



JUNIO DE 2014

MÓDULO RECOMENDADOR DE LIBROS PARA LA TIENDA VIRTUAL DEL PORTAL CUBALITERARIA.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORA: YANET DE LOS ÁNGELES CÁRDENAS ALMIRA

Tutor: Ing. Rafael Jacobo Hidalgo Urbino

Tutora: Ing. Magdalys Martínez Jiménez

Declaración de Autoría

Declaro ser la única autora del trabajo de diploma Módulo recomendador de libros para la tienda virtual del portal CubaLiteraria y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo el presente a los 23 días del mes de junio del año 2014.

Firma de la tesista

Yanet de los Angeles Cárdenas
Almira

Firma del tutor

Ing. Rafael Jacobo Hidalgo
Urbino

Firma de la tutora

Ing. Magdalys Martínez
Jiménez

“



El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes es la oportunidad



- Paulo Coelho



”

Agradecimientos

Se me acaba la UCI y es ahora cuando me toca agradecerle a todos aquellos que han formado parte de mi vida durante estos maravillosos años. Muchas son las personas cuyo apoyo ha sido imprescindible a lo largo de esta etapa a todos los que aportaron su granito de arena, esto va para ustedes.

Le agradezco a mi mamá, a mi papá y a mi hermana porque sin ellos no hubiera podido llegar tan lejos, su apoyo incondicional, su dedicación y consejos me han convertido en la profesional que soy hoy.

A mi mamá Virgen, Mimi gracias por haber estado siempre conmigo apoyando mis decisiones, aceptando mis malcriadeces, cuidando de mí como si todavía fuera una niñita, por la confianza que siempre has depositado en mí. Gracias por enseñarme a vivir la vida y disfrutar de mi juventud, porque me has enseñado a ser tolerante y optimista, por ser más que mi madre mi gran amiga, te adoro.

A mi papá Fidel, por impulsarme siempre a luchar, a no renunciar a mis sueños, por mimarme, cuidarme, protegerme y por confiar siempre en mí. Por luchar siempre por mí y estar presente en cada uno de los pasos que doy. Gracias por llevar al barrio entero para que hablaran conmigo por teléfono para sentir que estaba en la casa. Siempre trataré de retribuirte todo el amor y el cariño incondicional que me has brindado. Te amo con toda mi alma.

A mi hermanita Lianet, por siempre porfiarme para ver cuánto sé, por enseñarme a superarme, por las peleas de hermanas que hemos tenido, por brindarme su amor, cariño, consejos, por cuidarnos una a la otra, por ser tan celosa conmigo. Gracias por

siempre estar a mi lado, siempre confiar en mí, y defenderme cuando alguien habla mal de mí, que esta tesis te sirva de ejemplo y de inspiración.

A mi sweetie porque sé que soportarme tanto tiempo no es fácil, a veces soy un poco inmadura y un poco peleona, pero él me ha entendido, me ha aguantado mis malcriadeces, siempre ha estado a mi lado apoyándome, junto a él he aprendido muchísimo, por ayudarme en todo el tiempo que hemos estado juntos, por complacerme en todo en lo que ha podido, a ti mi “boyfriencito” Yoan gracias por ser lo mejor que me ha pasado en la vida, porque nunca te olvidaré, por hacer de mi toda una mujer, por hacerme ver la vida desde otro punto de vista, por compartir conmigo momentos de alegría y tristeza, por todo lo que has hecho por mí. Te amo.

Agradezco a mi familia, en especial a mi segundo padre y tío Eldi y a mi tía y Arili por quererme como una hija y estar a mi lado en todo momento, por enseñarme desde pequeña a luchar por alcanzar mis metas, sin importar las críticas. A Yahima, Yariuska, Aliuska, a mi tía Cuca por estar siempre cuando las necesito, a mi primos y casi hermanos Katyleidis, Alma, Bella, Eldito, Maykel, Sandrito para que esta tesis les sirva de ejemplo.

A mis abuelos por inspirarme a convertirme en una profesional, porque hoy estarían orgullosos pero la vida no le dio la oportunidad de verme graduada, siempre los recordaré y amaré.

A mi familia matancera por acogerme en su casa como una más, por preocuparse y por cuidarme, por brindarme tanto amor y cariño. Gracias a todos.

Gracias a mi familia habanera por acogerme y brindarme todo su apoyo cuando lo necesitaba, sin esperar nada a cambio.

También quisiera agradecerle a todas las amistades que han estado junto a mí en todos estos años, a Analiet por soportarme de compañera de cuarto, a Rosalia y Alberto que siempre han estado apoyándome, confiando en mí, esperándome para ir al laboratorio o al comedor, a Elena que fue la primera persona que sin conocerme siempre confió en mí, cuando tenía un problema siempre corría hacia ella y estaba ahí para escucharme, porque siempre fuimos las tres inseparables, gracias por hacerme pasar la mejor etapa de mi vida junto a las “tres maravillas”, en fin, a todos mis compañeros de aula y apartamento durante estos 5 años y todos que de una forma u otra se preocuparon por mí.

A la editorial CubaLiteraria por ayudarme en la realización de este trabajo. A mi tutora Magdalys y su esposo Marcel por haber tenido tanta paciencia conmigo y haber dedicado tanto empeño en la realización de la presente investigación, que siempre estaban en la disposición de ayudarme y de no ser por los dos no hubiera podido realizar esta tesis.

Gracias a todos los profes que de una forma u otra me hicieron convertirme en una profesional desde la primaria hasta estos momentos. A los miembros del tribunal, por demostrar grato interés en el trabajo de diploma y al oponente por su preocupación por el mismo.

Tal vez sean muchas palabras y yo solo quiero decir GRACIAS por estar a mi lado!!!!

Dedicatoria

A mis padres Virgen y Fidel por haberme traído al mundo, por brindarme todo su amistad, amor y cariño sin esperar nada a cambio, por educarme, por los sabios consejos que siempre me da, y forjar buenos valores para poder ser hoy una persona de bien. A mi hermana Lianet por consentirme, mimarme, por estar a mi lado en todos momentos, por su sacrificio, dedicación, ternura, amor, por darme las fuerzas para seguir siempre hacia delante. A mi novio Yoan por sus buenos consejos y por darme el apoyo que nadie me podría dar en los momentos más difíciles.

Resumen

Los sistemas de recomendación son herramientas de software y técnicas encargadas de realizar determinadas sugerencias, ayudando a obtener la información que se necesita de acuerdo a las preferencias del usuario por un grupo de elementos con características específicas. Actualmente, en la editorial electrónica CubaLiteraria se desarrolla una tienda virtual dedicada a vender los libros publicados por el Instituto Cubano del Libro. Este sistema posee un catálogo en línea que permite la búsqueda de libros según el criterio de los usuarios, sin embargo carece de un servicio que permita agilizar los procesos de búsqueda. Para dar solución a este problema se concibe un sistema de recomendación que permite la sugerencia de libros en la tienda virtual. El desarrollo del módulo se divide en las fases propuestas por la metodología Programación Extrema, documentando cada paso y cada artefacto generado. Como resultado se obtiene un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la propuesta de solución.

Palabras clave: *catálogo en línea, tienda virtual, sistema de recomendación.*

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	4
1.1 Conceptos asociados al dominio del problema	4
1.2 Sistemas recomendadores	4
1.2.1 Generalidades de los sistemas recomendadores	5
1.2.2 Clasificación de los sistemas de recomendación	5
1.2.3 El problema de la recomendación	9
1.2.4 Fundamentación de la técnica algorítmica seleccionada	9
1.2.5 Algoritmos generación de recomendaciones	10
1.2.6 Entradas de los sistemas de recomendación	11
1.2.7 Estudio de sistemas similares	11
1.2.8 Desventajas de los sistemas recomendadores	13
1.2.9 Salidas de los sistemas de recomendación	14
1.2.10 Ingeniería de conocimiento	14
1.3 Metodologías de desarrollo de software	16
1.4 Herramientas y tecnologías a utilizar	19
1.4.1 Sistemas de gestión de contenidos	19
1.4.2 Lenguaje de programación	20
1.4.3 Sistema gestor de base de datos	21
1.4.4 Entorno de desarrollo integrado	22
1.4.5 Servidor web	22
Capítulo 2: Características del sistema	24
2.1 Objetivos vinculados al campo de acción	24
2.2 Usuarios relacionados con el sistema	24
2.3 Requisitos del sistema	25
2.4 Historias de usuario	25
2.5 Estimación de esfuerzos por historias de usuario	28
2.6 Plan de iteraciones	28
2.7 Plan de duración de las iteraciones	29
2.8 Plan de entregas	29
2.9 Prototipo no funcional de interfaz de usuario	30
2.10 Tarjetas contenido responsabilidad colaboración	31

Capítulo 3: Implementación y pruebas	34
3.1 Fase de implementación	34
3.1.2 Iteración 2	35
3.1.3 Iteración 3	35
3.2 Trabajo con Drupal	36
3.3 Diseño de la base de datos	37
3.4 Módulos en Drupal	39
3.4.1 Módulo desarrollado para el sistema.....	39
3.4.2 Instalación del módulo	40
3.5 Módulo recomendador	41
3.6 Técnicas para el filtrado de la información.....	42
3.7 Roles de usuario.....	45
3.8 Pruebas	45
3.8.1 Pruebas unitarias	46
3.8.2 Pruebas de aceptación	47
Conclusiones	50
Recomendaciones	51
Referencias bibliográficas	52
Glosario de términos	58
Anexo I.....	59
Anexo II.....	63

ÍNDICE DE TABLAS

TABLA 1. USUARIOS DEL SISTEMA	24
TABLA 2. HU PERFIL DE USUARIO	26
TABLA 3. HU GENERAR LAS RECOMENDACIONES.....	26
TABLA 4. HU MOSTRAR LIBROS RECOMENDADOS.	27
TABLA 5. HU MOSTRAR LIBROS MÁS VISTOS.	27
TABLA 6. ESTIMACIÓN DE ESFUERZOS POR HU.....	28
TABLA 7. DURACIÓN DE LAS ITERACIONES	29
TABLA 8. PLAN DE ENTREGAS	29
TABLA 9. TARJETA CRC GESTIONAR PERFIL DE USUARIO.....	31
TABLA 10. TARJETA CRC RECOMENDADOR DE LIBROS	32
TABLA 11. TARJETA CRC MOSTRAR LIBROS RECOMENDADOS.....	32
TABLA 12. TARJETA CRC MOSTRAR LIBROS MÁS VISTOS	32
TABLA 13. FUNCIONALIDADES EN LA ITERACIÓN 1	34
TABLA 14. TAREA DE INGENIERÍA PARA LA PRIMERA ITERACIÓN.....	35
TABLA 15. FUNCIONALIDAD ABORDADA EN LA SEGUNDA ITERACIÓN.....	35
TABLA 16. FUNCIONALIDAD ABORDADA EN LA TERCERA ITERACIÓN	35
TABLA 17. CASO DE PRUEBA DE ACEPTACIÓN.....	47
TABLA 18. TAREAS DE INGENIERÍA PARA LA PRIMERA ITERACIÓN.	59
TABLA 19. TAREAS DE INGENIERÍA PARA LA SEGUNDA ITERACIÓN.....	60
TABLA 20. TAREAS DE INGENIERÍA PARA LA TERCERA ITERACIÓN.	62
TABLA 21. CASO DE PRUEBA DE ACEPTACIÓN HU2_P2.....	64
TABLA 22. CASO DE PRUEBA DE ACEPTACIÓN HU3_P3.....	64
TABLA 23. CASO DE PRUEBA DE ACEPTACIÓN HU4_P4.....	65

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1. PIRÁMIDE DEL CONOCIMIENTO..... 15

ILUSTRACIÓN 2. DIFERENCIAS ENTRE LOS TIPOS DE METODOLOGÍAS..... 17

ILUSTRACIÓN 3. FASES DE XP 19

ILUSTRACIÓN 4. PROTOTIPO DE INTERFAZ NO FUNCIONAL DE USUARIO..... 31

ILUSTRACIÓN 5. UBICACIÓN DEL MÓDULO RECOMENDADOR..... 40

ILUSTRACIÓN 6. NO CONFORMIDADES SIGNIFICATIVAS, NO SIGNIFICATIVAS Y RECOMENDACIONES. 49

Introducción

El crecimiento en los últimos años de las Tecnologías de la Información y las Comunicaciones (TIC) ha revolucionado la concepción tradicional de la comunicación, el aprendizaje, la economía y el mundo en general.

Generalmente las empresas o instituciones necesitan promocionar sus productos, una de las formas para lograrlo sería la realización de una tienda virtual, la cual se considera un comercio convencional que usa como medio principal para realizar sus transacciones un sitio web de internet (1). Esta tienda podrá utilizarse en la red nacional o internacional sin tener la necesidad de contar con la presencia física de usuarios interesados en las ofertas del sitio. En la actualidad, estas tiendas son frecuentadas diariamente por las comodidades y ventajas que brindan, lo que permite que los clientes interesados en sus ofertas realicen sus transacciones y compras. Entre los elementos que conforman una tienda virtual se encuentran los catálogos donde se muestra el abanico de productos ofertados.

La editorial electrónica CubaLiteraria está desarrollando una tienda virtual que se dedica a vender los libros que son publicados por el Instituto Cubano del Libro. Este sistema posee un catálogo en línea que permite la visualización de libros, sin embargo carece de un servicio que permita agilizar los procesos de búsqueda.

Este sistema, al no contar con un método que filtre libros de relevancia para el usuario, hace que sea muy difícil encontrar con facilidad dicha información, pues no siempre los materiales relevantes están ubicados en primera posición, es por esto que en ocasiones, los usuarios abandonan la búsqueda sin tener resultado alguno.

De la situación antes expuesta se origina el siguiente **problema a resolver**: ¿Cómo facilitar la localización de libros relevantes para los usuarios del portal CubaLiteraria en la tienda virtual?

A partir del problema de investigación identificado se definió como **objeto de estudio**: las técnicas de filtrado de información.

El **campo de acción** se centra en los sistemas recomendadores para tiendas virtuales de libros.

En el marco de este trabajo se trazó como **objetivo general**: Desarrollar un módulo de recomendación para la tienda virtual de libros del portal CubaLiteraria que facilite a sus usuarios la localización de libros relevantes para ellos.

En correspondencia con el objetivo general se propone realizar los siguientes **objetivos específicos**:

- ✓ Realizar el estado del arte de los sistemas de recomendación.
- ✓ Definir el algoritmo que se utilizará para la recomendación del libro considerando las características de la tienda virtual.
- ✓ Realizar el análisis y diseño del módulo propuesto.
- ✓ Implementar el módulo de recomendación para la tienda virtual del portal CubaLiteraria.
- ✓ Validar el módulo desarrollado mediante pruebas de aceptación y unitarias.

