

Universidad de las Ciencias Informáticas

Facultad 1



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

Título:

**Desarrollo del Catálogo de productos
para la librería de la Universidad de
las Ciencias Informáticas**

Autora: Yoanna González Aymerich

Tutores: Ing. Jorge Arias Sojo

Ing. Yohana González Moreno

**La Habana, Cuba
Junio de 2013**

PENSAMIENTO



***“Quien quiere hacer encuentra medios,
quien no quiere hacer encuentra
justificaciones”***

“Ernesto Guevara”

DECLARACIÓN DE AUTORÍA

Declaración de autoría

Declaro que soy la única autora de este trabajo y autorizo al Centro de Informatización Universitaria (CENIA) de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autora
Yoanna González Aymerich

Tutor
Ing. Jorge Arias Sojo

Tutor
Ing. Yohana González Moreno

DATOS DE CONTACTO

Datos de contacto.

Ing. Jorge Arias Sojo: Ingeniero en Ciencias Informáticas, graduado en el 2008 en la Universidad de las Ciencias Informáticas. Se desempeña como jefe del proyecto Intranet UCI 3.0 del Centro de Informatización Universitaria. Dirección de correo jarias@uci.cu, Edificio 46 Apto 46303, teléfono 8372120.

Ing. Yohana González Moreno: Ingeniera en Ciencias Informáticas, graduada en el 2009 en la Universidad de las Ciencias Informáticas. Se desempeña como analista principal del proyecto Intranet UCI 3.0 del Centro de Informatización Universitaria. Dirección de correo ygmoreno@uci.cu.

AGRADECIMIENTOS

Agradecimientos

Primero que nada darle gracias a la vida por el maravilloso bebé que tengo que ha sido mi principal motivación para culminar este largo y difícil camino que he tenido en la Universidad. Eres mi razón de ser y mi mayor logro.

Agradecerles eternamente a mis padres Miladys y Omar, por la educación, amor y cariño que me ha brindado durante toda mi vida y en especial a mi mami por ser mi principal apoyo, sustento y fuerzas para realizar mis sueños. Por creer en mí aún cuando yo había perdido las esperanzas y por apoyarme en las decisiones que he tomado en la vida, por tus consejos, por tu paciencia, por tu ayuda infinita e incondicional, gracias. Te amo con la vida.

A mi esposo y padre de mi hijo, por alegrar mis días y mis noches, por la comprensión, ayuda y confianza que has tenido conmigo y por hacerme sentir una persona especial en tu vida. Te amo.

A mi familia, por la preocupación, apoyo y ayuda, en especial a mis tías Ani y Yolanda, mi prima Sandra y mi primo Misael por estar presente cuando los he necesitado.

AGRADECIMIENTOS

A mis buenas amigas y hermanas Doralys, Isis y Yanet, por compartir buenos e inolvidables momentos, por los consejos y por estar presente en los momentos que me han hecho falta.

A mis tutores, Jorgito y Yohana por las críticas, ayuda, sinceridad y grandísima paciencia que han tenido conmigo, en especial a Jorgito por llevar juntos esta gran batalla, de corazón mil gracias.

Agradecerle a la Facultad 1, a los profesores que ayudaron en mi formación, a Mayelin, al decano Damián, a la profesora Eileen, por su inmensa comprensión y darme la oportunidad de finalizar mis estudios.

A los miembros del tribunal, por sus críticas constructivas y sugerencias.

A todos mis más sinceros agradecimientos.

Yoanna González Aymerich

DEDICATORIA

Dedicatoria

*A mi bebito Christopher
y a mi mamá por ser las personas más
importantes en mi vida.*

Yoanna

RESUMEN

Resumen.

La presente investigación forma parte de la línea Soluciones Intranet's y Portales, perteneciente al Centro de Informatización Universitaria (CENIA). Tiene entre sus labores la informatización de la librería de la Universidad de las Ciencias Informáticas. Este trabajo de diploma se realiza con el fin de dar solución al problema existente en dicha librería, debido a que todo el trabajo del personal se realiza de forma manual resultando un tanto engorroso para los mismos. Por tal situación es necesaria la implementación de un módulo capaz de resumir en un catálogo de productos todo el proceso de venta que se realiza en la librería por sus trabajadores. Para ello se realizó un estudio del arte y de las tecnologías, arquitecturas y herramientas dictadas por el CENIA, como Sistema de Gestión de Contenidos se escogió Drupal y como entorno de desarrollo integrado NetBeans. El levantamiento de los requisitos funcionales y no funcionales, las descripciones de las historias de usuarios ayudaron en la implementación de la aplicación.

Palabras clave:

Portales, librería, Intranet's.

ÍNDICE

Índice.

Introducción.....	1
Capítulo 1. Fundamentación teórica.	4
1.1 Conceptos fundamentales asociados al dominio del problema.....	4
1.2 Sistemas existentes de catálogos de librerías.	4
1.2.1 Sistemas internacionales.	5
1.2.2 Principales características, ventajas y desventajas de los catálogos de librerías a nivel internacional.....	5
1.2.3 Aportes de los sistemas de catálogo en línea a nivel internacional.....	6
1.3 Sistema de Gestión y Contenidos (CMS).....	6
1.3.1 Características de Drupal 7.....	8
1.4 Metodologías de desarrollo de software.....	9
1.4.1 Integración de las metodologías XP y SCRUM con el Proceso de Administración de Requisitos del nivel 2 de CMMI.	9
1.5 Lenguajes de programación utilizados.	11
1.5.1 Lenguajes de marcas.....	11
1.5.2 Lenguajes del lado del cliente.....	12
1.5.3 Lenguaje del lado del servidor.	12
1.6 Herramientas utilizadas para el desarrollo del sistema.....	13
1.6.1 Sistema gestor de base de datos.....	13

ÍNDICE

1.6.2	PostgreSQL v 9.1.....	13
1.6.3	Servidor web.....	14
1.6.4	Apache v2.2.21.....	14
1.6.5	Visual Paradigm v8.0.....	15
1.6.6	NetBeans IDE v7.2.....	16
1.7	Conclusiones parciales.....	16
Capítulo 2. Características del Sistema.....		18
2.1	Propuesta de solución.....	18
2.2	Descripción de la propuesta de solución.....	18
2.3	Especificación de los requisitos del software.....	19
2.3.1	Lista de Reserva del Producto.....	19
2.3.2	Requisitos funcionales.....	20
2.3.3	Requisitos no funcionales.....	22
2.4	Historias de Usuarios.....	23
2.4.1	Descripción de las historias de usuarios.....	24
2.5	Modelo de dominio.....	28
2.6	Diagrama de despliegue.....	29
2.7	Beneficios del sistema.....	30
2.8	Conclusiones parciales.....	30

ÍNDICE

Capítulo 3. Implementación y prueba de la solución propuesta	31
3.1 Arquitectura del sistema	31
3.2 Patrones de diseño.....	33
3.3 Estándares de codificación.....	34
3.3.1 Estándares de codificación de la base de datos.....	34
3.3.2 Estándares de codificación utilizados en la programación.	35
3.4 Diagrama de componentes	37
3.5 Modelo de prueba	38
3.5.1 Pruebas realizadas al sistema	39
3.5.2 Pruebas de caja negra.....	39
3.5.3 Resultados de las pruebas de caja negra.....	40
3.5.4 Ejemplo de casos de pruebas.....	41
3.5.5 Pruebas de carga y estrés.	47
3.6 Conclusiones parciales.....	48
Conclusiones Generales.....	49
Recomendaciones.....	50
Bibliografía referenciada.	51
Bibliografía consultada.	53
Glosario de Términos.....	54

INTRODUCCIÓN

Introducción

La creciente necesidad del uso de la Internet ha sido la impulsora principal del acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). Es por tal motivo, que muchas empresas promocionan sus productos por medio de herramientas digitales como lo son los catálogos en línea, los cuales contienen la información técnica en forma de registros y que corresponde a los títulos de las obras representados físicamente en libros, revistas y periódicos, que se ofrecen para su consulta [1]. Su uso gestiona con mayor eficiencia sus recursos; la cual se mide por las grandes posibilidades que brindan en el almacenamiento, control y acceso de los elementos de su inventario. Dichas herramientas forman un estrecho vínculo con la Internet, ya que comparten la comunicación y la transforman en adquisición de bienes y servicios, además de beneficiar al cliente al proporcionar una mayor cantidad de datos.

Cuba no está ajena al creciente auge tecnológico que se manifiesta en este momento, por tal razón, uno de los principales proyectos fundados por la revolución cubana, ha sido la Universidad de las Ciencias Informáticas (UCI), que tiene como principal objetivo formar ingenieros aptos para desarrollar aplicaciones y sistemas informáticos.

La idea de relacionar la informática con otras ramas especializadas, ha hecho que en la Universidad existan diversos medios de entretenimiento dedicados a estudiantes y trabajadores. Uno de estos lugares es la librería ubicada dentro de la propia Universidad, la misma se dedica a la venta de libros, revistas y folletos. En estos momentos el local no cuenta con un sistema computarizado capaz de agilizar todo el proceso de venta que realizan sus trabajadores, trayendo consigo por su realización manual un gasto considerable de materiales y la pérdida o deterioro de las planillas elaboradas. La lentitud de ejecutar una búsqueda manual resulta un tanto tediosa, además de no poder mantener informadas a las personas con relación al abastecimiento de los nuevos productos que se ofertan, disminuyendo de esta forma el número de ventas y la promoción de sus artículos; quedando identificado su **situación problemática**.

Con el propósito de resolver la problemática se define como **problema a resolver**: ¿cómo mejorar el proceso de venta en la librería de la Universidad de las Ciencias Informáticas?

La presente investigación tiene como **objeto de estudio** la organización de la información en librerías en línea, donde el **campo de acción** se define como la organización de la información en librerías a través de catálogos en línea.

INTRODUCCIÓN

Este trabajo de diploma tiene como **objetivo general**: desarrollar un catálogo para registrar todo el proceso de venta que se realiza en la librería de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se trazaron los siguientes **objetivos específicos**:

- Elaborar el marco teórico-conceptual de la investigación.
- Realizar un análisis y diseño del sistema.
- Implementar un catálogo que cumpla con las características específicas de la librería.
- Validar la solución desarrollada.

Se plantea como **idea a defender**: con el desarrollo de un catálogo que gestione las ventas de los productos de la librería de la Universidad, se permitirá organizar y actualizar sistemáticamente los artículos existentes, se agilizarán las búsquedas a realizar, se minimizará el gasto de materiales y el tiempo de sus trabajadores.

Para dar cumplimiento al objetivo general se definen las siguientes **tareas**:

- Realizar un estudio de homólogos a nivel nacional e internacional
- Entrevistar a los responsables de la librería de la Universidad de las Ciencias Informáticas.
- Realizar estudios bibliográficos relacionados con el tema.
- Estudiar los procesos de venta de la librería de la Universidad de las Ciencias Informáticas.
- Identificar los requisitos funcionales y no funcionales que debe cumplir el sistema informático.
- Fundamentar el uso de metodologías, tecnologías y herramientas a utilizar.
- Implementar un catálogo que cumpla con las características específicas de la librería.
- Realizar las pruebas para validar el funcionamiento eficiente del módulo.

