



# **SISTEMA RECOMENDADOR DE NOTICIAS PARA EL PORTAL OCTAVITOS**

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

**Autores:**

Yamilé Peña Cruz  
Franklin Mustelier Ramírez

**Tutores:**

Ing. Ernesto Yordi Plasencia  
MSc. Roberto López Dosagües

**Co-Tutores:**

Ing. Yordanis Rodríguez Rodríguez  
Ing. Yunior Orosa Velázquez  
Ing. Yasmani Ceballos Izquierdo

La Habana, junio 2013  
“Año 55 de la Revolución”

# Declaración de Autoría

Declaramos ser autores del presente trabajo de diploma y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos el presente a los 13 días del mes de junio del año 2013.

---

Firma de la tesista  
Yamilé Peña Cruz

---

Firma del tesista  
Franklin Mustelier Ramírez

---

Firma del tutor  
Ing. Ernesto Yordi Plasencia

---

Firma del tutor  
MSc. Roberto López Dosagües



*“La ciencia se compone de errores, que a su vez, son los pasos hacia la verdad.”*

*Julio Verne*

***De Yamilé***

*A mis padres Adaris Cruz Pupo y Otsenre Rueda Matos.*

*A mi abuela Gladis Pupo Telles y mi hermana Marlin Peña Marrero.*

***De Franklin***

*A mis padres Osálida Ramírez Peña y Francisco Mustelier Barroso.*

*A mis abuelos paternos y maternos.*

.

### **De Yamilé**

*A mis padres y mis abuelos maternos, por hacer todo para que yo sea la persona que soy, por luchar siempre por mí y estar presente en cada uno de los pasos que doy.*

*A mi mamá y mi otro papá Rueda por impulsarme siempre a luchar, a no renunciar a mis sueños, que al final, es el sueño de ellos también, y por confiar siempre en mí.*

*A mi papá Osvaldo, a mis tíos maternos y paternos, a mis hermanos Osvi y Marlin, porque siempre están pendientes de mí y apoyándome en todo.*

*A todos mis compañeros en los mejores 5 años de mi vida, en especial a mis amigas del alma Nersa, Nani y Mailín.*

*A mis tutores Yordanis, Yuniór, Yordi, Dosagües y Yasmani por su esfuerzo y dedicación a lo largo de este trabajo.*

*A mi compañero de tesis, que lo admiro muchísimo por tener siempre mente positiva y enfrentar los retos que la vida le depara, por apoyarme y alentarme en todo el recorrido de este trabajo.*

*En fin, a todos aquellos que me ayudaron a escalar la cima de este sueño.*

### **De Franklin**

*A Dios, por permitirme cumplir las expectativas que un grupo de personas esperaban de mí.*

*A mis padres y a mi familia por darme su apoyo incondicional, que permitieron que me formara como una persona de provecho y útil a la sociedad.*

*A mis hermanos de la vida Leo y Rey, estos han sido padre vital en mi formación tanto profesional como social.*

*A mi novia Lisandra por su amor infinito y su apoyo a mi preparación.*

*A todas las personas que he compartido en estos años y que han sido en múltiples ocasiones compañeros de valor.*

*A todos los profesores que me han formado y han compartido sus experiencias para lograr prepararme como un profesional capacitado.*

*A mi compañera de tesis por su dedicación al trabajo, ha sido muestra de lo que se puede alcanzar cuando nos proponemos una meta.*

Recommender systems are software tools and/or techniques responsible for making certain suggestions, helping to get the required information according to the user's preference for a group of items with specific characteristics. This work aims to develop a news recommender system that improves the personalization of the contents in the website Octavitos. This system allows Octavitos website users to satisfy specific information interests, providing a recommended newsgroup from a user profile where it selects the topics that it wants to be advised and evidence left in the website navigation. To guide the development was used Extreme Programming (XP) generating proposed methodology's fundamental artifacts for each stage of work, PHP programming language, the Cascading Style Sheets (CSS) for the presentation of content and Drupal 7.15 as Content Management System (CMS), achieving an efficient integration. Finally, to validate the conducted proposed solution, appropriated tests were performed, and to determine the degree of satisfaction it has been made use of the Iadov's technique, obtaining as results of it an acceptable level, verifying compliance with the objectives.

**Key words:** *news, Octavitos website, recommender systems*

---

Los sistemas de recomendación son herramientas de software y/o técnicas encargadas de realizar determinadas sugerencias, ayudando a obtener la información que se necesita de acuerdo a las preferencias del usuario por un grupo de elementos con características específicas. El presente trabajo tiene como objetivo desarrollar un sistema recomendador de noticias, que mejore la personalización de los contenidos presentes en el Portal Octavitos. Este sistema permite a los usuarios del Portal Octavitos satisfacer intereses informativos específicos, proporcionando un grupo de noticias recomendadas a partir de un perfil de usuario donde el mismo selecciona las temáticas que desea que se le recomiende y las evidencias de navegación dejadas en el portal. Para guiar el desarrollo se utilizó Extreme Programming (XP) generándose los artefactos fundamentales que propone la metodología para cada etapa de trabajo, PHP como lenguaje de programación, las hojas de estilo en cascada (CSS) para la presentación de los contenidos y Drupal 7.15 como Sistema de Gestión de Contenido (CMS), logrando una integración eficiente. Finalmente, para la validación de la propuesta de solución llevada a cabo se realizaron las pruebas correspondientes, y para determinar el grado de satisfacción se ha hecho uso de la técnica Iadov, obteniendo como resultados de la misma un nivel aceptable, verificándose el cumplimiento de los objetivos propuestos.

**Palabras claves:** *noticia, Portal Octavitos, sistema de recomendación*

---

---

INTRODUCCIÓN .....	1
1 FUNDAMENTACIÓN TEÓRICA .....	5
1.1 Introducción .....	5
1.2 Principales conceptos asociados al dominio del problema.....	5
1.3 Sistemas recomendadores .....	7
1.3.1 Clasificación de los sistemas de recomendación.....	9
1.3.1.1 Recomendadores basados en contenido.....	9
1.3.1.2 Sistemas basados en filtrado colaborativo .....	10
1.3.1.3 Sistemas basados en datos demográficos.....	11
1.3.1.4 Sistemas basados en el conocimiento .....	11
1.3.1.5 Sistemas de recomendación híbridos .....	11
1.3.2 Aplicaciones de los sistemas de recomendación.....	12
1.3.3 Limitaciones de los sistemas recomendadores .....	14
1.3.4 El problema de la recomendación .....	16
1.3.5 Algoritmos de recomendación .....	16
1.3.6 Análisis de sistemas homólogos.....	17
1.4 Tendencias y tecnologías actuales a considerar.....	18
1.4.1 Metodologías de desarrollo de <i>software</i> .....	18
1.4.2 Metodologías ágiles .....	19
1.4.2.1 SCRUM .....	20
1.4.2.2 Extreme Programming (XP).....	22
1.4.2.3 Metodología a utilizar.....	24
1.4.3 Sistemas de gestión de contenidos (CMS).....	25
1.4.3.1 Drupal 7.15.....	26
1.4.4 Lenguajes de programación .....	28
1.4.4.1 Lenguajes y tecnologías del lado del cliente .....	28
JavaScript 1.5.....	29
XHTML 1.1.....	29
CSS 3 .....	31
1.4.4.2 Lenguajes del lado del servidor .....	31
PHP 5.3.5 .....	31
1.4.5 Sistema gestor de base de datos (SGBD).....	32
1.4.5.1 MySQL 5.5.....	32
1.4.6 Entorno de desarrollo integrado (IDE) .....	33
1.4.6.1 Zend Studio .....	34



1.4.6.2	NetBeans 7.14 .....	34
1.4.6.3	IDE seleccionado.....	35
1.4.7	Servidor web .....	35
1.4.7.1	Apache 2.2.22.....	36
1.5	Conclusiones .....	36
2	DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN .....	38
2.1	Introducción .....	38
2.2	Objetivos del sistema.....	39
2.3	Definición de la audiencia .....	39
2.4	Usuarios relacionados con el sistema.....	39
2.5	Definición de servicios .....	40
2.6	Historias de usuario (HU).....	41
2.7	Estimación de esfuerzos por HU.....	45
2.8	Plan de iteraciones .....	45
2.9	Plan de duración de las iteraciones .....	46
2.10	Plan de entregas .....	46
2.11	Prototipo de interfaz de usuario no funcional.....	47
2.12	Tarjetas CRC .....	47
2.13	Conclusiones.....	48
3	IMPLEMENTACIÓN Y PRUEBAS .....	50
3.1	Introducción .....	50
3.2	Fase de implementación.....	51
3.2.1	Iteración 1 .....	51
3.2.2	Iteración 2 .....	51
3.2.3	Iteración 3 .....	52
3.3	Estructura de Drupal.....	52
3.4	Creación de las tablas adicionales en la base de datos .....	54
3.5	Módulo desarrollado para el SR.....	55
3.6	Técnicas para el filtrado de la información .....	58
3.7	Arquitectura del SR.....	59
3.8	Instalación del módulo .....	60
3.9	Configuración del SR.....	60
3.9.1	Roles de usuarios.....	61
3.10	Pruebas.....	62
3.10.1	Pruebas de Aceptación .....	64

3.11	Técnica de iadov .....	66
3.12	Conclusiones.....	70
CONCLUSIONES.....		71
RECOMENDACIONES .....		72
TRABAJOS CITADOS .....		73
GLOSARIO DE TÉRMINOS.....		77
ANEXOS .....		80

Figura 1. PM = Soporte de la gestión del proyecto. PD = Se describe un proceso en el método que incluye esta etapa. PA = El método describe prácticas, actividades y artefactos para la etapa. El color degradado indica que no se incluye en el método. ....	21
Figura 2. SCRUM Ciclo de vida. ....	21
Figura 3. Prácticas comunes de la metodología de desarrollo XP. Las prácticas se refuerzan entre sí. ....	23
Figura 4. Fases de XP. ....	24
Figura 5. Representación esquemática del patrón MVC en Drupal. ....	26
Figura 6. Prototipo de interfaz de usuario no funcional de la solución (bloque azul) dentro del Portal Octavitos. ....	47
Figura 7. Estructura de capas y datos en Drupal. ....	53
Figura 8. Tablas adicionadas a la base de datos de Drupal. ....	55
Figura 9. Ubicación del módulo recomendador en la estructura de archivos de Drupal. ....	56
Figura 10. Arquitectura del sistema recomendador. ....	59
Figura 11. Niveles de acceso por roles. ....	62
Figura 12. Resultado. ....	64
Figura 13. No conformidades significativas y no significativas. ....	65

Tabla 1 Usuarios relacionados con el sistema y su descripción. ....	40
Tabla 2 HU Perfil de usuario. ....	42
Tabla 3 HU Generar recomendaciones. ....	42
Tabla 4 Mostrar listado de noticias recomendadas. ....	43
Tabla 5 HU Mostrar noticias más populares. ....	43
Tabla 6 HU Apariencia. ....	43
Tabla 7 HU Seguridad. ....	44
Tabla 8 HU Condiciones tecnológicas. ....	44
Tabla 9 HU Condiciones Legales. ....	44
Tabla 10 Estimación de esfuerzos por HU. ....	45
Tabla 11 Plan de duración de las iteraciones. ....	46
Tabla 12 Plan de entregas. ....	46
Tabla 13 Tarjeta CRC 1. Recomendador de noticias. ....	48
Tabla 14 Funcionalidades abordadas en la primera iteración. ....	51
Tabla 15 Funcionalidad abordada en la segunda iteración. ....	51
Tabla 16 Funcionalidad abordada en la tercera iteración. ....	52
Tabla 17 Módulo Recomendador. ....	55
Tabla 18 Caso de prueba de aceptación HU1_P1. ....	64
Tabla 19 Cuadro lógico de ladov. ....	66
Tabla 20 Índice de satisfacción grupal. ....	67
Tabla 21 Resultados de la técnica iadov. ....	68
Tabla 22 Variables de la fórmula ISG. ....	68

# INTRODUCCIÓN

El crecimiento en los últimos años de las Tecnologías de la Información y las Comunicaciones (TIC), y su utilización en todos los campos está transformando nuestra sociedad. Internet ha revolucionado la concepción tradicional de la comunicación, el aprendizaje, la economía y el mundo en general. Es la etapa del desarrollo de la humanidad donde la aglomeración de conocimientos, noticias y el dinamismo constante de tecnologías constituye un hecho.

Cada día aumenta la cantidad de personas y medios digitales, que comparten información mediante la red, no solo para procesar los datos almacenados en cualquier formato digital, sino también como medio de comunicación, permitiendo el acceso a noticias, recursos y servicios prestados por computadoras remotas así como sistema de publicación y difusión de la información (1).

Este incesante desarrollo de la tecnología, hace muy difícil poseer un conocimiento actualizado en determinada esfera del saber (2). El incremento del volumen de la información no se asemeja con las habilidades, experiencias y recursos necesarios que tienen las personas, para administrar esa cantidad de datos que de forma constante se publica por los diferentes medios existentes.

La Universidad de las Ciencias Informáticas (en lo adelante UCI), juega un papel muy importante en el desarrollo informático del país. Esta institución trabaja en aras de convertir aplicaciones y servicios informáticos en un espacio de difusión de sus resultados, enlace y comunicación permanente entre docentes, investigadores, desarrolladores y las comunidades universitarias en general.

Cada facultad en la UCI posee un portal web, espacio creado con el fin de resolver necesidades de información específicas, además de actuar como medio publicitario, educativo, formativo y de superación cultural. Estos portales recopilan información general, agrupando por categorías las temáticas de mayor interés para quienes los visiten, además brindan un punto común de acceso donde obtener las noticias e informaciones necesarias a cada una de las áreas, así como aquellos servicios que permitan interactuar con el resto de la comunidad.

Octavitos (disponible en: <https://octavitos.uci.cu/>) es un portal web encaminado a la publicación y administración de noticias, desarrollado sobre la base de herramientas libres. Tiene como objetivo reflejar para la comunidad universitaria y en especial los miembros de la Facultad 4 la imagen, historia e identidad de la misma, así como la posibilidad de brindar determinados niveles de información, y es la forma principal de comunicación con sus estudiantes, profesores y trabajadores, para escuchar sus inquietudes, sus sugerencias y de esta manera conocer criterios con respecto al trabajo de las diferentes áreas (3).

Entre las principales temáticas que publica el portal se encuentran: noticias locales, económicas, deportivas, culturales, políticas, de opinión, entre otras.

Octavitos ha logrado captar la atención de la Facultad y el resto de la Universidad en los cuatro años que lleva prestando servicios, durante ese tiempo el número de usuarios ha crecido y cuenta con un alto promedio de visitas diarias.

La interacción de los clientes con el Portal permite enriquecerlo; a partir del estudio de las estadísticas resultantes de esta interacción y los comentarios de los visitantes, es posible mejorar los servicios que se brindan, posibilitando un incremento de su nivel de usabilidad, elemento que adquiere un significado específico en el campo de la informática e Internet, definiéndose la usabilidad por la ISO/IEC 9241-11 como: *“el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso”* (4).

Los autores de la presente investigación coinciden que el grado de usabilidad es un requisito valioso para una aplicación web, constituyendo un concepto importante dentro de la interacción usuario – aplicación, puesto que es una necesidad fundamental para crear de una forma sencilla y eficaz un entorno gráfico relacionado con la web, además de ser un instrumento para comprobar la efectividad desde el punto de vista del usuario final.

Por otra parte, el alto flujo de información, aparejado a la alta frecuencia de publicación de los artículos periodísticos en el Portal, provoca que los usuarios, por cuestiones de interés y tiempo, no accedan o visualicen los contenidos de su preferencia. Por otro lado, son los administradores del sistema los encargados de seleccionar, publicar y destacar aquellas informaciones que a su consideración constituyan las más relevantes. Este criterio de selección no siempre coincide con las preferencias informativas de los usuarios. Por otro lado, se necesita de mayor fidelidad de los usuarios con el Portal, a partir de una mejor orientación de sus servicios, proporcionándole un espacio de consulta permanente, más interactivo y más usable.

Lo anterior conlleva a resolver el siguiente **problema**: ¿Cómo contribuir al mejoramiento de la personalización de los contenidos en el Portal Octavitos?

El **objetivo general**: Desarrollar un sistema recomendador (SR) de noticias, que mejore la personalización de los contenidos presentes en el Portal Octavitos.

La parte de la ciencia que será **objeto de estudio** son los procesos de desarrollo de los sistemas recomendadores.

El **campo de acción** se centra en el desarrollo de los sistemas recomendadores para portales web.

En correspondencia con el objetivo general de la investigación que se presenta se propone realizar los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación.
- ✓ Desarrollar el análisis y diseño del SR para el Portal Octavitos.
- ✓ Implementar la solución propuesta de acuerdo a la estructura de diseño definida.
- ✓ Evaluar la calidad del sistema a través de pruebas.

La **idea a defender** que sustenta esta investigación es que el desarrollo de un SR de noticias, contribuirá al mejoramiento de la personalización de los contenidos presentes en el Portal, que permita satisfacer intereses informativos de sus usuarios.

Para la realización de esta investigación se utilizan 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 tener en cuenta la caracterización de la evolución histórica de los sistemas de recomendación como herramientas bases para concebir el sistema actual.

**Analítico-sintético:** El empleo de este método se evidencia cuando se realiza un análisis de toda la teoría y documentación, que permiten la extracción de los elementos fundamentales relacionados con el objeto de estudio.

En cuanto a los empíricos se emplea la *observación*, que permitió valorar las diferentes manifestaciones y comportamientos de los procesos y fenómenos, así como describir y explicar las características fenomenológicas del objeto de estudio y someterlo a elaboración racional.

Las **tareas de investigación** definidas para dar cumplimiento al objetivo de la investigación fueron las siguientes:

- ✓ Elaboración del estado del arte de los sistemas recomendadores y los principales conceptos y elementos teóricos del tema a tratar.
- ✓ Descripción de los artefactos generados durante el desarrollo de la solución propuesta.

- ✓ Confección de un prototipo no funcional de interfaz de usuario.
- ✓ Implementación de las principales funcionalidades de la solución informática propuesta.
- ✓ Integración del sistema de recomendación con el Portal Octavitos.
- ✓ Análisis de los resultados arrojados de las pruebas internas realizadas al SR.

La presente investigación está conformada por la siguiente estructura: introducción, tres capítulos, conclusiones, recomendaciones, trabajos citados, glosario de términos y anexos. Los capítulos abordan los siguientes temas:

**Capítulo 1: Fundamentación teórica.** En este capítulo se realiza un análisis de los principales conceptos relacionados con el objeto de estudio, así como un análisis del estado del arte de los sistemas recomendadores en apoyo a los portales web y en sentido general. También se realiza un estudio de las distintas herramientas y tecnologías a utilizar en el desarrollo del sistema propuesto.

**Capítulo 2: Descripción de la propuesta de solución.** En este capítulo se determinan los servicios que brindará el portal, 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 capítulo abarca todo lo relacionado con la implementación del sistema y el proceso de pruebas utilizado. Se implementan todas las funcionalidades identificadas, logrando un sistema que permita cumplir el objetivo general de los estudios que se presentan. Se detallan también las pruebas que se le realizaron al sistema ya finalizado, con el objetivo de asegurar la eficiencia de la solución.



# CAPÍTULO 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

Uno de los retos principales que en la actualidad tienen que afrontar los sistemas de información es la gestión eficaz del gran volumen de datos que almacenan para poder facilitar a los consumidores de información el acceso a recursos que satisfagan sus necesidades de una manera sencilla y ágil. Esta necesidad se vuelve más apremiante en una sociedad como la actual donde el nivel de exigencia de los usuarios es cada vez mayor (5).

En el ámbito de las aplicaciones web, el problema de la sobrecarga de información ha sido abordado adoptando diferentes medidas para atenuar estos problemas, tal es el caso de los sistemas recomendadores. Además, en un ambiente dominado por los medios sociales para la información, la relación y la comunicación; la dinámica que se establece entre contenidos, personas y tecnología cambia radicalmente. Ante la relevancia que cobra el contenido generado por los usuarios, la localización de las mejores fuentes de información requiere sistemas recomendadores que incorporen la naturaleza social a este escenario. Es por esta razón que en la actualidad, los sistemas recomendadores han evolucionado y es posible encontrarlos en diversos ámbitos de aplicación, convirtiéndose en una herramienta fundamental que apoya servicios tales como: el comercio electrónico, servicios de información científica, proveedores en línea, entre otros.

En el presente capítulo se elabora la fundamentación teórica de la investigación destacando los conceptos relacionados con el problema existente. Posteriormente, se presenta un análisis de las metodologías de desarrollo de software, herramientas, lenguajes y tecnologías necesarias para dar cumplimiento a las necesidades de la solución que se propone.

### 1.2 Principales conceptos asociados al dominio del problema

Para el desarrollo y mayor entendimiento de la presente investigación, es necesario puntualizar algunos conceptos fundamentales que son referidos en el marco metodológico.

A partir de la bibliografía consultada los autores consideran que las definiciones más concisas de portal web proceden de países foráneos, existiendo una gran variedad de opiniones al respecto, aunque la

mayoría coinciden en que un portal web se define como: “...un punto de entrada a internet donde se organizan sus contenidos, ayudando al usuario y concentrando servicios y productos, de forma que le permitan realizar cuanto necesite hacer en la Red a diario, o al menos que pueda encontrar allí todo cuanto utiliza cotidianamente sin necesidad de salir de dicho sitio” (6).

No obstante la variedad de criterios existentes, para esta investigación se asume un criterio elaborado a partir de los objetivos de la investigación que se presenta. Por lo tanto, se define como portal web un sitio web donde el usuario obtiene contenidos y servicios comunes que son imprescindibles para poder satisfacer sus necesidades.

