

Universidad de las Ciencias Informáticas

Facultad 3.



**Título: “Sistema de Gestión de Noticias de Cuba en la
Prensa Internacional”.**

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

Autor: Yanelkis Alba Torres.

Tutores: Ing. Adniel Quintana Muñoz.
Ing. Eduardo M. Macías Sotolongo.

La Habana, Junio de 2011
“Año 53 de la Revolución”



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento”

Fidel Castro Ruz.



Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Yanelkis Alba Torres

Autor

Ing. Adniel Quintana Muñoz

Tutor

Ing. Eduardo M. Macías Sotolongo.

Tutor

Datos de contacto

Tutor: Ing. Adniel Quintana Muñoz.

Edad: 26 años.

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente: Profesor en adiestramiento.

Email: aqmunoz@uci.cu

Graduado de la UCI en el curso 2008-2009.

Tutor: Ing. Eduardo Manuel Macías Sotolongo.

Edad: 27 años

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente: Instructor

Email: emmacias@uci.cu

Graduado de la UCI en el curso 2007-2008.



Dedicatoria

A mi madre, ya que si no hubiese sido por ella yo no estaría donde estoy, por depositar toda su confianza en mí y por tener la paciencia y fuerza de esperar en todos estos años que hemos estado separadas.

A mi padre, que aunque no lo tengo mucho tiempo a mi lado, este trabajo es un logro para él también.

A mi hermanito querido, que a pesar que no lo veo casi, este trabajo es para él también.

A mi novio Adriel, por tener tanta paciencia para conmigo y por darme tanto amor en todo momento.

A mi abuelo Pipe, que se que está muy orgulloso de su nieta, abue este trabajo te lo dedico a ti también.

A mis tíos, primas y primos, abuelos, en fin a toda mi familia.

A mis 2 hermanitas de la UCI, Yasmín y Yaira, loqis, esta tesis se la dedico a ustedes también.

A mi amigas del lejano Oriente jajaja, Yanicel, Suleidis, Tania, Bárbara y Yanet.

Todo este esfuerzo es para ustedes.



Agradecimientos

Me parece que en una hoja de este documento no me es suficiente para agradecerles a todas las personas que de una forma u otra han aportado un granito de arena para que yo cumpliera mis sueños de ser ingeniera. Pero bueno aquí les van mis agradecimientos:

A mi mamá, por estar siempre conmigo cuando hace falta, apoyándome en los buenos y malos momentos, sin ella no hubiese sido posible el éxito de mi carrera.

A mi papá, mi ratoncito querido, que aunque siempre no estemos juntos, estoy agradecida por darme tanto amor cuando nos vemos.

A toda mi familia, en especial a todos mis tíos, primas y primos, mis abuelos, gracias por estar siempre pendientes de mí.

A mi novio Adniel, gracias por apoyarme y ser el sostén de mi vida lejos de mi familia.

A mis amistades que siempre compartieron momentos tristes y alegres conmigo, que lloramos y reímos juntos, en especial, Yadira, Yasmín, Suleidis, Tania, Yanicel, Bárbara y Yanet. Independientemente de las vueltas que de la vida, seguiremos siempre siendo amigas.

A la familia de mi novio, especialmente a mi suegra Maritza, por su constante preocupación, por todos sus consejos y poder contar con su apoyo. Gracias por estar presente.

A todos mis compañeros de clases, amistades de la UCI y a todas aquellas amistades que se graduaron, no voy a mencionar nombres porque serían demasiados y tengo miedo que se me queden algunos, a todos ustedes gracias por soportarme estos 5 años. Siempre los recordaré.

A Nadier y Ariel, gracias por ayudarme tanto, sin ustedes no hubiese sido posible que este sueño se volviera realidad. Gracias de todo corazón.

A mis compañeras de cuarto, Dayana, Beylen e Idanis. Gracias por soportarme. Jajaja.

A Albín, Sabrina, Yordan y Leydis, gracias por soportarnos a mí y a Pandion.

A mis tutores Adniel y Eduardo, gracias por haberme apoyado tanto.

A Dios, por haberme permitido tener una mascota como mi bebé Pandion, que con sus juegos y malacrianzas he podido relajarme en mis momentos más estresantes de este curso.

A la Revolución y a nuestros comandantes Fidel y Raúl Castro por darnos la oportunidad de estudiar en esta universidad y formarnos como profesionales.



Resumen

Internet es uno de los principales medios de información que existe en la actualidad, pero cuenta con un gran volumen de información, muy dispersa. Por ello, el almacenamiento de noticias para sus posteriores análisis y las herramientas que sirven de soporte a dicha actividad son de vital importancia. Sobre todo cuando se está bajo constante ataque mediático como lo está Cuba. Por lo que en la Universidad de las Ciencias Informáticas (UCI), se le dio la tarea en coordinación con el Centro de Información para la Prensa (CIP), y por orientación del Comité Central del Partido Comunista de Cuba (CCPCC) la creación de una herramienta que sea capaz de gestionar las noticias que traten sobre Cuba en la prensa extranjera, para sus posteriores análisis. Y en base a dichos análisis trazarse una posible estrategia a seguir.

El presente trabajo de diploma tiene como objetivo: Desarrollar un sistema de gestión de noticias para facilitar la toma de decisiones.

El sistema fue desarrollado sobre el CMS Drupal, con PHP como lenguaje de programación del lado del servidor. Del lado del cliente se utilizaron: JavaScript, HTML y CSS. Utilizando como sistema gestor de base de datos a MySQL y como entorno de desarrollo Netbeans.

El Sistema de Gestión de Noticias de Cuba en la Prensa Internacional permite la edición, almacenamiento y actualización de las noticias recopiladas durante el proceso de seguimiento de noticias llevado a cabo por el grupo de Vigilancia Tecnológica y Política, perteneciente al CIDI.

Palabras Claves.

Gestión de noticias, CMS Drupal.



Índice

INTRODUCCION.....	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	15
Introducción.....	15
1.1 Conceptos fundamentales asociados al negocio.....	15
1.1.1 Sistemas de Gestión de Contenidos (Content Management Systems, CMS).....	15
1.2 Tendencias y Tecnologías actuales.....	20
1.2.1 Sistemas Gestores de Bases de Datos (SGBD).....	20
1.2.2 Herramientas CASE.....	22
1.2.3 Metodologías de desarrollo de software.....	24
1.2.4 Lenguaje de programación del lado del servidor.....	29
1.2.5 Tecnologías del lado del cliente.....	32
1.2.6 Entornos de Desarrollo Integrado (Integrated Development Enviroment o IDE).....	34
1.2.7 Servidor Web.....	36
Conclusiones parciales del capítulo.....	37
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....	38
Introducción.....	38
2.1 Modelo de dominio.....	38
2.1.1 Conceptos principales del modelo de dominio.....	38
2.1.2 Diagrama de clases del dominio.....	39
2.1.3 Descripción del modelo de dominio.....	39
2.1.4 Roles y permisos.....	40
2.1.5 Reglas del negocio.....	40



2.2 Especificación de los requisitos de software.....	40
2.2.1 Requisitos funcionales.	41
2.2.2 Requisitos no funcionales.	46
2.3 Patrones de casos de uso a utilizar.....	48
Múltiples actores: Roles comunes.....	48
CRUD Completo.	48
2.4 Descripción del sistema propuesto.....	49
2.5 Modelo del sistema. Definición de actores y casos de usos del sistema.....	49
2.5.1 Definición de los actores del sistema.....	49
2.5.2 Definición de casos de usos del sistema.....	50
2.5.3 Diagrama de casos de uso del sistema.....	54
2.6 Modelo de Análisis.....	55
2.6.1 Diagramas de clases del análisis.....	55
2.6.2 Diagramas de interacción.....	55
2.7 Modelo de Diseño.	56
2.7.1 Diagramas de clases del diseño.	56
Diagramas de clases del diseño web.....	58
2.8 Visión general de la arquitectura de software y patrones utilizados por Drupal.....	58
2.8.1 Arquitectura de Software.	59
2.8.2 Patrones utilizados por Drupal.....	60
2.9 Diseño de la Base de Datos.	61
2.9.1 Diagrama de clases persistentes.....	62
2.9.2 Modelo de datos de clases persistentes.....	63



Conclusiones parciales del capítulo.	63
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA.	64
Introducción.....	64
3.1 Diagrama de componentes.....	64
3.2 Diagrama de despliegue.	65
3.2.1 Descripción de los nodos.	66
3.2.2 Protocolos.....	67
3.3 Usuarios y permisos.....	67
3.4 Análisis de posibles implementaciones (reutilización de código).	68
3.4.1 Módulos contribuidos de Drupal.....	69
3.5 Módulos implementados.....	69
3.5 Pruebas.	70
Conclusiones parciales del capítulo.	71
CONCLUSIONES	72
RECOMENDACIONES.....	73
REFERENCIAS BIBLIOGRÁFICAS.....	74
BIBLIOGRAFÍA.....	76



INTRODUCCION

En el mundo moderno, las noticias han pasado a ser un elemento de gran importancia en todas las sociedades. Ya que mediante la divulgación de las mismas se garantiza que los distintos sucesos que surjan, sean conocidos por todas las personas interesadas.

Con el surgimiento de Internet se dio un paso trascendente para los medios de comunicación ya que mediante él es posible llegar a cualquier parte del mundo casi instantáneamente y con costos mínimos. Este fenómeno trajo consigo un desarrollo exponencial de los medios de comunicación en Internet, convirtiendo el ciberespacio en un escenario internacional para promover estados de opinión y movilizar masas.

Las posibilidades que brinda el ciberespacio para controlar la opinión pública internacional han sido explotadas por los grandes monopolios de la información al servicio de los países que siguen una política imperialista, fundamentalmente de Estados Unidos (EE.UU) y parte de una unión Europea, para imponer sus principios, ideales y lograr sus objetivos de dominación mundial sobre los pueblos y sus recursos naturales. Esto queda demostrado en el discurso de toma de posesión de la Secretaria de Estado de EE.UU Hillary Clinton cuando expresa:

“Es necesario utilizar la fuerza de Internet contra los países que combaten los medios de comunicación estadounidenses, sobre todo empleando Facebook, YouTube, Flickr y Twitter para hacer llegar allí las voces de EE.UU.”

Cuba es centro de atención en diversos medios de comunicación en Internet, muchos de estos medios sirven de plataforma para desacreditar a la Revolución Cubana, desinformando y tergiversando la realidad política, económica y social.

Ante esta situación se hace necesario realizar seguimiento constantemente a los medios de comunicación internacionales para identificar oportunamente cuando se fomenta una campaña mediática en contra de Cuba. A partir de una orientación dada por el Departamento Ideológico del Comité Central del Partido Comunista de Cuba (CCPCC) se decide crear un proyecto de Observatorio de Noticias en Internet y se decide que el Centro de Información de la Prensa (CIP) sea la institución encargada de coordinarlo y el Centro de Investigación y Desarrollo sobre Internet (CIDI) de la Universidad de las Ciencias Informáticas



(UCI) sería el encargado de desarrollar las herramientas informáticas necesarias. Dicho proyecto debía servir de base para la toma de decisiones por el CCPCC para elaborar estrategias de respuesta ante posibles campañas mediáticas en contra de Cuba.

En un estudio realizado por el CIP se identificó la necesidad de gestionar, procesar y almacenar las noticias publicadas por los medios en Internet. De esta manera se podrían generar estadísticas, reportes y análisis históricos del comportamiento de los medios de comunicación, permitiendo crear perfiles ideológicos de los medios y de los autores de las noticias.

A partir de lo antes expuesto surge el **problema a resolver**: ¿Cómo contribuir a la gestión y procesamiento de las noticias publicadas en Internet referentes a Cuba que facilite la toma de decisiones? En correspondencia con lo anterior se plantea como **objeto de estudio**: Proceso de desarrollo de software.

Para darle solución al problema existente se trazó como **objetivo general**: Desarrollar un sistema de gestión de noticias para facilitar la toma de decisiones. La investigación se enmarca en el siguiente **campo de acción**: Desarrollo de software de gestión.

La investigación estará guiada por la siguiente **idea a defender**: Con el desarrollo de un sistema de gestión de noticias, se viabilizará la toma de decisiones.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- ✓ Desarrollar el marco teórico conceptual respecto a los sistemas de gestión de noticias.
- ✓ Diseñar un sistema para gestionar noticias.
- ✓ Implementar el sistema de gestión de noticias.
- ✓ Validar la aplicación desarrollada.

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

- ✓ Estudio de las nuevas tendencias de las tecnologías de la información relacionadas con la gestión de noticias.
- ✓ Estudio y definición de conceptos sobre los temas relacionados con la gestión de noticias.
- ✓ Análisis de las metodologías para definir cuál se va a utilizar en el desarrollo del sistema de gestión de noticias.



- ✓ Estudio de los procesos de gestión de noticias.
- ✓ Identificación de necesidades de información, requisitos funcionales y no funcionales.
- ✓ Desarrollo de los artefactos establecidos por la metodología seleccionada.
- ✓ Implementación del sistema de gestión de noticias.
- ✓ Aplicación de las pruebas de caja blanca y caja negra a la solución desarrollada.

Como en todo proceso investigativo se utilizaron **métodos científicos** para el desarrollo de la investigación:

Métodos Teóricos:

- ✓ **Histórico-Lógico:** Este método se utilizó para el estudio de los temas relacionados con el desarrollo de los sistemas de gestión de noticias, su evolución y para la metodología a utilizar en el desarrollo.
- ✓ **Analítico – Sintético:** Se utilizó para el análisis de documentos, materiales, y temas relacionados con los sistemas de gestión de noticias. Permitió definir los conceptos fundamentales del tema y se usó para especificar la metodología a utilizar.
- ✓ **Hipotético – Deductivo:** para plantear la idea a defender a partir de la problemática detectada y los objetivos definidos.
- ✓ **Modelación:** Se utilizó para la modelación de los artefactos del sistema en el flujo de trabajo de diseño.

Métodos Empíricos:

- ✓ **Entrevista:** para obtener la información acerca de las necesidades del cliente, requisitos funcionales.
- ✓ **Experimento:** este método se utilizó a la hora de realizarle las pruebas pertinentes al sistema.
- ✓ **Observación:** para realizar un registro visual de lo que ocurre en el entorno del problema y aportar nuevos elementos que puedan ser de interés científico.