Para llevar a cabo la investigación se tuvieron en cuenta los siguientes **métodos científicos y técnicos**:

Métodos teóricos:

Histórico-lógico: se empleó para el estudio y profundización de la evolución de las aplicaciones y de sistemas similares, facilitando la búsqueda de soluciones al problema planteado.

Analítico-sintético: se utilizó en el estudio de la documentación especializada relacionada con el tema lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.

Métodos empíricos:

INTRODUCCIÓN

Entrevista: se efectuaron entrevistas de forma informal a los trabajadores de la librería para conocer cómo funciona el proceso de venta en esta entidad.

El trabajo de diploma consta de 3 capítulos, a continuación se explicará una breve descripción de los mismos.

Capítulo 1. “Fundamentación Teórica”: se realiza un estudio del marco conceptual, haciendo referencia a los principales conceptos vinculados a los catálogos en línea. Incluye un estudio del estado del arte tanto a nivel nacional como internacional. Se determinan las posibles herramientas, tecnologías y metodologías a emplear con sus respectivas fundamentaciones para un adecuado desarrollo del producto final.

Capítulo 2. “Características del sistema”: en este capítulo se plasman las descripciones de los requisitos funcionales y no funcionales que debe cumplir el sistema, además del diseño de los prototipos para la herramienta a desarrollar, donde se obtendrán las características suficientes para dar paso a la futura implementación.

Capítulo 3. “Implementación y prueba de la solución propuesta”: se efectúa la implementación del catálogo y se exponen las características de los módulos implementados, se exponen los aspectos relacionados con la fase de Pruebas, se valida la solución propuesta, la evaluación de su ejecución y los resultados obtenidos.

CAPÍTULO 1

Capítulo 1. Fundamentación teórica.

En el presente capítulo se hace referencia al marco conceptual, aludiendo a los principales conceptos vinculados con el tema propuesto. Se realiza un análisis del estado del arte de las posibles tecnologías y herramientas a utilizar para el desarrollo de la aplicación; pudiendo llegar a cercanas conclusiones sobre lo que sería más óptimo a utilizar y de esta forma dar cumplimiento a los objetivos generales de la investigación y a la solución del problema planteado.

1.1 Conceptos fundamentales asociados al dominio del problema.

Un buen proceso de desarrollo de software es eficiente cuando se logra una total sincronización entre los clientes, programadores y usuarios finales al entender lo que cada uno desea, siendo imprescindible la definición y argumentación de los principales conceptos de un trabajo investigativo.

Librería virtual o digital: recoge en un sistema una colección determinada de libros, folletos, diarios, revistas u otros ejemplares habilitándolos en la red para ser comprados o reservados. Es importante considerar que en el concepto de librería digital está presente el efecto de la integración de la informática y las comunicaciones cuyo exponente esencial es la red o Internet.

Catálogo: es una comunicación visual pero también es algo así como el punto de encuentro que tendrá la compañía con sus potenciales clientes, por eso casi siempre además del producto o servicio presentado a toda pompa visual con sus funciones, apariencia, materiales y especificaciones, el catálogo es una buena oportunidad para presentar a la empresa. El catálogo está compuesto principalmente por imágenes de los productos o servicios que se ofrecen en la empresa y que pueden ir acompañadas de breves descripciones, como por ejemplo el precio o algunos de los principales beneficios del producto.

En términos generales, según el diccionario de definiciones ABC, un catálogo es la lista ordenada o clasificada que se hará sobre cualquier tipo de objetos (monedas, bienes a la venta, documentos, entre otros) o en su defecto personas y también catálogo será aquel conjunto de publicaciones u objetos que se encuentran clasificados normalmente para la venta [2].

1.2 Sistemas existentes de catálogos de librerías.

El uso de los catálogos en línea ha tenido un inmenso auge en el mercado internacional. Esto se debe a las grandes ventajas que proporcionan a las empresas al promocionar y vender a través de Internet. Luego de investigaciones y entrevistas realizadas con el cliente, se pudo concluir que significaría un gran

CAPÍTULO 1

alivio para los trabajadores de la librería de la Universidad, la realización de un sistema que permita informatizar los procesos manuales que se realizan en la misma, además de poder transmitir a sus clientes todo lo nuevo que pueda suceder en un determinado tiempo.

A continuación, se describen algunos sistemas de catálogos de librerías a nivel internacional.

1.2.1 Sistemas internacionales.

El almacén de libros: Este sitio tiene una interfaz muy sencilla, muy fácil de navegar por él, y rápido acceso a la información. Está dividido en 3 sesiones, las cuales presenta vínculos que se muestran en todo momento en el sitio, siendo solo modificada en cada clic la sesión del centro. El usuario puede realizar búsquedas sencillas o avanzadas. El sitio presenta un inventario de todos los productos que posee, además de tener una total integración de todo el negocio en una sola aplicación.

Amazon: Es un sitio de interfaz muy agradable a la vista del usuario y de fácil navegación. Es un sitio de catálogos muy extenso con diferentes departamentos. Presenta una arquitectura de diseño muy bien estructurada, permitiendo al usuario en todo momento conocer lo que está haciendo en el sitio además de mostrar constantemente una ayuda de cómo trabajar o realizar cada acción en el catálogo en línea. Las búsquedas se pueden realizar de forma sencilla o avanzada según lo desee el usuario. A pesar de toda la información que almacena el sitio tiene una velocidad de carga bastante rápida.

1.2.2 Principales características, ventajas y desventajas de los catálogos de librerías a nivel internacional.

Características:

- Multiusuario y multiplataforma.
- Controlan a los usuarios por niveles de acceso, contraseña, por perfil de acceso, áreas autorizadas y restringidas.
- Interfaz sencilla.
- Fácil navegación.
- Menú de acceso disponible durante toda la navegación.
- Integración de todo el negocio en una sola aplicación.
- Búsqueda simple y por criterios.

CAPÍTULO 1

Ventajas:

- Colocan las especificaciones que puedan ayudar a facilitar el proceso de decisión de compra (precios, cantidad disponible, descripción).
- Ordenar los productos por categorías para que los clientes los encuentren fácilmente.
- Separar las promociones u ofertas del total de productos.

Desventajas:

Estos catálogos en líneas están muy bien pensados y muy sofisticados, pero no responden a las necesidades del cliente y expectativas de la Universidad, ya que presenta como inconveniente la venta en línea, puesto que las ventas en la UCI se realizan en el mismo local, o sea el usuario debe dirigirse a la entidad para adquirir el producto que compró. Proceso totalmente contrario a estos sitios en línea, pues son ellos quienes distribuyen los productos a sus clientes. Función principal por la que son creados.

1.2.3 Aportes de los sistemas de catálogo en línea a nivel internacional.

Estos sistemas estudiados fueron de gran ayuda pues sirvieron para definir una organización referente a las funcionalidades que debe brindar el catálogo además de una mejor estructura en el mismo.

Luego del estudio de homólogos, se continúa la investigación con el análisis de las herramientas que propuso CENIA, para que el sistema cumpla con las condiciones requeridas por la Universidad para una posterior versión actualizada del producto.

1.3 Sistema de Gestión y Contenidos (CMS).

CMS son las siglas de *Content Management System*, que se traduce directamente al español como Sistema Gestor de Contenidos. Como su propio nombre indica, es un sistema que permite gestionar contenidos. En líneas generales, un CMS permitiría administrar contenidos en un medio digital y para el caso particular, un CMS permitiría gestionar los contenidos de una web. Dicho de otra forma, un CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza una base de datos, incluyendo nueva información o editando la existente. Una herramienta CMS generalmente contendrá una interfaz basada en formularios, a los que habitualmente se accede con el navegador, donde se pueden dar de alta los contenidos fácilmente. Esos contenidos luego aparecerán en la página en los lugares donde se ha indicado al darlos de alta. Por lo tanto, un CMS estará

CAPÍTULO 1

compuesto de dos partes, un back (detrás) y un front (frente), siendo el back la parte donde los administradores publican las informaciones y el front la parte donde los visitantes visualizan las mismas [3].

Los CMS por su funcionalidad están divididos en cuatro categorías: creación de contenidos, gestión de contenidos, publicación y presentación.

Puntos más importantes que hacen útil y necesaria una aplicación CMS:

- Cambios del aspecto de la web.
- Consistencia de la web.
- Control de acceso.
- Inclusión de nuevas funcionalidades en la web.
- Mantenimiento de gran cantidad de páginas.
- Reutilización de objetos o componentes.
- Páginas interactivas.

El empleo de los CMS (Sistema de Gestión de Contenidos) se ha ido extendiendo precipitadamente debido a las grandes posibilidades que brindan. Los beneficios y peculiaridades que presentan, son extremadamente numerosos por lo que dificulta la selección del CMS más adecuado para desarrollar la implementación del catálogo de la librería UCI.

Para la selección del CMS a utilizar se tuvo en cuenta diferentes aspectos a relacionar, tales como los CMS más empleados en la Universidad y en el Centro de Informatización Universitaria CENIA.

Luego de realizar una pequeña investigación se determinó como denominador común el CMS Drupal como el gestor de contenidos más utilizados hasta el momento.

Primeramente se acoge este CMS porque el Centro de Informatización Universitaria estableció en su política la utilización del mismo en la producción de sus proyectos debido a que Drupal es un sistema robusto y potente, de código abierto y multiplataforma, que brinda a los desarrolladores una serie de funcionalidades y beneficios. En la UCI existe una comunidad con conocimiento acerca del CMS y que ha trabajado con buenos resultados, por ejemplo, la creación de los portales Intranet2, el periódico de la Universidad Mella y el portal de la Federación Estudiantil Universitario (FEU), por solo mencionar algunos. Otro argumento acerca de Drupal, es que la Universidad lo ha utilizado para el desarrollo de proyectos importantes, tanto nacionales como para los de colaboración con la República Bolivariana de Venezuela.

CAPÍTULO 1

También se han creado con el uso de este CMS las comunidades de C++ y PHP (*Hypertext Preprocessor*). Seguidamente se profundizará en las características y ventajas de esta herramienta.

1.3.1 Características de Drupal 7.

Drupal es el equivalente fonético en inglés a la palabra neerlandesa *druppel* que significa 'gota'. Drupal es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitio web. Puede usarse para aplicaciones como por ejemplo: Portales comunitarios, Foros de discusión, Sitios web corporativos, Aplicaciones de Intranet, Sitios personales o blogs, Aplicaciones de comercio electrónico, Directorio de recursos y Sitios de redes sociales.

Entre sus principales características se encuentran:

Ayuda on-line: un robusto sistema de ayuda en línea y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.

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

Código abierto: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.

Módulos: la comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como página de categorías, autenticación mediante jabber, mensajes privados, bookmarks.

Personalización: un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.

Independencia de la base de datos: Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos que

CAPÍTULO 1

actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma: 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: 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

Rendimiento y escalabilidad: 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 Caché: El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor [4].

1.4 Metodologías de desarrollo de software.

La ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [5]. Es decir, una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software.