Un portal web tiene como objetivo fidelizar a sus usuarios, y conseguir que éstos no accedan al mismo de forma eventual, sino que se habitúen a usarlo diariamente, y establecer algún tipo de vínculo casi personal entre el usuario y el portal. A este vínculo podemos llamarlo interacción entre el usuario y la aplicación.

Los conceptos que forman la base del campo de interacción usuario-aplicación, pueden sintetizarse en varios objetivos clave (7):

- ✓ Eficacia: la interfaz debe permitir que las metas y objetivos del usuario se cumplan de una forma práctica y directa, observando los niveles de precisión, fiabilidad y rigurosidad requeridos por la aplicación de interés.
- ✓ Eficiencia: la interfaz debe potenciar la productividad del usuario, a la vez que disminuye o elimina la posibilidad de errores.
- ✓ Usabilidad: la interfaz debe ser fácil de usar, minimizando o disminuyendo la carga cognitiva del usuario.

Para que exista en el portal una buena interacción se necesita que el mismo posea los objetivos planteados anteriormente pero principalmente para lograr un grado de interacción eficiente detallamos específicamente el significado de usabilidad.

El concepto de usabilidad se define en el estándar ISO 9241 como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” (4). El grado de usabilidad de un portal web es un requisito importante desde el punto de vista de su aceptación por el usuario. Estos requisitos son más importantes cuando la competencia por atraer la atención del usuario aumenta con la proliferación de la Web.

Para que exista una buena usabilidad el portal debe poseer una gran personalización, ya que la misma constituye un criterio de usabilidad. La personalización de lo que trata es de intentar desarrollar sitios web cuya interfaz y cuyas funcionalidades se adapten a los usuarios en función de sus expectativas, necesidades, preferencias y deseos (8). Es decir, ofrece al usuario aquello que más se ajuste a sus necesidades además de captar su atención. De esta forma, es preciso que ambos conceptos se apliquen en la propuesta de solución.

### 1.3 Sistemas recomendadores

La literatura consultada permitió determinar que el concepto que se analiza es poco abordado en la literatura cubana, aunque en la actualidad existen numerosas definiciones de sistemas recomendadores de autores foráneos (9) (10) (11).

Según (12) los sistemas recomendadores (también conocidos como sistemas de filtrado de información) *“son herramientas de software y técnicas que ofrecen determinadas sugerencias de empleo y utilidad para el usuario”*. Los autores de la presente investigación consideran poco abarcador este concepto ya que el mismo no deja muy amplio el tema de las sugerencias, por lo que es necesario agregar que un SR es una herramienta de software y/o técnica que ofrece determinada sugerencia de empleo, ayudando a obtener la información que se necesita de acuerdo a las preferencias del usuario por un grupo de elementos con características específicas.

Estos sistemas aplican técnicas de filtrado de información que pueden facilitar el acceso de los usuarios a la información que necesitan. En dominios textuales, los sistemas de filtrado evalúan y filtran los recursos disponibles en la Web básicamente para asistir a los usuarios en tareas de recuperación de información, [ver (13)] aunque también se utilizan para predecir la valoración de los usuarios sobre elementos que aún no han evaluado.

El origen de los sistemas recomendadores se remonta a principios de la década de los '90, surgiendo como un área de investigación independiente cuando comienzan a emerger los servicios de filtrado de noticias dentro de los servicios de grupos de noticias, permitiendo a la comunidad de usuarios acceder exclusivamente a aquellas que potencialmente podían ser de su interés (13). En aquel tiempo, los investigadores comenzaron a centrarse en los problemas de recomendación que explícitamente dependen de las clasificaciones de estructuras.

Uno de los grupos de investigación pioneros en el desarrollo de sistemas recomendadores fue el proyecto GroupLens de la Universidad de Minnesota que ha proporcionado una gran parte de la base algorítmica de muchos sistemas recomendadores. Fueron los primeros en introducirlos usando un algoritmo de búsqueda de vecinos para proporcionar predicciones en los grupos de noticias de USENET (14).

En la actualidad, aunque la investigación sobre sistemas recomendadores es un campo que se encuentra muy activo y que genera mundialmente un gran número de publicaciones y congresos todos los años, Cuba se encuentra en una etapa inicial en el tema, si bien cada año se incrementan trabajos relacionados aplicados principalmente en universidades del país. Esta tendencia provoca que en la actualidad es difícil encontrar un portal web que no provea un SR que se encargue de proporcionar nuevos servicios que amplíen los ya tradicionales.

Por la importancia que reviste para esta investigación los tipos de datos usados por los sistemas recomendadores, los autores de la tesis que se presenta coinciden con la opinión de (12) al agruparlos en tres tipos de objetos, entre ellos “... *los ítems, los usuarios y las transacciones*”.

El término **ítem**, fue utilizado por (12) para referirse a lo que el sistema recomienda a los usuarios. Los sistemas recomendadores se centran en un tipo específico de ítems (productos, sitios web, personas, películas, música, libros, noticias, imágenes, etc.) 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.

Otro de los grupos de datos que utilizan los sistemas recomendadores son los **usuarios**. Según (12) refiere la persona a la que se le van a proporcionar recomendaciones, los cuales pueden tener diversidad de preferencias cuando accede a un software. Para poder ofrecer las recomendaciones adecuadas el sistema debe explorar toda la información disponible acerca de dicho usuario, ya sea la que inicialmente se haya solicitado o la que se va recopilando a lo largo de la interacción de la persona con el sistema.

Por último las **transacciones** genéricamente se refieren a la interacción que existe entre el usuario y el SR. De esta forma, se puede almacenar información acerca de las preferencias de los usuarios sobre los ítems. Frecuentemente las transacciones se pueden ver a través de los **ratings**. Este término se refiere a la evaluación o calificación que realiza una persona sobre un ítem que es sometido a su consideración, de esta forma se puede almacenar el criterio que tiene el usuario sobre un ítem determinado (12).

En resumen, los autores de los estudios que se presentan, en correspondencia con el cumplimiento del objetivo general de la investigación puntualizan que básicamente, los sistemas recomendadores ayudan a las personas a encontrar lo que necesitan especialmente de la Web y han tenido una amplia aceptación entre los usuarios. Su objetivo es explorar y filtrar las mejores opciones a partir de un perfil de usuario (preferencias, características, etc.) considerando un importante número de posibilidades diferentes, muchas de ellas provenientes de la Web.

### 1.3.1 Clasificación de los sistemas de recomendación

A partir de la bibliografía consultada y considerando los aspectos establecidos anteriormente, los sistemas recomendadores pueden recibir diferentes clasificaciones dependiendo del tipo de información que utilizan para realizar sus recomendaciones, en diferentes tipos. La clasificación más aceptada y reconocida por los principales autores en el tema son los que a continuación se exponen:

#### 1.3.1.1 Recomendadores basados en contenido

Los recomendadores basados en el contenido (“*Content based*”) utilizan solamente las preferencias del usuario; intentan recomendar elementos que son similares a aquellos que el usuario previamente valoró en el pasado (o está examinando en la actualidad). 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. (Por ejemplo: palabras en el documento) y el perfil de usuario se basa en como el usuario valora esas características.

Entre las ventajas que posee este tipo de recomendador se puede destacar como una de las más importantes, que el sistema puede generar recomendaciones sin la necesidad de contar con un historial previo, por lo que permite realizar predicciones independientemente del historial del usuario. Como principal desventaja se tiene la sobre-especialización, donde el usuario está limitado a que le recomienden ítems similares a los que recomendó. Es decir, las recomendaciones resultantes suelen ser siempre muy similares dado que se basan en la misma información (15).

Las ventajas y desventajas que poseen los sistemas de recomendación basados en contenido se pueden mencionar en (16) y con más profundización en (17).

### 1.3.1.2 Sistemas basados en filtrado colaborativo

El filtrado colaborativo (*“Collaborative filter”*) es otra de las formas de filtrado de información que permite recomendar los elementos que son preferidos por usuarios similares o predecir la utilidad de ciertos elementos para un usuario particular. Algunos de los autores más reconocidos en el tema definen los sistemas de recomendación basados en filtrado colaborativo de la siguiente manera: *“recomienda elementos a las usuarios basados en las opiniones de otras personas”* [Ver (18)].

Las preferencias de los usuarios utilizadas en el sistema, generalmente, son representadas por una evaluación numérica, las cuales pueden obtenerse a partir de diferentes acciones que realice el usuario, como por ejemplo registros de compra, los tiempos de acceso a las páginas web, etc.

Una vez que los usuarios expresan sus preferencias a través de la evaluación de ítems que el sistema les presenta, toda esta información se utiliza para crear un perfil aproximado del usuario. Luego el sistema busca entre los usuarios que posean características similares y forma los grupos de usuarios llamados “usuarios más cercanos”, es decir, el sistema asocia los “ratings” suministrados por el usuario que solicita la recomendación contra los “ratings” que conforman el perfil de los usuarios anteriores. Se combinan estos grupos de usuarios y se devuelve un listado de las recomendaciones que contiene los ítems mejor evaluados por estos usuarios y que aún no han sido evaluados por uno nuevo.

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

Los sistemas recomendadores colaborativos basados en los usuarios identifican a los usuarios similares (usuarios más cercanos), al usuario actual y predice la valoración que el mismo daría a un ítem basándose en la que dan sus usuarios más cercanos. Los sistemas que son basados en el ítem son los que por cada elemento determinado se calculan los elementos más similares. A partir de las valoraciones de los mismos entonces se predice la valoración para el usuario [Ver (19)].

Muchos sitios web han incorporado mecanismos de recomendación basado en el filtrado colaborativo como sucede con Amazon ([www.amazon.com](http://www.amazon.com)), el mayor portal de ventas de bienes en línea, Barnes and Noble, una librería virtual, o Netflix, que ofrece servicios de alquiler de DVD. En general la aplicabilidad de estas técnicas son tan diversas que se observan trabajos desde el comercio electrónico hasta propuestas asociadas a la enseñanza a distancia.

Otras clasificaciones de los sistemas recomendadores en la literatura de Burke y Ricci que proponen en (20) y (12) respectivamente distinguen los siguientes tipos:

### 1.3.1.3 Sistemas basados en datos demográficos

Los sistemas basados en datos demográficos (*“Demographic based”*) son similares a los colaborativos, solo que en el cálculo de los usuarios más cercanos usan información demográfica, es decir, elementos basados en el perfil demográfico de los usuarios como la edad, el sexo, etc. En otras palabras, las diferentes recomendaciones deben ser generadas a partir de los diferentes elementos demográficos.

Muchos sitios web adoptan simples y efectivas soluciones de personalización basadas en la demografía. Por ejemplo, se les recomienda información a los usuarios de determinados sitios web basados en su idioma o país. Dichas sugerencias pueden ser personalizadas de acuerdo a la edad del usuario.

Los beneficios que posee este sistema frente a los sistemas basados en filtrado colaborativo es que no requieren un largo historial de valoraciones para hacer recomendaciones a un usuario. Además, los sistemas basados en datos demográficos clasifican a los usuarios en grupos y hace recomendaciones de acuerdo con el grupo.

### 1.3.1.4 Sistemas basados en el conocimiento

En los sistemas basados en el conocimiento (*“Knowledge based”*) las sugerencias de los ítems se basan en inferencias sobre las necesidades de los usuarios y sus preferencias. Para ello se utiliza conocimiento donde se tiene información sobre cómo un ítem específico responde a una necesidad en particular del usuario y, por lo tanto, la razón sobre la relación entre la necesidad y una posible recomendación. Varios ejemplos se encuentran en (12) donde se profundiza sobre este tipo de sistemas de recomendación.

### 1.3.1.5 Sistemas de recomendación híbridos

Los autores de esta investigación coinciden con la definición que nos expone el autor Robin Burke donde define a 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” [ver (20)].

Como se ha podido apreciar, las diferentes clases de enfoques de recomendación antes expuestos, han sido de gran utilidad en diversos campos y escenarios. En general cada uno de ellos presentan algunas desventajas cuando trabajan por separado. Algunas de las situaciones más preocupantes pueden ser la sobre-especialización en el caso de los sistemas basados en contenido y la necesidad de una gran cantidad de usuarios cuando se trata de un sistema basado en filtrado colaborativo.

Una opción que se ha encontrado para mitigar las desventajas que presentan por separado algunos de los sistemas anteriormente expuestos, es combinar la recomendación basado en contenido con el filtrado colaborativo. De esta manera se pueden usar las ventajas que ofrece la primera para atenuar las desventajas de la segunda y viceversa.

La tendencia actual se inclina por el desarrollo de sistemas de filtrado híbrido que combinan características de los sistemas basados en contenido y colaborativos, para atenuar las desventajas de cada uno de ellos y mejorar de esta manera la eficiencia global del funcionamiento del sistema.

El análisis bibliográfico permitió asumir para esta investigación el SR basado en contenido por las ventajas que se señalan:

- ✓ El sistema puede generar recomendaciones por el contenido y no por opiniones subjetivas de otros usuarios.
- ✓ El sistema 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.

### 1.3.2 Aplicaciones de los sistemas de recomendación

Dado que la presente investigación trata de un SR, se mencionan algunos ejemplos reales de los mismos. Existen sistemas recomendadores para multitud de dominios de aplicación: libros, películas, viajes, ropa, música, páginas web, etc. Cabe destacar que alguno de ellos ha servido de inspiración para el diseño del sistema de recomendación desarrollado en esta investigación. A continuación expondremos algunos de estos recomendadores.

Un SR muy popular en Internet es **Amazon**. En esta página se pueden buscar libros, películas, productos electrónicos, etc. Para todas estas áreas la aplicación propone inicialmente una lista de



productos de oferta, últimas novedades, los más vendidos, etc. A partir de lo que el usuario seleccione de esa lista inicial, el sistema utiliza su SR para guiar al usuario hacia el producto que esté buscando. Amazon también permite que los usuarios se registren rellenando un formulario con sus preferencias. De esta forma el proceso de recomendación puede ser más exitoso porque el sistema tiene más información de lo que el usuario está buscando.

Los sistemas recomendadores son ampliamente usados para fines comerciales, además de otras aplicaciones en Internet que los implementan, muestra de ello es el sitio de ventas anteriormente expuesto, pero podemos citar otros ejemplos como **Amie Street**, **Genius** y **Pandora** que son páginas creadas con el objetivo de brindar servicios de música, **Netflix**, por su parte proporciona servicios de alquiler de DVD, **FAB** es otro de los ejemplos de este tipo de sistemas, el cual es orientado a la recomendación de URL que combina el uso de información por extensión por el enfoque colaborativo, **GroupLens** es otro de los proyectos que diseña, implementa y evalúa un sistema de filtrado colaborativo para USENET, un servicio de listas de discusión con un alto volumen de negocio en Internet.

Internacionalmente existen disímiles ejemplos de recomendadores que tienen un alto nivel de aceptación por parte de los usuarios de la red global.

En (21) se presenta un SR colaborativo el cual pretende ayudar al paciente diabético recientemente diagnosticado, donde la herramienta desarrollada utiliza datos y mediciones analíticas del historial del paciente, en base a los cuáles recomienda una serie de consejos y contenidos formativos. El objetivo final es ayudar al paciente a mantener una buena salud y aumentar su calidad de vida.

Cuba es un país donde el desarrollo de sistemas recomendadores, es aún incipiente. Sin embargo la investigación acerca de estos sistemas que contribuyan al aumento de la calidad de las aplicaciones es un tema al cual los ingenieros, diseñadores y desarrolladores en general están prestando más atención. Además, en la bibliografía consultada los trabajos e investigaciones que se han encontrado provienen fundamentalmente de la UCI, sin embargo pueden citarse algunas investigaciones que muestran resultados enfocados a implementaciones reales de sistemas recomendadores.

En este proceso se destaca la UCI. Por otra parte, (22) muestra la implementación de un SR de ejercicios para jurados online de programación. Su filosofía de recomendación está basada en la técnica de filtrado colaborativo donde genera automáticamente paquetes de actividades a realizar por los estudiantes, basados en su rendimiento y progreso actual. Consiste además en sugerir al usuario aquellos problemas que a pesar de haber sido resueltos por usuarios semejantes a él, aún no han sido resueltos por este.

Otro de los trabajos que pueden citarse orientados a la implementación de herramientas recomendadoras en la UCI aplicados al campo de la enseñanza es en (23), que tiene como objetivo desarrollar un SR de cuestionarios interactivos para los estudiantes que utilizan la colección El Navegante<sup>1</sup>, mejorando así la consolidación del conocimiento.

De manera general, al realizar un estudio de las aplicaciones de los sistemas recomendadores en nuestro país se puede apreciar que este desarrollo ocurre mayormente en la UCI, como anteriormente se había mencionado. Para la presente investigación se han encontrado muy pocos trabajos y aplicaciones relacionados con el tema en nuestro país, destacar además que las investigaciones que se han logrado encontrar han sido confeccionadas a partir de bibliografía consultada internacionalmente.

### 1.3.3 Limitaciones de los sistemas recomendadores

En muchos de los sistemas recomendadores existen algunos problemas para representar la subjetividad e imprecisión asociadas típicamente a las opiniones o recomendaciones de los usuarios.

Según afirma (9) los sistemas recomendadores poseen una serie de problemas que se deben considerar para su diseño y se describen a continuación:

Macmanus coincide con Resnick en (24) pues plantea algunos indicios de los problemas que deben superar las organizaciones para construir sistemas recomendadores eficaces. Estas son de ámbito general e independiente del sistema que se utilice:

- ✓ Carencia de información (“*Lack of data*”): Los sistemas recomendadores necesitan de mucha información para hacer efectivas sus recomendaciones. Para el funcionamiento de los algoritmos de recomendación, se necesita información de los ítems y los usuarios. No siempre es necesario saber de ambos componentes, pero sí como mínimo de uno de ellos. Cuanta más información, más precisa será la recomendación.
- ✓ Información cambiante (“*Changing data*”): Hace referencia a los contenidos que su uso, modo o costumbre, están de moda. Estos productos representan preferencias de usuarios en espacios de tiempo muy puntuales. El problema de este tipo de ítems radica en que cuando un determinado ítem está “de moda”, recibe gran cantidad de valoraciones muy positivas. Por otro lado, cuando se termina el período de tiempo en que el ítem está “de moda”, los usuarios

---

<sup>1</sup> Conjunto de software educativo que tributa a la formación de una cultura general integral en la educación secundaria.

no desean recomendaciones del ítem; no obstante, el sistema de recomendación los sigue recomendando pues tiene una gran cantidad de votos positivos del ítem (que fueron realizados en el período de tiempo en que el ítem estaba “de moda”).

- ✓ Cambio de preferencias de usuario (“*Changing user preferences*”): Usualmente los usuarios buscan recomendaciones para ellos mismos, pero eventualmente, estos mismos usuarios pueden buscar recomendaciones para otros usuarios de diferente perfil. Por ejemplo: Un usuario determinado “A”, del cual se tiene establecido un perfil con sus preferencias y gustos, le puede interesar buscar un cierto ítem para una persona “B” (por ejemplo para hacer un regalo de cumpleaños) con unas preferencias y gustos totalmente diferentes a los suyos.
- ✓ Ítems impredecibles (“*Unpredictable items*”): Existen una serie de ítems que es difícil recomendarlos pues la reacción del usuario hacia ellos puede ser diversa e impredecible. Por ejemplo, en el caso de alguna película específica. Para la gente, este ítem tiende a ser o muy buena recomendación o muy mala, pero no un término medio.
- ✓ Voto pronto y a menudo (“*Vote early and often*”): Según (9), si cualquiera puede hacer recomendaciones, los propietarios de un ítem pueden hacer recomendaciones positivas de ese ítem y recomendaciones negativas para los ítems de sus competidores. Este aspecto se puede considerar como un ataque a los sistemas recomendadores pues se intenta sesgar las recomendaciones de cierto(s) ítem(s).
- ✓ Complejidad computacional (“*Computer complexity*”): Los sistemas recomendadores tienen una mayor exactitud de sus predicciones cuando cuentan con una mayor cantidad de información disponible. No obstante, a mayor cantidad de información en el sistema, más costosos (en tiempo) son los cálculos para obtener los resultados. Una manera de abordar el problema consiste en la creación de grupos o “clusters” de elementos parecidos de tal forma que los elementos similares se agrupan y se computan como si fueran el mismo.

El hecho de trabajar en sentido general con información con alto grado de incertidumbre ocasiona que los métodos para establecer recomendaciones se encuentren con limitaciones que pueden dificultar la consecución de su objetivo final. A la hora de concebir un SR, es importante no dejar de tener en cuenta estas limitantes.

### 1.3.4 El problema de la recomendación

El problema de la recomendación es reducida al problema de estimar las evaluaciones para los elementos que no han sido vistos por un usuario. Esta estimación se basa en las calificaciones (“*ratings*”) otorgadas por este usuario a otros “ítems”. Una vez que se pueda estimar calificaciones sobre los ítems sin calificación, se puede recomendar a los usuarios los ítems con el más alto rating estimado.

El problema de la recomendación se formula 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, tales como libros, películas o restaurantes, pudiendo ser bien grande 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. Más formalmente [ver (25)]:

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

### 1.3.5 Algoritmos de recomendación

En general, los algoritmos de recomendación (26) deben encontrar la preferencia (“*likeliness*”), que se expresa en una de estas 2 formas:

- ✓ **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  $U_a$ .
- ✓ **Recomendación:** una lista de los  $N$  primeros ítems que el usuario  $U_a$  posiblemente prefiere, no usados aun por  $U_a$ , y se llaman los top- $N$ .

