

Universidad de las Ciencias Informáticas

Facultad 1



Título: Implementación de un sistema para la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas: UCILITERARIA.

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Autor: Kerlins Martínez Borrego

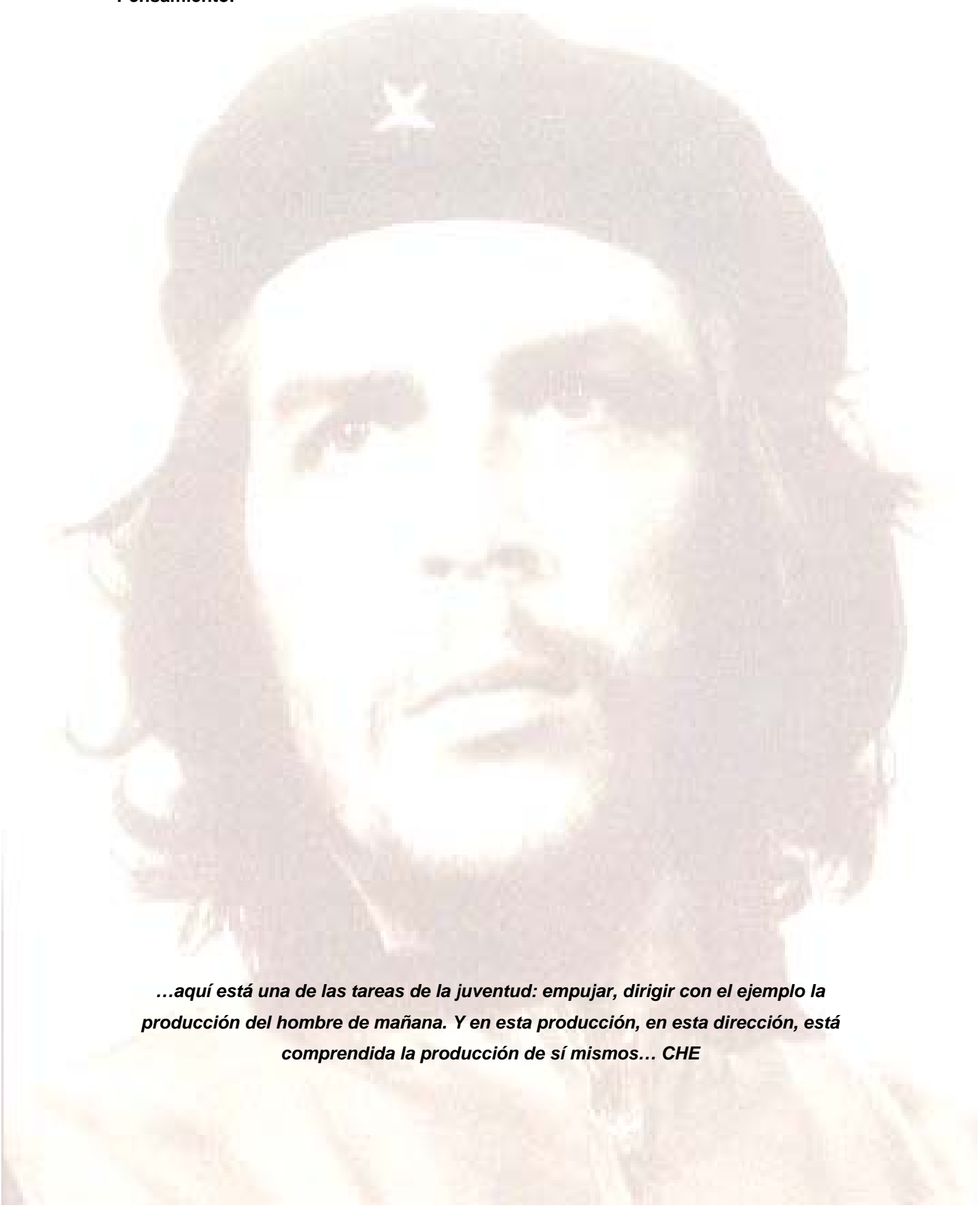
Tutores: Lic. Joel Arencibia Ramírez

Ing. Yoan Sánchez Mesa

Ciudad de La Habana, Cuba.
Junio, 2010

PENSAMIENTO

Pensamiento.



...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos... CHE

DECLARACIÓN DE AUTORÍA

Declaración de autoría.

Declaro que soy el único autor del trabajo titulado:

Implementación de un sistema para la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas: UCILITERARIA.

Y le otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2010.

Autor: Kerlins Martínez Borrego

Tutor: Ing. Yoan Sánchez Mesa

Tutor: Lic. Joel Arencibia Ramírez

AGRADECIMIENTOS

Agradecimientos.

Agradezco principalmente al Comandante y a la Revolución por brindarme la posibilidad de estudiar, a papi y a mami por ser quien soy, y a mis hermanitas lindas por todo el cariño del mundo.

Agradezco además:

-A mamá Margó y a papá Blas.

-A mamá Fela y a papá Félix.

-A mis tías Nery, Ana, Mada y Martha.

-A mi Putyn por todo el amor.

-Al Truji por sus consejos.

-A mis tutores, en especial al Joe.

-A los profesores que me han educado.

-A Rosa e Irene por su colaboración.

-A todos mis amigos, a esos que siempre estuvieron ahí, en especial a Vismark y Brian.

-Por último a mis compañeros de aula y a todos aquellos que de una forma u otra me han apoyado.

DEDICATORIA

Dedicatoria.

Dedico este trabajo a Fidel Castro Ruz, a todos los que han creído en mí y muy en especial a mis padres, Noel y Edairis.

RESUMEN

Resumen.

Uno de los objetivos fundamentales del área de Extensión Universitaria es la gestión de eventos y servicios literarios que se desarrollan en la Universidad de las Ciencias Informáticas (UCI). Estos eventos tienen como núcleo a la Biblioteca, la Librería y el Centro Cultural.

La idea de un sistema para la gestión de eventos y servicios literarios surge a partir de una necesidad de la comunidad universitaria. Dicha necesidad está sustentada en que no existía un espacio capaz de unir a la Librería y el Centro Cultural, en materia de contenido literario. Otro de los problemas era la lejanía entre la residencia y los centros mencionados, lo cual provocaba que no llegara a todos la información deseada, y que el personal laboral de estas instituciones gestionara de forma lenta y trabajosa todo este flujo de información. El objetivo del presente trabajo es desarrollar un sistema que gestione el proceso artístico literario, desde noticias y ofertas de libros, hasta festivales culturales y eventos literarios. En el proceso de desarrollo se utilizó la metodología XP, PHP como lenguaje de programación y el CMS Drupal para minimizar el tiempo de entrega al cliente. Al sistema desarrollado se le realizaron pruebas que validan su calidad y se encuentra listo para su despliegue.

ÍNDICE DE CONTENIDO

Índice.

PENSAMIENTO.	I
DECLARACIÓN DE AUTORÍA.	I
AGRADECIMIENTOS.	II
DEDICATORIA.	III
RESUMEN.	IV
INTRODUCCIÓN.	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.	6
1.1 Introducción.	6
1.2 Conceptos fundamentales.	6
1.3 Arquitectura de los sistemas de gestión.	6
1.4 Sistema para la gestión de eventos y servicios literarios, necesidad o imposición.	7
1.5 Estado del arte.	8
1.5.1 Análisis de sistemas existentes tanto nacionales como internacionales.	8
1.5.2 Análisis general y deficiencias de los sistemas estudiados.	10
1.6 Tendencias y tecnologías actuales.	10
1.6.1 Metodologías de desarrollo.	10
1.6.2 Sistema gestor de base de datos.	14
1.6.3 Servidor Web.	16
1.6.5 Zend Studio, un IDE significativo.	19
1.6.6 Visual Paradigm como Herramienta CASE.	19
1.6.7 ¿Por qué los CMS?	20
1.7 Conclusiones del capítulo.	24
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO.	25
2.1 Introducción.	25

2.2 Fase de Exploración y Planificación.-----	25
2.2.1 Historias de usuario.-----	25
2.2.2 Personas relacionadas con el sistema.-----	34
2.2.3 Requisitos No Funcionales del Sistema.-----	35
2.2.3.1 Requisitos de apariencia o interfaz externa.-----	36
2.2.3.2 Requisitos de usabilidad.-----	36
2.2.3.3 Requisitos de rendimiento.-----	36
2.2.3.4 Requisitos de portabilidad.-----	36
2.2.3.5 Requisitos de seguridad.-----	36
2.2.3.6 Requisitos de software.-----	37
2.2.3.7 Requisitos de hardware.-----	37
2.2.4 Estimación de esfuerzo por Historia de usuario.-----	37
2.2.5 Plan de iteraciones.-----	38
2.2.6 Plan de duración de iteraciones.-----	39
2.2.7 Tareas de ingeniería.-----	40
2.2.8 Plan de entregas.-----	40
2.3 Fase de Diseño.-----	41
2.3.1 Metáfora para el sistema.-----	41
2.3.2 Diseño de la base de datos.-----	42
2.3.3 Diagrama de despliegue.-----	44
2.4 Beneficios del sistema propuesto.-----	45
2.5 Conclusiones del Capítulo.-----	45
 CAPÍTULO 3. DESARROLLO Y VALIDACIÓN DEL SISTEMA PROPUESTO.-----	 47
3.1 Introducción.-----	47
3.2 Estudio detallado de la técnica de programación, plataforma y librerías usadas para el desarrollo de UCILITERARIA.-----	47
3.2.1 Estudio de los Módulos de Drupal.-----	47
3.2.2 Técnicas de programación y creación de módulos con Drupal.-----	49
3.2.3 Estándares de codificación.-----	52
3.2.4 Análisis de posibles componentes, módulos e implementaciones ya existentes que pueden ser reutilizados.-----	53
3.2.5 Descripción del módulo Vlibrería y del algoritmo más complejo en su desarrollo.-----	56
3.3 Pruebas, métodos de pruebas.-----	58
3.3.1 Pruebas unitarias.-----	58

3.3.2 Pruebas de aceptación.-----	58
3.4 Conclusiones del capítulo.-----	59
CONCLUSIONES.-----	60
RECOMENDACIONES.-----	61
BIBLIOGRAFIA REFERENCIADA.-----	62
BIBLIOGRAFIA.-----	64
GLOSARIO.-----	67

INTRODUCCIÓN

Introducción.

La Literatura es una manifestación artística que llega a toda la sociedad. De una forma u otra, estamos en constante intercambio con dicha manifestación. El arte literario va más allá, va desde la comprensión casi perfecta de un texto a la creación divina del mismo.

En todas las universidades del mundo existen movimientos literarios, impulsados por actitudes que presentan profesores, trabajadores y estudiantes. Se trata no solo de universidades donde se estudian carreras de perfil humanístico; pues de alguna manera la ciencia está estrechamente relacionada con la literatura. Este principio parte de que muchos avances científicos están sustentados en obras escritas muchos años antes. Una verdadera forma de imaginar el futuro y dejar volar la imaginación ha sido a través de los años la Ciencia Ficción.

Las universidades cubanas no están ajenas a los movimientos literarios. Como se mencionó anteriormente la ciencia y la literatura caminan a la par y un ejemplo latente de esta afirmación es la Universidad de las Ciencias Informáticas (UCI), un centro de altos estudios encargado de formar ingenieros altamente preparados en esta rama científica.

La Universidad de las Ciencias Informáticas (UCI) cuenta con una comunidad universitaria de más de 10 000 residentes, entre profesores y estudiantes, más una población flotante conformada por trabajadores y profesores externos, que durante el día, llega a ser de unas 2 000 personas. Existe un área llamada Extensión Universitaria, esta surge con la necesidad que tiene todo el personal que concurre en la UCI, de hacer más cómoda su estancia y poder intercambiar ideas, problemas, arte, etc.

Son diversas las manifestaciones artísticas existentes en la UCI, pero el presente estudio se centrará en la literatura y gran parte de los procesos relacionados con la misma; ya que ella ocupa un espacio considerable y carece actualmente de un sistema que la promueva.

Existen tres instituciones, por llamarlas de alguna manera, cuyo trabajo se relaciona con todos los eventos y servicios literarios que se ofrecen en la universidad. Ellas son:

- Centro Cultural.
- Librería.
- Biblioteca.

El Centro Cultural maneja todo lo relacionado con Talleres literarios, Concursos y Festivales de Artistas Aficionados. Estos tres procesos cuentan con una amplia participación del

INTRODUCCIÓN

personal de la UCI y en ellos se maneja un gran cúmulo de información de forma manual. Manejar toda esa información manualmente tiene los inconvenientes siguientes: pérdida de información, mala divulgación de las actividades y por consiguiente, poca participación de la comunidad, atrasos en la entrega de trabajos en los festivales de literatura y con ello, en la revisión de los mismos.