El presente trabajo consta de 3 capítulos:

Capítulo 1 titulado “Fundamentación teórica”: Incluye el estado del arte del tema tratado, describe el objeto de estudio, el campo de acción y la problemática existente. Además, se selecciona la plataforma, lenguaje, herramientas y metodología con las que se llevará a cabo el sistema.

Capítulo 2 titulado “Análisis y diseño del sistema”: se encuentra la definición del negocio, el tema de análisis, las reglas del negocio, los requisitos (los funcionales y los no funcionales) y los casos de uso del sistema, se identifica la estructura de usuarios y permisos.

Capítulo 3 titulado “Implementación y prueba del sistema”: contiene todos los elementos referentes a la implementación del sistema donde se generó el modelo físico de la base de datos, concluyendo con la realización de pruebas para comprobar la calidad del sistema por parte de los desarrolladores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

En este capítulo se realiza un estudio de los sistemas de gestión de noticias, así como los conceptos fundamentales asociados al negocio. Se explican y justifican las tecnologías y herramientas a utilizar para el desarrollo del sistema. Se fundamentará la selección realizada de la metodología de desarrollo, el lenguaje de modelado, el lenguaje de programación y la herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) que se emplearán en la realización de la aplicación.

1.1 Conceptos fundamentales asociados al negocio.

1.1.1 Sistemas de Gestión de Contenidos (Content Management Systems, CMS).

En el mundo moderno el seguimiento de las noticias es un elemento clave a la hora de la toma de decisiones, debido a esto se hace necesario utilizar herramientas que ayuden a gestionar de una forma correcta los contenidos publicados en los diferentes medios noticiosos de interés, surgiendo como alternativa los sistemas de gestión de contenidos.

El concepto de CMS se centra en un ciclo de creación, administración y publicación de contenidos:



Figura 1 Ciclo de vida de un CMS

El mismo concepto de CMS deja claro algunos beneficios decisivos en el manejo de contenidos, en particular:

- ✓ Mejor control del proceso de autoría de contenidos.
- ✓ Soporte a procesos de autoría descentralizados.
- ✓ Tiempos mínimos en el cambio de páginas en la Web.
- ✓ Mayor consistencia de los contenidos.
- ✓ Facilidad de navegación.
- ✓ Mayor flexibilidad.
- ✓ Mejora de los niveles de seguridad.
- ✓ Reduce la posibilidad de duplicación de información.
- ✓ Posibilidad de crecimiento controlado.
- ✓ Reducción de los costos de mantenimiento de sitios. (scribd.com)

A grandes rasgos se puede decir que los sistemas de gestión de contenidos son un conjunto de herramientas y procesos que recopilan y gestionan contenidos además de ser un software que se utiliza principalmente para facilitar la gestión web.

James Robertson¹ propone una división de la funcionalidad de los sistemas de gestión de contenidos en cuatro categorías: creación de contenido, gestión de contenido, publicación y presentación.

Creación de contenido.

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG², en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de

¹ James Robertson es el fundador y Director de Step Two Designs, una consultora independiente de proveedores ubicados en Australia. James es reconocido como uno de los líderes de opinión en todo el mundo sobre los temas de gestión de contenido web y la estrategia de la intranet. Ha trabajado con muchas organizaciones en los sectores público y privado, incluyendo organismos del Gobierno Federal.

² WYSIWYG acrónimo de What You See Is What You Get (en inglés, lo que ves es lo que obtienes).

formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio web. (Cuerda, 2004)

Para la creación del sitio, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente. (Cuerda, 2004)

Gestión de contenido.

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, como son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad), datos, preferencias de los usuarios y la estructura de la web. (Cuerda, 2004)

Esta estructura se puede configurar con una herramienta que habitualmente presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. Eso es imprescindible para facilitar el ciclo de trabajo con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del ciclo de trabajo. (Cuerda, 2004)

Publicación.

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas. (Cuerda, 2004)

Presentación.

Un CMS puede gestionar automáticamente la accesibilidad del web, con soporte de normas internacionales de accesibilidad como WAI³ y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm) (Cuerda, 2004)

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro del web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que se incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otra web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado. (Cuerda, 2004)

Las características de los CMS pueden variar, pero la mayoría incluyen publicación basada en la web, indexación, revisión, búsquedas y diversas herramientas como calendario, foros, galería de imágenes, administrador de documentos, administrador de noticias, administrador de artículos, administrador de menús, creador de bloques y encuestas.

Los CMS presentan varias ventajas dentro de las cuales se pueden destacar, no debe ser necesario tener experiencia en la programación web para trabajar con ellos, la mayoría de los CMS tienen páginas dinámicas que muestran información guardada en una base de datos, y poseen un panel para que el administrador del sitio pueda agregar, modificar y eliminar cualquier información.

A continuación se muestra un análisis comparativo entre Drupal y Joomla realizado por un consultor de la IBM. (Graduados.uci.cu)

³ WAI (Iniciativa de Accesibilidad Web), se formó por el Consorcio de la World Wide Web (W3C) para brindar las consideraciones de accesibilidad en el desarrollo de tecnología del Consorcio Web y determinar las pautas para la tecnología accesible incluyendo desarrollo web y agentes del usuario (navegadores). (Avila, 2009)

Características	Drupal vs Joomla
Flexibilidad y Potencia.	Drupal es significativamente más potente y mucho más flexible. Parece que Views, CCK, Panels, le dan a Drupal una enorme ventaja sobre Joomla. Los desarrolladores de Joomla cuando lo diseñaron no tuvieron en cuenta la flexibilidad.
Rendimiento.	Drupal supera ampliamente a Joomla.
Curva de aprendizaje	Joomla es mucho más fácil de configurar y poner en marcha. Drupal todavía es un desafío.
Plantillas	Joomla gana por un amplio margen.
Editor de contenido	El sistema de taxonomías de Drupal es excelente. La aproximación de Joomla es pobre (los tipos de contenido están confinados a una sección/categoría).
Construcción de contenido	Las herramientas de Drupal son muy buenas, son increíblemente mejores que las de Joomla. Con Joomla puedes conseguir componente que ofrezca el 60-80% de lo que necesitas. Por ejemplo, Joomla tiene un tremendo componente de noticias, pero nunca llegará a las capacidades de CCK, Views, que tiene Drupal.

Tabla 1 Análisis comparativo entre los CMS Drupal y Joomla por un consultor de la IBM.

Fundamentación del CMS a utilizar.

De acuerdo a la comparación de los resultados de la Tabla 1, se determinó utilizar como gestor de contenido para el desarrollo de la aplicación a Drupal, por su alto rendimiento, flexibilidad y gestión de contenidos, además de ser la política a seguir definida por el proyecto.

1.2 Tendencias y Tecnologías actuales.

1.2.1 Sistemas Gestores de Bases de Datos (SGBD).

Los SGBD son una colección de programas cuyo objetivo es servir de interfaz entre la Base de Datos (BD), el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (Leyva, y otros, 2010)

Para el desarrollo del sistema es necesario realizar un estudio de los diferentes SGBD existentes, especialmente los que se rigen por las licencias de software libre, evaluando sus características principales, así como las ventajas y desventajas, para determinar cuál de ellos es el más adecuado para la realización del mismo.

PostgresSQL.

Es un SGBD relacionales orientadas a objetos, distribuido bajo licencia de software libre permisiva y con su código fuente disponible libremente. Actualmente, es el SGBD de código abierto más avanzado, soportando casi todas las sintaxis SQL. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. (Leyva, y otros, 2010)

Características de PostgresSQL. (Leyva, y otros, 2010)

- ✓ Soporta diferentes tipos de datos como son: fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de bits, además del soporte para los tipos base y permite la creación de tipos propios.
- ✓ Incorpora una estructura de datos de arreglos.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas y entre gestores de objetos relacionales.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada usuario.

MySQL.

Es un SGBD relacionales. Utiliza como lenguaje de programación Structured Query Language (SQL)

Características de MySQL.

En las últimas versiones de MySQL resaltan las siguientes características:

- ✓ Su principal objetivo es velocidad.
- ✓ Puede trabajar en distintas plataformas y sistemas operativos.
- ✓ Cada base de datos generada por MySQL cuenta con 3 archivos: uno de estructura, uno de datos y uno de índice, soportando hasta 32 índices por tablas.
- ✓ Gracias a su implementación multihilo aprovecha la potencia de sistemas multiproceso.
- ✓ Presenta un buen nivel de seguridad en los datos, ya que posee un sistema de contraseñas y gestión de usuarios.
- ✓ Soporta gran cantidad de tipos de datos.

Ventajas de MySQL

- ✓ Velocidad al realizar las operaciones, es uno de los de mejor rendimiento.
- ✓ Bajo costo en requisitos tecnológicos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos.
- ✓ Facilidad de configuración e instalación.
- ✓ Soportado por gran variedad de Sistemas Operativos.

Fundamentación del SGBD seleccionado.

A pesar de PostgreSQL ser considerado el SGBD más avanzado y poder ser utilizado con Drupal, se decide utilizar para el desarrollo de la aplicación MySQL, por política del proyecto GIDI. La principal razón es por la poderosa combinación del servidor Apache + PHP con MySQL. Además brinda un alto nivel de seguridad en cuanto a permisos y privilegios de usuarios, contando con cinco niveles distintos de privilegios, permitiendo de esta forma limitar las acciones que tales usuarios puedan llevar a cabo:

- ✓ **Nivel global:** Es el nivel más alto de privilegio, en el sentido de que su ámbito es el más general.
- ✓ **Nivel de base de datos:** este nivel se refiere a las bases de datos individuales y a todos los objetos que tiene cada base de datos.

- ✓ **Nivel de tabla:** se aplica a tablas individuales y a todas las columnas de esas tablas.
- ✓ **Nivel de columna:** se aplica a una columna en una tabla concreta.
- ✓ **Nivel de rutina:** se aplican a los procedimientos almacenados.

Tiene una excelente integración con el CMS Drupal, además de que es multiplataforma, es rápido y fácil de usar. Su bajo consumo de recursos lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

1.2.2 Herramientas CASE.

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida del software en tareas como realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Visual Paradigm.

Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Facilita la realización de los diagramas de modelado que sigue el estándar UML como:

- ✓ Diagramas de clase.
- ✓ Casos de Uso.
- ✓ Comunicación.
- ✓ Secuencia.
- ✓ Estado.
- ✓ Actividad.
- ✓ Componentes.

Esta herramienta tiene **características** entre las que se encuentran:

- ✓ Soporta aplicaciones web.
- ✓ Varios idiomas.

- ✓ Generación de código para Java y exportación como HTML.
- ✓ Fácil de instalar y actualizar.
- ✓ Compatibilidad entre ediciones. (Giraldo, y otros, 2005)

RationalRose

Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Es una herramienta de modelado visual, diseñado específicamente para desarrollar las fases de análisis, diseño y construcción de aplicaciones orientadas a objetos. El navegador UML de Rational Rose permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. (VARGAS, 2005)

Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. (VARGAS, 2005)

Características

- ✓ Mantiene la consistencia de los modelos del sistema software
- ✓ Chequeo de la sintaxis UML
- ✓ Generación Documentación automáticamente
- ✓ Generación de Código a partir de los Modelos
- ✓ Ingeniería Inversa (crear modelo a partir código)

Fundamentación de la herramienta CASE a utilizar.

Se decidió trabajar con el Visual Paradigm 6.4 for UML por su robustez, usabilidad y portabilidad. Además por ser una herramienta que se conoce bastante en la universidad, se cuenta con la licencia para su utilización y soporta el ciclo completo del proceso de desarrollo de software. Utiliza UML como lenguaje de modelado, que tiene como propósito visualizar, especificar, construir y documentar proyectos de software.

Visual Paradigm permite dibujar todos los tipos de diagramas que se generan a partir de los artefactos creados por la metodología de desarrollo empleada. Puede ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma y se puede integrar con Netbeans.

1.2.3 Metodologías de desarrollo de software.

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software.

Proceso Unificado de Desarrollo (RUP).

RUP define cuatro fases (inicio, elaboración, construcción y transición) y dentro de cada una de ellas el equipo de trabajo pasa por todos los flujos que son transversales a las fases, inclusive en varias iteraciones.

Inicio: Se describe el negocio y se delimita el proyecto, describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los Requisitos (funcionales y no funcionales) identificados de acuerdo con el alcance definido.

Construcción: Se logra un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene una o varias versiones del producto que han pasado las pruebas. Se ponen estos entregables a consideración de un subconjunto de usuarios.

Transición: La versión ya está lista para su instalación en las condiciones reales. Puede implicar reparación de errores.

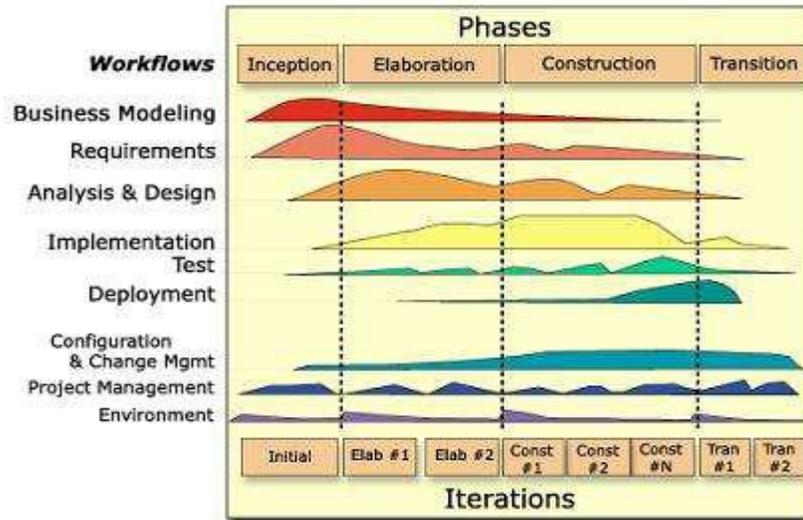


Imagen 1 Fases e Iteraciones de la metodología RUP

El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- ✓ Ingeniería de Negocios: Permite entender las necesidades del negocio.
- ✓ Requisitos: Permite trasladar las necesidades del negocio a un sistema automatizado.
- ✓ Análisis y Diseño: Permite trasladar los requisitos dentro de la arquitectura de software.
- ✓ Implementación: Permite crear software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ✓ Pruebas: Permite asegurarse que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- ✓ Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- ✓ Administrando el proyecto: Administrando horarios y recursos.
- ✓ Ambiente: Administrando el ambiente de desarrollo.
- ✓ Distribución: Hacer todo lo necesario para la salida del proyecto.