Existen diferentes metodologías de desarrollo como las ágiles dentro de las que se encuentran Scrum, Crystal Clear, Extreme Modeling, XP (Programación Extrema), entre otras; el otro término en que se clasifican son en metodologías pesadas como son Métrica3 y RUP (Proceso Unificado de Rational).

1.4.1 Integración de las metodologías XP y SCRUM con el Proceso de Administración de Requisitos del nivel 2 de CMMI.

CMMI nivel 2.

El Modelo de Calidad CMMI (Modelo de Madurez de Capacidad Integrado) tiene como objetivo conseguir que en los proyectos de la organización haya una gestión de los requisitos y que los procesos (formas de hacer las cosas) estén planeados, ejecutados, medidos y controlados. O sea, explicando un poco más lo que se pretende se puede decir que:

CAPÍTULO 1

- El uso de los procesos al nivel dos ayuda a que la forma de trabajar se mantenga cuando hay problemas de fechas. Cuando se realizan estas prácticas, **los proyectos se ejecutan y gestionan de acuerdo con los planes de proyecto.**
- El estado de los elementos de trabajo (análisis, diseño, código, documentación,...) están visibles (estado de avance) a la gerencia en puntos definidos (hitos del proyecto). **Se sabe cuánto trabajo está hecho y cuánto queda por hacer.**
- Los compromisos adquiridos con todas las personas involucradas en el proyecto se revisan de acuerdo a las necesidades. **Los elementos de trabajo se revisan con las personas involucradas y son controlados.** Estos elementos de trabajo satisfacen las especificaciones, estándares y objetivos [6].

Scrum.

Es una metodología ágil enfocada a la gestión de proyectos. Sus principales características se pueden resumir en dos: el desarrollo de iteraciones y reuniones a lo largo del desarrollo. Las iteraciones en Scrum tienen una duración máxima de 30 días y el resultado de cada uno de ellas define un incremento del producto a desarrollar. La evolución del proyecto por la metodología se define a través de reuniones diarias donde el trabajo del día anterior es revisado por el equipo, previendo además la labor a realizar el día siguiente. Dentro de las prácticas definidas por la metodología Scrum se encuentran:

- Planificación de la iteración o sprint
- Revisión de la iteración o sprint
- Reunión diaria
- Pila del producto
- Incremento
- Propietario del producto
- Auto organización [6]

XP (Programación Extrema).

Programación Extrema es una metodología ágil de desarrollo de software que posee cuatro tareas fundamentales: planificación, diseño, desarrollo y pruebas. Esta metodología está basada en la simplicidad durante el desarrollo, la comunicación entre las partes implicadas (clientes y desarrolladores) y

CAPÍTULO 1

la retroalimentación para poder reutilizar el código desarrollado. En su concepción establece entregas frecuentes con posibilidad de refactorización continua, permitiendo mejorar el diseño cada vez que se añade una funcionalidad. Para su implementación XP establece un conjunto de prácticas que deben ser empleadas en los proyectos de desarrollo, las mismas se mencionan a continuación:

- El juego de la planificación
- Entregas pequeñas
- Metáfora
- Diseño simple
- Pruebas
- Refactorización (*Refactoring*)
- Programación en parejas
- Propiedad colectiva del código
- Integración continua
- 40 horas por semana
- Cliente in-situ
- Estándares de programación [7].

1.5 Lenguajes de programación utilizados.

1.5.1 Lenguajes de marcas.

Aquellas aplicaciones que les permite a los usuarios acceder a un servidor web a través de una Intranet o Internet mediante un navegador para adquirir información de forma interactiva, se les nombra aplicación web. Dicha aplicación se codifica en un lenguaje soportado por los navegadores web como son: HTML, Java Script, Java, XML (Lenguaje de Marcas Extensible), en la que se confía la ejecución al navegador.

HTML (Lenguaje de Marca de Hipertexto).

Es un lenguaje muy sencillo que posibilita describir hipertexto. Es decir, texto presentado de forma estructurada y agradable, con enlaces o vínculos que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos y sonido). Una de las características

CAPÍTULO 1

esenciales de este lenguaje es la universalidad, y significa que prácticamente cualquier ordenador, independientemente del sistema operativo, puede leer o interpretar una página web.

1.5.2 Lenguajes del lado del cliente.

Java Script.

Lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es orientado a eventos con manejo de objetos, multiplataforma, con una sintaxis semejante al lenguaje Java, cuyo código se incluye directamente en el mismo documento HTML. Se usa para la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

1.5.3 Lenguaje del lado del servidor.

PHP v5.4.3.

El PHP (*Hypertext Preprocessor*), se define como un lenguaje de programación para la creación rápida de contenidos dinámicos de sitios web, como son los foros, blogs, sistemas de noticias, entre otros. También, crea aplicaciones gráficas independientes del navegador y aplicaciones para servidores. Es un lenguaje de script dentro del HTML. La principal función del PHP es permitir la interacción de la página web con el visitante que pudo haber realizado cambios en ella, y cada usuario que ingrese a la página podrá ver la información anteriormente modificada [12].

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC (*Open DataBase Connectivity*), Microsoft SQL Server.

El Centro de Informatización Universitaria propone la utilización de PHP siendo este un lenguaje que se integra a los CMS, completamente orientado a la web, se basa en la POO (Programación Orientada a Objetos), es una alternativa de fácil acceso para todos y posee característica que se adecuan a las necesidades de este proyecto. PHP es un lenguaje implícito en el funcionamiento del CMS Drupal.

CAPÍTULO 1

1.6 Herramientas utilizadas para el desarrollo del sistema.

1.6.1 Sistema gestor de base de datos.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Entre los Sistema Gestores de Base de Datos se encuentran Oracle, DB2, PostgreSQL, MySQL, MS SQL Server.

Para la realización de esta aplicación web CENIA dicta utilizar PostgreSQL por la fácil integración con el CMS Drupal.

1.6.2 PostgreSQL v 9.1.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (Berkeley Software Distribution). Está considerado como la base de datos de código abierto más avanzada del mundo.

PostgreSQL ofrece diversas ventajas como son:

- El código fuente está disponible para todos sin costo, lo que lo hace extensible.

Usa una estrategia de almacenamiento de filas llamada MVCC (Control de Concurrencia para Múltiples Versiones de PostgreSQL)

- para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- Está disponible en casi cualquier sistema Unix.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Posee una amplia variedad de tipos nativos.

CAPÍTULO 1

- Tiene mejor soporte para *triggers* y procedimientos en el servidor.
- Altamente adaptable a las necesidades del cliente.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos.
- Probado con torres de discos duros de alta disponibilidad [13].

1.6.3 Servidor web.

Un servidor web es un programa que sirve datos en forma de páginas web, hipertextos o páginas HTML 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 (HiperText Transfer Protocol). Con esto, un servidor web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que se conoce como un navegador web [14].

1.6.4 Apache v2.2.21.

Características.

Es un servidor web de software libre desarrollado por la *Apache Software Foundation* cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan. La arquitectura utilizada es cliente/servidor, es decir, el equipo cliente hace una solicitud o petición al equipo servidor y éste la atiende.

En el equipo cliente se ejecuta una aplicación llamada navegador o cliente web que:

- Sirve de interfaz con el usuario: atiende sus peticiones, muestra los resultados de las consultas y proporciona al usuario un conjunto de herramientas que facilitan su comunicación con el servidor.
- Se comunica con el servidor web: transmite las peticiones de los usuarios.

El protocolo utilizado para la transferencia de hipertexto es HTTP que está basado en el envío de mensajes y establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una web y la respuesta de esa web.

CAPÍTULO 1

HTTP es un protocolo sin estado, es decir, no recuerda nada relativo a conexiones anteriores a la actual. La conexión sólo tiene la duración correspondiente a la transmisión de la página solicitada si la encuentra, y si no la encuentra devuelve un código de error [15].

Principales ventajas.

- Código abierto.
- Multiplataforma.
- Extensible.
- Popular (fácil conseguir ayuda/soporte).

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. El Centro de Informatización Universitaria decide seleccionar Apache como servidor además de sus ventajas por ser éste 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.

1.6.5 Visual Paradigm v8.0.

¿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 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.

La estructura de las herramientas Case se basa en la siguiente terminología:

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.

CAPÍTULO 1

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 [16].

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, demostraciones interactivas y proyectos de UML [17].

1.6.6 NetBeans IDE v7.2.

NetBeans es un entorno de desarrollo integrado de código libre licenciado bajo la *Common Development and Distribution License* (CDDL) y de distribución gratuita que proporciona herramientas muy cómodas y de fácil uso para el desarrollo de aplicaciones sobre la plataforma Java.

Entre las características que proporciona están:

Desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux.

Desarrollo de aplicaciones para Móvil, Web gráfico, integración con SOA (Arquitectura orientada a servicios), optimización de aplicaciones y desarrollo con C y C++.

Ciente CVS integrado.

Crecimiento de plataforma por medio de plugins.

1.7 Conclusiones parciales.

Después de realizar un estudio de los diferentes catálogos en línea que existen a nivel internacional, se puede concluir que, fueron de gran ayuda para tener una mejor conceptualización y dominio del tema, además de aportar mejores ideas de cómo organizar y mostrar la información en la aplicación que se desea implementar. Para un mejor desarrollo de la misma se empleará el CMS Drupal por las grandes posibilidades que brinda al gestionar contenidos, vistas, búsquedas y otros elementos muy necesarios en

CAPÍTULO 1

el desarrollo de la aplicación para la librería de la Universidad. Como lenguaje del lado del servidor se empleará PHP en su versión 5.4.3, por ser compatible e integrado con el CMS Drupal y como gestor de base de datos a PostgreSQL en su versión 9.1. Se seleccionó como servidor web el Apache v2.2.22, por ser uno de los más usados en la Universidad, como herramienta de modelado se empleará a Visual Paradigm v8.0 y para apoyar el proceso de implementación de software se utilizará el IDE NetBeans en su versión 7.2 con la ventaja de lograr un desarrollo dinámico junto al cliente y una rápida culminación del producto.

CAPÍTULO 2

Capítulo 2. Características del Sistema.

En el presente capítulo se expone un análisis de la propuesta del sistema aplicando la metodología ágil Scrum + XP y el modelo de calidad CMMI nivel 2 donde se especifican los requisitos funcionales y no funcionales, se describen textualmente las historias de usuario, el diagrama de dominio y de despliegue, la propuesta de solución y los beneficios que aporta el sistema quedan plasmados.

2.1 Propuesta de solución.

El sistema surge por la necesidad de informatizar la gestión de ventas que se realiza en la librería de la Universidad de las Ciencias Informáticas. Su principal objetivo, ayudar en la gestión del proceso de venta y promoción de los productos de la entidad. La comunidad universitaria podrá obtener notificaciones acerca de las ofertas que brinda esta entidad y acceder en cualquier momento del día a la librería de manera virtual sin tener que trasladarse obligatoriamente a la entidad.

El sistema será desarrollado sobre el núcleo básico, de la versión 7.18 del Sistema Gestor de Contenido Drupal, con el propósito de reutilizar muchas de las ventajas que dicho CMS brinda, además de ser lo suficientemente fácil de modificar y extender.