Se tiene como **idea a defender** que el desarrollo de un módulo de recomendación para la tienda virtual de libros permitirá a los usuarios la localización de libros, teniendo en cuenta sus preferencias y comportamiento en el sistema.

Para la realización de esta investigación se utiliza la combinación dialéctica de los **métodos teóricos y empíricos**, los que permitieron develar la parte de la ciencia que está siendo objeto de estudio. Entre los primeros se emplean:

Histórico-lógico: Utilizado al analizar la evolución histórica de los sistemas de recomendación como una de las herramientas para concebir el sistema actual.

Analítico-sintético: Este método es empleado al analizar la documentación y elementos teóricos relacionados con las herramientas para recomendación en tiendas virtuales.

Se realizaron **Entrevistas** al cliente, usuarios y especialistas en el tema para determinar las funcionalidades a implementar para desarrollar el sistema deseado.

A continuación se muestra un resumen de las diferentes temáticas que se abordan en el presente trabajo de diploma.

Capítulo 1: Fundamentación Teórica

Este capítulo contiene la fundamentación teórica que soporta la investigación que permite develar la parte de la ciencia que está siendo estudiada enfatizando en el estudio de metodologías de desarrollo, en las tecnologías y lenguajes de programación que se utilizan en el desarrollo del módulo y el estado del arte de soluciones existentes.

Capítulo 2: Descripción de la propuesta de solución.

En este apartado se determinan los servicios que brindará el módulo recomendador, definiéndose las funcionalidades que debe cumplir, así como el diseño del mismo y se generan los artefactos que propone la metodología.

Capítulo 3: Implementación y pruebas.

Este último capítulo se realiza la implementación del módulo a desarrollar así como las pruebas para lograr que el módulo brinde las funcionalidades requeridas.

Capítulo 1: Fundamentación teórica

Introducción

El desarrollo progresivo de internet, aparejado a las nuevas tendencias de visualización de los contenidos en línea, crea una variedad extensa de productos que deben satisfacer las necesidades del usuario final. En este ámbito, los individuos se enfrentan a nuevos escenarios caracterizados por la existencia de una voluminosa cantidad de información, por lo que las recomendaciones de terceros son una opción a la que acuden para tratar de solucionar los problemas derivados de su limitado conocimiento, respecto a categorías o alternativas en el consumo de materiales. Para dar solución a esta problemática, mejorar el tratamiento de los clientes y satisfacer sus necesidades, surgen los sistemas de recomendación, encargados de brindar a los usuarios propuestas personalizadas de los contenidos o servicios ofertados.

1.1 Conceptos asociados al dominio del problema

Una tienda virtual se define como: "...un sitio web que vende productos o servicios y por lo general tiene un carro de compra en línea asociado a ella. Tienen distintas formas de pago para que el cliente pueda acceder sin problemas como el sistema de tarjetas de crédito o el pago contra entrega, este último es el pago al momento de entregar el producto en el domicilio del cliente, también algunos ocupan la transferencia bancaria."

1.2 Sistemas recomendadores

Según Ricci, los sistemas recomendadores "*son herramientas de software y técnicas que ofrecen determinadas sugerencias de empleo y utilidad para el usuario*". Un sistema de recomendación es aquel sistema que tiene como principal tarea seleccionar ciertos objetos de acuerdo a los requerimientos del usuario. Constituyen una técnica de filtrado de información, la cual presenta distintos tipos de temas o ítems de información al usuario, basándose en la predicción del "ranking" o ponderación que este le daría a un ítem (2) que el sistema aún no ha considerado; mediando, automatizando o soportando de esta forma, el proceso de realizar recomendaciones, además de utilizarse para predecir la valoración de los usuarios sobre elementos que aún no han evaluado (3). En definitiva los sistemas recomendadores son sistemas para ayudar a relacionar o emparejar a usuarios con los productos.

1.2.1 Generalidades de los sistemas recomendadores

Los tipos de datos usados por los sistemas recomendadores se agrupan en tres tipos de objetos: los ítems, los usuarios y las transacciones.

El término **ítem**, son los objetos que se recomiendan a los usuarios (productos, sitios web, personas, películas, música, libros, noticias, imágenes) por lo cual elementos como el diseño, la interfaz gráfica y el algoritmo usado para generar las recomendaciones, son personalizados para obtener sugerencias efectivas para dicho ítem (2). El análisis basado en los ítems recomienda al usuario productos parecidos a los que ha elegido o le gustan al visitante o cliente potencial.

La persona a la que se le va a realizar las recomendaciones es el usuario, el cual es otro tipo de dato de los sistemas recomendadores que permite ofrecer las recomendaciones luego de explorar toda la información explícita o implícita sobre este usuario (2).

A la interacción que existe entre el usuario y el sistema recomendador se le conoce por **transacciones**, donde la similitud entre ítems y usuarios se calcula mediante complejos algoritmos diseñados para éste fin. Cada empresa dedicada a estas actividades utiliza alguno basado en sus propios estudios o experiencias. La evaluación o calificación que una persona realiza sobre un ítem se denomina **ratings**, demostrando así la transacción.

1.2.2 Clasificación de los sistemas de recomendación

Los sistemas recomendadores se pueden clasificar de acuerdo a la creación de su perfil de usuario. Existen dos grandes tipos de sistemas de recomendaciones en base a la fuente de conocimiento usada para realizar la recomendación: basados en contenidos, que son aquellos que realizan la recomendación al usuario en base a la descripción de los productos (4) y los colaborativos, que son aquellos que utilizan valoraciones asociadas a los productos dadas por el usuario y por otros usuarios (5).

Otras técnicas han sido propuestas para desarrollar recomendaciones, por ejemplo: las basadas en conocimientos (6) y recomendaciones basadas en perfiles demográficos pero son menos usadas (7). Para mejorar el rendimiento de las distintas técnicas surgieron las recomendaciones híbridas. Estas recomendaciones surgen como combinación de otros enfoques (por ejemplo: basados en contenidos y colaborativos).

1.2.2.1 Sistemas basados en filtrado colaborativo

Un sistema basado en filtrado colaborativo “*recomienda elementos a los usuarios basados en las opiniones de otras personas*” (8), es decir, recomiendan objetos que son preferidos por otros usuarios de intereses similares. Los usuarios realizan sus valoraciones utilizando los ratings y se crea un perfil de usuario aproximado buscando las características similares en otros usuarios devolviendo un listado con las recomendaciones que contienen los ítems con un mayor nivel de evaluación y que no han sido visitados por el usuario al que se le realizará la recomendación.

Existen dos tipos de filtrados colaborativos, basados en el usuario y basados en el ítem: Los basados en los usuarios identifican a los usuarios similares, al usuario actual y predice la valoración que le daría a un ítem basándose en la que sus usuarios más cercanos han realizado. Los sistemas basados en el ítem por cada elemento determinado, calculan los elementos más similares. A partir del rating de los mismos entonces se predice la valoración para el usuario (9).

El resultado de un sistema de filtrado colaborativo no coincide automáticamente con las propias preferencias. A menos que se logre una buena diversidad de opiniones y la independencia, un punto de vista siempre dominará a otro en una comunidad. Al igual que en el escenario de recomendación personalizado, el usuario nuevo o un nuevo elemento siempre puede causar un problema, puesto que no habrá datos suficientes para que el filtrado colaborativo pueda trabajar. Con el fin de hacer recomendaciones precisas, el sistema debe primero aprender las preferencias del usuario con evaluaciones anteriores o realizando nuevas. Para un nuevo elemento añadido el sistema de filtrado colaborativo no sería capaz de recomendarlo hasta que sea evaluado por un número sustancial de usuarios.

1.2.2.2 Sistemas basados en datos demográficos

Los sistemas basados en datos demográficos son los menos usados, se basan solo en el cálculo de los usuarios más cercanos usando información demográfica, como la edad y el sexo. Los beneficios que posee este sistema frente a los sistemas basados en filtrado colaborativo consisten en no requerir de un largo historial de valoraciones para hacer recomendaciones a un usuario.

1.2.2.3 Sistemas basados en el conocimiento

Los sistemas basados en el conocimiento se basan en inferencias sobre las necesidades de los usuarios y sus preferencias para realizar las sugerencias. Para ello se información sobre cómo un

ítem específico responde a una necesidad en particular del usuario y por tanto, la razón sobre la relación entre la necesidad y una posible recomendación (2).

1.2.2.4 Sistemas de recomendación híbridos

Los enfoques básicos de la recomendación anteriormente mencionados han sido utilizados de manera exitosa en disímiles campos y escenarios. Pese a esa difusión, de forma general cada uno de ellos presentan varias desventajas relacionadas principalmente con la necesidad de un gran número de usuarios adheridos al sistema para un correcto funcionamiento del mismo y con la tendencia de una sobre-especialización que va aumentando según aumenta la utilización de la aplicación por parte del usuario.

Con el objetivo de mitigar estas desventajas, una opción muy utilizada es la de combinar la recomendación basada en contenido con el filtrado colaborativo, conformando un sistema de recomendación con características híbridas. A través de estos, la construcción de preferencias se basa tanto en los intereses de usuarios, como en los intereses de usuarios similares, al coincidir así con los dos tipos específicos de sistemas de recomendación antes mencionados.

Varios autores han demostrado cómo a través de este enfoque la calidad de las recomendaciones aumentan considerablemente (10), constituyendo de esta forma una alternativa a tener en cuenta a la hora de diseñar un sistema de este tipo.

El autor Robin Burke define los sistemas recomendadores híbridos de la siguiente manera: "...combinan dos o más técnicas de recomendación para obtener mayor eficacia sin los inconvenientes de cada sistema hibridado" (11).

1.2.2.5 Recomendaciones basadas en contenido

Las recomendaciones basadas en contenido son aquellas que realizan la recomendación en base a la descripción de los productos a recomendar; intentan recomendar elementos que son similares a aquellos que el usuario previamente valoró en el pasado. Se enfocan en algoritmos de aprendizaje de preferencias de usuarios y filtran nuevos elementos, teniendo en cuenta los que más satisfacen las preferencias de estos.

En estos sistemas los ítems se definen según sus características y el perfil de usuario se basa en cómo el usuario valora esas características.

A continuación se muestran los pasos a seguir en las recomendaciones basadas en contenido:

1. Extraer las características de los objetos.
2. Comparar los objetos con el perfil del usuario para identificar las preferencias de este sobre dicho objeto.
3. Recomendar objetos similares en su contenido a los que forman parte de su perfil.

Entre las ventajas que posee este tipo de recomendador se puede destacar como una de las más importantes, que el sistema puede Generar las recomendaciones sin la necesidad de contar con un historial previo, por lo que permite realizar predicciones independientemente del historial del usuario. A continuación se presentan otras de las ventajas de los sistemas recomendadores basados en contenido:

- ✓ Puede Generar las recomendaciones por el contenido y no por opiniones subjetivas de otros usuarios.
- ✓ Puede generar explicaciones sobre la recomendación que hizo en base al historial del usuario.
- ✓ El modelado de la información está presente en las características del contenido y no necesitan proveerlas otros usuarios.

Como principal desventaja en este tipo de sistema se tiene la sobre-especialización, donde el usuario está limitado a que le recomienden ítems similares a los que recomendó, consiste en que las recomendaciones resultantes suelen ser siempre muy similares, dado que se basa en la misma información y que al depender del dominio de los objetos a recomendar, es difícil extraer buena información de estos para realizar un cálculo falible (12).

Después de realizar el análisis de algunos de los tipos de sistemas de recomendación teniendo en cuenta las características, ventajas y desventajas de los mismos, se puede concluir que los elementos comunes para estos sistemas son: la necesidad de contar con un perfil de usuario que almacene sus preferencias, una descripción del elemento que posiblemente sea recomendado y un sistema que determine lo cerca que está dicho elemento del usuario, con el fin de poder recomendárselo. Se decide utilizar un sistema de recomendación basado en contenido debido a que su principal ventaja radica en realizar recomendaciones sin la necesidad de contar con un historial previo, por lo cual se pueden efectuar predicciones independientemente del historial del usuario. No se necesita de gran cantidad de personas que utilicen el sistema para determinar a qué

grupo de usuarios con características similares debe pertenecer un nuevo individuo, cuestión en la que se basa el filtrado colaborativo. Además la recomendación no está definida para que un usuario realice recomendaciones a otro, ni teniendo en cuenta la edad del mismo, particularidades de los sistemas de soporte a la recomendación y los basados en datos demográficos, respectivamente.

1.2.3 El problema de la recomendación

El principio de funcionamiento de los sistemas de recomendación está resumido en el problema de la recomendación, el cual se formula de acuerdo a la ecuación 1, de la siguiente manera: sea C el conjunto de todos los usuarios y sea S el conjunto de todos los ítems posibles que se pueden recomendar, como libros, películas o restaurantes, sin importar la cardinalidad de ambos conjuntos. Sea u una función que mide la utilidad del ítem s al usuario c , o sea $u: C \times S \rightarrow R$, donde R es un orden total (enteros no negativos o números reales en un determinado rango). Con estos elementos se desea para cada usuario $c \in C$, aquel ítem $s \in S$ que maximice la utilidad al usuario (13).

$$\forall c \in C, S'_c = \operatorname{argmax}_{s \in S} u(c,s)$$

Ecuación 1. Problema de la recomendación.

1.2.4 Fundamentación de la técnica algorítmica seleccionada

La obtención de datos es necesaria en los sistemas de recomendación basados en contenidos pero no sirve de nada sin un mecanismo o algoritmo que procese y extraiga las recomendaciones. Estos algoritmos crean una función que modela los intereses de los usuarios. Por consiguiente, dado un modelo de usuario y un ítem nuevo, la función predice si el ítem será del agrado del usuario. A continuación se presentan diferentes formas de inferir en el conocimiento almacenado en el sistema:

- ✓ Personalización del usuario: El sistema proporciona una interfaz donde el usuario puede aportar las preferencias de sus gustos. Una vez que el usuario ha proporcionado sus preferencias, el sistema simplemente busca los elementos que encajan con los parámetros indicados por el usuario. El método implica que el usuario debe implicarse y rellenar un formulario para que el sistema pueda realizar las recomendaciones.

- ✓ Reglas de asociación: Se intenta descubrir relaciones entre los ítems, que luego pueden ser utilizadas para recomendación. Los ítems se comparan en base al comportamiento pasado de los usuarios respecto a ellos. Por ejemplo: la compra de cereales puede relacionarse con la compra de leche.
- ✓ *Feedback* relevante: Método que propone obtener información sobre los ítems recomendados al usuario cuando éste ha realizado una búsqueda. Por ejemplo: páginas web que ofrecen ayuda o soporte de sus programas, piden al usuario que valore la información aportada por el sistema conforme a lo que estaba buscando. Mediante la apreciación del usuario, el sistema conoce qué información ha sido útil para la búsqueda que se ha realizado y la tiene en cuenta para posteriores búsquedas, iguales o relacionadas, a la valorada por el usuario.
- ✓ *Clustering*: Dado que muchas veces es una tarea costosa tanto en tiempo de cálculo como de requisitos, el obtener una recomendación para un individuo en concreto, una alternativa es agrupar usuarios en categorías en base a comportamientos pasados de otros consumidores. Cada grupo o *cluster* tiene preferencias que son típicas en base a sus miembros. Los usuarios dentro de cada grupo recibirán recomendaciones calculadas a nivel grupal. La ventaja del *clustering* es que trabaja sobre los datos agregados, es decir, más eficientes. Esta técnica se suele usar como un primer paso para reducir las búsquedas del filtrado colaborativo. Por otro lado las recomendaciones (a nivel grupal) son menos relevantes o precisas que en el filtrado colaborativo (a nivel individual).

