



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 4

*Sistema Recomendador de ejercicios físicos para
estudiantes de la Universidad de las Ciencias
Informáticas*

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

Autores

Arianna Mesa Gómez

Ernesto Marcelino Haughton Samada

Tutores

MsC. Víctor Michel Hernández Gómez

Ing. Addiel Rodríguez Guadarrama

Co-tutores

MsC. Arlenys Palmero Ortega

Osvel Vargas Torres

*La Habana, Junio de 2013
"Año 55 de la Revolución"*



Declaración de autoría

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de junio de 2013.

Arianna Mesa Gómez

Firma del Autor

Ernesto Marcelino Haughton Samada

Firma del Autor

MSc. Víctor Michel Hernández Gómez

Firma del Tutor

Ing. Addiel Rodríguez Guadarrama

Firma del Tutor

MSc. Arlenys Palmero Ortega

Firma del Co-tutor

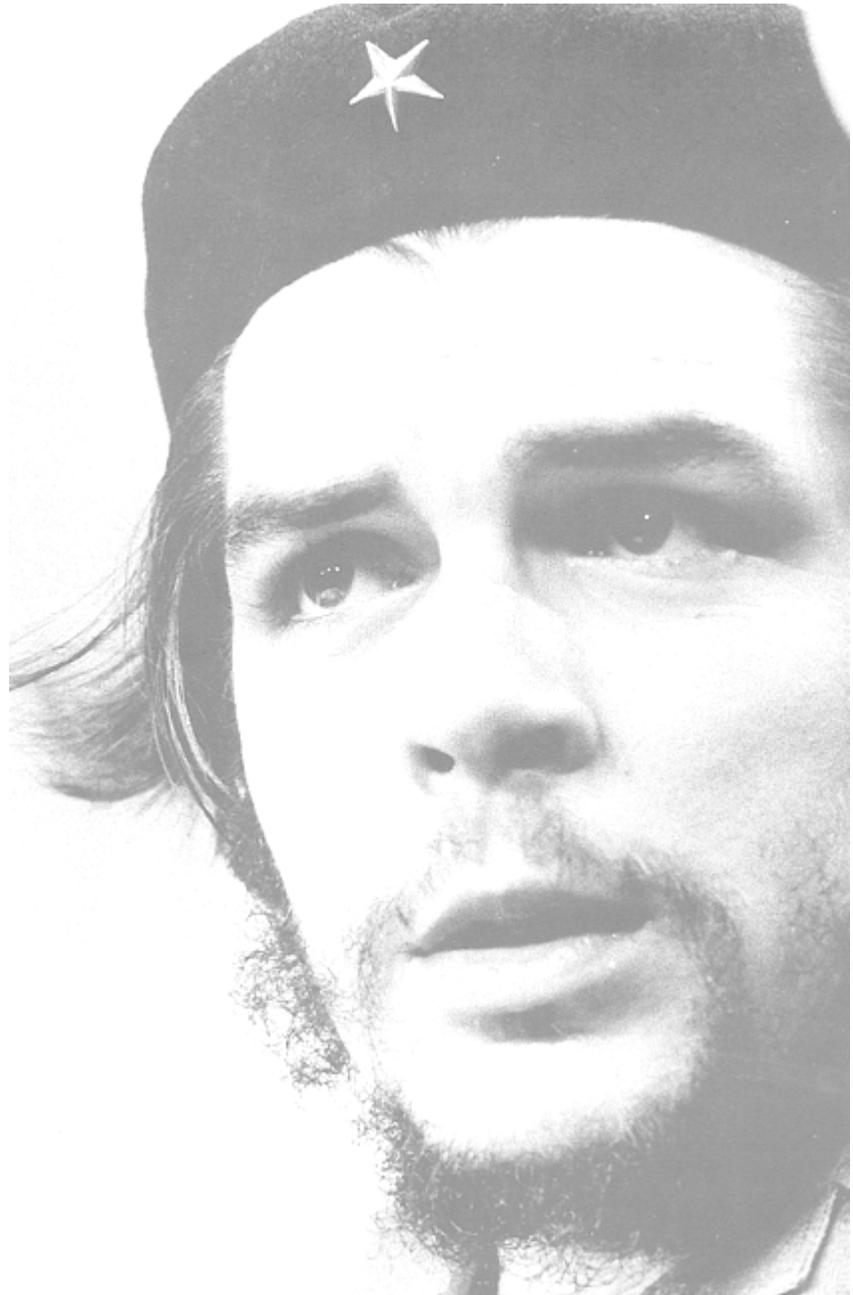
Ing. Osvel Vargas Torres

Firma del Co-tutor



*"La única lucha que se
pierde es la que se
abandona."*

*"Si el presente es de
lucha, el futuro es
nuestro."*





Agradecimientos

Se me acaba la UCI y es ahora cuando me toca agradecerle a todos aquellos que han formado parte de mi vida durante estos maravillosos años, a todos los que aportaron su granito de arena, esto va para ustedes.

A mi mamita gracias por haber estado siempre conmigo apoyando mis decisiones, aceptando mis malcriadeces, cuidando de mí como si todavía fuera una niña, por la confianza que siempre has depositado en mí, por ser más que mi madre mi gran amiga, te adoro.

A mi papá por impulsarme siempre a ser mejor, por escucharme y aconsejarme; por todo lo que me quieres y me malcrías.

A mi Dayi por ser la lucecita de mi vida, la cosita que llegó para cambiarme el mundo, por ser mi mayor inspiración, por ti lucho cada día con más fuerzas para siempre poder darte todo lo que necesites, por ti me he preparado para siempre tener una respuesta a tus preguntas, sabes que eres la persona que más amo en el mundo.

A mi tía Cary por ser como una madre para mí, te debo mucho en la vida.

A mi novio, por todo tu amor, tu comprensión, por llenarme de ilusiones, por tantos buenos momentos, por tu apoyo en todo este tiempo, sabes que has sido un factor fundamental en este camino que he recorrido a tu lado, te adoro con la vida.

A mis suegros Chava y Julio por haberme aceptado como un miembro más de su familia, por sus consejos y sus sobremesas.

A mis hermanas Vero y Mile por tantos años de amistad incondicional.



Agradecimientos

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

A mis amigas del alma Danae y Zoe por esta amistad que se mantiene a pesar de la distancia.

A mis amigos incondicionales de primer año: a Jessy que hoy más que mi amiga es parte de mi familia, gracias por tantas locuras juntas; A Randy, por todo el cariño que siempre me diste y a Linet, gracias por tus repasos y tu dedicación.

A los amigos que vinieron con el tiempo, a Nora por ser mi amiguís, por las meriendas, por el café y todo lo demás. A Yunior, a Heydi, a Darien, los quiero.

A mi compañero de tesis, mi negri del alma, si no fuera por ti hoy no lo estaríamos logrando.

A mis tutores, en especial a Arlenys por todo su tiempo, esfuerzo, paciencia y dedicación.

A los profes del proyecto en especial a Arlan, que se convirtió en un gran amigo, sabes que te debo mucho.

A los profes que más me ayudaron en estos años: Lara, Yudanis, Rosalba, Amado, Yuneiris, en fin a todos, siempre se queda alguien por mencionar.

A todos aquellos que de alguna forma influyeron en que hoy esté logrando este sueño, muchas gracias.

Arianna



Agradecimientos

Primero antes que todo quiero dar las gracias a Dios por haberme dado la sabiduría para graduarme, a mi mamá por darme todo su amor y cariño, a mi padre su fuerza para seguir adelante y la ayuda económica, a la Revolución por haberme dado diecisiete años de estudios gratuitos, a mi compañera de tesis Arianna que sin ella la tesis no hubiera salido con la calidad requerida, a mis tutores por haberme apoyado y aclarado las dudas cuando los necesitaba, a Liliana por siempre estar presente cuando la necesitaba ya sea como amiga o para ayudarme cuando no entendía algo de Drupal, a mis compañeros de cuarto por haberme soportado desde primer año y pasar con ellos buenos ratos, en especial Alexey y Basulto con los que mejores relaciones he tenido, a mi grupo de tecktonik, al piquete del fútbol, a mi equipo de atletismo, a mis compañeros de proyecto, a mi barbero Eduardo, a mis niñas del 141101 Anabel, Ivis, Anay, Aleyda y en especial a la persona que más he querido en la UCI a mi amiga más que amiga, mi hermana Adia.

Nesty



Dedicatoria

De Arianna:

A mi hermana para que le sirva de inspiración en la vida.

A mi mamá para que recoja el fruto de sus esfuerzos y sacrificios durante toda mi vida.

A mi papá, para que aprenda a creer...

De Nesty:

A mis padres por darme su apoyo en todo momento.

A mi hermano para que siga mi camino.

A mi abuela la cual estaría muy contenta con mi graduación.



Resumen

Actualmente la insuficiente disponibilidad de tiempo de los profesores y el poco conocimiento de los estudiantes relacionado con la práctica de la actividad física en la Universidad de las Ciencias Informáticas, hace necesario estudiar las herramientas más factibles para el desarrollo de aplicaciones que permitan la recomendación de ejercicios físicos a la comunidad universitaria. Se analizan algunos sistemas similares definiendo posibles funcionalidades para la solución propuesta, agregándole las necesidades del cliente. El desarrollo se divide en las fases propuestas por la metodología seleccionada, documentando cada paso y cada artefacto generado. Para verificar el cumplimiento de las funcionalidades propuestas se realizan las pruebas de aceptación. Como resultado se obtiene un producto funcional con una documentación que sirve de base para futuras investigaciones o modificaciones a la propuesta de solución.



Índice

Introducción.....	14
Capítulo1: Fundamentación teórica	18
1.1 <i>Introducción</i>	18
1.2 <i>Generalidades de los sistemas de recomendación</i>	18
1.2.1 El problema de la recomendación	19
1.2.2 El modelo del proceso de recomendación.....	19
1.3 <i>Clasificación de los sistemas de recomendación</i>	20
1.3.1 Sistemas basados en filtrado colaborativo	20
1.3.2 Sistemas basados en el conocimiento	21
1.3.3 Recomendadores basados en el contenido	21
1.3.4 Sistemas de recomendación híbridos	22
1.3.5 Fundamentación del sistema seleccionado	22
1.4 <i>Técnicas algorítmicas utilizadas en los sistemas recomendadores</i>	22
1.4.1 Técnicas algorítmicas basadas en memoria	23
1.4.2 Técnicas algorítmicas basadas en modelo	24
1.4.3 Fundamentación de la técnica algorítmica seleccionada	25
1.5 <i>Análisis de sistemas similares</i>	26
1.5.1 Aportes del estudio de los sistemas similares	26
1.6 <i>Metodologías de desarrollo de software</i>	27
1.6.1 Rational Unified Process (RUP)	27
1.6.2 SCRUM	28
1.6.3 Programación Extrema (XP).....	29
1.6.4 Fundamentación de la metodología seleccionada	30
1.7 <i>Sistema gestor de contenidos</i>	31
1.7.1 Joomla.....	32
1.7.2 Drupal	33
1.7.3 Fundamentación de la herramienta de desarrollo seleccionada.....	35



1.8 Lenguajes de programación del lado del cliente	35
1.8.1 Javascript	36
1.8.2 XHTML 5.....	36
1.8.3 CSS 3	37
1.9 Lenguajes de programación del lado servidor	37
1.9.1 PHP 5.3.8.....	37
1.10 Sistemas gestores de Bases de Datos	38
1.10.1 MySQL 5.1.37	38
1.10.2 PostgreSQL 8.3.5.....	39
1.10.3 Fundamentación del SGBD seleccionado	39
1.11 Entorno de desarrollo integrado	40
1.11.1 Eclipse	40
1.11.2 NetBeans 7.1.....	41
1.11.3 Fundamentación del IDE seleccionado	42
1.12 Servidor web	42
1.12.1 IIS 7.5	42
1.12.2 Apache 2.2.12	43
Conclusiones	43
Capítulo 2: Propuesta de solución	45
2.1 Introducción	45
2.2 Procesos vinculados al campo de acción	45
2.3 Personal relacionado con el sistema	46
2.4 Definición de servicios y contenidos	47
2.5 Historias de usuarios	47
2.6 Estimación de esfuerzos por HU	51
2.7 Plan de iteraciones	53
2.8 Plan de duración de las iteraciones	54
2.9 Plan de entregas.....	56
2.10 Prototipo de interfaz de usuario	57



2.11 Tarjetas CRC.....	59
Conclusiones	61
Capítulo 3: Implementación y prueba del sistema	62
3.1 Introducción	62
3.2 Estructura de las tablas agregadas.....	62
3.3 Temas en Drupal.....	63
3.3.1 Tema del Sistema Recomendador de ejercicios físicos.....	63
3.3.2 Instalación del tema SRTema	65
3.4 Módulos en Drupal.....	66
3.4.1 Módulos estudiados de Drupal 7.8.....	66
3.4.2 Paquetes de módulos desarrollados para el sistema.....	67
3.4.3 Instalación de módulos.....	71
3.5 Configuración del sistema.....	72
3.5.1 Menús.....	72
3.5.2 Bloques	72
3.5.3 Roles de usuario.....	73
3.5.4 Formato de entrada	74
3.6 Pruebas	74
3.6.1 Pruebas unitarias	74
3.6.2 Pruebas de aceptación.....	74
Conclusiones	78
Conclusiones generales.....	79
Recomendaciones.....	80
Referencias Bibliográficas	81
Bibliografía.....	85
Glosario de términos	86
Anexos	89

Índice de tablas

<i>Tabla 1: Usuarios relacionados con el sistema.....</i>	<i>46</i>
<i>Tabla 2: Ejemplo de una HU.....</i>	<i>48</i>
<i>Tabla 3: HU Gestionar perfil de usuario.....</i>	<i>49</i>
<i>Tabla 4: HU Gestionar información de ejercicios.....</i>	<i>49</i>
<i>Tabla 5: HU Gestionar información de ejercicios.....</i>	<i>50</i>
<i>Tabla 6: Estimación de esfuerzos por HU</i>	<i>51</i>
<i>Tabla 7: Plan de duración de las iteraciones</i>	<i>55</i>
<i>Tabla 8: Plan de entregas.....</i>	<i>56</i>
<i>Tabla 9: Tarjeta CRC Gestionar perfil de usuario</i>	<i>59</i>
<i>Tabla 10: Tarjeta CRC Gestionar información de ejercicios.....</i>	<i>60</i>
<i>Tabla 11: Tarjeta CRC Calcular el índice de masa corporal (IMC).....</i>	<i>60</i>
<i>Tabla 12: Caso de Prueba de Aceptación HU1_P1</i>	<i>75</i>
<i>Tabla 13: Caso de Prueba de Aceptación HU1_P2.....</i>	<i>76</i>
<i>Tabla 14: Caso de Prueba de Aceptación HU1_P3.....</i>	<i>77</i>
<i>Tabla 15: Caso de Prueba de Aceptación HU1_P4.....</i>	<i>77</i>



Índice de Figuras

<i>Figura 1: Cálculo del Coeficiente de correlación de Pearson.</i>	<i>23</i>
<i>Figura 2: Cálculo de la Similitud basada en Coseno propuesto por (47).....</i>	<i>24</i>
<i>Figura 3: Equipo de desarrollo, Prototipo de Interfaz de usuario, 2 de Abril de 2013</i>	<i>58</i>
<i>Figura 4: Estructura de las tablas añadidas a la base de datos de Drupal, 16 de Mayo de 2013.....</i>	<i>63</i>
<i>Figura 5: Equipo de desarrollo, Niveles de acceso por roles, 18 de mayo de 2013.....</i>	<i>73</i>



Introducción

La actividad física y el ejercicio inciden de forma positiva en la personas, siendo considerados como una de las bases fundamentales para un buen estado de salud. Su práctica regular y moderada produce una mejora en el estado de bienestar general del individuo.

El organismo humano está diseñado para moverse y realizar todo tipo de actividades con diversos propósitos, entre los que se encuentra la propia subsistencia. El ejercicio y la actividad física realizados de forma regular durante el período de crecimiento, contribuyen al desarrollo armónico y equilibrado del hombre, alejándolo de múltiples enfermedades desde las primeras etapas de su vida hasta la adultez.

Es necesario para mantener un hábito de vida físicamente activo conllevar estilos de vida saludables, lo que implica una correcta alimentación, mantenerse alejados del consumo de tóxicos y evitar el sedentarismo. De ahí surge la necesidad de la práctica continua de ejercicios que garanticen un correcto funcionamiento del organismo.

La Universidad de las Ciencias Informáticas (UCI), es una universidad de nuevo tipo que tiene como misión formar profesionales comprometidos con su patria y altamente calificados en la rama de la Informática; producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática. Esto implica la inserción de los estudiantes en proyectos productivos desde los primeros años de la carrera, facilitando su autopreparación como futuros ingenieros (45).

Paralelamente a su preparación profesional, muchos estudiantes practican diferentes deportes y realizan ejercicios físicos para mantener el cuerpo en la mejor forma posible.

Muchos de estos estudiantes no tienen una idea concreta de cuáles son los ejercicios que deben realizar según su condición física, peso corporal, edad, sexo, además de las enfermedades que padece, la práctica o no de ejercicios físicos y su sistematicidad. Aunque la universidad cuenta con profesores experimentados en el tema, estos no siempre están disponibles para guiar a los estudiantes en la realización de los ejercicios adecuados para mejorar su condición física, basándose en sus características propias. A pesar de que existe en Internet un cúmulo de sistemas que brindan información relacionada con el desarrollo de ejercicios físicos que permiten a los usuarios actualizarse en los temas relacionados con



las facilidades que brinda el desarrollo de actividades físico- deportivas para el mejoramiento de su calidad de vida, existe una insuficiente disponibilidad de herramientas informáticas de apoyo al mejoramiento de la condición física para estudiantes de la comunidad universitaria.