2.2 Descripción de la propuesta de solución.

El sistema permitirá 3 usuarios diferentes: administrador, autenticado, anónimo.

Administrador: destinado a aquellos usuarios que han sido señalados como administradores del sistema. A través de este rol se podrán configurar las distintas opciones relacionadas con los pedidos de compras, las gestiones de ventas y la actualización de los productos (precio, existencia).

Autenticado: destinado a los usuarios autenticados en el sistema. Los mismos tendrán la posibilidad de navegar por todo el sistema, además de contar con su propio perfil del sitio. Podrán realizar búsquedas en el catálogo, simple o avanzada, realizar pedidos compras, votar, comentar sobre un productos entre otros.

Anónimo: ideado para aquellos usuarios que visitan el sitio solo para ver los productos que se ofrecen sin necesidad de realizar ningún pedido.

La aplicación contará con un catálogo de todos los productos existentes en el inventario de la librería. El mismo mostrará una descripción de cada producto, tales como, imagen, resumen, existencia, precio, género y editorial. Permitirá a los trabajadores de la librería listar los pedidos de ventas por sus usuarios y

CAPÍTULO 2

poder modificar el estado en Aceptado, Rechazado o Pendiente, además de poder exportar a un documento Excel. Para ello contará con una base de datos que almacenará todos los datos necesarios.

La aplicación cuenta con un servicio de notificación el cual es enviado al usuario luego de haber realizado un pedido. En el mismo se indica su número de pedido (identificador), los productos que reservó, el monto a pagar y la fecha límite para pasar por el local.

2.3 Especificación de los requisitos del software.

2.3.1 Lista de Reserva del Producto.

A continuación se realizan diversas actividades, tales como el levantamiento de requisitos y elaboración de la Lista de Reserva del Producto (LRP), esta es una lista priorizada que define el trabajo que se va a realizar en el proyecto.

Dicha lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. La LRP presenta los requisitos funcionalidades que debe cumplir el sistema en un orden de prioridad (alta, media y baja). También aparecen los requisitos no funcionales que debe cumplir el sistema.

Código	Descripción del requisito funcional	Prioridad
RFMC01	Autenticar usuario	Alta
RFMC02	Realizar búsqueda simple	Alta
RFMC03	Realizar búsqueda avanzada	Alta
RFMC04	Mostrar resultado de búsqueda	Alta
RFMC05	Realizar pedido	Alta
RFMC06	Listar pedidos	Alta
RFMC07	Cambiar estado de pedido	Alta
RFMC08	Mostrar carrito de compras	Alta
RFMC09	Añadir productos	Alta
RFMC10	Enviar correo de pedido de compra	Alta
RFMC11	Enviar correo de pedido rechazado	Alta
RFMC12	Insertar literatura	Alta
RFMC13	Editar literatura	Alta

CAPÍTULO 2

RFMC14	Eliminar literatura	Alta
RFMC15	Insertar tipo de literatura	Alta
RFMC16	Editar tipo de literatura	Alta
RFMC17	Eliminar tipo de literatura	Alta
RFMC18	Generar reportes	Alta
RFMC19	Mostrar los 5 producto más vendido	Media
RFMC20	Mostrar los 5 producto más votado	Media
RFMC21	Mostrar los 5 nuevos productos	Media
RFMC22	Comentar producto	Media
RFMC23	Votar por un producto	Media
RFMC24	Mostrar quienes somos	Media
RFMC25	Eliminar pedidos	Media
RFMC26	Eliminar producto	Media
RFMC27	Cancelar pedido	Baja

2.3.2 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto, por lo que se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

- Autenticación
 - Autenticar usuario
- Búsquedas
 - Realizar búsqueda simple
 - Realizar búsqueda avanzada
 - Mostrar resultado de búsqueda
- Pedidos de compras
 - Realizar pedido
 - Cancelar pedido
 - Listar pedidos

CAPÍTULO 2

- Eliminar pedidos
- Cambiar estado de pedido
- Carrito de compras
 - Mostrar carrito de compras
 - Añadir productos
 - Eliminar productos
- Notificación
 - Enviar correo de pedido de compra
 - Enviar correo de pedido rechazado
- Preferencias
 - Mostrar los 5 producto más vendido
 - Mostrar los 5 producto más votado
 - Mostrar los 5 nuevos productos
 - Comentar producto
 - Votar por un producto
 - Mostrar quienes somos
- Reportes
 - Generar reportes
- Gestionar literatura
 - Insertar literatura
 - Editar literatura
 - Eliminar literatura
 - Insertar tipo de literatura
 - Editar tipo de literatura
 - Eliminar tipo de literatura

CAPÍTULO 2

2.3.3 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a los requisitos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener y cuán rápido debe ser. Por tanto, la aplicación debe cumplir con las siguientes propiedades:

Usabilidad:

1. Los administradores de la aplicación deben tener un control total sobre la misma, para poder dar solución a cada incidente ocurrido.
2. La interfaz del sistema debe ser agradable, que permita a los usuarios sin experiencia una rápida adaptación.
3. Debe permitir que, cuando sea conveniente, existan valores por defecto persistentes para la introducción de datos, entre ellos valores derivados del contexto.
4. Debe tener buena visibilidad en los principales navegadores web como Mozilla Firefox a partir de su versión 13.0 e Internet Explorer v8.0.

Fiabilidad:

5. El sistema debe ser multiplataforma.
6. El sistema debe estar bajo la modalidad de 8 horas laborables del año.

Eficiencia:

7. El sistema debe ser capaz de responder a cualquier petición antes de los 25 segundos.
8. El sistema debe requerir un consumo mínimo de recursos.
9. El sistema debe soportar una conexión simultánea de 2000 usuarios.

Restricciones del diseño:

10. El lenguaje de programación es PHP 5.4.3.
11. El marco de trabajo base de desarrollo que se utiliza es Drupal 7.18.
12. La herramienta IDE de desarrollo utilizada es NetBeans 7.2.
13. El Sistema Gestor de Base de Datos es PostgreSQL 9.1.
14. Usar la herramienta de modelado Visual Paradigm 8.0.

CAPÍTULO 2

Hardware:

15. Para el aprovechamiento del servidor se necesita un microprocesador Intel Core i3 con velocidad de 3GHZ, memoria RAM de 2 GB y una capacidad de almacenamiento de 250 GB.
16. Para el desarrollo del sistema se necesita un microprocesador Pentium IV con una velocidad de 3.8GHZ, memoria RAM 1 GB y una capacidad de almacenamiento de 160 GB.
17. Para el usuario con un navegador web será suficiente.
18. Debe existir una red de área local para la divulgación de la información.

Interfaz:

19. La interfaz debe ser sencilla y amigable de manera que potencie la comodidad del usuario para su trabajo.
20. El sistema estará optimizado para una resolución de 1024x768.
21. El diseño gráfico será acorde con las pautas de diseño de la Universidad.
22. La interfaz debe mantener el mismo formato en todas las páginas.
23. El sistema debe ser independiente del sistema operativo y ser completamente funcional en los principales sistemas operativos existentes en el mercado: Sistema Operativo Microsoft Windows 2000 o superior; Sistema Operativo Linux.

Interfaces de comunicación:

24. El sistema deberá consumir Servicios Web para autenticar usuario y la integración con sistemas externos.

Requisitos legales, de derecho de autor y otro:

25. El CMS escogido como plataforma para el desarrollo de la aplicación está basado en la licencia GNU/GPL.

2.4 Historias de Usuarios.

Utilizando la integración de las metodologías Scrum y XP con el modelo CMMI se analizó y diseñó el sistema, generando como artefacto principal las Historias de Usuario (HU) que son entregadas al programador para la implementación.

Las HU representan una breve descripción del comportamiento del sistema. Estas deben proporcionar solo el detalle suficiente como para poder hacer razonable la estimación de cuánto tiempo requiere la implementación de la historia. Difiere de los casos de uso, porque son escritos por el cliente, empleando

CAPÍTULO 2

terminología del cliente. Cada HU cuenta con una breve descripción de alguna funcionalidad en específico que debe cumplir el sistema y las interfaces correspondientes para la solución de las mismas. Para el desarrollo del Catálogo Librería UCI se definieron 27 HU.

2.4.1 Descripción de las historias de usuarios.

Las Historias de Usuarios (HU) son artefactos narrativos que describen el comportamiento del sistema desde el punto de vista del cliente. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y requisitos que debe cumplir el sistema. A continuación se describen 5 de las HU del sistema a desarrollar con prioridad alta.

Tabla 1 HU Autenticarse.

Número: RFMC01		Nombre del requisito: Autenticar usuario	
Programador: Yoanna González Aymerich		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 1	
Riesgo en Desarrollo: Medio		Tiempo Real: 1	
Descripción: La HU comienza cuando el usuario accede al sistema y se autentica de manera correcta introduciendo el usuario y la contraseña en los campos correspondientes.			
Observaciones:			
Prototipo de interfaz:			

CAPÍTULO 2



The screenshot shows a login form with the following elements:

- Title: Autenticarse
- Section: Inicio de sesión
- Field: Nombre de usuario * (containing 'Usuario UCI')
- Field: Contraseña *
- Button: Iniciar sesión

Tabla 2 HU Realizar búsqueda simple.

Requisito	
Número: RFMC02	Nombre del requisito: Realizar búsqueda simple
Programador: Yoanna González Aymerich	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1
Riesgo en Desarrollo: Alta	Tiempo Real: 1
Descripción: La HU inicia cuando el usuario accede al menú de catálogo. El usuario puede realizar la búsqueda por orden alfabético o por título.	
Observaciones: el usuario tiene que estar autenticado en el sistema.	
Prototipo de interfaz:	
 <p>The screenshot shows a search interface with the following elements:</p> <ul style="list-style-type: none">Title: Catálogo de literaturaSection: Búsqueda simpleInput field: Escriba nombre del autorButton: AplicarNavigation: A B D E H L M R T	

CAPÍTULO 2

Tabla 3 HU Realizar búsqueda avanzada.

Número: RFMC03		Nombre del requisito: Realizar búsqueda avanzada	
Programador: Yoanna González Aymerich		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 1	
Riesgo en Desarrollo: Alta		Tiempo Real: 1	
Descripción: La HU comienza cuando el usuario accede al sistema y en el menú selecciona Búsqueda. El sistema muestra los campos a llenar, ellos son: Título, género, editorial, año de edición, autor. Una vez llenado los campos el usuario debe dar clic en el botón Buscar			
Observaciones: el usuario debe estar autenticado en el sistema.			
Prototipo de interfaz:			
			

Tabla 4 HU Mostrar resultado de búsqueda

Número: RFMC04		Nombre del requisito: Mostrar resultado de búsqueda	
Programador: Yoanna González Aymerich		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 1	
Riesgo en Desarrollo: Medio		Tiempo Real: 1	
Descripción: La HU comienza cuando el usuario da clic en el botón Buscar o selecciona una letra del catálogo. Seguidamente el sistema muestra todos los productos que coincidan con la			

CAPÍTULO 2

búsqueda especificada. El sistema muestra los siguientes datos: imagen del producto, título, resumen, autor, código, existencia, precio y un botón de Agregar al carrito.