Dadas las características anteriormente descritas, se emplea para construir la solución, un sistema basado en reglas de asociación y la personalización del usuario debido a que se lograría realizar una efectiva inferencia de conocimiento debido a la estructura del sistema actual.

1.2.5 Algoritmos generación de recomendaciones

Existen dos formas de generar recomendaciones, los algoritmos son:

- **Predicción:** un valor numérico $P(a, j)$ que predice la preferencia del usuario activo (U_a), hacia el ítem (I_j), no usado aun por U_a (14).

- **Recomendación:** una lista de los primeros ítems (N) que el usuario U_a posiblemente prefiere, no usados aun por U_a y se llaman los top-N (14).
- **Basados en el usuario** utilizan estadísticas para buscar los usuarios más similares al U_a y para producir una predicción o recomendación (top-N) combinan las preferencias de los usuarios similares para U_a (15).
- **Basados en los ítems** deducen que determinado usuario esté interesado en ítems similares a los que ha visitado, calculan la similitud con el ítem objetivo y seleccionan los más similares. También pueden ser basados mediante la creación del perfil de usuario (15).

Para la realización de la investigación se decide seleccionar el algoritmo basado en ítems usando el algoritmo de recomendación pues responde a las necesidades de la investigación.

1.2.6 Entradas de los sistemas de recomendación

En las entradas del sistema de recomendación la información sobre los usuarios puede venir dada de dos formas que no son mutuamente excluyentes. Una de estas formas es por extensión, la cual se refiere a información que se tenga sobre las experiencias pasadas del usuario con respecto a los elementos encontrados, es lo que también se conoce como navegación implícita, pues el usuario no es consciente de estos seguimientos. La otra forma es la información expresada intencionalmente, también se le llama navegación explícita y consiste en que el usuario expresa intencionalmente sus preferencias, esta forma se evidencia en los datos recogidos en el perfil de usuario (10).

Para el sistema de recomendación de la tienda virtual puede adoptarse la obtención de información tanto implícita como explícita. Para ello se puede hacer uso de los datos del historial de navegación del usuario y del perfil del mismo con que cuenta la tienda virtual.

1.2.7 Estudio de sistemas similares

Los sistemas de recomendación tienen disímiles aplicaciones en diferentes áreas de la sociedad, dentro de estas se puede resaltar su amplia y estratégica utilización en el marco del *e-commerce*. En función de facilitar la concepción del sistema de recomendación se realiza un estudio de soluciones similares existentes para determinar elementos que presentan dichas herramientas y que pueden ser útiles en el desarrollo de la solución propuesta.

Hoy día con el desarrollo y la competitividad del comercio es de vital importancia para las grandes compañías conocer quiénes son sus clientes, cuáles son sus preferencias y cómo evalúan los productos existentes. Trabajar en base al criterio de los usuarios finales constituye una de las llaves del éxito de muchas empresas con presencia en Internet. Muchos sitios web de comercio electrónico han comenzado a retroalimentarse de sus usuarios de manera directa, al permitirle compartir opiniones y evaluar de manera cuantitativa los productos que compran.

Unos de los ejemplos de aplicaciones que poseen sistemas de recomendación en este campo lo constituye *Amazon*, sitio de venta de productos mencionado anteriormente, cuya sección de venta de libros contiene implícita varias funcionalidades basadas en esta filosofía, por ejemplo: *Customers who bought*, mostrada en la página de información de cada libro dentro del catálogo y que contiene dos listas de recomendaciones, una con los libros comprados por aquellos clientes que compran el libro y otra con los autores leídos por estos mismos clientes.

Otras aplicaciones populares en Internet que implementan sistemas de recomendaciones son: *CDNOW*, portal concebido para la venta de *CD* de música, *Levis*, creada con el objetivo de dar soporte a los clientes de la ropa de la marca homónima. Todos estos sitios tienen en común un alto nivel de aceptación por parte de los clientes de la red global.

Sistemas de recomendación a nivel nacional

El estudio realizado está enfocado en los diferentes sistemas recomendadores, en búsqueda de obtener las mejores características que ellos brindan. El empleo de técnicas de recomendación en la búsqueda de los contenidos educativos que son de interés para el usuario mejoraría el proceso de enseñanza. Luego de consultar la bibliografía se concluye que la mayoría de los sistemas recomendadores fueron desarrollados en la UCI.

Raciel Yera muestra la implementación de un sistema recomendador de ejercicios para jurados online de programación (16). Recomienda paquetes de actividades a realizar por los estudiantes, basados en su rendimiento y progreso actual, utilizando un sistema basado en filtrado colaborativo. Otro de los trabajos que pueden citarse es Yordi, que desarrolla un sistema recomendador de cuestionarios interactivos para los estudiantes que utilizan la colección *El Navegante*, mejorando así la consolidación del conocimiento, el sistema permite a los estudiantes atender aspectos en los

que presentan mayores dificultades y así poder erradicar las mismas (17). Por último, se analiza el trabajo que tiene como objetivo desarrollar un sistema recomendador de noticias basado en contenido, que mejore la personalización de los contenidos presentes en el Portal Octavitos (18). Los estudios realizados permitieron adquirir información sobre los sistemas de recomendación más relevantes en la actualidad. Mediante el análisis de las principales características de estos sistemas se llegó a la conclusión de que no existe ningún sistema de recomendación basado en contenido que cumpla con las necesidades de la tienda virtual del portal CubaLiteraria, debido a que cada uno de ellos está concebido para fines específicos de las entidades para las que fueron desarrollados, por lo que no pueden ser utilizados. A pesar de esto, es válido aclarar que algunos de estos sistemas poseen características como son: la creación de perfiles de usuarios y utilización de algoritmos que pudieran ser utilizadas en el desarrollo del sistema.

1.2.8 Desventajas de los sistemas recomendadores

En los sistemas recomendadores existen problemas que se deben considerar para su diseño, Macmanus plantea algunos indicios de los problemas que deben superar las organizaciones para construir sistemas recomendadores eficaces (19):

- ✓ **Carencia de información:** Para realizar una correcta recomendación se necesita gran información, ya sea por parte de los usuarios o de los ítems. Los sistemas recomendadores tienen mayor exactitud de sus predicciones cuando cuentan con mayor cantidad de información disponible.
- ✓ **Información cambiante:** El problema de este tipo radica cuando un determinado ítem está de moda y recibe gran cantidad de valoraciones. Cuando este período termina, los usuarios no desean que se les recomiende más; sin embargo, el sistema de recomendación los sigue recomendando pues tiene un rating elevado.
- ✓ **Ítems impredecibles:** En ocasiones existen ítems difíciles de recomendar pues a los usuarios no les resulta de interés.

- ✓ Cambio de preferencias de usuario: En ocasiones los usuarios buscan recomendaciones para ellos mismos, pero pueden buscar productos para otros usuarios con preferencias y gustos diferentes a los suyos.

1.2.9 Salidas de los sistemas de recomendación

Las salidas de los sistemas de recomendación están enfocadas en mostrar las recomendaciones finales al usuario. Pueden utilizarse diferentes representaciones, por ejemplo: una lista de sugerencias, predicciones del grado de satisfacción asociadas a cada recomendación o votaciones realizadas a cada elemento (10). Para mostrar las recomendaciones en la tienda se presentarán en la portada principal una lista de sugerencias con las medias de mayor preferencia para el usuario.

1.2.10 Ingeniería de conocimiento

La ingeniería del conocimiento forma parte de la Inteligencia Artificial y su fin es el diseño y desarrollo de sistemas expertos. Para ello, se apoya en metodologías instruccionales y en las ciencias de la computación y de la información, intentando representar el conocimiento y razonamiento humanos en un determinado dominio, dentro de un sistema artificial. El trabajo de los ingenieros del conocimiento consiste en extraer el conocimiento de los expertos humanos en una determinada área y en codificar dicho conocimiento de manera que pueda ser procesado por un sistema.

La ingeniería de conocimiento es una disciplina de gran utilidad para las organizaciones en los ámbitos de la investigación, el desarrollo y la innovación ya que facilitan y mejoran el funcionamiento de sistemas en la toma de decisiones. En la base están los datos y su reelaboración mediante sistemas de conocimiento conduce a la creación de nuevo conocimiento y su representación cartográfica que ayudan a la toma de decisiones.

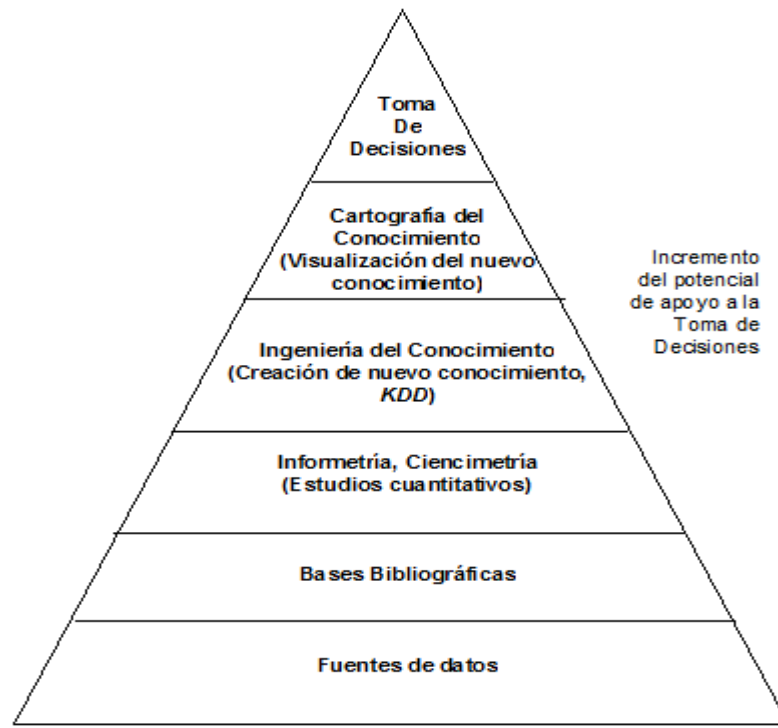


ILUSTRACIÓN 1. PIRÁMIDE DEL CONOCIMIENTO.

Uno de los métodos empleados en la ingeniería del conocimiento es el análisis de palabras asociadas que se basa en “leer” los documentos y generar redes socio-cognitivas a partir de la asociación de las palabras que hay en los documentos. Necesariamente este procedimiento debe llevarse a cabo mediante sistemas informáticos con software específicos. El resultado es lo que se denomina bases de conocimiento en oposición a bases de datos, que tan solo contienen datos.

El punto clave del desarrollo de un sistema basado en el conocimiento es el momento de traspasar el conocimiento que posee el experto a un sistema real. En este proceso no sólo se han de captar los elementos que componen el dominio del experto, sino que también se han de adquirir las metodologías de resolución que utilizan éstos. Este trabajo de extracción del conocimiento se realiza durante la interacción entre dos personajes, el ingeniero del conocimiento (IC) (persona que conoce el formalismo de representación que utilizará el SBC) y el experto (persona que posee el conocimiento, pero que no tiene por qué usar un formalismo para representarlo).

Durante las entrevistas entre el IC y el experto, el primero ha de ayudar a sistematizar el conocimiento del experto, consiguiendo que vaya explicitando las diferentes técnicas que utiliza para resolver los problemas de su dominio, de manera que se puedan representar en un formalismo computable.

Extracción del conocimiento

Esta metodología de extracción del conocimiento es bastante lenta. Varias son las dificultades que dan una producción tan baja a esta metodología:

- ✓ La naturaleza especializada del dominio hace que el IC deba aprender unas nociones básicas para que pueda establecerse una comunicación (vocabulario básico, elementos que intervienen en el dominio, formalismos que utilizan los expertos, entre otros).
- ✓ Los expertos se encuentran más cómodos pensando en términos de ejemplos típicos que razonando en términos generales, que son de los que realmente se podría hacer una mejor abstracción.
- ✓ La búsqueda de un formalismo de representación que se adapte adecuadamente al problema y que sea fácil de interpretar y adoptar por el experto. Este formalismo ha de ser susceptible de ser transformado en algo computable.
- ✓ Por lo general, a los expertos les es muy difícil explicitar los pasos que utilizan para resolver los problemas. Es la que se ha denominado paradoja del experto. Cuanta más experiencia, menos explícitos son los razonamientos del experto y más ocultos los métodos de resolución.

Con todas estas circunstancias, se puede observar que la auténtica dificultad de la adquisición del conocimiento estriba en descubrir los métodos mediante los que se usa el conocimiento en la resolución del problema y no tanto en la adquisición del conocimiento estático del problema, por este motivo no se utilizará para el desarrollo de esta investigación.

1.3 Metodologías de desarrollo de software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental para guiar el desarrollo de un

producto software, pero los requisitos de un software a otro son tan variados y cambiantes que ha dado lugar a que exista una gran variedad de metodología (20). Existen metodologías ágiles y tradicionales, a continuación se muestran sus diferencias:

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

ILUSTRACIÓN 2. DIFERENCIAS ENTRE LOS TIPOS DE METODOLOGÍAS

Para el desarrollo de la presente investigación se analizaron las metodologías ágiles SCRUM¹ y programación extrema (XP) debido a que la información que generan estas metodología se ajusta al entorno y condiciones del proyecto y tiene el potencial de acelerar el tiempo de desarrollo, ambas fomentan la interacción sistemática entre el cliente y el equipo de desarrollo y están suficientemente

¹ SCRUM es una metodología ágil principalmente indicada para proyectos con un rápido cambio de requisitos.

documentadas, características que son muy importantes para el desarrollo del presente trabajo, debido al intercambio de información que se realiza entre el cliente y el desarrollador.

SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints* con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Extreme Programming

Una de las prácticas más significativas que posee XP es que simplifica el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que no se necesiten, ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código (21).

