entrada/ Pasos de ejecución:	
<ul style="list-style-type: none"> El usuario llena su perfil seleccionando el rol o los roles que le interesan, datos que se almacenan en la base de datos. Se genera la recomendación a partir de su perfil. Las recomendaciones están dadas por los ítems visitados por usuarios que tengan perfil similar al usuario actual y que este no haya visitado. 	
Si no existen usuarios con un perfil semejante al del usuario actual no se realiza la recomendación. Se realizan las recomendaciones correspondientes cuando el usuario comience a navegar. El total de recomendaciones no excede de diez.	
Resultado esperado: los datos son actualizados satisfactoriamente.	
Evaluación de la prueba: satisfactoria	

Tabla 23: Caso de prueba de aceptación HU4_P4.

Caso de Prueba de Aceptación	
Código: HU4_P4	No. de HU: 4
Nombre: Generar recomendaciones según el perfil y la navegación del usuario.	
Descripción: prueba para la funcionalidad que permite generar recomendaciones según el perfil y la navegación del usuario.	

Condiciones de ejecución: el usuario debe estar autenticado.
Entrada/ Pasos de ejecución: <ul style="list-style-type: none"> El usuario llena su perfil seleccionando el rol o los roles que le interesan almacenándose los datos en la base de datos. <p>Si existen usuarios con perfil semejante a él y además han visitado algún ítem en común:</p> <ul style="list-style-type: none"> Se genera una recomendación a partir de su perfil y de su navegación. Deben mostrarse los ítems que son común en ambos casos, los correspondientes a su perfil y los que se corresponden a su navegación, siempre que el usuario no los haya visitado. Solo se realizan hasta un máximo de diez recomendaciones. En caso que existan más de diez, se recomiendan las que más visitas posean por parte de otros usuarios.
Resultado esperado: los datos son actualizados satisfactoriamente.
Evaluación de la prueba: satisfactoria

En la iteración 3 se hacen las pruebas de aceptación para las HU 5 y 6, comprobando junto al cliente que cada funcionalidad cumpla con su objetivo.

Tabla 24: Caso de prueba de aceptación HU5_P5.

Caso de Prueba de Aceptación	
Código: HU5_P5	No. de HU: 5
Nombre: Mostrar listado de recomendaciones	
Descripción: prueba para la funcionalidad que le permite al usuario visualizar el listado de recomendaciones.	
Condiciones de ejecución: el usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Debe mostrar al usuario mediante un bloque las recomendaciones generadas en las HU 2,3 y 4	
Resultado esperado: la aplicación debe mostrar el listado de recomendaciones.	
Evaluación de la prueba: satisfactoria	

Tabla 25: Caso de prueba de aceptación HU6_P6.

Caso de Prueba de Aceptación	
Código: HU6_P6	No. de HU: 6
Nombre: Mostrar listado de contenidos más visitados	
Descripción: prueba para la funcionalidad que le permite al usuario visualizar los contenidos más	

visitados por otros usuarios.

Condiciones de ejecución: el usuario debe estar autenticado.

Entrada/ Pasos de ejecución:

- Cuando el usuario entre al sistema, si nunca ha llenado su perfil ni ha navegado en el sitio, se muestran en un bloque los contenidos más visitados por otros usuarios.
- Si el usuario ya llenó su perfil pero no existe un usuario con perfil semejante a él y este no ha navegado nunca por el sitio, se siguen mostrando en el bloque los contenidos más visitados por otros usuarios.
- En caso que sí existan usuarios con perfil semejante, pero ya el usuario haya visitado los ítems correspondientes a la recomendación se muestran nuevamente en el bloque los contenidos más visitados por otros usuarios, que este no haya visitado. Si los visitó todos, entonces no se realiza la recomendación.
- Cuando el usuario ha navegado por el sitio sin llenar su perfil y no existen ítems a recomendar, se muestran en el bloque los ítems más visitados por otros usuarios que él no conoce. Si los visitó todos entonces no se realiza la recomendación.
- Si el usuario tiene lleno su perfil y ha navegado por el sitio, pero ya ha visitado todas las recomendaciones correspondientes, se muestran en el bloque los ítems más visitados que él no conoce. Si los visitó todos entonces no se realiza la recomendación.