Por otra parte está la Librería. Esta se dedica a la venta de libros en formato impreso y de dar publicidad a los mismos. La comunidad universitaria no cuenta actualmente con un sistema que gestione la publicación de estos ejemplares. Esto provoca que las personas que no tengan tiempo para pasar por la librería, no tengan conocimiento de los libros en venta y sus precios respectivos, lo cual limita la eficiencia del trabajo del personal de la librería y con ello, la incidencia de la misma en la comunidad

Por último y no menos importante que las demás, tenemos a la Biblioteca. Esta cuenta con múltiples servicios; entre ellos: el préstamo de libros en formato impreso, la consultoría de libros de corte docente en formato digital y brinda además un área para la lectura sana y de estudio. Actualmente no existe una relación Biblioteca-Librería que garantice información compartida, es decir, que exista un catálogo en la librería para que junto al de la Biblioteca, nos oriente si un título dado se encuentra en una u otra institución y en caso de estar en los dos lados, el cliente puede escoger si lo compra o solicita un préstamo.

A la hora de revisar los concursos y las obras de los festivales de artistas aficionados, existe cierta demora ya que no todos los autores entregan en tiempo los trabajos, teniendo en cuenta que las vías que actualmente existen para eso son el correo y el contacto personal. Además, existen dificultades a la hora de difundir las actividades y los eventos literarios porque existe una sola vía para esto: la Intranet. En ocasiones el estudiantado se fija en otras temáticas y pasa de largo el vínculo literario. Por otro lado, los residentes en la universidad no cuentan con la existencia de un sitio que englobe parcialmente los contenidos de la Biblioteca y los de la Librería, ya que si fuese así le brindaría la posibilidad de escoger entre comprar un libro o pedirlo prestado para consultarlo. Todo lo planteado anteriormente evidencia la existencia de una Situación Problemática.

El fin del presente trabajo es dar respuesta al siguiente Problema Científico:

¿Cómo perfeccionar la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas?

Para brindar solución a esta Situación Problemática y al Problema Científico, se plantea el siguiente Objetivo General:

INTRODUCCIÓN

Desarrollar un sistema de gestión de eventos y servicios literarios para la Universidad de las Ciencias Informáticas (UCI).

Para dar solución a este objetivo general los Objetivos Específicos trazados son los siguientes:

- Caracterizar la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas (UCI).
- Modelar los procesos asociados a los eventos y servicios literarios en la Universidad de las Ciencias Informáticas (UCI).
- Implementar un sistema para la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas (UCI).

El Objeto de Estudio de este trabajo son los procedimientos de gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas (UCI).

La implementación de este sistema tendrá como Campo de Acción los procesos relacionados con la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas (UCI).

Marco Conceptual de la Investigación.

Tareas:

1. Entrevista con los responsables de todos los eventos y servicios literarios en la UCI.
2. Diseño teórico de la investigación.
3. Análisis de sistemas con características similares dentro del territorio nacional y en el resto del mundo.
4. Modelado del proceso de negocio.
5. Modelado de la solución.
6. Implementación del sistema de gestión de eventos y servicios literarios para la UCI.
7. Ejecución de las pruebas al sistema desarrollado.

Hipótesis.

INTRODUCCIÓN

Si se logra implantar el sistema UCILITERARIA entonces se perfeccionará la gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas.

Variables: Las variables son características y propiedades cuantitativas o cualitativas de un fenómeno que toma distintos valores respecto a la unidad de estudio.

Variable Independiente: Sistema UCILITERARIA.

Variable Dependiente: Perfeccionar la gestión de eventos y servicios literarios Universidad de las Ciencias Informáticas.

Modelo metodológico.

Estrategia de investigación:

Descriptiva: esta permite descubrir la esencia del fenómeno, así como comprender el valor científico de los resultados obtenidos a partir de la profundidad teórica del planteamiento investigativo.

Métodos de Investigación Científica Empleados:

Métodos Teóricos:

Histórico Lógico: Permite estudiar de forma analítica la trayectoria histórica de los fenómenos, su evolución y desarrollo.

Analítico Sintético: Permite la división del fenómeno en sus múltiples relaciones y componentes, lo que facilita su estudio. Establece la unión entre las partes previamente analizadas, posibilita el descubrimiento de sus características generales y las relaciones entre ellas.

Es necesario partir de un análisis de toda la información que fue recopilada mediante otros métodos de investigación como por ejemplo la entrevista; para luego centrarla y sintetizarla para elaborar el procedimiento adecuado.

Métodos Empíricos:

Entrevista: Es una conversación planificada y dirigida hacia objetivos específicos entre el investigador y el entrevistado para obtener información.

INTRODUCCIÓN

Observación: La observación científica es la percepción planificada dirigida a un fin y relativamente prolongada, de un hecho o fenómeno. Es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado.

El contenido del presente trabajo se estructura en tres capítulos:

Capítulo 1: En este capítulo se explican los conceptos fundamentales que ayudan a comprender mejor el trabajo. Se hace un estudio de los sistemas similares a nivel nacional e internacional, así como de las características de las principales herramientas, tecnologías y metodología a utilizar en el desarrollo del sistema.

Capítulo 2: En este capítulo se realiza la exploración, planificación y diseño del sistema propuesto, y se muestran los componentes necesarios para su futura implementación.

Capítulo 3: En este capítulo se describen los principales algoritmos usados en la implementación y se realiza la validación del sistema mediante las pruebas unitarias y de aceptación. Además se relaciona toda la documentación obtenida de las pruebas y se evalúan sus resultados.

Posible resultado:

1. Sistema de gestión de eventos y servicios literarios.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

En el presente capítulo se muestran los principales conceptos relacionados con el problema planteado. Se pueden ver además las arquitecturas de los sistemas de gestión. Luego se muestra el estado del arte en el que se analizan las ventajas y desventajas de los sistemas existentes tanto nacional como internacional. Por último se exponen las valoraciones de las tendencias y tecnologías actuales con el objetivo de lograr un sistema eficiente.

1.2 Conceptos fundamentales.

Cuando está en proceso el desarrollo de un software, tiende a existir problemas en la comunicación y entendimiento entre clientes, desarrolladores y usuarios finales. Esto está dado producto de que de los diversos lados, se manejan términos y conceptos que pueden ser desconocidos para el personal externo; de ahí lo importante que es definir y argumentar los que pueden ser los principales conceptos de un trabajo investigativo.

En cualquier empresa o institución donde se necesite gestionar información de cualquier tipo, se impone la necesidad de elaborar y poner en ejecución un sistema de gestión de eventos y servicios. Este no es más que una herramienta para el control, planeamiento y administración de los servicios prestados a usuarios, integrando informaciones de las diferentes áreas de la institución en que se encuentre: académica, técnica, de investigación y administrativa.

Cuando estamos en presencia de un sistema en el que se necesite hacer búsquedas de contenido o materiales por parte de los usuarios, es obligatoria la confección de una herramienta en la que se introduzcan criterios de búsquedas o mejor dicho, un catálogo general de búsqueda. El mismo está concebido para tener acceso a toda la información en materia de contenido de una base de datos. Esta herramienta, como bien se decía anteriormente, funciona con criterios de búsquedas que pueden ser por contenido, autores, materia o definiciones.

1.3 Arquitectura de los sistemas de gestión.

En los sistemas de gestión la arquitectura usada en la mayoría de los casos es la arquitectura en 3 capas. La arquitectura 3 capas o programación 3 capas, consiste literalmente en separar un proyecto en *Capa de presentación*, *Capa de negocio* y *Capa de*

CAPÍTULO 1

datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles. Estas capas o niveles son [1]:

1. **Capa de Presentación:** Esta es la parte que ve el usuario, las pantallas que se le muestra para que el interaccione con el programa (también se le conoce como “capa de usuario”), comunicándole la información y recolectando la información suministrada por el usuario en un mínimo de proceso (realiza validaciones para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios, es la interfaz grafica del programa y debe ser lo más amena posible para una mejor comunicación con el usuario.
2. **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todos los procesos que deben realizarse.
3. **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Este tipo de arquitectura es la usada cuando se desarrolla con Drupal como CMS debido a que el mismo presenta bien distribuido el trabajo en los 3 niveles que propone la arquitectura en 3 capas.

1.4 Sistema para la gestión de eventos y servicios literarios, necesidad o imposición.

Cabe preguntarse si el hecho de que no exista un sistema que gestione todos los eventos y servicios literarios en la UCI, es o no una necesidad clara para su construcción. Después de varios meses de investigación y entrevistas con los que pudieran ser los principales clientes, se llegó a la conclusión de que significaría un gran avance en pos del desarrollo y la informatización ya que el abundante trabajo que antes se hacía manual, se puede simplificar a varios clics. El desarrollo de las TIC en el mundo y por supuesto en Cuba, está tomando gran relevancia, tanto así que lo que ayer fue la última versión hoy puede ser la versión pasada. Para toda institución es de gran significado que exista una transformación tecnológica en pos de informatizarse o simplemente que exista un cambio de tecnología en caso de existir un sistema, herramienta o programa obsoleto.

1.5 Estado del arte.

1.5.1 Análisis de sistemas existentes tanto nacionales como internacionales.

A continuación se muestran las características de algunos sistemas relacionados con servicios literarios, existentes tanto a nivel internacional como nacional.

Sistemas internacionales.

Ababolia es un portal Español de contenidos literarios que se puede visitar exclusivamente a través de Internet desde marzo del 2003. Desde entonces, más de 100 escritores colaboradores y más de 5000 miembros registrados en sus foros, contribuyen en la actualización constante del mismo. Es un sistema (Sitio Web) que cuenta con cinco módulos fundamentales: Servicios, Lecturas, Actualidad, Ababolízate y Recomendados. El primero de estos módulos presta servicios literarios diversos, entre ellos se destacan cursos, talleres literarios, accesoria y corrección, publicaciones de libros y tu espacio en la red. El área de lecturas cuenta con cuatro temáticas, poemas, relatos, pensamientos y opiniones, las mismas recogen una amplia gama de materia literaria para el deleite y debate de los usuarios. Actualidad, es un apartado que contiene un gran peso en este sistema ya que muestra un compendio de noticias del mundo literario, habla de los libros y sus escritores, y le ofrece a los escritores aficionados la oportunidad de participar en concursos. Algo original y sugerente resulta Ababolízate, un módulo que tiene a su cargo tres temáticas fundamentales, escribir en Ababolia, contacto y foros, estas le dan la oportunidad al usuario de compenetrarse con el sitio, ya sea escribiendo en el, contactando con el equipo desarrollador del sistema y participando de forma interactiva en los diferentes foros. El último de los módulos que presenta Ababolia es Recomendaciones, este ofrece un listado de vínculos a otros sistemas con características similares.

Premura es un sistema que gestiona servicios editoriales y funciona como una agencia literaria en la cual se pueden tener acceso a recursos, escritores, revistas, agencias, publicaciones, empresas y biografías. Esta gran empresa goza de gran prestigio internacionalmente y ello lo demuestra los miles de usuarios en casi todos los países de habla hispana. Premura, como su significado lo manifiesta, pretende llegar con rapidez y prontitud, a todos los usuarios. Posee una interfaz agradable acorde con las necesidades de quien lo visita. Quizás para muchos asiduos de este portal Web el atractivo principal es el Escritores-Club, un vínculo que guía a los escritores en muchos aspectos que son esenciales para futuras publicaciones.

CAPÍTULO 1

Doobooks es un portal que aborda todo un universo de temas literarios, una amplia gama de servicios y miles de usuarios en internet. Este sitio brinda la posibilidad de estar al día con las últimas noticias del mundo de los libros y de la literatura en todos los países de habla hispana. Otra de sus funciones vitales es permitir la compra de libros en diversas librerías de la red y brinda la posibilidad de que los usuarios puedan recibir estos ejemplares en sus casas. Para aquellos que se inicien en la literatura y desean hacerlo en serio, Doobooks ofrece cursos y talleres de escritura, por ejemplo: novela, relato, poesía, corrección literaria, creación de personajes, guión de cine, etc.

Sistemas nacionales.

Cubaliteraria salió a la luz en el año 2005. Desde sus inicios el soporte de este portal está a cargo de CITMATEL, la cual tiene entre sus funciones fundamentales proponer y ejecutar proyectos de investigación, así como la introducción de herramientas y métodos avanzados de diseño y programación necesarios para el desarrollo de aplicaciones informáticas. Este portal de la literatura cubana cuenta con seis módulos fundamentales, *portada, de la Cuba Literaria, autores, libros, revistas y periódicos, y editoriales*. Además este sitio presenta quince secciones temáticas, donde la actualidad y la primicia literaria van de la mano. El primero de estos módulos se encarga de mostrar la interfaz principal donde se puede observar un compendio de noticias actualizadas y eventos nacionales relacionados con la literatura y el libro. *De la Cuba Literaria* muestra un ambiente agradable entre los autores y sus libros o sus quehaceres en la literatura. En el apartado *autores* muestra a todos aquellos que han contribuido con el desarrollo de la literatura en Cuba y la han elevado al más alto nivel. El área de los *libros* es quizás la que más usuarios atraiga debido a que muestra todos los libros o la gran mayoría de los publicados por editoriales cubanas, *revistas y periódicos* al igual que libros muestra todos los ejemplares que se difunden entre la sociedad hoy en día. Por último, *editoriales* muestra todas las editoriales existentes en nuestro país y sus aspectos relevantes. Sin lugar a es un portal bien estructurado que gana usuarios cada día en la red de redes.

Sistemas en la UCI.

El portal Web de la **Biblioteca** de la Universidad de las Ciencias informáticas, sirve de apoyo a la institución que la originó. Este portal cuenta con dos áreas fundamentales, la informativa y la de servicios. En la primera de esta se ofrecen *biografías, trucos del bibliotecario, revistas, boletines, libros digitalizados y otros sitios*; la segunda área temática ofrece un *catálogo de búsqueda dentro del sitio, bases de datos y un amplio conjunto de tutoriales* con diferentes fines. Biblioteca.uci.cu es más que un simple sitio de apoyo, es un

CAPÍTULO 1

espacio obligado para la consulta y el estudio, un espacio para todo aquel que desee superarse y crecer.

1.5.2 Análisis general y deficiencias de los sistemas estudiados.

Después de haber estudiado los sistemas anteriormente expuestos, se puede ver como todos tienen en común un aspecto fundamental, y es que cada uno de ellos apuesta todo por la literatura y por mejorar los servicios que prestan. Sin quitarle méritos a estos portales web, se puede decir que no son del todo completos ya que ninguno de ellos brinda la posibilidad de compartir libros, siendo esta una necesidad actual de todos los adictos a la lectura en el mundo; tampoco brindan la posibilidad de tener información compartida entre una biblioteca y una librería donde el usuario pueda escoger si compra el libro o lo pide prestado, y por último, ninguno de ellos gestiona otros eventos que no sean los que vienen en su estructura originalmente. Después de haber estudiado a fondo el sistema de la Biblioteca se puede notar la existencia de una buena organización y un trabajo bien estructurado, mas si nos fijamos bien, se nota que carece de un vínculo con la librería y con los eventos literarios, en este caso los concursos literarios y los Festivales de Artistas Aficionados en la manifestación de literatura.

En el presente trabajo de diploma se propone un sistema donde se gestionen todos los eventos referentes a los Festivales de Artistas Aficionados en la manifestación de literatura, se gestionen además todos los servicios literarios que ofrece la universidad y también se propone una catálogo global, por llamarlo así, que recoja toda la información que comparte la librería con la biblioteca, en este caso información sobre los libros.

1.6 Tendencias y tecnologías actuales.

1.6.1 Metodologías de desarrollo.

Una metodología de desarrollo de software se refiere a un *framework* que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. [2]

Proceso Unificado de Desarrollo de Software (RUP).

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software [3]:

Inicio, el objetivo en esta etapa es determinar la visión del proyecto.

CAPÍTULO 1

Elaboración, en esta etapa el objetivo es determinar la arquitectura óptima.

Construcción, en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

Transmisión, el objetivo es llegar a obtener la liberación del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. [3]

Disciplina de desarrollo. [3]

Ingeniería de negocios: Se entienden las necesidades del negocio.

Requerimientos: Se trasladan las necesidades del negocio a un sistema automatizado.

Análisis y diseño: Se trasladan los requerimientos dentro de la arquitectura de software.

Implementación: Se crea un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

Pruebas: Se asegura que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. [3]

Extreme Programming (XP).

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final; pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP. [4]

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que, adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.

CAPÍTULO 1

- **Re fabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué es lo que propone XP? [4]

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Ventajas [4]:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Scrum.

Scrum es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. [4]

Aunque Scrum está enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums. [4]

Ideal para proyectos con un rápido cambio de requerimientos. Es compatible con las metodologías ágiles y en concreto, con la programación extrema. [4]

Características fundamentales de Scrum.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará

CAPÍTULO 1

durante un proyecto. Los roles principales en Scrum son el *ScrumMaster*, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *ProductOwner*, que representa a los involucrados (clientes externos o internos), y el *equipo* que incluye a los desarrolladores. [4]

Artefactos SCRUM.

Sprint. [4]

- Es la base del desarrollo Scrum.
- Su máxima duración es de 30 días.
- Se realizan tareas anteriormente establecidas y no se puede modificar el trabajo acordado en el *backlog*.
- Sólo el *ScrumMaster* puede abortar un sprint si lo considera no viable por alguna de las siguientes razones [4]:
 - Las circunstancias del negocio han cambiado.
 - La tecnología acordada no funciona.
 - El equipo ha tenido interferencias.

Sprint Backlog. [4]

- Especifica la serie de tareas que se van a desarrollar según los requisitos señalados.
- Estas tareas tienen una duración de entre 4 y 16 hrs. de trabajo.
- Las de mayor duración se intentan descomponerlas en sub-tareas dentro de ese rango de tiempo.
- Al final del sprint se busca un incremento en la funcionalidad.

Metodología a utilizar.

Se puede concluir que la metodología de desarrollo de software más idónea para la elaboración de este trabajo de diploma es XP, ya que esta ha sido diseñada especialmente para proyectos de corto plazo y que puedan cambiar de requisitos constantemente. Otra ventaja que nos ofrece esta metodología, es que los clientes forman parte del grupo de desarrollo que a su vez se caracteriza por ser sencillo, compacto y eficaz. Por otro lado se puede decir que XP refuerza con sus prácticas todo el proceso de desarrollo, logrando como resultado que este sea más completo y organizado.

1.6.2 Sistema gestor de base de datos.

MySQL.

¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño *multihilo* le permite soportar una gran carga de forma muy eficiente. MySQL fue creado por la empresa sueca MySQL AB, que mantiene el derecho de autor del código fuente del servidor SQL, así como también de la marca. [5]

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL. [5]

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. [5]

Características de MySQL v5.1.37.

Las principales características de este gestor de bases de datos son las siguientes [5]:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación *multihilo*.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de *API's* en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas de MySQL.

Es evidente que la gran mayoría de las personas usan este gestor en Internet, por lo que encontrar opiniones favorables no ha resultado en absoluto complicado [5]:

CAPÍTULO 1

- Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones (Freshmeat.net).

Desventajas de MySQL.

Debido a la gran aceptación con que goza este sistema gestor de bases de datos en Internet, gran parte de los inconvenientes que se exponen a continuación, han sido extraídos de comparativas con otras bases de datos [5]:

- Carece de soporte para transacciones, *rollback's* y *subconsultas*.
- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

¿Por qué MySQL?

MySQL v5.1.37 tiene una ventaja significativa y esta consiste en que el conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones, es además el más usado en sistemas realizados con Drupal debido a su gran compatibilidad y es *OpenSource* (Código libre), por lo que se concluye que es la más idóneo a utilizar para la realización del sistema propuesto.

1.6.3 Servidor web.

Un servidor web es un programa que sirve datos en forma de páginas web, hipertextos o páginas HTML (*HyperText Markup Language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio de un protocolo, concretamente del protocolo HTTP. Con esto, un servidor web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un navegador web. [6]

Apache v2.2.12 como servidor web.

Características.

Apache es un servidor web flexible, rápido y eficiente y continuamente actualizado. Entre sus características se destacan [7]:

- Es multi-plataforma.
- Es un servidor de web conforme al protocolo HTTP/1.1.
- Es modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Está basado en hebras en la versión 2.0.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.

Principales ventajas.

- Código abierto.
- Multi-plataforma.
- Extensible.
- Popular (fácil conseguir ayuda/suporte).

¿Por qué Apache?

Existen diversos tipos de servidores web, mas no se puede negar la superioridad de Apache sobre los demás, siendo este el más utilizado en el mundo. En el presente trabajo de diploma se decide seleccionar Apache v2.2.12 como servidor además de sus ventajas por

ser este el que utiliza la Intranet de la Universidad de las Ciencias Informáticas y el presente sistema en un futuro debe integrarse a los servidores de la universidad y con ello a la Intranet.

1.6.4 Programación web, lenguaje a utilizar.

La programación web, parte de las siglas WWW, que significan *World Wide Web* o telaraña mundial. Para realizar una página con la programación web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (*Uniform Resource Locators*), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita. [8]

El siguiente concepto dentro de la programación web, es el protocolo encargado de llevar la información que contiene una página web por toda la red de Internet, como es el HTTP (*Hypertext Transfer Protocol*). [8]

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página web, este lenguaje es el HTML (*Hypertext Markup Language*). En la programación web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página web. [8]

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor; pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica. [8]

Características de PHP v5.3.0.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas [9]:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.

CAPÍTULO 1

- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Un sitio con páginas web dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

Ventajas de PHP v5.3.0.

Las principales ventajas de PHP v5.3.0 son [9]:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados *ext's* o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

PHP como lenguaje a utilizar.

Es PHP el más idóneo para el desarrollo del presente trabajo de diploma ya que es un lenguaje multiplataforma, se destaca por su conectividad con MySQL y además es libre

CAPÍTULO 1

siendo de fácil acceso y con documentación al alcance de todos para su estudio en profundidad.

1.6.5 Zend Studio, un IDE significativo.

Zend Studio es un entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Características.

No requiere la instalación previa de PHP ni del entorno de ejecución de Java.

- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- Detección de errores de sintaxis en tiempo real.
- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversion (a elección del desarrollador).
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio como IDE.

Se decide utilizar Zend Studio v5.5 como IDE por todas las oportunidades que ofrece y por la perfecta compatibilidad con PHP como lenguaje de programación.

1.6.6 Visual Paradigm como Herramienta CASE.

¿Qué importancia tienen las Herramientas Case?

Hoy en día, muchas empresas se han extendido a la adquisición de herramientas CASE, con el fin de automatizar los aspectos clave de todo el proceso de desarrollo de un sistema, desde el principio hasta el final y así incrementar su posición en el mercado competitivo. Algunas de estas herramientas tienen un valor económico muy alto y requieren costos de

CAPÍTULO 1

entrenamiento de personal muy altos; además se enfrentan la falta de adaptación de la herramienta, a la arquitectura de la información en la que está compuesta y a las metodologías de desarrollo utilizadas por la organización. Por otra parte, algunas herramientas CASE no ofrecen o evalúan soluciones potenciales para los problemas relacionados con sistemas, o simplemente no llevan a cabo ningún análisis de los requerimientos de la aplicación. [10]

La estructura de las herramientas CASE se basa en la siguiente terminología [10]:

CASE de Alto Nivel son aquellas herramientas que automatizan o apoyan las fases iniciales del ciclo de vida del desarrollo de sistemas, como la planificación de sistemas, el análisis de sistemas y el diseño de sistemas.

CASE de Bajo Nivel son aquellas herramientas que automatizan o apoyan las fases finales o inferiores del ciclo de vida como el diseño detallado de sistemas, la implantación de sistemas y el soporte de sistemas.

CASE Cruzado de Ciclo de vida en donde se aplica a aquellas herramientas que apoyan actividades que tienen lugar a lo largo de todo el ciclo de vida, se incluyen actividades como la gestión de proyectos y la estimación.

Visto lo anterior se puede resaltar que Visual Paradigm para UML es una Herramienta Case Cruzado de Ciclo de Vida. [10]

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [11]

1.6.7 ¿Por qué los CMS?

Un **sistema de gestión de contenidos** (CMS) es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web.

CAPÍTULO 1

Consiste en una interfaz que controla una o varias bases de datos en las cuales se pueden realizar modificaciones del contenido existente. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido nuevamente, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo el público. [12]

Drupal, una herramienta de gestión de contenido.

Drupal es un sistema de gestión de contenido (CMS) que permite, tanto a individuos como a comunidades de usuarios, la publicación de contenidos en Internet, de manera sencilla, y sin tener conocimientos de lenguaje de marcas. Desarrollado dentro de la filosofía del Software Libre y de código fuente abierto bajo licencia GPL, Drupal es un sistema dinámico, puesto que el contenido de las páginas y otras configuraciones se pueden editar online mediante las herramientas que él mismo facilita. [12]

Desarrollado en PHP y mantenido por una amplia comunidad, permite múltiples usuarios y distintos lenguajes de marcas. Destaca la calidad de su código, el respeto por los estándares web, un énfasis especial en la usabilidad y consistencia de todo el sistema. Su flexibilidad y adaptabilidad, así como su estructura modular y su gran cantidad de módulos adicionales disponibles, hacen que sea adecuado para el desarrollo de sitios web con funcionalidades distintas. [12]

Características generales de Drupal v6.16. [13]

- **Ayuda on-line:** Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.
- **Hooks:** La documentación de Drupal tiene una sección dedicada a los *Hooks*. Los *Hooks* son el mecanismo que provee Drupal para interactuar con los distintos procesos que se ejecutan en un sitio web. Conocer su funcionamiento es fundamental para cualquier programador de módulos; así como también para aquellos diseñadores o *Themers* que deseen modificar aspectos al parecer imposibles de lograr.
- **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. Al contrario que otros sistemas de 'blogs' o de

CAPÍTULO 1

gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

- **Módulos:** La comunidad de Drupal ha contribuido con infinidad de módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante *jabber*, mensajes privados y *bookmarks*, entre otros.
- **Personalización:** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser personalizados de acuerdo con las preferencias definidas por el usuario.
- **URLs amigables:** Drupal usa el *mod_rewrite* de Apache para crear *URLs* que son manejables por los usuarios y los motores de búsqueda.
- **Gestión y 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.
- **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 deshacerlos, 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 a 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*, visualice el contenido publicado en la web desde el escritorio. Igualmente Drupal

CAPÍTULO 1

incluye un potente *Agregador* de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

- **La API de Blogger:** Permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que pueden ser 'herramientas web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.
- **Plataforma Independiente 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 ettext.
- **Administración y Análisis 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.
- **Características de comunidad:** Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión de un tema publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.
- **Encuestas:** Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.
- **Foros de discusión:** Drupal incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.

CAPÍTULO 1

- **Libro Colaborativo:** Esta característica es única de Drupal y permite crear un proyecto o “libro” cuyo contenido será la contribución de los mismos usuarios. El contenido se organiza en páginas cómodamente navegables.
- **Rendimiento y escalabilidad:** Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.
- **Sistema de Cache:** El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

¿Por qué este CMS?

Se decide utilizar este CMS en su versión 6.16 por sus magníficas características donde se destacan las de ser libre y multiplataforma, además contribuye a esta decisión el que en un futuro el sistema sea integrado a la intranet de la Universidad de las Ciencias Informáticas.

1.7 Conclusiones del capítulo.

En este capítulo se puede ver detalladamente el estado del arte y otros temas relacionados con el caso que se plantea, así como, la fundamentación de las decisiones tomadas acerca de las herramientas, tecnologías, lenguajes y metodologías a utilizar. Se decidió emplear como lenguaje de programación PHP v5.3.0 por todas las ventajas antes vistas y por ser además el lenguaje que utiliza Drupal; como sistema gestor de base de datos MySQL v5.1.37 por su manejabilidad y ventajas; como CMS a Drupal v6.16 por sus sobradas ventajas y decisión de los clientes; como servidor web Apache v2.2.12 principalmente por ser el más utilizado en la universidad; Visual Paradigm v6.4 como herramienta case por ser de libre distribución y de fácil manejo; Zend Studio v5.5 como IDE y la metodología seleccionada fue XP para lograr un desarrollo dinámico junto al cliente y una rápida culminación del producto.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA, EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO.

2.1 Introducción.

En el presente capítulo se aborda inicialmente las principales características del sistema propuesto, las fases de exploración, planificación y diseño pertenecientes a la metodología XP (*Extreme Programming*) utilizada para el desarrollo del sistema que se propone. Además se exponen los artefactos generados durante el transcurso de dichas fases.

2.2 Fase de Exploración y Planificación.

La metodología de desarrollo XP comienza con la fase de exploración y planificación. Durante esta se realiza el proceso de identificación de las historias de usuario, así como la familiarización de los equipos de trabajo con las tecnologías y herramientas seleccionadas para la construcción del proyecto y además se realiza una estimación del esfuerzo que costará implementar cada historia de usuario.

2.2.1 Historias de usuario.

Las historias de usuario (HU) son la forma en que se especifican en *XP* los requisitos del sistema. Estas se escriben desde la perspectiva del cliente; aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo. Durante la fase de exploración se identificaron 18 historias de usuario, las cuales se relacionan a continuación:

1. Lanzar convocatoria a concursos.
2. Enviar concursos.
3. Publicar resultados de concursos.
4. Lanzar convocatoria a FAAL.
5. Realizar inscripciones a FAAL.
6. Enviar trabajos de FAAL.
7. Publicar resultados de FAAL.
8. Gestionar temas en el foro.

CAPÍTULO 2

9. Gestionar comentarios en el foro.
10. Presentar libros.
11. Gestionar libros.
12. Gestionar noticias.
13. Gestionar administradores.
14. Gestionar técnicos.
15. Gestionar seguridad.
16. Gestionar restricciones.
17. Autenticar.
18. Cerrar sesión.

Las historias de usuario representan las funcionalidades que el cliente desea que estén presentes en el sistema; por lo tanto, todo el trabajo futuro debe girar en torno a satisfacer estas expectativas. Las historias de usuario son la primera y fundamental entrada para el proceso de desarrollo de software basado en la metodología *XP*. La descripción general de estas historias de usuario se muestran a continuación y el prototipo de interfaz de usuario perteneciente a cada una de ellas se encuentra en el Anexo I.

Tabla 1 HU Lanzar convocatoria a concursos

Historia de usuario	
Número: 1	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Lanzar convocatoria a concursos.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de lanzar convocatorias a los concursos literarios en el sistema, esta incluyen además: día de admisión de trabajos, día de exposición en talleres y día de entrega de premios.	

CAPÍTULO 2

Observaciones: Ninguna.

Tabla 2 HU Enviar concursos

Historia de usuario	
Número: 2	Usuario: Usuario Autenticado (Ver descripción en la Tabla 19).
Nombre historia: Enviar concursos.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Usuario Autenticado de enviar los trabajos al sistema, respondiendo a una convocatoria lanzada, cumpliendo con las condiciones de entrega.	
Observaciones: El Usuario Autenticado debe enviar solamente archivos del tipo estipulado de lo contrario no se admiten los trabajos y tampoco debe exceder de cinco trabajos.	

Tabla 3 HU Publicar resultados de concursos

Historia de usuario	
Número: 3	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Publicar resultados de concursos.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de publicar los resultados de los concursos en el sistema una vez evaluados los trabajos.	
Observaciones: Sólo se le publican resultados a un concurso que no se le hayan publicado antes.	

CAPÍTULO 2

Tabla 4 HU Lanzar convocatoria a FAAL

Historia de usuario	
Número: 4	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Lanzar convocatoria a FAAL.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Bajo.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de lanzar convocatorias a los Festivales de Artistas Aficionados de Literatura en el sistema.	
Observaciones: Sólo se le realizan convocatorias a un festival una vez al año.	

Tabla 5 HU Realizar inscripciones a FAAL

Historia de usuario	
Número: 5	Usuario: Usuario Autenticado (Ver descripción en la Tabla 19).
Nombre historia: Realizar inscripciones a FAAL.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Usuario Autenticado de inscribirse en los Festivales de Artistas Aficionados de Literatura en el sistema, esto ocurre una vez lanzada la convocatoria.	
Observaciones: Una vez inscrito no puede deshacer la inscripción.	

Tabla 6 HU Enviar trabajos de FAAL

Historia de usuario	
Número: 6	Usuario: Usuario Autenticado (Ver descripción en la Tabla 19).

CAPÍTULO 2

Nombre historia: Enviar trabajos de FAAL.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Medio.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Usuario Autenticado, una vez inscrito en el festival, de enviar los trabajos al sistema, cumpliendo con las condiciones de entrega.	
Observaciones: El Usuario Autenticado debe subir solamente archivos del tipo estipulado y no debe exceder de cinco trabajos.	

Tabla 7 HU Publicar resultados de FAAL

Historia de usuario	
Número: 7	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Publicar resultados de FAAL.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Bajo.
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de publicar los resultados de los Festivales de Artistas Aficionados de Literatura una vez evaluados los trabajos.	
Observaciones: Sólo se publican los resultados si no se ha hecho antes.	

Tabla 8 HU Gestionar temas en el foro

Historia de usuario	
Número: 8	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Gestionar temas en el foro.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Bajo.

CAPÍTULO 2

Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de que realice las acciones de crear y eliminar temas en el foro del sistema, además de editar y mostrar los datos de cada tema existente en el foro.	
Observaciones: Si existe un tema con un nombre igual a la que se desea crear entonces no podrá crearse.	

Tabla 9 HU Gestionar comentarios en el foro

Historia de usuario	
Número: 9	Usuario: Técnico y Usuario Autenticado (Ver descripción en la Tabla 19).
Nombre historia: Gestionar comentarios en el foro.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Bajo.
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se les brinda la posibilidad al Técnico y Usuario Autenticado de que realice las acciones de crear y modificar comentarios en el foro del sistema, además se le brinda la posibilidad al técnico de eliminar los comentarios existentes en el foro.	
Observaciones: Ninguna.	

Tabla 10 HU Presentar libros

Historia de usuario	
Número: 10	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Presentar libros.	
Prioridad en negocio: Media.	Riesgo en desarrollo: Media.
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego.	

CAPÍTULO 2

<p>Descripción: Se le brinda la posibilidad al Técnico de darle publicidad a los libros en el sistema. Estos libros son los nuevos ejemplares que llegan a la librería. La publicidad se basa en mostrar el título del libro, el autor y una breve reseña bibliográfica.</p>
<p>Observaciones: Sólo se le da publicidad a un libro una vez, cuando se registra como un nuevo libro en el sistema.</p>

Tabla 11 HU Gestionar libros

Historia de usuario	
Número: 11	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Gestionar libros.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alto.
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de que realice las acciones de crear y eliminar libros en el sistema, además de editar y mostrar los datos de cada libro existente en el sistema.	
Observaciones: Si existe un libro con un nombre y autor igual a la que se desea crear entonces no podrá crearse.	

Tabla 12 HU Gestionar noticias

Historia de usuario	
Número: 12	Usuario: Técnico (Ver descripción en la Tabla 19).
Nombre historia: Gestionar noticias.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alto.
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego.	
Descripción: Se le brinda la posibilidad al Técnico de que realice las acciones de insertar y eliminar noticias en el sistema, además de editar y mostrar los datos de cada noticia existente en el sistema.	

CAPÍTULO 2

Observaciones: Si existe una noticia con un nombre y autor igual a la que se desea crear entonces no podrá crearse.

Tabla 13 HU Gestionar administradores

Historia de usuario	
Número:13	Usuario: Administrador (Ver descripción en la Tabla 19).
Nombre historia: Gestionar administradores.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se le brinda la posibilidad al Administrador de que realice las acciones de crear y eliminar otros administradores del sistema, además de editar y mostrar los datos de cada administrador presente en el sistema.	
Observaciones: Si existe un administrador con un nombre igual al que se desea crear entonces no podrá crearse. Si el administrador intenta eliminar su propia cuenta el sistema debe impedirlo.	

Tabla 14 HU Gestionar técnicos

Historia de usuario	
Número:14	Usuario: Administrador (Ver descripción en la Tabla 19).
Nombre historia: Gestionar técnicos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se le brinda la posibilidad al Administrador de que realice las acciones de crear y eliminar técnicos del sistema, además de editar y mostrar los datos de cada tipo de turno presente en el sistema.	
Observaciones: Si existe un técnico con un nombre igual al que se desea crear entonces no podrá crearse.	

CAPÍTULO 2

Tabla 15 HU Gestionar seguridad

Historia de usuario	
Número:15	Usuario: Administrador (Ver descripción en la Tabla 19).
Nombre historia: Gestionar seguridad.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se le brinda la posibilidad al Administrador de que realice las acciones de crear, editar, mostrar y eliminar los datos de la configuración de seguridad presente en el sistema.	
Observaciones: Ninguna.	

Tabla 16 HU Gestionar restricciones

Historia de usuario	
Número:16	Usuario: Administrador (Ver descripción en la Tabla 19).
Nombre historia: Gestionar restricciones.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se le brinda la posibilidad al Administrador de que realice las acciones de insertar, editar, mostrar y eliminar datos de las restricciones presente en el sistema.	
Observaciones: Ninguna.	

Tabla 17 HU Autenticar

Historia de usuario	
Número:17	Usuario: Usuario (Ver descripción en la Tabla 19).

CAPÍTULO 2

Nombre historia: Autenticar.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se brinda la posibilidad al usuario a que introduzca sus datos (nombre de usuario y contraseña) con la finalidad de verificarlos y otorgarle los permisos según el rol que cumpla dentro de la aplicación.	
Observaciones: El usuario y la contraseña deben ser válidos. Se le otorgarán los permisos de acuerdo a la configuración de seguridad del sistema.	

Tabla 18 HU Cerrar sesión

Historia de usuario	
Número:18	Usuario: Administrador, Usuario Autenticado (Ver descripción en la Tabla 19).
Nombre historia: Cerrar sesión.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Kerlins Martínez Borrego	
Descripción: Se le brinda la posibilidad al Administrador y al Usuario Autenticado a que cierre la sesión y continúe navegando por el sistema sin privilegios.	
Observaciones: Ninguna.	

2.2.2 Personas relacionadas con el sistema.

Se denomina persona relacionada con el sistema a aquella que interactúa e intercambia con el sistema y obtiene resultados de los procesos desarrollados en la aplicación. Además, se incluye como persona relacionada, a aquella que interactúa con la aplicación sin poder hacer uso de las secciones privilegiadas del sistema. Las personas relacionadas con el sistema se infieren de una correcta lectura de las historias de usuario. A continuación se relacionan las personas vinculadas con el sistema propuesto.

Tabla 19 Personas relacionadas con el sistema.

Personas relacionadas con el sistema	Descripción
Usuario	Es la persona que navega por el sistema sin usar una cuenta específica y por tanto sin privilegios. Tiene solamente la posibilidad de consultar las noticias publicadas.
Usuario Autenticado	Es la persona que ya ha sido autenticada dentro de la aplicación y posee privilegios bajos, puede tratarse de un profesor, un alumno, un trabajador o un asociado a la universidad. Este podrá hacer uso de todas las áreas del sistema excepto las que son de carácter administrativas y de seguridad.
Administrador	Es la persona encargada de llevar a cabo la administración de todo el contenido dentro de la aplicación, gestiona todo el sistema.
Técnico	Es la persona encargada de llevar a cabo la gestión del contenido de su área de trabajo, gestiona el contenido de las clases persistentes en la base de datos.

2.2.3 Requisitos No Funcionales del Sistema.

Los requisitos no funcionales del sistema no se capturan en la metodología XP en forma de historias de usuario, puesto que los clientes generalmente no conocen la terminología técnica que se utiliza para describir este tipo de requisitos. La captura de requisitos no funcionales es por tanto una tarea del equipo de desarrollo completo, que haciendo un intercambio profundo de ideas con el cliente, determinan las mejores soluciones para crear un sistema que, además de implementar correctamente las historias de usuario, cumpla con determinados estándares de calidad y con las características propias del negocio que se pretende automatizar.

2.2.3.1 Requisitos de apariencia o interfaz externa.

La aplicación propuesta será usada por personas que tengan un conocimiento medio de informática; por lo que la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el trabajo con la misma, en especial para aquellos que poseen pocas habilidades en el manejo de aplicaciones web.

2.2.3.2 Requisitos de usabilidad.

A los administradores de la aplicación se les dará un adiestramiento básico en el uso de la aplicación. Estas personas tendrán un nivel de acceso amplio en la aplicación para que puedan dar respuesta a cada incidente ocurrido.

2.2.3.3 Requisitos de rendimiento.

Para un funcionamiento óptimo de la aplicación, se seguirán las diferentes técnicas de elaboración de la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo cliente/servidor, y la velocidad de las consultas de la base de datos. La aplicación propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el cliente requiere la respuesta a su petición.

2.2.3.4 Requisitos de portabilidad.

Las herramientas podrán ser usadas bajo cualquier sistema operativo de Windows NT en adelante o cualquier distribución de *Linux*. El servidor web y el servidor de Base de Datos pueden estar en la misma computadora sin ocasionar problemas de rendimiento.

2.2.3.5 Requisitos de seguridad.

Confiabilidad: La información manejada por el sistema debe estar protegida de acceso no autorizado.

Integridad: La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.

Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

2.2.3.6 Requisitos de software.

En las computadoras de los usuarios solo se requiere un navegador web moderno (Se recomienda Mozilla Firefox 2 o superior). Los usuarios pueden acceder al sistema desde cualquier sistema operativo. En el servidor de base de datos se requiere de Windows NT en adelante o cualquier distribución de Linux y como gestor de base de datos MySQL 5.0 o PostgreSQL 8.2. En el servidor web se requiere Apache 2 o superior, PHP 5.2 con las extensiones *php_mysql*, *php_pgsql*, *php_pdo*, *php_pdo_mysql*, *php_pdo_pgsql* y *php_ldap* activadas.

2.2.3.7 Requisitos de hardware.

En el cliente se requiere una máquina con 128 MB de RAM como mínimo. El servidor web junto con el servidor de base de datos, debe tener 256 MB de RAM y 20 GB de disco duro mínimo y todas las máquinas implicadas en el funcionamiento de la aplicación deben estar conectadas a la red.

2.2.4 Estimación de esfuerzo por Historia de usuario.

Para el desarrollo del sistema propuesto se realizó una estimación del esfuerzo para cada una de las historias de usuario. Este se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario, por ejemplo: las pruebas unitarias, la integración y refactorización del código y la preparación y ejecución de las pruebas de aceptación. Llegándose a los resultados que se muestran en la Tabla 2.

Tabla 10 Estimación de esfuerzo por historia de usuario

Historia de Usuario	Puntos estimados
Lanzar convocatorias a concursos.	0.5
Enviar concursos.	0.5
Publicar resultados de concursos.	0.5
Lanzar convocatoria a FAAL.	0.5
Realizar inscripciones a FAAL.	0.5
Enviar trabajos de FAAL.	0.5

CAPÍTULO 2

Publicar resultados de FAAL.	0.5
Gestionar temas en el foro.	1
Gestionar comentarios en el foro.	1
Presentar libros.	0.5
Gestionar libros.	2
Gestionar noticias.	1
Gestionar administradores.	1
Gestionar técnicos.	1
Gestionar seguridad.	1
Gestionar restricciones.	1
Autenticar.	0.5
Cerrar sesión.	0.5

2.2.5 Plan de iteraciones.

Una vez identificadas las historias de usuario del sistema y estimado el esfuerzo requerido para la implementación de cada una de estas, se procede a realizar el plan de iteraciones del proyecto. Teniendo en cuenta el riesgo para desarrollar cada una de las historias de usuario, el tamaño del equipo de desarrollo, así como otros factores subjetivos, se decidió dividir el proyecto en tres iteraciones, detalladas a continuación.

Iteración 1.

En esta iteración se implementan las historias de usuarios más sencillas, la realización de las mismas proporciona una idea de cómo funcionará la aplicación aunque todavía esté en sus inicios.

Iteración 2.

En esta iteración se realizarán las historias de usuarios más complejas y que inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el sistema se encontrará en un estado plenamente funcional.

Iteración 3.

En esta iteración se realizarán las historias de usuarios que se consideren opcionales y que no inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el sistema se encontrará completamente concluido y listo para su explotación.

Las HU asociados a cada iteración pueden verse en la tabla 21.

2.2.6 Plan de duración de iteraciones.

Como parte del ciclo de vida de un proyecto utilizando XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con que se cuenta. Este plan se encarga de mostrar las historias de usuario que serán abordadas en cada una de las iteraciones, así como la duración estimada de estas últimas y el orden en que se implementarán. La tabla 21 muestra el plan de duración de iteraciones para el desarrollo de este sistema.

Tabla 21 Plan de duración de iteraciones

Iteración	Historia de Usuario	Puntos estimados	Duración de cada iteración
Iteración 1	Lanzar convocatoria a concursos.	0.5	3.5 semanas
	Enviar concursos.	0.5	
	Publicar resultados de concursos.	0.5	
	Lanzar convocatoria a FAAL.	0.5	
	Realizar inscripciones a FAAL.	0.5	
	Enviar trabajos de FAAL.	0.5	
	Publicar resultados de FAAL.	0.5	
Iteración 2	Gestionar temas en el foro.	1	10 semanas
	Gestionar comentarios en el foro.	1	

CAPÍTULO 2

	Presentar libros.	0.5	
	Gestionar libros.	2	
	Gestionar noticias.	1	
	Gestionar administradores.	1	
	Gestionar técnicos.	1	
	Gestionar seguridad.	1	
	Gestionar restricciones.	1	
	Autenticar.	0.5	
Iteración 3	Cerrar sesión.	0.5	0.5 semana

2.2.7 Tareas de ingeniería.

Para dar solución a las HU es necesario realizar un grupo de acciones llamadas tareas de ingeniería. Estas son descritas una vez terminada toda la documentación necesaria para generar el código y permiten una mejor organización del proceso de implementación del módulo. Para revisar la relación de todas las tareas de ingenierías realizadas en el desarrollo de este sistema, ver Anexo II.

2.2.8 Plan de entregas.

A continuación se presenta el plan de entregas ideado para la fase de implementación. Como producto del mismo se harán entregable del sistema al finalizar cada iteración en la fecha aproximada que se indica en la Tabla 22.

Tabla 22 Plan de entregas

Sistema	Hito	Fecha Aproximada
Sistema de gestión de eventos y servicios literarios para la UCI versión 0.1	Fin de la primera iteración.	Segunda semana de marzo de 2010.
Sistema de gestión de eventos y servicios literarios para la UCI versión 0.2	Fin de la segunda iteración.	Cuarta semana de mayo de 2010.

Sistema de gestión de eventos y servicios literarios para la UCI versión 1.0	Fin de la tercera iteración.	Primera semana de junio de 2010.
--	------------------------------	----------------------------------

2.3 Fase de Diseño.

Siguiendo las guías de la metodología XP, debe aparecer un diseño que describa qué clases hay y cómo interactúan. La metodología *XP* no requiere la representación del sistema mediante diagramas de clases utilizando el Lenguaje Unificado de Modelado (UML); en su lugar usa una técnica excelente para representar clases, esta es la que usa tarjeta *Clase-Responsabilidad-Colaboración* o CRC. No obstante el uso de diagramas UML puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

Con Drupal las tarjetas *CRC* son un tanto innecesarias, a tal punto que no se aplican, ya que este está compuesto por módulos y no por clases. Estos módulos son implementados de manera independiente para resolver una o varias funcionalidades. Después de implementados son añadidos al sistema y no colaboran con otros módulos debido a la independencia y al fin con que se implementan. La programación de estos módulos es una programación estructurada o por funciones.

2.3.1 Metáfora para el sistema.

Una metáfora es una historia que todo el mundo puede contar acerca de cómo funciona el sistema. Algunas veces podremos encontrar metáforas sencillas “*Programa de gestión de compras, ventas, con gestión de cartera y almacén*”. Las metáforas ayudan a cualquier persona a entender el objeto del programa. [14]

El sistema de gestión de eventos y servicios literarios (UCILITERARIA) para la Universidad de las Ciencias Informáticas, puede ser integrado a la Intranet de dicha universidad y a cualquier sitio web dentro de la misma. Este planteamiento está sustentado en que él gestiona los principales procesos en materia literaria dentro de la UCI. El sistema tiene la responsabilidad de brindarle a todo aquel que lo visite la oportunidad de revisar en un catálogo global los libros existentes en la Librería y en la Biblioteca, de esta forma puede decidir si lo compra o lo pide prestado. La información que se muestra por parte de la Biblioteca es sólo si está o no, y por la Librería muestra el precio y la cantidad en existencia. El sistema además debe brindar la posibilidad de autenticarse para así poder desplegar

CAPÍTULO 2

otras funcionalidades que son solamente para el personal que estudia o trabaja en la UCI, que posee usuario en el dominio. Estas funcionalidades consisten en la participación en los FAAL y concursos propuestos por el Centro Cultural. También debe brindar facilidades para que los Técnicos y Administradores gestionen todo lo referente a contenido y mantenimiento.

2.3.2 Diseño de la base de datos.

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la misma. El objetivo principal del diseño de bases de datos es generar tablas que modelan los registros en los que se guardará información. Un buen diseño de una base de datos evita redundancias en los datos almacenados y por consiguiente evita fallos de integridad, este aspecto es de vital importancia, aunque en ocasiones, por motivos de velocidad, es conveniente tener alguna información redundante siempre y cuando su uso esté justificado.

Drupal maneja la base de datos de forma independiente, creando así tablas a medida que las vaya necesitando y las relaciones las trabaja de igual forma para cada tabla ya que la base de datos es gestionada por él internamente. Aunque nosotros influimos en la gestión de tablas de forma directa nos sería bastante dificultoso hacer un modelo de datos para todas ellas ya que Drupal crea una serie de tablas que desde la perspectiva del negocio no son relevantes. Para ello se hace un modelo de datos con las tablas de la BD persistentes en negocio y que tienen una connotación significativa en la elaboración del sistema propuesto.

A continuación se muestra en dos ilustraciones los respectivos modelos, lógico (Figura 1) y físico (Figura 2).