Existen disimiles autores que definen el ciclo de vida de XP, definiendo sus fases, la autora de la investigación emplea la bibliografía de José Carlos Carvajal, donde define que está compuesto de seis fases (ver Ilustración 1): *Exploración*: donde los clientes escriben las historias de usuario (HU) describiendo las funcionalidades que serán añadidas al sistema, con actualizaciones regulares en caso de modificación; *Planificación*: donde se establece la prioridad de las HU y se realiza una estimación temporal para una primera entrega del proyecto; *Iteraciones*: donde se realizan iteraciones continuas decidiendo las historias de usuario a implementar, así como las pruebas funcionales ejecutadas al final con una retroalimentación continua; *Producción*: donde se realizan pruebas de rendimiento y funcionamiento que son necesarias antes de entregar el producto; realizando una pequeña entrega; *Mantenimiento*: donde se actualiza el sistema una vez entregada la primera versión, el sistema se debe mantenerse en mantenimiento mientras existan iteraciones en la fase de producción; *Cierre del proyecto*: es donde se realiza la entrega final cumpliendo las necesidades del cliente y documentando todos los pasos (22).

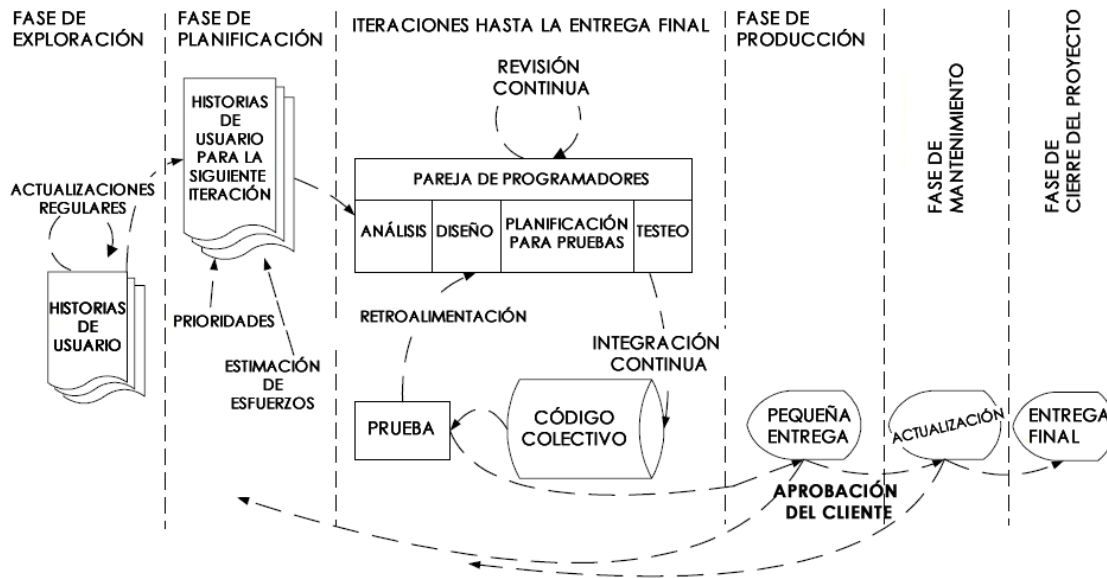


ILUSTRACIÓN 3. FASES DE XP

1.4 Herramientas y tecnologías a utilizar

Para el desarrollo del sistema de recomendación es de vital importancia la correcta selección de las herramientas y tecnologías a utilizar para garantizar un óptimo funcionamiento del sistema.

1.4.1 Sistemas de gestión de contenidos

El sistema de gestión de contenidos (CMS²) es: "...un software que se utiliza principalmente para facilitar la gestión de la web..." (23). Entre los CMS genéricos se encuentra Drupal, en el cual se basó el desarrollo de la tienda virtual de libros y que por su flexibilidad y estructura modular es considerado como potencial para continuar agregándole funcionalidades.

Drupal 7.15

Drupal está desarrollado con el lenguaje de programación PHP y utiliza maquetación con hojas de estilo CSS para la presentación de los contenidos, con lo que es posible construir sitios web totalmente accesibles (24). La tienda virtual de libros está desarrollada con el CMS Drupal, el módulo a desarrollar debe cumplir con la misma característica para facilitar su integración. El

² Content Management System, CMS por sus siglas en inglés.

módulo de recomendación estará desarrollado para el núcleo de Drupal independientemente de la versión, lo que establece que se utilizarán las funcionalidades encapsuladas por Drupal para dicha versión del núcleo. En la tienda virtual web oficial de Drupal (disponible en <http://drupal.org>) es posible encontrar módulos y temas que proporcionan nuevas características y que transforman la apariencia del sitio.

1.4.2 Lenguaje de programación

La aplicación a desarrollar es, por definición, una aplicación web. En esta sección se especifican las características más importantes de los lenguajes de programación utilizados para el desarrollo de la solución propuesta.

Lenguajes y tecnologías del lado del cliente

Las tecnologías del lado del cliente son las páginas dinámicas que procesa el usuario. En estas páginas toda la carga del procesamiento de los efectos y funcionalidades la soporta el navegador (25).

La programación web brinda la posibilidad de crear sitios dinámicos como periódicos digitales o tiendas virtuales, generando los contenidos del sitio a través de una base de datos, utilizando alguno de los diferentes lenguajes de programación para Web que existen hoy en día. Estos lenguajes se clasifican en dos partes fundamentales:

CSS 3

Las hojas de estilo en cascada, conocidos como CSS se define según Rivera como "...un lenguaje de hojas de estilo creado para controlar la presentación de los documentos" (26) , para seguir la línea de diseño de la tienda virtual de libros se decide seleccionar CSS para mejorar el diseño visual de la solución si fuera necesario, separar contenidos de presentación y realizar cambios a múltiples elementos dentro del código, agilizando el proceso de cambios.

JavaScript 1.5

JavaScript es un lenguaje de programación interpretado usado para crear páginas dinámicas, las variables no necesitan ser introducidas antes de su uso y los tipos de variables se resuelven dinámicamente durante su ejecución. Se trata de un lenguaje de programación del lado del cliente,

porque es el navegador el que soporta la carga de procesamiento, el código JavaScript que se encuentra dentro de las páginas web puede ser interpretado por todos los navegadores. Es un lenguaje de alto nivel, multiplataforma y no necesita compilación. Está basado en objetos, admite la programación estructurada y maneja la mayoría de los eventos que se pueden producir sobre la página web. En Drupal 7 se han agregado funcionalidades para facilitar el desarrollo con JavaScript alcanzando mejoras en la interfaz administrativa y una mayor interacción entre este lenguaje y el núcleo de Drupal, lo que permite la creación de páginas más dinámicas con funcionalidades nuevas y modernas que se esperan en los sitios web de la actualidad.

Lenguajes del lado del servidor

Se analiza el lenguaje PHP³ pues la tienda virtual de libros está desarrollada con Drupal, lo cual representa un factor importante en la selección de este lenguaje para el desarrollo del módulo, además, posibilita mantener una compatibilidad con los demás módulos que utiliza la tienda virtual.

PHP 5.3.5

PHP es un lenguaje diseñado para la creación de páginas web dinámicas, y con acceso a información almacenada en base de datos. Entre sus ventajas se destaca que consume pocos recursos, por lo que se ejecuta con rapidez y no tiende a ralentizar el resto de los procesos del sistema operativo (27). Otra ventaja es su gran capacidad de conectividad, al utilizar un sistema de extensiones modular hacia interfaces de variado tipo. Entre sus desventajas se pueden mencionar que para la realización de proyectos complejos requiere un alto nivel de experiencia (28).

1.4.3 Sistema gestor de base de datos

Un SGBD⁴ es un sistema de software que permite procesar, describir, administrar y recuperar los datos almacenados en una base de datos (29). Estos sistemas proporcionan funcionalidades, procedimientos y lenguajes que permiten a los usuarios realizar tareas con datos, garantizando la seguridad de los mismos. Se analiza las ventajas que brinda el SGBD MySQL 5.5, que soporta el almacenamiento de datos en la tienda virtual.

³ Hipertext Preprocessor, PHP por sus siglas en inglés.

⁴ Sistema Gestor de Base de Datos.

MySQL 5.5

MySQL en su versión 5.5 es un SGBD multiusuario, que cuenta con mejoras constantes por parte de desarrolladores a nivel mundial. Es un programa que almacena datos variados convirtiéndose en una solución altamente viable para la administración de datos (30). Por ser una herramienta pública y multiplataforma, la tienda virtual utiliza este gestor y es necesario mantener la compatibilidad para la próxima integración del módulo con la misma, se decide utilizar MySQL como gestor de base de datos en su versión 5.5.

1.4.4 Entorno de desarrollo integrado

Entorno de Desarrollo Integrado (IDE⁵) es un programa informático que agrupa diferentes herramientas de programación que facilitan la tarea al programador y se obtiene mayor rapidez en el desarrollo.

NetBeans 7.3

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas, está escrito en Java, pero puede servir para cualquier otro lenguaje de programación (31). Existe además un número importante de módulos para extender el IDE NetBeans, además es un producto libre y gratuito sin restricciones de uso. Se selecciona NetBeans en su versión 7.3 pues es compatible con lenguaje en el cual ha sido desarrollada la tienda virtual de libros, así como los módulos que utiliza.

1.4.5 Servidor web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Según Travis, Drupal necesita un servidor web para ejecutar código en PHP. Los servidores que utiliza para esto son Apache y el servidor IIS (27), teniendo en cuenta que la tienda virtual de libros se encuentra alojada en un servidor web Apache, se decide usar esta herramienta para su desarrollo.

⁵ Por sus siglas en inglés, IDE, Integrated Development Environment.

Apache 2.2.22

Para el desarrollo de la propuesta de solución se seleccionó el servidor web Apache en su versión 2.2.12, por ser una herramienta libre y brindar más seguridad que IIS. Apache funciona en una multitud de SO, mientras que IIS solo puede ser ejecutado en Windows. Además IIS es un software propiedad de Windows, lo que dificulta la obtención de la licencia para su uso, mientras que Apache es un software libre, facilitando su uso a toda la comunidad. Apache es el servidor web hecho por excelencia, su configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es un servidor web de código libre por lo que permite realizar cualquier cambio en el código fuente (32).

A continuación se muestran las principales características de este servidor web:

- ✓ Corre en una multitud de Sistemas Operativos (SO), lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código fuente abierto.
- ✓ Es un servidor altamente configurable de diseño modular.
- ✓ Trabaja con gran cantidad de lenguajes como Perl, PHP y otros lenguajes de script.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Conclusiones

En este capítulo, apoyado en los métodos de la investigación científicos definidos se construyó el marco conceptual que soporta la investigación y se adquirió el conocimiento necesario sobre las metodologías, herramientas y lenguajes, seleccionándose las adecuadas para el cumplimiento del objetivo general propuesto. Se analizaron las diferentes clasificaciones de los sistemas recomendadores basándose principalmente en la forma de procesar la información.

Capítulo 2: Características del sistema

Introducción

La metodología XP propone técnicas importantes de diseño entre ellas las historias de usuario y las tarjetas CRC que facilitan la comprensión del sistema a desarrollar y por esto se realizaron en el siguiente capítulo.

2.1 Objetivos vinculados al campo de acción

En la presente investigación se propone el desarrollo de un módulo de recomendación que tendrá el objetivo de brindar mayor elección, divulgación y acceso a los libros en venta, ofreciendo a los usuarios los libros de su preferencia. La solución permitirá a los usuarios seleccionar los géneros de su preferencia mediante un formulario, esto permite extraer la información necesaria para realizar la recomendación.

2.2 Usuarios relacionados con el sistema

Se define como usuario relacionado al sistema todo aquel que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellos que participan en ellos pero no obtienen ningún resultado de valor. En la presente investigación se definieron los siguientes usuarios.

TABLA 1. USUARIOS DEL SISTEMA

Usuarios	Justificación
Anónimo	Todos los usuarios deben autenticarse para acceder a las funcionalidades implementadas por el módulo. Este usuario puede navegar por el sistema sin obtener ningún privilegio.
Usuario autenticado	Persona que tendrá acceso al sistema como usuario estándar, teniendo la posibilidad de recibir recomendaciones.
Administrador	Los administradores disponen de posibilidades ilimitadas para ejecutar todas las funciones administrativas de la tienda y el módulo.

2.3 Requisitos del sistema

Mediante reuniones y encuentros entre el cliente y la autora del presente trabajo se definen características que debe poseer el módulo para cumplir con los objetivos trazados y satisfacer las necesidades de los usuarios:

- ✓ Mostrar un formulario con los géneros de libros existentes en el sistema.
- ✓ Realizar encuestas para determinar la aceptación de la recomendación.
- ✓ Realizar enlaces a otros sitios de interés vinculados a la tienda virtual.
- ✓ Proponer al usuario libros que no han sido vistos por él.
- ✓ Brindar un bloque con los libros más visitados por los usuarios en la tienda.

2.4 Historias de usuario

Las historias de usuarios son la forma en que se especifican en XP los requisitos del sistema, las mismas no deben ser descritas en más de tres líneas e idealmente es el cliente quien las redacta y prioriza por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica (33). Luego de esto se le suma un tiempo estimado de desarrollo que lo define el desarrollador. Deben describirse con claridad, de manera sencilla y concreta para su mejor entendimiento (34). A continuación se muestra una HU donde se observan sus principales elementos y la explicación de los mismos:

Historia de Usuario	
No: Es el número que se asigna a la HU.	Nombre: Atributo que contiene el nombre de la HU.
Usuario: El usuario que interactúa o protagoniza la HU.	

Prioridad en el negocio: El nivel de prioridad de la HU en el negocio.	Puntos Estimados: Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como cinco días hábiles. Cuando el valor de dicho atributo es de 0.5 equivale a dos días y medio de trabajo, lo que se traduce en veinte horas.
Riesgo de desarrollo: El nivel de riesgo en caso de no realizarse la HU.	Iteración asignada: Iteración en la que se implementará la HU.
Descripción: Describe lo que realizará la HU.	
Observaciones: Propone información adicional para hacer más entendible la HU	

Se muestran las HU del sistema a implementar:

TABLA 2. HU PERFIL DE USUARIO

Historia de Usuario	
No: 1 Usuario: Usuario autenticado.	Nombre: Perfil de usuario.
Prioridad en el negocio: Alta	Puntos Estimados: 1.5
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: El usuario especifica los géneros preferidos para que se le puedan recomendar.	

TABLA 3. HU GENERAR LAS RECOMENDACIONES.

Historia de Usuario	
No: 2	Nombre: Generar las recomendaciones.
Usuario: Administrador	

Prioridad en el negocio: Alta	Puntos Estimados: 3
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Permite recomendar a los usuarios los libros que hayan llenado el perfil de usuario, también recomienda los libros mediante la navegación en el sistema.	

TABLA 4. HU MOSTRAR LIBROS RECOMENDADOS.

Historia de Usuario	
No: 3	Nombre: Mostrar libros recomendados.
Usuario: Usuario autenticado.	
Prioridad en el negocio: Media	Puntos Estimados: 2.5
Riesgo de desarrollo: Alta	Iteración asignada: 2
Descripción: El usuario podrá ver los libros que recomendará el sistema teniendo en cuenta las decisiones de este en su perfil de usuario.	