En muchas ocasiones, los estudiantes al interactuar con este tipo de aplicaciones disponibles en Internet u otros sitios relacionados con el deporte, encuentran información que no tiene en cuenta las características propias de cada usuario, lo que provoca que este no pueda identificar la solución a sus necesidades para realizar la ejercitación.

Considerando la anterior situación problemática, unida a la necesidad de contribuir al mejoramiento de la condición física en los estudiantes de la comunidad universitaria se identifica como **problema de investigación**: *¿Cómo orientar ejercicios físicos a partir del estado inicial de condición física de los estudiantes de la UCI?*

En el marco de este trabajo sus autores se trazaron como **objetivo general**: *desarrollar un sistema que oriente ejercicios físicos a los estudiantes de la UCI a partir del estado inicial de condición física.*

A partir del problema de investigación identificado se definió como **objeto de estudio**: *los sistemas recomendadores.*

El **campo de acción** está enfocado en *los sistemas de recomendación de ejercicios físicos.*

Como **idea a defender** se plantea: *la práctica de ejercicios propuestos por un sistema recomendador contribuirá al mejoramiento de la condición física en los estudiantes de la comunidad universitaria.*

Como **objetivos específicos** se definieron los siguientes:

- Realizar el estudio del estado del arte de los sistemas recomendadores.
- Diseñar los principales procesos a informatizar de un sistema que permita recomendar ejercicios físicos a partir del estado inicial de condición física de los estudiantes de la UCI.
- Implementar y validar las funcionalidades de un sistema que permita recomendar ejercicios físicos a partir del estado inicial de condición física de los estudiantes de la UCI.

Para dar cumplimiento a los objetivos específicos se plantean las siguientes **tareas**:



Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

- Análisis de los fundamentos teóricos de la investigación relacionados con los sistemas recomendadores que más se adecuen al dominio de un sistema de recomendación de ejercicios físicos.
- Estudio de las técnicas algorítmicas referenciadas más efectivas para el desarrollo de los sistemas de recomendación.
- Estudio y definición de la metodología de desarrollo de software, tecnologías y herramientas a utilizar adecuándose a las particularidades del problema a resolver y del propósito del producto final.
- Elaboración de los artefactos generados por la metodología utilizada.
- Desarrollo del sistema de recomendación de ejercicios físicos.
- Validación del sistema de recomendación según la metodología de desarrollo de software seleccionada.

Para la realización de esta investigación se utilizaron los siguientes **métodos teóricos**:

1. Analítico-sintético: Al analizar la documentación y elementos teóricos relacionados con las herramientas para recomendación de ejercicios físicos.
2. Inductivo-deductivo: Al determinar las características de los sistemas de recomendación que sean aplicables a un sistema recomendador de ejercicios físicos para la comunidad universitaria.
3. Histórico-lógico: Al analizar la evolución histórica de los sistemas de recomendación como una de las herramientas para concebir el sistema actual.
4. Modelación: Como forma de representación de todos los datos inherentes a un sistema recomendador de ejercicios físicos para la comunidad universitaria.

El uso de los **métodos empíricos**, por otro lado, conllevó a una serie de procedimientos prácticos con el objeto y los medios de investigación; permitiendo revelar las características fundamentales y relaciones esenciales del primero. Se empleó:



Entrevistas al cliente y especialistas en el tema para determinar las funcionalidades a implementar para desarrollar un sistema que oriente ejercicios físicos a los estudiantes de la UCI a partir del estado inicial de condición física.

El documento está organizado en tres capítulos, de la siguiente forma:

Capítulo 1. Fundamentación teórica: Se hace un análisis de los principales conceptos relacionados con los sistemas recomendadores. Además, se realiza un estudio del estado del arte de las herramientas recomendadoras y de los sistemas de recomendación, haciendo énfasis en la evolución histórica y teórica. Se realiza un estudio de las distintas metodologías y herramientas a utilizar en su desarrollo.

Capítulo 2. Propuesta de solución: En este capítulo se presenta el diseño de la propuesta de solución al problema planteado, que contará con la determinación de los contenidos y servicios que brindará el sistema, definiéndose las funcionalidades que debe cumplir, así como el diseño y la estructuración del mismo.

Capítulo 3. Implementación y prueba: Se describe el funcionamiento del CMS Drupal en su versión 7.8, haciendo referencia a los módulos de esta herramienta. Además, se implementan las funcionalidades que requiere el sistema y se describen las pruebas propuestas por la metodología seleccionada.

Al final del documento se incluyen las referencias bibliográficas, un glosario de términos y los anexos referenciados a lo largo del mismo.



Capítulo 1: Fundamentación teórica

1.1 Introducción

En este capítulo se realiza un análisis de los principales conceptos relacionados con sistemas de recomendación, sus generalidades y características principales. Además, se lleva a cabo un estudio del estado del arte de los sistemas recomendadores orientados al desarrollo de ejercicios para mejorar la condición física. Finalmente, se hace una breve descripción de las herramientas, tendencias y tecnologías a utilizar en el desarrollo de la propuesta.

1.2 Generalidades de los sistemas de recomendación.

En la actualidad existe una gran cantidad de sitios en Internet, ofreciendo múltiples servicios y productos, convirtiéndose en un caos de información sin que el usuario pueda encontrar a veces lo que realmente desea. Los sistemas de recomendación surgen para solucionar este problema.

A lo largo de los años la definición de los sistemas de recomendación ha ido evolucionando y siempre ha ido ligada con el avance de las nuevas técnicas o ideas que iban surgiendo en la literatura de los sistemas de recomendación. Un sistema de recomendación es aquel sistema que tiene como principal tarea seleccionar ciertos objetos de acuerdo a los requerimientos del usuario, además este sistema utiliza las opiniones de los usuarios de una comunidad para ayudar a usuarios de la misma a encontrar contenidos de su gusto entre un conjunto sobrecargado de posibles elecciones (12).

A principios de la década de los 90 empezaron a surgir dentro de los servicios de grupos de noticias (*newsgroups*), servicios de filtrado de noticias que permitían a su comunidad de usuarios acceder exclusivamente a aquellas noticias que potencialmente podían ser de su interés (31, 32). No obstante, el primer sistema de recomendación que apareció fue el llamado “*Tapestry*” (33), desarrollado por *XeroXPARC*, el cual recomendaba listas de noticias, boletines electrónicos de artículos a usuarios de correo electrónico, permitía que cualquier usuario almacenara anotaciones sobre los mensajes y los usuarios del sistema podían hacer consultas que combinaban información textual con consultas semánticas de metadatos.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Los sistemas de recomendación se distinguen por dos criterios fundamentales: por un lado se encuentran aquellos que producen recomendaciones individualizadas en forma de "output" y, por otro, aquellos que tienen el efecto de guiar al usuario de forma personalizada para los intereses de éste dentro de un dominio con grandes cantidades de ítems posibles a elegir. Teniendo esto en cuenta, se puede afirmar que los sistemas de recomendación son un tipo específico de filtro de información cuyo objetivo es mostrar ítems (libros, artículos, películas, imágenes, sitios web, entre otros) al usuario que le sean relevantes o de interés. Se entiende por filtro de información un sistema que elimina información inadecuada o no deseada de un flujo de información de forma automática o semiautomática para ser presentada a los usuarios (7).

La amplia utilización y aceptación de este nuevo paradigma ha devenido en que se pueda afirmar que se está abandonando la época de la información y se está iniciando la época de la recomendación.

Tomando en consideración los criterios abordados anteriormente los autores de esta tesis consideran que los sistemas de recomendación reciben información del usuario acerca de productos o servicios en los que el usuario se encuentra interesado y le recomienda aquellos que estén más cercanos a sus preferencias.

1.2.1 El problema de la recomendación

El principio de funcionamiento de los sistemas de recomendación está resumido en el problema de la recomendación (34), el cual puede ser formulado de la siguiente forma:

Sea C un conjunto de usuarios, y S el conjunto de todos los ítems posibles a ser recomendados, tales como libros, películas o restaurantes, pudiendo ser bien grande la cardinalidad de ambos conjuntos. Sea además 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:

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

1.2.2 El modelo del proceso de recomendación

Dentro de la concepción de sistemas de recomendación, ocupan un papel central el buscador de recomendaciones, el proveedor de preferencias y el sistema recomendador propiamente dicho. El buscador de recomendaciones (generalmente un individuo) puede solicitar sugerencias al sistema recomendador, o este puede automáticamente ofrecerlas sin una solicitud previa. Asimismo, para la



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

construcción de las recomendaciones, el sistema puede capturar de manera directa las preferencias específicas de los individuos, o puede inferirlas a través de preguntas indirectas. Una vez recopilados todos los datos sobre los usuarios, el sistema está listo para recomendar los ítems que el usuario probablemente prefiera, basándose en su perfil, el perfil de otros usuarios, los datos del proveedor de las posibles preferencias, entre otros criterios a tomar en cuenta, siendo utilizadas estas recomendaciones para seleccionar ítems del universo de alternativas. En la práctica, a la hora de concebir un sistema recomendador, muchas veces no se instancian algunas partes de este modelo, dependiendo esto de los requerimientos y de los objetivos a cumplir (34).

1.3 Clasificación de los sistemas de recomendación

Los sistemas de recomendación se pueden clasificar en diferentes tipos dependiendo del tipo de información que utilizan para realizar sus recomendaciones. Tradicionalmente existen dos paradigmas para la selección de elementos o filtrado de acuerdo con la manera de realizar las recomendaciones (32).

1.3.1 Sistemas basados en filtrado colaborativo

Los sistemas de recomendación basados en un filtrado colaborativo son aquellos en los que las recomendaciones se realizan basándose en términos de similitud entre los usuarios (9). Es decir, recomiendan objetos que son del gusto de otros usuarios de intereses similares.

Para la realización de un buen sistema de recomendación colaborativo (es decir, un sistema que ofrezca recomendaciones de calidad) es necesario utilizar un adecuado algoritmo de filtrado colaborativo. Conforme la utilización de estos sistemas se ha ido popularizando se han detectado una serie de problemas en un modelo de funcionamiento como son: la escasez, que consiste en el insuficiente número de elementos para permitir que el sistema pueda recopilar las preferencias de los usuarios con precisión y por lo tanto proporcionar recomendaciones fiables a partir de otras ya existentes; la escalabilidad, que sucede cuando el número de usuarios y elementos es muy grande y por tanto aumenta la complejidad de los algoritmos que se utilizan para la recomendación, lo que provoca que el sistema se tome más tiempo del normal para realizar la recomendación; y el problema del ítem nuevo, el cual por tener esa condición es probable que tenga pocas puntuaciones y por ello será difícil que sea recomendado (13). Multitud de estudios y experimentos se han llevado a cabo en los últimos tiempos con la intención de minimizar estos problemas.



1.3.2 Sistemas basados en el conocimiento

Los sistemas de recomendación basados en conocimiento realizan un razonamiento basado en casos entre las necesidades y preferencias de cada usuario para sugerir recomendaciones (14).

A diferencia de otros sistemas de recomendación lo único que necesitan estos sistemas es tener un conocimiento general sobre el conjunto de objetos y un conocimiento informal de las necesidades del usuario.

El principal problema de estos sistemas de recomendación es que aunque no requieren mucha información sí requieren un gran esfuerzo humano para realizar las recomendaciones mediante todo tipo de heurísticas de inferencia.

1.3.3 Recomendadores basados en el contenido

Un sistema basado en el contenido es aquel en el cual las recomendaciones son realizadas basándose solamente en un perfil creado a partir del análisis del contenido de los objetos que el usuario ha evaluado en un pasado (9).

Los sistemas de recomendación basados en el contenido generan recomendaciones teniendo en cuenta la valoración que el usuario ha hecho de los ítems en el pasado. De esta forma, el sistema tiene en cuenta las valoraciones que el usuario realiza de los ítems y las utiliza para recomendar nuevos ítems que tienen características similares a los evaluados positivamente por el usuario. En otras palabras: extraen características de los objetos y las comparan con el perfil del usuario para predecir las preferencias de los usuarios sobre tales objetos. Lo que se pretende es recomendar objetos similares en su contenido a objetos que ya sabemos que son del agrado del usuario en cuestión, o sea, los que forman parte de su perfil.

El perfil del usuario es utilizado en los sistemas de recomendación basados en el contenido para almacenar información referente al usuario. Esta información puede ser proporcionada directamente por el usuario de forma explícita (por ejemplo, mediante un formulario) o deducida por el sistema (por ejemplo, mediante el *feedback* de los ítems que el usuario proporciona al sistema).

La obtención de datos es necesaria en el sistema de recomendaciones basada en el contenido pero no sirve de nada sin un mecanismo o algoritmo que los procese y extraiga conclusiones o, en este caso, recomendaciones. Estos algoritmos crean una función que modela los intereses de los usuarios. Por



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

consiguiente, dado un modelo de usuario y un ítem nuevo, la función predice si el mismo será del agrado del usuario.

1.3.4 Sistemas de recomendación híbridos

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

Con vista a mitigar estas desventajas, una opción muy utilizada es la de combinar la recomendación basada en contenido con el filtrado colaborativo, conformando un sistema de recomendación con características híbridas. A través de estos, la construcción de preferencias se basa tanto en los intereses del usuario, como en los intereses de los usuarios similares, coincidiendo así con dos de las técnicas específicas antes expuestas.

Varios autores han demostrado cómo a través de este enfoque la calidad de las recomendaciones aumenta considerablemente (14), constituyendo de esta forma una alternativa a tener en cuenta a la hora de diseñar un sistema de este tipo.

1.3.5 Fundamentación del sistema seleccionado

Teniendo en cuenta que el sistema a desarrollar debe permitir al usuario crear un perfil con sus características y especificar sus preferencias acerca de los ejercicios físicos que desea realizar, además de valorar su similitud con otros usuarios del sistema, para determinar la recomendación adecuada de ejercicios, se considera que el sistema de recomendación híbrido es el que más se adecua a las especificaciones del mismo.

1.4 Técnicas algorítmicas utilizadas en los sistemas recomendadores

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



1.4.1 Técnicas algorítmicas basadas en memoria

Las técnicas algorítmicas basadas en memoria utilizan la Base de Datos completa de usuarios-ítems para generar las predicciones, el Sistema Recomendador usa una técnica estadística para hallar un conjunto de usuarios que tengan una historia de concordancia con el usuario objetivo (vecinos), una vez es formada una vecindad se utiliza un algoritmo que combine preferencias de los vecinos para producir una predicción y un ranking de los “N principales” para el usuario objeto (17).

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

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

Este fue uno de los primeros algoritmos de filtrado colaborativo en implementarse. Se basa fundamentalmente en 3 pasos: la representación de los datos, formación de vecinos y generación de recomendaciones (34).

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

Coeficiente de correlación de Pearson

El Coeficiente de correlación de Pearson es una métrica típica de similitud entre funciones de preferencias de usuarios o (menos frecuentemente) distancias de vectores o productos puntos. La fórmula da una aproximación de qué tan bien los vectores comparados (perfiles, ítems) coinciden en la escala desde cero (no similares) a uno (total coincidencia) ó -1 (total diferencia) (17).

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

Figura 1: Cálculo del Coeficiente de correlación de Pearson.



Similitud basada en coseno

Se considera cada elemento como un vector dentro de un espacio vectorial de m dimensiones y se calcula la similitud como el coseno del ángulo que forman. Es decir si se tienen dos vectores $x1$, $x2$ consistentes en un *array* cuyos elementos son las votaciones recibidas de cada usuario. Su similitud será:

$$sim(x1, x2) = \frac{\sum_{u \in U} (R_{u,x1} - \bar{R}_{x1}) (R_{u,x2} - \bar{R}_{x2})}{\sqrt{\sum_{u \in U} (R_{u,x1} - \bar{R}_{x1})^2} \sqrt{\sum_{u \in U} (R_{u,x2} - \bar{R}_{x2})^2}}$$

Figura 2: Cálculo de la Similitud basada en Coseno propuesto por (35).

1.4.2 Técnicas algorítmicas basadas en modelo

Estos algoritmos primero modelan los votos de los usuarios. Tratan el problema como un problema de predicción estadística y calculan el valor esperado para cada ítem en función de los votos anteriores. Para ello se utilizan distintos algoritmos de aprendizaje: *Clustering*, Redes Neuronales como las Redes de Funciones de Base Radial (RBFN), *Hortig*, Encasillamiento, Redes Bayesianas. En general, ante las consultas responden más rápido que los basados en memoria, pero necesitan de un proceso de aprendizaje intensivo (35).

Redes Bayesianas

Una Red Bayesiana es un grafo acíclico en el que los nodos representan variables proposicionales y los arcos identifican dependencias causales entre ellos. Así, un arco entre dos nodos de la red significa que el nodo origen (también llamado nodo padre) tiene un impacto causal sobre el nodo destino (nodo hijo), o lo que es lo mismo, que este último depende del primero. Para poder cuantificar el efecto que los nodos padres tienen sobre un nodo específico de la red, se asocia a cada uno de éstos una tabla de probabilidades condicionales (calculadas mediante el teorema de Bayes), de forma que el valor de cada nodo es función de los valores de sus nodos padres. Además, los nodos hojas representan proposiciones cuyos valores son determinados por observación, a partir de los datos almacenados en un conjunto de entrenamiento del que se extrae la información representada en la red (18).