Los elementos del RUP son:

- ✓ **Actividades**, son los procesos que se llegan a determinar en cada iteración.

- ✓ **Trabajadores**, vienen hacer las personas o entes involucrados en cada proceso.
- ✓ **Artefactos**, un artefacto puede ser un documento, un modelo, o un elemento de modelo

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

El ciclo de vida de RUP se caracteriza por estar: (Luna, y otros, 2010)

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los Requisitos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, describiendo los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.

El modelo de arquitectura se representa a través de vistas(4+1)⁴, o sea, perspectivas del sistema, que logran una abstracción particular en cada uno de los casos, en otras palabras, se trata de que en cada una de las llamadas vistas se represente el sistema en su totalidad, teniendo en cuenta solo determinados aspectos.

Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, los incrementos y al crecimiento del producto.

⁴ 4+1 vistas de la arquitectura: Vista de casos de uso, vista lógica, vista de procesos, vista de componentes y vista de despliegue.

Extreme Programming (XP).

Esta metodología es muy utilizada a la hora de realizar proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

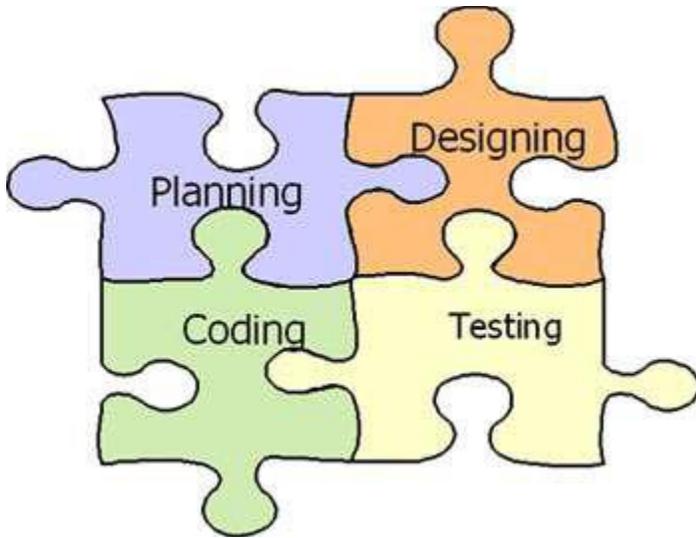


Imagen 2 Metodología Extreme Programming.

La metodología se basa en:

- ✓ Pruebas Unitarias: pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir. Es como si se obtuvieran los posibles errores.
- ✓ Refabricación: reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

XP propone: (Sanchez, 2004)

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

Lo fundamental en este tipo de metodología es: (Sanchez, 2004)

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

Fundamentación de la metodología a utilizar.

Luego de un preciso estudio de las metodologías anteriores se decidió utilizar para el desarrollo del sistema a RUP, por las potencialidades que presenta. Además de propiciar una buena organización y utilización de la información producida a lo largo de todo el proceso de desarrollo del software pudiendo darle el alcance que se espera tenga esta herramienta.

Define claramente actividades realizadas por roles generando a su paso artefactos que sustentan el proceso de desarrollo del producto. RUP constituye una metodología adaptable al proyecto, utilizada para el análisis, implementación y documentación de sistemas a través de UML.

Además por ser una de sus características principales el desarrollo incremental posee las ventajas siguientes:

- ✓ Los clientes no necesitan esperar hasta el fin del desarrollo para utilizar el sistema, pueden empezar a usarlo desde el primer incremento.
- ✓ Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.

1.2.4 Lenguaje de programación del lado del servidor.

Los lenguajes del lado del servidor son los encargados de realizar la conexión entre la base de datos a utilizar y el servidor, generan las páginas antes de enviarlas al cliente, en sentido general son los encargados de todos los aspectos relacionados con la funcionalidad del sistema.

Debido a que el sistema será desarrollado con Drupal el lenguaje de programación del lado del servidor a utilizar será PHP en su versión 5, a continuación se realiza una descripción del mismo.

Lenguaje PHP (Hypertext Pre-procesor).

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles y funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Es un lenguaje interpretado, débilmente tipado y orientado a objeto.

PHP se ejecuta en el servidor, permite acceder a los recursos del servidor, por ejemplo podría ser una base de datos. Las sentencias PHP son ejecutadas en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML



Imagen 3 Funcionamiento de PHP. (Hinostroza, 2007)

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. (Anónimo)

PHP es un interprete que puede ser incluido en un servidor web como un módulo. Con él se pueden realizar accesos a ficheros y conexiones de red. Está diseñado para ser más seguro que cualquier otro lenguaje de programación como C.

Dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas.

Características de PHP: (Hinostroza, 2007)

- ✓ Soporte para una gran cantidad de SGBD: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF y hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.
- ✓ Fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- ✓ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- ✓ Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Ventajas:

- ✓ Es un lenguaje multiplataforma.
- ✓ Completamente orientado a la web.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- ✓ Posee una amplia documentación en su página oficial entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).
- ✓ PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz.
- ✓ Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- ✓ Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.

Desventajas:

- ✓ Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- ✓ La ofuscación de código es la única forma de ocultar los fuentes.

- ✓ El manejo de errores no es tan sofisticado como ASP. (ANONIMOa)

Entre las principales características de PHP se destacan: su rapidez, facilidad de aprendizaje, soporte multiplataforma tanto de diversos sistemas operativos, como servidores http y de bases de datos y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

1.2.5 Tecnologías del lado del cliente.

HTML (HyperText Markup Language)

Es el lenguaje utilizado como base para crear las páginas web. Este lenguaje permite aglutinar textos, sonidos e imágenes y combinarlos para lograr una mejor apariencia de la página web, permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

Describe la estructura y el contenido en forma de texto empleando etiquetas (<Body>...</Body> o <P>...</P>) para ello. Debido a que es un estándar independiente de fabricantes y marcas puede ser interpretado por todos los navegadores, siendo sus aplicaciones muy rápidas, con mucho desarrollo y ocupan poco espacio.

La mayoría de las etiquetas del lenguaje HTML son semánticas. HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo.

CSS (Cascading Style Sheets)

Es una tecnología que permite crear páginas web de una manera más exacta. Son, además, un lenguaje formal que define cómo se va a mostrar un documento estructurado, escrito en HTML o XML y persigue separar la estructura de la presentación

La gran ventaja que poseen las hojas de estilo en cascadas es que un cambio en la hoja de estilo modifica inmediatamente todas las páginas que estén relacionadas con esta sin necesidad de hacerlo directamente en cada una.

La sintaxis CSS permite aplicar al documento un formato de modo mucho más exacto. Si antes el HTML no era suficiente para maquetar las páginas y se tenían que utilizar trucos para conseguir los efectos, ahora con el uso de las CSS se tienen muchas más herramientas que permiten definir esta forma:

- ✓ Definir la distancia entre líneas del documento.
- ✓ Se puede aplicar identado a las primeras líneas del párrafo.
- ✓ Colocar elementos en la página con mayor precisión, y sin lugar a errores.
- ✓ Definir la visibilidad de los elementos, márgenes, subrayados y tachados.

Con el HTML tan sólo se podían definir atributos en las páginas con pixeles y porcentajes, ahora se pueden definir utilizando muchas más unidades como:

- ✓ Pixels (px) y porcentaje (%), como antes.
- ✓ Pulgadas (in).
- ✓ Puntos (pt).
- ✓ Centímetros (cm).

JavaScript

Es un lenguaje de programación del lado del cliente, utilizado para crear pequeños programas encargados de realizar acciones dentro de una página web. El navegador del cliente, es el encargado de interpretar las instrucciones y ejecutarlas para realizar efectos e interactividades, de modo que el mayor recurso y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Es bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Entre las acciones típicas que se pueden realizar en Javascript se destacan los efectos especiales sobre páginas web, la creación de contenidos dinámicos, elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo y validaciones de datos en los formularios.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeñas secuencias de comandos, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejos. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Permitiendo que este se convierta en el verdadero dueño y controlador, de cada cosa que ocurre en la página cuando la está visualizando el cliente.

Fundamentación de las tecnologías del lado del cliente a utilizar.

Se seleccionó HTML, CSS y JavaScript por la gran integración que poseen entre ellos y las buenas funcionalidades que se logran a través de su combinación, obteniéndose unas interfaces totalmente dinámicas, seguras e innovadoras. Además de ser compatibles con la mayoría de los navegadores web.

Gracias a la utilización de CSS en la aplicación, un cambio realizado en la hoja de estilo modifica inmediatamente todas las páginas que estén relacionadas con esta sin necesidad de hacerlo directamente en cada una, agilizándose de esta forma el trabajo de los desarrolladores, ya que les facilita el mismo y les ahorra tiempo.

JavaScript es un lenguaje bastante sencillo que permite realizar acciones típicas sobre páginas web, destacándose los efectos especiales para crear contenidos totalmente dinámicos y permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

1.2.6 Entornos de Desarrollo Integrado (Integrated Development Environment o IDE).

Un entorno de desarrollo integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos.

Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. (integrado)

Netbeans.

El Netbeans es un IDE, basado en estándares, escrito en el lenguaje de programación Java. El proyecto Netbeans consiste en un IDE de código abierto y una plataforma de aplicaciones, que puede ser utilizado como un marco genérico para construir cualquier tipo de aplicación. El enfoque de Netbeans es la mejora de la productividad a través de un editor más inteligente, más rápido, y la integración de todos los productos en un IDE. (Muñoz, 2009)

Netbeans propone un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS. Además posee un sistema para examinar todo los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

Permite indentar automáticamente el código, mediante una combinación de teclas. Entre sus principales características se encuentran:

- ✓ Brinda completamiento automático de código PHP, así como coloreado de código sintáctico y semántico.
- ✓ Permite depurar el código usando Xdebug⁵.
- ✓ Genera fragmentos de código para bases de datos MySQL.

Zend Studio

Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. Se trata de un programa, uno de los mayores impulsores de PHP, orientada a desarrollar aplicaciones web. (Salazar, 2009)

Características y beneficios. (Salazar, 2009)

- ✓ Combina todas las herramientas que trabajan regularmente para desarrollar su aplicación en una interfaz unificada.
- ✓ Se puede editar PHP, HTML, JavaScript y código fuente.
- ✓ Posee código para estructurar las plantillas de código PHP rápidamente.

⁵ Es una extensión PHP que proporciona capacidades de depuración y perfiles.

- ✓ Presenta paquetes de múltiples archivos y directorios en una sola entidad del proyecto haciendo la aplicación simple mediante la navegación y la búsqueda.

Fundamentación del IDE a utilizar.

Para el desarrollo del sistema se decide utilizar NetBeans 6.8 por ser el IDE de desarrollo definido en las pautas de arquitectura del proyecto. Además en un estudio realizado por Mariana Raquel Salazar Quispillo Moyota y Paulina Monserrath Tasambat en su tesis de grado, (Salazar, 2009), plantean que Zend Studio es poco eficiente en cuanto al manejo de CSS, no cuenta con una facilidad de aprendizaje y uso de los elementos gráficos y presenta un insuficiente manejo visual de elementos de un formulario.

1.2.7 Servidor Web.

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada).

Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

A partir del anterior esquema se han diseñado y desarrollado todos los servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI⁶, Servlets) que pueden atender, en función de que sean o no sean multi-proceso o multi-hilados. (Cibernetia)

⁶ Interfaz de entrada común es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

Apache

Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuito es importante pero no tanto como que se trate de código fuente abierto.

Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este. Otra cosa importante es que cualquiera que posea una experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

Apache trabaja con PHP y otros lenguajes de script. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

Tiene una alta configurabilidad en la creación y gestión de log. Permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. (Ciberaula)

Por lo anteriormente expuesto para el desarrollo del sistema se utilizará como servidor web Apache.

Conclusiones parciales del capítulo.

En el capítulo se realizó un estudio de las tecnologías y tendencias actuales en cuanto a los sistemas de gestión de noticias. Se analizaron y se eligieron las principales herramientas, gestores de base de datos, metodologías de desarrollo y CMS para un mejor desarrollo del sistema de gestión de noticias.

Teniendo en cuenta el análisis realizado previamente, se toma como decisión desarrollar el sistema de gestión de noticias con el CMS Drupal, se propone además, para garantizar la realización de la aplicación de forma adecuada, el uso de la herramienta CASE Visual Paradigm, como metodología de desarrollo RUP, teniendo como IDE a Netbeans, lenguaje de programación del lado del servidor PHP, y utilizando las tecnologías del lado del cliente HTML, CSS y JavaScript; como gestor de base de datos MySQL.



CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.

Introducción

El capítulo está enmarcado en exponer las diferentes características del sistema propuesto con el único objetivo de cumplir con las expectativas del cliente y usuarios finales. Se explica el por qué se realizó un modelo de dominio. Se exponen los requisitos funcionales, no funcionales, los casos de uso del sistema y se realiza el análisis y diseño de lo que constituirá el sistema.

2.1 Modelo de dominio.

Como los procesos de negocios no están visibles claramente se opta por la realización de un modelo de dominio, el cual mediante la representación de un conjunto de conceptos y las relaciones que se establecen entre ellos ayudarán tanto a usuarios como a desarrolladores a usar un vocabulario común que les permita entender el contexto en que se emplaza el sistema a desarrollar.

2.1.1 Conceptos principales del modelo de dominio.

Usuario: Persona encargada de realizar la gestión de las noticias, la gestión de estadísticas y generación de reportes.

Reportes: Documento que en su estructura contiene aspectos relevantes de una noticia.

Noticia: Las noticias presentan una estructura definida que en su cuerpo incluye: título de la noticia, localización (URL), fecha de publicación, fuente de publicación, suministrado por, género, idioma, contenido.

Comentarios: Opinión de un usuario sobre una noticia publicada, consiste en leer primero el texto en cuestión para luego proceder a su valoración.

Estadísticas: Datos específicos de diferentes noticias que posibilitan la toma de decisiones.

Rol: Papel que desempeña un usuario o grupo en cualquier actividad.

Permisos: Autorizaciones que le serán concedidas a los diferentes roles, a través del cual se permita realizar solo las funcionalidades necesarias por los mismos.