Resultado esperado: la aplicación debe mostrar el listado de recomendaciones.

Evaluación de la prueba: satisfactoria

Las pruebas a las funcionalidades se realizaron en 3 iteraciones, detectándose un total de 9 no conformidades significativas, 2 no significativas y 3 recomendaciones. A continuación se muestra la cantidad de no conformidades identificadas por iteraciones (ver Figura 9):

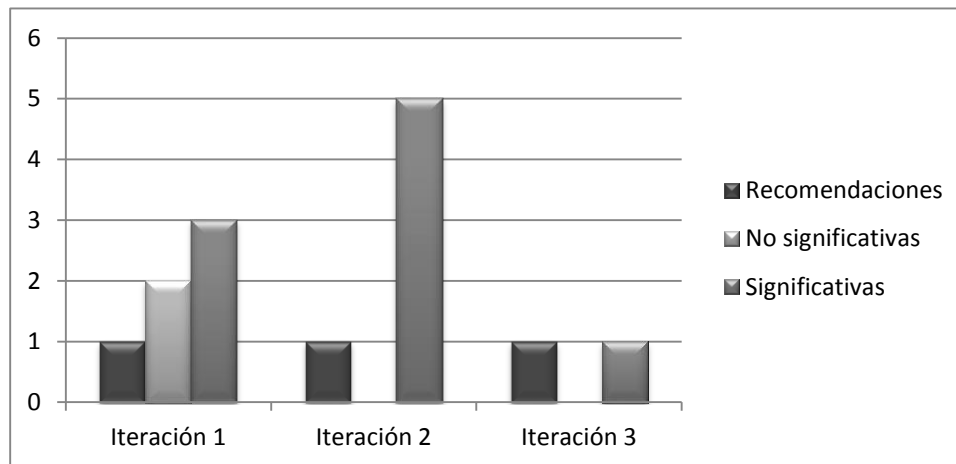


Figura 9. Resultados de las pruebas de aceptación.

Las no conformidades (NC) no significativas, se centraron en errores ortográficos (omisión de tilde y cambio de mayúscula por minúscula). Las significativas estuvieron centradas en las respuestas que proporcionaba el módulo ante las acciones del usuario y a errores de validación. Por otro lado las recomendaciones estuvieron dadas en función de mejorar el proceso de recomendación: se sugirió cuándo realizar una recomendación o no, sobre el diseño del formulario y cuándo mostrar los cinco elementos más visitados.

Las NC encontradas luego de concluida cada iteración de pruebas se analizaron por parte del equipo de desarrollo para luego corregir los errores detectados, lo que contribuyó a mejorar la calidad y funcionalidad del software.

Conclusiones parciales

La descripción de la propuesta de solución en el capítulo anterior permitió desarrollar las funcionalidades descritas utilizando las herramientas, lenguajes y tecnologías seleccionados en el capítulo 1. Los estándares de codificación permitieron que ambos desarrolladores pudieran trabajar de manera coordinada y pudiesen interpretar eficientemente el código. Mediante las pruebas de aceptación se pudo documentar las no conformidades detectadas en el módulo durante su desarrollo, las cuales fueron corregidas. Al finalizar la última iteración se obtuvo un módulo que cumple con las especificaciones establecidas por el cliente y que permite que los usuarios reciban recomendaciones de manera personalizada, teniendo en cuenta sus áreas de interés.

Conclusiones generales

Después de desarrollar el presente trabajo y analizar los resultados obtenidos, se arriban a las siguientes conclusiones:

- Con el desarrollo del marco teórico referencial se determinó confeccionar un módulo recomendador de recursos de información híbrido.
- El diseño del módulo recomendador permitió identificar las funcionalidades a implementar y la manera de llevar a cabo el proceso de desarrollo.
- Con la implementación se obtuvo un módulo recomendador de recursos de información, para la intranet del Centro FORTES, teniendo en cuenta las áreas de interés de los usuarios.
- Las pruebas realizadas al módulo recomendador corroboraron que la aplicación satisface las necesidades del cliente, pues cumple con las funcionalidades y características identificadas.