Las Redes Bayesianas son aplicables al campo de los sistemas de recomendación, son herramientas adecuadas tanto para modelar las preferencias de los usuarios, como para utilizarlas posteriormente durante el proceso de recomendación. El perfil de usuario resultante sería un grafo en el que los nodos y



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

arcos identificarían los conceptos/atributos que interesan a los usuarios y las relaciones causales establecidas entre ellos, respectivamente. Para elaborar recomendaciones personalizadas, el clasificador automático decide si un producto objetivo es interesante para el usuario activo, basándose en las relaciones probabilísticas existentes entre los atributos de dicho producto y las preferencias representadas en su red Bayesiana (19).

Redes Neuronales

Las Redes Neuronales (RN) proporcionan una forma muy conveniente de representación del conocimiento en tareas de Recuperación de Información (RI), donde los nodos representan objetos del proceso de RI como palabras claves, autores y citas y los enlaces representan la asociación ponderada de estos (relevancia). Las propiedades de aprendizaje de las RN tipo retropropagación del error y las propiedades de búsqueda en paralelo de las Redes tipo *Hopfield* proporcionan mecanismos efectivos para identificar información relevante de ítems en bases de datos (17).

Horting

Es una técnica basada en grafos en la cual los nodos son los usuarios y las aristas entre nodos son indicadores de los grados de similitud entre dos usuarios. Las predicciones se producen al recorrer el grafo entre nodos cercanos y combinando las opiniones entre usuario cercanos.

Esta técnica difiere de los algoritmos de vecindad más cercana, en la forma como el grafo puede ser recorrido por otros usuarios que no han valorado los ítems, luego esta técnica explora las relaciones transitivas que los algoritmos de vecindad más cercana no tienen en cuenta (17).

1.4.3 Fundamentación de la técnica algorítmica seleccionada

El sistema a desarrollar está apoyado en la técnica algorítmica basada en memoria, específicamente tomando como base el algoritmo de *k-nearest neighbor*, el cual no fue utilizado porque sus características no cumplen con los requerimientos del sistema. Se realizó una modificación del mismo a partir de la idea general que plantea este algoritmo, lo cual permitió desarrollar un perfil algorítmico que se basa en la filosofía de recomendar al usuario un paquete de ejercicios basado en sus características y según sus necesidades y otro paquete apoyándose en las características de usuarios semejantes a él.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

1.5 Análisis de sistemas similares

Como se explica anteriormente los sistemas recomendadores son los encargados de sugerir a un determinado usuario cierta información que se considere de interés para él. Su surgimiento y empleo en la educación está propiciado por la gran cantidad de *software* o plataformas educativas que han surgido como una medida de apoyo al proceso de enseñanza y aprendizaje.

En la presente investigación se realiza un estudio de varios Sistemas Recomendadores disponibles en Internet con características similares a las del sistema que se pretende desarrollar:

- Sistema Recomendador y Adaptativo para Bibliotecas Virtuales, el cual está basado en las preferencias estáticas de los usuarios y en los patrones de comportamiento de los usuarios al interactuar con el catálogo de la biblioteca, además de tener en cuenta la evaluación estadística de los datos de utilización.
- Sistema de recomendación y planificación turística de la ciudad de Valencia vía web, el cual utiliza las técnicas de recomendación básicas más apropiadas al dominio y las une utilizando una técnica de recomendación híbrida mezclada. Detalla también un núcleo planificador para la organización de las recomendaciones, y la elaboración de una agenda turística teniendo en cuenta las restricciones del entorno. Además se incluyen pequeños módulos que estudian y organizan la información de los usuarios y un módulo independiente que interactúa con la base de datos.
- Sistema Recomendador híbrido de cuestionarios interactivos, el cual fue desarrollado en la Universidad de las Ciencias Informáticas y ayuda a los usuarios a ejercitar el contenido a través de baterías de preguntas que sean más apropiadas de acuerdo a sus necesidades y preferencias educativas. El sistema cuenta con un repositorio de cuestionarios interactivos basado en el estándar IMS QTI donde cada objeto tiene sus metadatos descriptivos. Se utilizan estos metadatos con el objetivo de recuperar aquellos cuestionarios que satisfagan las necesidades del usuario.

1.5.1 Aportes del estudio de los sistemas similares

Después del análisis anterior sobre las tendencias actuales de desarrollo de los Sistemas Recomendadores se encontraron características comunes entre estos basando esta afirmación en que todos los sistemas analizados son sistemas híbridos, sin embargo, no se observan funcionalidades que contribuyan directamente a la solución del problema planteado, además de que no se encontró un Sistema Recomendador de ejercicios físicos en la investigación realizada.



1.6 Metodologías de desarrollo de software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental para guiar el desarrollo de un producto software, pero los requisitos de un software a otro son tan variados y cambiantes que ha dado lugar a que exista una gran variedad de metodologías (28).

Existen dos grandes grupos: 1) las metodologías robustas y 2) las metodologías ágiles. Las primeras son las que establecen rigurosamente las actividades a desarrollar, son las tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida (28), por estas razones es poco recomendable utilizar este tipo de metodologías para el desarrollo de proyectos pequeños pues demoraría innecesariamente su implementación y generaría una documentación que no sería utilizada.

Las metodologías ágiles están orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando (29). Estas metodologías están definidas para el desarrollo de proyectos con requisitos poco definidos o cambiantes, son aplicables para equipos de trabajo pequeños resolviendo problemas concretos y divide el trabajo minimizando los fallos y los costes. Tienen como objetivo principal crear un producto que funcione y cumpla las exigencias del cliente antes de escribir mucha documentación, contando con la colaboración del cliente en todo momento.

1.6.1 Rational Unified Process (RUP)

Según Pressman RUP es *“la metodología de desarrollo más popular y una de las más utilizadas a nivel mundial en grandes proyectos por su robustez. Muchas metodologías de surgimiento posterior son derivadas o adaptaciones de esta metodología tradicional. RUP es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular”* (27).



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

RUP es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. Está basado en componentes. El software está formado por componentes software interconectados a través de interfaces (27).

Está caracterizado por los siguientes aspectos:

Dirigido por casos de uso: Los cuales guían el proceso de desarrollo del software ya que modelan los requerimientos funcionales del sistema proporcionando un resultado de valor al usuario.

Centrado en la arquitectura: La arquitectura del sistema es donde se toma el conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, las interfaces entre ellos, su comportamiento, sus colaboraciones, y su composición.

Iterativo e incremental: En cada ciclo de iteración se produce una nueva versión del software.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del mismo. Cada uno consta de cuatro fases: inicio, elaboración, construcción, y transición. Utiliza Lenguaje Unificado de Modelado (UML) como lenguaje de modelado y cuenta con varias fases de trabajo en las cuales se desarrolla una serie de flujos fundamentales del desarrollo del proyecto.

Entre los beneficios que se pueden obtener de RUP es que unifica todo el equipo de desarrollo de software, lo que favorece la comunicación, asegurando la asignación de recursos en forma eficiente, la entrega de los artefactos correctos y el cumplimiento de los tiempos límite.

Sin embargo también presenta deficiencias, entre las cuales se encuentran: constituye una metodología pesada en la cual se genera documentación en exceso. Se aconseja solo a proyectos de larga duración y gran cantidad de recursos, de lo contrario retrasaría el proceso de desarrollo del software. Además es poco flexible y menos adaptable a los cambios de requisitos durante el proceso de desarrollo.

1.6.2 SCRUM

SCRUM es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80. Esta metodología no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Su modo de desarrollo es adaptable antes de predictivo, orientado a las personas más que a los procesos y emplea un modelo de desarrollo incremental basado en iteraciones y revisiones, cada iteración es denominada sprint. El proceso comienza con una visión general del producto, dando detalles de las funcionalidades o las partes del negocio con mayor prioridad, que pueden llevarse a cabo en un período de tiempo breve. Cada período de desarrollo es una iteración que finaliza con un incremento operativo del producto (46). El ciclo de vida está dividido en tres fases: planificación del sprint, seguimiento del sprint y revisión del sprint.

La metodología SCRUM propone para la gestión de sus fases varias prácticas y herramientas, sin embargo ninguna está orientada en concreto para los desarrolladores. Los grupos de usuarios se dividen en cinco roles diferentes entre los que se encuentran, el propietario del producto, el cual es el responsable entre otras cosas de la financiación del proyecto, SCRUM máster, el cual garantiza el funcionamiento de los procesos y la metodología, el gestor que se encarga de tomar las decisiones finales de los objetivos y requisitos, entre otros. Estos roles no se corresponden con la estructura del equipo de desarrollo de la presente investigación, por lo que no se considera adecuada esta metodología para el desarrollo del sistema propuesto (46).

1.6.3 Programación Extrema (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (29).

La programación extrema es una metodología de desarrollo ligera basada en cuatro valores fundamentales que debe cumplir el equipo de desarrollo y una docena de prácticas, entre las que se encuentran las entregas pequeñas, la refactorización constante, el diseño simple y un desarrollo orientado a las pruebas, entre otras, que propician un aumento en la productividad a la hora de generar software (28).

Los cuatro valores son:



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

- **Comunicación:** XP ayuda mediante sus prácticas a fomentar la comunicación, ya que el cliente forma parte del equipo de desarrollo.
- **Coraje:** Un miembro de un equipo de desarrollo extremo debe tener el coraje de exponer sus dudas, esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros.
- **Simplicidad:** Dado que no se puede predecir cómo va a ser en el futuro el software que se está desarrollando; un equipo de programación extrema intenta mantener el software lo más sencillo posible.
- **Retroalimentación:** La retroalimentación actúa junto con la sencillez y la comunicación, cuanto mayor es la retroalimentación más fácil es la comunicación. Cuanto más simple un sistema, más fácil de probar.

Esta metodología divide su desarrollo en cuatro fases principales: *Planificación*, donde se definen las Historias de Usuario con el cliente a partir de las características que deben ser adicionadas al sistema que fueron especificadas por este último; *Diseño*, donde se crean diseños simples y claros, siendo constantemente revisados y probablemente modificados durante el desarrollo, implementando con el orden definido en las iteraciones planificadas; *Desarrollo*, para ir cumpliendo con el objetivo de realizar entregas frecuentes al cliente, cuya presencia es también indispensable en esta fase; *Pruebas*, las cuales permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento, por tal motivo son realizadas en todo momento asegurando el cumplimiento de lo planteado en el diseño (28).

1.6.4 Fundamentación de la metodología seleccionada

Luego del análisis de algunas metodologías para el desarrollo del software se considera que la metodología XP es la más apropiada para el desarrollo del software pues el equipo de trabajo está conformado solo por dos personas, se necesita hacer en un tiempo mínimo y el objetivo es satisfacer al cliente con el producto sin generar artefactos innecesarios ya que esta metodología está centrada en potenciar las relaciones interpersonales como clave para el éxito, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se tiene en cuenta además que el cliente no tiene ninguna experiencia en cuanto a desarrollo web, lo que



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

significaría cambios futuros en las funcionalidades y en este aspecto la metodología es flexible ya que permite administrar estos de forma óptima.

1.7 Sistema gestor de contenidos

Un Sistema de Gestión de Contenidos, en inglés **Content Management System (CMS)**, es una aplicación Web que posibilita desde un Navegador de Internet, y sin necesidad de tener grandes conocimientos informáticos, desarrollar sitios Web. Además permite gestionar y presentar los contenidos del mismo. En la actualidad uso de los CMS se encuentra muy difundido en la creación de Portales Web pues brindan funcionalidades y facilidades a las personas, expertas o no en el tema, en el momento de crear o mantener un sitio Web.

Los Sistemas de Gestión de Contenido brindan distintas funcionalidades atendiendo a su propósito de creación y su tipo, pero las comunes o las que sustentan su concepto, CMS, se muestran a continuación (30):

Inclusión de nuevas funcionalidades en el Web. Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la Web. El sistema puede crecer y adaptarse a las necesidades futuras.

Mantenimiento de gran cantidad de páginas. En una Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.

Reutilización de objetos o componentes. Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado. Páginas interactivas. Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la Web.

Cambios del aspecto de la Web. Si no hay una correcta separación entre contenido y presentación, un cambio de diseño puede implicar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (*Cascading Style Sheets* u hojas de estilo en cascada).

Consistencia de la Web. La consistencia en un sistema Web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual). Un usuario nota enseguida cuándo una página no es igual que el resto dentro de la misma Web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el sistema Web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.

Control de acceso. Controlar el acceso a un sitio Web no consiste simplemente en permitir la entrada al sitio, sino que permite gestionar los diferentes permisos a cada área del mismo, aplicados a grupos o individuos.

Existen diferentes CMS, todos creados con propósitos específicos, entre ellos los genéricos, foros, blogs, wikis, *eCommerce*, portal, galería, *e-Learning*, entre otros (30).

1.7.1 Joomla

Joomla es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL, que permite crear sitios Web de alta interactividad, profesionalidad y eficiencia. Dentro de sus características generales se encuentran (1):

- Software libre (Licencias GNU/GPL), ampliable al disponer el código fuente.
- Completa y fácil administración por la web.
- Creación y administración rápida de una comunidad *on-line*.
- Creación de la web por inserción de módulos y componentes independientes.
- Creación y actualización dinámica de secciones, sub secciones y contenidos (públicos y privados).



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

- Creación de perfiles y privilegios con niveles jerárquicos para diferentes niveles de usuarios (súper administrador, administrador, gerente, editor y usuario registrado).
- Plantillas para transformar el diseño visual de la Web de forma automática en pocos minutos.
- Estadísticas de acceso a los contenidos.

El CMS Joomla proporciona gran variedad de extensiones destinadas al desarrollo de la Web 2.0, lo cual facilitaría el desarrollo de la propuesta de solución, aunque presenta algunas desventajas:

1.7.2 Drupal

Drupal es un sistema de gestión de contenido, modular, de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Reconocido por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web, y un énfasis especial en la usabilidad y consistencia de todo el sistema (22).

Características generales:

Ayuda on-line. Un robusto sistema de ayuda online y páginas de ayuda para los módulos del “núcleo”, tanto para usuarios como para administradores.

Búsqueda. Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

Código abierto. El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL.

Módulos. Las funcionalidades que brinda el CMS se pueden ampliar instalándoles nuevos módulos que implementen dichas funcionalidades. La comunidad de Drupal ha contribuido con muchos módulos que proporcionan disímiles funcionalidades.

URLs amigables. Drupal usa el *mod_rewrite* de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.

Gestión de usuarios

Autenticación de usuarios. Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, *Blogger*, *LiveJournal* u otro sitio Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.

Permisos basados en roles. Los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un rol y agrupar los usuarios por roles.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Gestión de contenido

Control de versiones. El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.

Enlaces permanentes. Todo el contenido creado en Drupal 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 (Nodos). El contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios, o no, sobre cada objeto.

Plantillas (Templates). El sistema de temas de Drupal 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.

Sindicación del contenido. Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios Web. Esto permite que cualquiera con un “Agregador de Noticias” visualice el contenido publicado en la Web desde el escritorio.

Plataforma

Independencia de la base de datos. Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una “capa de abstracción de base de datos” que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma. Drupal ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor Web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Múltiples idiomas y Localización. Drupal está pensado para una audiencia internacional y proporciona opciones para crear un Portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz Web, importando traducciones existentes o integrando otras herramientas de traducción como *GNU gettext*.

Administración y Análisis



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Administración vía Web. La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.

Análisis, Seguimiento y Estadísticas. Drupal puede mostrar en las páginas Web de administración informes sobre *referrals* (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan por el sitio.

Registros e Informes. Toda la actividad y los sucesos del sistema son capturados en un “registro de eventos”, que puede ser visualizado por un administrador (43).

1.7.3 Fundamentación de la herramienta de desarrollo seleccionada.

La herramienta seleccionada para desarrollar el Sistema Web fue el CMS Drupal en su versión 7.8, basándose esta selección en las ventajas que este presenta con respecto a Joomla, entre las que se encuentran: diseño modular, soporte de varias bases de datos, como MySQL y PostgreSQL; permite la autenticación por LDAP. Es una buena alternativa para el sistema a desarrollar pues el mismo presenta una complejidad alta, desde el punto de vista de los usuarios y permisos (por ejemplo para abordar la construcción de una comunidad de usuarios) y desde el punto de vista de tipos de contenidos personalizados. En este tipo de proyectos permitirá que se ahorre tiempo y recursos.

1.8 Lenguajes de programación del lado del cliente

Un lenguaje de programación es utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente (12). En la actualidad existen diversos lenguajes de programación para diferentes fines, los cuales han surgido según las necesidades de las diferentes plataformas y las tendencias al avance en la programación web en los últimos tiempos.

Drupal utiliza el lenguaje XHTML como base para toda la estructura del sistema del lado del cliente dando formato a la información, CSS para definir la apariencia visual de los elementos en el diseño web de manera que se presente de forma estructurada y agradable y JavaScript para facilitar la interacción del documento y el navegador.