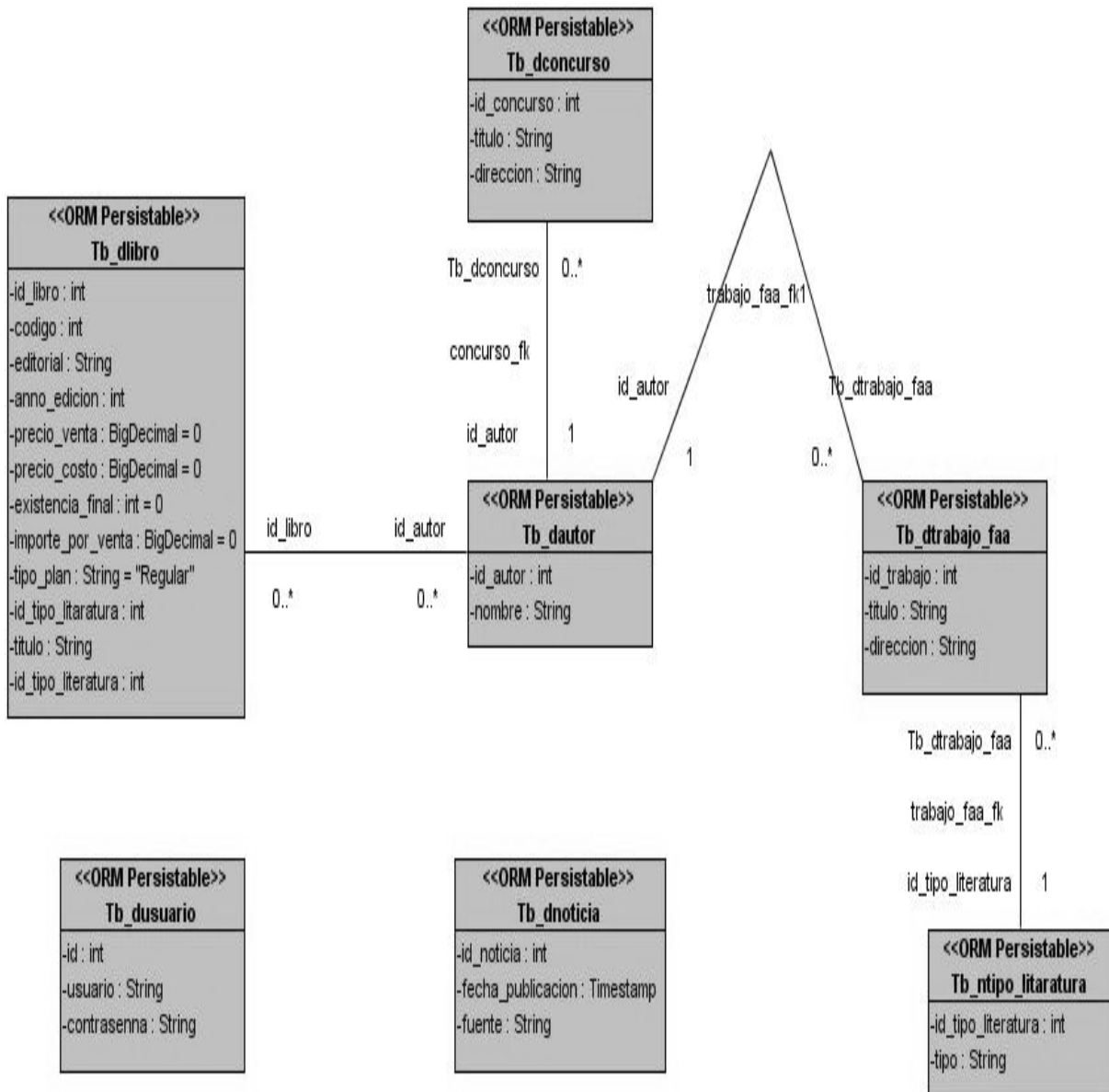


Ilustración 1 Modelo lógico

CAPÍTULO 2

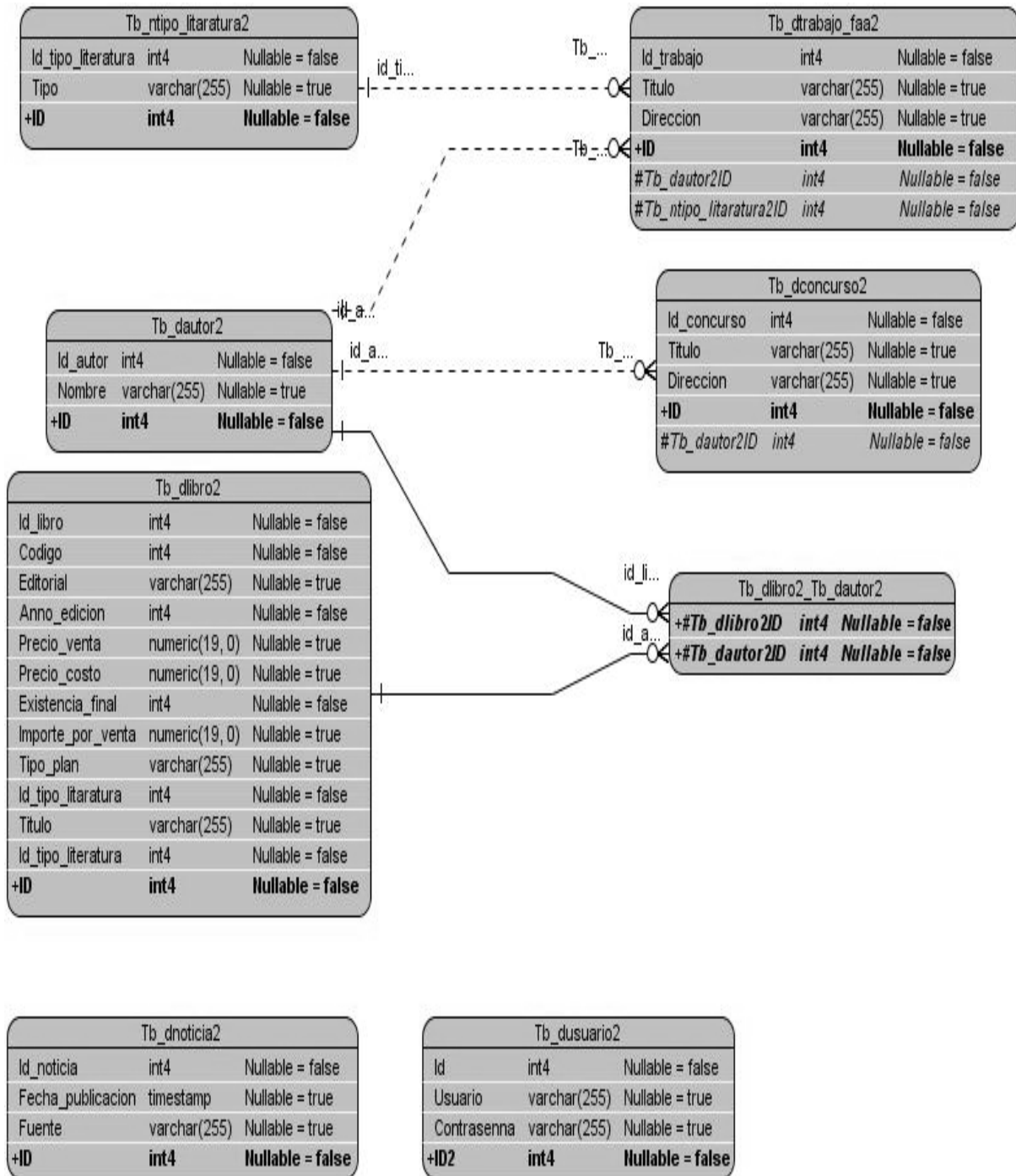


Ilustración 2 Modelo físico

2.3.3 Diagrama de despliegue.

Un *diagrama de Despliegue* muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los

CAPÍTULO 2

nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. [15]

A continuación se muestra el diagrama de despliegue del sistema propuesto.

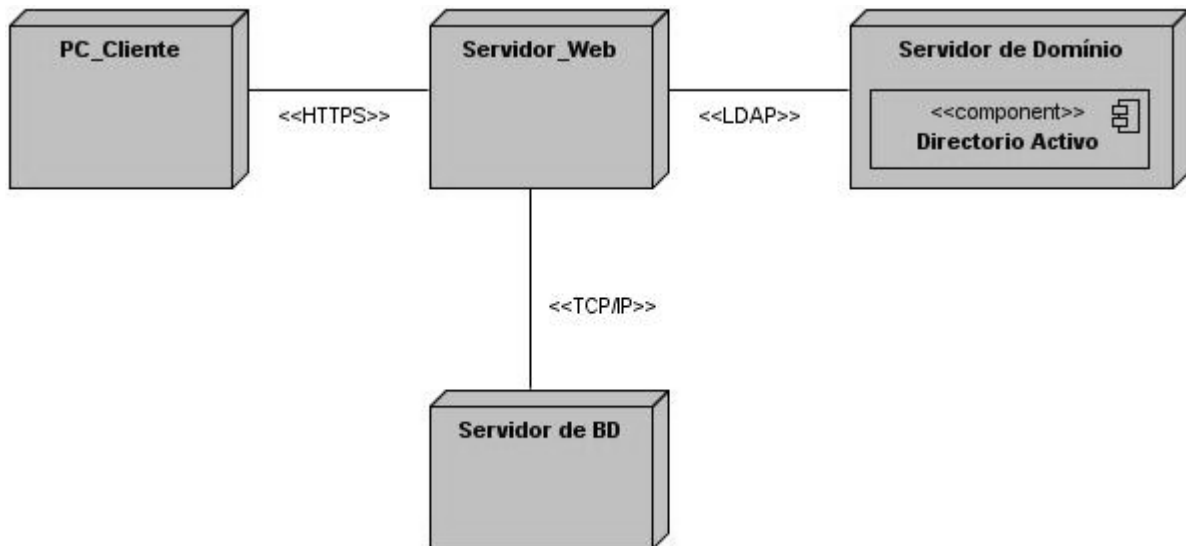


Ilustración 3 Diagrama de despliegue

2.4 Beneficios del sistema propuesto.

El sistema que se propone tiene como premisas: la gestión más dinámica de los eventos y servicios literarios, el perfeccionamiento técnico de los trabajadores encargados de la gestión, un mayor acercamiento de la comunidad universitaria a los procesos literarios y una verdadera transformación tecnológica en pos del desarrollo.

2.5 Conclusiones del capítulo.

En el presente capítulo se abordaron los temas referentes a las fases de exploración, planificación y diseño del proyecto basado en la metodología *XP*. Como resultado de la aplicación de las técnicas de ingeniería propuestas por esta metodología para estas fases se obtuvieron los artefactos necesarios para comenzar la implementación del sistema. Una mirada exhaustiva a los artefactos aquí generados brinda el primer acercamiento al sistema cuyo desarrollo se expone en el próximo capítulo. Las HU permitieron un mejor entendimiento del negocio ya que el cliente pudo describir cada una de las funcionalidades

CAPÍTULO 2

que deseaban llevar al sistema. Por otra parte, las tareas de ingeniería una vez realizadas permitieron resolver cada una de estas HU.

CAPÍTULO 3. DESARROLLO Y VALIDACIÓN DEL SISTEMA PROPUESTO.

3.1 Introducción.

En el presente capítulo se realizará un estudio detallado de la técnica de programación, plataforma y librerías usadas en el desarrollo del sistema propuesto, se observará además cómo es el funcionamiento de un módulo (apéndice fundamental en Drupal) y cómo se procede a la construcción del mismo, se describirá el algoritmo más complejo implementado y finalmente se describirán las pruebas de aceptación realizadas al sistema desarrollado.

3.2 Estudio detallado de la técnica de programación, plataforma y librerías usadas para el desarrollo de UCILITERARIA.

3.2.1 Estudio de los Módulos de Drupal.

Un módulo es la asociación de un conjunto de varias funciones en Drupal y ayudan a ofrecerle mayor funcionalidad a la web. Para crear un módulo en Drupal son necesarios tres tipos de ficheros de tipo PHP, de los cuales uno es de extensión .info, para portar la información que es pertinente al módulo como tal, como la versión y el autor del módulo, etc.; otro fichero es el de extensión .install, para crear las tablas y demás que se van a necesitar para que el módulo trabaje; y por último un fichero de extensión .module, que será en donde se escriban funciones que actúen en forma de enganche, al ser llamadas por Drupal a la hora de construir una página web y gestionar el contenido. Ejemplo de estas funciones escritas son: hook_help, hook_perm, hook_menu. Los módulos se pueden desactivar temporalmente para reducir la carga del servidor. En Drupal existen varios tipos de módulos importantes, como son el módulo de bloque, que da la posibilidad de crear contenido abreviado, que aparecen en la mayoría de los temas en los bloques de la izquierda y la derecha; y están además los módulos de nodo, que generan el contenido de la página del sitio, como puede ser la página del foro entre otros. A continuación se ofrece una descripción de los módulos del núcleo de Drupal y los opcionales [16]:

Módulo Filter. (Filtro)

- Permite la entrada de contenido para el sitio.

Módulo Watchdog. (Vigilante o Registro)

- Es una lista de los eventos almacenados, que contiene datos de uso, de rendimiento, de errores, de avisos y de información sobre las operaciones que se realizan. Es vital

CAPÍTULO 3

verificar el informe del vigilante con cierta regularidad, porque a menudo es el único que dice qué está pasando.

Módulo Node.

- Todo el contenido en un sitio web de Drupal se almacena y se trata como “nodos”. Un nodo es por ejemplo una página, una encuesta, una historia, un texto del foro, o una entrada del blog. Este módulo es el que posibilita que se enumere, clasifique y maneje todo el contenido en el sitio.

Módulo Block.

- Controla los bloques que se muestran alrededor del contenido principal.

Módulo User.

- Permite que los usuarios se registren, ingresen y terminen la sesión.

Módulo System. (Sistema)

- Proporciona opciones predefinidas para todo el sistema, se encarga de acceder a todas las páginas de administración.

Módulos Opcionales incluidos en Drupal.

Módulo Path.

- Permite indicar alias a los URLs de Drupal ayudando a mejorar la legibilidad de los mismos para los usuarios. Una página puede tener más de un alias.

Módulo Menu. (Menú)

- Brinda una interfaz para controlar y personalizar el completo sistema de menús que tiene Drupal.

Módulo Upload. (Subir archivos)

- Este módulo permite que los usuarios carguen y adjunten archivos al contenido.

Módulo Taxonomy.

- Activa la categorización del contenido.

Módulo Search. (Búsqueda)

- Este módulo permite la realización de búsquedas en todo el sitio creado.

3.2.2 Técnicas de programación y creación de módulos con Drupal.

Como ya se ha explicado anteriormente para comenzar a crear un módulo es necesario crear un archivo PHP, y guardarlo con el patrón nombre_módulo.module, otro con nombre_módulo.install, este sería para crear las tablas en las base de datos y un último archivo nombre_módulo.info para ofrecer información acerca del módulo. Todas las funciones del módulo que son usadas por Drupal, siguen la siguiente sintaxis: se nombran como {hook}_{nombre_módulo}, donde "hook" es una función predefinida llamada prefijo. Y Drupal llamará a esas funciones para conseguir datos específicos, teniendo así estos nombres bien definidos. Lo primero que se debe incluir al archivo .module debe ser el siguiente fragmento de código [16]:

```
<?php /* $Id$ */
```

Según el Estándar de Código de Drupal se puede omitir el *tag* de cierre `>`, la cadena `Id` permitirá mantener un seguimiento del número y fecha del módulo cuando sea convertido en CVS. Esto significa que cada vez que se modifique este módulo el controlador de versiones de Drupal le asignará un valor distinto al módulo y una fecha de modificación, permitiendo tener un control total sobre los cambios realizados al código y la persona que los realizó.

El formato general que debe tener el archivo .info es el siguiente [16]:

```
; $Id$
```

```
name = Nombre del módulo
```

```
description = "Una descripción de lo que hace el módulo."
```

```
dependencies = "Módulos de los que depende."
```

A continuación se muestran los ganchos (*hooks*) más importantes, que se utilizan a la hora de crear un nuevo módulo [16]:

Hook_help.

- Define temas de ayuda para que sea más fácil el uso del módulo que lo implemente.

Hook_access.

- Limita restricciones de acceso a los nodos.

Hook_perm.

- Se especifican quiénes tienen permisos del módulo. Se define quién puede acceder al contenido del sitio o administrar el módulo.

Hook_menu.

- La función hook_menu es llamada siempre por Drupal antes de crear las páginas, antes de mandarlas al navegador, y en ella pueden indicarse los puntos concretos en los que el módulo interactuará con la web.

Hook_view.

- Es un gancho usado por módulos de nodo. Esto permite a un módulo definir un método de mostrar los nodos.

Hook_form_alter.

- Permite hacer alteraciones a cualquier formulario antes que sea renderizado.

Hook_form.

- Es llamado cuando se necesita crear o editar algún artículo y devuelve el título de nodo, el área de texto, y otro campo específico del tipo de nodo.

Hook_submit.

- Se usa para las acciones que deben ocurrir cuando el módulo es guardado. Y también es utilizado después de que la validación ha tenido éxito y antes de insertar o actualizar.

Hook_validate.

- Esto es un gancho usado por módulos de nodo. Es utilizado para permitir al módulo verificar que el nodo se encuentra en un formato válido para añadirse al sitio.

Hook_delete.

- Es un gancho usado por módulos de nodo. Permite al módulo tomar medidas cuando un nodo es eliminado de la base de datos, por ejemplo, borrando la información de tablas relacionadas.

Hook_update.

- Indica cuando un nodo se actualiza. Es utilizado para permitir al módulo tomar acciones cuando un nodo es actualizado en la base de datos.

Hook_link.

- Define los enlaces internos de Drupal. Este *hook* posibilita añadir enlaces en diversos lugares en Drupal, como en bloques de navegación o en nodos.

Hook_settings.

- Para que el administrador pueda configurar lo que se va a mostrar, se crea la página de configuración. Casi siempre el administrador es quien tiene acceso a esta página.

Hook_search_item.

- Este módulo permite modificar los resultados de una búsqueda.

Hook_search.

- Permite a un módulo realizar búsquedas sobre el contenido que este define cuando una búsqueda de sitio es realizada.

Hook_nodeapi.

CAPÍTULO 3

- Actúa sobre nodos definidos por otros módulos. Este *hook* no es reservado para módulos de nodo. Al contrario, esto permite a módulos reaccionar a acciones que afectan todas las clases de nodos, independientemente de si algún módulo definió el nodo.

Hook_filter.

- Define filtros de contenidos. El contenido en Drupal es pasado por todos los filtros permitidos, antes que sea mostrado. Con este se puede modificar el contenido al gusto del administrador de sitio.

3.2.3 Estándares de codificación.

Los estándares de codificación son las reglas que se van a seguir para escribir el código. A continuación se explican más detalladamente, los estándares de codificación seguidos en el desarrollo de este sistema:

➤ Sangrías y Longitud de línea.

Usar una sangría de 4 espacios sin usar la tecla “*tab*”. Esto ayuda a evitar problemas con *diffs*, *patches*, historia del CVS y anotaciones.

➤ Llamadas de Función.

Las funciones deberá de ser llamadas sin espacios entre el nombre de la función, el paréntesis abierto y el primer parámetro; espacios entre comas y cada parámetro, y sin espacios entre el ultimo parámetro, el paréntesis cerrado y el punto y coma.

➤ Definiciones de Función.

En las declaraciones de función, la llave de apertura empieza al final de la línea de definición después de un espacio.

➤ Inclusión de Código.

En cualquier parte que se esté incluyendo incondicionalmente un archivo de una clase, se usa `required_once`. En cualquier parte donde se esté condicionalmente incluyendo un archivo de clase (por ejemplo, métodos de fábrica) se usa `include_once`. Cualquiera de esas asegurará que el archivo sea incluido únicamente una vez.

➤ **Convención de Nombres de Variables, Funciones, Clases.**

- **Variables:**

Estas deberán nombrarse con un prefijo de tres letras el cual define el tipo de dato de la misma, seguido de un guión bajo y el nombre descriptivo de la variable.

- **Funciones:**

Estas deberán nombrarse con palabras minúsculas seguido de un guión bajo, para separar palabras en el nombre de la misma.

- **Clases:**

Estas deberán nombrarse con la primera palabra iniciando con minúscula y separando las demás palabras con letra mayúscula.

3.2.4 Análisis de posibles componentes, módulos e implementaciones ya existentes que pueden ser reutilizados.

A continuación se relaciona un conjunto de componentes, módulos e implementaciones ya existentes que pueden ser reutilizados así como sus características fundamentales [17]:

Administration Menu.

El módulo de Administración de menú proporciona una interfaz de administración tema independiente para Drupal. Es una ayuda para los usuarios principiante de Drupal procedentes de otros CMS, un ahorro de tiempo para los administradores del sitio de Drupal, y definitivamente una necesidad para los desarrolladores de Drupal y constructores sitio.

El módulo está basado en CSS atractivo y exclusivamente en la parte superior del sitio web. No sólo contiene los elementos del menú regular (tareas locales).

Admin Role.

Este módulo es de ayuda para mantener una función de administrador que tiene todos los permisos disponibles. Por defecto, Drupal sólo tiene un súper usuario y este módulo ayuda a mejorar esto de forma radical.

Automatic Nodetitles.

CAPÍTULO 3

"Auto_nodetitle" es un módulo pequeño y eficiente que permite ocultar el campo de título de contenido en el formulario. Se puede configurar para generar el título por un patrón determinado. Los usuarios avanzados también pueden proporcionar algo de código PHP, que se utiliza para la generación automática de un título adecuado.

Automated Logout.

Este módulo proporciona un administrador de sitio que desconecta a un usuario tras pasar un periodo de inactividad especificado. Es altamente personalizable e incluye "las políticas del sitio" por el papel en la aplicación de cierre de sesión.

Counter.

El módulo contador cuenta el número de usuarios en el sitio.

Características:

- Cuenta en la web la entrada de usuarios por día, minuto, incluso segundos.
- Cuenta los usuarios registrados y no registrados.
- Emite estadísticas de visitantes por días, semanas, meses y años.

FCKeditor.

Este módulo permite a Drupal reemplazar campos de textos con un editor HTML visual, a veces llamado WYSIWYG editor. Este editor de texto HTML trae muchas de las funciones de edición WYSIWYG de gran alcance de los editores de escritorio conocidos, como Word, a la web. Es relativamente liviano y no requiere ningún tipo de instalación en el equipo cliente.

LDAP Integration.

Módulo que le permitirá autenticar su sitio contra un dominio. Este es válido para cualquier dominio, además tiene acceso pleno o restringido a toda la información de la base de datos del dominio.

Filter.

La lista de módulos puede llegar a ser bastante grande cuando se trata de un sitio grande. La capacidad de encontrar rápidamente el módulo que está buscando sin tener que depender de la función de búsqueda de los navegadores.

CAPÍTULO 3

Cuando se habilita mediante la configuración del filtro del módulo de página, un nuevo tema de diseño del módulo se aplica. Este diseño proporciona una ficha para cada paquete, así como una ficha en que se muestran todos los módulos en orden alfabético. El campo de texto filtro está disponible en cada ficha, pero actualmente no es compatible con autocompletado.

Pathauto.

El módulo de creación automática genera automáticamente alias de ruta para los diversos tipos de contenido (nodos, categorías, usuarios), sin necesidad de que el usuario especifique manualmente el alias de ruta. Esto le permite obtener los alias como / categoría / mi-nodo-title.html en lugar de node/123. Los alias se basan en un "patrón" que el administrador del sistema puede controlar.

Poormanscron.

Un módulo que ejecuta la operación cron de Drupal usando el navegador normal o solicitudes de página en lugar de tener que crear un crontab para solicitar el script cron.php. El módulo inserta una pequeña cantidad de código JavaScript en cada página de su sitio para que cuando transcurra cierto tiempo desde la última ejecución del cron, llame a una petición AJAX para ejecutar las tareas cron. Los usuarios no notarán ningún tipo de demora o interrupción durante la visualización de su sitio. Sin embargo, este enfoque requiere que su sitio reciba el tráfico ordinario de los visitantes con el fin de activar la solicitud cron.

Token.

Los créditos son pequeños trozos de texto que puede ser colocado en grandes documentos a través de simples marcadores de posición, como site-nombre o usuario. El módulo proporciona un token central de la API para los módulos utilizar estas fichas, así como exponer sus valores simbólicos propios.

Tenga en cuenta que el módulo Token no proporciona todas las funciones visibles para el usuario por su cuenta, sólo proporciona servicios de asistencia en razón de otros módulos.

Content Construction Kit (CCK).

Es un potente módulo que permite a los desarrolladores definir campos personalizados y los tipos de contenido. Existe una variedad de módulos de extensión para CCK que permite

CAPÍTULO 3

definiciones espaciales en campos tales como imágenes, fechas, y de valores computarizados.

3.2.5 Descripción del módulo Vlibrería y del algoritmo más complejo en su desarrollo.

En la elaboración del sistema fue necesario implementar un módulo llamado Vlibrería (Librería virtual). Este módulo no es más que un catálogo general de búsqueda de libros almacenados en la base de datos del sistema. El mismo funciona con criterios de búsquedas definidos por el usuario, brindándole así la posibilidad de acceder a la información de un libro determinado. El módulo fue implementado con lenguaje PHP y siguiendo el paradigma de una programación no orientada a objetos, sino estructurada.

Las funcionalidades implementadas tienen como objetivo responder a las necesidades de los usuarios mediante una búsqueda en la tabla libro de la base de datos. Estas necesidades están enmarcadas en querer tener información en materia de valor y existencia de un título dado, en la librería de la universidad. Respondiendo a esto se implementa el módulo *vlibreria*, el mismo está compuesto por cinco tipos de archivos: *vlibreria.info*, que muestra la información general del módulo; *vlibreria.css*, donde se le aplican los estilos; *vlibreria.js*, donde se encuentra el código *javascript* y se especifican los tipos de filtro que se llevaran a cabo para realizar la búsqueda; *vlibreria.pages.inc*, donde se construyen los formularios para introducir los datos y en la forma que serán mostrados; y por último está *vlibreria.module*, que sería el archivo principal donde se procede a implementar todas las funciones. Estas funciones son las encargadas de responder mediante un código en el lenguaje PHP a cada necesidad del usuario.

En el capítulo anterior se explica por qué cuando se utiliza la metodología XP es innecesario usar tarjetas CRC si se desarrolla con Drupal. La no necesidad de este está dada en que la implementación en Drupal es mediante módulos y no por clases. Estos módulos se implementan mediante funciones siendo así una programación estructurada. Teniendo en cuenta esto y siguiendo la estrategia utilizada por XP en sus tarjetas CRC, se conforma una tabla Modulo-Responsabilidad-Colaboración que nos muestra los datos principales del módulo y la colaboración de este con los demás en del sistema.

Módulo: Vlibreria.	
Responsabilidad	Colaboración

Realizar búsquedas de libros en la base de datos mediante consultas.	CCK
Filtrar los libros mediante criterios de búsquedas.	
Mostrar información necesaria de cada libro para su futura compra.	

Algoritmo más complejo.

La función *vliblioteca_search_data* implementa el algoritmo más complejo entre los que fueron implementados en el módulo *Viblioteca*. Esta se encarga de realizar búsqueda de libros siguiendo criterios definidos por: título del libro, autor, tipo de literatura y rango de precios. Primeramente la función se conecta a la base de datos, después define los criterios de búsqueda, se realiza una consulta a la base de datos del sistema para listar los libros que cumplan con dichos criterios, y por último la función devuelve el resultado en una vista del sistema. Para ver el código completo de esta función remitirse al Anexo IV.

Diagrama de componente.

Este diagrama muestra componentes, subsistemas de implementación y sus relaciones. Además, muestra la organización y las dependencias lógicas entre un conjunto de componentes de software. Para la realización del módulo *Viblioteca* se modeló el siguiente diagrama de componentes:

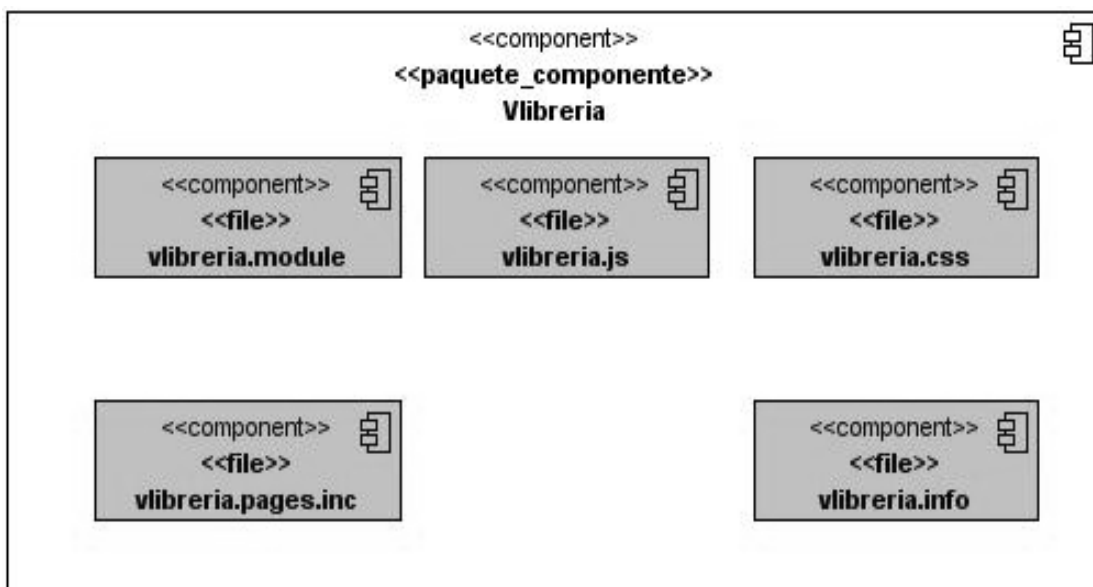


Ilustración 4 Diagrama de componentes, *Viblioteca*.

3.3 Pruebas, métodos de pruebas.

Las pruebas son acciones en las cuales el sistema es ejecutado bajo condiciones o requerimientos determinados. Los resultados son chequeados y registrados. Las pruebas verifican los resultados de la implementación del sistema. El modelo de prueba indica cómo han de ser probados aspectos determinados del producto o software. Hay multitud de conceptos asociadas a las tareas de prueba. Clasificarlas es difícil, pues no son mutuamente disjuntas, sino muy entrelazadas. En lo que sigue se intentará una pequeña organización de acuerdo a sus características.

3.3.1 Pruebas unitarias.

Escribir pruebas unitarias es una buena práctica de desarrollo web, aunque puede resultar difícil de poner en práctica. La metodología *XP* propone escribir estas pruebas antes de la implementación de la función a probar. Las pruebas unitarias son generalmente de caja negra, o sea que solo se observa el resultado obtenido de la llamada a las funciones a probar de acuerdo a los parámetros de entrada, si una función retorna al menos un valor no esperado se procede a su depuración paso a paso para detectar los errores que pueda contener.

3.3.2 Pruebas de aceptación.

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones las historias de usuarios seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una historia de usuario no se considera completa hasta que no ha pasado por sus pruebas de aceptación. Durante el desarrollo de las tres iteraciones del proyecto se realizaron las pruebas de aceptación para las 18 historias de usuario que conforman el sistema. Todas las pruebas de aceptación arrojaron resultados satisfactorios con lo que el cliente expresa su conformidad y satisfacción con el sistema creado. (Ver Anexo III).

3.4 Conclusiones del capítulo.

En el presente capítulo se realizó un estudio de las técnicas de programación y de los módulos de Drupal, además se pudo apreciar cómo se procede a la creación de los mismos, una mirada a los estándares de codificación aquí expuestos significa el primer paso para comenzar a desarrollar. Se ha podido apreciar cómo se realiza la integración de módulos ya existentes y sus descripciones respectivas. La implementación del módulo Vlibrería resultó ser la más compleja en el desarrollo del sistema, en el presente capítulo se muestra una amplia descripción del mismo. Una vez implementado el sistema se llevan a cabo las pruebas, validando así que funcione correctamente todo lo desarrollado en correspondencia con las funcionalidades definidas.

CONCLUSIONES

CONCLUSIONES.

En la actualidad y con el desarrollo de las TIC se hace necesario informatizar todas las instituciones, para de esta forma poder asistir el ingenio humano por computadoras.

Como parte de este trabajo de diploma se desarrolló satisfactoriamente un sistema de gestión de eventos y servicios literarios para la Universidad de las Ciencias Informáticas, lográndose caracterizar, modelar e implementar la gestión de eventos y servicios literarios en dicho centro. Todo este desarrollo trajo consigo que finalmente se perfeccionara todo el proceso literario en la universidad, consiguiéndose un importante avance en las áreas involucradas.

El análisis de las tecnologías relacionadas con el objeto de estudio permitió seleccionar la metodología y herramientas adecuadas para el desarrollo de la propuesta de solución. La utilización de tecnologías de libre distribución para el desarrollo del sistema hace de este un producto factible y recomendado para ser utilizado en todas aquellas instituciones que necesiten gestionar eventos y servicios literarios similares a los planteados en el caso de estudio.

El desarrollo del proyecto utilizando las guías de la metodología *XP* permitió minimizar el tiempo en que los usuarios finales obtuvieron un producto listo para utilizarse, sin sacrificar la calidad de la documentación generada y dejando las puertas abiertas para futuras mejoras.

Se documentó todo el trabajo desde sus inicios, lo cual facilitará su estudio por parte de futuros desarrolladores, permitiendo una comprensión rápida y sencilla de la concepción general del sistema.

RECOMENDACIONES

RECOMENDACIONES.

1. Implementar una nueva funcionalidad que permita la compra en línea de libros en la Librería.
2. Agregarle las funcionalidades necesarias al sistema para llevarlo a nivel superior y gestionar la Feria Internacional del Libro en Cuba.
3. Implementar un catalogo de búsqueda general que comparta el contenido entre la Biblioteca y la Librería.

BIBLIOGRAFÍA REFERENCIADA

BIBLIOGRAFIA REFERENCIADA.

1. *Kernel Error*. (s.f.). Citado el 15 de abril de 2010, de Kernel Error: <http://kernelerror.net/programacion/php/arquitectura-3-capas/>
2. *Wapedia*. (s.f.). Citado el 05 de 05 de 2010, de http://wapedia.mobi/es/Metodolog%C3%ADa_de_desarrollo_de_software
3. Sanchez, M. A. *Metodologías De Desarrollo De Software*. [Citado el 27 de marzo del 2010].
4. Gracia, J. (s.f.). *Patrones de diseño. Diseño de Software Orientado a Objetos*. Citado el 6 de marzo de 2010, de IngenieroSoftware.: <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
5. *Manual de MySQL y Manual de PostGreSQL*. (s.f.). Citado el 9 de marzo de 2010, de <http://www.mysql.com/documentation/index.html> y <http://www.mysql.com/documentation/index.html>
6. Fernández, S. P. (07 de julio de 2006). *El servidor Web. Arquitectura y funcionamiento*. Citado el 11 de marzo de 2010, de <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=366>
7. Félix, A. d. (s.f.). *El servidor de web Apache: Introducción práctica*. Citado el 12 de marzo de 2010, de <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/index.html>
8. *Programación Web*. (s.f.). Citado el 15 de marzo de 2010, de <http://www.lenguajes-de-programacion.com/programacion-web.shtml>
9. Paz, L. P. *Perl I primera parte*. [Citado el 20 de marzo 2010].
10. *Visual Paradigm For Uml - Presentation Transcript*. (s.f.). Citado el 25 de marzo de 2010, de <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>
11. *Free Download Manager*. (s.f.). Citado el 26 de marzo de 2010, de http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/

BIBLIOGRAFÍA REFERENCIADA

12. *Accesibilidad y CMS: Drupal, una herramienta de gestión de contenido con vocabulario.* (s.f.). Citado el 26 de marzo de 2010, de <http://hugodonairemiles.blogspot.com/2006/11/accesibilidad-y-cms-drupal-una.html>
13. *27 Características de DRUPAL.* (s.f.). Citado el 1 de abril de 2010, de <http://www.pilos.com.co/drupal/27-caracteristicas-de-drupal>
14. Solís., M. C. (n.d.). *willydev.* Retrieved marzo 20, 2010, from willydev: <http://www.willydev.net/descargas/prev/ExplicaXp.pdf>
15. *Guía de Usuario de Enterprise Architect 7.0.* (s.f.). Recuperado el 28 de marzo de 2010, de Guía de Usuario de Enterprise Architect 7.0: <http://www.sparxsystems.com.ar/download/ayuda/index.html?deploymentdiagram.htm>
16. Drupal., S. o. (s.f.). *Drupal.* Citado el 10 de mayo de 2010, de <http://drupal.org/handbook>
17. Drupal, S. o. (s.f.). *Drupal.* Citado el 10 de mayo de 2010, de <http://drupal.org/project/Modules>

BIBLIOGRAFÍA

BIBLIOGRAFIA.

1. Ingeniería de software 1. Scribd. [En línea] [Consultada el: 7 de abril de 2009.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>
2. Gracia Joaquín. Patrones de diseño. Diseño de software orientado a objetos. [En línea] 27 de mayo de 2005. [Consultada el: 29 de abril de 2009.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
3. Kniberg Henrik. Scrum y XP desde las trincheras. 2007.
4. Coding standards. Drupal. [En línea] [Consultada el: 11 de mayo de 2009.]. Disponible en: <http://drupal.org/coding-standards>
5. Hook. Drupal. [En línea] <http://api.drupal.org/api/group/hooks>
6. Letelier Patricio. Pruebas de Aceptación como conductor del Proceso Software. [En línea] 2007. <http://zweb.iti.upv.es/groups/squac/events/JTS2007/slides/3demayo16.45-PatricioLetelier.pdf/attach/3demayo16.45-PatricioLetelier.pdf>.
7. Gracia Joaquín. Patrones de diseño. Diseño de Software Orientado a Objetos. Ingeniero de Software. [En línea] 21 de abril de 2005. [Consultada el: 5 de mayo de 2009.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
8. Rojas Johanna; Barrios Emilio; Grupo ARQUISOFT. Métodos de prueba de caja negra. [En línea] 2007. [Consultada el: 1 de mayo de 2009.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node28.html>
9. Bartle Phil. Información para la gestión y gestión de la información. Potenciación comunitaria. [En línea] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

BIBLIOGRAFÍA

10. Cascante Patricia y Fonseca Mima. Intranet: Una ventana a la comunicación interna de las organizaciones. [En línea] 2006. <http://www.uned.ac.cr/Redti/cuarta/art2.pdf>.
11. Técnicas de programación. Autorneto. [En línea] mayo de 2008. <http://autorneto.com/tecnologia/software/tecnicas-de-programacion/>.
12. Rego Castillo Danis y Chaveco Laurencio Alfonso. Desarrollo de una intranet para el grupo empresarial de transporte por ómnibus. Ciudad Habana: s.n., 2007.
13. Blanco Cuaresma Sergi. Metodología de desarrollo. Marble Station. [En línea] <http://www.marblestation.com/?p=644>.
14. Canós José H., Letelier Patricio y Penadés M. Carmen. Metodologías Ágiles en el Desarrollo de Software. [En línea] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
15. Fases de la Programación Extrema. [En línea] <http://programacionextrema.tripod.com/fases.htm>.
16. Escribano, Fernández. Ingeniería de software II. Introducción a Extreme Programming. [En línea] <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf>.
17. Rodríguez Villar, Malay. de Diploma Introducción de procedimientos ágiles en la producción de software en la facultad 7 de la Universidad de las Ciencias Informáticas. Ciudad Habana: s.n., 2007.
18. Dondo, Agustín. ¿Por qué elegir PHP? PHP en Castellano. [En línea] <http://www.programacion.com/php/articulo/porquephp/>.
19. Programación en PHP. <http://www.desarrolloweb.com/manuales/12/>. [En línea] <http://www.desarrolloweb.com/manuales/12/>.

BIBLIOGRAFÍA

20. The Features, Mission, and Principles of the Drupal. Drupal. [En línea] 2008. <http://drupal.org/getting-started/project-features>.
21. Alarcón Medina, José Manuel. Administración SGBD PostgreSQL. Valencia : s.n., 2006.
22. Gómez, Carlos. PostgreSQL SGBD de código abierto. TechHerald.com. [En línea] 2008. <http://techerald.com/view.jsp?postgresql-301220081345.html>
23. ¿Qué es un servidor web (Web Servers)? - Definición de servidor web. masadelante.com. [En línea] 2008. <http://www.masadelante.com/faq-servidor.html>
24. Apache. CAMPUS VIRTUAL. [En línea] <http://www.cvirtualuees.edu.sv/mod/book/view.php?id=552>
25. Pressman, Roger S. Ingeniería del Software. un enfoque práctico. s.l.: quinta edición.

GLOSARIO.

CITMATEL: Tecnologías de la Información y Servicios Telemáticos.

CASE: Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y de dinero.

CMS: Un sistema de gestión de contenidos.

CVS: Es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto y permite que distintos desarrolladores colaboren.

FEU: Federación Estudiantil Universitaria.

FAAL: Festival de Artistas Aficionados de Literatura.

HU: Historias de usuarios.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

JCE: Jornada Científica Estudiantil.

RUP: Proceso Unificado de Desarrollo de Software.

Scrum: Metodología Ágil de Desarrollo de Software.

Scrum-XP: La unión de dos Metodologías Ágiles de Desarrollo de Software.

TIC: Tecnologías de la Informática y las Comunicaciones.

UCILITERARIA: Nombre del sistema de gestión de eventos y servicios literarios en la Universidad de las Ciencias Informáticas.

UCI: Universidad de las Ciencias Informáticas.

URL: Localizador uniforme de recursos, es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

UML: Lenguaje Unificado de Modelado.

WWW: Significan World Wide Web o telaraña mundial.

XP (Extreme Programing): Metodología Ágil de Desarrollo de Software.