Observaciones:

Prototipo de interfaz:

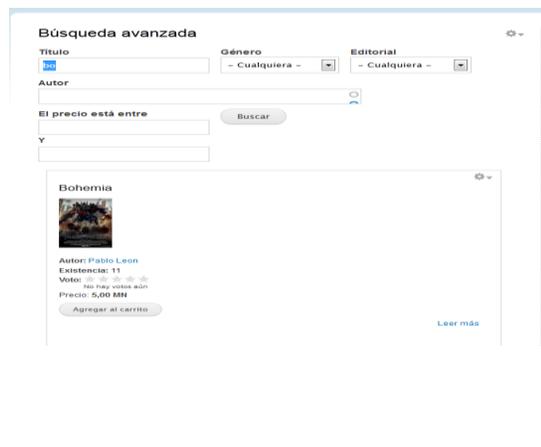


Tabla 5 HU Realizar pedido

Número: RFMC05		Nombre del requisito: Realizar pedido	
Programador: Yoanna González Aymerich		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 1	
Riesgo en Desarrollo: Medio		Tiempo Real: 1	
Descripción: La HU comienza una vez que el usuario añade productos a su carrito de compras y presione el botón Ver carrito. El sistema le muestra los productos a comprar y el botón Realizar pedido. Luego de presionado el sistema muestra otro botón de Enviar pedido. El sistema envía un correo de notificación al usuario.			
Observaciones:			
Prototipo de interfaz:			

CAPÍTULO 2



2.5 Modelo de dominio.

El Modelo de dominio es una representación visual estática del entorno real objeto del proyecto, es decir, un diagrama con los objetos reales que existen relacionados con el sistema que se desarrolla y las relaciones referente a la multiplicidad que hay entre ellos. Ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

Seguidamente se explica cada una de las clases que se muestran en el modelo:

Librería UCI: es la entidad de la UCI en la que se realiza el trabajo de forma manual

Inventario: donde se guarda la información referente a todas las operaciones que realiza la librería, dichas operaciones se recogen en un excel.

Ficha técnica: ficha donde se guardan los datos de los libros, folletos y las revistas, estos son: autor, editorial, título, año de edición, código, precio de costo, precio de venta, cantidad específica de un mismo tipo.

Ejemplares: son los libros, las revistas y los folletos.

Revistas, Folletos y Libros: ejemplares que se encuentran en la librería.

A continuación queda plasmado el modelo de dominio referente al sistema donde se muestran las clases mencionadas anteriormente y la relación que se establece entre las mismas mostrando en una imagen como funciona el trabajo en la entidad:

CAPÍTULO 2

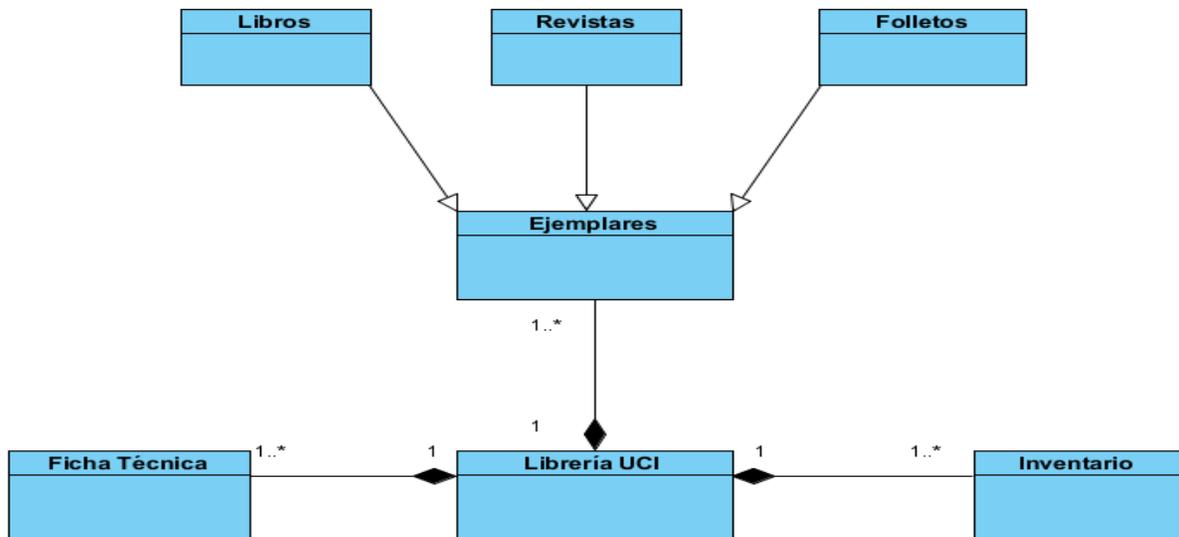


Figura 1: Modelo de dominio.

2.6 Diagrama de despliegue.

El diagrama de despliegue describe los componentes, dispositivos, hardware y software que se desplegará en el ambiente seleccionado. Se propone que el sistema propuesto quede distribuido de la siguiente forma:

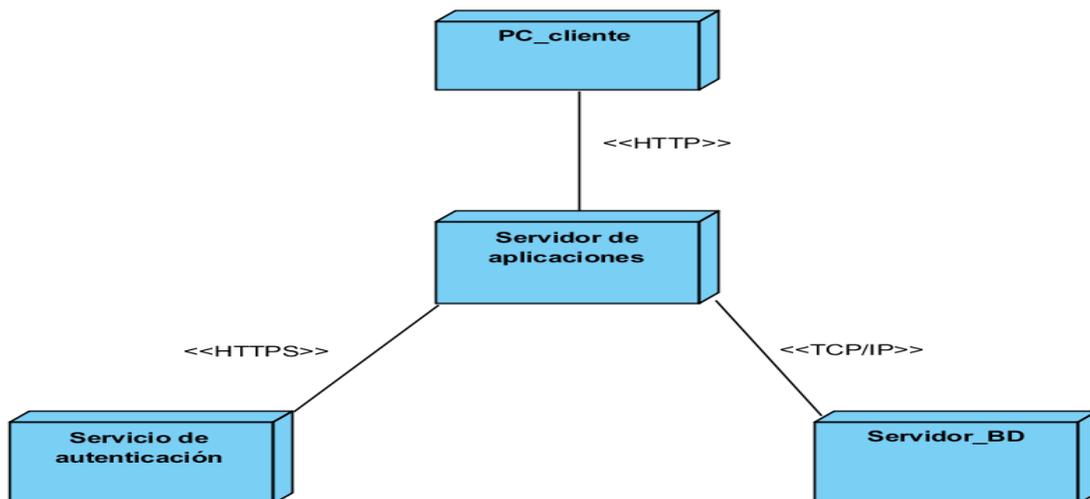


Figura: 2 Diagrama de despliegue.

CAPÍTULO 2

- **PC-Cliente:** representa el conjunto de computadoras donde los usuarios pueden acceder al servidor web mediante cualquier navegador web y consultar la información que brinda el sistema.
- **Servidor Web:** se encuentra el sistema de Catálogo de Librería UCI.
- **Servidor de Base de Datos:** representa el servidor PostgreSQL, donde se encuentra ubicada toda la información persistente del sistema.
- El nodo **pasarela de servicios** representa un servidor web que actúa como interfaz única de acceso a los servidores de dominio de la Universidad.
- El servidor de dominio almacena todos los datos de los usuarios del Directorio Activo de la UCI.

2.7 Beneficios del sistema.

El sistema que se propone tiene múltiples beneficios quizás el más importante sea la informatización de todo el proceso de venta de la librería de la Universidad de las Ciencias Informáticas haciéndole el trabajo más fácil y agradable al personal que allí laboran. Por medio de esta aplicación se le dará a conocer a todos los usuarios de la Universidad el movimiento que se realiza en la librería y las nuevas opciones literarias que se brindan en la misma. La entidad contará con una capacidad de almacenamiento superior para los artículos, una mayor organización de los ejemplares, más eficiencia y rapidez a la hora de gestionar la venta o búsqueda de los productos y aumentará la expectativa de los trabajadores en su labor.

2.8 Conclusiones parciales.

Como resultado de la realización de este capítulo se describieron los requisitos funcionales y no funcionales como proceso esencial para el desarrollo del sistema, dando así una visión del futuro de la aplicación, se realizó las descripciones de las historias de usuarios, donde se identificó la prioridad de cada una para la realización de estas. De esta forma quedan planteadas, las condiciones y características del sistema propuesto y marcando la línea a seguir para la realización del análisis y el diseño de la aplicación.

CAPÍTULO 3

Capítulo 3. Implementación y prueba de la solución propuesta.

Luego de obtener las bases del software a través del análisis correspondiente, se realizó la fase de implementación la cual se rige estrictamente por los resultados y artefactos obtenidos. A dicho sistema se les realizó varias pruebas para así ratificar y comprobar la calidad del mismo en cuanto al código generado y su funcionamiento.

3.1 Arquitectura del sistema.

La arquitectura del CMS Drupal es totalmente modular, ya que incluye un conjunto de herramientas y pautas que se deben seguir para desarrollar e integrar nuevas funcionalidades a través de módulos adicionales. En última instancia un módulo consiste en un conjunto de archivos con código PHP, que utiliza la arquitectura y las APIs de Drupal para incorporar nuevas características funcionales al sitio web. A continuación se describen algunos elementos que componen la arquitectura de Drupal.

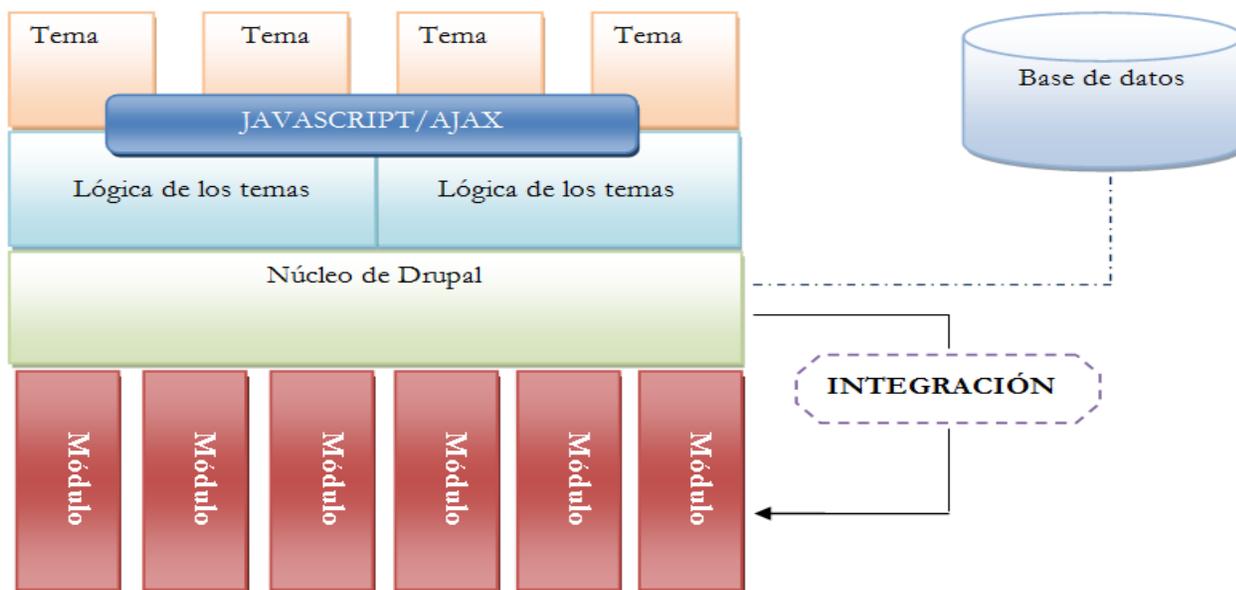


Figura 3: Esquema de los elementos que conforman un sistema Drupal.