1.8.1 Javascript

JavaScript es un lenguaje de programación interpretado usado para crear páginas dinámicas, las variables no necesitan ser introducidas antes de su uso y los tipos de variables se resuelven dinámicamente durante su ejecución. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Fue creado por Brendan Eich en la empresa *Netscape Communications*. El código JavaScript que se encuentra dentro de las páginas web puede ser interpretado por todos los navegadores. Permite que las definiciones de funciones y otro tipo de código sean modificados mientras el programa se esté ejecutando. El modelo de ejecución de JavaScript se basa en la interpretación del código fuente. Es un lenguaje de alto nivel, multiplataforma y no necesita compilación. Está basado en objetos, admite la programación estructurada y maneja la mayoría de los eventos que se pueden producir sobre la página web. La mayoría de los navegadores en sus últimas versiones interpretan el código JavaScript integrado dentro de las páginas web (48).

En Drupal 7 se han agregado funcionalidades para facilitar el desarrollo con JavaScript (se actualiza el núcleo de JavaScript a jQuery versión 1.4.2, se actualiza la librería de jQuery a la versión 2.36 y se añade jQuery UI 1.8), alcanzando mejoras en la interfaz administrativa y una mayor interacción entre este lenguaje y el núcleo de Drupal, lo que permite la creación de páginas más dinámicas con funcionalidades nuevas y modernas que se esperan en los sitios web de la actualidad.

1.8.2 XHTML 5

El XHTML (*eXtensible Hypertext Markup Language*) o Lenguaje de Etiquetado Hipertextual Extensible es una reformulación del HTML (es lo que se utiliza para crear todas las páginas web de Internet) como aplicación XML. Su finalidad es que pueda ser usado como lenguaje de contenidos que sea a su vez acorde a XML y funciona en agentes de usuario (las aplicaciones que leen y procesan documentos). XHTML incorpora una nueva concepción o filosofía de modelación de las páginas web, que busca la creación de una web semántica. XHTML suprime todas las etiquetas y atributos que sirven para definir el aspecto y sólo se dejan las etiquetas que sirven para definir el significado de cada elemento de la página, dejando para Javascript su aspecto y diseño en distintos medios tales como ordenadores, teléfonos móviles, impresoras, entre otros (40).



1.8.3 CSS 3

Hojas de Estilo Cascada (*Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. Permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la regla CSS afectará a todas las páginas vinculadas a esa regla en las que aparezca ese elemento (49).

1.9 Lenguajes de programación del lado servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. Los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son ASP, JSP, PERL y PHP (30). En la presente investigación se seleccionó el CMS Drupal en su versión 7.8 el cual utiliza para la implementación de sus módulos el lenguaje PHP.

1.9.1 PHP 5.3.8

Su nombre proviene de las siglas del término en inglés *Hypertext Preprocessor* (inicialmente *PHP Tools* o *Personal Home Page Tools*). Fue creado en 1994 por Rasmus Lerdorf. No cuenta con licencia, es un software libre, gratuito y multiplataforma. Generalmente no se ejecuta en la computadora sino en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida.

Todo en su diseño está orientado a facilitar la creación de funciones claves para las páginas web, de forma sencilla. Permite además la conexión a diferentes tipos de servidores de bases de datos: MySQL, PostgreSQL, Oracle y Microsoft SQL Server, entre otros. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows. Es además, una alternativa de software libre a las tecnologías que requieren licencias para poder funcionar (47).



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Drupal utiliza PHP como lenguaje para construir y hacer las conexiones a la base de datos, mostrando una línea de documentación excelente.

1.10 Sistemas gestores de Bases de Datos

Un sistema gestor de bases de datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y su manipulación (4).

Drupal requiere un servidor de base de datos SQL compatible con PHP. La instalación por defecto incluye controladores para MySQL, PostgreSQL y SQLite. MySQL y PostgreSQL son las bases de datos que son utilizadas con más frecuencia en Drupal. Si se utiliza MySQL, es necesaria la versión 5.0.15 o superior, en caso de utilizarse PostgreSQL, es necesaria la versión 8.3.0 o posterior (5).

1.10.1 MySQL 5.1.37

MySQL es un SGBD relacional, multihilo y multiusuario. Su idea originaria fundada en 1995 por la empresa Open Source MySQL AB. EL objetivo que persigue MySQL es cumplir el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad (6).

Las principales características de este gestor de bases de datos que se tuvieron en cuenta son:

- Implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de APIs en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseña, manteniendo un muy buen nivel de seguridad en los datos.
- Alta velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

- Bajo consumo, lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Gran facilidad de configuración e instalación.
- Presenta una probabilidad muy reducida de corromper los datos.

MySQL es muy popular en aplicaciones web y actúa como un componente de bases de datos para las plataformas LAMP, MAMP y WAMP (Linux/MAC/Windows-Apache-MySQL-PHP/Perl/Python), también trabaja en numerosas plataformas como AIX, HP-UX, GNU/Linux, Mac OS X, Novell NetWare, OpenBSD, OS/2, Solaris, SunOS, y todas las versiones de Windows. Su mayor desempeño se logra cuando se combina con el lenguaje de programación PHP (6).

1.10.2 PostgreSQL 8.3.5

Es un SGBD objeto-relacional, ya que aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos), por nombrar solo unos ejemplos. Es altamente extensible: soporta operadores y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (*joins*) SQL92. Cuenta con una API (del inglés *Application Program Interface*) flexible lo cual ha permitido dar soporte para el desarrollo con PostgreSQL en diversos lenguajes de programación. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos ofrece, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación a objetos o la programación funcional.

1.10.3 Fundamentación del SGBD seleccionado

Se seleccionó MySQL pues su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es uno de los gestores más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. MySQL es muy utilizado en aplicaciones web, pues en este tipo de aplicaciones hay baja



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

conurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de sistemas, por tanto se considera que este gestor es el más adecuado para almacenar la información del sistema que se desea desarrollar.

1.11 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (por sus siglas en inglés *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C (50).

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación. Este es el caso de Eclipse, NetBeans, MS Visual Studio.NET, Zend Studio, entre otros. Zend Studio es un potente IDE desarrollado para el lenguaje PHP pero como desventaja tiene que es un software privativo, por lo que queda descartado por los problemas que origina obtener la licencia, así como MS Visual Studio.NET. En la presente investigación se definió PHP, el cual no necesita de compilación por ser un lenguaje interpretado. A continuación se analizan el IDE Net Beans y el editor de texto Notepad++ para definir cuál utilizar en el desarrollo de la propuesta de solución.

1.11.1 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto, Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados. Dentro de sus características fundamentales se encuentran: editor de texto, resaltado de sintaxis, compilación en tiempo real. Como desventaja tiene que para poder desarrollar entornos de diseño visual requiere la instalación de *plugins* por ejemplo el Visual Edition.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

El Eclipse Project está subdividido a su vez en tres sub-proyectos que son la propia plataforma, JDT (*Java Development Tool*) y PDE (*Plugin Development Environment*). Mediante este IDE se pueden crear diversas aplicaciones como pueden ser, sitios web, programas Java, PHP, C++ y Enterprise Java Beans. Su interfaz de usuario está compuesta de un conjunto de vistas, editores y perspectivas. Los editores permiten crear, modificar y salvar objetos, las vistas proveen información acerca de los objetos con los que se están trabajando y las perspectivas proveen distintas formas de organización del proyecto.

1.11.2 NetBeans 7.1

El NetBeans es un IDE para programadores, pensado para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans, además es un producto libre y gratuito sin restricciones de uso (50).

Características:

- Autocompletación de código.
- Utilidades de búsqueda y reemplazo muy intuitivos, amigables y sobre todo realmente útiles para el programador.
- Coloración de código.
- Refactorización inteligente.
- Diseñadores de interfaces gráficas para Swing y JSF.
- Asistentes para escritura de código rápido de acceso a datos basados en JPA (Java Persistence API).
- Asistentes para escritura de servicios Web.
- Soporte para múltiples máquinas virtuales.
- Soporte para múltiples servidores contenedores J2EE.
- Soporte para librerías de terceros.



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

También es importante recordar la variedad de *plugins* (componentes de software conectables a NetBeans) disponibles para incrementar las funcionalidades del IDE.

1.11.3 Fundamentación del IDE seleccionado

Para realizar la aplicación, se ha decidido utilizar el IDE NetBeans 7.1, ya que es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Es un producto libre y gratuito sin restricciones de uso y el desarrollador obtiene todas las herramientas que necesita para crear aplicaciones profesionales de escritorio, empresariales y móviles. Este IDE corre sobre varias plataformas y es muy fácil de instalar. Además es un entorno con grandes prestaciones para codificar y probar el software que se desarrolla.

1.12 Servidor web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados (44).

Drupal necesita un servidor web para ejecutar código en PHP. Los servidores que utiliza para esto son Apache y el servidor IIS (*Internet Information Server*, por sus siglas en inglés).

1.12.1 IIS 7.5

IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS (41).

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente. Los servicios de IIS proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro (41).



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de *Active Server Pages* (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl (41).

1.12.2 Apache 2.2.12

Para el desarrollo de la propuesta de solución se seleccionó el servidor web Apache en su versión 2.2.12, por ser una herramienta libre y brindar más seguridad que IIS. Apache funciona en una multitud de SO, mientras que IIS solo puede ser ejecutado en Windows. Además IIS es un software propiedad de Windows, lo que dificulta la obtención de la licencia para su uso, mientras que Apache es un software libre, facilitando su uso a toda la comunidad.

Apache es el servidor web hecho por excelencia, su configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es un servidor web de código libre por lo que permite realizar cualquier cambio en el código fuente (incluso *forks* y productos propietarios) (42).

A continuación se muestran las principales razones por las que este software libre es ampliamente reconocido en muchos ámbitos empresariales (42):

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

Conclusiones

En este capítulo se realizó un estudio del estado del arte de los sistemas recomendadores, mostrando sus principales características. Se analizaron las diferentes clasificaciones de los mismos basándose principalmente en la forma de procesar la información. Se llegó a la conclusión de que tienen una amplia aplicación en el mundo informático, comprobando que son aplicables al software que se va a desarrollar. Fueron además analizadas las diferentes herramientas a utilizar concluyendo que la metodología de desarrollo seleccionada es Programación Extrema (XP) y que el perfil tecnológico estará compuesto por



Capítulo 1: Fundamentación teórica

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

PHP, CSS, Javascript y XHTML como lenguajes de programación; Drupal, como sistema gestor de contenidos; NetBeans como IDE de desarrollo y MySQL como sistema gestor de la base de datos.

Capítulo 2: Propuesta de solución

2.1 Introducción

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. En la presente investigación, la metáfora del sistema que se propone es: “un espacio donde los usuarios puedan obtener recomendaciones de ejercicios físicos de acuerdo a sus características propias”. La metodología XP sugiere utilizar conjuntamente con la metáfora, otras técnicas importantes de diseño que serán presentadas más adelante en este capítulo.

En el mismo se describe la propuesta de solución basada en las dos primeras fases en las que se divide la metodología XP: Planeación y Diseño. Se centra en abordar temas relacionados con las necesidades del software y conceptos propios de la metodología como son las Historias de Usuario y las Tarjetas CRC (Clase-Responsabilidad-Colaboración) identificando las responsabilidades. Además se definen las funcionalidades y se generan los artefactos propuestos por la metodología seleccionada, concretando la cantidad de entregas y tiempo de ejecución.

2.2 Procesos vinculados al campo de acción

La práctica de ejercicios físicos en la Universidad de las Ciencias Informáticas tiene como objetivo el mejoramiento de la condición física en la comunidad universitaria. El exceso de información a través de la red y la escasez de tiempo del personal capacitado en el tema para guiar a los practicantes en la realización de los ejercicios adecuados para mejorar su condición física, basándose en sus características propias provocan que la mayoría de estos no realicen los ejercicios correctos. Además la universidad no cuenta con un sistema capaz de permitir que los estudiantes puedan obtener la información necesaria y que facilite la recomendación de ejercicios físicos acorde a las necesidades y características propias de los interesados en practicar ejercicios.

2.3 Personal relacionado con el sistema

El sistema que se desarrollará mostrará un grupo de funcionalidades y servicios para cumplir con los objetivos planteados. Su implementación se realiza a través de módulos en los que se establece un grupo de roles para asignar los diferentes permisos para su acceso. Tanto los roles como los permisos determinan el nivel de acceso de cualquier usuario que interactúe con el sistema a las funciones del mismo. Un usuario es cualquier persona que se relacione con el sistema, así sea a través de la vinculación al desarrollo del mismo o que de una forma u otra va a interactuar con la aplicación, incluyendo aquellos que se encargan de mantener el sistema actualizado y funcionando. El sistema propuesto dispone de los siguientes roles:

Tabla 1: Usuarios relacionados con el sistema

Usuario relacionado con el sistema	Descripción
Autenticado	Todos los usuarios deben autenticarse para acceder a cualquier opción que así lo requiera. Puede acceder a todas las opciones, menos a las de administración.
Administrador	Son aquellos usuarios que tienen todos los permisos para administrar las funcionalidades del sistema. Gestionan los contenidos, los usuarios y todos los servicios que se brindan.
Anónimo	Es la persona que navega por el sistema sin

	haberse registrado aún, interactúa con este sin privilegios y tiene la posibilidad de visualizar las diferentes opciones que brinda el mismo.
Especialista	Es el encargado de insertar los ejercicios que se recomendarán a los usuarios autenticados.

2.4 Definición de servicios y contenidos

Durante el transcurso de reuniones y encuentros entre el cliente y el equipo de desarrollo se definen varias funcionalidades y características que debe ofrecer el sistema para satisfacer las necesidades de los usuarios y cumplir con los objetivos trazados para el sistema. A continuación se mencionan varios de los servicios y contenidos que ofrece el mismo:

- Información relacionada con los diferentes tipos de ejercicios que se pueden realizar.
- Información sobre la musculatura.
- Noticias informativas sobre eventos deportivos a nivel nacional e internacional.
- Galería de imágenes.
- Videos.
- Encuestas.
- Foros.
- Enlaces a otros sitios de interés.
- Sistema recomendador.
- Evaluador de índice de masa muscular.

2.5 Historias de usuarios

La metodología XP utiliza la técnica de las historias de usuarios (HU) para describir las funcionalidades que debe tener el sistema. Una HU es una representación de un requerimiento de software escrito en una o dos frases utilizando el lenguaje común del usuario (acompañadas de las discusiones con los usuarios y las pruebas de validación). Dentro de la metodología XP las HU deben ser escritas por los clientes. Estas son una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran



Capítulo 2: Propuesta de solución

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos y permiten responder rápidamente a los requerimientos cambiantes. Estas deben ser independientes unas de otras, negociables, valoradas por los clientes o usuarios, estimables, pequeñas y verificables. Generalmente se espera que la estimación de tiempo de cada historia de usuario se sitúe entre una 1 y 2 semanas. Estimaciones mayores a dos semanas son indicativo de que la historia es muy compleja y debe ser dividida en varias historias. A continuación se muestran las HU que representan a las funcionalidades críticas que fueron implementadas para la propuesta de solución. En los anexos se encuentran las HU de las restantes funcionalidades junto con las características y cualidades del sistema (ver Anexo 1).

Tabla 2: Ejemplo de una HU

Historia de usuario	
<p>No: Número consecutivo a partir del 1.</p> <p>Usuario: ¿Quién ejecuta la HU?</p>	<p>Nombre: Identifica la HU.</p>
<p>Prioridad de Negocio: Define la relevancia e impacto de la HU para el negocio de acuerdo con las necesidades del usuario.</p>	<p>Puntos estimados: Permite estimar la duración de la implementación, representando con 1, una semana de trabajo.</p>
<p>Nivel de complejidad: Define la dificultad técnica que supone desarrollar la HU desde el punto de vista del programador.</p>	<p>Iteración asignada: Precisa la iteración en que será desarrollada la HU.</p>
<p>Descripción: Explica en que consiste la HU, teniendo en cuenta las acciones realizadas por el usuario y</p>	



Capítulo 2: Propuesta de solución
*Sistema Recomendador de ejercicios físicos para estudiantes de la
 Universidad de las Ciencias Informáticas*

la respuesta brindada por el sistema.

Observaciones: Brinda información extra que se estime agregar para hacer más comprensible la HU. Por ejemplo conceptos, post-condiciones, relación con otros requisitos, entre otras.

Tabla 3: HU Gestionar perfil de usuario.

Historia de usuario	
No: 1. Usuario: Autenticado, Administrador, Especialista.	Nombre: Gestionar perfil de usuario.
Prioridad de Negocio: Alta.	Puntos estimados: 2.
Nivel de complejidad: Media.	Iteración asignada: 1.
Descripción: Se podrá adicionar, modificar, eliminar y ver los datos del perfil del usuario.	
Observaciones: Solo el usuario puede gestionar sus propios datos.	

Tabla 4: HU Gestionar información de ejercicios.



Capítulo 2: Propuesta de solución
*Sistema Recomendador de ejercicios físicos para estudiantes de la
Universidad de las Ciencias Informáticas*

Historia de usuario	
No: 2 Usuario: Administrador, Especialista	Nombre: Gestionar información de ejercicios
Prioridad de Negocio: Alta	Puntos estimados: 2.5
Nivel de complejidad: Alta	Iteración asignada: 1
Descripción: Se podrá adicionar, modificar, eliminar y ver las noticias de ejercicios físicos. Permite al administrador y especialista adicionar, eliminar, modificar y ver las noticias, además de insertar los metadatos que describen la información para su posterior recomendación a los usuarios, mientras que al usuario autenticado solo le brindara la información del artículo.	
Observaciones:	

Tabla 5: HU Gestionar información de ejercicios.