Recomendaciones

A partir de la investigación realizada se sugieren las siguientes recomendaciones con el objetivo de que muchas de las consideraciones dadas aquí sean objeto de revisión, de completamiento y de perfeccionamiento en futuros trabajos. Se recomienda:

- Investigar el rango óptimo de recomendaciones que puede recibir una persona, así como la cantidad de vecinos, para lograr una mejor recomendación por parte del sistema.
- Implantar este módulo en otros sitios web de la universidad.
- Estudiar la manera de realizar el servicio de recomendaciones a través del correo electrónico.

Referencias bibliográficas

1. **Ojeda, F.** *Las personas son el valor más importante de una empresa o una organización*, 2009
2. **Ponjuan, Gloria.** *Introducción a la gestión del conocimiento*, 2006
3. **Rebolledo, Gustavo.** *Gestión, Calidad y Agregación de valor en información*, 1993
4. **de Chiavenato, Idalberto.** *Introducción a la Teoría General de la Administración*. Séptima Edición. s.l. : McGraw-Hill Interamericana, 2006. pág. 110
5. **Curto, Josep.** Information Management. *¿Qué es la gestión de la información?* [En línea] 28 de Noviembre de 2006. [Citado el: 2 de Mayo de 2014.] [Disponible en: <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>]
6. **Ponjuán, G.** *Gestión de la información en las organizaciones: principios, conceptos y aplicaciones*. Santiago de Chile : s.n., 1998
7. **Cronin, B.** *Information management from strategies to action*. London: Aslib : s.n., 1985
8. **Morales Campos, E. y Rodríguez Gallardo, A.** *La Biblioteca del futuro México*, 2005
9. **Sánchez Pereira, A.** *La Diseminación Selectiva de la Información a través de internet: Propuesta de Boletín Electrónico de información Bibliográfica Especializada en Economía, utilizando las bases de datos de Clases*. México : s.n., 2003
10. **Winer, D.** ScriptingXML. [En línea] 15 de Diciembre de 1997. [Citado el: 18 de Marzo de 2014.] [Disponible en: <http://davenet.scripting.com/1997/12/15/scriptingNewsInXML>]
11. RSS. *¿Qué es RSS?* [En línea] [Citado el: 1 de Mayo de 2014.] [Disponible en: <http://www.rss.nom.es>]
12. **Konstan, Joseph A.** *Introduction to Recommender Systems*. University of Minnesota: s.n., 2008
13. **Yager, Ronald R.** *Fuzzy logic methods in recommender systems*. Elsevier North-Holland : s.n., 2003
14. **Konstan, Josep.** *Introduction To Recommender Systems: Algorithms and Evaluations*, 2004
15. **Peña Henríquez, Fernando A. y Riffo Carrillo, Ricardo E.** *Revisión, selección e implementación de un algoritmo de recomendación de material bibliográfico utilizando tecnología j2ee*. Concepción, Chile : s.n., 2008
16. **Santana, Anabel y Pozo, Yandry.** *Sistema de recomendación de objetos de aprendizaje en el Repositorio de Objetos de Aprendizaje RHODA*. La Habana : s.n., 2013
17. **Pupo, H.** *Sistema recomendador híbrido de cuestionarios interactivos como herramienta de apoyo al proceso de enseñanza-aprendizaje*, 2012
18. **Pazzani, M. y Billsus, DI.** *Content-Based Recommendation System. The Adaptive Web: Methods and Strategies of Web Personalization*. Springer Berlin / Heidelberg : s.n., 2007
19. **Sarwar, B.** *Item-based collaborative filtering recommendation algorithms*, 2001
20. **Galán Nieto, S.** *Filtrado Colaborativo y Sistemas de Recomendación*, 2007
21. **Oswaldo, C.S.** *Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos*, 2006