TABLA 5. HU MOSTRAR LIBROS MÁS VISTOS.

Historia de Usuario	
No: 4	Nombre: Mostrar libros más vistos.
Usuario: Usuario autenticado	
Prioridad en el negocio: Baja	Puntos Estimados: 1.5
Riesgo de desarrollo: Media	Iteración asignada: 3
Descripción: El usuario podrá ver los libros con más visitas en la tienda.	

2.5 Estimación de esfuerzos por historias de usuario

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar todas las historias de usuario siendo uno de los artefactos generados en esta fase. Una vez terminado esto, se procede a organizarlas en las iteraciones correspondientes, en dependencia de la prioridad especificada por el cliente y del tiempo de desarrollo de cada una (35).

TABLA 6. ESTIMACIÓN DE ESFUERZOS POR HU

No	Historia de Usuario	Estimación (semanas)
1	Perfil de usuario	1.5
2	Generar las recomendaciones	3
3	Mostrar libros recomendados	2.5
4	Mostrar libros más vistos	1.5

2.6 Plan de iteraciones

Después de identificar las historias de usuarios y realizar una previa estimación de esfuerzos se procede a la planificación de la implementación del sistema. En este plan se especifica la prioridad con que se implementarán las historias de usuarios organizadas por iteraciones así como posibles fechas de liberación.

Las funcionalidades son planificadas generando en cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. De esta manera se garantiza que el cliente tenga un prototipo del sistema con un porcentaje de funcionalidades listas para ser probadas y tomar decisiones en cuanto a cambios que puedan requerirse para mantener el camino correcto rumbo a los objetivos propuestos (36).

En la primera iteración se entregarán las funcionalidades con prioridad alta para el cliente, al finalizar se contará con las funcionalidades descritas en las historias de usuario asignadas. El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad alta, además, de otras con prioridad media. Con la culminación de la misma se tendrán implementadas las peticiones del cliente descritas en las historias de usuario asignadas.

Por último se implementan las historias de usuario de prioridad baja para el cliente, esta función tiene como objetivo mostrar los libros que más visitas han tenido en el sistema.

2.7 Plan de duración de las iteraciones

Siguiendo el desarrollo de la metodología XP se crea el plan de duración de las iteraciones. En este plan se especifica detalladamente el orden de desarrollo de las historias de usuarios dentro de cada iteración así como la estimación completa de dicha iteración.

TABLA 7. DURACIÓN DE LAS ITERACIONES

Iteración	Orden de las HU a implementar	Duración total
1	Perfil de usuario Generar las recomendaciones	4.5 semanas
2	Mostrar libros recomendados	2.5 semanas
3	Mostrar libros más populares	1.5 semana

2.8 Plan de entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (clientes, desarrolladores). Este plan se acopla las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la fase de implementación (37). A continuación se muestra el plan generado en cada una de las iteraciones:

TABLA 8. PLAN DE ENTREGAS

Historia de usuario	Primera iteración	Segunda iteración	Tercera iteración
Perfil de usuario	V 0.5	V 1.0	Finalizado
Generar las	V 0.5	V 0.7	V 1.0

recomendaciones			
Mostrar libros recomendados	-	V 1.0	Finalizado
Mostrar libros más populares	-	-	V 1.0

2.9 Prototipo no funcional de interfaz de usuario

Un prototipo no funcional de interfaz de usuario es una representación parcial de la interfaz de usuario que tendrá el *software* y se utiliza para que el cliente pueda refinar sus necesidades y comunicarlas al desarrollador (37).

Para mostrar una vista preliminar del sistema que se va a desarrollar se crea un prototipo de interfaz, el cual es de fácil modificación conteniendo todas las características definidas de la propuesta de solución. La propuesta que se muestra a continuación parte de la interfaz inicial que presenta la tienda virtual de libros, dentro de la misma, se encuentra un bloque en la parte inferior derecha donde se muestra el módulo recomendador, además en el menú de navegación se muestra un enlace para llenar el perfil de usuario. Esta propuesta puede cambiar en algún momento, según los criterios del cliente.



ILUSTRACIÓN 4. PROTOTIPO DE INTERFAZ NO FUNCIONAL DE USUARIO.

2.10 Tarjetas contenido responsabilidad colaboración

Para el diseño de aplicaciones, 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 (Contenido, Responsabilidad y Colaboración) (37). No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

Cada tarjeta CRC representa a un módulo, donde las responsabilidades definidas son las funciones que realiza y los colaboradores son los métodos del módulo (38). La forma de diseño y organización que se adopta es de diseñar una tarjeta CRC por cada uno de los módulos que brindan una funcionalidad directa al negocio, es decir aquellos que fueron desarrollados desde la raíz. De esta forma se obtiene un diseño simple y no se implementan características que no son necesarias.

TABLA 9. TARJETA CRC GESTIONAR PERFIL DE USUARIO

Perfil de usuario	
Responsabilidades	Colaboraciones
Perfil de usuario	recomendador_configuracion recomendador_configuracion_submit

TABLA 10. TARJETA CRC RECOMENDADOR DE LIBROS

Módulo: Recomendador de libros	
Responsabilidades	Colaboraciones
Generar las recomendaciones	recomendador_metodo1 recomomendador_categorias_Visitados cant_nodos_termino_visitados_user recomendador_mejores_categorias_recomendar nodos_visitados_termino

TABLA 11. TARJETA CRC MOSTRAR LIBROS RECOMENDADOS

Mostrar libros recomendados	
Responsabilidades	Colaboraciones
Mostrar libros recomendados	recomendador_theme recomendador_menu recomendador_block_view recomendador_block

TABLA 12. TARJETA CRC MOSTRAR LIBROS MÁS VISTOS

Mostrar libros más vistos	
Responsabilidades	Colaboraciones

Mostrar libros más vistos	recomendador_metodo1
---------------------------	----------------------

Conclusiones

Los artefactos generados por la metodología XP aportaron las buenas prácticas y procedimientos para la implementación del módulo a desarrollar, construyéndose un cronograma para desarrollar las historias de usuario definidas, con el objetivo de organizar el trabajo y realizar entregas frecuentes al cliente así como la construcción de las tarjetas CRC definiendo las responsabilidades y las colaboraciones de las funcionalidades necesarias.

Capítulo 3: Implementación y pruebas

Introducción

Otras de las fases propuestas por la metodología utilizada son la de producción y mantenimiento que se lleva a cabo mediante la implementación y prueba del sistema. El proceso de implementación será guiado por el análisis y diseño realizado al módulo. Se implementará el algoritmo que dará solución a la propuesta de módulo recomendador de libros en la tienda virtual, una vez concluido el módulo se realizarán las pruebas propuestas por la metodología seleccionada, documentando todos los resultados obtenidos los cuales demostrarán su nivel de calidad.

3.1 Fase de implementación

En esta fase, XP plantea la implementación de cada una de las historias de usuarios. Primeramente se hace un chequeo de cada una de las HU junto con el plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se crean tareas para ayudar a organizar la implementación exitosa de las HU, asignando al responsable de su implementación. Estas tareas pueden escribirse en un lenguaje técnico, al contrario de las historias de usuario que son escritas en el lenguaje del cliente. Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se explican cada una de las iteraciones.

3.1.1 Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad, críticas para ser mostrado al cliente y tomar nuevas iniciativas de forma.

TABLA 13. FUNCIONALIDADES EN LA ITERACIÓN 1

Módulo	Historias de usuario	Tiempo de implementación	
		Estimación	Real
recomendador	Perfil de usuario	1.5	1

	Generar las recomendaciones	3	3.5
--	-----------------------------	---	-----

A continuación se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración, las demás tareas se muestran en el Anexo I.

TABLA 14. TAREA DE INGENIERÍA PARA LA PRIMERA ITERACIÓN

Tarea	
No. de tarea: 1	No. de HU: 1
Nombre: Creación del perfil de usuario.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.3
Fecha inicio: 6/1/2014	Fecha fin: 7/1/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se especificarán los datos que mostrará el perfil de usuario. Se crearán los archivos .info y formulario.inc para la creación del perfil.	

3.1.2 Iteración 2

Durante el transcurso de esta iteración se realiza la historia de usuario Mostrar libros recomendados, teniendo prioridad media.

TABLA 15. FUNCIONALIDAD ABORDADA EN LA SEGUNDA ITERACIÓN

Módulo	Historias de usuario	Tiempo de implementación	
		Estimación	Real
recomendador	Mostrar libros recomendados	2.5	2

3.1.3 Iteración 3

Esta iteración permite implementar las historias de usuario de prioridad baja, que tiene como objetivo brindarle al usuario los libros más populares del sistema.

TABLA 16. FUNCIONALIDAD ABORDADA EN LA TERCERA ITERACIÓN

Módulo	Historias de usuario	Tiempo de implementación	
		Estimación	Real
recomendador	Mostrar libros más vistos	1.5	1.75

3.2 Trabajo con Drupal

El elemento más importante de Drupal son los nodos, la unidad básica de contenido. Los nodos pueden ser de distintos tipos: una página de contenido estático, un mensaje del foro, un artículo que se vende en una tienda *online*, un formulario web, una noticia en un diario digital, una imagen de una galería multimedia, entre otros. Drupal categoriza los nodos a través de los tipos de contenido. Los tipos de contenido constituyen una potente herramienta para implementar contenidos estructurados. Pueden ampliarse con otros campos e incluso extenderse con funcionalidades que provean otros módulos de Drupal.

Los módulos son conjuntos de funcionalidades con la capacidad de integrarse en una instalación de Drupal, añadiendo y mejorando el funcionamiento del sitio en varias áreas: administración, gestión de contenidos, gestión de usuarios, comercio electrónico, entre otros. Los módulos son desarrollados y compartidos libremente por la comunidad de usuarios y desarrolladores de Drupal. Esto hace que sea común encontrar errores e incompatibilidades entre módulos y por ello se debe mantener el sitio web actualizado y en un estado permanente de revisión y mejora continua. Una vez subidos los módulos al sitio, se podrá consultar los módulos disponibles en el área de gestión de módulos (39).

La seguridad y control de los usuarios se garantiza con la capa de permisos de usuario, subsistema que es clave para manejar la integridad de la información almacenada en el sitio. Los permisos hacen referencia a determinadas funcionalidades de los módulos, que son los encargados de generarlos y se vinculan a los roles. Los usuarios tendrán o no permiso para esas funcionalidades en función de los roles que se les haya asignado a cada uno. Un rol se define como un conjunto de reglas que se aplican de forma automática a todos los usuarios (39).

Los bloques son trozos de código o contenido con distintas funcionalidades que típicamente se colocan en los laterales del sitio, aunque en realidad se pueden colocar en cualquiera de las regiones definidas del tema (40).

El sistema de menús de Drupal tiene una doble funcionalidad. Por una parte controla la navegación y permite organizar jerárquicamente los contenidos del sitio a través de los menús y sus elementos. Por otra parte se encarga de direccionar cualquier página del sitio, llamando a las funciones de retorno correspondientes a cada URL que se intenta cargar desde el navegador. Estas funciones de retorno son las encargadas de establecer el contenido y funcionalidad de cada página. Por tanto, sin el sistema de menús no sería posible crear y referenciar páginas en el sitio (40).

El tema es un conjunto de archivos que le da al sitio web un aspecto visual determinado. Cuando se cambia un tema por otro es posible modificar el aspecto completo de la web, sin cambiar con ello las funcionalidades implementadas y los contenidos creados. Con los temas, Drupal separa los contenidos de la presentación o diseño gráfico con que se muestran. Los temas están diseñados con hojas de estilos en cascada lo que hace que sea relativamente sencillo modificarlos y adaptarlos a necesidades específicas (40).

Lo primero para crear el tema es seleccionar el nombre, utilizando en este caso `admire_grunge` y ubicarlo en el directorio `tesis/sites/all/themes` usando el nombre definido, a partir de lo cual el tema a desarrollar se encontrará en la dirección `tesis/sites/all/themes/admire_grunge`. Luego se crea un archivo con la información básica tal como el nombre, versión, regiones, CSS, entre otras y se guarda con la extensión `.info` con lo que se agrega a los registros de Drupal el nuevo tema. Las plantillas son ficheros de código con extensión `.tpl.php` y nombre fijo (ejemplo: **recomendador.tpl.php**), que definen la apariencia y organización general de los elementos en las páginas y su ubicación dentro de las mismas. Estas contienen código HTML y PHP que sobrescriben adicionalmente el contenido o la parte dinámica, declarando una función con un nombre apropiado.

3.3 Diseño de la base de datos

Drupal necesita un gestor de bases de datos para su funcionamiento. Para evitar la dependencia con un gestor de bases de datos en particular, muchas aplicaciones incluyen una capa de

abstracción de la base de datos que hace de intermediario entre el sistema y la base de datos. En este modelo no se realizan llamadas directas a la base de datos, sino que se hacen a través de funciones genéricas definidas por la capa de abstracción de la base de datos. Estas funciones serán siempre iguales, independientemente del gestor de bases de datos utilizado MySQL, SQLite, PostgreSQL, Oracle, entre otros, lo que nos permite cambiar el tipo de base de datos sin que la aplicación se vea afectada.

Selección de la base de datos durante la instalación

Durante el proceso de instalación de Drupal se configuran los parámetros relativos a la base de datos:

- Tipo de base de datos: MySQL, SQLite, PostgreSQL.
- Nombre de la base de datos.
- Nombre de usuario que usa Drupal para conectar a la base de datos.
- Contraseña que usa Drupal para validar al usuario en su proceso de conexión.

El proceso de instalación se realiza a través del navegador, pero implica la modificación del fichero *settings.php*. La instalación de Drupal crea 74 tablas que son utilizadas para almacenar y manejar los datos (como ejemplo: nodos, usuarios, bloques, menús, contenidos, entre otros). Este número de tablas crece con el desarrollo de las nuevas funcionalidades incluidas en los módulos debido a la necesidad de almacenar datos de importancia para los usuarios. 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 la implementación del módulo de recomendación se deben realizar transacciones que establecen entre el usuario y los ítems utilizando el módulo *taxonomy*. Otro elemento fundamental de los sistemas recomendadores son los usuarios utilizándose la tabla *history* que contiene la información referente al tiempo de visita de los usuarios a determinado nodo y la tabla *node* que contiene el título del contenido así como la tabla *taxonomy_term_data* contiene las categorías de

los contenidos, además una nueva tabla *recomendador* que registra la relación de los usuarios con las categorías de los libros.