Historia de usuario



Capítulo 2: Propuesta de solución
Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

No: 3. Usuario: Todos los usuarios del sistema	Nombre: Calcular el índice de masa corporal (IMC)
Prioridad de Negocio: Alta	Puntos estimados: 1
Nivel de complejidad: Baja	Iteración asignada: 1
Descripción: Permite calcular a todos los usuarios del sistema su índice de masa corporal, si el usuario tiene un perfil creado le calcula el IMC automáticamente y le muestra un mensaje informándole su condición física, de no tener un perfil creado el sistema muestra una página donde tendrá que insertar el tamaño en metros y el peso en kilogramos y luego el sistema muestra un mensaje.	
Observaciones:	

2.6 Estimación de esfuerzos por HU

La estimación de esfuerzos la establecen los programadores utilizando las semanas como medida. Las HU deben ser programadas en un tiempo estimado máximo de 3 semanas. Si la estimación supera este tiempo la HU deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. En caso de que el esfuerzo sea menor a 1 semana la HU deberá ser combinada con otra HU.

Tabla 6: Estimación de esfuerzos por HU

--	--	--



Capítulo 2: Propuesta de solución
*Sistema Recomendador de ejercicios físicos para estudiantes de la
Universidad de las Ciencias Informáticas*

No.	Historia de Usuario	Estimación (semanas)
1	Gestionar perfil de usuario	2
2	Gestionar información de ejercicios	2.5
3	Calcular índice de masa corporal	1
4	Recomendar ejercicios físicos	3.0
5	Enviar solicitud	1.5
6	Eliminar solicitud	1.5
7	Historial del usuario	1
8	Crear contenido	2.0
9	Autenticar usuario	1



Capítulo 2: Propuesta de solución
*Sistema Recomendador de ejercicios físicos para estudiantes de la
Universidad de las Ciencias Informáticas*

10	Buscar información	1
11	Gestionar encuesta	1
12	Gestionar noticia	1
13	Gestionar foro	1
14	Gestionar galería de imagen	1
15	Gestionar enlace	1
16	Gestionar galería de video	1
17	Enviar correo	1

2.7 Plan de iteraciones

En el plan de iteraciones se especifican cuáles son las HU definidas por el cliente que serán implementadas en cada iteración del sistema. Cada HU se traduce en tareas específicas de programación y se definen las pruebas de aceptación que serán realizadas al final de cada iteración. Al finalizar cada iteración la aplicación tendrá implementadas funcionalidades que darán cumplimiento a los objetivos propuestos. De esta manera se garantiza que el cliente tenga un prototipo del sistema con un porcentaje de funcionalidades listas para ser probadas y tomar decisiones en cuanto a cambios que puedan requerirse para mantener el camino correcto rumbo a los objetivos propuestos.

Iteración 1: En la primera iteración se entregarán las funcionalidades que tienen prioridad alta para el cliente, correspondiendo a las HU 1, 2, 3, las cuales son:

- 1- Gestionar perfil de usuario
- 2- Gestionar información de ejercicios
- 3- Calcular índice de masa corporal

Iteración 2: En esta iteración se realizarán las restantes HU que son importantes para el cliente, siendo estas 4, 5, 6, 7, 8, las cuales tendrán como funcionalidad:

- 4- Recomendar ejercicios físicos
- 5- Enviar solicitud
- 6- Eliminar solicitud
- 7- Historial del usuario
- 8- Crear contenido

Iteración 3: En esta iteración se implementan las HU de baja prioridad para el cliente pero no menos importantes que las anteriores para los desarrolladores. Estas son: 9, 10, 11, 12, 13, 14, 15, 16, 17 que se definen por:

- 9- Autenticar usuario
- 10- Buscar información
- 11- Gestionar encuesta
- 12- Gestionar noticia
- 13- Gestionar foro
- 14- Gestionar galería de imagen
- 15- Gestionar enlace
- 16- Gestionar galería de video
- 17- Enviar correo

2.8 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 la prioridad que el cliente le asigna a cada HU y la complejidad que estas tengan.



Capítulo 2: Propuesta de solución
*Sistema Recomendador de ejercicios físicos para estudiantes de la
Universidad de las Ciencias Informáticas*

Tabla 7: Plan de duración de las iteraciones

Iteración	Orden de las HU	Duración total
1	<ul style="list-style-type: none">➤ Gestionar perfil de usuario➤ Gestionar información➤ Calcular el índice de masa corporal (IMC)	5.5 semanas
2	<ul style="list-style-type: none">➤ Recomendar ejercicios físicos➤ Enviar solicitud➤ Eliminar solicitud➤ Historial del usuario➤ Crear contenido	9.3 semanas
3	<ul style="list-style-type: none">➤ Autenticar usuario.➤ Buscar información.➤ Gestionar encuesta.➤ Gestionar noticias.➤ Gestionar foro.➤ Gestionar galería de imágenes.➤ Gestionar enlaces.➤ Gestionar galerías de videos.➤ Enviar correo.	8 semanas

2.9 Plan de entregas

En el plan de entregas se establecen el orden y las HU que se agrupan para conformar una entrega. Este plan tiene como finalidad reflejar la duración de cada iteración, lo que ayuda a tener una idea aproximada del tiempo que durará la confección del sistema en su totalidad. Este cronograma será el resultado de una reunión entre el equipo de desarrollo y el cliente, permitiendo alcanzar un mayor entendimiento en la implementación del sistema.

Tabla 8: Plan de entregas

Historia de Usuario	1ra Iteración	2da Iteración	3ra Iteración
Gestionar perfil de usuario	V 1.0	Finalizado	Finalizado
Gestionar información	V 1.0	Finalizado	Finalizado
Calcular el índice de masa corporal (IMC)	V 1.0	Finalizado	Finalizado
Recomendar ejercicios físicos	-	V 1.0	Finalizado
Enviar solicitud	-	V 1.0	Finalizado
Eliminar solicitud	-	V 1.0	Finalizado
Historial del usuario		V 1.0	Finalizado
Crear contenido	-	V 1.0	Finalizado
Autenticar usuario	-	-	V 1.0
Buscar información	-	-	V 1.0



Capítulo 2: Propuesta de solución

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Gestionar encuesta			V 1.0
Gestionar noticias	-	-	V 1.0
Gestionar foro	-	-	V 1.0
Gestionar galería de imágenes	-	-	V 1.0
Gestionar enlaces	-		V 1.0
Gestionar galerías de videos	-	-	V 1.0
Enviar correo	-	-	V 1.0

2.10 Prototipo de interfaz de usuario

Para mostrar una vista preliminar del sistema que se va a desarrollar se crea un prototipo de interfaz, el cual es de fácil modificación conteniendo todas las características definidas de la propuesta de solución. La propuesta que se muestra a continuación parte de la interfaz inicial que presentará el Sistema Recomendador de ejercicios físicos para los estudiantes de la Universidad de las Ciencias Informáticas. Esta propuesta puede cambiar en algún momento, según los criterios del cliente.



Banner

INICIO - Videos - Galería - Contenido - Objetivos - Bibliografía - Autores - Acerca de – Idioma

Pruebas funcionales <u>iniciales</u> <ul style="list-style-type: none">➤ Abdominales de tronco➤ Planchas➤ Test de Cooper➤ Test de <u>Ruffier</u>	Área de contenido	<input type="text"/> buscador								
Propuesta de ejercicios <ul style="list-style-type: none">➤ Ejercicios de calentamiento➤ Ejercicios de estiramiento➤ Ejercicios de movilidad articular		Descargas <ul style="list-style-type: none">➤ <u>Biceps</u>➤ <u>Triceps</u>➤ Trapecio➤ Espalda➤ Pecho➤ Pierna➤ Abdomen➤ Antebrazo								
Ejercicios para el desarrollo de la fuerza <ul style="list-style-type: none">➤ Planchas➤ Paralelas➤ Barra fija➤ Abdominales de tronco		<u>Fisiculturismo</u>								
Sitios Recomendados <ul style="list-style-type: none">- Primer sitio- Segundo sitio- Tercer sitio	Blog de Deportes <ul style="list-style-type: none">- Primer blog- Segundo blog- Tercer blog	VISITAS 189291 <table><tr><td>Hoy</td><td>318</td></tr><tr><td>Ayer</td><td>311</td></tr><tr><td>Esta semana</td><td>629</td></tr><tr><td>Total</td><td>189291</td></tr></table>	Hoy	318	Ayer	311	Esta semana	629	Total	189291
Hoy	318									
Ayer	311									
Esta semana	629									
Total	189291									

Figura 3: Equipo de desarrollo, Prototipo de Interfaz de usuario, 2 de Abril de 2013



2.11 Tarjetas CRC

Según **Calero Solís, M. 2003**: "el diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar" y a su vez Joskowicz expresa que "la metodología XP hace especial énfasis en los diseños simples y claros".

XP a diferencia de la mayoría de las metodologías que usan diagramas para el desarrollo de modelos, usa tarjetas para el desarrollo de las clases. Estas tarjetas son llamadas CRC (clases, responsabilidad, colaboración). Son sencillas de utilizar y entender y permiten al equipo de trabajo en su totalidad participar en el diseño del sistema. Además estas tarjetas son una herramienta de reflexión en el diseño de software orientado a objeto. A continuación se muestran las tarjetas CRC de las HU de las funcionalidades críticas de la propuesta de solución. En los anexos se encuentran las tarjetas CRC de las restantes HU (Ver Anexo 2):

Tabla 9: Tarjeta CRC Gestionar perfil de usuario

Gestionar perfil de usuario.	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> ✓ Adicionar perfil ✓ Eliminar perfil ✓ Modificar perfil ✓ Ver perfil 	<ul style="list-style-type: none"> ✓ mb_overview ✓ adminScmb_menu ✓ usuarios ✓ insertar_medio_m ✓ insertar_medio_m_validate ✓ insertar_medio_m_submit ✓ update_mb ✓ update_mb_validate ✓ update_mb_submit ✓ modificar_usuario ✓ modificar_usuario_submit ✓ modificar_usuario_validate



Capítulo 2: Propuesta de solución
Sistema Recomendador de ejercicios físicos para estudiantes de la
Universidad de las Ciencias Informáticas

	<ul style="list-style-type: none"> ✓ delete_mb ✓ adminScmb_schema ✓ adminScmb_install ✓ llena_select ✓ datos_de
--	--

Tabla 10: Tarjeta CRC Gestionar información de ejercicios

Gestionar información de ejercicios	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> ✓ Adicionar información de ejercicios ✓ Eliminar información de ejercicios ✓ Modificar información de ejercicios ✓ Ver información de ejercicios 	<ul style="list-style-type: none"> ✓ nuevo_ejercicio ✓ nuevo_ejercicio_validate ✓ nuevo_ejercicio_menu ✓ nuevo_ejercicio_submit ✓ nuevo_ejercicio_validate ✓ nuevo_ejercicio_schema ✓ nuevo_contenido_ejercicio ✓ nuevo_contenido_ejercicio_submit ✓ nuevo_tipo_ejercicio ✓ nuevo_tipo_ejercicio_submit ✓ nuevo_categoria ✓ nuevo_categoria_submit ✓ nuevo_clasificardia ✓ nuevo_clasificardia_submit ✓ ne_overview ✓ update_ejercicio ✓ delete_ejercicio

Tabla 11: Tarjeta CRC Calcular el índice de masa corporal (IMC)

Calcular el índice de masa corporal (IMC)	
Responsabilidades	Colaboradores
✓ Calcular el índice de masa corporal del usuario.	✓ imc2_menu ✓ usuario ✓ imc2 ✓ imc2_validate ✓ imc2_submit

Conclusiones

En este capítulo se describen las fases de Planeación y Diseño según la metodología XP, haciendo énfasis en los artefactos generados en las mismas. Se define el personal relacionado con el sistema especificando las funcionalidades de las cuales podrá hacer uso cada uno de los usuarios en el futuro desarrollo de las mismas. Se definieron y redactaron las historias de usuario, así como las tareas de ingeniería correspondientes a cada una de ellas. Se elaboraron 8 tarjetas CRC que representan las funcionalidades que se implementan definiendo las responsabilidades y los colaboradores.



Capítulo 3: Implementación y prueba del sistema

3.1 Introducción

El presente capítulo se centra en las fases de Desarrollo y Prueba de la metodología seleccionada, tomando como base el proceso de Planificación y Diseño descrito en el capítulo anterior, lo que dio paso al desarrollo de entregas funcionales que dan cumplimiento a las necesidades del cliente. Se describe el tema usado y los módulos configurados, junto a los desarrollados por el equipo de trabajo que dan cumplimiento a los requerimientos definidos por el cliente. Se realizan las pruebas propuestas por la metodología seleccionada, documentando todos los resultados arrojados.

3.2 Estructura de las tablas agregadas

Con la instalación del CMS Drupal 7.8 se crean un total de 74 tablas para el manejo y almacenamiento de los datos (como ejemplo: los contenidos, bloques, datos de nodos, usuarios, entre otros). Este número de tablas crece con el desarrollo de las nuevas funcionalidades incluidas en los módulos debido a la necesidad de almacenar datos de importancia para los usuarios, obteniéndose como resultado las tablas que se muestran en la Figura 4.



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

nuevo_categoria		
+idc	integer(11)	Nullable = false
category	varchar(255)	Nullable = false

nuevo_contenido		
+idc	integer(11)	Nullable = false
conten	varchar(255)	Nullable = false

nuevo_tipo		
+idt	integer(11)	Nullable = false
tipo	varchar(255)	Nullable = false

primer_historial		
id_history	integer(11)	Nullable = false
+id_usuario	integer(11)	Nullable = false
+user	varchar(255)	Nullable = false
altura	integer(11)	Nullable = false
peso	integer(11)	Nullable = false
sexo	varchar(255)	Nullable = false
tep	varchar(255)	Nullable = false
categoria	varchar(255)	Nullable = false
cfure	varchar(255)	Nullable = false
dtheuv	varchar(255)	Nullable = false
enfermedades	varchar(255)	Nullable = false
nombre	varchar(255)	Nullable = false
tiempo	integer(11)	Nullable = false
tabla	integer(10)	Nullable = true
base	integer(10)	Nullable = true
datos	integer(10)	Nullable = true

nuevo_ejercicio		
+id	integer(11)	Nullable = false
titulo	varchar(255)	Nullable = false
cuerpo	varchar(255)	Nullable = false
seleccion	varchar(255)	Nullable = false
clasificardia	varchar(255)	Nullable = false
tipoejer	varchar(255)	Nullable = false
pesomenor	integer(11)	Nullable = false
pesomayor	integer(11)	Nullable = false
sexo	varchar(255)	Nullable = false
category	varchar(255)	Nullable = false
catatle	varchar(255)	Nullable = false
temeje	varchar(255)	Nullable = false
enfermo	varchar(255)	Nullable = false

segundo_historial		
id_history	integer(11)	Nullable = false
+id_usuario	integer(11)	Nullable = false
+user	varchar(255)	Nullable = false
altura	integer(11)	Nullable = false
peso	integer(11)	Nullable = false
sexo	varchar(255)	Nullable = false
tep	varchar(255)	Nullable = false
categoria	varchar(255)	Nullable = false
cfure	varchar(255)	Nullable = false
dtheuv	varchar(255)	Nullable = false
enfermedades	varchar(255)	Nullable = false
nombre	varchar(255)	Nullable = false
tiempo	integer(11)	Nullable = false
tabla	integer(10)	Nullable = true
base	integer(10)	Nullable = true
datos	integer(10)	Nullable = true

deportes		
id_usuario	integer(11)	Nullable = false
+user	varchar(255)	Nullable = false
altura	integer(11)	Nullable = false
peso	integer(11)	Nullable = false
sexo	varchar(255)	Nullable = false
tep	varchar(255)	Nullable = false
categoria	varchar(255)	Nullable = false
cfure	varchar(255)	Nullable = false
dtheuv	varchar(255)	Nullable = false
enfermedades	varchar(255)	Nullable = false
nombre	varchar(255)	Nullable = false
tiempo	integer(11)	Nullable = false
tabla	integer(10)	Nullable = true
base	integer(10)	Nullable = true
datos	integer(10)	Nullable = true

nuevo_clasificardia		
+idcl	integer(11)	Nullable = false
clasificardia	varchar(255)	Nullable = false

contador_pagina		
idcp	integer(11)	Nullable = false
+id_page	integer(11)	Nullable = false
cont_page	integer(11)	Nullable = false

clasificacion		
idcl	integer(11)	Nullable = false
+user	varchar(255)	Nullable = false
dias	varchar(255)	Nullable = false

Figura 4: Estructura de las tablas añadidas a la base de datos de Drupal, 16 de Mayo de 2013

3.3 Temas en Drupal

Un tema (*theme*) es una máscara para un sitio que puede ser modificado sin cambiar las funcionalidades implementadas y los contenidos creados. En el contexto de Drupal, el término tema es un conjunto de archivos inter-relacionados responsables del aspecto visual. El empleo del patrón Modelo Vista Controlador (MVC) y el uso de temas en Drupal hacen que este sea excepcionalmente flexible para trabajar con la interfaz, debido a que las funcionalidades del sitio están en gran medida desvinculadas de la presentación y es muy fácil cambiar el aspecto visual (25).

3.3.1 Tema del Sistema Recomendador de ejercicios físicos

Es posible crear un nuevo tema desarrollándolo desde cero, o partiendo de uno ya creado. Hay cientos de temas pre-construidos para Drupal 7.8 (25), por supuesto, con atributos muy diferentes, pero si se desea comenzar de uno previamente realizado, es importante encontrar el tema lo más parecido posible a lo que



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

se está buscando con el fin de reducir la cantidad de trabajo necesario para hacerlo coincidir con el diseño previsto. Por las especificaciones y necesidades del cliente, se crea el tema SRTema, partiendo de la estructura de las páginas y el diseño definidos en el capítulo anterior.

Lo primero para crear el tema es seleccionar el nombre, utilizando en este caso SRTema y ubicarlo en el directorio *deportes/sites/all/themes* usando el nombre definido, a partir de lo cual el tema a desarrollar se encontrará en la dirección *deportes/sites/all/themes/SRTema*. Luego se crea un archivo con la información básica tal como el nombre, versión, regiones, CSS, entre otras y se guarda con la extensión **.info** con lo que se agrega a los registros de Drupal el nuevo tema. Para personalizar la apariencia visual se utilizan los CSS definiendo las clases y los identificadores, se usa como herramienta de apoyo la extensión Firebug versión 1.10.6 de Firefox 20.01 para facilitar el trabajo.

El tema SRTema incluye los siguientes archivos:

- *block.tpl.php*
- *comment.tpl.php*
- *comment-wrapper.tpl.php*
- *html.tpl.php*
- *maintenance-page.tpl.php*
- *node.tpl.php*
- *page.tpl.php*
- *style.css*
- *suckerfish-menu.css*
- *superfish.css*
- *slideshow.js*
- *superfish.js*
- *template.php*
- *theme-settings.php*

A continuación se describen brevemente las plantillas necesarias en casi cualquier proyecto y que por su importancia fueron críticas para el desarrollo del tema SRTema y su funcionamiento:

html.tpl.php

De todas las plantillas desarrolladas para el tema esta es la principal que provee el diseño global del sitio.



page.tpl.php

La plantilla `page.tpl.php` define el cuerpo de las páginas entre las etiquetas `<body>`, donde se construye la estructura de la página mediante el uso de etiquetas `<div>`.

node.tpl.php

Esta plantilla define como los nodos son visualizados en una página. A diferencia de la plantilla de la página, que se interpreta una sola vez, esta se interpreta una vez por cada nodo que se muestre en la página.

block.tpl.php

Plantilla del tema que define como serán visualizados los bloques.

comment.tpl.php

Esta plantilla del tema corresponde al módulo de comentarios. Es la más sencilla de todas. Compone una sección con el título, autor, fecha y el comentario en sí mismo.

style.css

Este fichero es una hoja de estilos donde se incluyeron todos los identificadores y clases con sus atributos, utilizados en el tema.

styles.ie6.css, styles.ie7.css

Estas dos hojas de estilo fueron creadas para mantener la apariencia del tema en el navegador Internet Explorer en su versión 6 o 7 debido a que este navegador no interpreta de igual forma las especificaciones HTML y CSS.

3.3.2 Instalación del tema SRTema

El CMS Drupal 7 permite instalar los temas mediante dos vías: instalación automática desde la página oficial de Drupal (www.drupal.org) o mediante la forma manual ya existente en versiones anteriores que requiere de conocimientos básicos para mover archivos desde el equipo local hasta el servidor (19). A continuación se explica la instalación manual (21):

Instalación Manual

- Una vez que tenga acceso a su servidor copie la carpeta contenedora del tema en el directorio `deportes/sites/all/themes`.



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

- Para los próximos pasos, acceder al vínculo Apariencia de la interfaz de administración del sitio a través de su navegador, mostrando un listado con todos los temas disponibles del sitio.
- Para utilizar el nuevo tema primero debe ser activado haciendo clic en el enlace Activar debajo de la descripción del tema.
- A continuación para establecerlo como predeterminado se hace clic en el enlace *Activar y Cambiar por Defecto*.
- Por último hacer clic en el botón *Guardar Configuración* y el sistema instalará el tema

3.4 Módulos en Drupal

Los módulos en Drupal son un conjunto de funcionalidades con la capacidad de integrarse en una instalación añadiendo y/o mejorando el funcionamiento del sitio en áreas como: administración, gestión de contenidos, gestión de usuarios, entre otros. Estos módulos son desarrollados y compartidos libremente por la comunidad de usuarios y desarrolladores de Drupal, por lo que se hace necesario mantener el sitio que se realiza actualizado y en un estado permanente de revisión y mejora continua (21).

Para la instalación de nuevos módulos, como mínimo se deben tener los conocimientos necesarios para acceder al servidor y copiar los ficheros del módulo en el directorio `sites/all/modules`. Su complejidad depende de cada módulo en particular, siendo por lo general muy sencillo. La activación del módulo es mediante el vínculo *módulo* que brinda el menú con los privilegios administrativos en la parte superior del sistema (22).

Muchas de las características del funcionamiento básico están disponibles en una gran cantidad de módulos, brindando numerosas temáticas de alta calidad. Estos módulos pueden proveer nuevos tipos de contenidos tales como foros, blogs, entre otros, o brindar servicios tales como la búsqueda de información, envío de notificaciones, entre otras.

3.4.1 Módulos estudiados de Drupal 7.8

Para la versión 7.8, Drupal tiene una amplia variedad de módulos implementados. Se estudiaron algunos de los módulos existentes para seleccionar los que pueden ser utilizados en el desarrollo del sistema. A continuación se listan los módulos que aportan funcionalidades para la propuesta de solución.

Locale



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

El módulo *locale* agrega la capacidad de gestionar idiomas y permitió la traducción de la interfaz de usuario del portal al idioma español.

Search

Este módulo permitió agregar un formulario de búsqueda básica y/o avanzada en el portal.

Poll

Con este módulo se incluyó un sistema de votos sobre determinados tópicos en forma de encuesta cuyos resultados se muestran en un bloque.

Path

Permitió la creación de URL personalizadas para direcciones de Drupal.

Galeryformatter

Este módulo permitió crear una galería dinámica de imágenes a partir de un conjunto de medias de ese tipo.

Visitors

Muestra las estadísticas de las visitas en el sistema.

3.4.2 Paquetes de módulos desarrollados para el sistema

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

A continuación se describen los nuevos módulos desarrollados e incluidos en la instalación de Drupal. En su desarrollo están reflejados los patrones: *Bridge*, *Chain of Responsibility*, *Command* y *Singleton* debido a la forma que Drupal establece para escribir el código.

El primer paso en la creación de estos módulos es un archivo con la información de los mismos y otro para crear la estructura de las tablas que utilizan.

Para dar cumplimiento a las necesidades planteadas por el cliente, además de configurar los módulos de Drupal descritos en el epígrafe anterior, se precisa implementar las siguientes funcionalidades:

REF (Recomendar Ejercicios Físicos)

Este módulo permite adicionar, eliminar, modificar y ver los perfiles de los usuarios. Cuando el usuario se autentica tiene el privilegio de realizar todas estas acciones pero no puede modificar, eliminar, ver y



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

adicionar los datos de otros usuarios. Los administradores solo pueden realizar algún tipo de cambios del usuario cuando este lo solicite y directamente en la base de datos.

Historias de usuario que implementa: Gestionar perfil de usuario, Recomendador de ejercicios físicos, Enviar solicitud.

Tablas: *deportes, solicitud_user, clasificacion, contador_pagina*

Para la implementación del módulo se definieron varios métodos: `usuarios()` es el encargado de mostrarle al usuario si tiene un perfil creado con los datos introducidos anteriormente, de no tener datos automáticamente llama a la función `insertar_medio_m()` que es la encargada de mostrarle un formulario para que el usuario inserte sus datos físicos, luego el método `insertar_medio_m_validate()` se encarga de verificar que los datos del usuario sean válidos, mientras que el método `insertar_medio_m_submit()` se encarga de realizar una consulta a la base de datos para guardar los parámetros de forma interna. Por petición del cliente el sistema debe permitir eliminar y modificar el perfil por el propio usuario realizando esta acción mediante el método `insertar_medio_m_delete()` e `insertar_medio_m_update()` respectivamente, que realiza estos cambios en la base de datos, por último se creó el método `mb_overview()` que permite mostrar los datos de todos los usuarios con un perfil creado.

En este módulo no solo se administran los perfiles de usuarios, además se lleva el control de las solicitudes de los usuarios que no tengan información suficiente en la base de datos para recomendarles ejercicios, mediante el método `enviar_datos()` que muestra un formulario donde el usuario debe insertar los datos que describan su solicitud, luego el método `enviar_datos_validate()` se encarga de verificar los datos que insertó el usuario, mientras que `enviar_datos_submit()` realiza una consulta a la base de datos y guarda los datos pertinentes en la tabla *solicitud_user* para que los administradores del sistema puedan ver la solicitud y luego eliminarla mediante el método `eliminar_usuario_solicitante()` después de haber creado el paquete de ejercicio del usuario. Cuando el usuario envía la solicitud se le envía un correo a todos los administradores y especialistas del sistema notificando que existe una nueva solicitud a través del método `email_enviar_solicitud_send()` y se le envía un correo al usuario cuando el especialista o administrador del sistema crea un paquete de ejercicios a través del método `email_eliminar_solicitud_send()`.

Por último se realizaron los métodos, `ejercicios_overview()` que es el más importante del sistema ya que realiza la recomendación de ejercicios físicos a los usuarios con un perfil creado, este método por su



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

complejidad necesita del apoyo de otros métodos que le faciliten el trabajo como son: `llenar_dias_clasificacion()` que recoge la clasificación de los días existentes en la base de datos `nuevo_ejercicio`; el método `var_aux()` que su función es tomar una posición del arreglo que devuelve el método `llenar_dias_clasificacion()`; `cambiar_variable_clasificacion()` cambia el valor de la variable que devuelve el método `var_aux()`; `cambiar_clasificacion()` cuya función es cambiar la clasificación del día que tiene el usuario en la base de datos `clasificación`, y por último la función `enviar_datos_2()` que realiza la misma función de `enviar_datos()` solo que esta vez el usuario sí encontró parámetros disponibles en su perfil pero no encontró en el sistema suficiente información para recomendarle ejercicios. El otro método de peso de este módulo es `otros_ejercicios_overview()` que se encarga de recomendar los ejercicios más usados por otros usuarios con características similares al usuario on-line.

La implementación del módulo anteriormente explicado está basado en el algoritmo expuesto en el Capítulo 1, epígrafe 1.4.3, el cual plantea: realizar la sumatoria de la cantidad de visitas de todos los usuarios a una página (X) determinada más la sumatoria del tiempo que permanecieron los usuarios en dicha página (Y), dividido entre 2. La fórmula es la siguiente: $(\sum_{j=1}^t X_{ij} + \sum_{j=1}^t Y_{ij})/2$.

nuevoEjercicio

Este módulo permite adicionar, eliminar, modificar y ver las informaciones de los ejercicios a recomendar a los usuarios. El administrador o especialista tiene todos los permisos para gestionar este módulo mientras que los usuarios autenticados solo podrán ver los la información de estos, mientras que los metadatos del mismo coincidan con las descripciones del perfil del usuario, además este módulo tiene implementadas las funcionalidades que permiten eliminar solicitud y crear el paquete de ejercicios a los usuarios solicitantes además de listar los datos de los usuarios solicitantes.

Historias de usuario que implementa: Gestionar información de ejercicios, Eliminar solicitud, Crear contenido a usuarios solicitantes, Listar usuarios solicitantes.

Tablas: `nuevo_ejercicio`, `nuevo_contenido`, `nuevo_tipo`, `nuevo_categoria`, `nuevo_clasificardia`, `solicitud_temporal`, `solicitud_user`.

Los métodos que se implementaron en este módulo son de gran soporte para el Sistema Recomendador. El método `nuevo_ejercicio()` se encarga de mostrar un formulario para insertar las descripciones del



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

ejercicio y los metadatos que lo identificaron para recomendar ejercicios a los usuario, esta funcionalidad tiene también los métodos `nuevo_ejercicio_validate()` y `nuevo_ejercicio_submit()` que se encargan de verificar los datos entrados por el usuario y realizar una consulta en la base de datos para guardar los datos respectivamente. El método `nuevo_ejercicio()` de ser necesario hace un llamado a los métodos `llena_select_contenido()`, `llena_select_clasificacion()`, `llena_select_tipo()` y `llena_select_categoria()` su función es llenar los *select* que se encuentran en el formulario para llamar los atributos que existen en la base de datos, para adicionar estos atributos a estas tablas se implementaron los métodos `nuevo_contenido_ejercicio()`, `nuevo_tipo_ejercicio()`, `nuevo_categoria()` y `nuevo_clasificardia()`. Cada uno de estos métodos se encarga de adicionar un nuevo contenido, nuevo tipo, nueva categoría y nueva clasificación de día respectivamente. Estas funciones también tienen su *validate* y *submit*. Otro de los métodos es `adicionar_contador_pagina()` creando una base de datos con los datos de la cantidad de visitas y el tiempo que se demoran observando su información para ayudar al Sistema Recomendador elegir la mejor opción. Otros de los métodos implementados para la historia de usuario Gestionar información es `update_ejercicio()` para modificar los datos de los contenidos insertados, esta funcionalidad (como todas las que muestran un formulario) tiene implementado `update_ejercicio_validate()` y `update_ejercicio_submit()`. Otra de las historias de usuarios que se implementan en este módulo es Eliminar solicitud y los métodos que la conforman son `delete_solicitud()` que elimina las solicitudes de los usuarios, antes de ejecutar esta eliminación se llama al método `verificar Eliminacion()`, el cual verifica si la solicitud tiene un paquete de ejercicios con la información suficiente para recomendárselo al usuario. Si la verificación es correcta se adicionan los datos a la tabla `nuevo_ejercicio` llamando al método `adicionar_ejercicio()`. Luego se envía una notificación por correo electrónico al usuario para informarle que tiene una nueva serie de ejercicios recomendados en el sistema, y por último, se eliminan los datos guardados en la tabla `solicitud_temporal`.

historial_usuario

Este módulo permite ver la historia de usuario con las descripciones de los datos personales que ha tenido el usuario durante su período de inscripción en el sistema. A este módulo pueden acceder todos los usuarios que estén autenticados en el sistema.

Historia de usuario que implementa: Historial del usuario.



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Tablas: *primer_historial, segundo_historial.*

En este módulo solo se implementó un método: `historial_usuario_overview()` que mostrará una tabla con tres filas mostrando el historial del usuario, en la primera fila se mostrarán los primeros datos que insertó el usuario en su perfil, en la segunda fila los penúltimos datos que insertó o modificó el usuario en su perfil y en la tercera fila se mostrarán los últimos datos modificados por el usuario.

IMC

Este módulo permite calcular el índice de masa corporal (IMC) del usuario. A este módulo pueden acceder todos los usuarios del sistema.

Historia de usuario que implementa: Calcular IMC

Uno de los métodos implementados en este módulo es `usuario()` cuya función es verificar si el usuario que está realizando la petición de calcular su IMC tiene un perfil creado, de tener un perfil creado se calcula su IMC a través de la fórmula: peso del usuario (p) dividido por su altura al cuadrado (h), (p_{kg}/h_m^2). Si el usuario no tiene un perfil creado se llama al método `imc2()` que muestra un formulario para que el usuario inserte su altura y peso en metros y kilogramos respectivamente. Luego la función `imc2_validate()` valida que los datos sean válidos y la función `imc2_submit()` realiza el cálculo con la fórmula descrita anteriormente. Al final el sistema muestra un mensaje al usuario (teniendo o no un perfil creado) diciendo cómo se comporta su cuerpo en correspondencia de su altura y peso.

3.4.3 Instalación de módulos

Para instalar un módulo en el sistema lo primero que se debe realizar es copiar la carpeta contenedora de los archivos al directorio `sites/all/modules` con el nombre definido. Luego en la interfaz administrativa se visita el vínculo "Módulos" en el menú superior del sistema mostrando una lista de los 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. El grupo de módulos implementados para el portal están agrupados en el paquete *Desarrollo*. En ocasiones al instalar un módulo es necesario dirigirse hacia la página de permisos para establecer los privilegios referentes al módulo o hacia la página de configuración para cambiar las opciones del módulo. Algunas de las posibles opciones de configuración son explicadas a continuación.



3.5 Configuración del sistema

Luego de instalar el tema SRTema, se requiere realizar configuraciones básicas:

- Actualización del cron (Tareas Programadas de Windows) para iniciar las tareas automáticas del sistema.
- Registrar la información del sitio.
- Establecer el español como idioma.
- Habilitar la opción de URL limpias.
- Configurar las opciones para el correcto funcionamiento del motor de búsqueda.
- Fijar la disposición de bloques y menús.
- Crear los roles definidos con sus privilegios.
- Determinar los formatos de entradas.

Dentro de estas configuraciones, algunas se pueden llevar a cabo mediante las opciones de la interfaz de administración, habilitando y deshabilitando opciones, mientras que otras son un poco más complejas y se explicarán a continuación.

3.5.1 Menús

Los menús son un conjunto de enlaces responsables de la navegación de un sitio. Un menú consta de tres entidades: el árbol, los elementos y el enlace. Los elementos de un menú tienen propiedades que indican el estado del menú, que puede ser hoja, ampliada o colapso (51).

Existen varios tipos de menú estándares en Drupal, dos de ellos (“menú principal” y “menú secundario”) son creados por el administrador. En el tema SRTema el menú que incluye los enlaces: “Inicio”, “Mejora tu físico”, “Contactos”, “Asistente on-line”, “Mi cuenta” constituye el menú principal, siendo habilitado el vínculo cuando se activan los módulos pertenecientes a estas funcionalidades.