Autor: Nombre de las personas que crearon las noticias.



Postura ideológica: Describe la noticia desde el punto de vista ideológico.

Descriptor: Palabra que describe la noticia.

País: País de la noticia.

Idioma: Idioma de la noticia.

Fuente: La fuente noticiosa donde se publicó la noticia.

Tipo fuente: Clasificación de las fuentes.

Género: Género de la noticia, ejemplo: noticia, artículo de opinión, entrevista.

Temáticas: Clasificación de la noticia sobre el tema que trata.

2.1.2 Diagrama de clases del dominio.

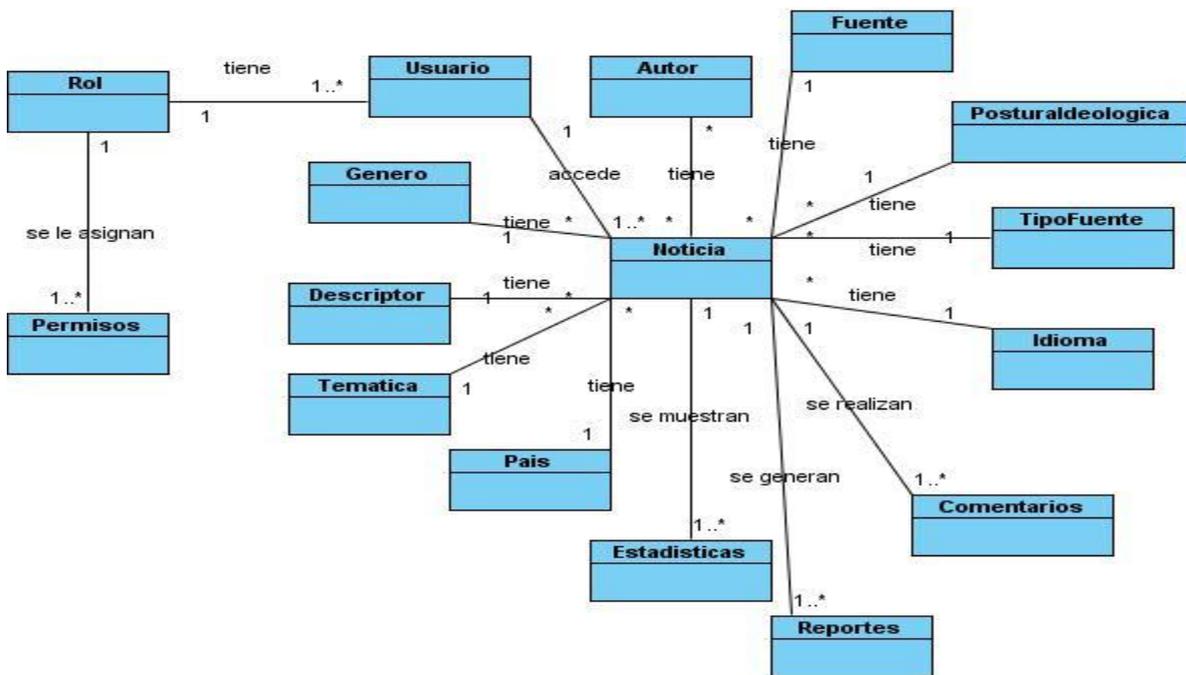


Imagen 4 Diagrama de clases del dominio.

2.1.3 Descripción del modelo de dominio.

El usuario es la persona encargada de acceder a las diferentes noticias, el mismo tiene un rol que lo identifica, a cuyo rol se le asignan permisos específicos para que solo pueda realizar las funcionalidades necesarias. A las noticias que accede el usuario se le realizan comentarios, se



generan reportes y se muestran estadísticas. Las noticias a su vez tienen: autor, fuente, tipo fuente, idioma, país, temática, descriptor y género.

2.1.4 Roles y permisos.

Los roles de usuarios en el sistema se establecen con el objetivo de proteger el flujo de contenidos. Son la base para establecer permisos y restricciones.

Administrador: Responsable del mantenimiento y futuras actualizaciones. Controla el flujo de contenidos de la aplicación. Es el encargado de gestionar los usuarios, los roles y los comentarios.

Editor: Es el que gestiona las noticias y los reportes que se les realizarán a las mismas.

Usuario: Puede realizar las siguientes operaciones: configurar su propio perfil, mostrar las diferentes estadísticas existentes, visualizar las noticias según una temática en específico y realizarle comentarios.

2.1.5 Reglas del negocio.

Las reglas del negocio son aquellas condiciones, validaciones y normas que se deben cumplir y controlar dentro de la aplicación y que son definidas de acuerdo con el comportamiento esperado de la misma.

En la realización del sistema se tendrán en cuenta algunas reglas que deben de cumplirse para un buen funcionamiento del mismo. Las reglas son las siguientes:

- ✓ Solo tiene acceso a la información el personal autorizado.

2.2 Especificación de los requisitos de software.

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema. (Jacobson, y otros, 2000)

Los requisitos de software se clasifican en 2 categorías: los requisitos funcionales y los requisitos no funcionales.



- ✓ Los requisitos funcionales: definen las funciones que el sistema será capaz de realizar.
- ✓ Los requisitos no funcionales: especifican las propiedades del sistema, tienen que ver con las características que de una forma u otra puedan limitar el sistema, como por ejemplo: restricciones del entorno o la implementación, el rendimiento (en tiempo y espacio), fiabilidad, seguridad, portabilidad y interfaces de usuarios.

2.2.1 Requisitos funcionales.

RF1: Autenticar usuario.

El sistema debe permitir que los usuarios se autenticuen introduciendo sus credenciales (usuario y contraseña).

RF2. Administrar usuario

RF2.1 Agregar un usuario.

El sistema debe permitir agregar un usuario especificando toda la información necesaria para ello.

RF2.2 Editar un usuario.

El sistema debe permitir editar la información persistente almacenada previamente de un usuario.

RF2.3 Eliminar un usuario.

El sistema debe permitir eliminar completamente un usuario del sistema y todas sus relaciones existentes con otros elementos del sistema.

RF3. Administrar rol.

RF3.1 Agregar rol.

El sistema debe permitir agregar un rol determinado.

RF3.2 Eliminar rol.

El sistema debe permitir eliminar completamente el rol del sistema.

RF3.3 Editar rol.

El sistema debe permitir editar el rol.

RF4. Administrar noticias.

RF4.1 Insertar una noticia.

Un usuario puede insertar una nueva noticia en un grupo mediante la opción “Nueva noticia”. Los campos para cada noticia son:



- ✓ Título de la noticia.
- ✓ Localización (URL).
- ✓ Fecha de publicación.
- ✓ Fuente de publicación.
- ✓ Tomado de (Opcional).
- ✓ Suministrado por.
- ✓ Género.
- ✓ Idioma.
- ✓ Contenido.
- ✓ Imagen (Opcional).
- ✓ Postura ideológica, (Opcional). Este campo no se utiliza en el caso del grupo de Noticias de Tecnología.
- ✓ Temática.
- ✓ Descriptor(es) (Opcional).
- ✓ Autor(es) (Opcional).
- ✓ Replicado de (Opcional).

RF4.2 Editar una noticia.

Se mostrará la opción de “Editar” al lado de cada noticia en la lista.

RF4.3 Eliminar una noticia.

Se mostrará la opción de “Eliminar” al lado de cada noticia en la lista. Se pedirá confirmación antes de ejecutar la acción.

RF5. Mostrar estadísticas por autor.

Se muestra la cantidad de autores insertados en la base de datos, sus nombres y para cada uno la cantidad de noticias que ha publicado.

RF6. Mostrar estadísticas por fuente.

Se muestra la cantidad de fuentes insertadas en la base de datos y la cantidad de noticias que han sido tomadas de ella.

RF 7 Generar reportes.

El sistema debe permitir generar reportes estadísticos.

RF 7.1 Mostrar reportes por día.

Después de escoger una fecha se mostrará:



- ✓ Cantidad de noticias para esa fecha.
- ✓ Cantidad de autores para esa fecha.
- ✓ Cantidad de fuentes para esa fecha.

Desglosado más abajo y en forma de tabla:

- ✓ Autores por cantidad de noticias.
- ✓ Fuentes por cantidad de noticias.

RF 7.2. Mostrar reportes por semana.

El usuario debe indicar el día que precise consultar y el sistema interpreta esa fecha como fecha final y antepone la fecha de siete días anteriores a la misma para establecer la semana. El sistema muestra por defecto la fecha actual en que se realiza la consulta.

Después de escoger una fecha determinada se mostrará:

- ✓ Cantidad de autores en esa fecha.
- ✓ Cantidad de fuentes en esa fecha.
- ✓ Cantidad de temáticas en esa fecha.

Desglosado más abajo y en forma de tabla:

- ✓ Los autores por cantidad de noticias
- ✓ Las fuentes por cantidad de noticias.
- ✓ Las temáticas por cantidad de noticias.

RF 7.3. Mostrar reportes por mes.

Después de escoger un mes se mostrará:

- ✓ Cantidad de noticias en esa fecha.
- ✓ Cantidad de autores en esa fecha.
- ✓ Cantidad de fuentes en esa fecha.
- ✓ Cantidad de temáticas en esa fecha.

Desglosado más abajo y en forma de tabla:

- ✓ Los autores por cantidad de noticias
- ✓ Las fuentes por cantidad de noticias.
- ✓ Las temáticas por cantidad de noticias.

RF 7.4. Mostrar reportes por año.

Después de escoger un año determinado se mostrará:

- ✓ Cantidad total de noticias en ese año.
- ✓ Cantidad total de autores.
- ✓ Cantidad total de fuentes.



- ✓ Temáticas más abordadas en el año.

RF 8. Mostrar todas las noticias de una temática.

El sistema debe permitir acceder a todas las noticias de una temática y mostrarlas ordenadas por fecha de publicación.

RF9. Configurar perfiles de usuario.

RF 9.1 Configurar datos personales

El sistema debe permitir configurar los datos personales del usuario.

RF 9.2 Configurar preferencias

El sistema debe permitir configurar las preferencias del usuario, para personalizar la aplicación a su complacencia.

RF10 Realizar comentarios a las noticias.

El sistema debe permitir a los usuarios poder realizar comentarios a las noticias.

RF 11. Administrar comentarios.

RF 11.1 Eliminar comentarios.

El sistema debe permitir eliminar los comentarios.

RF 11.2 Editar comentarios.

El sistema debe editar los comentarios.

RF 11.3 Insertar comentarios.

El sistema debe permitir insertar comentarios.

RF 12. Administrar países.

RF 12.1 Insertar país.

El sistema debe permitir insertar países.

RF 12.2 Editar país.

El sistema debe permitir editar los países.

RF 12.3 Eliminar país.

El sistema debe permitir eliminar países.

RF 13. Administrar géneros.

RF 13.1 Insertar género.

El sistema debe permitir insertar géneros.



RF 13.2 Editar género.

El sistema debe permitir editar los géneros.

RF 13.3 Eliminar género.

El sistema debe permitir eliminar géneros.

RF 14. Administrar autores.

RF 14.1 Insertar autor.

El sistema debe permitir insertar autores.

RF 14.2 Editar autor.

El sistema debe permitir editar los autores.

RF 14.3 Eliminar autor.

El sistema debe permitir eliminar autores.

RF 15. Administrar fuentes.

RF 15.1 Insertar fuente.

El sistema debe permitir insertar fuentes.

RF 15.2 Editar fuente.

El sistema debe permitir editar las fuentes.

RF 15.3 Eliminar fuente.

El sistema debe permitir eliminar las fuentes.

RF 16. Administrar temáticas.

RF 16.1 Insertar temática.

El sistema debe permitir insertar temáticas.

RF 16.2 Editar temática.

El sistema debe permitir editar las temáticas.

RF 16.3 Eliminar temática.

El sistema debe permitir eliminar temáticas.

RF 17. Administrar descriptores.

RF 17.1 Insertar descriptor.

El sistema debe permitir insertar descriptores.

RF 17.2 Editar descriptor.

El sistema debe permitir editar los descriptores.



RF 17.3 Eliminar descriptor.

El sistema debe permitir eliminar descriptores.

RF 18. Administrar tipo de fuente.

RF 18.1 Insertar tipo de fuente.

El sistema debe permitir insertar tipos de fuente.

RF 18.2 Editar tipo de fuente.

El sistema debe permitir editar los tipos de fuente.

RF 18.3 Eliminar tipo de fuente.

El sistema debe permitir eliminar tipos de fuente.

RF 19. Administrar idiomas.

RF 19.1 Insertar idioma.

El sistema debe permitir insertar idiomas.

RF 19.2 Editar idioma.

El sistema debe permitir editar los idiomas.

RF 19.3 Eliminar idioma.

El sistema debe permitir eliminar idiomas.

RF 20. Administrar postura ideológica.

RF 20.1 Insertar postura ideológica.

El sistema debe permitir insertar posturas ideológicas.

RF 20.2 Editar país.

El sistema debe permitir editar las posturas ideológicas.

RF 20.3 Eliminar país.

El sistema debe permitir eliminar posturas ideológicas.

2.2.2 Requisitos no funcionales.

Requisitos de apariencia o interfaz externa.

- ✓ El sistema mostrará el nombre del producto.
- ✓ Diseño sencillo y con una interfaz amigable para el usuario, permitiéndoles la fácil manipulación sin necesidad de adiestramiento.
- ✓ Interfaz compatible con los diferentes navegadores webs existentes.



- ✓ El sistema debe poderse ejecutar en máquinas con una resolución mínima de 800x600 y 32 bits de color y superiores.

Requisitos de usabilidad.

- ✓ El sistema será lo suficientemente sencillo como para ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web.

Requisitos de rendimiento.

- ✓ **Tiempo de respuesta:** Las consultas a la base de datos no serán de un nivel de complejidad alto y los niveles de concurrencia de usuarios que accedan al servidor de aplicaciones no serán muy elevados, el tiempo de respuesta de la aplicación será rápido, es decir no debe de exceder los 3 segundos.

Requisitos de Software.

- ✓ Sistema Operativo: Windows o Linux.
- ✓ Servidor: Apache.
- ✓ Cliente: Navegadores como el Internet Explorer, Firefox o Netscape.

Requisitos de Hardware.

- ✓ Servidor: Requisitos mínimos: 512 Mb RAM.
- ✓ Cliente: Requisitos mínimos: 128 Mb RAM.

Requisitos de portabilidad.

- ✓ El sistema podrá ser usado bajo cualquier sistema operativo ya sea Linux o Windows.

Requisitos de seguridad y privacidad

- ✓ Las funcionalidades del sistema serán mostradas de acuerdo con el rol que desempeña el usuario en el sistema, los módulos y los bloques pueden crear sus propias tablas en la base de datos, por lo que no hay riesgo de que la información se pierda.
- ✓ El sistema autentica a cada usuario para que sea identificado antes de realizar cualquier acción. Si no está registrado no lo dejará realizar ninguna actividad, con el propósito de proteger toda la información sensible a la visibilidad de determinados usuarios.

Requisitos de diseño e implementación.

- ✓ La herramienta para el desarrollo de la aplicación será Drupal 6.19.



- ✓ Utilizando PHP, HTML, CSS y JavaScript para el desarrollo del sistema.
- ✓ La arquitectura del sistema se realizó según la arquitectura definida por Drupal, arquitectura en capas.

2.3 Patrones de casos de uso a utilizar.

Utilizando los patrones de casos de uso, arquitectos, analistas, ingenieros, y gerentes pueden lograr mejores resultados de forma más rápida, en la realización de un sistema.

Múltiples actores: Roles comunes

En ocasiones sucede que varios actores juegan un mismo rol sobre un caso de uso, este rol es representado por otro actor, heredando de él los actores que jueguen su mismo rol sobre el caso de uso.

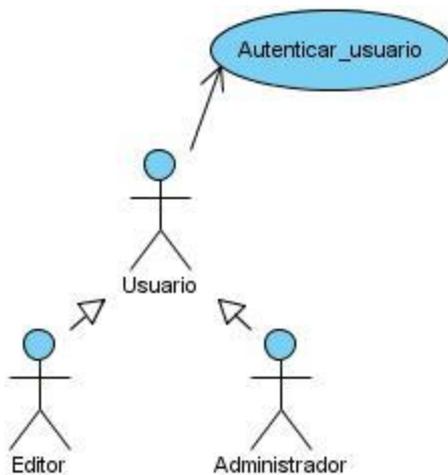


Imagen 5 Múltiples actores. Roles común.

CRUD Completo.

Este patrón consta de un caso de uso, llamado **Gestionar Información**. El mismo modela todas las operaciones que pueden ser realizadas sobre una información, tales como creación, modificación, lectura y eliminación.

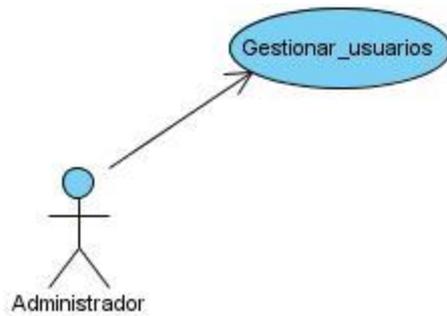


Imagen 6 CRUD Completo.

2.4 Descripción del sistema propuesto.

El sistema está destinado al grupo de Vigilancia Tecnológica y Política del proyecto Grupo de Investigación y Desarrollo sobre Internet (GIDI). El Sistema de Gestión de Noticias de Cuba en la Prensa Internacional es una aplicación web donde sus usuarios tienen determinados roles y de acuerdo a dichos roles podrán realizar determinadas funcionalidades, como por ejemplo:

- ✓ Los usuarios del rol editor podrán gestionar las noticias.
- ✓ Los usuarios del rol administrador podrán administrar el sistema en general.
- ✓ Los usuarios de rol usuario solo podrán realizar las acciones de visualización de las noticias, comentarlas, así como ver las estadísticas existentes y los tipos de reportes.

La aplicación permitirá que los miembros del Grupo de Vigilancia tecnológica y Política puedan almacenar, editar y eliminar noticias obtenidas durante el proceso de recopilación de noticias en los diferentes medios noticiosos existentes en la web referente a Cuba. Para su posterior análisis por parte de la dirección o de los analistas del proyecto.

2.5 Modelo del sistema. Definición de actores y casos de usos del sistema.

2.5.1 Definición de los actores del sistema.

Un actor del sistema es una entidad externa que va a interactuar con el sistema.

Los actores del sistema:

- ✓ No son parte de él.
- ✓ Pueden intercambiar información con él.
- ✓ Pueden ser un recipiente pasivo de información.



- ✓ Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. (UCI, 2010-2011)



Imagen 7 Actor del sistema.

Actor	Justificación
Usuario	Actor genérico que garantiza la autenticación de todos los usuarios del sistema.
Administrador	Actor que se encarga de la administración total del sistema.
Editor	Actor que se encarga de trabajar con las noticias.

Tabla 2 Definición de actores del sistema.

2.5.2 Definición de casos de usos del sistema.

Los casos de uso (CU) son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. (UCI, 2010-2011)

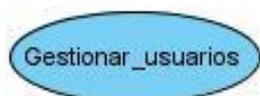


Imagen 8 Caso de uso del sistema.

Definición del CU Autenticar.

Caso de Uso	Autenticar usuario
Actores	Usuario
Descripción	El caso de uso autenticar le permite autenticarse a los usuarios para interactuar con el sistema.
Referencia	RF 1

Tabla 3 Definición del CU Autenticar.

Definición del CU Gestionar usuarios.

Caso de Uso	Gestionar usuarios
Actores	Administrador



Descripción	Este caso de uso permite al administrador gestionar todas las cuentas de los usuarios que tienen permiso para acceder al sistema.
Referencia	RF 2 completo.

Tabla 4 Definición del CU Gestionar usuarios.

Definición del CU Gestionar rol.

Caso de Uso	Gestionar rol
Actores	Administrador
Descripción	Este caso de uso permite al administrador gestionar todos los roles que le serán asignados a los usuarios del sistema.
Referencia	RF 3 completo.

Tabla 5 Definición del CU Gestionar rol.

Definición del CU Gestionar noticias.

Caso de Uso	Gestionar noticias
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar todas las noticias.
Referencia	RF 4 completo.

Tabla 6 Definición del CU Gestionar noticias.

Definición del CU Realizar comentarios.

Caso de Uso	Realizar comentarios
Actores	Usuario
Descripción	Este caso de uso permite a los usuarios realizar comentarios a las noticias.
Referencia	RF 10.

Tabla 7 Definición del CU Realizar comentarios.

Definición del CU Mostrar noticias de una temática.

Caso de Uso	Mostrar noticias de una temática.
Actores	Usuario
Descripción	Este caso de uso permite al usuario mostrar todas las noticias existentes según una temática.
Referencia	RF 8.

Tabla 8 Definición del CU Mostrar noticias de una temática.

Definición del CU Configurar perfil.

Caso de Uso	Configurar perfil.
Actores	Usuario
Descripción	Este caso de uso permite al usuario configurar su perfil, realizarle cambios a sus datos personales.
Referencia	RF 9 completo.

Tabla 9 Definición del CU Configurar perfil.



Definición del CU Mostrar estadísticas por autor.

Caso de Uso	Mostrar estadísticas por autor
Actores	Usuario
Descripción	Este caso de uso permite al usuario mostrar estadísticas de las noticias según el autor de las mismas.
Referencia	RF 5.

Tabla 10 Definición del CU Mostrar estadísticas por autor.

Definición del CU Mostrar estadísticas por fuente.

Caso de Uso	Mostrar estadísticas por fuente
Actores	Usuario
Descripción	Este caso de uso permite al usuario mostrar estadísticas de las noticias según la fuente.
Referencia	RF 6.

Tabla 11 Definición del CU Mostrar estadísticas por fuente.

Definición del CU Gestionar comentarios.

Caso de Uso	Gestionar comentarios
Actores	Administrador
Descripción	Este caso de uso permite al administrador gestionar los comentarios que se les realicen a las noticias.
Referencia	RF 11 completo.

Tabla 12 Definición del CU Gestionar comentarios.

Definición del CU Generar reportes

Caso de Uso	Generar reportes
Actores	Editor
Descripción	Este caso de uso permite al editor generar reportes estadísticos diarios, semanales, mensuales y anuales.
Referencia	RF 7 completo.

Tabla 13 Definición del CU Generar reportes.

Definición del CU Gestionar país.

Caso de Uso	Gestionar país
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar los países, los cuales serán utilizados para la creación de una noticia.
Referencia	RF completo.

Tabla 14 Definición del CU Gestionar país.

Definición del CU Gestionar género.

Caso de Uso	Gestionar género
Actores	Editor



Descripción	Este caso de uso permite al editor gestionar los géneros, los cuales serán utilizados para la creación de una noticia.
Referencia	RF completo.

Tabla 15 Definición del CU Gestionar género.

Definición del CU Gestionar autores.

Caso de Uso	Gestionar autores
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar los autores, los cuales serán utilizados para la creación de una noticia.
Referencia	RF completo.

Tabla 16 Definición del CU Gestionar autores.

Definición del CU Gestionar fuentes.

Caso de Uso	Gestionar fuentes
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar las fuentes de publicación de las noticias.
Referencia	RF completo.

Tabla 17 Definición del CU Gestionar fuentes.

Definición del CU Gestionar temáticas.

Caso de Uso	Gestionar temáticas
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar las temáticas, las cuales serán utilizadas para la creación de una noticia.
Referencia	RF completo.

Tabla 18 Definición del CU Gestionar temáticas.

Definición del CU Gestionar descriptores.

Caso de Uso	Gestionar descriptores
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar los países, los cuales serán utilizados para la creación de una noticia.
Referencia	RF completo.

Tabla 19 Definición del CU Gestionar descriptores.

Definición del CU Gestionar tipo fuente.

Caso de Uso	Gestionar tipo fuente
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar los tipos de fuente de publicación de las noticias.
Referencia	RF completo.

Tabla 20 Definición del CU Gestionar tipo fuente.



Definición del CU Gestionar postura ideológica.

Caso de Uso	Gestionar postura ideológica
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar las diferentes posturas ideológicas.
Referencia	RF completo.

Tabla 21 Definición del CU Gestionar postura ideológica.

Definición del CU Gestionar fuente.

Caso de Uso	Gestionar fuente
Actores	Editor
Descripción	Este caso de uso permite al editor gestionar las fuentes de publicación de las noticias.
Referencia	RF completo.

Tabla 22 Definición del CU Gestionar fuente.

Remitirse al documento *Modelo de casos de uso del sistema* para poder obtener una descripción detallada de los mismos.

2.5.3 Diagrama de casos de uso del sistema.

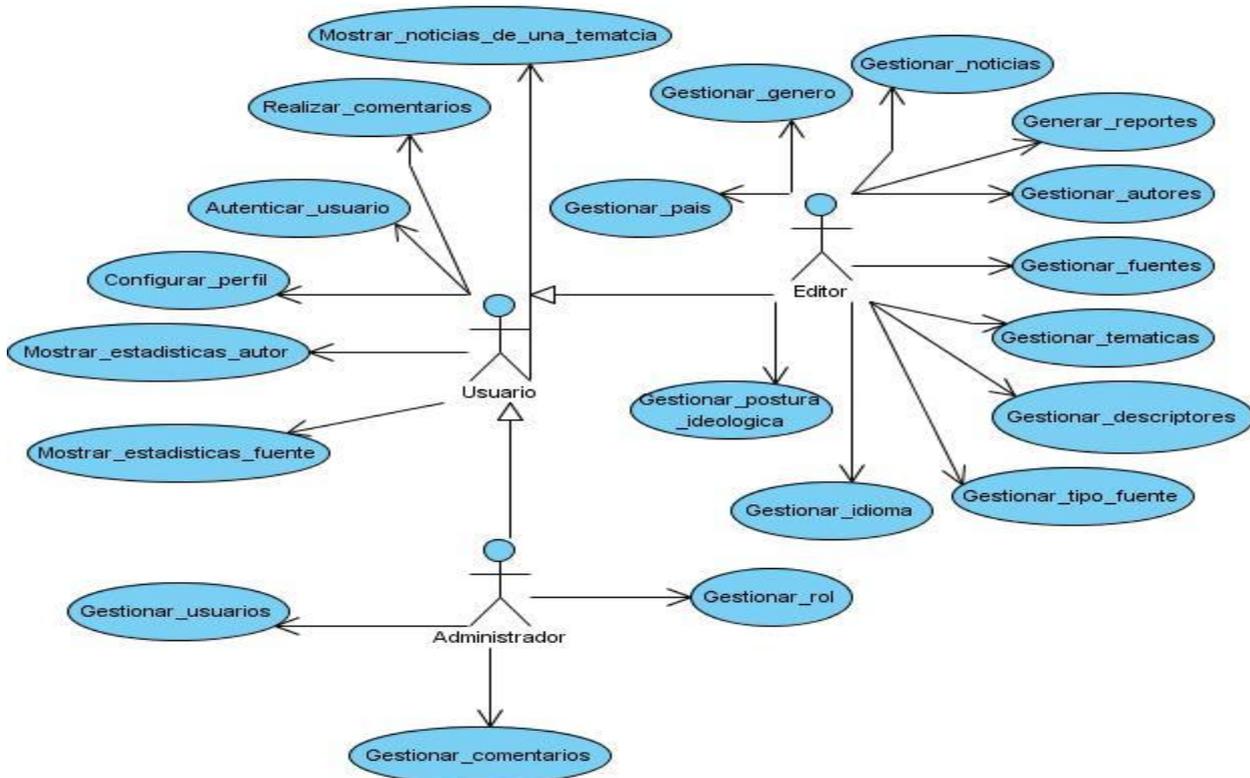


Imagen 9 Diagrama de CU del sistema.



2.6 Modelo de Análisis

Con el análisis se obtiene una mejor visión del sistema a implementar ya que este está centrado en los requisitos funcionales que se definieron anteriormente. Además representa la relación entre los actores del sistema y las clases de análisis.

El modelo de análisis es un modelo conceptual, ya que en él se identifican una serie de clases y relaciones que conectadas entre sí, permiten una mejor comprensión de los requisitos de la aplicación que se está modelando.

2.6.1 Diagramas de clases del análisis.

Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar las clases en: clase de interfaz, clase controladora y clase entidad.

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. (Informáticas, 2010-2011)

Para ver los restantes diagramas de clases del análisis remitirse al documento *Modelo de análisis*.



Imagen 10 Diagrama de clases del análisis del CU Autenticar usuario.

2.6.2 Diagramas de interacción.

Los diagramas de interacción tienen diferentes formas, basadas todas ellas en una misma información subyacente pero resaltando cada una un punto de vista de la misma: diagramas de secuencia, diagramas de colaboración.

Para ver los diferentes diagramas de interacción remitirse al documento *Modelo de análisis*.

Diagramas de colaboración



Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que preceden a las descripciones del mensaje.

Diagramas de secuencia.

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo de la página. La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.

Se muestra un mensaje como una flecha desde la línea de vida de un objeto a la del otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo. (Informáticas, 2010-2011)

2.7 Modelo de Diseño.

El Diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida, y crear un plano del modelo de implementación.

En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema. (UCIISW)

2.7.1 Diagramas de clases del diseño.

El diagrama de clases se agrupó en paquetes para dividir el modelo en partes manejables mediante la agrupación de las diferentes clases. Teniéndose 4 paquetes generales, que los mismos en su interior tienen varios paquetes. Para entender el siguiente diagrama de clases, primeramente es necesario saber cuál es la funcionalidad de cada paquete.



- ✓ **Sistema de Gestión de Noticias**, agrupa todos los casos de uso del sistema, mediante los cuales es posible darle cumplimiento a las funcionalidades requeridas para que el sistema cumpla con las expectativas del cliente.

- ✓ **Drupal Core**, conforma el núcleo de Drupal, agrupando los siguientes subpaquetes:

Theme: Genera la interfaz gráfica que le proporciona el Core de Drupal y cada uno de los módulos utilizados, procesa la información y la visualiza en el sitio.

Form: Integra las funciones para generar y controlar dinámicamente todos los formularios de la aplicación.

Database: Es la capa de acceso a datos, tramita el acceso y las acciones realizadas sobre la base de datos del sistema.

User: Aquí es donde se gestionan los usuarios y roles que tendrán los mismos, así como la autenticación y sus permisos.

Node: En los nodos en donde se almacena toda la información del sistema, provee una plataforma base para el proceso de gestión de contenidos, que puede ser extendida por otros módulos.

Block: Permite configurar el contenido que aparece en bloques localizados en las barras laterales del sitio y otras regiones.

- ✓ **Drupal Contrib**, en este paquete están agrupados todos los módulos que se le adicionaron al sistema con el fin de incluir nuevas funcionalidades. Ver la sección módulos contribuidos de Drupal del capítulo 3 de este documento.
- ✓ **Módulos implementados**, para darle cumplimiento a determinadas funcionalidades fue necesario la implementación de 2 módulos, estadísticas y reportes. Para observar una descripción detallada de los mismos remitirse al epígrafe 3.5.

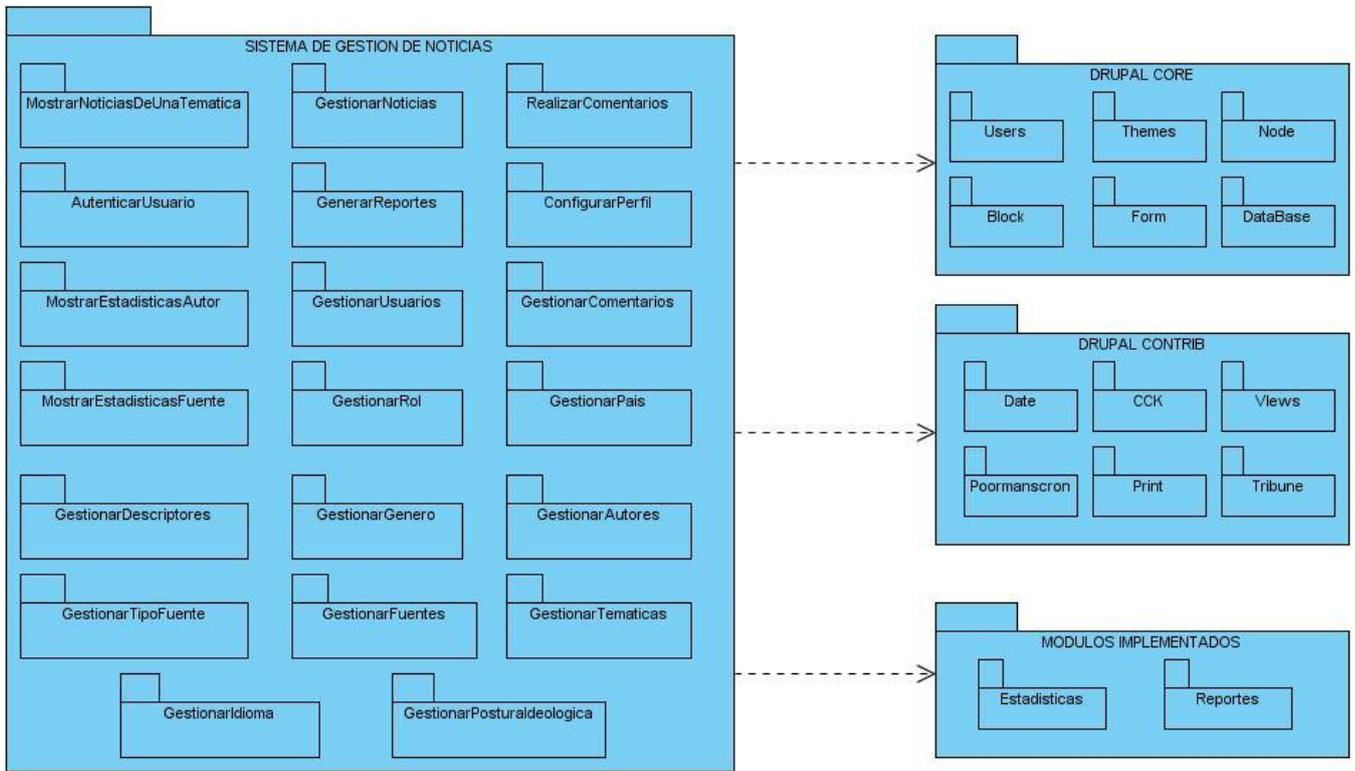


Imagen 11 Diagrama de clases, agrupado por paquetes.

Diagramas de clases del diseño web.

Para ver los restantes diagramas del diseño web, remitirse al documento *Modelo de diseño*.

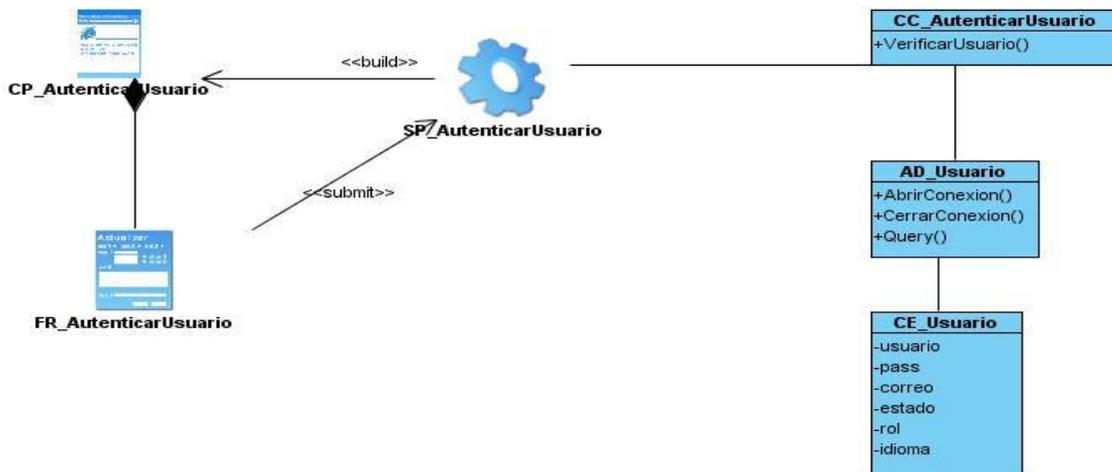


Imagen 12 Diagrama de clases del diseño web del CU Autenticar usuario.

2.8 Visión general de la arquitectura de software y patrones utilizados por Drupal.



2.8.1 Arquitectura de Software.

Drupal utiliza una arquitectura flexible basada en la abstracción y organización en capas para conseguir flexibilidad y facilidad a la hora de realizar sitios web. En la siguiente figura se muestra el flujo de contenidos de Drupal.

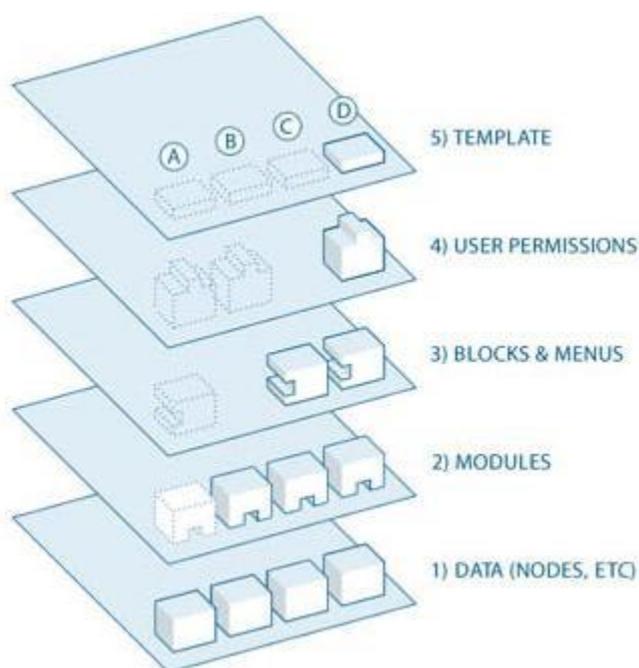


Imagen 13 Arquitectura de Drupal. (Drupal)

Nodos: son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio web crece, lo va haciendo el número de nodos los cuales van formando un “depósito de nodos” cada vez mayor.

Módulos: son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. Son como Plug-Ins que se instalan en el sitio web proporcionándole nuevas funcionalidades.

Bloques y menús: Estos permiten estructurar y organizar los contenidos en la página web. Es decir que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

Usuarios y permisos: Drupal dispone de un registro de usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario. Es decir que las



operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal.

Temas: es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio web sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que queremos es renovar la apariencia de un sitio web.

2.8.2 Patrones utilizados por Drupal.

Drupal, a pesar de no utilizar muchas de las características de la Programación Orientada a Objetos (POO), muchos paradigmas orientados a objetos se utilizan aún en su diseño. El diseño de Drupal está conformado a través de una orientación a objetos, mostrando características como las que se describen a continuación:

- ✓ **Objetos:** Algunos de los más prominentes componentes de Drupal que podrían ser considerados objetos son los módulos, temas, nodos, y los usuarios.
- ✓ **Abstracción:** El sistema de ganchos de Drupal (los hooks) definen las operaciones que se pueden realizar en o por un módulo. Si un módulo implementa un hook, se entra en un contrato para realizar una tarea en particular cuando el hook se invoca. El código no necesita saber nada sino acerca del módulo o la forma en que el hook se aplica con el fin de obtener un trabajo útil hecho invocando el hook.
- ✓ **Encapsulación:** Como la mayoría de otros sistemas orientados a objetos, Drupal no tiene estrictamente una forma de limitar el acceso a un objeto del funcionamiento interno, sino que se basa en la convención para lograr esto. El código de Drupal se basa alrededor de las funciones, que comparten un mismo nombre, este nombre está subdividido por el uso de prefijos. Siguiendo esta simple convención, cada módulo puede declarar sus propias funciones y variables sin la preocupación de los conflictos con los demás.



- ✓ **Polimorfismo:** Los nodos son polimórficos en el sentido clásico. Los temas son polimórficos de la misma manera, el tema se pasa a un mensaje y responde a ella de una manera diferente dependiendo de la aplicación del tema activo, aunque la interfaz es constante.
- ✓ **Herencia:** Los módulos y los temas pueden definir funciones. Sin embargo, puede ser pensado a heredar su comportamiento de una clase base abstracta. En el caso de los temas, el comportamiento de esta clase está determinada por las funciones de theme.inc, si un tema no anula existe una función definida, por defecto la versión de un componente de interfaz se utiliza, pero el tema puede prever su propia renderizado. Los módulos tienen el mismo modo de selección de todos los ganchos de Drupal para anular a voluntad, y puede seleccionar y elegir los que se apliquen. (Martín)

Drupal también hace uso de patrones de diseño tales como: *Singleton*, *Observer*, *Bridge*, *Chain of Responsibility* y *el Command*. Por solo especificar uno, el caso del patrón Singleton: si hemos de pensar de los módulos y los temas como objetos, y luego siguen el patrón singleton. En general, estos objetos no encapsulan datos, lo que separa a un módulo desde otro es el conjunto de funciones que contiene, por lo que debe ser pensado como una clase con un ejemplo aislado. (Martín)

2.9 Diseño de la Base de Datos.

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos. Si las tablas no son definidas apropiadamente, se podrían presentar muchos problemas al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de información. No importa si la base de datos tiene sólo 20 registros, o algunos cuantos miles, es importante asegurarse que la base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo tiempo.

Una base de datos bien diseñada permite obtener acceso a la información exacta y actualizada, por lo que una estructura adecuada es esencial para lograr los objetivos fijados para el sistema en cuestión. Independientemente de que la propuesta está basada en el CMS Drupal, el cual consta con una base de datos ya diseñada y probada por una comunidad internacional de varios miles de usuarios, se hará una representación de los principales aspectos para lograr un mejor entendimiento del funcionamiento de la aplicación.



Para lograr un diseño de base de datos apropiado y eficiente, el mismo debe cumplir con las siguientes cualidades:

- ✓ Reflejar la estructura del problema en el mundo real.
- ✓ Ser capaz de representar todos los datos esperados, incluso con el paso del tiempo.
- ✓ Evitar el almacenamiento de información redundante.
- ✓ Proporcionar un acceso eficaz a los datos.
- ✓ Mantener la integridad de los datos a lo largo del tiempo.
- ✓ Ser claro, coherente y de fácil comprensión. (Grayson, 2002)

En el diagrama de clases persistentes solo se contemplan las entidades de significativa relevancia para la aplicación, o sea, las clases que van a persistir en el tiempo. No será necesario el análisis de todas las entidades pues el CMS se encarga de la gestión de las mismas. Para diseñar la Base de Datos es necesario primeramente realizar el diagrama de clases persistentes con sus atributos y relaciones.

2.9.1 Diagrama de clases persistentes.

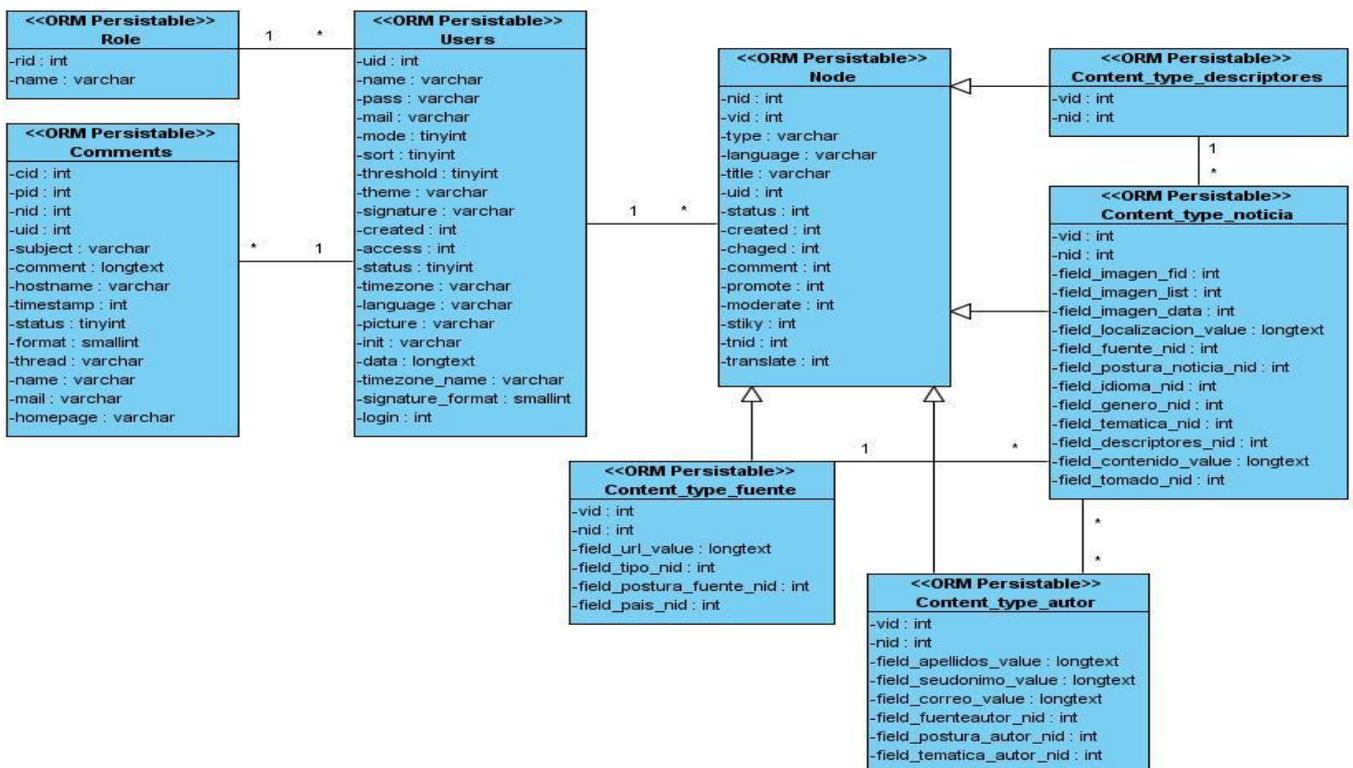


Imagen 14 Diagrama de clases persistentes



2.9.2 Modelo de datos de clases persistentes.

A partir del diagrama de clases persistentes se define el modelo de datos que se muestra a continuación.

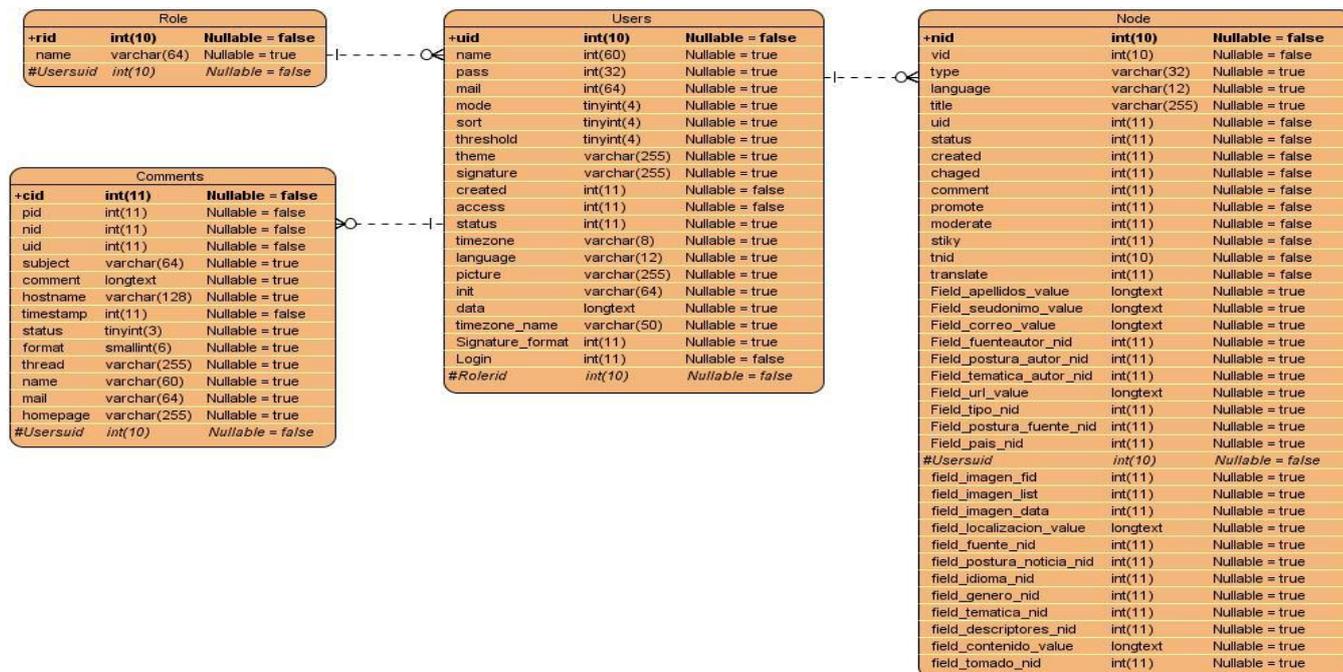


Imagen 15 Modelo de datos

Conclusiones parciales del capítulo.

El análisis y diseño de un software brinda la primera visión de lo que pudiera ser la solución en el desarrollo del mismo. En este capítulo se han mostrado las funcionalidades que tendrá el sistema que se describe en el presente trabajo de diploma, además de las reglas del negocio que se deberán tener en cuenta, los roles y permisos.

Se ha desarrollado el modelado del diagrama de casos de uso del sistema utilizando los patrones correspondientes, la descripción detallada de cada uno de los mismos en un lenguaje sencillo y su clasificación según la prioridad requerida.

Además se desarrolló el análisis y diseño de la solución propuesta obteniéndose los diagramas de clases del análisis y del diseño. También se conoció el modelo de datos a utilizar en el sistema. El resultado del capítulo posibilitará la guía para la futura implementación de la aplicación.



CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA.

Introducción

El presente capítulo contempla la implementación del Sistema de Gestión de Noticias de Cuba en la Prensa Internacional integrado al CMS Drupal. Se realizará el modelo de implementación para una mejor descripción de la solución propuesta. Se describen los usuarios, roles y privilegios de la política de acceso al sistema. Además, se realizan las pruebas de caja blanca y caja negra al sistema desarrollado

3.1 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Pueden ser utilizados para modelar sistema de software de cualquier tamaño y complejidad.

Para un mejor entendimiento del diagrama de componentes de la aplicación desarrollada se describen mediante una tabla los aspectos del mismo.

Componente	Descripción
Themes	Son los temas que se pueden utilizar en la aplicación.
Misc	Aquí se encuentran una serie de archivos Javascript y CSS necesarios para el funcionamiento del sistema.
Modules	Aquí se localizan todos los módulos de Drupal.
Includes	Aquí se encuentran funciones y scripts necesarios para el funcionamiento de los módulos y el sistema Drupal, incluyendo las APIs de conexión a base datos.
Base de datos	Es la base datos de Drupal, dentro de la cual se incluyen las tablas del Sistema de Gestión de



	Noticias.
Estadísticas	Módulo para la generación de estadísticas en cuanto a la fuente de publicación y los autores de las noticias.
Reportes	Módulo para realizar reportes estadísticos diarios, semanales, mensuales y anuales.

Tabla 23 Descripción de los componentes.

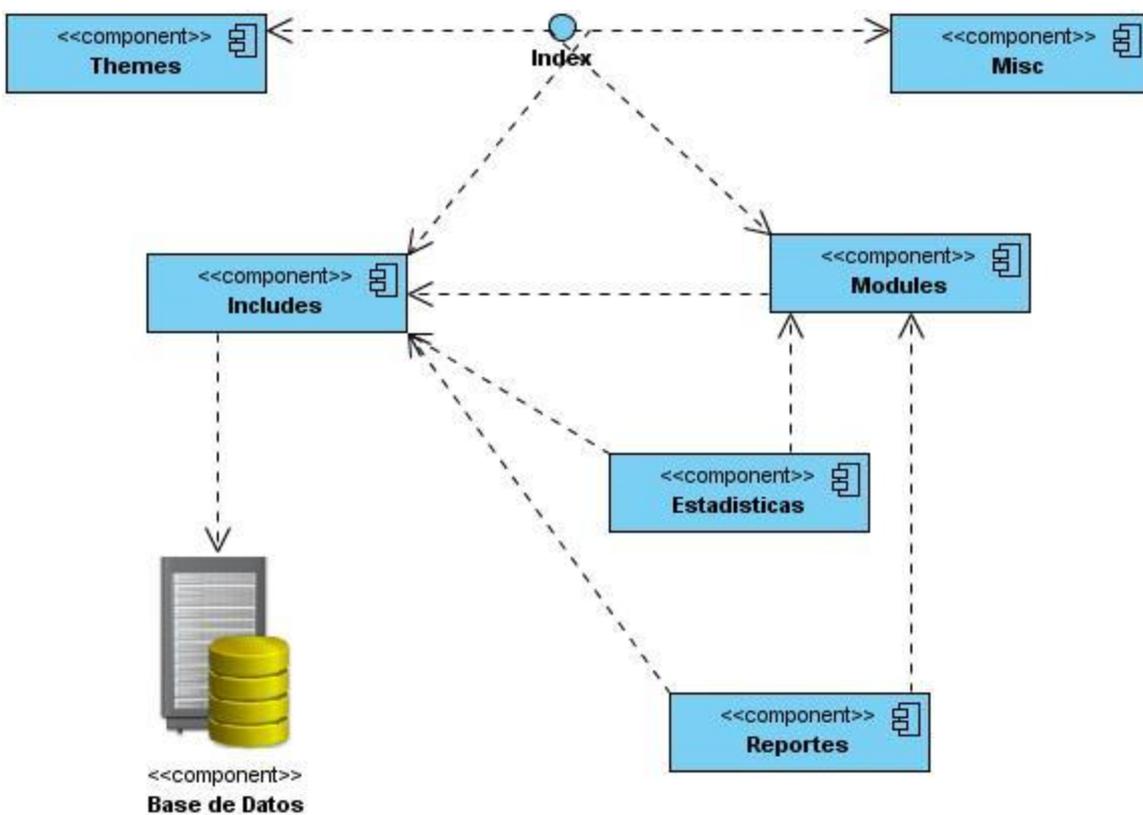


Imagen 16 Diagrama de componentes

3.2 Diagrama de despliegue.

Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.



Para acceder al Sistema de Gestión de Noticias, las PC Clientes utilizan un navegador web, las mismas tiene conectado una impresora para si el usuario desea imprimir las noticias. La comunicación entre las PC clientes y el servidor web se establece utilizando el protocolo HTTPS y el servidor web hace peticiones mediante TCP/IP a la base de datos.

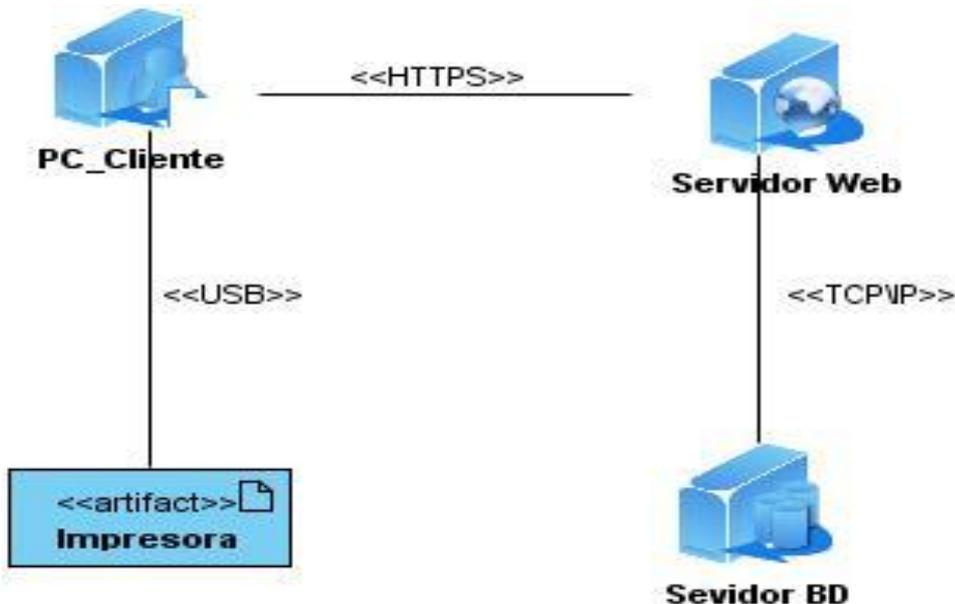


Imagen 17 Diagrama de despliegue

3.2.1 Descripción de los nodos.

El nodo **Servidor de Base de Datos** representa un servidor MySQL, en el cual se ubica toda la información persistente del sistema, almacenándose los datos que son actualizados y consultados por los usuarios del sistema.

En el **Servidor web**, se ubican íntegramente las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.

La **PC cliente** representa el conjunto de computadoras a través de las cuales los usuarios pueden actualizar y consultar la información que se encuentra en el servidor web.

La **impresora** representa un artefacto el cual será utilizado para imprimir los reportes o estadísticas generadas por el sistema.



3.2.2 Protocolos.

- ✓ **TCP/IP:** Se utiliza en la comunicación entre el servidor y la base de datos para realizar operaciones sobre la información de las tablas.
- ✓ **HTTPS** (Protocolo de transferencia de hipertexto seguro): Establece un esquema de comunicación cliente – servidor. El cliente es el navegador web que realiza las peticiones a las que el servidor se encarga de dar respuesta.

3.3 Usuarios y permisos.

Los usuarios son creados para mantener el control de las personas que acceden al sistema por lo que a cada uno de ellos se le asignan roles y privilegios a la hora de manejar la información que se encuentra en el sistema, la tabla muestra a continuación los usuarios que participan en el sistema y sus respectivos roles.

Usuarios	Roles	Permisos	Descripción
Administrador	Administrador	Control total.	Administra el sistema en general.
Editor	Editor	<ul style="list-style-type: none">- Acceder a los comentarios.- Publicar comentarios sin aprobación.- Acceso al formulario de contacto del sitio.- Observar y editar los campos de los tipos de contenidos.- Permitir el acceso a sus contenidos y a la de los demás.- Acceder al contenido.- Administrar los tipos de contenidos.- Acceder al módulo print.- Poder convertir las noticias a formato PDF.- Buscar contenido y usar la búsqueda avanzada.	Control en cuanto a la gestión de noticias, la generación de reportes y otras funcionalidades del sitio.



		<ul style="list-style-type: none"> - Acceder a la sala de chat, a su historial y poder enviar mensajes. - Acceder a perfiles de usuarios y cambiar su nombre propio. - Acceder a las diferentes vistas del sitio. - Acceder a los reportes. - Acceder a las estadísticas. 	
Usuario	Usuario	<ul style="list-style-type: none"> - Acceder a los comentarios. - Publicar comentarios. - Acceder al formulario de contactos del sitio. - Permitir el acceso a los contenidos. - Acceder al contenido. - Acceder al módulo print. - Poder convertir las noticias a formato PDF. - Buscar contenido y usar la búsqueda avanzada. - Acceder a la sala de chat, a su historial y poder enviar mensajes. - Acceder a perfiles de usuarios y cambiar su nombre propio. - Acceder a los reportes. - Acceder a las estadísticas. 	Sólo podrá ver los contenidos publicados, realizarle comentarios a los mismos, ver las estadísticas, los reportes y participar en el chat.

Tabla 24 Usuarios, roles y permisos

3.4 Análisis de posibles implementaciones (reutilización de código).

La reutilización de código es un aspecto de suma importancia cuando se trata de reducir el tiempo de desarrollo. Consiste en la utilización de programas o fragmentos de códigos ya existentes, que pueden fortalecer o desempeñar una funcionalidad en un producto determinado.



3.4.1 Módulos contribuidos de Drupal.

Los módulos contribuidos de Drupal son programas o conjunto de programas que extienden, modifican o agregan una funcionalidad al núcleo de Drupal. A continuación se detallan algunos módulos contribuidos utilizados en la aplicación.

Content Construction Kit (CCK): Integra todos los módulos destinados a la creación de tipos de contenidos. Con este módulo es posible definir los tipos de contenidos que tendrá el sitio, así como agregarle nuevos campos para ampliar las funcionalidades de los mismos. Además, posibilita establecer permisos a los tipos de contenido, basándose en los roles existentes en la aplicación.

Date: Permite agregar campos de tipo fecha y hora a los tipos de contenidos. Facilita el uso de calendarios emergentes para estos campos y la validación de fechas. Brinda la posibilidad de agregarle un calendario a la aplicación.

Poormanscron: Permite la ejecución automática del cron, el cuál es una aplicación que se ejecuta periódicamente con la finalidad de realizar las tareas básicas de mantenimiento del sitio web, como limpiar los registros, indexar los nuevos contenidos.

Print: Permite exportar los contenidos a formato PDF con el uso de librerías como DOMPDPF⁷, añade un enlace de versión para impresora a las páginas con contenido y de administración.

Views: Permite realizar lecturas en la base de datos de la aplicación y listar contenidos según lo requiera el usuario.

Tribune: Es un tipo de sala de chat, usado para la discusión en tiempo real de algún tema en específico. Está completamente basado en la web y no depende de ningún servicio externo.

3.5 Módulos implementados.

Para cumplir con determinadas funcionalidades se hizo necesaria la implementación de algunos módulos, los cuáles se describen a continuación:

Estadísticas: Módulo para mostrar estadísticas en cuanto a autores y fuentes de publicación.

⁷ La DOMPDPF es una librería escrita en PHP y destinada a convertir del formato HTML a PDF.



Reportes: Da la posibilidad a los usuarios de realizar reportes estadísticos sobre las noticias, estos reportes pueden ser: diarios, semanales, mensuales y anuales.

3.5 Pruebas.

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requisitos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (ISW)

A la aplicación desarrollada se le realizaron un conjunto de pruebas de caja blanca y caja negra con el objetivo de evaluar la calidad del producto desarrollado y garantizar que la aplicación cumpla con las funcionalidades. Para ver en detalles las mismas remitirse a los documentos de *Diseño de casos de prueba*.

Para la realización de las pruebas de caja negra se tuvieron en cuenta los casos de uso de mayor importancia, que necesitan la entrada de datos y que cada uno está dividido en tantas secciones como escenarios tenga el caso de uso, incluyendo sus flujos alternos, dentro de estos casos de uso se encuentran: gestionar usuarios, gestionar rol y gestionar noticias.

Se pudo comprobar con la ejecución de estos casos de prueba que las funciones del software son operativas, que las entradas realizadas se aceptan de forma adecuada y que se produce una salida correcta.

Al sistema también se le realizaron pruebas de caja blanca, que son pruebas con acceso al código, comprobándose los caminos lógicos del software de forma satisfactoria.



Se llevaron además a cabo pruebas de seguridad con el objetivo de asegurar que el sistema sólo fuera accedido por los actores definidos, se logró comprobar que el editor y el administrador son los únicos que pueden realizar trabajo con las noticias. A su vez el usuario sólo puede acceder a las funcionalidades correspondientes a los casos de usos inicializados por él en el diagrama de CU del sistema.

El proceso de pruebas del sistema se realizó en dos iteraciones, ya que en la primera iteración se detectó una no conformidad referente a la internacionalización del idioma, dándole solución en la segunda iteración y el sistema encontrándose listo para su explotación.

Conclusiones parciales del capítulo.

En el presente capítulo se elaboraron diagramas del flujo de trabajo de implementación que permiten entender de una manera concisa la construcción de la aplicación. El administrador, el usuario y el editor son los usuarios que tienen acceso sobre la aplicación y a los cuáles se les asignaron determinados privilegios para garantizar la seguridad del sistema. Además se realizaron pruebas a dicho sistema lo que permitió verificar su buen funcionamiento.



CONCLUSIONES

Con el desarrollo del sistema de gestión de noticias, se dio respuesta a la necesidad de contar con una herramienta informática que contribuyera a la toma de decisiones. Se puede plantear que la investigación cumplió con los objetivos trazados cuando se comenzó la misma. Lo que se sustenta en las siguientes razones:

- ✓ El estudio de las principales tecnologías actuales permitió escoger las herramientas factibles para el diseño e implementación del sistema.
- ✓ Con la utilización del CMS Drupal para la realización de la propuesta se logró que la misma fuera flexible y fácil de mantener, permitiendo alta adaptabilidad a la hora de incorporar nuevos módulos.
- ✓ La realización de las diferentes pruebas hizo posible que la propuesta de solución contara con la calidad requerida.



RECOMENDACIONES.

Teniendo en cuenta que la solución propuesta para este Trabajo de Diploma es una versión inicial se recomienda dar continuidad al trabajo basándose en los siguientes aspectos:

- ✓ Desarrollar nuevas versiones donde se incluyan mejoras al diseño de interfaz propuesto para dicha aplicación.
- ✓ Incluir funcionalidades que permitan la generación de boletines.
- ✓ Emplear información como nombres y apellidos y foto del usuario, proporcionada por el servidor LDAP de la UCI, para incorporarla a los datos de usuarios.
- ✓ Realizar un manual de usuario y de administración.
- ✓ Validar la aplicación desarrollada por Calidad UCI.



REFERENCIAS BIBLIOGRÁFICAS.

Avila, M. P. (2009). Recuperado el 13 de junio de 2011, de <http://dspace.esPOCH.edu.ec/bitstream/123456789/101/1/18T00378.pdf>

ANONIMOa. Ventajas y desventajas del Personal Home Page. [En línea] [Citado el: 5 de Diciembre de 2010.] <http://www.creargratisunapaginaweb.com/PHP/Ventajas-y-desventajas-del-Personal-Home-Page-4/>.

Anónimob. Taringa. [En línea] [Citado el: 5 de Diciembre de 2010.] http://www.taringa.net/posts/downloads/1481705/Visual-Paradigm_-una-herramienta-de-lo-mas-util.html.

Ciberaula. [En línea] [Citado el: 15 de Enero de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

Cibernetia. [En línea] [Citado el: 14 de Enero de 2011.] http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.

Cuerda, Xavier García. 2004. Mosaic. [En línea] 29 de Noviembre de 2004. [Citado el: 24 de Noviembre de 2010.] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>.

Drupal. [En línea] [Citado el: 15 de febrero de 2011.] <http://www.cursosdrupal.com/content/arquitectura>.

Giraldo, Luis y Zapata, Yuliana. 2005. *Herramientas de Desarrollo de ISW para Linux.* 2005.

Hinostroza, Raul Rodas. 2007. LinuxCentro.net. [En línea] 22 de Febrero de 2007. [Citado el: 1 de Diciembre de 2010.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

Informática, Sub-Jefatura de. 1999. *Herramientas Case.* 1999.

integrado, Entornos de desarrollo. [En línea] [Citado el: 13 de Enero de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.



ISW, Departamento. eva. [En línea] [Citado el: 18 de mayo de 2011.]

ISWII. Eva. [En línea] [Citado el: 20 de mayo de 2011.] http://eva.uci.cu/file.php/259/Curso_2010-2011/Semana_11/Clase_Practica_7/Materiales_Complementarios/Material_de_caja_b_y_caja_n.pdf.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El proceso unificado de desarrollo de software.* 2000.

Leyva, Yanelis Rosales y Rodríguez Labrada, Leonardo. 2010. *Análisis, Diseño e Implementación del Mercado de Datos sobre las Inversiones para la Oficina Nacional de Estadísticas.* Ciudad Habana : s.n., 2010.

Martín, Liber Matos. Se soluciona programando. [En línea] [Citado el: 12 de 04 de 2011.] <http://jitcode.blogspot.com/2010/06/poo-y-patrones-de-diseno-en-drupal.html>.

Muñoz, Adniel Quintana. 2009. *Implementación del Módulo de Búsqueda del Sistema de Monitoreo y Análisis de Noticias.* Ciudad Habana : s.n., 2009.

Pérez, M. T. (2010). Biblioteca. Citado el 14 de marzo de 2011, de <http://biblioteca.uci.cu/>

Pintado, Eliana Acosta. [En línea] [Citado el: 12 de Enero de 2011.] <http://www.scribd.com/doc/2413026/Lenguaje-de-programacion-para-Web-ASP>.

Salazar, M. R. (2009). Recuperado el 13 de junio de 2011, de <http://dspace.espech.edu.ec/bitstream/123456789/35/1/18T00396.pdf>

Sanchez, María A. Mendoza. 2004. *Metodologías De Desarrollo De Software.* Perú : s.n., 2004.

scribd.com. Que es un CMS. [En línea] [Citado el: 24 de Noviembre de 2010.] <http://www.scribd.com/doc/9784583/Que-es-un-CMS>.

UCI. 2010-2011. *Asigantura Ingeniería de Software I.* 2010-2011.

UCIISW. eva. [En línea] [Citado el: 2 de abril de 2011.]

VARGAS, P. A. (11 de octubre de 2005). Recuperado el 17 de junio de 2011, de <http://repositorio.espe.edu.ec/bitstream/21000/678/1/T-ESPE-014103.pdf>

Graduados.uci.cu. (s.f.). Recuperado el 18 de junio de 2011, de <http://graduados.uci.cu/?q=node/2466>



BIBLIOGRAFÍA.

Anadic Sinaloa. [En línea]

http://www.anadicsinaloa.com/index.php?option=com_content&view=article&id=207.

A., Ernesto Quiñones. [En línea] http://postgresql.org.pe/articles/introduccion_a_postgresql.pdf.

Alvarez, Sara. Desarrollo web.com. [En línea] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.

Cantillo's, Carmen. Carmen Cantillo's Blog. [En línea] <http://carmencantillo.wordpress.com/about/>.

CIPRE. CIPRE. [En línea] <http://quienesquien.cip.cu/faq>.

CIPREa. CIPRE. [En línea] <http://www.cip.cu/contenido/menu2/servic>.

GoldScripts. [En línea] <http://www.goldscripsts.com/2007/11/14/wordpress-caracteristicas-generales/>.

Jomlaos.net. [En línea] <http://www.jomlaos.net/caracteristicas-de-joomla>.

José H. Canós, Patricio Letelier y M^a Carmen Penadés. [En línea]

<http://www.willydev.net/descargas/prev/TodoAgil.pdf>.

Masa, Pablo Villaverde. [En línea] http://tgp0607.awardspace.com/Recursos_ASP.pdf.

Manuales de ayuda. [En línea] <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/>.

Sánchez, Dr. José Ignacio Peláez. [En línea]

http://www.lcc.uma.es/~jignacio/index_archivos/TEMA4.pdf.

Uaem. [En línea] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>.

WordPress.org. [En línea] http://codex.wordpress.org/es:WordPress_Features.

web, Desarrollo. [En línea] <http://www.desarrolloweb.com/articulos/182.php>.

WebEstilo. [En línea] <http://www.webestilo.com/php/php00.phtml>.