3.4 Módulos en Drupal

Para la instalación de nuevos módulos, como mínimo se deben tener los conocimientos necesarios para acceder al servidor y copiar los ficheros del módulo en el directorio `sites/all/modules`. Su complejidad depende de cada módulo en particular, siendo por lo general muy sencillo. La activación del módulo es mediante el vínculo `módulo` que brinda el menú con los privilegios administrativos en la parte superior del sistema (41). Para la versión 7.15, Drupal tiene una amplia variedad de módulos implementados. Se estudiaron algunos de los módulos existentes para seleccionar los que pueden ser utilizados en el desarrollo del sistema. A continuación se listan los módulos que aportan funcionalidades para la propuesta de solución.

Poll: Este módulo permite crear encuestas, con el que podremos plantear preguntas a los usuarios del sitio y podrán expresar su opinión votando por alguna de las opciones disponibles.

Path: Este módulo sirve para asignar alias de *URL* a las rutas internas del sitio. De esta forma, las rutas del tipo `node/50`, pueden sustituirse por una *URL* amigable, que se identifique con el contenido. Viene activado por defecto y es altamente recomendable su uso para que los contenidos se posicionen adecuadamente en los buscadores

Visitors: Muestra las estadísticas de las visitas en el sistema.

3.4.1 Módulo desarrollado para el sistema

Drupal incluye un núcleo de 41 módulos dirigidos a las funcionalidades básicas de cualquier sitio y permite la adición de módulos contribuidos, pero ninguno de estos módulos contiene las funcionalidades básicas para la solución.

Para dar cumplimiento a las necesidades planteadas por el cliente, además de configurar los módulos de Drupal descritos en el epígrafe anterior, se precisa implementar el módulo que lleva como nombre Recomendador.

Recomendador

Este módulo permite extraer características de los libros no conocidos aún por el usuario actual, se recomienda luego de compararlo con su perfil, además se puede recomendar de acuerdo a la navegación del usuario en el sistema. Luego se muestra al usuario el listado de libros recomendados de acuerdo a lo que haya definido en su perfil y se muestra en un bloque según las evidencias registradas en la aplicación.

Historias de usuario que implementa: Generar las recomendaciones y Mostrar listados de libros recomendados.

Tablas: node, taxonomy_index, history, taxonomy_term_data y recomendador.

El módulo se encuentra en el directorio `sites/all/modules/recomendador` mostrándose en la Ilustración 4.

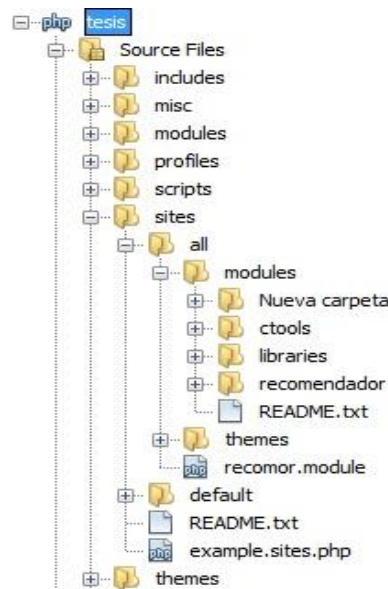


ILUSTRACIÓN 5. UBICACIÓN DEL MÓDULO RECOMENDADOR

3.4.2 Instalación del módulo

En Drupal cada módulo dispone de su propio directorio, para la realización del nuevo módulo se utiliza la carpeta `sites/all/modules` con el mismo nombre que el módulo para simplificar la organización. Una vez creados la carpeta y los archivos de módulo, el siguiente paso será la activación del módulo. La instalación y activación del módulo se realiza como con cualquier otro



módulo, accediendo a Administración Módulos. Para realizar la activación de un módulo basta con seleccionarlo y guardar los cambios. El grupo de módulos implementados para la tienda virtual están agrupados en el paquete Tienda. En ocasiones al instalar un módulo es necesario dirigirse hacia la página de permisos para establecer los privilegios referentes al módulo o hacia la página de configuración para cambiar las opciones del módulo (42).

En cuanto a la integración del módulo con la tienda virtual, se aplicó una integración continua tal y como propone XP, es decir, se fueron integrando pequeños fragmentos de código continuamente, para evitar que al finalizar el proyecto se tuviera que invertir grandes esfuerzos en la integración final. En todo buen proyecto de XP, tendría que existir una versión al día integrada, de manera que los cambios siempre se realicen en esta última versión (34).

3.5 Módulo recomendador

Luego de realizar la instalación el módulo recomendador se deben realizar algunas configuraciones que se pueden llevar a cabo mediante las opciones de la interfaz de administración y otras que se explican a continuación, la primera configuración es crear los roles definidos con sus privilegios, de esta forma se puede activar el bloque en la parte que se desea ver y que usuarios pueden verlos.

Los menús son enlaces responsables de la navegación de un sitio. Un menú consta de tres entidades: el árbol, los elementos y el enlace. Los elementos de un menú tienen propiedades que indican el estado del menú, que puede ser hoja, ampliada o colapso (43). Existen varios tipos de menú estándares en Drupal, dos de ellos (menú principal y menú secundario) son creados por el administrador. Cada menú crea automáticamente un bloque del mismo nombre. Los nuevos elementos de menú se colocan dentro de un menú integrado y etiquetado de navegación, pero los administradores también pueden crear menús personalizados (44).

Un bloque es esencialmente un contenedor independiente que se puede utilizar para alojar los fragmentos de texto o las funcionalidades. Los bloques normalmente se colocan en las barras laterales, el encabezado o pie de página de una plantilla mostrando los enlaces a otras páginas, información del sistema y las encuestas por citar algunos ejemplos (42).

El enlace (Obtener libros de su preferencia) es mostrado en el menú principal del módulo recomendador siendo habilitado el vínculo inmediatamente que el usuario se autentica. Luego de

seleccionar la opción y llenar los datos requeridos, el módulo crea un bloque que muestra un listado con los libros recomendados comparándolos con el perfil.

3.6 Técnicas para el filtrado de la información

Los sistemas recomendadores aplican técnicas de minería de datos para la extracción del conocimiento, para el desarrollo del módulo Recomendador se utiliza la minería de secuencias y reglas de asociación.

La minería de secuencias consiste en encontrar patrones estadísticamente relevantes en colecciones de datos que están representados de forma secuencial debido a la frecuencia con que aparecen este tipo de datos en escenarios de aplicaciones reales, esta técnica constituye uno de los métodos más populares de descubrimiento de patrones, en general, en las bases de datos de transacciones están disponibles los datos en un período de tiempo y se dispone de la fecha en que se realizó la transacción

El descubrimiento de patrones de secuencia en el Log puede utilizarse para predecir las futuras visitas y así poder organizar mejor los accesos y publicidades para determinados períodos de tiempo (45).

Reglas de asociación

Las reglas de asociación se utilizan para descubrir hechos que ocurren en común dentro de un determinado conjunto de datos, es decir, expresa la probabilidad de que la ocurrencia de un evento implique la ocurrencia de otro; esta información se puede utilizar como base para tomar decisiones a la hora de realizar la recomendación.

Según la definición original de Agrawal el problema de minería de reglas de asociación se define como (46):

- ✓ Sea $I = \{i_1, i_2, i_3, i_{n-1}, i_n\}$ un conjunto de atributos binarios llamados ítems.
- ✓ Sea $T = \{t_1, t_2, t_3, t_{n-1}, t_n\}$ un conjunto de transacciones almacenadas en una base de datos.

Cada transacción en T tiene un identificador único y contiene un subconjunto de ítems de I . Una regla se define como una implicación de la forma:

$$X \Rightarrow Y \text{ donde: } X, Y \subseteq I \text{ y } X \cap Y = \emptyset$$

Los conjuntos de ítems X y Y se denominan respectivamente antecedente o parte izquierda y consecuente o parte derecha de la regla.

Una regla necesita un soporte de varias transacciones antes de que ésta pueda considerarse significativa desde un punto de vista estadístico. Para seleccionar reglas interesantes del conjunto de todas las reglas posibles que se pueden derivar de un conjunto de datos se pueden utilizar restricciones sobre diversas medidas de significancia e interés. Las restricciones más conocidas son los umbrales mínimos de soporte y confianza.

El soporte de un conjunto de ítems X en una base de datos D se define como la proporción de transacciones en la base de datos que contiene dicho conjunto de ítems: $\text{sop}(X) = \frac{|X|}{|D|}$

La confianza de una regla se define como: $\text{conf}(X \Rightarrow Y) = \frac{\text{sop}(X \cup Y)}{\text{sop}(X)} = \frac{|X \cup Y|}{|X|}$

Este cálculo significa que el porcentaje de las reglas de la base de datos que contienen X en el antecedente también tienen Y en el consecuente; en otras palabras, que la regla es cierta en el porcentaje dado de los casos, es decir, la confianza puede interpretarse como la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda (46).

Las reglas de asociación deben satisfacer las especificaciones del usuario en cuanto a umbrales mínimos de soporte y confianza. Para conseguir esto el proceso de generación de reglas de asociación se realiza en dos pasos. Primero se aplica el soporte mínimo para encontrar los conjuntos de ítems más frecuentes en la base de datos. En segundo lugar se forman las reglas partiendo de estos conjuntos frecuentes de ítems y de la restricción de confianza mínima.

Reglas de asociación con datos secuenciales.

Obtener reglas de asociación a partir de secuencias, se basa principalmente en el descubrimiento de patrones secuenciales frecuentes. Un ejemplo de uso de esta técnica es la representación de operaciones de distribución y marketing, a partir de secuencias, y las asociaciones obtenidas,

ayudarían a identificar los productos de mayor promoción, de acuerdo con los patrones de compra, por este motivo se decide la implementación de un algoritmo basado en reglas de asociación y secuencias frecuentes. Se utilizan reglas $X \rightarrow Y$, donde X son géneros de libros que se sugieren al usuario. En este caso, al calcular el soporte de los géneros de cuento y novela, se tiene que es del 40 % (2 de cada 5 transacciones son realizadas a estos géneros): $\text{sup}(X) = \frac{2}{5} = 0.4$

Al calcular la confianza este dio como resultado que el 80% de las reglas de la base de datos que contienen el género en el antecedente también tienen el autor en el consecuente; en otras palabras, que la regla: es cierta en el 80% de los casos, donde el usuario accede a los libros de determinado género sienten un nivel de preferencia alto por otros géneros del autor del género seleccionado. Se utiliza secuencias frecuentes para obtener la cantidad de visitas a una categoría. Entonces si la cantidad de visitas a una categoría es mayor que el promedio de visitas a las demás categorías, se propone la recomendación de la categoría con más visitas.

Aplicación de patrones de diseño

Los desarrolladores de aplicaciones acumulan un conjunto tanto de principios generales como de soluciones enfocadas en aplicar ciertos estilos que les guíen en el proceso de creación de software, recibiendo estos principios y estilos la denominación de patrones de software. Dentro de los patrones de software revisten especial protagonismo los patrones de diseño. Según Jakob, un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño en un contexto particular. Su objetivo es reutilizar, además de código, otros artefactos de los cuales se tiene certeza respecto a su correcto funcionamiento (46). En el diseño de la aplicación se hace un uso intensivo de los patrones *Chain of Responsibility*, *Command* y *Singleton*. El patrón de diseño *singleton* (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto., de ahí se puede decir que Drupal sigue el patrón de diseño creacional *Singleton*.

El patrón de diseño *Chain of Responsibility* es un patrón de comportamiento que evita acoplar el emisor de una petición a su receptor dando a más de un objeto la posibilidad de responder a una petición. Para ello, se encadenan los receptores y pasa la petición a través de la cadena hasta que es procesada por algún objeto. Este patrón es utilizado a menudo en el contexto de las interfaces

gráficas de usuario donde un objeto puede contener varios objetos. El sistema de menús de Drupal sigue el patrón de diseño de comportamiento *Chain of Responsibility*. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y que la función se llama para hacer el trabajo.

Muchos de los *hooks* de Drupal utilizan el patrón de diseño de comportamiento *Command* para reducir el número de funciones que son necesarias para la aplicación, los módulos no tienen que definir cada *hook*, sino sólo los que vayan a aplicar. Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto, con lo que además se facilita la parametrización de los métodos.

3.7 Roles de usuario

Con el sistema deben interactuar un conjunto de usuarios con niveles de acceso y objetivos diferentes. Drupal posee un sistema de permisos basado en roles que permite especificar qué tipos de usuarios tienen la capacidad de crear, eliminar, editar o incluso ver los libros recomendados. Los roles no son más que una vía de colocar a los usuarios autenticados dentro de categorías, donde las mismas limitan los niveles de acceso (40). Por ejemplo: los usuarios anónimos pueden tener la capacidad de aportar evidencias para la base de conocimientos del recomendador y ver los libros más visitados, pero solo los usuarios autenticados que han creado su perfil de usuario pueden ver los libros recomendados. Solo los administradores poseen el control total del recomendador y de los libros que se publican. El administrador accede al vínculo *Personas\Permisos*, en la interfaz de administración para configurar los permisos de usuarios asignando los privilegios.

3.8 Pruebas

La fase de pruebas es una de las más importantes de la metodología XP. Permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones (34).

Uno de los pilares fundamentales de XP es el proceso de pruebas, en el cual los desarrolladores prueban constantemente tanto como sea posible. De esta manera se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección [28]. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores y las pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente (34).

3.8.1 Pruebas unitarias

Las pruebas unitarias o pruebas de unidad deben ser construidas antes que el código, permitiéndole a los programadores tener máxima claridad de lo que van a programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberán pasar, lo que optimizará el trabajo y el código será de mayor calidad (47).

Una de las características incorporadas en Drupal 7 es el framework de pruebas. Tiene una herramienta de prueba integrada en el núcleo que le permite definir y automatizar las pruebas de la tienda virtual. La ejecución automática de las pruebas permite verificar que cuando se inserta un fragmento de código en Drupal, funciona como se espera en el contexto de la aplicación. En lugar de evaluar una función directamente, muchas veces una prueba funcional ejecutará toda la aplicación, en condiciones que hacen que sea fácil de comprobar si el código que se está probando está trabajando correctamente.

El primer paso en la configuración del entorno de prueba es a la página Módulos, y activar el módulo de pruebas Testing, luego de hacer clic en el botón "Guardar configuración" la herramienta de prueba está lista para ser utilizada. Antes de ver la creación de nuestro propio conjunto de casos de prueba, echemos un vistazo a los casos de prueba que se suministran con Drupal 7 (48).

Para el desarrollo de los casos de pruebas automatizados utilizando el módulo Testing implementaron los casos de pruebas basados en el paradigma de la POO, pues una prueba es un objeto que se crea a partir de la clase *DrupalWebTestCase*. Los casos de prueba implementaron siguiendo los pasos siguientes:

1. Crear una nueva clase que extiende de *DrupalWebTestCase*.
2. Definir las variables *instance* cada una representando a un usuario específico con privilegios asignados que se utilizarán a lo largo del proceso de pruebas.
3. Definir el nombre la descripción y el grupo asociado, utilizando la función `getInfo ()`.
4. Configurar el método `setup ()`.
5. Crear los árboles de usuarios asignando los permisos referenciándolos con la variable `$this`.
6. Escribir los métodos de prueba empezando con la palabra `test`.
7. En cada método de prueba, utilizar una o más afirmaciones (`assert`) para poner a prueba los valores reales.