3.5.2 Bloques

Un bloque es esencialmente un contenedor independiente que se puede utilizar para alojar los fragmentos de texto o las funcionalidades. Los bloques normalmente se colocan en las barras laterales, el encabezado o pie de página de una plantilla mostrando los enlaces a otras páginas, información del sistema y las encuestas por citar algunos ejemplos (53).



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

En el tema SRTema se crearon un total de 8 bloques que agrupan información y vínculos específicos, algunos son agregados mediante la activación de los módulos y otros son creados mediante la interfaz de administración, Estructura\Bloques. En la barra lateral derecha están ubicados los bloques: “Pruebas”, “Ejercicios”, “Desarrollo fuerza”, “Descargas”, “Mejora tu físico” y “Navegación”. En el pie de página se muestran los bloques “Blog de deportes” y “Sitios recomendados”.

3.5.3 Roles de usuario

Con el sistema deben interactuar un conjunto de usuarios con niveles de acceso y objetivos diferentes (ver Figura 4). Drupal posee un sistema de permisos basado en roles (54). Los roles no son más que una vía de colocar a los usuarios autenticados dentro de categorías, donde las mismas limitan los niveles de acceso (22).



Figura 5: Equipo de desarrollo, Niveles de acceso por roles, 18 de mayo de 2013



3.5.4 Formato de entrada

Los formatos de entrada se definen para establecer la forma en que se procesan los textos introducidos en los formularios de Drupal. Se puede configurar un formato de texto existente o crear uno nuevo haciendo clic en el enlace *Configuración* en la parte superior de la página, seguido por el enlace para agregar un nuevo formato de texto (21).

La instalación por defecto de Drupal 7.8 incluye un grupo de filtros (que pueden extenderse individualmente o mediante módulos contribuidos que crean sus propios filtros para fines específicos), básicamente son tres: *FilteredHTML*, *FullHTML* y *Plain Text*.

3.6 Pruebas

La fase de pruebas es una de las más importantes de la metodología XP. Permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Además, permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones (25).

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores y las pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente (25).

3.6.1 Pruebas unitarias

Las pruebas unitarias o pruebas de unidad deben ser construidas antes que el código, permitiéndole a los programadores tener máxima claridad de lo que van a programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberán pasar, lo que optimizará el trabajo y el código será de mayor calidad (25) (26).

3.6.2 Pruebas de aceptación

Se considera que las pruebas de aceptación poseen un peso superior a las unitarias, pues las mismas muestran a los desarrolladores la satisfacción del cliente. Por este motivo se puede decir que estas ponen fin a una iteración dando inicio a la siguiente.

Las pruebas de aceptación se describen mediante una tabla llamada Caso de Prueba de Aceptación y son destinadas a evaluar si al final de cada iteración se obtuvo la funcionalidad requerida y son creadas sobre



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

la base de las HU. El cliente desde su punto de vista creará los casos de prueba en conjunto con los desarrolladores para probar que una HU ha sido implementada correctamente. Por cada HU se elaborarán todas las pruebas de aceptación que se necesiten para asegurar su correcto funcionamiento.

A continuación se muestran los casos de prueba de aceptación realizados para la HU Gestionar perfil de usuarios. Los restantes casos de prueba de aceptación se muestran en el Anexo 3.

Tabla 12: Caso de Prueba de Aceptación HU1_P1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: Gestionar perfil de usuario.
Nombre: Adicionar perfil de usuario.	
Descripción: Prueba para la funcionalidad que le permite al usuario adicionar los datos de su perfil.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución El usuario, en el bloque navegación, selecciona la opción “Lista de usuarios”, luego, en la página que se muestra ir a la pestaña “Crear mi perfil”. Se le muestran los campos a llenar y procede a insertar los datos requeridos del perfil. Finalmente el usuario ejecuta la acción “Aceptar” y el sistema le muestra un mensaje confirmando que la acción fue llevada a cabo de manera satisfactoria. Otra manera de adicionar el perfil es dirigiéndose nuevamente al bloque de navegación, al link “Mi perfil”, si el usuario aún no tiene un perfil creado, se muestra un página con los campos a llenar del perfil. Finalmente el usuario pulsa “Aceptar” y el sistema muestra un mensaje confirmando que la acción fue realizada con éxito.	
Resultado esperado: Los datos son registrados satisfactoriamente.	



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Evaluación de la prueba: Satisfactoria.

Tabla 13: Caso de Prueba de Aceptación HU1_P2

Caso de prueba de aceptación	
Código: HU1_P2	Historia de usuario: Gestionar perfil de usuario.
Nombre: Modificar los datos del perfil de usuario.	
Descripción: Prueba para la funcionalidad que permite modificar los datos del perfil de usuario.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de usuarios” y seguidamente en la tabla de la lista de usuarios que muestra el sistema, busca su nombre y en esa misma fila pulsa el link “Modificar”. Luego se genera una página con los datos del perfil y el usuario modifica los datos que desee, presionando después de la modificación el botón “Aceptar”. Automáticamente el sistema mostrará un mensaje confirmando que la acción fue ejecutada satisfactoriamente. Otra manera de modificar los datos es dirigiéndose al bloque de navegación y seleccionar el link “Mi perfil”, se muestra una página con los datos del perfil del usuario, se pulsa en el link “Modificar datos” y se muestra una página con los datos del perfil, el usuario modifica los datos que desee y al final de la operación presiona el botón “Aceptar”, mostrando un mensaje confirmando la satisfacción de la operación. El usuario no puede modificar otro perfil que no sea el suyo propio.	
Resultado esperado: Los datos son registrados satisfactoriamente.	



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Evaluación de la prueba: Satisfactoria.

Tabla 14: Caso de Prueba de Aceptación HU1_P3

Caso de prueba de aceptación	
Código: HU1_P3	Historia de usuario: Gestionar perfil de usuario.
Nombre: Mostrar los datos del perfil de usuario.	
Descripción: Prueba para la funcionalidad que permite mostrar los datos del perfil de usuario.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona la opción “Mi perfil”, seguidamente se muestra una página con los datos del perfil de usuario.	
Resultado esperado: El perfil de usuario es mostrado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 15: Caso de Prueba de Aceptación HU1_P4

Caso de prueba de aceptación	
Código: HU1_P4	Historia de usuario: Gestionar perfil de usuario.
Nombre: Eliminar el perfil de usuario.	
Descripción: Prueba para la funcionalidad que permite eliminar el perfil del usuario.	



Capítulo 3: Implementación y pruebas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

Condiciones de ejecución: El usuario debe estar autenticado.

Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de usuarios”, seguidamente se muestra una página con el listado de todos los usuarios que tienen un perfil creado organizados en una tabla, el usuario busca su nombre en la lista y en la misma fila que aparece pulsa el link “Eliminar” y automáticamente el sistema mostrará un mensaje confirmando que la acción fue ejecutada satisfactoriamente. El usuario no puede eliminar otro perfil que no sea el suyo propio.

Resultado esperado: El perfil de usuario es eliminado satisfactoriamente.

Evaluación de la prueba: Satisfactoria.

Conclusiones

Tomando como punto de partida el diseño propuesto en el capítulo anterior, junto a las herramientas, metodología y tecnología seleccionadas en el Capítulo 1, en el presente capítulo se abordaron los temas referentes a la implementación de la solución propuesta y las pruebas realizadas a la misma con el objetivo de evaluar el producto y determinar el nivel de conformidad del cliente. Mediante las pruebas de aceptación se pudo documentar las no conformidades detectadas en el sistema durante su desarrollo, las cuales fueron posteriormente corregidas. Al finalizar la última iteración se obtuvo un sistema completamente funcional que cumple con las especificaciones establecidas por el cliente.



Conclusiones generales

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

- El estudio del estado del arte sobre sistemas recomendadores permitió seleccionar el sistema Híbrido como el más aplicable a la propuesta de solución.
- La metodología seleccionada permitió satisfacer las necesidades del cliente mediante la definición de las HU y la construcción de las Tarjetas CRC que dieron paso a la fase de codificación.
- El diseño desarrollado para la aplicación, permitió la implementación de funcionalidades que dieron solución a la problemática descrita.
- Las pruebas de aceptación realizadas por el cliente permitieron validar el sistema demostrando el correcto funcionamiento del mismo.



Recomendaciones

Como parte del proceso investigativo llevado a cabo se recomienda:

- Enriquecer la base de conocimientos para permitir la recomendación de ejercicios a usuarios de otras edades.
- A la comunidad científica, tomar el presente trabajo como material de estudio en el desarrollo de funcionalidades similares.
- Al cliente, poner el sistema al servicio de toda la comunidad universitaria, tanto de la UCI como de otras universidades del país, adicionándole al mismo nuevas funcionalidades e integrándole servicios necesarios.



Referencias Bibliográficas

1. **Dhar, Santonu Kumar.** *Building job sites with Joomla!* s.l. : Akshara Aware, 2010. ISBN: 978-1-849512-22-0.
2. **Hejlsberg , Anders y de Icaza, Miguel.** *Framework Design Guidelines.* 2008. ISBN: 978-0-321-54561-9.
3. **Tomlinson, Todd.** *Beginning Drupal 7.* 2010. ISBN: 978-1-4302-2859-2.
4. **EllisLab, Inc.** *Sistemas Gestores de Bases de Datos.* 2010. ISBN: 1988-6047.
5. **Noble, Mark.** *Drupal 7 First Look.* s.l. : Packt Publishing, Noviembre, 2010.
6. **Welling, Luke y Thomson, Laura.** *Desarrollo Web con PHP y MySQL.* Madrid : Anaya Multimedia, 2005.
7. **FARHOOMAND, Ali F y DRURY, Don H.** *Managerial information overload.* s.l. : Communications of the ACM. ACM, Oct. 2002. Vol. 45. 127-131.
8. **HULL, R. and SU, J.** *Tools for composite web services: a short overview.* 2005.
9. **Balabanovic, M. y Shoham, Y.** Content-based, collaborative recommendation. *Communications of ACM* . 1997. 40.
10. **BECHHOFFER, S, y otros.** OilEd: a Reason-able Ontology Editor for the Semantic Web . *Lecture Notes in Computer Science.* 2001. Vol. 2174, 396.
11. **BONHARD, P, HARRIES, C y MCCARTHY, J.** Accounting for taste: using profiles imilarity to improve recommender systems . *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Montréal, Québec, Canada : ACM Press, April 22, 2006. 1057-1066.
12. **Herlocker, J, y otros.** Evaluating Collaborative Filtering Recommender Systems en. s.l. : ACM, 2004. Vol. *Transaction on Information Systems*, 22.
13. **Dai, H. y Mobasher, B.** "Integrated semantic knowledge with web usage mining and personalization" . *Web Mining: Applications and Techniques.* s.l. : IRM Press .



Referencias bibliográficas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

14. **Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. y Sartin, M.** Combining content-based and collaborative filters in an online newspaper” . *Proceedings of ACM SIGIR'99 Workshp on Recommender Systems: Algorithms and Evaluation*. 1999.
15. **Guo, X.** Personalized Government Online Services with Recommendation Techniques. University Graduate School, University of Technology Sydney : s.n., 2006.
16. *V Encuentro xBase. Calero Solís , Manuel.* MADRID : s.n., 2003.
17. *Sistemas Recomendadores: Un enfoque desde los algoritmos genéticos. Santos, Carlos Oswaldo.* 2006, Vol. 9.
18. **Jensen, Finn V. y Nielsen, Thomas D.** *Bayesian Networks and Decision Graphs. 2nd.* s.l. : Springer Publishing Company, Incorporated, 2007.
19. **Blanco, Yolanda.** *Tesis doctoral propuesta metodológica para el razonamiento semántico en sistemas de recomendación personalizada y automática. Aplicación al caso de contenidos audiovisuales.* 2007.
20. **Mercer, David.** *Drupal 7.* s.l. : Packt Publishing, Septiembre, 2010. .
21. **Noble, Mark.** *Drupal 7 First Look.* s.l. : Packt Publishing, Noviembre, 2010.
22. **Tomlinson, Todd y VanDyk, John K.** *Pro Drupal 7 Development, 3rd.Edition.* s.l. : Apress, Diciembre, 2010.
23. **Miles, Earl y Miles, Lynette.** *DrupalsBuilding Blocks.* s.l. : Addison Wesley, Diciembre, 2010.
24. **Travis, Brian.** *Pro Drupal 7 for Windows Developers.* s.l. : Apress, Febrero, 2011.
25. **Tobón, Luis Miguel Echeverry y Carmona, Luz Elena Delgado.** *Caso práctico de la metodología XP al desarrollo de software.* Universidad Tecnológica de Pereira : s.n., 2007.
26. **Joskowicz, José.** *Reglas y Prácticas en eXtremeProgramming. Doctorado de Ingeniería Telemática de la Universidad de Vigo.* España : s.n., 2008.
27. **Pressman, Roger.** *Ingeniería de software: un enfoque práctico.* 2005.
28. **Rodríguez Corbea, Maite y Ordóñez Pérez, Meylin .** *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA.* La Habana : s.n., julio, 2007.



Referencias bibliográficas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

29. **Canós, José H, Letelier, Patricio y Penadés, María del Carmen.** *Metodologías ágiles en el desarrollo del software.* Valencia : s.n.
30. **ALFONSO, X. Y. J. M.** *Introducción a los Sistemas de Gestión de Contenido.* 2008.
31. **HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., and RIEDL, J. T.** *Evaluating collaborative filtering recommender systems.* 2004.
32. **BAEZA-YATES, R. and RIBEIRO-NETO, B.** *Modern Information Retrieval.* AdissonWesley. 1999.
33. **BERNERS-LEE, T.** *Weavingthe web.* Harper, San Francisco : s.n., 1999.
34. **Vozalis, E.** *Analysis of Recommender Systems' Algorithms.*
35. **Galán, Sergio y Nieto, Manuel.** *Filtrado colaborativo y Sistemas de Recomendación.* 2007.
36. Comunidad CodeIgniter. *Comunidad CodeIgniter en español.* [En línea] 14 de noviembre de 2009. [Citado el: 25 de octubre de 2012.] <http://comunidadcodeigniter.wordpress.com/>.
37. xprogramming.com. [En línea] 2012. <http://xprogramming.com/index.php>.
38. **Casas, Ester.** Conferencia Internacional de Software Libre. [En línea] [Citado el: 14 de noviembre de 2012.] http://malaga08.opensourceworldconference.com/papers/Dia22/Sala%204/Casas_235.pdf.
39. W3C Recommendation. [En línea] 2000. <http://www.w3.org/TR/xhtml1/>.
40. desarrolloweb.com. [En línea] 2012. <http://desarrolloweb.com/>.
41. Drupal. [En línea] 2013.
https://drupal.org/project/themes?solsort=iss_project_release_usage%20desc.
42. Slideshare. [En línea] <http://www.slideshare.net/>.
43. Drupal Hispano. [En línea] <http://drupal.org.es/Drupal>.
44. Instituto Tecnológico de Veracruz. [En línea]
<http://www.prograweb.com.mx/pweb/0203ladoServidor.html>.
45. Universidad de las Ciencias Informáticas. [En línea] 2013. <http://www.uci.cu/mision>.



Referencias bibliográficas

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

46. **Palacio, Juan.** Navegapolis.com. [En línea] 2006. [Citado el: 25 de febrero de 2013.] http://www.navegapolis.net/component/option,com_frontpage/Itemid,1/.
47. Maestros del web. [En línea] 2007. <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
48. Mozilla Developer Network. [En línea] <https://developer.mozilla.org/en-US/docs/Web/JavaScript?redirectlocale=en-US&redirectslug=JavaScript>.
49. Mozilla Developer Network. [En línea] <https://developer.mozilla.org/es/docs/CSS>.
50. Luciano. IDEs Programación. [En línea] 2008. [Citado el: 15 de marzo de 2013.] <http://luauf.com/entornos-de-desarrollo-integrado-para-java>.



Bibliografía

1. **Dai, H. y Mobasher, B.** "Integrated semantic knowledge with web usage mining and personalization" . *Web Mining: Applications and Techniques*. s.l. : IRM Press .
2. **Vozalis, E.** *Analysis of Recommender Systems' Algorithms*.
3. **Casas, Ester.** Conferencia Internacional de Software Libre. [En línea] [Citado el: 14 de noviembre de 2012.] http://malaga08.opensourceworldconference.com/papers/Dia22/Sala%204/Casas_235.pdf.
4. **Herlocker, J, y otros.** Evaluating Collaborative Filtering Recommender Systemsen. s.l. : ACM, 2004. Vol. *TransactionsonInformationSystems*, 22.
5. **Pressman, Roger.** *Ingeniería de software: un enfoque práctico*. 2005.
6. **Welling, Luke y Thomson, Laura.** *Desarrollo Web con PHP y MySQL*. Madrid : Anaya Multimedia, 2005.
7. **Blanco, Yolanda.** *Tesis doctoral propuesta metodológica para el razonamiento semántico en sistemas de recomendación personalizada y automática. Aplicación al caso de contenidos audiovisuales*. 2007.
8. Comunidad CodeIgniter. *Comunidad CodeIgniter en español*. [En línea] 14 de noviembre de 2009. [Citado el: 25 de octubre de 2012.] <http://comunidadcodeigniter.wordpress.com/>.
9. **Dhar, Santonu Kumar.** *Building job sites with Joomla!* s.l. : Akshara Aware, 2010. ISBN: 978-1-849512-22-0.
10. xprogramming.com. [En línea] 2012. <http://xprogramming.com/index.php>.
11. **Rodríguez Corbea, Maite y Ordóñez Pérez, Meylin .** *LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA*. La Habana : s.n., julio, 2007.
12. **Blanqué, Javier.** *La Mejor Metodología _Liviana de Desarrollo de Software: eXtreme Programming*. ISBN 201-61641-6.
13. **ALONSO DE HERRERA, G.** La Condición Física . [En línea] <http://www.iesgaherrera.com/ef/a> .
14. **BOUCHARD, C.** *Actividad física, condición física y salud*. Chicago, Champaign IL : s.n., 1991.