Los algoritmos de recomendación pueden basarse en el usuario (en la memoria), o en el ítem (en el modelo).

- ✓ Los algoritmos basados en el usuario utilizan estadísticas para buscar el conjunto de los usuarios más similares al  $U_a$  (vecinos cercanos, o *nearest neighbors*) y a continuación combinan las preferencias de esos vecinos para producir una predicción o recomendación (top- $N$ ) para  $U_a$ .
- ✓ Los algoritmos basados en los ítems suponen que un usuario estaría interesado en ítems similares a los que le interesaron antes a ese mismo usuario, y buscan entre el conjunto de

ítems que el usuario ha usado y jerarquizado, calculan la similaridad con el ítem objetivo, y seleccionan los  $k$  más similares. También se les llama “basados en el modelo”, pues desarrollan primero un modelo de las opiniones de los usuarios, usando técnicas de aprendizaje automático.

En el SR que se desarrolla se decide utilizar el algoritmo de recomendación debido a que es la forma de recomendación que mejor se adapta a las necesidades del portal y por consiguiente se ajusta de forma clara al dominio del problema que se plantea.

### 1.3.6 Análisis de sistemas homólogos

Luego del estudio de los sistemas recomendadores de noticias disponibles en la web, se llega a la conclusión de que internacionalmente existen sistemas recomendadores similares al que se pretende alcanzar y se obtuvo una visión de cómo manejar algunos de los procesos que se encuentran inmersos dentro de la estructura de los sistemas recomendadores, lo que facilita el trabajo a la hora de tomar ideas para la construcción del SR de noticias.

Algunos ejemplos de sistemas recomendadores existentes en Internet que han servido de base para numerosos estudios relacionados con la recomendación de noticias en portales web son:

- ✓ **Daily Me** (disponible en: <http://www.dailyme.com>) es un servicio personalizado de noticias en línea que permite a los lectores recibir contenidos informativos adaptados a sus intereses. Presenta además, un resumen de las noticias más importantes del día de los principales periódicos, servicios de noticias y otras fuentes superiores.
- ✓ **Reddit** (disponible en: <http://www.reddit.com>). El mismo es un agregador de noticias donde los usuarios pueden dejar enlaces a contenidos web. Otros usuarios pueden votar a favor o en contra de los enlaces, haciendo que aparezcan más o menos destacados.

Ninguno de los sistemas homólogos analizados satisfacen las necesidades planteadas en la situación problémica de la presente investigación, sin embargo, su estudio fue provechoso, ya que se identificaron características y funcionalidades comunes que sirven como base para el desarrollo del SR como por ejemplo, estos sistemas muestran aproximadamente de 5 a 10 noticias recomendadas, y clasifican las noticias por temáticas.

## 1.4 Tendencias y tecnologías actuales a considerar

Al no existir un SR que solucione la situación problemática existente, es preciso desarrollar una nueva solución y para garantizar un proceso de desarrollo satisfactorio se procede entonces a efectuar un estudio de las metodologías de desarrollo de *software*, herramientas y tecnologías actuales, para identificar las apropiadas que serán empleadas en la solución. En el caso de una aplicación web, para completar una implementación eficiente existen básicamente dos estrategias: 1) la utilización de un *framework server-side scripting*<sup>2</sup>, y 2) la utilización de un sistema de gestión de contenidos (CMS). Dado que el Portal Octavitos está desarrollado sobre la base del CMS Drupal, parte del estudio a continuación está sujeto a esta condición y al uso del *software* libre (27).

### 1.4.1 Metodologías de desarrollo de *software*

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un *software* (28). Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo (29). Por tanto, los autores de la presente investigación consideran que una metodología describe una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción del sistema.

Básicamente, las metodologías de desarrollo de *software* suelen dividirse en dos grupos (28): metodologías pesadas (tradicionales) y metodologías ágiles.

Entre las metodologías tradicionales se encuentran: MSF (por sus siglas en inglés, *Microsoft Solution Framework*), RUP (por sus siglas en inglés, *Rational Unified Process*), y sus híbridos.

Generalmente las metodologías tradicionales se caracterizan por (30): 1) requisitos fijados a lo largo de todo el proyecto, 2) basadas en los procesos, 3) procesos muy bien documentados, 4) gestión predictiva de los proyectos, y 5) no siguen ni los principios, ni las técnicas de las metodologías ágiles. Además estas metodologías suelen aplicarse a proyectos fundamentados con un desarrollo orientado a objetos y un equipo de desarrollo grande. Con tales características, a pesar de que han demostrado ser bastante efectivas y necesarias en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo

---

<sup>2</sup> Término se puede traducir como “para la escritura de código del lado del servidor”.

general se obtienen resultados satisfactorios cuando el equipo de trabajo tiene amplia experiencia en su aplicación, este enfoque no resulta ser el más adecuado para proyectos donde el entorno del sistema es muy cambiante y donde se requiere reducir drásticamente el tiempo de desarrollo manteniendo una alta calidad.

Además del enfoque tradicional y ágil, algunos investigadores proponen metodologías de desarrollo de *software* centradas específicamente en el desarrollo web, entre las que se encuentran: *Navigational Development Techniques* (NDT), *Ubiquitous Web Applications* (UWA), *Object Oriented Hypermedia Design Model* (OOHDM) y otras que actualmente son consideradas obsoletas. Estas metodologías presentan algunas deficiencias, por ejemplo: 1) la mayoría de estas propuestas conceden una menor importancia al tratamiento de requisitos, 2) centran su trabajo en las etapas de diseño e implementación, haciendo mucho énfasis en la navegabilidad del usuario, y la apariencia visual de la aplicación, 3) no incluyen demasiada documentación, y 4) algunas de ellas están concebidas para el uso de paradigma orientado a objetos (31).

Por los planteamientos mencionados en los párrafos anteriores, las características propias de la solución y el ambiente de desarrollo (dígase cliente, equipo de desarrollo, tiempo de desarrollo, etc.), se decide desechar la aplicación de las metodologías anteriores en el proceso de desarrollo y se opta por la aplicación de una metodología ágil.

#### 1.4.2 Metodologías ágiles

Las metodologías ágiles están especialmente definidas para el desarrollo de proyectos con requisitos poco definidos o cambiantes y son aplicables a equipos pequeños de trabajo que resuelven problemas concretos minimizando los fallos y los costes. Las metodologías ágiles surgen como una alternativa, o una reacción a las metodologías tradicionales y principalmente a su burocracia (30). Su objetivo principal es crear un producto que funcione y cumpla las exigencias del cliente (contando con su colaboración cliente en todo momento) antes que escribir mucha documentación.

Dentro de las metodologías ágiles se incluyen: *Extreme Programming* (XP), *SCRUM*, *Crystal Methodologies*, *XBreed*, *Adaptive Software Development* (ASD), *Dynamic Systems Development Method* (DSDM), entre otras (32).

Para el desarrollo de la presente investigación se analizaron las metodologías ágiles SCRUM y XP debido a que ambas promueven el trabajo en equipo, fomentan la interacción sistemática entre el cliente y

el equipo de desarrollo y están suficientemente documentadas. Vale destacar además que estas metodologías ágiles permiten la realización de reuniones y entregas al cliente continuamente, en cortos períodos de tiempo.

### 1.4.2.1 SCRUM

La metodología ágil SCRUM<sup>3</sup> se basa fundamentalmente en entregas iterativas con ciclos de corta duración (24-30 días). A estos ciclos la metodología SCRUM los denomina *sprint*. En estos ciclos se desarrollan las mejoras de los productos que se hayan planificado (33).

Según (34) es posible identificar tres fases en SCRUM: 1) planificación del *sprint*, 2) seguimiento del *sprint*, y 3) revisión del *sprint* (ver Figura 2). Estas tres fases pueden encontrarse también en la literatura sobre el tema como: fase antes del juego, fase del juego o desarrollo, y fase después del juego (35).

SCRUM es una metodología ágil principalmente indicada para proyectos con un rápido cambio de requisitos (34), dispone de herramientas para la gestión de cada una de sus fases, y es ideal para pequeños equipos de diez o menos miembros; sin embargo no indica y/o provee de ninguna práctica concreta para el desarrollo de *software* (30). Una comparativa de las metodologías ágiles SCRUM y XP indica que la metodología XP está más enfocada a presentar diferentes prácticas, y SCRUM a la gestión de proyectos (ver Figura 1). Además, a pesar de las ventajas que pudiera suponer la aplicación de la metodología ágil SCRUM, es necesario señalar que los roles que esta propone (propietario del producto, *Scrum Master*, equipo de desarrollo, cliente, y gestor) (30) no están visiblemente representados en el presente proyecto.

---

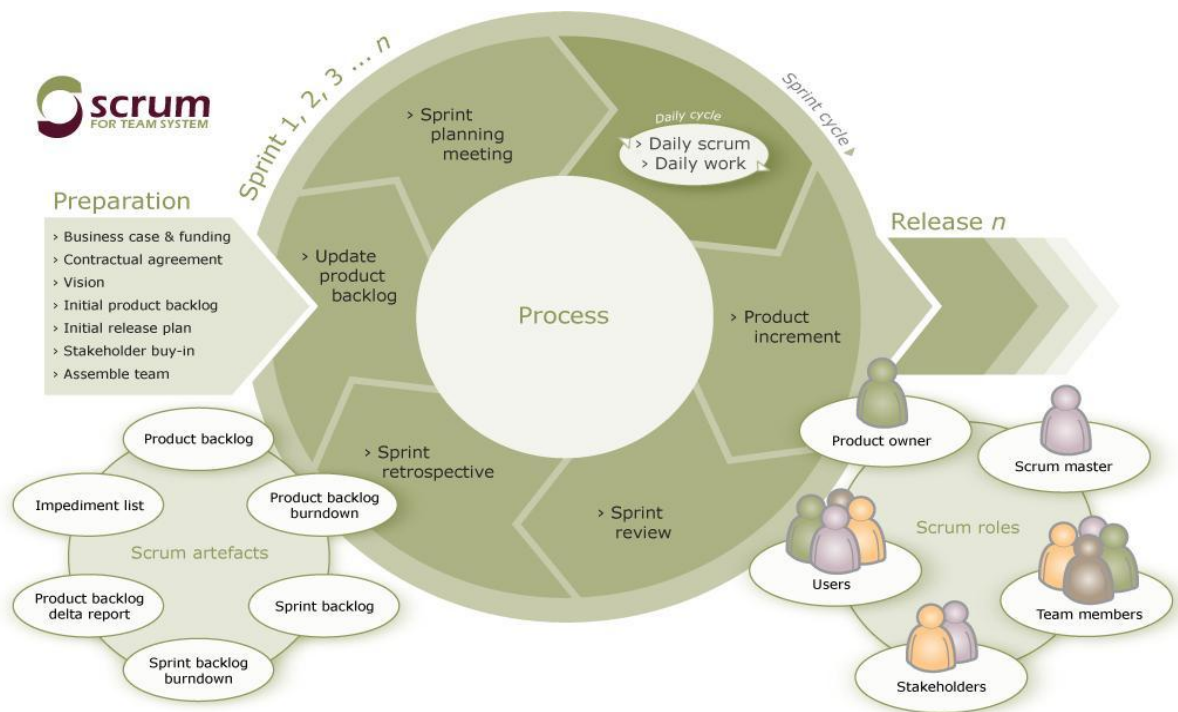
<sup>3</sup> Término en rugby, que hace referencia a como se devuelve un balón que ha salido fuera del campo, al terreno de juego de una manera colectiva, la traducción al castellano sería melé.



		Principio del proyecto	Especificación de requisitos	Diseño	Codificación	Test Unitarios	Test de integración	Test del sistema	Test de aceptación	Mantenimiento
SCRUM	PM									
	PD									
	PA									
XP	PM									
	PD									
	PA									

**Figura 1.** PM = Soporte de la gestión del proyecto. PD = Se describe un proceso en el método que incluye esta etapa. PA = El método describe prácticas, actividades y artefactos para la etapa. El color degradado indica que no se incluye en el método.

Algunos autores (35) proponen aplicar una combinación adecuada de SCRUM y XP debido a que opinan que estas son metodologías ágiles complementarias, pero los autores de la presente investigación consideran que estos híbridos no están suficientemente documentados.



**Figura 2.** SCRUM Ciclo de vida.

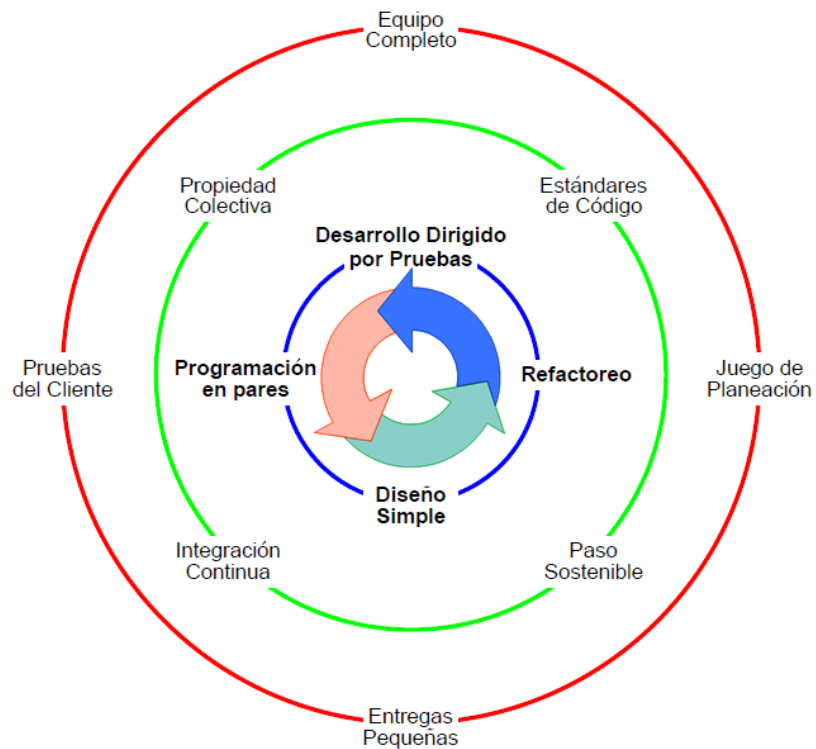
### 1.4.2.2 Extreme Programming (XP)

La Programación Extrema o *Extreme Programming* (XP) es una metodología ágil formulada por Kent Beck en 1996 a partir de un conjunto de principios, prácticas y técnicas que facilitan de manera exitosa la finalización de proyectos, que surgen como respuesta y posible solución a los problemas derivados del constante cambio en los requerimientos (36). Centrada en potenciar las relaciones interpersonales como clave imprescindible para el éxito en el desarrollo de software, la metodología XP promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (32).

Una de las prácticas más significativas que posee la metodología XP es la programación en pares, que proclama que dos personas escriban código en el mismo ordenador, lo que posibilita que parcialmente se puedan evitar y/o reducir errores y malos diseños en la creación del código, pues de esta forma se controla cada línea de código y decisión del diseño instantáneamente. Además, la buena conexión entre ambos desarrolladores generará discusiones que sin duda conducirá a mejores estructuras y algoritmos, aumentando así la calidad del software.

XP no es un conjunto de reglas a seguir, sino más bien un gran abanico de principios, técnicas y prácticas (ver Figura 3), con suficientemente eficiencia probada, que definen una forma de trabajar en armonía con los valores personales y organizacionales, y teniendo su punto de partida en cinco valores fundamentales que debe cumplir el equipo de desarrollo (37):

- ✓ Simplicidad: Se simplifica el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que realmente no necesitamos.
- ✓ Comunicación: La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo.
- ✓ Retroalimentación (*feedback*): Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código.
- ✓ Coraje: El equipo de desarrollo debe estar preparado para enfrentarse a los continuos cambios que se presentarán en el transcurso de la vida del proyecto.
- ✓ Respeto: Los miembros del equipo de desarrollo deben tratarse entre ellos y los demás con dignidad, reconociendo la experiencia de todos y su deseo de éxito.



**Figura 3.** Prácticas comunes de la metodología de desarrollo XP. Las prácticas se refuerzan entre sí.

El ciclo de vida de la metodología XP está compuesto de seis fases (ver Figura 4): *Exploración*, donde los clientes escriben las HU que quieren que sean incluidas en la primera versión, que describen las funcionalidades que serán añadidas al sistema; *Planificación*, donde se establece la prioridad de las HU y se acuerda el contenido de la primera entrega del proyecto, así como su estimación temporal; *Iteraciones*, donde se decide que historias se realizan en cada iteración, así como las pruebas funcionales ejecutadas al final de cada iteración; *Producción*, donde se llevan a cabo un conjunto de pruebas extras, de rendimiento y funcionamiento que son necesarias antes de entregar el producto; *Mantenimiento*, una vez sea liberada la primera versión a los usuarios, el sistema se debe mantener en el entorno de producción siempre y cuando aún hayan iteraciones en fase de producción; *Cierre del proyecto*, donde ya no hay más HU que deban ser implementadas, las necesidades del cliente han sido satisfechas, y otros aspectos como fiabilidad, rendimiento, etc., así como la documentación (38).

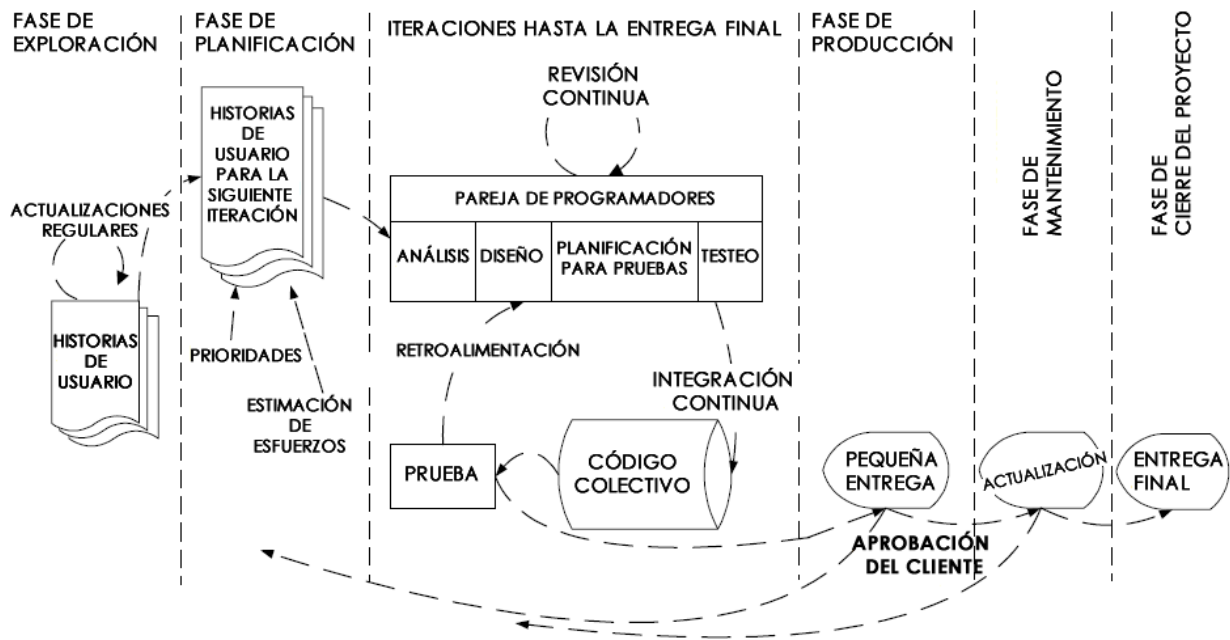


Figura 4. Fases de XP.

XP es difícil de implementar de una sola vez y es recomendable ir escogiendo con cuidado que prácticas aplica a un determinado proyecto (36). Cada práctica debe ser adaptada a los diferentes proyectos y no adaptar un determinado proyecto a las prácticas existentes (39).

En XP, las iteraciones son relativamente cortas ya que entre más rápido se entregue un resultado al cliente, más retroalimentación se obtiene y esto va a representar una mejor calidad del producto a largo plazo. De manera general, esta última característica de la metodología XP es relevante y resulta conveniente para el desarrollo de la presente solución.

### 1.4.2.3 Metodología a utilizar

Luego del análisis anterior de las metodologías ágiles para el desarrollo del *software* y teniendo en cuenta que el presente proyecto es pequeño, con un equipo de desarrollo pequeño y la disponibilidad del cliente, se decide seleccionar XP pues esta metodología propone: 1) una estructura de roles adaptable al equipo de desarrollo, 2) frecuente comunicación entre equipo de desarrollo y cliente, 3) ambiente de desarrollo basado en un único local y ordenador utilizando espacios abiertos, 4) programación por pares, 5) producción de pequeñas entregas funcionales para mostrar un resultado rápido, 6) integración continua, 7) empleo de estándares de codificación, 8) propiedad colectiva del código, y 9) horario de trabajo de un

máximo de 40 horas por semana sin trabajar horas extras. Estas características se ajustan perfectamente a las condiciones reales del presente proyecto.

Además es necesario señalar: 1) que la herramienta a desarrollar formará parte del Portal Octavitos, cuyo desarrollo fue guiado por la metodología XP; por tanto, en aras de continuar una única línea para generar la documentación asociada se decide aplicar la misma metodología, y 2) que el uso de otras metodologías podría traer inconveniencias en el cronograma definido por el proyecto.

### 1.4.3 Sistemas de gestión de contenidos (CMS)

El análisis de la bibliografía refiere a los sistemas de gestión de contenidos (*Content Management System* en inglés, abreviado CMS) como: "...un software que se utiliza principalmente para facilitar la gestión de la web..." (40). Aunque esta definición ha evolucionado en los últimos años, los autores de la presente investigación coincidimos con (40) y concluimos además que son herramientas que permiten crear y mantener una web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores.

Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web (41). Los diferentes CMS existentes se pueden agrupar según varias categorías pero la más usada es según el tipo de sitio que permiten gestionar: genéricos, *blogs*, *wikis*, foros, galerías de imágenes, comercio electrónico. Cada una de estas clasificaciones con características propias responde a necesidades específicas de los usuarios.

Dentro de los CMS genéricos se encuentra incluido Drupal, sobre el cual se basó el desarrollo del Portal Octavitos y que por su flexibilidad y estructura modular es considerado por los autores como potencial para continuar agregándole funcionalidades a este, y de esta forma ideal para el desarrollo de la presente solución, en vez de comenzar a desarrollar sobre una base arquitectónica diferente como pudiera ser otro CMS o un *framework server-side-scripting* como *Symfony*.

Algunas de las ventajas más importantes que hacen útil, necesaria, o se manifiestan con la utilización de un CMS y que se pueden profundizar más en (42), pero que por su relevancia influyen en desarrollo de la presente solución son:

- ✓ Inclusión de nuevas funcionalidades en la web

- ✓ Reutilización de objetos o componentes
- ✓ Páginas interactivas
- ✓ Consistencia de la web
- ✓ Control de acceso

### 1.4.3.1 Drupal 7.15

Drupal es un CMS de código abierto, considerado como un *framework* modular (41) que aplica el patrón arquitectónico Modelo-Vista-Controlador (ver Figura 5), multipropósito y muy configurable, que permite publicar artículos, imágenes, u otros archivos, y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Es de libre distribución, con licencia GNU/GPL, desarrollado y mantenido por una activa comunidad de usuarios. 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 (41).

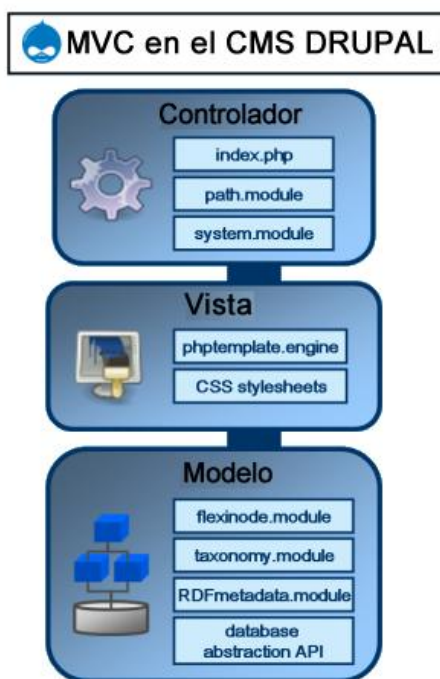


Figura 5. Representación esquemática del patrón MVC en Drupal.

Las principales características del CMS Drupal según (43) son:

- ✓ Ayuda en línea: posee un robusto sistema de ayuda en línea y páginas de ayuda para sus módulos, tanto para los usuarios como para los administradores.
- ✓ Búsqueda: el contenido en Drupal es indexado en tiempo real y se puede consultar en cualquier momento.
- ✓ Código abierto: código fuente libremente disponible bajo los términos de la licencia GNU/GPL.
- ✓ Módulos: las funcionalidades que brinda el CMS se pueden ampliar instalándose nuevos módulos contribuidos que implementen dichas funcionalidades.
- ✓ Personalización: tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario, producto al entorno de personalización implementado en su núcleo.
- ✓ Autenticación de usuarios: se puede registrar e iniciar sesión de forma local, utilizando un sistema de autenticación externo como *Jabber*, *Blogger*, *LiveJournal*, otro sitio Drupal o integrando con un servidor LDAP (*Lighthouse Access Protocol*).
- ✓ Permisos basados en roles: los administradores de Drupal no tienen que establecer permisos para cada usuario, en lugar de eso, pueden asignarse permisos a un rol determinado y agrupar los usuarios por roles.
- ✓ Control de versiones: permite seguir y auditar las actualizaciones del contenido.
- ✓ Enlaces permanentes: todo el contenido creado tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.
- ✓ Objetos de contenido: el contenido creado es funcionalmente un objeto, permitiendo el tratamiento uniforme de la información, con una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios sobre cada objeto.
- ✓ Plantillas: el sistema de temas separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

- ✓ Independencia de la base de datos: aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones, debido a que incorpora una capa de abstracción de base de datos mantenida para MySQL y PostgreSQL.
- ✓ Internacionalización: proporciona opciones para crear un portal multilingüe, pudiendo ser traducido fácilmente utilizando una interfaz Web, importando traducciones existentes o integrando otras herramientas de traducción.

Teniendo en cuenta su nivel de usabilidad, la comunidad de Drupal posee gran conciencia y un alto nivel de motivación, elevando sistemáticamente las prestaciones o funcionalidades del CMS. Como resultado, las capacidades de Drupal se pueden extender con la inclusión de nuevos módulos, temas y perfiles de instalación. En el portal web oficial de Drupal (disponible en <http://drupal.org>) es posible encontrar miles de módulos y temas que proporcionan nuevas características y que transforman la apariencia del sitio (44).

#### 1.4.4 Lenguajes de programación

Un lenguaje de programación “es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes” (45).

En el desarrollo de aplicaciones web, los lenguajes de programación se suelen clasificar como: lenguajes del lado del cliente y del lado del servidor.

##### 1.4.4.1 Lenguajes y tecnologías del lado del cliente

Las tecnologías del lado del cliente son las que se ejecutan en el navegador del usuario, son las páginas dinámicas que se procesan en el cliente. En estas páginas, toda la carga del procesamiento de los efectos y funcionalidades la soporta el navegador (46).

Drupal utiliza el lenguaje XHTML (Lenguaje de Marcado de Hipertexto Extensible) como base para toda la estructura del sistema del lado del cliente dando formato a la información y CSS (*Cascading Style Sheets* en inglés, abreviado como CSS) para cambiar la apariencia visual de los elementos en el diseño



web de manera que se presente de forma estructurada y agradable, e incluye el *framework* JQuery para alcanzar mayor dinamismo mediante el lenguaje JavaScript.

## JavaScript 1.5

El lenguaje de programación JavaScript es utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas, así como las interactividades con el usuario.

Técnicamente es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (47).

Actualmente es utilizado por millones de páginas web para validar formularios, crear cookies, detectar navegadores y mejorar el diseño, su fácil aprendizaje lo hace un lenguaje muy demandado.

JavaScript, como una parte integrada de Drupal, proporciona características dinámicas con la inclusión de una librería potente para los desarrolladores de módulos. Este *framework*, llamado **JQuery** es un tipo de librería de código para facilitar la manipulación del mismo. Es un estándar más rápido y conciso que simplifica el trabajo con los documentos HTML (*HyperText Markup Language* en inglés, abreviado como HTML) y el manejo de eventos, animaciones e interacciones de Ajax para el rápido desarrollo web. JQuery se diseña para cambiar la manera de escribir en JavaScript.

Según Miguel Ángel Álvarez, autor reconocido en el tema, refiere este estándar como un producto que nos simplificará la vida para programar en JavaScript. Nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente (48). Además JQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio.

## XHTML 1.1

El lenguaje XHTML, es la versión XML (*eXtensible Markup Language* en inglés, abreviado como XML) de HTML, por lo que tiene las mismas funcionalidades y características de HTML, además de ser una importante mejora. Al mismo tiempo, utiliza la sintaxis de XML, gana toda la potencia y flexibilidad del mismo y es una base perfecta para CSS.

Los documentos XHTML creados son más flexibles, ya que se adaptan mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, impresoras y dispositivos utilizados por personas discapacitadas (49).

En el desarrollo de la solución, el uso de este lenguaje propone diversas ventajas. La extensibilidad de XHTML mejora las posibilidades de estructuración, eliminación de características repetidas en XML, usabilidad, accesibilidad, mejores formularios, etc. Por todo lo antes expuesto y la flexibilidad que brinda este lenguaje, los autores de la presente investigación coinciden en utilizar esta potente combinación de estándares en el Portal Octavitos para el desarrollo de esta tesis.

Por otro lado, en la actualidad HTML es el lenguaje predominante para el desarrollo de páginas web y su más reciente versión (**HTML5**) se encuentra en constante mejora para aprovechar todos sus beneficios y conseguir una superior experiencia del usuario.

Algunas de las principales ventajas que propone la utilización de HTML5 para el diseño de los sitios web son (50):

- ✓ Mejora de la estructura de las páginas web: utilizando HTML5 es posible mejorar considerablemente la estructura de las páginas de un sitio web con determinadas etiquetas.
- ✓ Contenido multimedia: HTML5 incluye nuevas etiquetas de manera tal que la reproducción de un video en un sitio web se convierta en algo muy sencillo y con una gran diferencia en el rendimiento.
- ✓ Trabajar *offline*: Brinda la posibilidad de trabajar con utilidades web sin necesidad de estar *online*.
- ✓ Limpieza de etiquetas: HTML5 se encarga de limpiar aquellas etiquetas obsoletas. (estilos en las tablas, *<frames>*, *<font>*, *<center>* y *<b>*, entre otras).
- ✓ Sitios web más rápidos: HTML5 trabaja descargando elementos livianos primero e inmediatamente los muestra en pantalla mientras se cargan los de mayor peso. Las páginas cargan mucho más rápido que en un sitio desarrollado en versiones anteriores de HTML.

Por todas las posibilidades anteriormente expuestas que brinda HTML5 se decide emplear esta última versión de HTML para el desarrollo y diseño de la propuesta objetivo de la presente investigación.

## CSS 3

Las hojas de estilo en cascada, conocidos como CSS se define según (51) como “...un lenguaje de hojas de estilo creado para controlar la presentación de los documentos...”, y se utilizan para darle forma y aplicar diseño o presentación al contenido o estructura HTML y XML que forma una página o aplicación web (52). CSS es la mejor forma de separar contenidos de presentación y es imprescindible para la creación de páginas web complejas ya que permite realizar cambios a múltiples elementos dentro del código, agilizando considerablemente así el proceso de cambios.

En los últimos años el código CSS ha evolucionado hasta su tercera revisión (CSS 3), la cual incluye propiedades que permiten alcanzar efectos complejos, facilitando la aplicación, edición y actualización de formato en todo un sitio web.

Continuando con la línea de desarrollo anterior del Portal Octavito se decide optar por CSS como lenguaje para mejorar el diseño visual de la solución si fuera necesario.

### 1.4.4.2 Lenguajes del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

En la presente investigación se analiza el lenguaje PHP pues es el utilizado en Drupal y sus módulos.

## PHP 5.3.5

PHP (*Hipertext Preprocessor*) es un lenguaje del lado del servidor para la generación de HTML. Actualmente, PHP se puede utilizar solo (*standalone scripting language*), pero para Drupal, PHP es utilizado como un poderoso lenguaje para extraer información almacenada en base de datos, aplicar reglas del negocio, e instrucciones para temas, y eventualmente crear HTML que será enviado al servidor (53).

El lenguaje PHP es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. El tiempo de aprendizaje de PHP es muy corto gracias a su potencial y simplicidad (54). Las principales características de PHP son: rapidez, puede ser utilizado en diversos sistemas operativos, servidores web y la característica más potente y destacable de PHP es su soporte

para una gran cantidad de bases de datos (55). Ha evolucionado como lenguaje orientado a objetos permitiendo incluir nuevas funcionalidades, mejoras, y reutilización de código en sitios web (53).

Drupal está escrito puramente en PHP, lenguaje que utiliza arreglos multidimensionales para mucho de su procesamiento, y Drupal continúa con esta metodología (principalmente en sus módulos) (53). Por esta razón es necesario el empleo de PHP en su versión 5.3.5 en la mayor parte de la presente solución.

### 1.4.5 Sistema gestor de base de datos (SGBD)

El estudio de la bibliografía consultada indica que un SGBD (*Data Base Management System* en inglés, abreviado como DBMS:) es un sistema de *software* que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos (56). En estos sistemas se proporciona un conjunto de funcionalidades, procedimientos y lenguajes que permiten a los distintos usuarios realizar tareas habituales con los datos, garantizando además la seguridad de los mismos.

Drupal soporta la integración con diferentes SGBD como: *MySQL*, *SQLite*, *PostgreSQL* y *Oracle*. Esto se consigue mediante una capa de abstracción de la base de datos que convierte las instrucciones genéricas proporcionadas por Drupal en instrucciones particulares de cada base de datos. De esta forma es posible cambiar el gestor de base de datos sin necesidad de cambiar el código de programación de Drupal o de los módulos desarrollados (41). Es por esta razón que de forma convencional, se decide analizar aquí las posibles ventajas que brinda el SGBD MySQL 5.5, que actualmente soporta el almacenamiento de datos en el Portal Octavitos.

#### 1.4.5.1 MySQL 5.5

MySQL en su versión 5.5 es un SGBD relacional y multiusuario con más de seis millones de instalaciones. Actualmente son uno de los mejores sistemas de administración de base de datos y cuenta con mejoras constantes de parte de desarrolladores a nivel mundial. Es un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización [ver (57)] convirtiéndose en una solución altamente viable para la administración de datos.

El SGBD MySQL como solución para la administración de datos del Portal Octavitos involucra:

- ✓ **Velocidad:** es mucho más rápido que la mayor parte de sus rivales.

- ✓ **Funcionalidad:** dispone de muchas de las funciones que exigen la mayor parte de los desarrolladores profesionales. Así mismo, se desarrolla y actualiza de forma mucho más rápida.
- ✓ **Portabilidad:** se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- ✓ **Facilidad de uso:** resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

#### 1.4.6 Entorno de desarrollo integrado (IDE)

En el análisis de la bibliografía consultada se coincide en que: un Entorno de Desarrollo Integrado o IDE (*Integrated Development Environment*) es un programa informático que agrupa diversas herramientas de programación para facilitar la tarea al programador y obtener mayor rapidez en el desarrollo. En otras palabras, es un sistema que facilita el trabajo del desarrollador de software, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, las medidas de rendimiento, la incorporación de las fuentes a un sistema de control de fuentes, etc., normalmente de forma modular (58). Un buen IDE debe incluir las siguientes características:

- ✓ Multiplataforma.
- ✓ Soporte para diversos lenguajes de programación.
- ✓ Integración con sistemas de control de versiones.
- ✓ Reconocimiento de sintaxis.
- ✓ Extensiones y componentes para el IDE.
- ✓ Integración con *framework* populares.
- ✓ Depurador.
- ✓ Importar y exportar proyectos.
- ✓ Múltiples idiomas.
- ✓ Manual de usuarios y ayuda.

Entre los ejemplos de IDE más utilizados en entornos de Linux se encuentran: NetBeans y Zend Studio, los cuales se analizan a continuación:

#### 1.4.6.1 Zend Studio

Zend Studio es uno de los IDE que agrupa todos los componentes de desarrollo necesarios para el ciclo de desarrollo de aplicaciones PHP, disponible para desarrolladores profesionales.

Incluye un conjunto de herramientas de edición, depurado, análisis y optimización, lo que permite simplificar los proyectos complejos. Proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. Consta de dos partes: las funcionalidades de parte del cliente (contiene interfaz de edición y ayuda) y las del servidor (instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración). A pesar de todas estas ventajas se distribuye a través de una licencia comercial.

#### 1.4.6.2 NetBeans 7.14

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, aunque permite programar en diferentes lenguajes (59). NetBeans es una base sólida para desarrollar aplicaciones complejas con un enfoque modular y posee características como la extensibilidad y la escalabilidad.

Se selecciona NetBeans en su versión 7.14 pues ofrece un excelente entorno y proporciona un rendimiento superior para los desarrolladores de PHP, proporcionando editores y herramientas integrales para sus tecnologías relacionadas. Este IDE tiene editores y herramientas para los lenguajes XML, HTML, PHP, JavaScript, etc., debido a que es extensible. También posee un gran soporte para la edición en HTML5, además tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código.

A continuación se resaltan otras de las características de esta herramienta (60):

- ✓ Interfaz de usuario para la personalización de la aplicación.
- ✓ *Framework* para la creación de interfaces de usuario.
- ✓ Internacionalización.

- ✓ Ayudas del sistema.
- ✓ Rendimiento óptimo en tiempo de ejecución y optimización de recursos.

Este IDE posee una larga comunidad de usuarios y desarrolladores mundialmente. Las características mencionadas anteriormente y otras, además de las descripciones de muchas de sus funcionalidades están disponibles en el portal web oficial de NetBeans que se encuentra disponible en: (<http://netbeans.org>).

Las ventajas de usar NetBeans son muchas, a continuación se mencionan solo las más importantes:

- ✓ Tener el respaldo de una empresa tan grande y seria como es Oracle.
- ✓ Poder usar las licencias *open source* para realizar mejoras futuras.
- ✓ Tener un respaldo online por parte de otros programadores.
- ✓ Tener un IDE que soporta el desarrollo de todos los tipos de aplicación Java.

### 1.4.6.3 IDE seleccionado

Para el desarrollo de la propuesta de solución se selecciona NetBeans 7.14 debido a que Zend Studio tiene carácter privativo y además de las características anteriormente mencionadas NetBeans ofrece formato de código acorde a los estándares de código de Drupal, y reconocimiento a los principales tipos de archivo específicos de Drupal (.info, .module .install etc.).

### 1.4.7 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error (61).

Para su correcta ejecución, una aplicación en Drupal necesita alojarse en un servidor web. Según (53), los servidores web recomendados para Drupal son IIS (*Internet Information Server*) y Apache. Teniendo en cuenta que el Portal Octavitos fue desarrollado soportado por un servidor web Apache (27) y que actualmente se encuentra alojado en un servidor web Apache, se decide también que este sea la

herramienta usada para el desarrollo local, hasta tanto el sistema no se considere terminado y se integre con el Portal Octavitos de forma definitiva.

#### 1.4.7.1 Apache 2.2.22

El servidor Apache es un *software* de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Apache es un potente servidor web de software libre cuyo objetivo es servir o suministrar páginas web a los clientes o navegadores que las solicitan.

Entre las principales características de este potente servidor web, se destacan las siguientes:

- ✓ Apache se ha concentrado en la escalabilidad, en la seguridad y en el rendimiento.
- ✓ Es un servidor web altamente configurable.
- ✓ Plataforma de servidores web de código fuente abierto.
- ✓ Trabaja con diversas tecnologías como PHP, *mod\_perl*, *servlets* de Java, etc.

Seleccionamos como servidor web Apache en su versión 2.2.22 debido a que es una tecnología gratuita, de código abierto, coincidiendo con Kabir cuando refiere que “...es la *plataforma de servidores Web de código fuente abierto más poderosa en el mundo...*” (62). Además es compatible con el CMS Drupal, pues se aplica este CMS para la administración del sistema. Consideramos que es la solución perfecta para la mayor parte de los sitios Web y por tanto del portal.

### 1.5 Conclusiones

Luego del estudio y análisis realizado del objeto de investigación del presente trabajo de diploma, apoyado en los métodos de la investigación científicos definidos, se concluye lo siguiente:

- ✓ Se construyó el marco conceptual que soporta la investigación y se adquirió el conocimiento necesario sobre las metodologías, herramientas y lenguajes, seleccionándose las más adecuadas para el cumplimiento del objetivo general propuesto.
- ✓ La bibliografía consultada aportó que la definición y características del objeto de estudio se centran en autores de países foráneos y por consiguiente los sistemas similares existentes abundan mayormente en países extranjeros.



- ✓ Se identificó la metodología XP como guía del proceso de desarrollo, pues provee un marco metodológico integrado por principios, prácticas, y técnicas concretas que se ajustan al entorno y/o condiciones del proyecto, y tienen el potencial de acelerar el tiempo de desarrollo.
- ✓ La integración del CMS Drupal 7.15 soportado por los lenguajes del lado del cliente XHTML, CSS3, y JavaScript, y del lado del servidor PHP 5.3.5, así como del SGBD MySQL 5.5, el IDE NetBeans 7.14, y el servidor web Apache 2.2.22; guiados por la metodología XP, permite continuar parcialmente con la misma línea inicial de desarrollo del proyecto y al mismo tiempo constituye una base sólida para el desarrollo de la solución.
- ✓ El patrón MVC, la capa de abstracción de datos, y la estructura modular presente en el CMS Drupal permite el acoplamiento de nuevas funcionalidades (módulos), lo cual garantiza un proceso de integración seguro de la solución.

# CAPÍTULO 2

## DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

### 2.1 Introducción

En XP no se enfatiza la definición temprana de una arquitectura bien especificada para el sistema. Dicha arquitectura evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. En el libro *Extreme Programming Explained*, Kent Beck plantea: “*la metáfora del sistema es algo que todos los clientes y programadores entienden sobre cómo funciona el sistema. Las metáforas ayudan con la abstracción y modelado del sistema. Encontramos más útil el uso de una metáfora que un modelo, como por ejemplo el modelo de dominio*” (63).

Este concepto es una manera bastante más sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. Por otra parte, Martin Fowler en (64) explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Las palabras usadas para identificar entidades deben estar consistentemente tomadas de la metáfora principal. Mientras avanza el desarrollo del sistema y la metáfora madura, el equipo de desarrollo podrá hallar nueva inspiración a partir de la examinación de la metáfora (63) De esta forma, este conjunto de palabras puede ayudar además a la nomenclatura de clases y métodos del sistema.