Luego de la ejecución las pruebas aparece una pantalla que muestra los resultados al finalizar.

3.8.2 Pruebas de aceptación

Se considera que las pruebas de aceptación poseen un peso superior a las unitarias, pues las mismas muestran a los desarrolladores la satisfacción del cliente. Por este motivo se puede decir que estas ponen fin a una iteración dando inicio a la siguiente. Las pruebas de aceptación se describen mediante una tabla llamada Caso de Prueba de Aceptación y son destinadas a evaluar si al final de cada iteración se obtuvo la funcionalidad requerida y son creadas sobre la base de las HU. El cliente desde su punto de vista creará los casos de prueba en conjunto con los desarrolladores para probar que una HU ha sido implementada correctamente. Por cada HU se elaborarán todas las pruebas de aceptación que se necesiten para asegurar su correcto funcionamiento. A continuación se muestran los casos de prueba de aceptación realizados para la HU Perfil de usuarios, los restantes casos de prueba de aceptación se muestran en el Anexo IV:

TABLA 17. CASO DE PRUEBA DE ACEPTACIÓN

Caso de prueba de aceptación.

Código: HU1_P1	No. de HU: 1
Nombre: Perfil de usuario	
Descripción: Prueba para llenar el formulario con los géneros de libros de la tienda virtual.	
Condiciones de ejecución: Usuario autenticado.	
Pasos de ejecución: El usuario selecciona el enlace “Obtener libros de su preferencia”, mostrada en el menú de navegación, se muestra un formulario con los géneros de los libros ofertados en la tienda, un vez seleccionado el o los g{eneros el usuario oprime el botón Enviar y se le muestra un listado con los libros más leídos perteneciente a ese género.	
Resultado esperado: Los datos se seleccionan y se muestra satisfactoriamente los libros a recomendar.	
Evaluación de la prueba: Satisfactoria.	

Se realizaron tres iteraciones donde se encontraron siete no conformidades entre la primera y la segunda iteración, las que fueron mitigadas en la tercera iteración para lograr la calidad y el correcto funcionamiento del sistema (ver Ilustración 7).

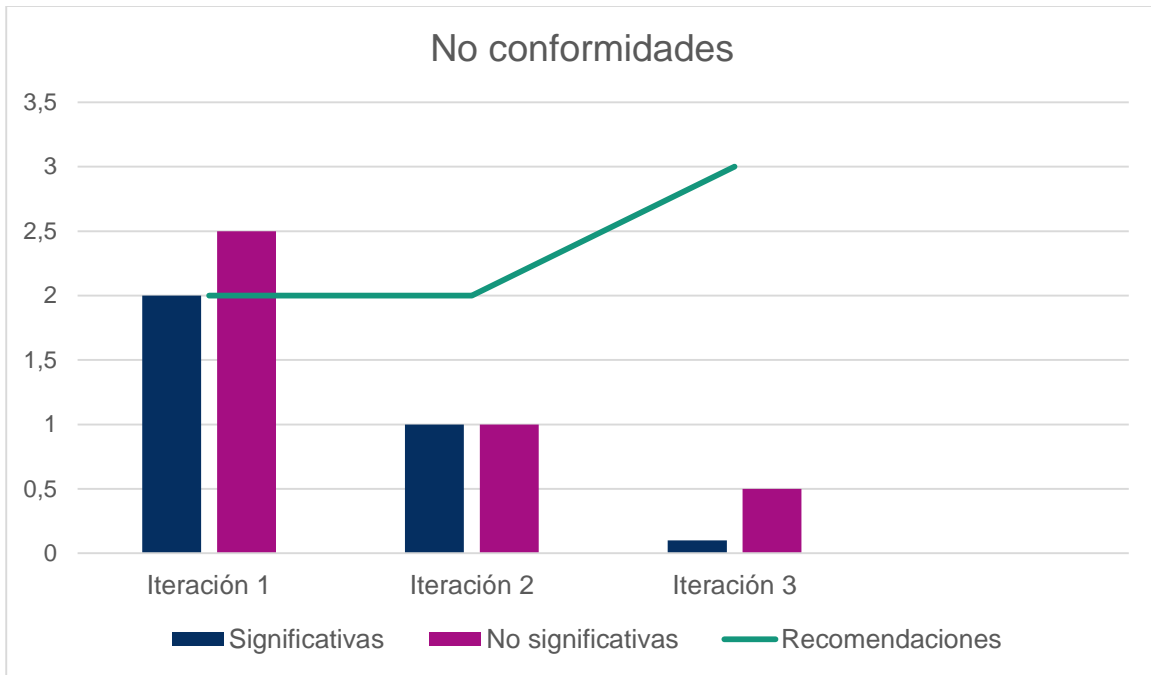


ILUSTRACIÓN 6. NO CONFORMIDADES SIGNIFICATIVAS, NO SIGNIFICATIVAS Y RECOMENDACIONES.

Se detectaron errores ortográficos fundamentalmente en las no conformidades no significativas, en el caso de las significativas, se detectaron errores de validación. Las pruebas se realizaron de forma iterativa e incremental, comprobando en cada iteración que hubiesen sido corregidos los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del software.

Conclusiones

Mediante el desarrollo de este capítulo se logró la implementación del módulo recomendador respondiendo a las necesidades del cliente, documentando cada paso para su mejor entendimiento y futuro cambio. El correcto uso de la implementación del módulo se probó mediante las pruebas de funcionalidad y las pruebas de aceptación demostraron el cumplimiento de los requerimientos del cliente.

Conclusiones

Al finalizar la investigación se arribaron a las siguientes conclusiones:

- ✓ El estudio del estado de arte arrojó un punto de partida a la solución definiendo características y métodos a utilizar.
- ✓ El análisis y diseño de la solución propuesta estuvo guiado por la metodología de desarrollo de software XP, la cual permitió generar los artefactos que documentaron la solución, para continuar con el desarrollo futuro de nuevas versiones y centró al equipo de desarrollo en los requerimientos de esta.
- ✓ La integración del módulo de recomendación a la tienda virtual permitió contar con una nueva funcionalidad que permite presentar los contenidos publicados de acuerdo a las preferencias particulares de cada usuario.
- ✓ El cliente quedó satisfecho con el módulo Recomendador pues la tienda virtual se ajusta a las tendencias de personalización de las ofertas que utilizan los sitios de distribución de contenido, para mantener la fiabilidad de sus clientes.

Recomendaciones

Teniendo en cuenta los resultados durante el desarrollo de la investigación se proponen las siguientes recomendaciones:

- ✓ Incorporar nuevos criterios de selección que respondan a los intereses de usuarios exigentes.
- ✓ Aplicar buenas prácticas de programación orientada a objetos para hacer extensible el código.

Referencias bibliográficas

1. **Richard, Sandhusen L.** *Mercadotecnia*. s.l. : Compañía Editorial Continental, 2002. pág. 637 y 638. Vol. Primera Edición.
2. **Ricci, F., Rokach, L. y Shapira, B.** *Introduction to Recommender Systems Handbook*. 2011. 978-0-387-85819-7.
3. **Resnick, M.L.** *Persuasive Design Through Intelligent Recommendation Systems*. s.l. : Proc. of the Human Factors and Ergonomics Society Annual Meeting, 2004. págs. 1503-1507.
4. **J.PZZANI, MICHAEL., BILLSUS, DANIEL.** Content-Based Recommendation Systems. *Springerlink*. [En línea] 2007. [Citado el: 25 de noviembre de 2013.] <http://www.springerlink.com/content/qq35wt68l6774261>.
5. **SCHAFER, J.BEN., FRANKOWSKI, DAN., HERLOCKER, JON., SEND, CHILAD.** *Collaborative Filtering Recommender Systems*. 2007. *Sitio web Information Search and Retrieval*. [En línea] [Citado el: 25 de noviembre de 2013.] <http://www.inf.unibz.it/~ricci/ISR>.
6. **BURQUE, ROBIN.** Knowledge-based recommender systems. . [En línea] [Citado el: 25 de diciembre de 2013.] www.cs.odu.edu/~mukka/cs795sum10dm/Lecturenotes/Day6/burke-elis00.pdf.
7. **J.PAZZANI, MICHAEL.** Content-Based and Demographic Filtering Disponible en: . [En línea] 2000. [Citado el: 15 de noviembre de 2013.] http://www.cs.northwestern.edu/~pardo/courses/mmml/papers/collaborative_filtering/a_framework_for_content_based_demographic_filtering_AIR99.pdf.
8. **Manouselis, N. y Costopoulou, C.** *Analysis and Classification of Multi-Criteria*. 2007. págs. 415-441. Vol. 10.
9. **Sinbad2, Grupo de investigación.** Personalización y los Sistemas de Recomendación. [En línea] 13 de marzo de 2012. [Citado el: 28 de noviembre de 2013.]
10. **Herrera, Viedma Enrique, Porcel, Carlos y Hidalgo, Lorenzo.** *Sistemas De Recomendaciones: Herramientas de filtrado de información*. Granada : Dpto. de Ciencias de la Computación, 2011.
11. **Burke, R.** *Hybrid recommender systems: Survey and experiments*. Estados Unidos : Kluwer Academic Publishers, 2002. págs. 331-370. Vol. 12.
12. **Peinado Rodríguez, S.A., y otros.** Diseño E Implementación De Un Software Recomendador Y Adaptativo De Educación Básica Secundaria En Las Instituciones Educativas Del Municipio De Lorica - Córdoba. *BuenasTareas.com*. [En línea] septiembre de 2011. [Citado el: 2 de octubre de

2013.] <http://www.buenastareas.com/ensayos/Dise%C3%B1o-e-Implementaci%C3%B3n-De-Un-Software/2714571.html>.

13. **Adomavicius, G. y Tuzhilin, A.** *Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.* 2005.

14. **Saboya Vargas, Aniceto.** *Uso de Recomendadores, Asistentes y Ayudantes en sistemas Tutores.* . Catalunya, España : Depto. LSI-U. Politécnica de Catalunya, 2005.

15. *Sistema recomendador de noticias para el portal Octavitos.* **Yamile, Franklin Mustelier y.** UCI : s.n., 2013.

16. **Yera Toledo, R.** Concepción y desarrollo de un sistema de recomendación para jurados online de programación. . [En línea] UCI. Dirección de Información, junio de 2010. [Citado el: 2 de noviembre de 2013.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_02825_10.

17. **Yordi Plasencia, E. y Milán Cristo, N.** *Herramienta de selección didáctica para guiar el aprendizaje interactivo en el módulo Ejercicios de la colección El Navegante.* La Habana : s.n., 2012.

18. **Frankilin Mustelier, Yamile Peña.** *Sistema recomendador de noticias para el portal Octavitos.* UCI : s.n., 2013.

19. **Resnick, P., Varian, H.R. y Editors, Guest.** *Recommender Systems. Communications of the ACM.* 1997. págs. 56-89.

20. **Avison, D.E. y Fitzgerald, G.** *Information Systems Development: Methodologies, Techniques, and Tools.* . McGraw-Hill : s.n., 1995.

21. **Beck, Kent.** *Embracing Change with Extreme Programming.* . Addison Wesley Longman : Inc, 1999. ISBN-10: 0201616416 | ISBN-13: 978-0201616415.

22. **Riola, Jose Carlos Carvajal.** *Metodologías Ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial.* Barcelona : UPC - Barcelona, 2008.

23. **Serrano-Cobos, Jorge.** *Evolución de los sistemas de gestión de contenidos (CMS). Del mainframe al open source.* 2007. págs. 213-215. Vol. 3.

24. **Gil, F.** *Experto en Drupal 7.* s.l. : Forcontu, 2011.

25. **Padilla Matos, Reynier.** Programación Web y Tecnologías Informáticas. [En línea] 28 de febrero de 2008. [Citado el: 14 de enero de 2014.]

26. **Rivera, E.A., Zamora, R.G. y Soria, M.G.** *Sistema de Educación a Distancia.* s.l. : IV Congreso de Tecnología en Educación, 2012.

27. **Travis, B.** *Drupal 7 for Windows Developers.* s.l. : Apress, 2011.

28. **Stewart, Celeste.** The advantage of PHP. Designer's Playground. . [En línea] 3 de enero de 2006. [Citado el: 22 de enero de 2014.] <http://www.designersplayground.com/articles/118/1/The-Advantages-of-PHP/Page1.html>.
29. **Asenjo Sánchez, Jorge.** *Sistemas Gestores de Bases de Datos*. España : Creative Commons, 2009.
30. **Gilfillan, Ian.** *La Biblia de MySQL*. España : Kluwer Academic, 2003.
31. **NETBEANS.ORG.** Bienvenido a NetBeans . *www.netbeans.org*. [En línea] Portal del IDE Java de Código Abierto, 2011. [Citado el: 20 de noviembre de 2013.] http://netbeans.org/index_es.html.
32. **Kabir, Mohammed J.** *La Biblia Servidor Apache 2*. s.l. : Anaya Multimedia, 1999. ISBN: 8441514682.
33. **Priolo, Ing. Sebastian.** *Programación Extrema*. 2011.
34. **Beck, K.** *Extreme Programming Explained*. s.l. : Addison-Wesley, 2000.
35. **Escribano, Gerardo Fernández.** *Introducción a Extreme Programming*. 2009.
36. **Solis, Ricardo Tercero.** XP Programación Extrema para Desarrollo de Sistema Basados en Web. *Sistematización de Control del Par que Vehicular de la Secretaría del Medio Ambiente y Recursos Naturales, Delegación Morelos*. 21 de abril de 2010. 1.1.
37. **Bajo de Luque, M.J. y et.al.** Metodologías ágiles. *Agile Alliance*. [En línea] 2011. [Citado el: 25 de 2 de 2014.] <http://metologiasagiles.herobo.com/index.php/2011-12-05-16-09-55/metologia-xp>.
38. **Chaffer, Jonathan.** Drupal. *Drupal programming from an object-oriented perspective*. [En línea] 29 de Junio de 2012. [Citado el: 25 de Febrero de 2014.] <http://drupal.org/node/547518>.
39. **Noble, Mark.** *Drupal 7 First Look*. s.l. : Packt Publishing, Noviembre, 2010.
40. **Travis, Brian.** *Pro Drupal 7 for Windows Developers*. s.l. : Apress, Febrero, 2011.
41. **Tomlinson, Todd y VanDyk, John K.** *Pro Drupal 7 Development*. s.l. : Apress,, Diciembre, 2010. Vol. 3rd.Edition.
42. **Gil, Fran.** *FORCONTU-Experto en Drupal nivel I*. s.l. : FORCONTU, 2011.
43. —. *FORCONTU-Experto en Drupal nivel III*. s.l. : FORCONTU, 2011.
44. **Buytaert, Dries.** Drupal. *Working with Menus*. [En línea] 23 de Enero de 2013. [Citado el: 25 de Abril de 2014.] <http://drupal.org/documentation/modules/menu>.
45. **Morales, Cristóbal Romero, Soto, Sebastián Ventura y Martínez, Cesar Hervás.** *Estado actual de la aplicación de la minería de datos a los sistemas basados en web*. Córdoba : s.n., 2005.
46. **Nielsen, Jakob.** useit.com. [En línea] [Citado el: 10 de abril de 2014.]