Glosario de términos

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

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

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.

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.



Glosario de términos

Sistema Recomendador de ejercicios físicos para estudiantes de la Universidad de las Ciencias Informáticas

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

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: En español Licencia Pública General, es una licencia creada por la Free Software Foundation 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.

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. Cuándo 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.



Anexos

Anexo 1:

Tabla 16: HU Recomendar ejercicios físicos.

Historia de usuario	
No: 4. Usuario: Todos los usuarios autenticados del sistema que tengan un perfil de usuario.	Nombre: Recomendar ejercicios físicos.
Prioridad de Negocio: Media.	Puntos estimados: 3
Nivel de complejidad: Alta.	Iteración asignada: 2
Descripción: Permite recomendar a todos los usuarios del sistema que estén autenticados y tengan un perfil de usuario, ejercicios físicos en dependencia de sus datos personales, también recomienda las informaciones de ejercicios más usadas por los usuarios del sistema que tengan características similares al usuario autenticado.	
Observaciones:	

Tabla 17: HU Enviar solicitud.



Historia de usuario	
No: 5 Usuario: Todos los usuarios del sistema que estén autenticados.	Nombre: Enviar solicitud.
Prioridad de Negocio: Media.	Puntos estimados: 1.5
Nivel de complejidad: Alta.	Iteración asignada: 2
Descripción: Permite enviar a los administradores y especialistas del sistema una solicitud de recomendación de ejercicios. La solicitud se envía cuando el usuario no encuentra ningún parámetro acorde a sus características e inserta otros datos inexistentes en la base de datos, al presionar el botón enviar el sistema guarda la solicitud y envía un correo a los administradores y especialistas del sistema. Otra forma de enviar la solicitud es cuando el usuario llenó los datos que se ajustaban a sus características en su perfil pero no existe en la base de datos información para recomendarle, automáticamente se le envía un correo a los administradores y especialistas y se guarda la solicitud en la base de datos.	
Observaciones:	

Tabla 18: HU Eliminar solicitud.

Historia de usuario



No: 6 Usuario: Administrador y Especialista.	Nombre: Eliminar solicitud.
Prioridad de Negocio: Media.	Puntos estimados: 1.5
Nivel de complejidad: Alta.	Iteración asignada: 2
Descripción: Permite eliminar las solicitudes enviadas por los usuarios, antes de eliminar dicha solicitud el especialista o administrador debe crear un paquete de ejercicios de acorde con las características de la solicitud, después de creado este paquete se elimina la solicitud y se envía un correo al usuario solicitante.	
Observaciones:	

Tabla 19: HU Historial del usuario.

Historia de usuario	
No: 7. Usuario: Todos los usuarios del sistema que estén autenticados y tengan un perfil creado.	Nombre: Historial del usuario.
Prioridad de Negocio: Baja.	Puntos estimados: 1



Nivel de complejidad: Baja.	Iteración asignada: 2
Descripción: Permite mostrar a los usuarios que tengan un perfil creado, un historial de como se ha comportado su cuerpo y los ejercicios que ha pedido.	
Observaciones:	

Tabla 20: HU Crear contenido.

Historia de usuario	
No: 8 Usuario: Administrador, Especialista.	Nombre: Crear contenido.
Prioridad de Negocio: Alta.	Puntos estimados: 2
Nivel de complejidad: Alta.	Iteración asignada: 2.
Descripción: Se podrá adicionar un paquete de recomendación de ejercicios físicos a los usuarios que han enviado una solicitud.	
Observaciones:	

Tabla 21: HU Autenticar Usuario.



Historia de usuario	
No: 9 Usuario: Todos los usuarios que pertenezcan al dominio UCI.	Nombre: Autenticar usuario.
Prioridad de Negocio: Alta.	Puntos estimados: 1.
Nivel de complejidad: Baja.	Iteración asignada: 3.
Descripción: Se podrán autenticar todos los usuarios que pertenezcan al dominio UCI.	
Observaciones:	

Tabla 22: HU Buscar información en el sistema.

Historia de usuario	
No: 10 Usuario: Todos los usuarios del sistema.	Nombre: Buscar información.
Prioridad de Negocio: Baja	Puntos estimados: 1



Nivel de complejidad: Baja.	Iteración asignada: 3.
Descripción: Se podrá buscar cualquier información que exista en el sistema insertando una cadena de palabras.	
Observaciones:	

Tabla 23: HU Gestionar encuesta.

Historia de usuario	
No: 11 Usuario: Administrador y especialista.	Nombre: Gestionar encuesta.
Prioridad de Negocio: Baja.	Puntos estimados: 1
Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver las encuestas realizadas por el administrador o especialista.	
Observaciones:	

Tabla 24: HU Gestionar noticias.



Historia de usuario	
No: 12 Usuario: Administrador y especialistas.	Nombre: Gestionar noticias.
Prioridad de Negocio: Baja.	Puntos estimados: 1
Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver las noticias realizadas por el administrador o especialista.	
Observaciones:	

Tabla 25: HU Gestionar foro.

Historia de usuario	
No: 13 Usuario: Todos los usuarios autenticados en el sistema.	Nombre: Gestionar foro.
Prioridad de Negocio: Baja.	Puntos estimados: 1



Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver los temas del foro por los administradores del sistema, los demás usuarios solo podrán ver y adicionar temas del foro.	
Observaciones:	

Tabla 26: HU Gestionar galería de imagen.

Historia de usuario	
No: 14 Usuario: Administrador y especialista.	Nombre: Gestionar galería de imágenes.
Prioridad de Negocio: Baja.	Puntos estimados: 1
Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver galerías de imágenes publicadas en un artículo.	
Observaciones:	

Tabla 27: HU Gestionar enlaces.



Historia de usuario	
No: 15 Usuario: Todos los usuarios del sistema.	Nombre: Gestionar enlaces.
Prioridad de Negocio: Baja.	Puntos estimados: 1
Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver enlaces a otros sitios de la universidad.	
Observaciones:	

Tabla 28: HU Gestionar galerías de videos.

Historia de usuario	
No: 16 Usuario: Todos los usuarios del sistema.	Nombre: Gestionar galerías de videos.
Prioridad de Negocio: Baja.	Puntos estimados: 1



Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá adicionar, modificar, eliminar y ver galerías de videos publicadas en un artículo.	
Observaciones:	

Tabla 29: HU Enviar correos.

Historia de usuario	
No: 17 Usuario: Todos los usuarios del sistema.	Nombre: Enviar correos.
Prioridad de Negocio: Baja.	Puntos estimados: 1
Nivel de complejidad: Baja.	Iteración asignada: 3
Descripción: Se podrá enviar un correo desde el sistema.	
Observaciones:	

Anexo 2:

Tabla 30: Tarjeta CRC Recomendar ejercicios físicos.

--



Recomendar ejercicios físicos.	
Responsabilidades	Colaboradores
✓ Recomendar ejercicios físicos según las características del usuario.	✓ ejercicios_overview ✓ llenar_dias_clasificacion ✓ var_aux ✓ cambiar_variable_clasificacion ✓ cambiar_clasificacion

Tabla 31: Tarjeta CRC Enviar solicitud.

Enviar solicitud.	
Responsabilidades	Colaboradores
✓ Guardar la solicitud en la base de datos y enviarle una notificación por el correo a los administradores del sistema.	✓ enviar_datos_2 ✓ enviar_datos ✓ eliminar_usuario_solicitante ✓ email_enviar_solicitud_send ✓ enviar_datos_submit ✓ email_enviar_solicitud_mail ✓ email_enviar_solicitud_send

Tabla 32: Tarjeta CRC Eliminar solicitud.

Eliminar solicitud.	
Responsabilidades	Colaboradores
✓ Eliminar la solicitud de la base de datos y enviarle una notificación por el correo al usuario solicitante.	✓ delete_solicitud ✓ verificar Eliminacion ✓ adicionar_ejercicio



	<ul style="list-style-type: none">✓ email_eliminar_solicitud_mail_send✓ eliminar_solicitud_temporal
--	--

Tabla 33: Tarjeta CRC Historial del usuario.

Historial del usuario.	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">✓ Mostrar los datos personales que ha tenido el usuario desde la primera vez que insertó su perfil hasta la última vez que modificó sus datos.	<ul style="list-style-type: none">✓ historial_usuario_menu✓ historial_usuario_overview✓ historial_usuario_schema

Tabla 34: Tarjeta CRC Crear contenido.

Crear contenido.	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">✓ Crearle un contenido a los usuarios que solicitaron recomendación de ejercicios.	<ul style="list-style-type: none">✓ resolver_lista_solicitudes✓ resolver_lista_solicitudes_submit

Anexo 3:

Tabla 35: Caso de Prueba de Aceptación HU2_P1

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: Gestionar información.



Nombre: Adicionar información de ejercicios.
Descripción: Prueba para la funcionalidad que permite insertar información de ejercicios físicos para el sistema.
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link "Insertar ejercicio", seguidamente el sistema muestra una página con los campos a llenar de la información del ejercicio. Luego se pulsa el botón "Enviar" y se muestra un mensaje confirmando que la acción fue ejecutada satisfactoriamente.
Resultado esperado: La información es adicionada satisfactoriamente.
Evaluación de la prueba: Satisfactoria.

Tabla 36: Caso de prueba de aceptación HU2_P2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: Gestionar información.
Nombre: Mostrar información del ejercicio.	
Descripción: Prueba para la funcionalidad que permite mostrar la información de un ejercicio.	
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.	



Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de ejercicios”, seguidamente el sistema muestra un listado con todos los ejercicios que hay en el sistema.

Resultado esperado: El listado es mostrado satisfactoriamente.

Evaluación de la prueba: Satisfactoria.

Tabla 37: Caso de prueba de aceptación HU2_P3

Caso de prueba de aceptación	
Código: HU2_P3	Historia de usuario: Gestionar información.
Nombre: Actualizar datos de los ejercicios.	
Descripción: Prueba para la funcionalidad que permite actualizar los datos de los ejercicios.	
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de ejercicios”, seguidamente el sistema muestra un listado con todos los ejercicios que hay en el sistema. Se busca el nombre del ejercicio que se quiera modificar los datos y se presiona en el link “Editar”, mostrando el sistema una página con los datos del ejercicio a modificar, se modifican los datos que desee el usuario y se presiona al final de la operación del botón modificar, el sistema muestra un mensaje notificando la satisfacción de la operación.	
Resultado esperado: Los datos son actualizados satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	



Tabla 38: Caso de prueba de aceptación HU2_P4

Caso de prueba de aceptación	
Código: HU2_P4	Historia de usuario: Gestionar información.
Nombre: Eliminar información de ejercicio.	
Descripción: Prueba para la funcionalidad que permite eliminar la información de un ejercicio.	
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de ejercicios”, seguidamente el sistema muestra un listado con todos los ejercicios que hay en el sistema. Se busca el nombre del ejercicio que se quiera eliminar pulsando en el link “Eliminar”, el sistema muestra un mensaje notificando la satisfacción de la operación.	
Resultado esperado: El listado es eliminado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 39: Caso de prueba de aceptación HU3_P1

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: Calcular el índice de masa corporal (IMC).
Nombre: Calcular el índice de masa corporal del usuario.	



Descripción: Prueba para la funcionalidad que permite calcular el IMC de un usuario.
Condiciones de ejecución: Acceder al sistema con cualquier rol.
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Calculador de IMC”, seguidamente si el usuario tiene un perfil creado el sistema muestra un mensaje con el resultado del IMC. Si el usuario no tiene un perfil creado el sistema muestra una página con los campos a llenar para calcular el IMC, seguidamente se pulsa el botón “Calcular” y el sistema muestra un mensaje con el resultado del cálculo.
Resultado esperado: El IMC es calculado satisfactoriamente.
Evaluación de la prueba: Satisfactoria.

Tabla 40: Caso de prueba de aceptación HU4_P1

Caso de prueba de aceptación	
Código: HU4_P1	Historia de usuario: Recomendar ejercicios físicos.
Nombre: Recomendar ejercicios físicos	
Descripción: Prueba para la funcionalidad que permite recomendar ejercicios físicos a los usuarios.	
Condiciones de ejecución: El usuario debe estar autenticado.	



Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Ejercicios recomendados”, seguidamente el sistema muestra un listado con todos los ejercicios que el mismo le va recomendar en el día y al final de la lista un link que al pulsarlo el sistema muestra una página con otros ejercicios recomendados, sino se accede a la página se debe mostrar un mensaje explicando las causas de no haber recomendado otro tipo de ejercicios. Si al pulsar el link “Ejercicios recomendados” no se muestra la página con los ejercicios ver caso de prueba de aceptación

HU5_P1

Resultado esperado: El listado de ejercicios recomendados es mostrado satisfactoriamente.

Evaluación de la prueba: Satisfactoria.

Tabla 41: Caso de prueba de aceptación HU5_P1

Caso de prueba de aceptación	
Código: HU5_P1	Historia de usuario: Enviar solicitud.
Nombre: Enviar solicitud.	
Descripción: Prueba para la funcionalidad que permite enviar una solicitud.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución: El usuario, cuando va a adicionar su perfil descrito en el caso de prueba HU1_P1, si no encuentra los datos que se adecuen a las características de su perfil accede al link que aparece al lado del botón “Enviar” de la página de crear perfil y el sistema muestra una página con los campos a llenar para enviar la solicitud, después de llenado los campos se pulsa el botón enviar y se envía una solicitud. El sistema muestra un mensaje de satisfacción de la operación. Otra manera de enviar una solicitud es cuando el usuario llena los datos en la ventana de	



crear perfil y al enviar los datos el sistema no encuentra una serie de ejercicios a recomendar, entonces se muestra un mensaje notificando que una solicitud ha sido enviada a los administradores del sitio y la última manera de enviar una solicitud es cuando el usuario pulsa en el link “Ejercicios recomendados” y el sistema no tiene ejercicios a recomendar, se le muestra un mensaje notificando el envío de la solicitud.

Resultado esperado: La solicitud es enviada satisfactoriamente.

Evaluación de la prueba: Satisfactoria.

Tabla 42. Caso de prueba de aceptación HU6_P1

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: Eliminar solicitud.
Nombre: Eliminar solicitud.	
Descripción: Prueba para la funcionalidad que permite eliminar una solicitud.	
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de solicitudes”, seguidamente el sistema muestra un listado con todas las solicitudes que hay registradas. Se busca la solicitud que se quiera eliminar siempre y cuando tenga esta solicitud un paquete de ejercicios creado, se pulsa en el link “Eliminar” y el sistema muestra un mensaje de notificación de la operación y le envía un mensaje por el correo al usuario solicitante.	
Resultado esperado: La solicitud es eliminada satisfactoria mente.	
Evaluación de la prueba: Satisfactoria.	



Tabla 43: Caso de prueba de aceptación HU7_P1

Caso de prueba de aceptación	
Código: HU7_P1	Historia de usuario: Historial de usuario.
Nombre: Historial de usuario.	
Descripción: Prueba para la funcionalidad que permite mostrar el historial del usuario.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Historial del usuario”, seguidamente el sistema muestra un listado con el historial que ha tenido el usuario en el sistema.	
Resultado esperado: El historial de usuario es mostrado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria.	

Tabla 44: Caso de prueba de aceptación HU8_P1

Caso de prueba de aceptación	
Código: HU8_P1	Historia de usuario: Crear contenido
Nombre: Crear contenido	
Descripción: Prueba para la funcionalidad que permite crear un contenido al usuario solicitante.	
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador o especialista.	



Entrada/Pasos de ejecución: El usuario, en el bloque de navegación, selecciona el link “Lista de solicitudes”, seguidamente el sistema muestra un listado con las solicitudes que hay registradas. Se busca el usuario o la solicitud a la cual se le va a crear el contenido pulsando el link “Crear contenido”, mostrando el sistema una ventana con algunos campos a llenar y otros con los datos de la solicitud del usuario, a medida que el especialista va creando el paquete de contenido pulsando el botón de “Enviar” el sistema muestra un mensaje notificando la cantidad de ejercicios que le falta por insertar al módulo.

Resultado esperado: Contenido creado satisfactoriamente.

Evaluación de la prueba: Satisfactoria.

Anexo 4:

Entrevista realizada al especialista y cliente.

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

1. ¿En qué consiste el producto que desea desarrollar?
2. ¿A qué tipo de usuarios está destinado este producto?
3. ¿Tiene algún producto creado actualmente?
4. ¿Qué servicios brindará esta aplicación?
5. ¿Cómo los administradores de la aplicación gestionaran la información de ejercicios requerida?
6. ¿Bajo qué criterio de búsqueda será mostrada y ordenada la información de ejercicios físicos?
7. Algún criterio en especial para lograr un producto personalizado y generalizable para cualquier tipo de usuario.