Los módulos del núcleo de Drupal:

Estos módulos constituyen el núcleo de Drupal y forman parte de la distribución básica del entorno. Tanto los módulos que constituyen el núcleo de Drupal como cualquier otro módulo desarrollado por la comunidad, siguen idénticas directrices en su desarrollo, haciendo uso de las mismas APIs.

CAPÍTULO 3

Entidades y campos:

Separa los campos del tipo de contenido y genera el concepto de entidad. En Drupal 7 una entidad es un elemento más genérico, al que se le pueden añadir campos. Ahora los tipos de contenido (o nodos), los usuarios, los términos de taxonomía y otros elementos son definidos como entidades, y como tales se podrán gestionar los campos asociados y su presentación.

Los nodos en Drupal:

Drupal almacena los contenidos de una página Web en nodos. Los nodos constituyen por tanto los elementos básicos de información en un sitio Web con Drupal. Aunque lo habitual es que los nodos estén constituidos por contenido de tipo texto (o HTML), Drupal puede manejar tipos de nodos cuyos elementos fundamentales no sean sólo texto sino ficheros de audio, video. Los nodos se almacenan en la base de datos, siendo la tabla *node* la encargada de almacenar la información básica.

Los comentarios en Drupal:

En Drupal 7 los comentarios también son entidades, por lo que se podrán gestionar sus campos y presentaciones como se harán con el resto de las entidades. La tabla *comment* almacena la información relativa a los comentarios. Los comentarios son un ejemplo de la flexibilidad que aporta Drupal con el nuevo concepto de entidad.

Los usuarios en Drupal:

Otro tipo de elemento importante son los usuarios. La información relativa a los usuarios se encuentra almacenada en la base de datos y se utiliza en diversos procesos: autenticación, preferencias y permisos. El uso de la *user API* permite obtener información de los usuarios, comprobar sus permisos y obtener sus preferencias.

Los formularios en Drupal:

El mecanismo principal de enviar contenidos a través de la web son los formularios HTML. A través de los *Forms API* (también conocida como FAPI) se ofrecen unas potentes herramientas de desarrollo para la generación de formularios web, de manera que a partir de la definición del formulario, Drupal se encarga de gestionar su construcción, visualización, captura de datos y validación [17].

CAPÍTULO 3

3.2 Patrones de diseño.

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de un diseño estructurado común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Definen una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones.

Los patrones de diseño tienen como propósito:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores. Estandarizar el modo en que se realiza el diseño.

Facilitar el aprendizaje de las nuevas generaciones de diseñadores, condensando conocimiento ya existente [18].

Los patrones GOF (*Gang-Of-Four*) son utilizados en el diseño de la aplicación y están relacionados por su funcionalidad, entre los que se destacan los creacionales, los estructurales y los de comportamiento. Para el diseño del Catálogo Librería UCI se tuvo en cuenta 3 patrones: Observador, Fachada, Cadena de Responsabilidad.

El patrón Observador (*Observer*): Es del tipo comportamiento y define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifiquen y actualicen automáticamente todos los objetos que dependen de él. Este patrón se evidencia en el código de la aplicación a través del uso de los ganchos (*hook*). Al implementar un gancho los módulos se registran como observadores del objeto, las modificaciones que se realicen en el mismo, serán notificadas en los demás módulos que dependan de él.

El patrón Fachada (*Facade*): Incluido en los patrones estructurales, es utilizado para acceder a una interfaz o grupo de interfaces de un subsistema mediante una interfaz unificada; este patrón se evidencia mediante el acceso al servicio web del Directorio Activo de la UCI.

El patrón Puente (*Jumper*): Es de tipo estructural y se aplica de una forma similar al patrón de diseño puente. Los módulos necesitan ser escritos en una forma que es independiente del sistema que se está utilizando en la base de datos, y para esto proporciona la capa de abstracción. La nueva capa puede

CAPÍTULO 3

escribirse conforme a la API definida por el puente, añadiendo soporte para más sistemas de bases de datos sin la necesidad de modificar el código del módulo.

El patrón Cadena de responsabilidades (*Chain of Responsibility*): El sistema de menús de Drupal sigue este patrón. En cada solicitud de la página, el menú determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, se llama la función para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud, si el elemento de menú no puede manejar la petición, se pasa de la cadena. Esto continúa hasta que un módulo se encarga de la petición, se niega el acceso para el usuario o la cadena se agota.

3.3 Estándares de codificación.

Un estándar de codificación constituye un modelo a seguir donde se le da una estructura específica al código, es además una guía probada y efectiva para los programadores. Para que el estándar se considere completo debe comprender todos los aspectos de la generación de código y su puesta en práctica por todos los programadores de un software logrará mantener una correcta legibilidad y mantenimiento del código.

Para la presente investigación son de vital interés los estándares de codificación para base de datos y para el lenguaje de programación PHP.

3.3.1 Estándares de codificación de la base de datos.

Durante el ciclo de vida de desarrollo de un software es necesario asegurar la calidad del mismo. Uno de los instrumentos que facilitan esta tarea es la adopción de estándares de diseño. Su uso trae consigo cientos de ventajas en el diseño y desarrollo de la base de datos, entre ellas se encuentran las siguientes:

- Asegurar la legibilidad del modelo de datos, inclusive para personas que no están familiarizadas con el ambiente informático.
- Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones.
- Facilitar la tarea de los programadores en el desarrollo de sistemas.

Por ende, es necesario que la codificación de las tablas de la base de datos deba cumplir con los requisitos expuestos en el siguiente estándar.

1. Los nombres de las tablas deben introducirse entre llaves: {nombre tabla}.

CAPÍTULO 3

2. Si el nombre de las tablas se compone de varias palabras deben separarse por un guión bajo y escribirse en minúscula.
3. Los atributos se escriben con minúscula, de forma sugerente procurando que transmitan una idea lo más cercana posible a la descripción de la característica real del objeto. Ejemplo: título, autor, editorial.

3.3.2 Estándares de codificación utilizados en la programación.

Establecer un estándar de codificación en la programación es muy importante para lograr hacer los programas más entendibles y fáciles de leer. Con el cumplimiento estricto de cualquier estándar de codificación que se establezca se pueden dar información sobre la función de cualquier identificador que se use, por ejemplo, cuando es una constante, un paquete, o una clase, que puede ser útil para entender el código o corregir algún error que se presente durante el proceso de implementación.

Indentación:

Se debe indentar con 2 espacios, nunca con tabuladores. No se debe dejar espacios en blanco al final de cada línea.

Etiquetas de apertura y cierre de PHP:

Siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<? y ?>`.

Operadores:

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

Uso de comillas:

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

Estructura de control:

Debe haber un espacio entre el comando que define la estructura (`if`, `while`, `for`) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones. La llave de apertura `{` se situará en la misma línea que la definición de la estructura, separada por un espacio.

CAPÍTULO 3

Se recomienda usar siempre las llaves {} aún en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).

Las estructuras else y elseif se escribirán en la línea siguiente al cierre de la sentencia anterior.

Funciones:

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guión bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

Ejemplo: `function basic_cart_menu (arg1, arg2)`

Arreglos (array):

Los valores dentro de un array (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados.

Constantes:

Los nombres de las constantes deben escribirse en mayúsculas, con guiones bajos para separar palabras. Los nombres de las constantes deben tener como prefijo el nombre del módulo (o tema) en el que se utilizan. Este prefijo también se escribirá en mayúsculas.

Comentarios de documentación:

Utilizar las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea.

Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto.

Comentarios:

El estilo de comentarios que se aplicará debe ser (`/* */` ó `//`). Deben estar elaborados con un lenguaje formal sin faltas de ortografía y con carácter explicativo.

Definiciones de función:

En las declaraciones de función, los argumentos con valores por defecto van al final de la lista de argumentos. La apertura de la llave debe encontrarse luego del cierre del paréntesis, el cuerpo de la función se encuentra en una línea posterior.

Llamadas de función:

CAPÍTULO 3

Las funciones se llaman sin espacios entre el nombre de la función, la apertura del paréntesis y el primer parámetro y sin espacios entre el último parámetro y el cierre del paréntesis, seguido de un punto y coma. Los parámetros se separan por coma y sin espacios entre ellos.

Convención de Nombres de Variables, Funciones y Clases:

Tanto las variables como las funciones deben nombrarse de forma sencilla y descriptiva.

3.4 Diagrama de componentes.

Es una representación de cómo se compone el sistema, muestra la organización y las relaciones de dependencias entre un conjunto de componentes. Define las estructuras de datos, los algoritmos, las características de la interfaz y los mecanismos de comunicación asignados a cada componente del software [19].

El sistema se desarrolla sobre las bases de Drupal por lo tanto su diagrama de componentes está relacionado con el CMS.

Descripción del diagrama de componentes:

- **Index:** Este componente funciona como un controlador, canaliza todo y es el encargado de realizar las llamadas a los módulos del CMS.
- **Themes:** Almacena las plantillas o temas que serán utilizadas para la interfaz de Drupal.
- **Modules:** Incluye los módulos del núcleo que vienen por defecto con el CMS, para su inicial funcionamiento.
- **Include:** Contiene la infraestructura del núcleo, donde se sitúan las librerías de las funciones comunes y necesarias para el levantamiento del gestor.
- **Scripts:** Contiene scripts para tareas como: verificar sintaxis, limpiar el código o ejecutar Drupal desde la línea de comandos manejando casos especiales como el cron o test suites.
- **Profiles:** Consta de diferentes perfiles de instalación para un sitio, con el objetivo de que los perfiles habiliten automáticamente ciertos módulos del núcleo o contribuidos.
- **Misc:** Almacena los ficheros javascript y varios iconos/imágenes disponibles para usarlos en la instalación de Drupal.