47. **Saboya Vargas, Aniceto.** . *Uso de Recomendadores, Asistentes y Ayudantes en sistemas.* Catalunya, España : Politécnica de Catalunya, 2005.
48. **Butcher, M., y otros.** *Drupal 7 Module Development.* Birmingham,Mumbai : s.n., 2010. ISBN 978-1-849511-16-2.
49. **Gómez García, J. C.** *Portales de internet: concepto, tipología básica y desarrollo.* s.l. : El profesional de la información, 2001. págs. 4-13. Vol. 10. 7.
50. **Ezquerro, Norberto.** *Interacción hombre-máquina y usabilidad: Diseño centrado en el usuario.* Universidad de Coruña : s.n., 2008.
51. **ISO, 9241-11.** *Ergonomic requirements for office work with visual display terminals (VDT)s – Part 11 Guidance on usability.* Londres : s.n., 1998.
52. **SEGUIDO FONT, MIGUEL.** *Sistemas de recomendación para webs de información sobre la salud. . Sitio Web UPCOMmons.* [En línea] 2009. [Citado el: 30 de noviembre de 2013.] <http://upcommons.upc.edu/pfc/handle/2099.1/7193>.
53. **PLAZA, ERIC., MCGINTY, LORRAINE.** *Distributed case-based reasoning.* [En línea] 2005. [Citado el: 20 de noviembre de 2013.] http://www.iiia.csic.es/People/enric/papers/Distributed_CBR.pdf.
54. **Matos, V.L.** *Sistemas de Recomendación y Personalización.* Universidad Nacional del Sur : s.n., 2006.
55. **Drachsler, H., Hummel, H.G.K. y Koper, R.** *Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. , .* Netherlands : Inderscience Enterprises Ltd, 2008.
56. **BERTATE, LETICIA., MACHADO, ROFRIGO., VALERIA.** *PGMúsica Sistema de Recomendación de música. Inco.* [En línea] 18 de 4 de 2006. [Citado el: 25 de octubre de 2013.] www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf..
57. **Carrillo de Comas, A.F., Llovet Rodríguez, J. M. y Rodríguez Paulete, I.** *Sistema recomendador orientado a la educación terapéutica del paciente diabético.* Madrid : Universidad Complutense de Madrid (UCM), 2010.
58. **MacManus, Richard.** *Problems of Recommender Systems.* ReadWrite. [En línea] SAY Media, Inc., 28 de enero de 2009. [Citado el: 13 de diciembre de 2013.] http://readwrite.com/2009/01/28/5_problems_of_recommender_systems#feed=/search?keyword=Problems%20of%20Recommender%20Systems.
59. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El proceso unificado de desarrollo.* Addison-Wesley : s.n., 2000.

60. **Carvajal Riola, J.C.** *Metodologías ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial*. Barcelona : Tesis de maestría, 2008.
61. **Suárez Jorge, Alinoet.** *Estudio Comparativo sobre 11 Metodologías de Desarrollos Web*. . La Habana : s.n., 2009.
62. **Letelier, P. y Penadés, M^a C.** *Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP)*). Valencia : Universidad de Valencia, 2008. Vol. 05. ISSN 1666-1680..
63. **Arellano Moncayo, Fabricio Gerardo.** Desarrollo e implantación del sistema de reservación de laboratorios para el Laboratorio de la Facultad de Ingeniería de Sistemas. [En línea] Space Escuela Politécnica Nacional, 3 de mayo de 2011. [Citado el: 13 de enero de 2014.] <http://bibdigital.epn.edu.ec/handle/15000/3824>.
64. **Palacio, J.** *Flexibilidad con Scrum, principios de diseño e implantación en campos Scrum*. s.l. : SafeCreative, 2007.
65. **Schwaber, K. y Beedle, M.** *Agile Software Development with Scrum*. Upper Saddle River : Prentice Hall, 2002.
66. **Cuerda García, Xavier. Mosaic.** Introducción a los sistemas de gestión de contenidos (CMS) de código abierto. [En línea] 29 de noviembre de 2004. [Citado el: 13 de diciembre de 2013.] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-los-contenidos-cms-de-codigo-abierto/>. ISSN: 1696-3296.
67. **Hispano, Drupal.** . [En línea] Drupal Hispano., 11 de abril de 2005. [Citado el: 25 de enero de 2014.] <http://drupal.org.es/caracteristicas>.
68. **Eguíluz Pérez, J.** *Introducción a Javascript*. 2008.
69. **Seco, José Antonio González.** *El lenguaje de programación C#* . 2002.
70. **Pérez Eguíluz, Javier.** *Introducción a JavaScript*. Madrid : Autoedición, 2009.
71. **Pérez, J.F.** *Introducción a XHTML*. s.l. : Serie Científica, 2012. Vol. 1.
72. **Converse, Tim, Park, Joyce y Morgan, Clark.** *PHP5 and MySQL Bible*. Indiana : Wiley Publishing,Inc. ISBN: 0-7645-5746-7.
73. **Rea Inca, Paúl Fernando.** *Diseño y Desarrollo de un Prototipo de Control mediante SMS para Casas Inteligentes*. Sangolquí : s.n., 2012.
74. **Drupal.** [En línea] https://drupal.org/project/themes?solsort=iss_project_release_usage%20desc.
75. **Beck, K.** *Extreme Programming Explained*. Addison-Wesley : s.n., 2000.
76. **Fowler, Martin.** *Who Needs an Architect?* . s.l. : IEEE SOFTWARE, 2003.

77. Cerda, Felipe. *NetBeans el unico IDE que necesitas*. 2009.

Glosario de términos

Algoritmo: Lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

Bloques: Los bloques son elementos dentro del CMS Drupal que se emplean para incluir información ya sea un menú o cualquier otra función que se desee, casi siempre aparecen en las columnas izquierda y derecha de la página.

Clustering: es una técnica estadística que permite una generación automática de grupos en los datos. Incluso, existen algoritmos de clustering que permiten la generación de grupos jerárquicos, consiguiendo una mayor abstracción y representación de la información para poder recuperarla más eficiente.

e-commerce (comercio electrónico): consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas.

Hook: Del inglés gancho, palabra usada en la comunidad de Drupal para nombrar los métodos de la interfaz que deben implementar los módulos desarrollados para este CMS.

Minería de datos: Consiste en la extracción no trivial de información, que reside de manera implícita en los datos. Dicha información, previamente desconocida y podrá resultar útil para algún proceso. La minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos. En inglés se le reconoce por las siglas DM, Data Mining.

Objetos: En el paradigma de POO, un objeto se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico se define como la instancia de una clase. Estos objetos interactúan unos con otros, en contraposición a la visión tradicional en la cual un programa es una colección de subrutinas (funciones o procedimientos), o simplemente una lista de instrucciones para el computador.

Anexo I.

TABLA 18.TAREAS DE INGENIERÍA PARA LA PRIMERA ITERACIÓN.

Tarea	
No. de tarea: 2	No. de HU: 1
Nombre: Construcción de la funcionalidad del perfil de usuario.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.7
Fecha inicio: 08/01/2014	Fecha fin: 10/01/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se implementará el código que permitirá al usuario seleccionar los géneros de su preferencia, para su desarrollo se utilizan los hooks: hook_menu, hook_form y hook_submit y el método ver_recomendados.	
Tarea	
No. de tarea: 3	No. de HU: 2
Nombre: Configuración del generador de recomendaciones.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.5
Fecha inicio: 13/01/2014	Fecha fin: 15/01/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se crearán los archivos .module, .tpl.php, .info, para implementar las funcionalidades que tendrá el módulo recomendador.	
Tarea	
No. de tarea: 4	No. de HU: 2
Nombre: Creación de la tabla en la base de datos correspondiente a la funcionalidad Generar las recomendaciones.	
Tipo de tarea: desarrollo	Puntos de estimación: 1

Fecha inicio: 16/01/2014	Fecha fin: 21/01/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira.	
Descripción: Se creará el archivo de tipo .install con el objetivo de crear la base de datos para guardar la información y extraerla en caso necesario.	
Tarea	
No. de tarea: 5	No. de HU: 2
Nombre: Implementación de la funcionalidad generador de recomendaciones.	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Fecha inicio: 22/01/2014	Fecha fin: 05/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se implementará el código que permitirá realizar las recomendaciones de los libros tomando como objetivo sus preferencias a partir del perfil y las evidencias dejadas en la tienda virtual. Para su creación se utilizaron los hooks, hook_theme, hook_block_info, hook_block, hook_block_view, hook_menu, así como varios métodos para su correcto funcionamiento.	
Tarea	
No. de tarea: 6	No. de HU: 2
Nombre: Obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.1
Fecha inicio: 06/02/2014	Fecha inicio: 07/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se obtendrá el código fuente para posibilitar su futura modificación	

TABLA 19. TAREAS DE INGENIERÍA PARA LA SEGUNDA ITERACIÓN.

Tarea

No. de tarea: 1	No. de HU: 3
Nombre: Creación de Mostrar libros recomendados.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.3
Fecha inicio: 10/02/2014	Fecha fin: 12/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda ver los libros que se les recomendaron a partir de su perfil o de la navegación por el sistema.	
Tarea	
No. de tarea: 2	No. de HU: 3
Nombre: Creación de la tabla en la base de datos correspondiente a la funcionalidad Mostrar libros recomendados.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.4
Fecha inicio: 14/02/2014	Fecha fin: 17/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se creará el archivo de tipo .install donde se crea la tabla "recomendador" para almacenar la información.	
Tarea	
No. de tarea: 3	No. de HU: 3
Nombre: Implementación de la funcionalidad Mostrar libros recomendados.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.8
Fecha inicio: 18/02/2014	Fecha inicio: 24/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	

Descripción: Se implementará la funcionalidad Mostrar libros recomendados que permitirá al usuario ver los libros recomendados a partir de su perfil de usuario y las evidencias dejadas en la tienda virtual, mostrando estas últimas en un bloque. Para la implementación se utilizarán los hooks: hook_theme, hook_block, hook_block_view, entre otros, para posibilitar el resultado esperado.	
Tarea	
No. de tarea: 4	No. de HU: 3
Nombre: obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.1
Fecha inicio: 25/02/2014	Fecha inicio: 26/02/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se guardará el código fuente para futuros cambios y modificaciones.	
Tarea	
No. de tarea: 5	No. de HU: 3
Nombre: Configuración de Mostrar libros recomendados.	
Tipo de tarea: configuración	Puntos de estimación: 0.4
Fecha inicio: 03/03/2014	Fecha fin: 05/03/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se gestionarán los archivos incluidos sobre esta funcionalidad para acceder a la base de datos y presentar del listado de libros recomendados en un bloque.	

TABLA 20. TAREAS DE INGENIERÍA PARA LA TERCERA ITERACIÓN.

Tarea	
No. de tarea: 1	No. de HU: 4
Nombre: Configuración de la funcionalidad Mostrar libros más vistos.	

Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.4
Fecha inicio: 10/03/2014	Fecha fin: 12/03/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se especifican los datos que mostrará el módulo recomendador, creándose los archivos .module, .tpl.php, .info, para la creación e implementación de la funcionalidad.	
Tarea	
No. de tarea: 2	No. de HU: 4
Nombre: Creación de la tabla en la base de datos.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.4
Fecha inicio: 17/03/2014	Fecha fin: 19/03/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se creará el archivo de tipo .install para almacenar la información.	
Tarea	
No. de tarea: 3	No. de HU: 4
Nombre: Construcción de Mostrar libros más vistos.	
Tipo de tarea: desarrollo	Puntos de estimación: 1
Fecha inicio: 19/04/2014	Fecha inicio: 26/04/2014
Programador responsable: Yanet de los Ángeles Cárdenas Almira	
Descripción: Se implementará el código utilizando del hook_theme para crear la funcionalidad mostrar libros más populares que le permitirá al usuario ver los libros más vistos publicados en la tienda virtual.	

Anexo II.

Casos de prueba de aceptación.

TABLA 21. CASO DE PRUEBA DE ACEPTACIÓN HU2_P2.

Caso de prueba de aceptación.	
Código: HU2_P2	No. De HU: 2
Nombre: Generar las recomendaciones	
Descripción: Prueba para la funcionalidad que permite generar las recomendaciones.	
Condiciones de ejecución: El usuario tiene que estar autenticado.	
Pasos de ejecución: <ul style="list-style-type: none"> ✓ Verificar si el usuario llenó el perfil. ✓ Si no se ha llenado el perfil, se analizan la navegación en la tienda virtual jerarquizando los libros de su preferencia. 	
Resultado esperado: Los datos son actualizados satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	

TABLA 22. CASO DE PRUEBA DE ACEPTACIÓN HU3_P3.

Caso de prueba de aceptación.	
Código: HU3_P3	No. De HU: 3
Nombre: Mostrar libros recomendados	
Descripción: Prueba para la funcionalidad que le permite al usuario ver el listado de libros que se les recomienda.	
Condiciones de ejecución: El usuario tiene que estar autenticado.	
Pasos de ejecución: Si el usuario no llena los datos en el formulario, el sistema muestra el listado de los libros recomendados de acuerdo a su navegación. En caso contrario, el usuario puede seleccionar la opción de Obtener libros de su preferencia donde se le mostrará un formulario (caso de prueba HU1_P1) para seleccionar los géneros de su preferencia para seguidamente visualizar el listado de libros recomendados.	
Resultado esperado: La aplicación muestra el listado de libros recomendados correctamente.	

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: Mostrar libros más vistos.	
Descripción: Prueba para la funcionalidad de mostrar libros más vistos.	
Condiciones de ejecución: El usuario no debe estar autenticado.	
Pasos de ejecución: La aplicación mostrará en un bloque los libros que más visitas han tenido en la tienda virtual.	
Resultado esperado: Se muestran los libros más vistos.	
Evaluación de la prueba: Satisfactoria.	

ENTREVISTA REALIZADA AL CLIENTE.

Objetivo: Obtener información relacionada con el módulo que se desea desarrollar.

1. ¿En qué consiste el módulo que se desea desarrollar?
2. ¿A qué tipo de usuarios está destinado este producto?
3. ¿Qué servicios brindará la tienda virtual y el módulo a desarrollar?
4. ¿Cómo los administradores de la aplicación gestionarán la información requerida?
5. ¿Bajo qué criterio de búsqueda será mostrada y ordenada la información?