22. **Matos, V.** *Sistemas de Recomendación y Personalización*, 2006
23. **Ricci, F., Rokach, L. y Shapira, B.** *Introduction to Recommender Systems Handbook*. 2011 ISBN:978-0-387-85819-7.
24. **Burke, R.** *Hybrid recommender systems: Survey and experiments*. Estados Unidos: Kluwer Academic Publishers : s.n., 2002. págs. 331–370, Vol. 12
25. **Vozalis, E.** *Analysis of Recommender Systems Algorithms*, 2009
26. **Blanco, Y.** *Tesis doctoral propuesta metodológica para el razonamiento semántico en sistemas de recomendación personalizada y automática. Aplicación al caso de contenidos audiovisuales*, 2007
27. **Lemire, D y Maclachlan, A.** *Slope One Predictors for Online Rating-Based Collaborative Filtering*, *SDM05*, 2005
28. **Calabuig, D y Ruescas, Ana V.** *Sistemas Recomendadores en Redes Sociales*. Universidad Politécnica de Valencia. 2013
29. **Pressman, R. S.** *Ingeniería del Software: Un enfoque práctico*. Madrid: Mc Graw-Hill : s.n., 1998
30. **Reynoso, C.** *Métodos Heterodoxos en Desarrollo de Software*, 2004
31. **XP-Extreme Programing** Ingeniería de Software. [En línea] [Citado el: 9 de Febrero de 2014.] [Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html]
32. **Shore, J. y Warden, S.** *The Art of Agile Development*. Estados Unidos : O'Reilly Media : s.n., 2008. ISBN -13: 978-0-596-52767-9
33. **Joskowicz, J.** *Reglas y Prácticas en eXtremeProgramming*. España : s.n., 2008
34. **Gil Rodríguez, F.** *Experto en Drupal 7. Nivel Avanzado*. Forcontu, 2011. ISBN: 978-84-939410-2-4
35. **Alfonso, X.** *Introducción a los Sistemas de Gestión de Contenido*, 2008
36. **Tomlinson, Todd y Vandyk, John K.** *Pro Drupal 7 Development*, 2010
37. **Centro de apoyo tecnológico a emprendedores.** *Estudio de los sistemas gestión de contenido web*, 2012
38. **Eguíluz Pérez, J.** *Introducción a Javascript*, 2008
39. **Dpto de Ingeniería electrónica.** *Introducción a XHTML y CSS*, 2006
40. **XHTML 1.0 Estricto: ¿XHTML o HTML?** [En línea] [Citado el: 2 de Abril de 2014.] [Disponible en: <http://xhtml-estricto.awardspace.com/articulos/xhtml-o-html>]
41. **de la Torre, A.** *PHPNuke*, 2006
42. **PHP: ¿Qué puede hacer PHP? - Manual.** [En línea] [Citado el: 10 de Febrero de 2014.] [Disponible en: <http://www.php.net/manual/es/intro-whatcando.php>]
43. **Leyva, D. y García, Miguel R.** *Diseño e implementación de un módulo para la realización de presentaciones web reusables sobre Moodle*, 2011
44. **Cabello, V. N.** *Introducción a las Base de Datos Relacionales*. Madrid : s.n., 2008. ISBN 987-84-9886-809-8.
45. **Gilfillan, Ian.** *La Biblia de MySQL*. España : s.n., 2003

46. MySQL: MySQL 5.0 Reference Manual. El servidor MySQL incrustado (embedded). [En línea] [Citado el: 3 de Abril de 2014.] [Disponible en: <https://dev.mysql.com/doc/refman/5.0/es/nutshell-embedded-mysql.html>]
47. **González, J., Seoane, J. y Robles, G.** *Introducción al software libre*, 2013
48. NetBeans IDE 7.4 Release Information. [En línea] [Citado el: 10 de Febrero de 2014.] [Disponible en: <https://netbeans.org/community/releases/74/>]
49. VISUAL PARADIGM©. Visual modeling tool for building enterprise applications. [En línea] [Citado el: 17 de febrero de 2014.] [Disponible en: <http://www.visual-paradigm.com/product/vpuml/features>]
50. **Jacobson, I., Booch, G. y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de Software*, 2000
51. Visual Paradigm para UML. [En línea] [Citado el: 12 de Abril de 2014.] [Disponible en: <http://www.software.com.ar/visual-paradigm-para-uml.html>]
52. Servidor-web. [En línea] [Citado el: 10 de Febrero de 2014.] [Disponible en: <http://mundogeek.net/etiqueta/servidor-web/>]
53. Welcome to The Apache Software Foundation! [En línea] [Citado el: 19 de Marzo de 2014.] [Disponible en: <http://www.apache.org/>]
54. **Kabir, Mohammed J.** *La Biblia Servidor Apache 2*, 1999. ISBN: 8441514682
55. **Bisquerra, R.** *Métodos de investigación educativa. Guía práctica*. Barcelona: CEAC : s.n., 1989
56. **Letelier, P. y Penadés, M.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia : s.n., 2008
57. **Beck, K.** *Extreme Programming Explained*. Addison-Wesley : s.n., 2000
58. **Bajo de Luke, M.J.** *Metodologías ágiles. Agile Alliance*, 2011
59. **Chaffer, Jonathan.** *Drupal programming from an object-oriented perspective*, 2012
60. **Gamma, E.** *Design Patterns. Elements of Reusable Object-Oriented Software*, 2003
61. **Beck, Kent.** *Embracing Change with Extreme Programming*. Addison Wesley Longman : s.n., 1999. ISBN-10: 0201616416 | ISBN-13: 978-0201616415
62. **Valdés, Abner G.** *Pruebas de sistema y pruebas de aceptación*, 2013

Bibliografía

1. **Alfonso X.** Introducción a los Sistemas de Gestión de Contenido, 2008
2. API, Interface de Programación de Aplicaciones [En línea] [Citado el: 18 de Marzo de 2014.] [Disponible en: <http://www.elwebmaster.com/referencia/api-interface-de-programacion-de-aplicaciones>]
3. **Bajo de Luke M.J.** Metodologías ágiles. Agile Alliance, 2011
4. **Beck K.** Extreme Programming Explained. Addison-Wesley, 2000
5. **Beck Kent** Embracing Change with Extreme Programming. Addison Wesley Longman, 1999. ISBN-10: 0201616416 | ISBN-13: 978-0201616415
6. Biblioteca Temporal de la Universidad de la Habana [En línea] [Citado el: 14 de diciembre de 2013] [Disponible en: <http://www.dict.uh.cu/diseminacion.html>]
7. **Bisquerra R.** Métodos de investigación educativa. Guía práctica. Barcelona: CEAC, 1989
8. **Blanco Y.** Tesis doctoral propuesta metodológica para el razonamiento semántico en sistemas de recomendación personalizada y automática. Aplicación al caso de contenidos audiovisuales, 2007
9. **Burke R.** Hybrid recommender systems: Survey and experiments. Estados Unidos: Kluwer Academic Publishers, 2002, Vol. 12 : págs. 331–370
10. **Cabello V. N.** Introducción a las Base de Datos Relacionales. Madrid, 2008. ISBN 987-84-9886-809-8
11. **Calabuig D y Ruescas Ana V.** Sistemas Recomendadores en Redes Sociales. Universidad Politécnica de Valencia, 2013
12. **Centro de apoyo tecnológico a emprendedores.** Estudio de los sistemas gestión de contenido web, 2012
13. **Chaffer Jonathan.** Drupal programming from an object-oriented perspective, 2012
14. **Chesani F.** Implicit Interest Indicators, Proceedings of the 6th International conference on intelligent user interfaces, 2001
15. **Chua T. S., Lim S. K., and Pung H. K.** Content-based retrieval of segmented images," ACM Multimedia, 1994
16. **Corniel Marla** Re-diseño de un sistema recomendador de estudios basado en minería web semántica. Venezuela, 2010
17. **Cronin B.** Information management from strategies to action. London: Aslib, 1985
18. **Curto Josep** Information Management. ¿Qué es la gestión de la información?. 28 de Noviembre de 2006. [Citado el: 2 de Mayo de 2014] [Disponible en: <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>]
19. **Davies, J., Studer, R. y Warren, P.** Semantic Web Technologies: Trends and Research in Ontology-Based Systems, 2006

20. **de Chiavenato Idalberto** Introducción a la Teoría General de la Administración. McGraw-Hill Interamericana. 2006. Séptima Edición: pág. 110
21. **de la Torre A.** PHPNuke, 2006
22. **Del Pino, J., Salazar, G. y Cedeño, V.** Adaptación de un Recomendador de Filtro Colaborativo Basado en el Usuario para la Creación de un Recomendador de Materias de Pregrado Basado en el Historial Académico de los Estudiantes, 2011
23. **Diz Monje, C., Guzmán Garate, J. y Sánchez García, J.** Catalogador automático de textos y recomendador de artículos del portal de revistas electrónicas de la biblioteca UCM. [Disponible en: <http://eprints.ucm.es/9887>]
24. **Dpto de Ingeniería electrónica.** Introducción a XHTML y CSS. Universidad de Huelva, 2006
25. **Drachsler, Hendrik, K., Hans G. y Koper, R.** Personal recommender systems for learners in lifelong learning networks the requirements, techniques and model, 2008
26. **Eguíluz Pérez J.** Introducción a Javascript, 2008
27. **Farzan R. and Brusilovsky P.** Social navigation support in a course recommender system. In Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2008
28. **Galán Nieto S.** Filtrado Colaborativo y Sistemas de Recomendación, 2007
29. **Gamma E.** Design Patterns. Elements of Reusable Object-Oriented Software, 2003
30. **Gerling V.** Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje. Universidad Nacional de Rosario. Argentina, 2009
31. **Gil Rodríguez Fran.** Experto en Drupal 7. Nivel Avanzado. Forcontu, 2011. ISBN: 978-84-939410-2-4.
32. **Gilfillan Ian.** La Biblia de MySQL. España, 2003
33. **González J., Seoane J. y Robles G.** Introducción al software libre, 2013
34. Guía Breve de Tecnologías XML [En línea] [Citado el: 25 de Marzo de 2014] [Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>]
35. HTML Source: HTML Tutorials [En línea] [Citado el: 25 de Marzo de 2014] [Disponible en: <http://www.yourhtmlsource.com/starthere/whatishtml.htm>]
36. **Jacobson I., Booch G. y Rumbaugh J.** El Proceso Unificado de Desarrollo de Software, 2000
37. **Jensen, Finn V. y Nielsen, Thomas D.** Bayesian Networks and Decision Graphs, 2007
38. **Joskowicz J.** Reglas y Prácticas en eXtremeProgramming. España, 2008
39. **Kabir Mohammed J.** La Biblia Servidor Apache 2, 1999. ISBN: 8441514682
40. **Konstan Josep** IntroductionTo Recommender Systems: Algorithms and Evaluations, 2004
41. **Konstan Joseph A.** Introduction to Recommender Systems. University of Minnesota, 2008

42. **Lemire D y Maclachlan A.** Slope One Predictors for Online Rating-Based Collaborative Filtering, SDM05, 2005
43. **Letelier P. y Penadés M.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia, 2008
44. **Leyva D. y García Miguel R.** Diseño e implementación de un módulo para la realización de presentaciones web reusables sobre Moodle, 2011
45. **Matos V.** Sistemas de Recomendación y Personalización. Universidad del Sur, 2006
46. **Morales Campos E. y Rodríguez Gallardo A.** La Biblioteca del futuro. México, 2005
47. MySQL: MySQL 5.0 Reference Manual. El servidor MySQL incrustado (embedded). [En línea] [Citado el: 3 de Abril de 2014] [Disponible en: <https://dev.mysql.com/doc/refman/5.0/es/nutshell-embedded-mysql.html>]
48. NetBeans IDE 7.4 Release Information [En línea] [Citado el: 10 de Febrero de 2014] [Disponible en: <https://netbeans.org/community/releases/74/>]
49. **Ojeda F.** Las personas son el valor más importante de una empresa o una organización, 2009
50. **Oswaldo C.S.** Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos, 2006
51. **Paterek, A.** Improving regularized singular value decomposition for collaborative filtering. In Proceedings of KDD Cup and Workshop, 2007
52. **Pazzani M. y Billsus DI.** Content-Based Recommendation System. The Adaptive Web: Methods and Strategies of Web Personalization. Springer Berlin/Heidelberg, 2007
53. **Peña Henríquez Fernando A. y Riffo Carrillo, Ricardo E.** Revisión, selección e implementación de un algoritmo de recomendación de material bibliográfico utilizando tecnología j2ee. Chile, 2008
54. PHP: ¿Qué puede hacer PHP? [En línea] [Citado el: 10 de Febrero de 2014] [Disponible en: <http://www.php.net/manual/es/intro-whatcando.php>]
55. **Ponjuan Gloria** Introducción a la gestión del conocimiento, 2006
56. **Pressman R. S.** Ingeniería del Software: Un enfoque práctico. Madrid: Mc Graw-Hill, 1998
57. Proyectos GNU. Licencias [En línea] [Citado el: 10 de Marzo de 2014] [Disponible en: <http://www.gnu.org/licenses/licenses.html>]
58. **Pupo H.** Sistema recomendador híbrido de cuestionarios interactivos como herramienta de apoyo al proceso de enseñanza-aprendizaje, 2012.
59. **Rebolledo Gustavo** Gestión, Calidad y Agregación de valor en información, 1993
60. **Reynoso C.** Métodos Heterodoxos en Desarrollo de Software, 2004
61. **Ricci F., Rokach, L. y Shapira, B.** Introduction to Recommender Systems Handbook, 2011. ISBN:978-0-387-85819-7

62. RSS. ¿Qué es RSS? [En línea] [Citado el: 1 de Mayo de 2014] [Disponible en: <http://www.rss.nom.es/>]
63. **Sánchez Pereira A.** La Diseminación Selectiva de la Información a través de internet: Propuesta de Boletín Electrónico de información Bibliográfica Especializada en Economía, utilizando las bases de datos de Clases. México, 2003
64. **Santana Anabel y Pozo Yandry** Sistema de recomendación de objetos de aprendizaje en el Repositorio de Objetos de Aprendizaje RHODA . La Habana, 2013.
65. **Sarwar B.** Item-based collaborative filtering recommendation algorithms, 2001
66. Servidor-web [En línea] [Citado el: 10 de Febrero de 2014] [Disponible en: <http://mundogeek.net/etiqueta/servidor-web/>]
67. **Shore J. y Warden S.** The Art of Agile Development. Estados Unidos : O'Reilly Media, 2008. ISBN -13: 978-0-596-52767-9
68. **S. H. et al.** Amed- a personalized TV recommendation system. In Interactive TV: a Shared Experience, 2007
69. **Tomlinson Todd y Vandyk John K.** Pro Drupal 7 Development, 2010
70. **Valdés Abner G.** Pruebas de sistema y pruebas de aceptación, 2013
71. Visual Paradigm para UML [En línea] [Citado el: 12 de Abril de 2014] [Disponible en: <http://www.software.com.ar/visual-paradigm-para-uml.html>]
72. VISUAL PARADIGM©. Visual modeling tool for building enterprise applications [En línea] [Citado el: 17 de febrero de 2014] [Disponible en: <http://www.visual-paradigm.com/product/vpuml/features>]
73. **Vozalis E.** Analysis of Recommender Systems Algorithms, 2009
74. Welcome to The Apache Software Foundation! [En línea] [Citado el: 19 de Marzo de 2014] [Disponible en: <http://www.apache.org/>]
75. **Winer D.** ScriptingXML [En línea] [Citado el: 18 de Marzo de 2014] [Disponible en: <http://davenet.scripting.com/1997/12/15/scriptingNewsInXML>]

4.1 Anexos

Anexo 1: Tareas de ingeniería

Tabla 26: Tareas de ingeniería para la segunda iteración.

Tarea de ingeniería	
No. de tarea: 1	No. de HU: 3
Nombre de la tarea: Configuración del generar recomendaciones según el perfil del usuario	
Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se actualizarán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las nuevas funcionalidades.	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 3
Nombre de la tarea: Construcción de funcionalidades para generar recomendaciones según el perfil del usuario.	
Tipo de tarea: desarrollo	Estimación: 2.6
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se modificará el código creando las funcionalidades para generar recomendaciones que permitirán recomendarle al usuario los contenidos de su preferencia a partir de su perfil y de las preferencias de usuarios con perfil similar.	
Tarea de ingeniería	
No. de tarea: 1	No. de HU: 4
Nombre de la tarea: Configuración del generar recomendaciones según el perfil y las evidencias de navegación del usuario.	
Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se actualizarán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las nuevas funcionalidades	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 4
Nombre de la tarea: Construcción de funcionalidades para generar recomendaciones según el perfil y las evidencias de navegación del usuario.	

Tipo de tarea: desarrollo	Estimación: 2.6
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se modificará el código creando las funcionalidades para generar recomendaciones que permitirán recomendarle al usuario los contenidos de su preferencia a partir de su navegación y a las preferencias de usuarios con perfil similar.	

Tabla 27: Tareas de ingeniería para la tercera iteración.

Tarea de ingeniería	
No. de tarea: 1	No. de HU: 5
Nombre de la tarea: Configuración 1 del mostrar listado de recomendaciones.	
Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se actualizarán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las nuevas funcionalidades.	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 5
Nombre de la tarea: Configuración 2 de la funcionalidad mostrar listado de recomendaciones	
Tipo de tarea: configuración-desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se gestionarán los archivos incluidos sobre esta funcionalidad para el acceso a la base de datos y la presentación del listado de recomendaciones en la interfaz gráfica. Se utiliza la integración de los ganchos: <i>hook_theme</i> , <i>hook_block_info</i> , <i>hook_block</i> , <i>hook_block_view</i> , entre otros para posibilitar el resultado esperado.	
Tarea de ingeniería	
No. de tarea: 1	No. de HU: 6
Nombre de la tarea: Configuración 1 del mostrar listado de contenidos más visitados.	
Tipo de tarea: configuración - desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se actualizarán los archivos .module, .tpl.php, .info, .inc para la creación e implementación de las nuevas funcionalidades.	
Tarea de ingeniería	
No. de tarea: 2	No. de HU: 6

Nombre de la tarea: Configuración 2 de la funcionalidad mostrar listado de contenidos más visitados	
Tipo de tarea: configuración-desarrollo	Estimación: 0.4
Programador responsable: Massiel Agüero Cabreja y Aliosky Martínez Hernández	
Descripción: se gestionarán los archivos incluidos sobre esta funcionalidad para el acceso a la base de datos y la presentación del listado de contenidos más visitados recomendados en la interfaz gráfica. Se utiliza la integración de los ganchos: <i>hook_theme</i> , <i>hook_block_info</i> , <i>hook_block</i> , <i>hook_block_view</i> , entre otros para posibilitar el resultado esperado.	

Anexo 2: Entrevista

Para conocer las herramientas que se utilizan para facilitar acceso a la información en algunas instituciones cubanas como la UH, el Centro DESOFT y la UCI se realizó la siguiente entrevista:

1. ¿Qué vías o medio se emplea en la institución para facilitar el acceso a recursos de información?
2. ¿Las vías antes mencionadas cumplen con las expectativas esperadas por parte de los trabajadores? ¿Qué deficiencias han detectado estos que se hacen notables en dichos mecanismos de divulgación?