CAPÍTULO 3

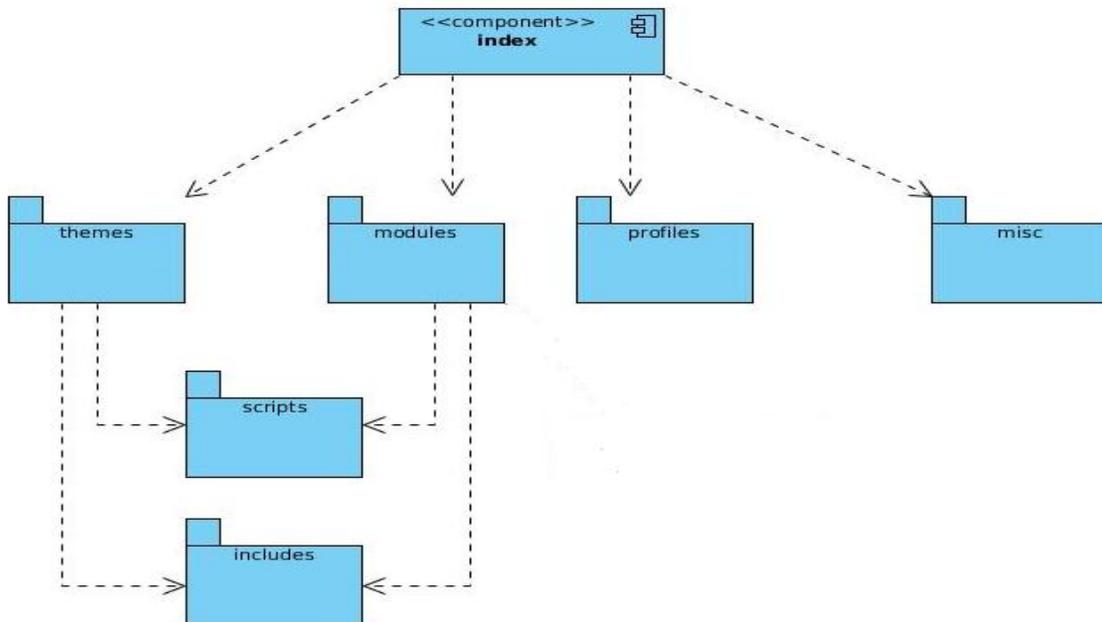


Figura 4: Diagrama de componentes del sistema Librería UCI.

3.5 Modelo de prueba.

El proceso de realización de pruebas al software constituye la etapa donde se evalúa si el producto cumple con las especificidades y requisitos definidos en sus inicios, si es de gran aceptación para el cliente y si puede optimizarse la solución propuesta. Se debe recalcar que esta validación enmarcada en la calidad del software debe realizarse durante todo el ciclo de desarrollo de la aplicación si se desea lograr excelentes resultados.

Existen diferentes niveles de pruebas los cuales enmarcan un objetivo específico a alcanzar con su aplicación estos niveles están dados por:

- Prueba de desarrollador.
- Prueba unitaria.
- Prueba independiente.
- Prueba de integración.
- Prueba de sistema.
- Prueba de aceptación.

CAPÍTULO 3

3.5.1 Pruebas realizadas al sistema.

En la presente investigación se aplicó el nivel correspondiente a las pruebas de aceptación con el objetivo de encontrar errores en el código y en el cumplimiento de las funciones del software y que pueda ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue concebido en sus inicios; dentro de este nivel se realizaron pruebas para validar tanto las funciones como los métodos, los servicios y requisitos presentes en el sistema. Para la puesta en práctica de dichas pruebas se empleó el método de Caja negra el cual contempla las pruebas que se aplican a la interfaz del software.

Las pruebas de aceptación son también pruebas de caja negra que se crean a partir de las historias de usuario. En el proceso de iteraciones las historias de usuarios seleccionadas serán traducidas a pruebas de aceptación. En ellas se señalan, desde la perspectiva que tiene el cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente.

3.5.2 Pruebas de caja negra.

Es el elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. De ellas se obtienen su forma de interactuar con el medio que les rodea, entendiendo qué es lo que hacen, pero sin dar importancia a cómo lo hacen. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, o sea, su interfaz.

Las pruebas de Caja negra intentan encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz [20].

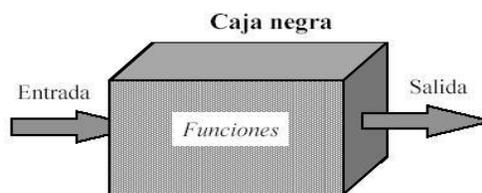


Figura 5: Modelo de caja negra

CAPÍTULO 3

A través de esta técnica, los casos de prueba desarrollados en el presente trabajo pretenden demostrar que las funciones del Catálogo Librería UCI son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

3.5.3 Resultados de las pruebas de caja negra.

Dentro del plan de pruebas definido para las validaciones al producto se decidió realizar 3 iteraciones de pruebas las cuales iniciaron cuando la aplicación estaba implementada con todos los requisitos definidos Número de la iteración para la primera iteración. En la siguiente tabla se recoge de manera explícita el resultado de las pruebas realizadas en cada iteración iniciados en 6.

Número de la iteración	Número de las No Conformidades	Descripción de las No Conformidades	Clasificación	Estado
1	1	Faltas de ortografías	Significativa	Pendiente
	2	Los campos vacíos no arrojaban mensajes de error	Significativa	Resuelta
	3	Los campos no estaban validados según el tipo de carácter que debía ser introducido.	Significativa	Resuelta
	4	El campo existencia no decremento su cantidad.	Significativa	Resuelta
	5	No se generaba el Excel	Significativa	Pendiente
	6	Los mensajes de error no se mostraban	Significativa	Pendiente
2	1	Faltas de ortografías	Significativa	Resuelta
	2	No se generaba el Excel	Significativa	Pendiente
	3	Los mensajes de error no se mostraban	Significativa	Pendiente
3	1	Los mensajes de error no se mostraban	Significativa	Resuelta
	2	No se generaba el Excel	Significativa	Resuelta

CAPÍTULO 3

4	1	No se encontraron		
---	---	-------------------	--	--

3.5.4 Ejemplo de casos de pruebas.

A continuación se muestran algunas de las planillas de casos de pruebas referentes a realizar búsqueda avanzada.

Tabla 1 Caso de prueba de la HU gestionar literatura

Escenario	Descripción	VARIABLE 1 T. Literatura	VARIABLE 2 Título	VARIABLE 3 Imagen	VARIABLE 4 Autor	VARIABLE 5 Género	VARIABLE 6 Editorial	VARIABLE 7 Año de edición	VARIABLE 8 código	VARIABLE 9 Existencia	VARIABLE 10 Precio	Respuesta del sistema	Flujo central
Insertar literatura correctamente	Mediante este escenario el usuario puede insertar un producto en el sistema	V libro	V La Edad de Oro	N/A	V José Martí	V Cuento	V Gente Nueva	V 1999	V 2334	V 20	V 20,00	El sistema permite insertar un libro mostrando un mensaje satisfactorio.	El usuario una vez autenticado en el sistema selecciona la opción agregar contenido. El sistema muestra la opción Literatura. El usuario llena los datos

CAPÍTULO 3

													correspon dientes y presiona el botón guardar
Campos vacíos	No permit e inserta r la literatu ra en el sistem a	I vací o	V La Edad de Oro	N/A	V José Mart í	V cuen to	I vací o	V 199 9	I vací o	I vacío	I vací o	La literatur a no se inserta	El usuario una vez autenticad o en el sistema seleccion a la opción agregar contenido. El sistema muestra la opción Literatura. El usuario llena los datos correspon dientes si deja algún campo vacío y presiona el botón guardar,

CAPÍTULO 3

Tabla 2 Caso de prueba de la HU editar literatura

Escenario	Descripción	Variable 1 T. literatura	Variable 2 Título	Variable 3 "imagen	Variable 4 autor	Variable 5 genero	Variable 6 editorial"	Variable 7 Año de edición	Variable 8 código	Variable 9 existencia	Variable 10 precio	Respuesta del sistema	Flujo central
Editar literatura	Mediante este escenario se puede editar una literatura en el sistema	V	V La edad de oro	N/A	V José Martí	V Cuento	V Gente nueva	V 1999	V 2334	V 20	V 10	El sistema permite editar un libro mostrando un mensaje satisfactorio	El usuario una vez autenticado en el sistema selecciona la opción gestionar literatura. El sistema muestra la opción editar. El usuario realiza los cambios correspondientes y presiona el botón guardar
Cambios	Se cambió	V	V La	N/A	V José	V Cuento	V Gente	V 1999	V 2334	V 15	V 20	La literatura	El usuario una vez autenticado

CAPÍTULO 3

cambios	an los datos	literatura	edad de oro		é Mart í	nto	te nueva	9	4			ura se edita	o en el sistema selecciona la opción gestionar literatura. El sistema muestra la opción editar. El usuario realiza los cambios correspondientes y presiona el botón guardar. El sistema lanza un mensaje satisfactorio.
---------	--------------	------------	-------------	--	----------	-----	----------	---	---	--	--	--------------	--

Tabla 2.1 Descripción de las variables del caso de prueba de la HU editar literatura.

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Tipo de literatura	Campo de acción	No	
2	Título	campo de texto	No	Sólo permite cadena de caracteres y enteros
3	Imagen	fichero	si	
4	Autor	campo de texto	No	Sólo permite cadena de caracteres.

CAPÍTULO 3

5	Género	campo de texto	No	Sólo permite cadena de caracteres.
6	Editorial	campo de texto	No	Sólo permite cadena de caracteres.
7	Año de edición	campo de texto	No	Sólo permite cadena de enteros
8	Código	campo de texto	No	Sólo permite cadena de enteros y de caracteres
9	Existencia	campo de texto	No	Sólo permite cadena de enteros
10	Precio	campo de texto	No	Sólo permite cadena de enteros y decimales.

Tabla 3 Caso de prueba de la HU Eliminar literatura.

Escenario	Descripción	Variable 1 "eliminar"	Respuesta del sistema	Flujo central
Eliminar una literatura correctamente	Mediante este escenario se puede eliminar una literatura del sistema	V Se selecciona la opción correspondiente a eliminar la literatura	El sistema permite al usuario autenticado eliminar una literatura de la base de datos	El usuario una vez autenticado en el sistema selecciona la opción gestionar literatura. El sistema muestra la opción eliminar. El sistema muestra un mensaje de advertencia. Si el usuario presiona la opción cancelar no se

CAPÍTULO 3

				elimina la literatura
--	--	--	--	-----------------------

Tabla 3.1 Descripción de las variables del caso de prueba de la HU eliminar literatura.

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	eliminar	Campo de acción	No	Sólo se acepta seleccionar la opción.

3.5.5 Pruebas de carga y estrés.

Las pruebas de estrés y carga se utilizan para tratar de determinar el rendimiento de un sistema, validando atributos de calidad como fiabilidad, estabilidad. Estas permiten evaluar la capacidad de la máquina y la infraestructura, basándose en requerimientos de negocio.

Se realizaron las pruebas de carga y estrés a la aplicación empleando la herramienta JMeter, se tomó una muestra de 1000 usuarios con una concurrencia de 100 a la vez, para ver el rendimiento de las funcionalidades.

Condiciones de ejecución:

1. Microprocesador Intel® Core (TM) i3 - 2120 CPU 3.30GHZ
2. Memoria RAM de 4 GB.
3. Disco duro de 320 GB.
4. Sistema Operativo Ubuntu 12.04.
5. Red a 100 Mbps

Los resultados que se obtuvieron fueron los siguientes:

CAPÍTULO 3

Funcionalidad	Tiempo mínimo (milisegundos)	Tiempo máximo (milisegundos)	Rendimiento (segundos)
Acceso a la portada	3606	21671	2.224
Acceso al catálogo	1189	19017	0.752
Acceso a la búsqueda avanzada	1482	16476	1.057
Acceso a quiénes somos	3834	15543	0.955
Total	27	21671	21.318

3.5.6 Resultados de las pruebas de carga y estrés.

En las pruebas realizadas al sistema “Librería UCI”, se obtuvo como resultado que para una muestra de 2000 usuarios conectados el rendimiento de la aplicación es de 21.318 segundos, siendo satisfactorio el tiempo de respuesta según las condiciones de las maquinas donde se aplicaron.