En la presente investigación, la metáfora del sistema que se propone es: “un **espacio** donde los usuarios del portal puedan de forma sencilla, acceder y encontrar toda la **información** de su interés relacionada con las temáticas de su preferencia”. La metodología XP sugiere utilizar conjuntamente con la metáfora, otras técnicas importantes de diseño que serán presentadas más adelante en este capítulo como son: historias de usuario y tarjetas CRC (Clase-Responsabilidad-Colaborador) (65). Además, en este capítulo se determinan los servicios que debe brindar la solución y se describen las posibles funcionalidades a través de la exposición de los principales artefactos generados por la metodología en las fases de Planificación y Diseño.

## **2.2 Objetivos del sistema**

La definición de los objetivos del sistema es la primera tarea que se debe realizar en la creación de un sistema informático. Los objetivos que se persiguen con la implementación de la solución son los siguientes:

- ✓ Mejorar los niveles de información que actualmente poseen los usuarios que interactúan con el Portal Octavitos.
- ✓ Brindar mayor divulgación, elección y acceso a las informaciones.
- ✓ Ofrecer a los usuarios las noticias de su preferencia.

## **2.3 Definición de la audiencia**

La audiencia es el público hacia el cual estará orientada la solución y a la que va dirigida la información que se publica en el portal, constituye uno de los pasos fundamentales y elementos más importantes para alcanzar el éxito en los objetivos propuestos. El público es dividido por categorías, determinando las necesidades y expectativas de cada uno. Además, se debe tener en cuenta quiénes son, como categorizarlos, cuáles son sus niveles de conocimiento, cultura, habilidades, formación y necesidades informativas, etc. A continuación se divide la audiencia en varios grupos atendiendo adecuadamente a los usuarios que visitarán el portal.

Los **usuarios potenciales**, serán principalmente todos los estudiantes, profesores y trabajadores de la facultad 4, así como el resto de los miembros de la comunidad universitaria que tenga acceso a la red de la UCI y que por tanto puedan convertirse en usuarios reales.

Y los **usuarios reales**, que son aquellos que visitan el Portal en dependencia de la frecuencia con que lo hagan, pudiendo o no ser de la Facultad 4.

## **2.4 Usuarios relacionados con el sistema**

Los usuarios relacionados con el portal son aquellas personas que de una forma u otra van a interactuar con la aplicación, tanto los que obtienen resultado de valor con los procesos que se ejecutan como los que no obtienen ningún resultado, incluyendo además los que mantienen el portal actualizado y en correcto funcionamiento. En la presente investigación se definieron los siguientes usuarios:

Tabla 1 Usuarios relacionados con el sistema y su descripción.

Usuarios	Justificación
Anónimo	Es la persona que navega por el sistema sin haber creado una cuenta aún, fluye dentro de éste sin privilegios de acceder a los recursos que el portal define para usuarios autenticados.
Usuario autenticado	Es la persona que se encuentra autenticada en el Portal cumpliendo este rol, además tendrá acceso a diferentes módulos y componentes debido al rol que cumple dentro de la aplicación.
Editor	Es la persona encargada de modificar diferentes contenidos en el Portal de acuerdo con la temática que represente.
Administrador	Es la persona autorizada para la gestión del sistema, encargado de actualizar, modificar, eliminar e insertar toda la información del Portal. Dispone de posibilidades ilimitadas para ejecutar todas las funciones administrativas del sistema.

## 2.5 Definición de servicios

Mediante el desarrollo de las funcionalidades definidas el Portal podrá ofrecerle a la audiencia un conjunto de contenidos y servicios con el fin de satisfacer sus necesidades y expectativas, alcanzando los objetivos trazados para la aplicación. Dentro de los servicios que brindará el sistema de recomendación se encuentran:

- ✓ Proponer al usuario un grupo de noticias que no han sido vistas aún por él, pero que han sido jerarquizadas por el usuario de forma explícita e implícita.
- ✓ Brindar un conjunto de evidencias de los usuarios, entre las que podemos citar:
  - ✓ Los grupos de categorías que más visitan los usuarios.
  - ✓ Los usuarios que más interactúan con el portal.
  - ✓ Noticias con mayor nivel de audiencia.

- ✓ Ofrecer un grupo de noticias clasificadas como las más populares, evidencia dejada por las actividades de los usuarios en el portal.

## 2.6 Historias de usuario (HU)

Las HU son la técnica utilizada para especificar los requisitos del *software* tanto funcionales como no funcionales. Las HU son escritas por el cliente en su propio lenguaje, como cortas descripciones de lo que el sistema debe realizar (66). El tratamiento de las HU es muy flexible con cada HU siendo lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

La información de una HU puede variar y ajustarse a las características específicas del proyecto. Por ejemplo, existen varias plantillas sugeridas para representar cada HU, pero no existe un consenso al respecto. A continuación se incluyen los siguientes aspectos esenciales de cualquier proyecto de desarrollo, que ha sido el resultado de consultar las HU presentadas como ejemplo en libros (36) (67):

- ❖ **Número:** Posee el número asignado a la HU.
- ❖ **Nombre de HU:** Atributo que contiene el nombre de la HU.
- ❖ **Usuario:** El usuario del sistema que utiliza o protagoniza a la HU.
- ❖ **Prioridad en el negocio:** Evidencia el nivel de prioridad de la HU en el negocio.
- ❖ **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU.
- ❖ **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 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias. Por lo que cuando el valor de dicho atributo es de 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas.
- ❖ **Iteración asignada:** Precisa la iteración en la que será desarrollada la HU.
- ❖ **Descripción:** Posee una breve descripción de lo que realizará la HU.
- ❖ **Observaciones:** Brinda información extra que se estime agregar para hacer más comprensible la HU.

Para las HU que describen las características y cualidades con que debe cumplir la solución, los autores de la presente investigación decidieron excluir los siguientes aspectos: riesgo de desarrollo, puntos estimados y puntos reales. A continuación se muestran las HU generadas, que representan funcionalidades que serán implementadas.

**Tabla 2 HU Perfil de usuario.**

<b>Historia de Usuario</b>	
No: 1	Nombre: Perfil de usuario
Usuario: Usuario autenticado	
Prioridad en el negocio: Alta	Puntos Estimados:2
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Da la posibilidad al usuario de llenar un formulario con las preferencias de categorías de noticias del portal.	
Observaciones:	

**Tabla 3 HU Generar recomendaciones.**

<b>Historia de Usuario</b>	
No: 2	Nombre: Generar recomendaciones
Usuario: Administrador	
Prioridad en el negocio: Alta	Puntos Estimados: 3
Riesgo de desarrollo: Alta	Iteración asignada: 1
Descripción: Se podrá extraer características de los objetos no conocidos aún por el usuario actual y se comparan con el perfil del mismo para predecir sus preferencias sobre tales objetos.	
Observaciones:	

**Tabla 4 Mostrar listado de noticias recomendadas.**

Historia de Usuario	
No: 3	Nombre: Mostrar listado de noticias recomendadas
Usuario: Usuario autenticado	
Prioridad en el negocio: Media	Puntos Estimados: 2
Riesgo de desarrollo: Alta	Iteración asignada: 2
Descripción: Muestra al usuario el listado de noticias recomendadas de acuerdo a lo que haya definido en su perfil de usuario y las evidencias registradas en la aplicación.	
Observaciones:	

**Tabla 5 HU Mostrar noticias más populares.**

Historia de Usuario	
No: 4	Nombre: Mostrar noticias más populares
Usuario: Usuario autenticado	
Prioridad en el negocio: Baja	Puntos Estimados: 1
Riesgo de desarrollo: Media	Iteración asignada: 3
Descripción: El usuario podrá visualizar las noticias más populares publicadas en el portal.	
Observaciones: Esta funcionalidad se genera si el usuario no tiene creado un perfil previo.	

**Tabla 6 HU Apariencia.**

Historia de Usuario	
No: 5	Nombre: Apariencia
Descripción: El sistema contará con una interfaz amigable, manteniendo el formato en los navegadores.	
Observaciones: El sistema contará con enlaces bien identificados, permitiendo acceder a cualquiera de las secciones con un número mínimo de clics y estará	

optimizado para una resolución de 1024x768.

**Tabla 7 HU Seguridad.**

<b>Historia de Usuario</b>	
No: 6	Nombre: Seguridad
Descripción: La autenticación está garantizada a través de los roles y permisos definidos a cada usuario en el sistema.	
Observaciones: Los usuarios no necesariamente deberán estar autenticados para visualizar algunos de los contenidos que brinda la aplicación. Sin embargo los administradores y moderadores se comportarán como usuarios, pero para poder además, realizar configuraciones de los contenidos o algún cambio en el sistema, tendrán obligatoriamente que estar autenticados. Cada usuario autenticado podrá realizar las operaciones correspondientes con su rol definido en el sistema.	

**Tabla 8 HU Condiciones tecnológicas.**

<b>Historia de Usuario</b>	
No: 7	Nombre: Condiciones tecnológicas
Descripción: Garantizar que el acceso al sistema se realice desde máquinas con características óptimas.	
Observaciones: El sistema debe tener al menos 5 MB de espacio libre en el disco duro. Las restantes condiciones tecnológicas del sistema están sujetas a las condiciones tecnológicas del Portal Octavitos.	

**Tabla 9 HU Condiciones Legales.**

<b>Historia de Usuario</b>	
No: 8	Nombre: Legales
Descripción: Usar herramientas de software libre bajo las licencias GNU/GPL.	



## 2.7 Estimación de esfuerzos por HU

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de uno a tres puntos. Por otra parte, los clientes deciden sobre el ámbito, tiempo de las entregas y de cada iteración (32).

**Tabla 10 Estimación de esfuerzos por HU.**

No	Historia de Usuario	Estimación (semanas)
1	Perfil de usuario	2
2	Generar recomendaciones	3
3	Mostrar listado de noticias recomendadas	2
4	Mostrar noticias más populares	1

## 2.8 Plan de iteraciones

En la metodología XP, la creación del sistema se divide en iteraciones. La duración ideal de una iteración está entre una y tres semanas. Para cada una de las iteraciones se establecen un conjunto de HU definidas por el cliente que serán implementadas en cada iteración del sistema, superando las pruebas de aceptación las cuales son realizadas al final de cada ciclo (65). Al terminar cada iteración la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos.

**Iteración 1:** En la primera iteración se entregarán las funcionalidades que tienen prioridad alta para el cliente correspondiendo a las HU 1 y 2, las cuales son:

Perfil de usuario

Generar recomendaciones

**Iteración 2:** En la segunda iteración se realizarán las restantes HU que son importantes para el cliente siendo esta: la 3, la cual tendrá como funcionalidad:

Mostrar listado de noticias recomendadas

**Iteración 3:** En esta iteración se implementan las HU de baja prioridad para el cliente pero no menos importante que las anteriores para los desarrolladores. La HU es: 4 que se define por:

Mostrar noticias más populares

## 2.9 Plan de duración de las iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en días que demora implementar cada HU. Se tendrá en cuenta además la prioridad que el cliente le asigna a cada historia y el nivel de complejidad que estas poseen.

Tabla 11 Plan de duración de las iteraciones.

Iteración	Orden de las HU	Duración total
1	Perfil de usuario Generar recomendaciones	5 semanas
2	Mostrar listado de noticias recomendadas	2 semanas
3	Mostrar noticias más populares	1 semana

## 2.10 Plan de entregas

En el plan de entregas se realiza un cronograma de entregas donde el cliente establece la prioridad de cada HU, cuáles serán agrupadas para conformar una entrega, el orden de las mismas. En correspondencia, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.) (65). Esta fase dura unos pocos días.

Tabla 12 Plan de entregas.

Historia de usuario	Primera iteración	Segunda iteración	Tercera iteración
Perfil de usuario	V 1.0	Finalizado	Finalizado
Generar recomendaciones	V 1.0	Finalizado	Finalizado
Mostrar listado de noticias recomendadas	-	V 1.0	Finalizado
Mostrar noticias más populares	-	-	V 1.0

### 2.11 Prototipo de interfaz de usuario no funcional

Un prototipo de interfaz de usuario no funcional 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 (68).

Para mostrar una vista preliminar de la solución se creó un prototipo de interfaz no funcional (Ver Figura 6) que básicamente incluye las características definidas de la propuesta de solución, y es de fácil modificación a partir de la retroalimentación del cliente. La propuesta realizada de la solución parte desde la interfaz inicial que presenta el Portal Octavitos y puede estar propensa a posibles cambios dentro de este.



**Figura 6.** Prototipo de interfaz de usuario no funcional de la solución (bloque azul) dentro del Portal Octavitos.

### 2.12 Tarjetas CRC

La metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Cada tarjeta CRC representa una clase, objeto, modulo,

paquete, etc., con su nombre en la parte superior. En la parte inferior izquierda se describen las responsabilidades y a la derecha las clases que le sirven de soporte (65).

La filosofía del CMS Drupal, no es basada en la Programación Orientada a Objeto (POO), pero aun así, utiliza en su diseño varios conceptos o técnicas del paradigma orientado a objeto (“objetos”, “abstracción”, “polimorfismo”, “encapsulación” y “herencia”), adaptando la técnica de las tarjetas CRC a estas características, representando cada tarjeta CRC a un módulo, quedando las responsabilidades definidas como las funciones que realiza y los colaboradores serían los métodos del módulo (69). A continuación se muestran las tarjetas CRC de la propuesta de solución:

**Tabla 13 Tarjeta CRC 1. Recomendador de noticias.**

<b>Módulo: Recomendador de noticias</b>	
<b>Responsabilidades</b>	<b>Colaboraciones</b>
Perfil de usuario	recomendador_configuracion recomendador_configuracion_submit
Generar recomendaciones	recomendador_generador compararFechas buscar
Mostrar listado de noticias recomendadas	mostrarNoticias recomendador_theme
Mostrar noticias más populares	recomendador_generador

**2.13 Conclusiones**

El estudio de la audiencia a la que va dirigida la solución delimitó que existen 4 tipos de usuarios que interactúan con el sistema con diferentes niveles de acceso.

Se definieron 3 iteraciones que abarcan un total de 4 HU (3 críticas y 1 no crítica), que describen los aspectos principales a tener en cuenta para el desarrollo de la solución, contemplando una visión futura de las funcionalidades que debe cumplir el sistema.

Asociado a estas HU se construyó el plan de entregas que enmarcó el tiempo de desarrollo de las HU en 8 semanas, determinando un cronograma que especifica las entregas que deben hacerse y conjuntamente se elaboró el prototipo de interfaz no funcional de la solución y 1 tarjeta CRC que representa las funcionalidades a implementar definiendo las responsabilidades y los colaboradores.

Los principios, prácticas y técnicas que propone la metodología XP aportó los artefactos necesarios que darán paso a la implementación de la solución.

### 3.1 Introducción

La implementación “es un pilar fundamental de la metodología XP” que se realiza durante la fase de producción, a partir del análisis y diseño consecuentes en la fase de iteraciones hasta la entrega final, y donde el equipo de desarrollo mantiene un constante intercambio con el cliente, no solo para aclarar las dudas que puedan surgir sino para formar parte de este.

Sin embargo, a pesar de que en las fases anteriores la metodología propone claramente artefactos basados en técnicas como son las HU, y las tarjetas CRC, para la fase de producción se plantean prácticas concretas como son la programación en parejas, integración continua y utilización de estándares de código, pero no existe una opinión única entre los investigadores sobre cuáles son los artefactos que genera la metodología XP, aunque implícitamente se sugiere que el código desarrollado forma parte de estos [ (32) (36) (44) (67)]. Debido a que el resultado esperado de la implementación de la presente investigación puede considerarse como un subsistema de un sistema mayor como lo es el Portal Octavitos, el código desarrollado introducido se encuentra incluido dentro de un nuevo módulo, que aporta las funcionalidades necesarias para el SR de noticias y que debe ser considerado como el artefacto fundamental obtenido en esta fase. Por lo tanto, al reflexionar en esta consideración, y para obtener un mayor entendimiento sobre este tema fue necesario comprender conceptos elementales en Drupal como son: nodos, bloques, ganchos, temas y menús, lo cual puede consultarse en la bibliografía reciente sobre el tema [ (44) (53) (70)].

Este capítulo está enmarcado en las fases de iteraciones hasta la entrega final y producción que propone la metodología XP (ver Figura 4), por lo tanto, se explica someramente cómo se desarrollaron las funcionalidades del SR, comenzando desde la creación de las tablas añadidas a la base de datos, hasta cómo se implementó el algoritmo fundamental que resuelve la recomendación de noticias del portal octavitos. Para concluir, se analizan los resultados de las pruebas diseñadas para cada una de las HU que se proponen en la fase de Exploración y Planificación.

### 3.2 Fase de implementación

Todas las HU se desarrollan según como fueron planificadas en las iteraciones, por lo cual XP plantea que se verifique el plan de iteraciones para proceder así a la implementación. Todo esto se manifiesta de igual manera con las tareas de cada HU.

Para el presente trabajo se propone realizar tres iteraciones y a continuación se detallan cada una de ellas.

#### 3.2.1 Iteración 1

Para la presente iteración se implementaron las HU de mayor prioridad y a continuación se detallan cada una de ellas.

**Tabla 14 Funcionalidades abordadas en la primera iteración.**

Módulo / componente	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
Recomendador	Perfil de usuario	2	0.98
	Generar recomendaciones	3	2.16

En el Anexo I se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración:

#### 3.2.2 Iteración 2

Para esta iteración se implementaron las HU que tienen prioridad media.

**Tabla 15 Funcionalidad abordada en la segunda iteración.**

Módulo / componente	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
Recomendador	Mostrar listado de noticias	2	2.12

	recomendadas		
--	--------------	--	--

En el Anexo II se muestran las tareas efectuadas para la funcionalidad implementada en esta iteración:

### 3.2.3 Iteración 3

Para esta iteración se implementaron las HU que tienen prioridad baja.

Tabla 16 Funcionalidad abordada en la tercera iteración.

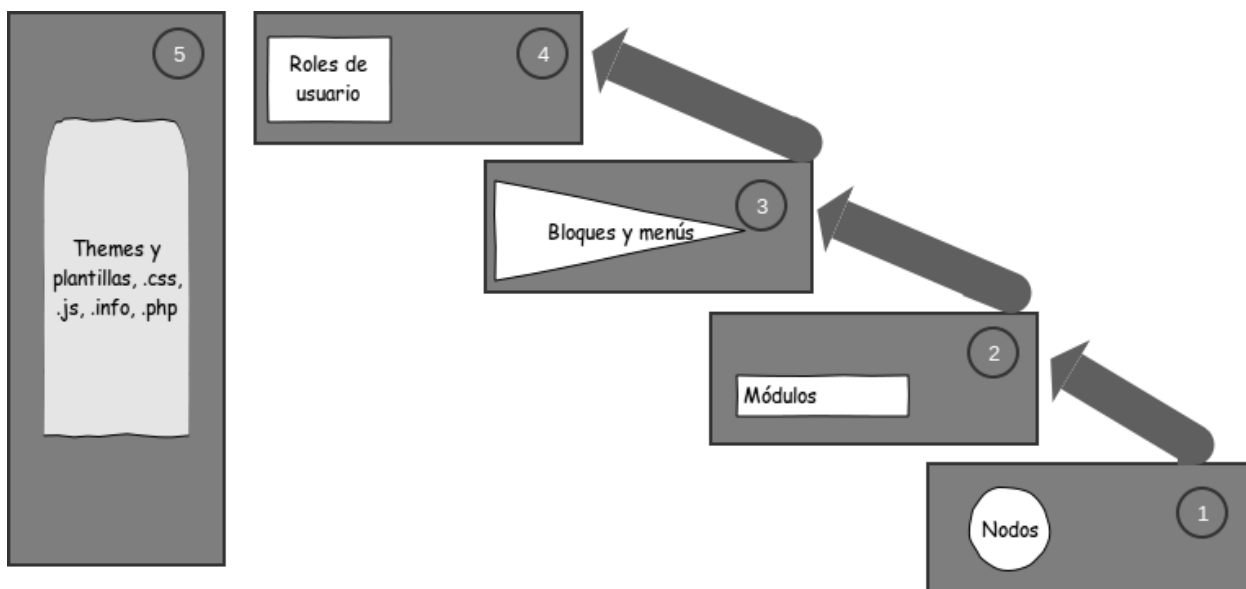
Módulo / componente	Historias de usuario	Tiempo de implementación (semanas)	
		Estimación	Real
Recomendador	Mostrar noticias más populares	1	1.10

En el Anexo III se muestran las tareas efectuadas para la funcionalidad implementada en esta iteración:

### 3.3 Estructura de Drupal

La estructura de capas y datos del CMS Drupal está conformada alrededor de 5 conceptos fundamentales que son los elementos base que mantienen los componentes organizados y flexibles (ver Figura 7):





**Figura 7.** Estructura de capas y datos en Drupal.

El más importante subsistema de Drupal son los nodos (*nodes*), los cuales son los elementos básicos en que Drupal almacena la información o los contenidos. Un nodo puede ser cualquier tipo de contenido que queramos mostrar en nuestro sitio. Básicamente, los nodos tienen una serie de campos de información asociada tales como: Autor, Fecha, Título y Cuerpo del contenido. Además pueden ampliarse con otros campos e incluso extenderse con funcionalidades que provean otros módulos de Drupal.

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

El subsistema de bloques (*blocks*) y menú (*menus*) de Drupal se encargan de la configuración y la visualización de los contenidos. Permiten estructurarlos y organizarlos en la página. Los bloques pueden ser configurados para ofrecer una salida de varias maneras, así como que solo se muestren en ciertas páginas definidas, o solo para cierto tipo de usuarios. Los menús de Drupal son una colección de enlaces a diferentes contenidos del sitio y que se muestran dentro de un bloque en una zona concreta.

La seguridad y control de los usuarios se garantiza con la capa de permisos de usuario (*user permissions*), subsistema que es clave para manejar la integridad de la información almacenada en el sitio. En Drupal podremos crear todos los roles de usuarios que necesitemos y asignarles los permisos sobre las funcionalidades que estimemos oportunos. Lo normal será definir un rol y después activar

mediante un listado de los módulos instalados que permisos tendrán, que podrán hacer y que no podrán hacer en nuestra plataforma.

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

La implementación de la presente investigación involucró el desarrollo de un módulo para la recomendación de noticias en el Portal Octavitos de la Facultad 4.

### 3.4 Creación de las tablas adicionales en la base de datos

Por defecto, la instalación de Drupal crea 74 tablas que son utilizadas por el CMS para almacenar y manejar datos de nodos, usuarios, bloques, menús, contenidos, etc. El diseño de la estructura de las tablas que se agregaron a la base de datos de Drupal es el primer paso para comenzar la codificación de la solución informática. Para completar el desarrollo del SR, y teniendo en cuenta que las nuevas funcionalidades incluyen nuevos contenidos, fue necesario crear 1 tabla en el módulo desarrollado, y utilizar 5 tablas del módulo taxonomía del núcleo de Drupal. La estructura de las tablas se muestran en la Figura 8.

De acuerdo a los resultados arrojados por la investigación para resolver el problema de la recomendación se debe establecer una conexión entre los usuarios y las categorías (módulo *taxonomy*) demostrando que esta relación son las transacciones que establecen entre el usuario y los ítems (noticias). Para el desarrollo de la aplicación se establece una relación de mucho a mucho entre la tabla {Users} (esta contiene la información referente a los usuarios del portal) y la tabla {taxonomy\_term\_data} (esta contiene las categorías de los contenidos) provocando el surgimiento de una nueva tabla {db\_recomendador} (esta contiene la relación de los usuarios con las categorías).

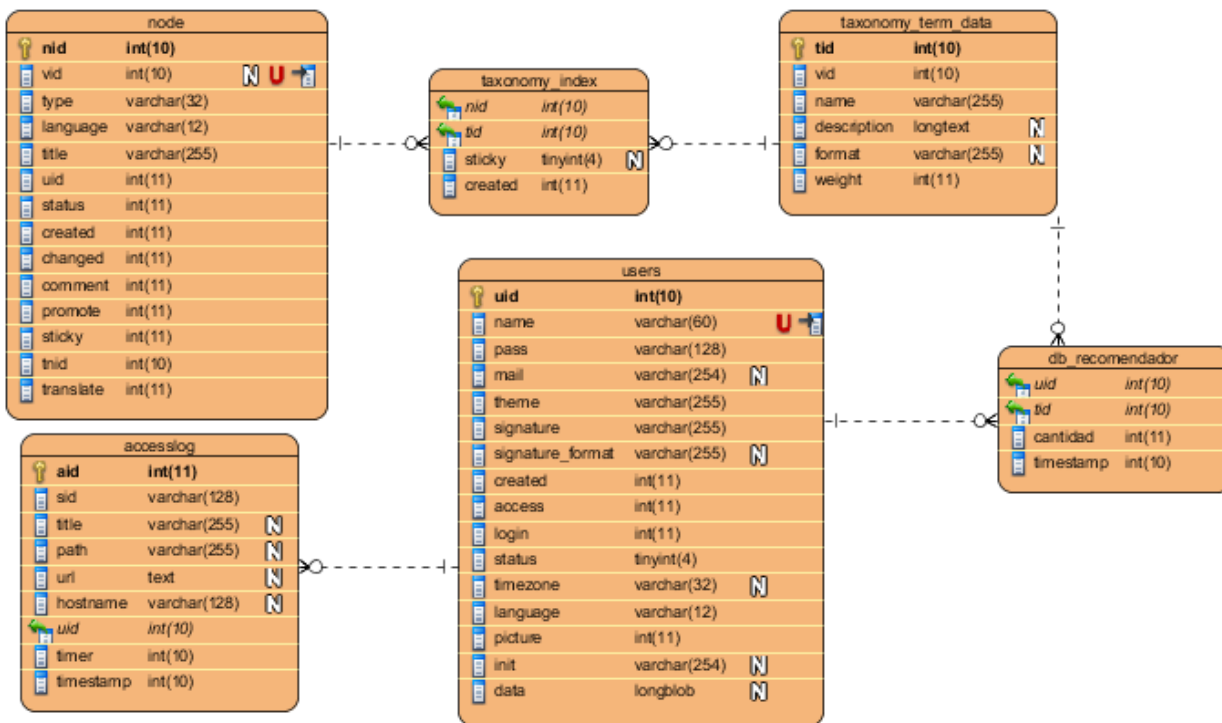


Figura 8. Tablas adicionadas a la base de datos de Drupal.

### 3.5 Módulo desarrollado para el SR

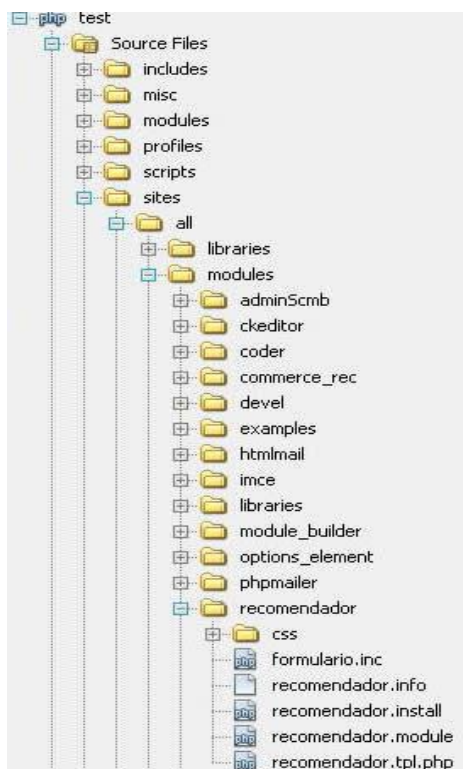
El modulo que se desarrolló lleva como nombre recomendador (ver Tabla 14), dicha tabla representa el artefacto de tareas de ingeniería que proponen algunas bibliografías (63).

Tabla 17 Módulo Recomendador.

Recomendador
<p><b>Propósito del módulo:</b> Extraer características de los objetos no conocidos aún por el usuario actual y se comparan con el perfil del mismo para predecir sus preferencias sobre tales objetos. Mostrar al usuario el listado de noticias recomendadas de acuerdo a lo que haya definido en su perfil de usuario y las evidencias registradas en la aplicación.</p>
<p><b>Tecnologías que utiliza:</b> Este módulo emplea PHP (<i>Hipertext Preprocessor</i>), las de hojas de estilo CCS 3 (<i>Cascading Style Sheets</i>) y el lenguaje de marcado HTML (<i>HyperText Markup Language</i>) en su versión 5.</p>
<p><b>Historias de usuario que implementa:</b> HU- Generar recomendaciones.</p>

**Tablas:** node, taxonomy\_index, users, taxonomy\_term\_data, db\_recomendador, accesslog.

La programación del módulo recomendador se realizó en parejas en una misma estación de trabajo, en un mismo local, y en horario de 9 de la mañana a 5 de la tarde, siguiendo el proceso descrito en los capítulos 2,4, 5, 6, 8 y 11 del libro Pro Drupal 7 Development (70). En una primera etapa de la implementación se creó el archivo con la información de los mismos (.info) y posteriormente otro archivo (.install;) para crear la estructura de las tablas asociadas. En un tercer archivo (.module) se implementaron las funcionalidades críticas. La estructura del módulo se encuentra en el directorio `sites/all/modules/recomendador` desde donde se verificó su ejecución (ver Figura 9).



**Figura 9.** Ubicación del módulo recomendador en la estructura de archivos de Drupal.

Además, es necesario destacar que debido a la manera particular que Drupal establece la escritura de módulos, en la implementación del módulo de la solución están reflejados patrones de diseño creacionales (*Singleton*) que son los que resuelven problemas relativos a la creación de objetos, y patrones de diseño de comportamiento (*Chain of Responsibility*, y *Command*) que resuelven problemas relativos a la interacción entre objetos. En el caso de los patrones de diseño estructurales, los cuales

resuelven problemas relativos a la composición de clases y objetos, no fue necesaria su utilización para el desarrollo del módulo. A continuación se explican estos patrones utilizados directamente en la solución.

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

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, pasando por la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de gancho utiliza este modelo, a fin de que los módulos no tienen que definir cada hook, sino sólo los que vayan a aplicar.

El módulo terminó con 468 líneas. Se utilizaron arreglos asociativos, arreglos bidimensionales, condicionales, consultas a base de datos con diferentes grados de complejidad, ciclos y asignaciones. Entre las funciones más complejas están `recomendador_generador` que es un  $O(n^3)$ . En esta función se utilizaron técnicas de minería de datos que fueron de suma importancia para eliminar ambigüedades e información innecesaria.

Por otra parte, los navegadores Internet Explorer, Chrome y Firefox tienen excelentes herramientas para desarrolladores para prestarle atención a varios aspectos de las páginas web, como los identificadores y clases de los elementos HTML. En este sentido se utilizó la extensión *Firebug* versión 1.11.2 del navegador web Firefox 18 para facilitar el trabajo con las hojas de estilo.

Además, si se desea personalizar la salida actual de HTML es necesario crear o modificar una plantilla. Las plantillas son ficheros de código con extensión `.tpl.php` y nombre fijo (por 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. Entonces Drupal utiliza esa función para crear esa parte de la página.

Por otra parte, XP promueve la programación basada en estándares, de manera tal que el código sea fácilmente entendible por todo el equipo, y facilite la recodificación (36). En ese sentido, se aplicó la práctica (*coding standards*) de la metodología XP que propone la utilización de estándares de código, lo que fortalece la comprensión de la pareja de programadores y hace que el código sea más legible.

En cuanto a la integración de los módulos con el Portal Octavitos, 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 (63).

### 3.6 Técnicas para el filtrado de la información

Los SR aplican técnicas de minería de datos para el filtrado de la información, con los datos de interacción que proveen estos ambientes es posible encontrar indicadores que con la aplicación de estas técnicas se pueda identificar información relevante, para la definición de recomendaciones. Entre estas técnicas se encuentra las reglas de asociación y las secuencias frecuentes.

La regla de asociación consiste en la búsqueda de ítems que aparezcan en los mismos conjuntos de transacciones, para, a partir de ahí establecer reglas que indiquen dependencia entre algunos de los ítems. Morales et al (72) plantearon que una regla de asociación expresa la probabilidad de que la ocurrencia de un evento implique la ocurrencia de otro, planteando la siguiente formalización:

Considérese un conjunto de ítems  $I = \{i_1, i_2, i_3, \dots, i_{n-1}, i_n\}$ , donde cada elemento “ $i$ ” perteneciente a  $I$  puede asumir valores binarios 1 o 0 (verdadero o falso) que expresan respectivamente su presencia o ausencia en el conjunto. Además, sobre los elementos que componen  $I$ , se tiene un conjunto de transacciones  $T = \{t_1, t_2, t_3, \dots, t_{n-1}, t_n\}$ , donde cada elemento “ $t$ ” perteneciente a  $T$  corresponde a un conjunto de ítems presentes en  $I$ , tal que  $t \subseteq I$ . Un ítem es considerado como una instancia de un atributo, o sea, su valor en un determinado registro (u objeto) que represente una transacción.

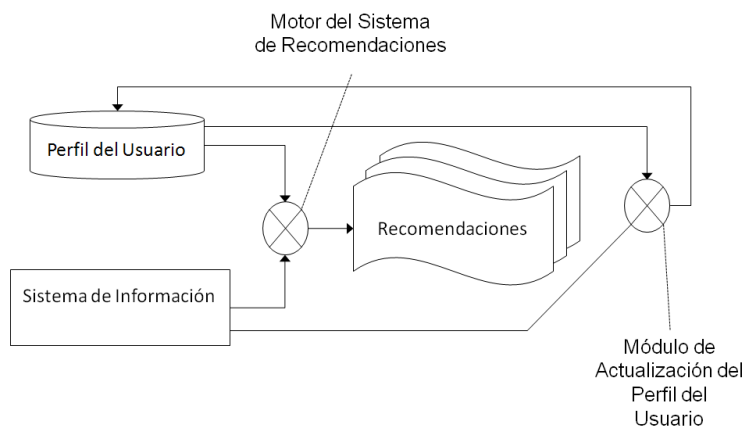
Por otra parte, la técnica de secuencias frecuentes permite descubrir el tiempo de las secuencias ordenadas de URLs que han seguido los usuarios y predecir los futuros. En general, en las bases de datos de transacciones están disponibles los datos en un período de tiempo y se dispone de la fecha en que se realizó la transacción. El descubrimiento de patrones de secuencia (*sequential patterns*) en el Log

puede utilizarse para predecir las futuras visitas y así poder organizar mejor los accesos y publicidades para determinados períodos de tiempo.

Se decide la implementación de un algoritmo basado en reglas de asociación y secuencias frecuentes. De esta forma, se utilizan reglas generadas por el sistema cliente (Portal Octavitos)  $P \rightarrow Q$ , donde P y Q son categorías de noticias que se le sugieren al usuario. Se entiende que si un usuario siente un nivel de preferencia alto por la categoría P entonces lo más probable es que le guste la categoría Q. Se utiliza secuencias frecuentes para obtener la cantidad de visitas a una categoría P y la fecha en que se realizaron estas visitas. Entonces si la cantidad de visitas a una categoría P es mayor que el promedio de visitas a las demás categorías, se propone la recomendación de la categoría Q.

### 3.7 Arquitectura del SR

La arquitectura que guió el proceso de desarrollo de la presente investigación es la que se muestra a continuación (ver Figura 10):



**Figura 10.** Arquitectura del sistema recomendador.

Está compuesta por 3 elementos fundamentales:

- ✓ Perfil de usuario: representa el registro de valoraciones del usuario hacia un grupo de categorías en el sistema de información.
- ✓ Sistema de Información: constituye el cúmulo de objetos (noticias) del cual se va a extraer aquellos que sean de interés del usuario.

- ✓ Recomendaciones: son las sugerencias de objetos extraídos del sistema de información que se la van a mostrar al usuario.

Entre los elementos perfil de usuario, sistema de información y recomendaciones, media el motor de recomendaciones. El cual aplicando técnicas de recopilación de información filtra las preferencias del usuario sobre categorías del sistema de información generando un grupo de objetos a recomendar.

### 3.8 Instalación del módulo

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

### 3.9 Configuración del SR

Luego de instalar el módulo definido, se requiere realizar algunas configuraciones básicas:

- ✓ Establecer los permisos.
- ✓ Activar el bloque en la parte que se desea ver.

Todas estas configuraciones se llevan a cabo mediante las opciones de la interfaz de administración.

Los menús son un conjunto de enlaces que se utilizan para navegar por un sitio web. En Drupal el módulo de menú proporciona una interfaz potente para controlar y personalizar el sistema de menú que viene con el CMS. Los menús se muestran principalmente como una lista jerárquica de vínculos con alta flexibilidad característica de los bloques de Drupal. Cada menú crea automáticamente un bloque del mismo nombre. Por defecto, los nuevos elementos de menú se colocan dentro de un menú integrado y etiquetado de navegación, pero los administradores también pueden crear menús personalizados (73).

En el módulo “recomendador” se activa un menú principal el cual está compuesto por el enlace: “Obtener noticias de su preferencia” siendo habilitado el vínculo inmediatamente que el usuario se autentica.



Esencialmente, un bloque es un trozo de contenido que puede ser habilitada o deshabilitada en un lugar específico en la plantilla de un portal web. Los bloques se colocan normalmente en las barras laterales de la plantilla, el encabezado o pie de página y pueden ser configurados para mostrar nodos de un cierto tipo de contenido solo en la primera página, o según otros criterios (70).

Inicialmente solo se muestran los bloques activos que trae por defecto la instalación del sistema (bloque de autenticación, bloque de búsqueda y bloque de navegación), pero adicionalmente el módulo recomendador crea un bloque en la barra lateral derecha que agrupan un listado de noticias recomendadas a los usuarios ya autenticados, si este no se ha autenticado aún y no ha llenado el formulario de su perfil de usuario, igualmente se le recomendará las noticias más populares.

### 3.9.1 Roles de usuarios

Uno de los puntos esenciales que brinda Drupal es el uso de permisos basados en roles. Este motor permite especificar qué tipos de usuarios tienen la capacidad de crear, eliminar, editar o incluso ver las noticias recomendadas. Con el uso de los permisos basados en roles se interactúa con el contenido del SR mediante distintos niveles de acceso. Estos están definidos mediante un rol que brinda determinados privilegios a sus usuarios. Por ejemplo, los usuarios anónimos pueden tener la capacidad de aportar evidencias para la base de conocimientos del recomendador, solo los usuarios autenticados que han creado su perfil de usuario pueden ver las noticias recomendadas. Y solo los administradores poseen el control total del recomendador y de las noticias que se publican.

Para configurar los permisos de usuarios, el Administrador se dirige al vínculo *Personas\Permisos*, de la interfaz de administración y asigna los privilegios por módulos activados. En el capítulo anterior se definieron 4 roles: anónimo, autenticado, editor y administrador. A continuación se muestran los respectivos permisos en la Figura 11.

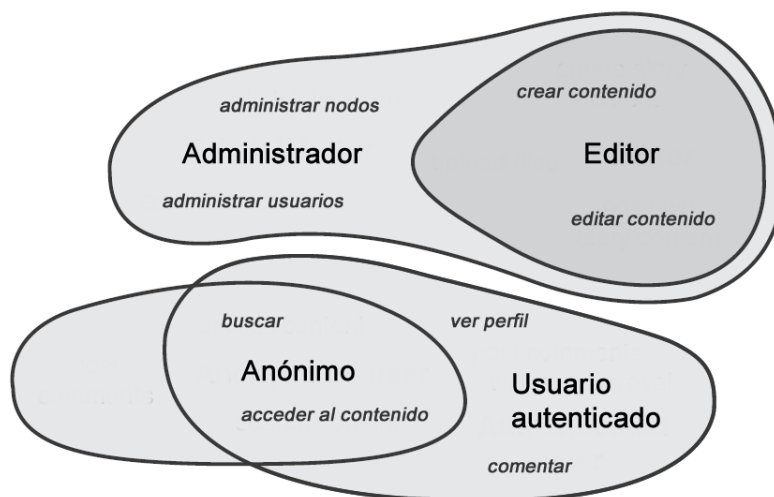


Figura 11. Niveles de acceso por roles.

### 3.10 Pruebas

Probar el código programáticamente es una práctica bien establecida en la industria de desarrollo de software. En Drupal 7, esta es una capacidad incorporada al núcleo de Drupal mediante el módulo *Testing*, que permite a los desarrolladores definir y automatizar las pruebas de un sitio hecho en Drupal. Las pruebas automatizadas se escriben para verificar que un sistema funciona como se espera. Algunas metodologías de desarrollo como XP exigen escribir pruebas antes de escribir el código, lo cual es perfectamente aplicable a módulos de Drupal. Así, utilizando el marco de trabajo que propone *Testing*, los desarrolladores pueden realizar pruebas unitarias y funcionales para validar su código.

Una prueba unitaria es un algoritmo independiente que se encarga de verificar de manera simple y rápida el comportamiento de una parte mínima de código, de forma individual, y sin alterar el funcionamiento de otras partes de la aplicación. En el paradigma de la POO, el enfoque de las pruebas unitarias es a menudo ejercitar todos los métodos de un objeto (o clase). En la programación estructurada, las pruebas unitarias se centran en funciones e incluso, en variables globales. El objetivo es simplemente asegurarse de que cada porción de código (cada unidad) está haciendo su trabajo como se espera.

El módulo *Testing* de Drupal no solo ofrece la posibilidad de hacer pruebas unitarias, sino que también permite realizar pruebas más complejas a todo el sistema, por ejemplo pruebas funcionales. Las pruebas funcionales están diseñadas para verificar que cuando se inserta un fragmento de código en Drupal, funciona como se espera en el contexto de la aplicación. Se trata de una categoría más amplia

que las pruebas unitarias. 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. Por ejemplo, en este contexto las pruebas funcionales a menudo inician Drupal, añaden un usuario, activan módulos, redireccionan páginas y, finalmente, comprueban la salida.

Las pruebas de integración, por otro lado, están más enfocadas a cómo las funciones individuales operan en conjunto para proporcionar un sistema de trabajo. Por ejemplo en Drupal, existen pequeñas funciones, pero existen muchas tareas que requieren de un sistema más amplio para hacer algo antes de que se pueda trabajar.

Para realizar las pruebas se implementaron 5 casos de pruebas. Para el desarrollo de los casos de pruebas automatizados utilizando el módulo Testing se siguió el capítulo 2, del libro *Drupal 7 Module Development*, el capítulo 8 del libro *Pro Drupal 7 for Windows Developers*, y el capítulo 25 del libro *Pro Drupal 7 Development* [ (44) (53) ]. La implementación de los casos de pruebas con *Testing* se basa en el paradigma de la POO, ya que una prueba es esencialmente un objeto que se crea a partir de la clase `DrupalWebTestCase` o `DrupalUnitTestCase`. La mayoría de los casos de prueba se guían por una estrategia simple:

1. Crear una nueva clase que extiende de `DrupalUnitTestCase` o `DrupalWebTestCase`.
2. Agregar la función `getInfo ()`.
3. Realizar cualquier configuración necesaria en el método `setup ()`.
4. Escribir uno o más métodos de prueba, empezando cada método con la palabra `test`.
5. En cada método de prueba, utilizar una o más afirmaciones (`assert`) para poner a prueba los valores reales.

Después de ejecutarse las pruebas y estas hayan finalizado, aparece una pantalla que muestra los resultados, como se muestra en la Figura 12.

MENSAJE	GRUPO	NOMBRE DE ARCHIVO	LÍNEA	FUNCIÓN	ESTADO
Enabled modules: <i>recomendador</i>	Other	recomendador.test	45	recomendadorTestCase2->setUp()	✓
User created with name <i>aOLZ8Hms</i> and pass <i>63pCXMYBDC</i>	User login	recomendador.test	46	recomendadorTestCase2->setUp()	✓
El modulo define 2 permisos	Other	recomendador.test	59	recomendadorTestCase2->testPermission()	✓
Enabled modules: <i>recomendador</i>	Other	recomendador.test	45	recomendadorTestCase2->setUp()	✓
User created with name <i>ySZIKdYc</i> and pass <i>fw7v7iuVc</i>	User login	recomendador.test	46	recomendadorTestCase2->setUp()	✓
El modulo define 2 items en el hook menu	Other	recomendador.test	64	recomendadorTestCase2->testhookmenu()	✓
Enabled modules: <i>recomendador</i>	Other	recomendador.test	45	recomendadorTestCase2->setUp()	✓
User created with name <i>nL0eXR8W</i> and pass <i>QWgMphR8AW</i>	User login	recomendador.test	46	recomendadorTestCase2->setUp()	✓
GET http://localhost/test/?q=user returned 200 (7.15 KB).	Browser	recomendador.test	69	recomendadorTestCase2->testGestionarCategoriaUnprivilegedUser()	✓
Valid HTML found on "http://localhost/test/?q=user"	Browser	recomendador.test	69	recomendadorTestCase2->testGestionarCategoriaUnprivilegedUser()	✓

**Figura 12.** Resultado.

34 pruebas se han ejecutado en 5 minutos y 32 segundos, y todas las pruebas pasan. El intento de hacerlo manualmente sin la automatización de pruebas habría tomado horas con la posibilidad de error humano durante la ejecución de las pruebas.

### 3.10.1 Pruebas de Aceptación

Para probar el sitio, se definen una serie de escenarios para correr a través de la aplicación y documentar los resultados de cada uno. En las pruebas de aceptación se deben especificar uno o varios escenarios para comprobar que una HU ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Los escenarios que se han definido para probar el SR se enumeran en las siguientes tablas:

**Tabla 18** Caso de prueba de aceptación HU1\_P1.

Caso de prueba de aceptación.	
<b>Código:</b> HU1_P1	<b>No. de HU:</b> 1
Nombre: Perfil de usuario	
Descripción: Prueba para la funcionalidad que le permite al usuario llenar un formulario	

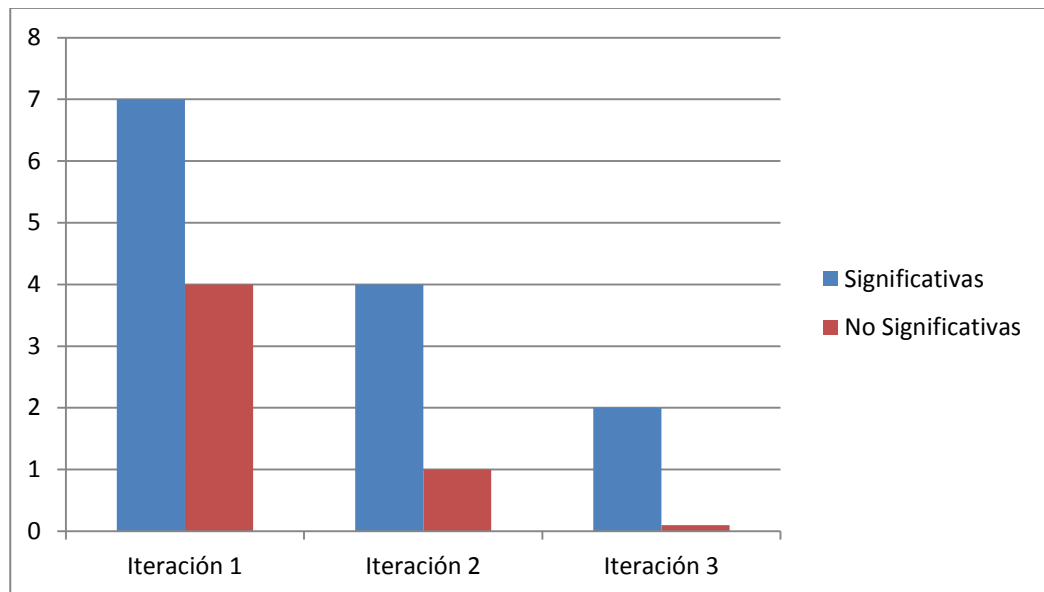
con las preferencias de categorías de noticias del portal.
Condiciones de ejecución: El usuario debe estar autenticado.
Pasos de ejecución: El usuario, después de haberse autenticado, selecciona el menú “Obtener noticias de su preferencia” y seguidamente se le muestran los campos a marcar y procede a insertar los datos requeridos del formulario para completar su perfil de usuario. Finalmente el usuario ejecuta la opción “Enviar” y el sistema guarda los datos registrados en la base de datos.
Resultado esperado: Los datos son registrados y el perfil de usuario se ha creado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Los restantes casos de prueba de aceptación (Ver Anexo IV).

Las pruebas a las funcionalidades se realizaron en 3 iteraciones detectándose para la primera 7 no conformidades significativas y 4 no significativas (Ver Figura 13).

Para la segunda iteración, 4 no conformidades significativas y 1 no significativas.

Para la tercera y última iteración, 2 no conformidades significativas.



**Figura 13.** No conformidades significativas y no significativas.

Las no conformidades (NC), no significativas, se centraron en errores ortográficos como: omisiones de tildes, paréntesis, cambio de mayúscula por minúscula y aceptar letras donde se esperaban valores numéricos, y las significativas, en errores de validación y cambios en el diseño. Las NC encontradas luego de concluida cada iteración de pruebas se analizaron por parte del equipo de desarrollo para luego corregir los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del software.

### 3.11 Técnica de ladov

La técnica de ladov fue creada en 1970 por N. V. Kusmina, empleada para el estudio de la satisfacción en carreras pedagógicas y actualmente aplicada y manejada en diversas ramas investigativas (73). Esta técnica es de suma importancia para conocer el grado de satisfacción del usuario con respecto al recomendador de noticias del Portal Octavitos, lo cual resulta una vía de inestimable valor para el administrador en el estudio de la calidad de la motivación de los usuarios del portal.

Como parte de la presente investigación "Sistema Recomendador de Noticias para el Portal Octavitos" se considera oportuno la aplicación de dicha técnica a través de un cuestionario para el estudio de la satisfacción de los usuarios con respecto al recomendador de noticias del Portal Octavitos.

En esta técnica los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario (preguntas 3, 5 y 9 del cuestionario que aparece en el Anexo V) y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov".

Tabla 19 Cuadro lógico de ladov.

	3. ¿Las noticias recomendadas no se corresponden con sus preferencias?								
	NO			NO SÉ			SI		
9. ¿Te gusta el recomendador de noticias?	5. ¿Considera adecuadas las categorías predefinidas en el perfil?								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	6	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	6	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	3	3	4

El número resultante de la interrelación de las tres preguntas nos indica la posición de cada sujeto en la escala de satisfacción.

La escala de satisfacción es la siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

**Tabla 20 Índice de satisfacción grupal.**

+1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula por la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 4; 5 ó 6; y donde N representa el número total de sujetos del grupo.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

Para medir el grado de complacencia del proceso resultante se tomó una muestra de treinta profesores y estudiantes de la Facultad teniendo en cuenta que los profesores tienen un tiempo de experiencia y uso del Portal Octavitos de más de un año, por otra parte los estudiantes encuestados

forman parte del proyecto al que pertenece el portal, de ellos la gran mayoría son de cuarto y quinto año. La siguiente tabla muestra los resultados de satisfacción individual de los entrevistados:

**Tabla 21 Resultados de la técnica iadov.**

Resultado	Cantidad	%
Total de usuarios de la muestra	30	100
Máximo de satisfacción	25	83.33
Más satisfecho que insatisfecho	3	10
No definida	0	0
Más insatisfecho que satisfecho	2	6.66
Clara insatisfacción	0	
Contradictoria	-	-

**Tabla 22 Variables de la fórmula ISG.**

A	25
B	3
C	0
D	2
E	0
N	30

$$ISG = \frac{25(+1) + 3(+0,5) + 0(0) + 2(-0,5) + 0(-1)}{30}$$

ISG = 0,85 grado de satisfacción alto

Según el valor del ISG obtenido: 0.85, se demuestra que existe un alto grado de satisfacción con la propuesta desarrollada por parte de los usuarios del Portal Octavitos; los mismos reconocen la utilidad del recomendador de noticias desarrollado para satisfacer los gustos y preferencias de los usuarios del portal.

La técnica de ladov contempla además dos preguntas abiertas de gran importancia en la investigación. Dichas preguntas abiertas permiten profundizar en la naturaleza de las causas que originan los diferentes niveles de satisfacción generados por los entrevistados. Estas son:

**Pregunta 4:** ¿Incluiría alguna modificación para el perfeccionamiento del recomendador?

De los usuarios entrevistados algunos no brindaron recomendaciones debido a que consideraban bastante completo el proceso de recomendación; otros, a su vez, aportaron algunas sugerencias, entre ellas:

- Recomendar noticias que a pesar de no estar acorde con el perfil sea de las más leídas.



- Permitir seleccionar la(s) categoría(s) específica(s) que no desea que se le recomiende.
- Incluirle recomendaciones de acuerdo a gustos de amigos míos.
- No se tienen en cuenta noticias que respondan a los intereses del proceso de enseñanza – aprendizaje.
- Cambiar el diseño de la interfaz del bloque de recomendaciones de noticias.

A partir de lo antes expuesto, se sugiere tener en cuenta para futuras versiones del recomendador, el desarrollo de las sugerencias brindadas.

**Pregunta 5:** Considerando que las noticias que se recomiendan están acorde a sus preferencias, ¿qué otras categorías incluiría a los criterios de selección para su perfil?

Gran parte de los usuarios encuestados ofrecieron su afirmación en la interrogante planteada, de ellos propusieron incluirle nuevas categorías al perfil tales como: sexualidad, juegos deportivos, festivales, entre otras.

A partir de la técnica IADOV aplicada quedó mostrada el grado existente en cuanto a la solución desarrollada, destacándose la preferencia por el uso de esta aplicación que ha cumplido con las expectativas esperadas.

#### **Otros aspectos complementarios que se evalúan a través del cuestionario.**

Dicho cuestionario permite evaluar además de la satisfacción con respecto al recomendador a través de la técnica IADOV, aspectos complementarios. Dichos aspectos son: la preferencia por los servicios (preguntas 2 y 7) y cómo ve el usuario al recomendador (pregunta 8). La pregunta 1 es introductoria y es solo de interés en caso de que se aprecie un número significativo de respuestas No y No sé.

#### **Consideraciones finales.**

El diagnóstico de la satisfacción con respecto al recomendador de noticias constituye un punto de partida importante para el diseño de estrategias concretas dirigidas a propiciar el desarrollo de la motivación del usuario por el Portal Octavitos.

### **3.12 Conclusiones**

La descripción de la propuesta de solución en el capítulo anterior permitió desarrollar las funcionalidades descritas a partir de las herramientas, lenguajes y tecnologías seleccionados en el capítulo 1.

Se documentó el desarrollo del módulo recomendador que brinda funcionalidades para erradicar las deficiencias existentes en el Portal Octavitos de la Facultad # 4.

Se validó funcionalmente la solución evaluando que cada funcionalidad estuviera en correspondencia con lo definido inicialmente por el cliente.

Se determinó el grado de conformidad y satisfacción de los usuarios respecto a la propuesta desarrollada.

## CONCLUSIONES

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

- ✓ Los métodos científicos empleados para investigar el objeto de estudio posibilitaron identificar la teorías y los conceptos que sustentan la presente investigación.
- ✓ El estudio que se realizó permitió seleccionar el CMS Drupal 7.15 que proporcionó una base sólida para el desarrollo del sistema, seleccionando como metodología para la guía del proceso a XP, sirviendo como guía de desarrollo generando los artefactos fundamentales para lograr un desarrollo con calidad y poder cumplir con las expectativas del cliente.
- ✓ Las herramientas, lenguajes de programación y tecnologías seleccionadas permitieron desarrollar un SR para satisfacer las necesidades definidas por el cliente, identificando las funcionalidades y características para eliminar los problemas existentes y convertirlo en una herramienta de comunicación.
- ✓ La validación funcional a través de las pruebas demostró el correcto funcionamiento de la solución.

## RECOMENDACIONES

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

- ✓ A la comunidad científica, tomar el presente trabajo como material de estudio en el desarrollo de las funcionalidades similares.
- ✓ Proponemos a la dirección de la Facultad, generalizar el presente trabajo en el resto de las Facultades de la UCI e incluso, en otras universidades del país.
- ✓ A partir del sistema recomendador que se presenta desarrollar un sistema similar basado en filtrado colaborativo.
- ✓ Incorporar nuevas variantes de criterio de selección que respondan a los intereses del proceso de formación y aprendizaje.

## TRABAJOS CITADOS

1. **Adell, Jordi.** Tendencias en educación en la sociedad de las tecnologías de la información. Universidad de las Islas Baleares : EDUTEC. Revista Electrónica de Tecnología Educativa, 1997. Núm. 7. ISSN: 1135-9250.
2. **Castro, Francisco Guerrero.** Blog Institucional monografias.com. *La importancia de las nuevas tecnologías de la información.* [En línea] 2012.  
<http://www.monografias.com/trabajos12/itecdein/itecdein.shtml>.
3. **Consejo Editorial Facultad 4.** *Perfil Editorial Portal Octavitos.* La Habana : s.n., 2011.
4. **ISO, 9241-11.** *Ergonomic requirements for office work with visual display terminals (VDT)s – Part 11 Guidance on usability.* Londres : s.n., 1998.
5. **Peis, E., Morales-del-Castillo, J. M. y Delgado-López, J. A.** Hipertext.net. *Sistemas de Recomendación Semánticos. Un análisis del estado de la cuestión.* [En línea] Universidad de Granada, 2008. [Citado el: 22 de Noviembre de 2012.] <http://www.upf.edu/hipertextnet/numero-6/recomendacion.html>.
6. **Gómez García, J. C.** Portales de internet: concepto, tipología básica y desarrollo. *El profesional de la información.* 2001. Vol. vol. 10, no 7,p. 4-13.
7. **Ezquerro, Norberto, et al.** *Interacción hombre-máquina y usabilidad: Diseño centrado en el usuario.* Universidad de Coruña : s.n., 2008.
8. **Cobos, J. S.** Estilo. *La personalización web, santo grial del comercio electrónico.* [En línea] Fundéu BBVA, 14 de Diciembre de 2011. [Citado el: 7 de Febrero de 2013.]  
<http://www.manualdeestilo.com/?s=La+personalizaci%C3%B3n+web%2C+santo+grial+del+comercio+electr%C3%B3nico>.
9. **Resnick, P., Varian, H.R. y Editors, Guest.** *Recommender Systems. Communications of the ACM.* 1997. 40 (3),56-89.
10. **Pascual-Miguel, F., Chaparro-Peláez, J. y Fumero-Reverón, A.** Presente y futuro de los sistemas recomendadores en la web 2.0. s.l. : Revista internacional de Información, Documentación, Biblioteconomía y Comunicación, 2011. Vol. 20, 6.
11. **Schafer, J.B., J.A., Konstan y J., Riedl.** *E-commerce recommendation applications. Data Mining and Knowledge Discovery.* 2001. pp.115-153.
12. **Ricci, F., Rokach, L. y Shapira, B.** *Introduction to Recommender Systems Handbook.* 2011. ISBN 978-0-387-85819-7.
13. **Resnick, M.L. et al.** *Persuasive Design Through Intelligent Recommendation Systems. Proc. of the Human Factors and Ergonomics Society Annual Meeting.* 2004. pp. 1503-1507.
14. **Nieto Galán, S.M.** *Filtrado Colaborativo y Sistemas de Recomendación.* Madrid : s.n., 2007.
15. **Peinado Rodríguez, S.A., y otros, 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 Noviembre de 2012.] <http://www.buenastareas.com/ensayos/Dise%C3%B1o-e-Implementaci%C3%B3n-De-Un-Software/2714517.html>.
16. **Matos, V.L.** *Sistemas de Recomendación y Personalización.* Universidad Nacional del Sur : s.n., 2006.
17. **Drachler, 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.
18. **Manouselis, N. y Costopoulou, C.** Analysis and Classification of Multi-Criteria. 2007. Vol. 10. págs. 415-441.

19. **Sinbad2, Grupo de investigación.** Personalización y los Sistemas de Recomendación. [En línea] 13 de Marzo de 2012. [Citado el: 28 de Noviembre de 2012.] <http://sinbad2.ujaen.es/cod/archivosPublicos/presentaciones/sistemasDeRecomendacion.pdf>.
20. **Burke, R.** Hybrid recommender systems: Survey and experiments. Estados Unidos : Kluwer Academic Publishers, 2002. Vol. 12. p. 331-370.
21. **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.
22. **Yera Toledo, R.** Concepción y desarrollo de un sistema de recomendación para jurados online de programación. *UCI.Dirección de Información*. [En línea] Junio de 2010. [Citado el: 2 de Noviembre de 2012.] [http://repositorio\\_institucional.uci.cu/jspui/handle/ident/TD\\_02825\\_10](http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_02825_10).
23. **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.
24. **MacManus, Richard.** 5 Problems of Recommender Systems. *ReadWrite*. [En línea] SAY Media, Inc., 28 de Enero de 2009. [Citado el: 13 de Diciembre de 2012.] [http://readwrite.com/2009/01/28/5\\_problems\\_of\\_recommender\\_systems#feed=/search?keyword=Problems%20of%20Recommender%20Systems](http://readwrite.com/2009/01/28/5_problems_of_recommender_systems#feed=/search?keyword=Problems%20of%20Recommender%20Systems).
25. **Adomavicius, G. y Tuzhilin, A.** *Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. 2005.
26. **Saboya Vargas, Aniceto.** *Uso de Recomendadores, Asistentes y Ayudantes en sistemas Tutores*. Catalunya : Depto. LSI-U. Politécnica de Catalunya.España, 2005.
27. **Rodríguez Rodríguez, Y. y Berroa Arias, O. E.** *Solucion de las limitaciones del Portal Octavitos para apoyar la gestión de la información y el trabajo colaborativo en la Facultad 8*. La Habana : s.n., 2010.
28. **Piattini Velthuis, M. y al., et.** Alarcos. [En línea] Universidad de Castilla La Mancha, 2013. [Citado el: 14 de Enero de 2013.] [alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf](http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf).
29. **Avison, D.E. y Fitzgerald, G.** *Information Systems Development: Methodologies, Techniques, and Tools*. s.l. : McGraw-Hill, 1995.
30. **Carvajal Riola, J.C.** *Metodologías ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial*. Barcelona : Tesis de maestría, 2008.
31. **Suárez Jorge, Alinoet .** *Estudio Comparativo sobre 11 Metodologías de Desarrollos Web*. La Habana : s.n., 2009.
32. **Letelier, P. y Penadés, M<sup>a</sup> C.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia : Universidad de Valencia, 2008. Vol. 05, 26. ISSN 1666-1680.
33. **Arellano Moncayo, Fabricio Gerardo.** Space Escuela Politécnica Nacional. *Desarrollo e implantación del sistema de reservación de laboratorios para el Laboratorio de la Facultad de Ingeniería de Sistemas*. [En línea] 3 de Mayo de 2011. [Citado el: 7 de Marzo de 2013.] <http://bibdigital.epn.edu.ec/handle/15000/3824>.
34. **Palacio, J.** *Flexibilidad con Scrum, principios de diseño e implantación en campos Scrum*. s.l. : SafeCreative, 2007.
35. **Schwaber, K. y Beedle, M.** *Agile Software Development with Scrum*. s.l. : Upper Saddle River, Prentice Hall, 2002.
36. **Beck, Kent.** *Embracing Change with Extreme Programming*. s.l. : Addison Wesley Longman, Inc, 1999. ISBN-10: 0201616416 | ISBN-13: 978-0201616415.
37. **Shore, James y Warden, Shane.** *The Art of Agile Development*. Estados Unidos : O'Reilly Media, Inc., 2008. ISBN-13: 978-0-596-52767-9.

38. **Riola, Jose Carlos Carvajal.** *Metodologías Agiles:Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial.* Barcelona : UPC - Barcelona, 2008.
39. **Grenning, J.** *Launching Xp at a Process-Intensive Company.* s.l. : IEEE Software, 2001. Vol. 18.
40. **Serrano-Cobos, Jorge.** Evolución de los sistemas de gestión de contenidos (CMS). Del mainframe al open source. 2007. Vol. 16, 3. p. 213-215.
41. **Gil, F.** *Experto en Drupal 7.* s.l. : Forcontu S.L., 2011.
42. **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 2012.] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>. ISSN: 1696-3296.
43. **Hispano, Drupal.** Drupal Hispano. [En línea] 11 de Abril de 2005. <http://drupal.org.es/caracteristicas>.
44. **Butcher, M., y otros, y otros.** *Drupal 7 Module Development.* Birmingham,Mumbai : s.n., 2010. ISBN 978-1-849511-16-2.
45. **Eguíluz Pérez, J.** *Introducción a Javascript.* 2008.
46. **Padilla Matos, Reynier.** Programación Web y Tecnologías Informáticas. [En línea] 28 de Febrero de 2008. [Citado el: 14 de Enero de 2013.] <http://zenkius.blogspot.com/2008/02/tecnologias-del-lado-del-cliente.html>.
47. **Pérez Eguíluz, Javier.** *Introducción a JavaScript.* Madrid : Autoedición, 2009.
48. **Alvarez, Miguel Angel.** *Manual de jQuery.* 2009.
49. **Pérez, J.F.** *Introducción a XHTML.* s.l. : Serie Científica, 2012. Vol. vol. 6, no 1.
50. **Studio, Ideadelivery.** Ideadelivery Studio. [En línea] 2011. [Citado el: 14 de Enero de 2013.] [http://www.ideadeliveryla.com/5\\_ventajas\\_del\\_HTML5\\_para\\_el\\_dise%C3%B1o\\_de\\_sitios\\_web\\_nota-65](http://www.ideadeliveryla.com/5_ventajas_del_HTML5_para_el_dise%C3%B1o_de_sitios_web_nota-65).
51. **Rivera, E.A., Zamora, R.G. y Soria, M.G.** *Sistema de Educación a Distancia.* s.l. : IV Congreso de Tecnología en Educación, 2012.
52. **De Luca, Damián y De Luca, A.A.** *CSS3 & HTML5. Los nuevos estándares para el diseño y desarrollo web.* [En línea] 2 de Septiembre de 2012. [Citado el: 14 de Diciembre de 2012.] <http://css3html5.com.ar/>.
53. **Travis, B.** *Drupal 7 for Windows Developers.* s.l. : Apress, 2011.
54. **Converse, Tim, Park, Joyce y Morgan, Clark.** *PHP5 and MySQL Bible.* Indiana : Wiley Publishing,Inc, 2004. ISBN: 0-7645-5746-7.
55. **KadaSoftware.** KadaSoftware. [En línea] Universidad Tecnologica de la Mixteca, 2013. [Citado el: 14 de Enero de 2013.] <http://www.kadasoftware.com/index.php/tecnologias-de-desarrollo.html>.
56. **Asenjo Sánchez, Jorge.** *Sistemas Gestores de Bases de Datos.* España : Creative Commons, 2009.
57. **Gilfillan, Ian.** *La Biblia de MySQL.* España : Kluwer Academic, 2003.
58. **González Barahona, J.M., Seoane Pascual, Joaquín y Robles, Gregorio.** *Introducción al software libre.* [En línea] 2003. [Citado el: 10 de Enero de 2013.] <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/sec-ide.html>.
59. **NETBEANS.ORG.** Bienvenido a NetBeans y [www.netbeans.org](http://www.netbeans.org), Portal del IDE Java de Código Abierto. [En línea] 2011. [Citado el: 20 de Noviembre de 2012.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
60. **Rea Inca, Paúl Fernando.** *Diseño y Desarrollo de un Prototipo de Control mediante SMS para Casas Inteligentes.* Sangolquí : s.n., 2012.

61. **Duque González, Raúl.** Mundo geek. [En línea] 2011. [Citado el: 9 de Enero de 2013.] <http://mundogeek.net/etiqueta/servidor-web/>.
62. **Kabir, Mohammed J.** *La Biblia Servidor Apache 2.* s.l. : Anaya Multimedia, 1999. ISBN: 8441514682.
63. **Beck, K.** *Extreme Programming Explained.* s.l. : Addison-Wesley, 2000.
64. **Fowler, Martin.** *Who Needs an Architect?* s.l. : IEEE SOFTWARE, 2003.
65. **Bajo de Luque, M.J. y et.al.** Metodologías ágiles. *Agile Alliance.* [En línea] 2011. [Citado el: 23 de Febrero de 2013.] <http://metodologiasagiles.herobo.com/index.php/es/2011-12-05-16-09-55/metodologia-xp>.
66. **Beck, Kent y Andres, Cynthia.** “*Extreme Programming Explained—Embrace Change*”. s.l. : Second Edition, Addison Wesley, 2005.
67. **Wake, William C.** *Extreme Programming Explored.* s.l. : Addison-Wesley, 2000.
68. **Oktaba, Hanna.** Scribd. [En línea] Agosto de 2010. [Citado el: 26 de Febrero de 2013.] [es.scribd.com/doc/111600953/25/Prototipo-de-interfaz-de-usuario](http://es.scribd.com/doc/111600953/25/Prototipo-de-interfaz-de-usuario).
69. **Chaffer, Jonathan.** Drupal. *Drupal programming from an object-oriented perspective.* [En línea] 29 de Junio de 2012. <http://drupal.org/node/547518>.
70. **Tomlinson, Todd y VanDyk, John K.** *Pro Drupal 7 Development .* s.l. : 3rd.Edition Apress, 2010. ISBN-13 (electronic): 978-1-4302-2839-4.
71. **Reyes, Sady C. Fuentes y Lobaina, Marina Ruiz.** Minería Web: un recurso insoslayable para el profesional de la información. [En línea] Acimed , 2007. [http://bvs.sld.cu/revistas/aci/vol16\\_4\\_07/aci111007.html](http://bvs.sld.cu/revistas/aci/vol16_4_07/aci111007.html).
72. **Morales, Cristóbal Romero, Soto, Sebastián Ventura y Martínez, Cesar Hervás.** *Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web.* Córdoba : Departamento de Informática y Análisis Numérico, 2005.
73. **Buytaert, Dries.** Drupal. *Working with Menus.* [En línea] 23 de Enero de 2013. [Citado el: 14 de Mayo de 2013.] <http://drupal.org/documentation/modules/menu>.
74. **Powell, M.J.D.** *Approximation theory and methods.* Cambridge : Cambridge university press, 1981.
75. **Jensen, P.A.** Operations Management / Industrial Engineering. *Forecasting Theory.* [En línea] 2004. [Citado el: 4 de Abril de 2013.] <http://www.me.utexas.edu/~jensen/ORMM/omie/operation/unit/forecast/>.
76. **Fodor, J.A.** *Concepts: Where cognitive science went wrong.* Oxford : Clarendon Press, 1998.
77. **MANNING, C.D., RAGHAVAN, P. y SCHÜTZE, H.** *Introduction to information retrieval.* Cambridge : Cambridge University Press, 2008.
78. **MURTHI, B.P.S. y SARKAR, S.** The role of the management sciences in research on personalization. *Management Science.* 2003. Vol. 49, 10. p. 1344-1362.
79. **CHANDUKALA, S.R., KIM, J. y OTTER, T.** *Choice Models in Marketing.* s.l. : Now Publishers Inc, 2008.
80. **Corporation, LinkedIn.** LinkedIn. *Web Mining.* [En línea] 2013. [http://www.linkedin.com/skills/skill/Web\\_Mining](http://www.linkedin.com/skills/skill/Web_Mining).



## GLOSARIO DE TÉRMINOS

**Ajax:** *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

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

**API:** Del inglés *Application Programming Interface* - Interfaz de Programación de Aplicaciones es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**Artefacto:** Un artefacto es un producto tangible resultante del proceso de desarrollo de software.

**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.

**CMS:** *Content Management System*, sistema que facilita la gestión de contenidos en todos sus aspectos: creación, publicación y presentación.

**Código Abierto:** Es una tendencia internacional del desarrollo de software que profesa la

**CSS:** Las hoja de estilo en cascada (en inglés *Cascading Style Sheets*) contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llama “cascada” porque una hoja puede heredar los formatos definidos en otra hoja de forma de que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación de un Sitio Web.

distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

**Entorno de desarrollo integrado (IDE):** Programa compuesto por un conjunto de herramientas para un programador que puede dedicarse en exclusiva a un sólo lenguaje de programación o bien puede utilizarse para varios. Consisten en un editor de código, un compilador, un depurador y un GUI.

**Frameworks:** Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

**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.

**HTML:** *Hipertext Transfer Protocol* (Protocolo de transferencia de hipertexto), es el método más común de intercambio de información en la *World Wide Web*, mediante el cual se transfieren las páginas web a un ordenador.

**HTTP:** *HyperText Transfer Protocol* (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la *World Wide Web*.

**JavaScript:** Lenguaje de programación interpretado, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Actualmente todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web.

**Licencia GNU/GPL:** *Free Software Foundation* (Licencia Pública General). Es una licencia creada por la y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

**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.

**PHP:** PHP (*Hypertext Preprocessor*) es un lenguaje multiplataforma, multiparadigma, script (no se compila para conseguir códigos máquina sino que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código) para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código.

**TIC:** Tecnologías de la Información y la Comunicación, se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

**UCI:** Universidad de las Ciencias Informáticas.

**Unix:** Sistema operativo atribuido a Ken Thompson y comercializado por la empresa ATT en la década de los 70 que alcanzó mucho éxito, sobre todo en las universidades y posteriormente en las empresas. Entre sus principales características tenemos que es: portable, robusto, y flexible actualmente goza de gran popularidad dentro de la tecnología de Internet.

**Web:** Es un sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

**WWW (World Wide Web):** O simplemente Web, es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano.

**XHTML:** *Extensible HyperText Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje pensado para sustituir a HTML como estándar para las páginas web.

**XML:** *Extensible Markup Language* (Lenguaje de marcado extensible), conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes.

## ANEXO I.

## Tareas de ingeniería para la iteración 1.

Tabla 23. Tareas de ingeniería para la primera iteración.

Tarea	
No. de tarea: 1	No. de HU: 1
Nombre: configuración 1 del perfil de usuario.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.4
Fecha inicio: 10/12/2012	Fecha fin: 12/12/2012
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se crearán los archivos .module, .tpl.php, .info, .admin, .inc, .test para la creación e implementación de las funcionalidades.	
Tarea	
<b>No. de tarea: 2</b>	<b>No. de HU: 1</b>
Nombre: creación de la tabla en la base de datos correspondiente a la funcionalidad perfil de usuario.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.8
Fecha inicio: 13/12/2012	Fecha fin: 17/12/2012
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se creará un archivo de tipo .install donde se crea la tabla "db_recomendador" para posibilitar la persistencia de los datos.	
Tarea	
<b>No. de tarea: 3</b>	<b>No. de HU: 1</b>
Nombre: construcción de la funcionalidad perfil de usuario.	

Tipo de tarea: desarrollo	Puntos de estimación: 0.8
Fecha inicio: 07/01/2013	Fecha fin: 11/01/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se modificará el código creando la funcionalidad perfil de usuario que le permitirá al usuario crearse su perfil. Se utiliza la integración de los hooks: hook_menu, hook_form y hook_submit para posibilitar el resultado esperado.	
<b>Tarea</b>	
<b>No. de tarea: 4</b>	<b>No. de HU: 1</b>
Nombre: configuración 2 de la funcionalidad perfil de usuario.	
Tipo de tarea: configuración	Puntos de estimación: 0.4
Fecha inicio: 14/01/2013	Fecha fin: 16/01/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se configurará la interfaz gráfica, permitiendo el acceso a la base de datos para actualizar los campos correspondientes.	
<b>Tarea</b>	
<b>No. de tarea: 5</b>	<b>No. de HU: 1</b>
Nombre: obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.2
Fecha inicio: 17/01/2013	Fecha fin: 18/01/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se obtendrá el código fuente del medio de almacenamiento donde se encuentre. Se crearán mecanismos que posibiliten la modificación de la estructura del código.	
<b>Tarea</b>	

<b>No. de tarea: 1</b>	<b>No. de HU: 2</b>
Nombre: configuración de generar recomendaciones.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.4
Fecha inicio: 21/01/2013	Fecha fin: 23/01/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se crearán los archivos .module, .tpl.php, .info, .admin, .inc, .test para la creación e implementación de las funcionalidades.	
<b>Tarea</b>	
<b>No. de tarea: 2</b>	<b>No. de HU: 2</b>
Nombre: creación de la tabla en la base de datos correspondiente a la funcionalidad generar recomendaciones.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.8
Fecha inicio: 24/01/2013	Fecha fin: 28/01/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se creará un archivo de tipo .install donde se crea la tabla "db_recomendador" para posibilitar la persistencia de los datos.	
<b>Tarea</b>	
<b>No. de tarea: 3</b>	<b>No. de HU: 2</b>
Nombre: construcción de la funcionalidad generar recomendaciones.	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Fecha inicio: 29/01/2013	Fecha inicio: 08/02/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se modificará el código creando la funcionalidad generar recomendaciones que permitirá recomendarle al usuario las noticias de su preferencia a partir de su perfil y	

las evidencias dejadas en el portal. Se utiliza la integración de los hooks: hook_theme, hook_help, hook_block_info, hook_permission, hook_block, hook_block_view, entre otros para posibilitar el resultado esperado.	
<b>Tarea</b>	
<b>No. de tarea: 4</b>	<b>No. de HU: 2</b>
Nombre: obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.4
Fecha inicio: 11/02/2013	Fecha inicio: 13/02/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se obtendrá el código fuente del medio de almacenamiento donde se encuentre. Se crearán mecanismos que posibiliten la modificación de la estructura del código.	

## ANEXO II.

## Tareas de ingeniería para la iteración 2.

Tabla 24. Tareas de ingeniería para la segunda iteración.

Tarea	
<b>No. de tarea: 1</b>	<b>No. de HU: 3</b>
Nombre: configuración 1 de mostrar listado de noticias recomendadas.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.4
Fecha inicio: 14/02/2013	Fecha fin: 18/02/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se crearán los archivos .module, .tpl.php, .info, .admin, .inc, .test para la creación e implementación de las funcionalidades.	
Tarea	
<b>No. de tarea: 2</b>	<b>No. de HU: 3</b>
Nombre: creación de la tabla en la base de datos correspondiente a la funcionalidad mostrar listado de noticias recomendadas.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.4
Fecha inicio: 19/02/2013	Fecha fin: 21/02/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se creará un archivo de tipo .install donde se crea la tabla "db_recomendador" para posibilitar la persistencia de los datos.	
Tarea	
<b>No. de tarea: 2</b>	<b>No. de HU: 3</b>
Nombre: construcción de la funcionalidad mostrar listado de noticias recomendadas.	
Tipo de tarea: desarrollo	Puntos de estimación: 1.8



Fecha inicio: 22/02/2013	Fecha inicio: 06/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se modificará el código creando la funcionalidad mostrar listado de noticias recomendadas que le permitirá al usuario visualizar las noticias recomendadas a partir de su perfil de usuario y las evidencias dejadas en el portal. Se utiliza la integración del hook_theme para posibilitar el resultado esperado.	
<b>Tarea</b>	
<b>No. de tarea: 3</b>	<b>No. de HU: 3</b>
Nombre: obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.2
Fecha inicio: 07/03/2013	Fecha inicio: 08/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se obtendrá el código fuente del medio de almacenamiento donde se encuentre. Se crearán mecanismos que posibiliten la modificación de la estructura del código.	
<b>Tarea</b>	
<b>No. de tarea: 4</b>	<b>No. de HU: 3</b>
Nombre: configuración 2 de la funcionalidad mostrar listado de noticias recomendadas	
Tipo de tarea: configuración	Puntos de estimación: 0.4
Fecha inicio: 11/03/2013	Fecha fin: 13/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se gestionarán los archivos incluidos sobre esta funcionalidad para el acceso a la base de datos y la presentación del listado de noticias recomendadas en la interfaz gráfica.	

## ANEXO III.

## Tareas de ingeniería para la iteración 3.

Tabla 25. Tareas de ingeniería para la tercera iteración.

Tarea	
<b>No. de tarea: 1</b>	<b>No. de HU: 4</b>
Nombre: configuración de mostrar noticias más populares.	
Tipo de tarea: configuración - desarrollo	Puntos de estimación: 0.4
Fecha inicio: 14/03/2012	Fecha fin: 18/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se especificarán los datos que mostrará el módulo recomendador. Se crearán los archivos .module, .tpl.php, .info, .admin, .inc, .test para la creación e implementación de las funcionalidades.	
Tarea	
<b>No. de tarea: 2</b>	<b>No. de HU: 4</b>
Nombre: creación de la tabla en la base de datos correspondiente a la funcionalidad mostrar noticias más populares.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.4
Fecha inicio: 19/03/2013	Fecha fin: 21/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se creará un archivo de tipo .install donde se crea la tabla "db_recomendador" para posibilitar la persistencia de los datos.	
Tarea	
<b>No. de tarea: 3</b>	<b>No. de HU: 4</b>
Nombre: construcción de la funcionalidad mostrar noticias más populares.	
Tipo de tarea: desarrollo	Puntos de estimación: 1

Fecha inicio: 22/03/2013	Fecha inicio: 29/03/2013
Programador responsable: Franklin Mustelier Ramírez – Yamilé Peña Cruz	
Descripción: se modificará el código creando la funcionalidad mostrar noticias más populares que le permitirá al usuario visualizar las noticias más populares publicadas en el portal. Se utiliza la integración del hook_theme para posibilitar el resultado esperado.	
<b>Tarea</b>	
<b>No. de tarea: 4</b>	<b>No. de HU: 4</b>
Nombre: obtención del código fuente.	
Tipo de tarea: desarrollo	Puntos de estimación: 0.2
Fecha inicio: 01/04/2013	Fecha inicio: 02/04/2013
Programador responsable: Franklin Mustelier Ramírez	
Descripción: se obtendrá el código fuente del medio de almacenamiento donde se encuentre. Se crearán mecanismos que posibiliten la modificación de la estructura del código.	

## ANEXO IV.

## Casos de prueba de aceptación.

Tabla 26. Caso de prueba de aceptación HU2\_P2.

Caso de prueba de aceptación.	
Código: HU2_P2	No. De HU: 2
Nombre: Generar recomendaciones	
Descripción: Prueba para la funcionalidad que le permite al administrador generar las recomendaciones.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ul style="list-style-type: none"> <li>✓ Verificar si el usuario que esta autenticado tiene un perfil creado.</li> <li>✓ Si no se ha conectado en el día, se analizan las evidencias dejadas desde su última conexión hasta el momento, y se actualiza su perfil de usuario.</li> </ul>	
Resultado esperado: Los datos son actualizados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 27. Caso de prueba de aceptación HU3\_P3.

Caso de prueba de aceptación.	
Código: HU3_P3	No. De HU: 3
Nombre: Mostrar listado de noticias recomendadas	
Descripción: Prueba para la funcionalidad que le permite al usuario visualizar el listado de noticias recomendadas.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: Si el usuario ya tiene creado su perfil, automáticamente el sistema le muestra el listado de las noticias recomendadas. En caso contrario se mostrará el escenario de perfil de usuario (caso de prueba HU1_P1) mostrándole un mensaje de confirmación para seguidamente visualizar el listado de noticias recomendadas.	
Resultado esperado: Que la aplicación muestre el listado de noticias recomendadas	

correctamente.
Evaluación de la prueba: Prueba satisfactoria.

**Tabla 28.** Caso de prueba de aceptación HU4\_P4.

<b>Caso de prueba de aceptación.</b>	
<b>Código: HU4_P4</b>	<b>No. De HU: 4</b>
Nombre: Mostrar noticias más populares.	
Descripción: Prueba para la funcionalidad de mostrar noticias más populares.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: La aplicación debe mostrar en el bloque lateral derecho del portal las noticias más populares en caso de que no se haya creado un perfil de usuario.	
Resultado esperado: Que la aplicación muestre las noticias más populares.	
Evaluación de la prueba: Prueba satisfactoria.	

**ANEXO V.****Cuestionario realizado para evaluar la satisfacción con respecto al recomendador de noticias del Portal Octavitos.**

Lee con cuidado cada pregunta antes de responder. En este cuestionario NO TIENES QUE PONER TU NOMBRE. Te agradecemos tu participación y tu franqueza al decirnos honestamente lo que piensas sobre lo que te preguntamos.

**1. ¿Te gusta el Portal Octavitos? Sí\_\_ No\_\_ No sé\_\_**

**2. ¿Cuáles son los tres servicios que más te gustan?**

1. Recomendador\_\_ 2. Octavitos 4x4\_\_ 3. El tiempo\_\_ 4. Cartelera TV\_\_ 5. Cartelera cultural\_\_  
6. Menú del comedor\_\_ 7. Efemérides\_\_ 8. Buscador\_\_ 9. Votar\_\_ 10. Comentar\_\_ 11.  
Otros\_\_\_\_\_

**3. ¿Las noticias recomendadas no se corresponden con sus preferencias?**

Sí\_\_ No\_\_ No sé\_\_

**4. ¿Incluiría alguna modificación para el perfeccionamiento del recomendador?**

**5. ¿Considera adecuadas las categorías predefinidas en el perfil?**

Sí\_\_ No\_\_ No sé\_\_

**6. Considerando que las noticias que se recomiendan están acorde a sus preferencias, ¿qué otras categorías incluiría a los criterios de selección para su perfil?**

**7. ¿Cuáles son los tres servicios del portal que más te disgustan?**

1. Recomendador\_\_ 2. Octavitos 4x4\_\_ 3. El tiempo\_\_ 4. Cartelera TV\_\_ 5. Cartelera cultural\_\_  
6. Menú del comedor\_\_ 7. Efemérides\_\_ 8. Buscador\_\_ 9. Votar\_\_ 10. Comentar\_\_ 11.  
Otros\_\_\_\_\_

**8. ¿Cómo es el recomendador de noticias del Portal Octavitos? Marca con una X la respuesta seleccionada.**

Bueno		Malo	
-------	--	------	--

**9. ¿Te gusta el recomendador de noticias?**

- Me gusta mucho  
 Me gusta más de lo que me disgusta  
 Me da lo mismo  
 Me disgusta más de lo que me gusta  
 No me gusta nada  
 No sé qué decir