3.6 Conclusiones parciales.

En el recién finalizado capítulo se implementó las historias de usuarios según su descripción textual cumpliendo con el plan de iteraciones, además de definir los patrones de diseño a utilizar en el desarrollo de la aplicación. Se describe los estándares de codificación que cumple el sistema y la base de datos. La realización de las pruebas de caja negra se desarrolló al culminar la primera iteración del plan de desarrollo con el objetivo de ir limando errores que arrojaban las no conformidades sobre la aplicación, se describe la prueba de carga y estrés a la que fue sometido el sistema, con la finalidad de asegurar que este cumpla con los requerimientos no funcionales. En todos los casos, las mismas devolvieron resultados satisfactorios, asegurando la fiabilidad y eficiencia del catálogo Librería UCI. Con la implementación de estas funcionalidades, se obtuvo una primera versión del producto, que permitió validar la arquitectura propuesta y obtener una mejora en el proceso de venta de la librería UCI.

CONCLUSIONES GENERALES

Conclusiones Generales

Con el desarrollo de este trabajo de diploma se da cumplimiento a los objetivos plasmados al inicio del mismo: con el estudio de diferentes sistemas de catálogos en el ámbito internacional se concluye que los mismos aportaron para la aplicación una mejor estructura y organización de las funcionalidades a brindar. Las entrevistas realizadas a los trabajadores de la librería valieron de gran ayuda para comprender cómo funciona el proceso de venta en esta entidad.

El extenso estudio sobre la arquitectura, tecnologías y herramientas dictadas por el Centro de Informatización Universitaria y atendiendo a las metodologías y herramientas de desarrollo establecidas, se realizó la fase de planificación y diseño de la solución propuesta para comprender la lógica del negocio y definir las deficiencias existentes en el mismo para puntualizar las funcionalidades con las que debe cumplir el sistema y la estructura que debe estar presente en el mismo, permitiendo de esta forma sentar las bases para una implementación exitosa.

Con el desarrollo del catálogo Librería UCI se creó una aplicación que elimina el trabajo manual respecto a la venta de productos literarios que se realiza en la entidad, permitiendo resolver las deficiencias que existen y contribuyendo además con otras funciones tales como, lograr un mejor conocimiento de los productos que se ofertan y mantener informado a los usuarios de los nuevos libros, revistas y folletos que ingresaron al local.

Se vincularon los componentes de la aplicación con su arquitectura, lo que facilitó el desarrollo del catálogo según las especificidades y necesidades del cliente, esto fue comprobado mediante la realización de varias iteraciones de pruebas de caja negra a la aplicación las cuales arrojaron algunas no conformidades que fueron solucionadas en las posteriores iteraciones y las pruebas de cargas y estrés propiciando que el producto final cumpliera con la calidad y validez requerida.

RECOMENDACIONES

Recomendaciones.

Una vez concluido el presente trabajo de diploma se exhorta a:

- La utilización de esta experiencia como material de estudio para el desarrollo de aplicaciones similares por parte de otros desarrolladores a lo largo del país.
- Incorporar la aplicación a la arquitectura del sistema de la Universidad.
- Agregar una funcionalidad que muestre los diferentes eventos que realiza librería en la Universidad.
- Incorporar una ayuda en línea de forma tal que los usuarios puedan intercambiar ideas e inquietudes con los administradores del sistema.

BIBLIOGRAFÍA REFERENCIADA

Bibliografía referenciada.

1. Biblioteca Nacional. [Citado el: 15 de Enero de 2013], <http://www.bibliotecanacional.gov.co/eContent/home.asp/index.php>.
2. Definición de Catálogo » Concepto en Definición ABC [Citado el: 15 de Enero de 2011], <http://www.definicionabc.com/general/catalogo.php>.
3. Que es un CMS [En línea]. [Citado el: 14 de Enero de 2013.] <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>
4. Drupal-características [En línea]. [Citado el: 14 de Abril de 2013.] <http://www.slideshare.net/pabloto/drupal-caracteristicas>
5. Peñalver, Romero. Trabajo de diploma: Metodología ágil para proyectos de software libre. Ciudad Habana : Universidad de Ciencias Informáticas, 2008.
6. Gracia, J. CMM - CMMI Nivel 2. Calidad. Ingeniería del Software. [Citado el: 2 de Abril de 2013] <http://www.ingenierossoftware.com/calidad/cmm-cmmi-nivel-2.php>.
7. Palacio, Juan. 2008. Flexibilidad con Scrum. 2008
8. Letelier, Patricio y Penadés, M^a Carmen . Metodologías ágiles para el desarrollo de
9. software:eXtreme Programming (XP). Valencia : s.n.
10. Accesibilidad y CMS : Drupal, una herramienta de gestión de contenido con vocabulario. [En línea]. [Citado el: 14 de Enero de 2013.] <http://hugodonairemiles.blogspot.com/2006/11/accesibilidad-y-cms-drupal-una.html>.
11. Sobre Drupal | Drupal Hispano[Citado el: 2 de Abril de 2012], <http://drupal.org.es/drupal>.
12. MONOGRÁFICO: Drupal 7 - 2 Características de Drupal 7. [Citado el: 2 de Abril de 2013], <http://recursostic.educacion.es/observatorio/web/es/software/software-general/1022-monograficodrupal-7?start=2>.
13. Qué es PHP ?. [Citado el: 2 de Abril de 2013], <http://www.misrespuestas.com/que-es-php.html>.

BIBLIOGRAFÍA REFERENCIADA

14. Sobre PostgreSQL | www.postgresql.org.es. [En línea]. [Citado el: 16 de Enero de 2013.]
http://www.postgresql.org.es/sobre_postgresql.
15. Servidor Web - Investigaciones - Carlitoswu. [Citado el: 16 de Enero de 2013],
<http://www.buenastareas.com/ensayos/Servidor-Web/1288892.html>.
16. Apache 2.2: servidor web | Observatorio Tecnológico. [Citado el: 2 de Abril de 2013],
<http://recursostic.educacion.es/observatorio/web/es/software/servidores/580-elvira-mifsud>.
17. Visual Paradigm For Uml . [En línea] [Citado el: 18 de Enero de 2013.]
<http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
18. Rodríguez, F.G., Experto en Drupal 7, Nivel Avanzado. Curso de creación y gestión de portales web con Drupal 7: 21/3/2012. 507.
19. Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objeto.
20. Pressman, R. S. (2005). Ingeniería de Software un enfoque práctico. 6 ed. 2005, [Citado el:: 30 de mayo del 2013].
21. Buenas tareas [En línea] [Citado el: 15 de Mayo de 2013.]
<http://www.buenastareas.com/temas/pruebas-software/0>.

BIBLIOGRAFÍA CONSULTADA

Bibliografía consultada.

1. Entorno virtual de aprendizaje de la Universidad de las Ciencias Informáticas. <http://eva.uci.cu/>.
2. Tesis.uci. <http://tesis.uci.cu/news.php>.
3. Rodríguez, F.G., Experto en Drupal 7, Nivel Avanzado. Curso de creación y gestión de portales web con Drupal 7: 21/3/2012. 507.
4. Rodríguez, F.G., Experto en Drupal 7, Nivel Intermedio. Curso de creación y gestión de portales web con Drupal 7: 21/3/2012. 507.
5. Rodríguez, F.G., Experto en Drupal 7, Nivel Básico. Curso de creación y gestión de portales web con Drupal 7: 21/3/2012. 507.
6. Natalia Juristo, Ana M. Moreno y Sira Vegas. Técnicas de Evaluación de Software. 2006.
7. Metodologías ágiles para el desarrollo de software SCRUM + XP.
http://unkasoft.googlepages.com/desarrollo_agil_UPSA.pdf.
8. Definición. De. [En línea] 2008. <http://definicion.de/pdf/>.
9. Curso de java [En línea] 2009. <http://www.cursodejava.com.mx/cursodejava102.php>.
10. MasAdmin. [En línea] 2009. [Citado el: 11 de Enero de 2011.] <http://www.masadmin.com>.
11. Drupal. [En línea] <http://www.drupal.org>.
12. Visual Paradigm for UML.
<http://www.visualparadigm.com/product/vpuml/editions/modeler.jsp>. Design Patterns of Drupal [En línea] 2012 [Citado el 20 Mayo 2013] <http://drupalwatchdog.com/1/1/design-patterns-of-drupal>.

Glosario de Términos

A

APIs: Interfaz de programación de aplicaciones (IPA) o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

B

Blogs: Sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

BSD: es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

C

CENIA: Centro de Informatización Universitaria.

CMS: Un sistema de gestión de contenidos (o CMS, del inglés Content Management System) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás roles.

Crystal Clear (): Es una metodología ágil que puede ser usado en proyectos pequeños y como casi todos los otros métodos, CC consiste en valores, técnicas y procesos..

ANEXOS

CMMI: (Capability Maturity Model Integration). Es el modelo de procesos más difundido a nivel mundial para la mejora en las organizaciones, tanto en la madurez de la organización como en la capacidad de los procesos.

C++: es un lenguaje de programación. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos.

D

Drupal: Es un sistema de gestión de contenidos, es modular, multipropósito y muy configurable que permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

DB2: Es una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos. Es un motor de base de datos relacional que integra XML de manera nativa, que permite almacenar documentos completos dentro del tipo de datos XML para realizar operaciones y búsquedas de manera jerárquica dentro de éste, e integrarlo con búsquedas relacionales.

E

Extreme Modeling: s la creación de modelos que abarcan el cambio. Puedo crear modelos para cambiar todos los días y para ser desechado cuando han cumplido su propósito. Esta agilidad se logra manteniéndolos peso ligero, capturando solamente los aspectos esenciales de lo que está tratando de comunicar.

F

FEU: Federación Estudiantil Universitaria

H

Hipertextos: Se trata de una herramienta de software con estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos.

ANEXOS

I

IDE: Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación.

Indentación: La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos.

J

Java: Java es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

M

Multiusuario: se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente.

Multiplataforma: En informática, multi-plataforma, es un atributo conferido a los programas informáticos o los métodos de cálculo y los conceptos que se ejecutan e interoperan en múltiples plataformas informáticas.

Métrica3: es una metodología de planificación, desarrollo y mantenimiento de sistemas de información.

MVCC: Control de Concurrencia para Múltiples Versiones de PostgreSQL.

Multihilos: La creación de un nuevo hilo es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente).

P

POO: La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos

ANEXOS

R

RUP: El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software.

S

SQL: El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

SGBD: Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar modificar y analizar los datos.

T

Triggers: Un trigger (o disparador) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

U

UCI: Universidad de las Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

X

XML: XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y

ANEXOS

permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes.