

Universidad de las Ciencias Informáticas
FACULTAD 1



Título: Generador de portales web

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

Autores: Marlon Galens Molina

Iván Camejo Monroy

Tutores: Ing. Daneysi Granado Rodríguez

Ing. Yurelkys de los Ángeles Carreras Riopedre

Ing. Nodelvis Hernández Rodríguez

26 de junio de 2015

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor

Firma del Tutor

Agradecimientos

A la revolución cubana por darnos la oportunidad de nuestra preparación profesional, pues como dijera Jose Martí ser cultos es la única manera de ser libres.

Al Lic. Juan Carlos Rodríguez, editor e investigador de innumerables textos entre ellos la obra "Vilma, una vida extraordinaria" de la editorial Capitán San Luis.

A nuestra familia en especial nuestros padres por su apoyo incondicional durante toda nuestra carrera que ha sido el impulso que necesitábamos para estar hoy donde estamos.

A nuestros amigos tantos los que crecieron con nosotros como los que estuvieron sentados en nuestra aula, cada uno aportó su granito de arena para ser posible este sueño.

Resumen

El departamento de Servicios Informáticos para Internet (SENIT) del Centro de Ideoinformática de la Facultad 1, se encarga del desarrollo de portales web a la medida; en la actualidad el mismo cuenta con recursos humanos limitados y pocas herramientas de apoyo para cubrir la alta demanda de estos. En la presente investigación se propone la concepción y desarrollo de un generador de portales web que permita agilizar el proceso de desarrollo de portales en el Departamento de Servicios Informáticos para Internet, para esto se realiza un estudio de las principales tendencias actuales en cuanto a la generación de portales con el objetivo de identificar elementos significativos que pudieran guiar su construcción. El desarrollo de la herramienta se rigió por las especificaciones propuestas en la metodología de desarrollo de software OpenUp, que facilitó el análisis, el diseño, la implementación y la validación de las funcionalidades del sistema. Se usará como Sistema de Gestión de Contenidos Drupal 7 y como lenguaje de programación PHP 5.5. Para la validación de la propuesta de solución se utilizarán las herramientas Apache JMeter2.3.1 y websecurify scanner 0.9.

Palabras clave: desarrollo, Drupal, gestión de contenidos, Generador web.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción.....	4
1.2 Análisis crítico de los sistemas estudiados	4
1.3 Sistemas homólogos estudiados.....	5
1.4 Lenguajes y tecnologías.....	6
1.5 Herramientas	9
1.6 Metodologías de desarrollo.....	10
1.7 Conclusiones parciales	12
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA PROPUESTA DE SOLUCIÓN	13
2.1 Introducción	13
2.2 Propuesta de solución.....	13
2.3 Modelo de dominio.....	13
2.4 Requisitos Funcionales	15
2.5 Requisitos no funcionales.....	16
2.6 Diagrama de casos de uso del sistema y especificación de casos de uso.....	17
2.7 Descripción de estilos arquitectónicos y patrones de diseño.....	21
2.8 Modelo de despliegue	23
2.9 Diagrama de clases del diseño web.....	24
2.10 Diagrama de secuencia.....	27
2.11 Conclusiones parciales	28
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA	29
Introducción	29
3.1 Diagrama de componente.....	29
3.2 Estándares de codificación	30
3.3 Pruebas.....	35
3.5 Resultados de las pruebas	39
3.6 Conclusiones.....	41
REFERENCIAS BIBLIOGRÁFICAS:	44

INTRODUCCIÓN

Internet se ha convertido en los últimos años en el principal espacio para el comercio y la gestión del conocimiento, dos áreas fundamentales para cualquier empresa o institución constituyendo una necesidad para las mismas el poseer un sitio o portal web. Con un portal web se dispone de la mejor carta de presentación ya que proporciona al instante información visible tanto a los clientes locales como a los posibles futuros clientes por internet, facilitando en todo momento la obtención de toda la información de la empresa o institución y el acceso a los productos o servicios que las mismas brindan.

En Cuba, a causa del bloqueo impuesto por el gobierno de Estados Unidos, los niveles de conectividad son bajos, la mayoría de las instituciones y empresas cubanas aún no cuentan con un portal que las represente, o el que poseen no se ajusta a las nuevas tendencias web, limitando las posibilidades de negocios e intercambio de información en esta época de interconexión, por lo que poseer un sitio web de calidad se ha convertido en su prioridad.

La UCI tiene como misiones principales la informatización de la sociedad y lograr una mayor visibilidad del país en Internet, de ahí que el desarrollo de los portales constituya una de sus principales líneas de producción. Esta línea de producción es desarrollada, fundamentalmente, por el departamento SENIT del Centro de Ideoinformática de la Facultad 1, el cual cuenta con recursos humanos limitados y pocas herramientas de apoyo para cubrir la alta demanda de proyectos en un plazo de tiempo corto, por lo que los clientes pueden estar un tiempo prolongado en una lista de espera.

Una de las etapas que puede implicar un período de realización largo es la del levantamiento de información si el cliente no tiene una idea clara de lo que desea y al especialista se le hace difícil identificar los requerimientos del sistema.

El levantamiento de información se realiza a través de entrevistas con el cliente, donde se aplican encuestas que le permiten a los especialistas identificar los requerimientos del sistema y las necesidades del cliente y de la audiencia; sin embargo, a través de estas encuestas se hace difícil identificar detalladamente la posible representación de los contenidos y su ubicación en la navegación del portal, por lo que es necesario realizar el proceso de diseño de la Arquitectura de Información(AI).

Este proceso lleva varias iteraciones donde se presentan al cliente los prototipos con las pantallas de primer nivel de navegación y luego del resto de los niveles y se van haciendo ajustes de acuerdo a las consideraciones del mismo hasta lograr su aprobación. Existen

proyectos que no presentan una complejidad alta en cuanto a funcionalidades requeridas y diseño, no obstante deben pasar por estas etapas del modelo de desarrollo vigente en el departamento.

La siguiente etapa es el desarrollo del portal, que puede demandar dos meses de esfuerzo profesional. Para el desarrollo web se utiliza el CMS Drupal el cual cuenta con su propia comunidad de usuarios y desarrolladores que contribuyen en la mejora del software o aportando nuevos módulos para incrementar o mejorar sus funcionalidades. La reutilización de estas funcionalidades o módulos disminuye el tiempo de desarrollo, pero su configuración y adaptabilidad al diseño web para ser lograda con éxito debe ser aplicada por personal experimentado en el trabajo con el CMS. Otras de las ventajas de Drupal es el uso de distribución, también conocida como perfil de instalación cuales cuentan con un conjunto de instrucciones que permite definir qué módulos y opciones se desea activar durante el proceso de instalación de Drupal. De esta forma se podrá crear portales Web sin necesidad de empezar desde cero, ganando en tiempo y productividad.

Se ha implementado un perfil de instalación para Drupal el cual define para todos los portales una serie de funcionalidades utilizadas en proyectos anteriores por el departamento, pero presenta la dificultad de que algunos proyectos no necesitan de ciertos módulos definidos en el perfil lo que trae consigo que después haya que desinstalarlos.

Al tomar en consideración lo anteriormente planteado se define como **problema científico**: ¿Cómo agilizar el proceso de desarrollo de portales web en el Departamento de Servicios Informáticos para Internet?

Para desarrollar esta investigación y teniendo en cuenta el problema definido anteriormente, **se determina como objeto de estudio** las herramientas de generación de páginas web. **El campo de acción** se centra en las herramientas online de generación de páginas web.

Para brindar una solución al problema planteado se propone como **objetivo general** desarrollar un sistema para la personalización de portales web que permita agilizar el proceso de desarrollo en el Departamento de Servicios Informáticos para Internet, desglosándolo en los siguientes **objetivos específicos**:

1. Construir el marco teórico conceptual de la investigación que permita la comprensión del escenario.
2. Definir perfiles de portales web.
3. Diseñar las interfaces para definir la arquitectura de información.

4. Desarrollar el portal base para implementar los procesos y generar el portal web.
5. Validar los resultados del sistema

Para lograr el objetivo general se plantean las siguientes **Tareas de la Investigación:**

1. Descripción de las características de un generador de portales web.
2. Definición del flujo de actividades para diseñar la arquitectura de información (AI).
3. Diseño de las interfaces para la definición de la arquitectura de información.
4. Definición de los procesos para la generación de perfiles de portales web.
5. Implementación de los mecanismos de levantamiento de información y diseño de la AI.
6. Implementación del generador de portales web.
7. Realización de pruebas al sistema.
8. Valoración del impacto de la solución implementada.

Capítulo I: Fundamentación teórica del desarrollo del generador Web: contiene los aspectos esenciales para entender el entorno del problema a resolver. Incluye los conceptos fundamentales y las tendencias actuales, así como una descripción de las herramientas y tecnologías utilizadas para el desarrollo de la solución.

Capítulo II: Características de la propuesta de solución: se determinan las características del sistema, se definen los requisitos funcionales y no funcionales con los que debe cumplir, se describen a profundidad las características de la solución propuesta mediante los diversos artefactos que especifica el proceso de software utilizado y se define quién interactúa con el sistema.

Capítulo III prueba de la propuesta de solución: en este capítulo se detallan las pruebas realizadas a la propuesta de solución del problema planteado. Además se mostrarán los resultados de cada una de las pruebas y la solución que se le dio a cada problema.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se realiza un estudio de los sistemas existentes que realizan funciones similares, descripción de las diferentes herramientas y lenguajes que se van a usar para la solución.

1.2 Análisis crítico de los sistemas estudiados

1.2.1 ¿Qué es un generador de portales web?

Un generador de portales web es una versión simplificada de los llamados sistemas de gestión de contenido (CMS, del inglés, *content management system*). (García, 2014)

Mientras que los sistemas clásicos de gestión de contenido se usan para el mantenimiento y gestión de grandes páginas web, los generadores de páginas web están pensados para proyectos web más pequeños, en los cuales el diseñador del proyecto no tiene que tener conocimientos de programación. Igualmente, tampoco son necesarios conocimientos técnicos como la instalación básica de un CMS, la creación de plantillas o la configuración de bases de datos. (García, 2014)

El usuario de un generador de páginas web puede montar un sitio web totalmente funcional y así tener su presencia en la red. El usuario puede introducir e implementar con facilidad actualizaciones de texto, nuevos artículos o distintos contenidos multimedia (como vídeo y audio) fácilmente mediante los distintos módulos directamente en la página. (García, 2014)

1.2.2 ¿A qué tipo de usuario están dirigidos los generadores de páginas web?

Los principales grupos de usuarios de estas herramientas son los, autónomos, fotógrafos, las pequeñas y medianas empresas y emprendedores con o sin una tienda en línea. (García, 2014)

1.2.3 Tipos de generadores web

Existen generadores de páginas web que están alojados en el sistema del proveedor que los ofrece y, por otra parte, existen también generadores de páginas web que deben instalarse como software en un ordenador. Ambas variantes presentan ventajas e inconvenientes: mientras que la opción de instalación suele tener un funcionamiento más rápido; la opción de *hosting online* (alojamiento en línea) implica que puede acceder en cualquier momento desde cualquier lugar. Con el rápido incremento y mejora del acceso a Internet, las herramientas en línea están ganando popularidad, esta es la razón por la que el estudio de los sistemas homólogos se centra en las soluciones en línea. (García, 2014)

1.3 Sistemas homólogos estudiados.

Para realizar el estudio de los sistemas homólogos se visitaron varios sitios teniendo en cuenta su popularidad. Entre los sitios visitados están:

1. <http://www.multisitios.net>

Permite armar una web de acuerdo las necesidades más comunes habilitando diferentes secciones, entre las que se encuentran diferentes tipos de galerías de imágenes, páginas de información, libros de visitas, formularios de contactos, además permite vincular el sitio creado a Facebook y configurar las páginas para una correcta indexación por parte de los buscadores

2. <http://es.wix.com>

Wix ofrece impresionantes diseños, posibilidad de añadir animaciones y muchas aplicaciones en su mercado de aplicaciones, por otra parte una vez se selecciona una plantilla, ésta no puede ser cambiada con facilidad.

3. <https://www.yola.com/es>

Yola es un servicio de construcción de sitios y alojamiento que ofrece todo lo que una pequeña empresa u organización necesita para sobresalir en la web. Desde nombres de dominio personalizados, diseño de plantillas, hasta alojamiento confiable y libre de publicidad. Yola es una de las mejores formas de crear y lanzar rápidamente un sitio web. Ofrece planes gratuitos, diseños clásicos y es fácil de utilizar.

4. <http://www.drupalgardens.com>

Drupal Gardens ofrece la posibilidad de generar instancias de Drupal 7, con un reducido pero selecto conjunto de módulos que permiten generar con un esfuerzo moderado una buena variedad de tipos de sitios web: desde páginas estáticas, a blogs para uno o varios usuarios, o incluso pequeñas comunidades digitales con foros, lista de correo básica, encuestas y contenidos personalizables.

5. <http://webnode.es>

Webnode es muy recomendado para páginas web multilingües además ofrece una gran variedad de funcionalidades profesionales y una buena optimización de motores de búsqueda.

1.3.1 Principales inconvenientes

En el estudio realizado sobre sitios generadores de portales web se destacan como principales problemas que los más usados son privativos o no están disponibles para su uso en Cuba. Otro de los inconvenientes es que otorgan poca seguridad pues no se tiene acceso al código fuente, ya que el sitio es hospedado dentro de la misma herramienta por lo que también los datos e información se alojan fuera del control del usuario, además el sitio no puede ser exportado. En otros casos se ajustan solo a las necesidades y características de las instituciones que los desarrollaron o no están orientados a todo tipo de usuarios.

1.3.2 Aspectos positivos

Entre los principales aspectos que aporta el estudio anteriormente mencionado y que constituyen puntos de partida para la propuesta de solución se pueden mencionar cuatro ideas fundamentales:

1. Se hacen preguntas sencillas relacionadas a los objetivos del sitio a confeccionar para así mostrar plantillas más acorde con la clasificación donde entrará este.
2. Se le permite al usuario añadir páginas opcionales a su web independientemente de las páginas que traen por defecto la configuración del prototipo seleccionado, como una galería y foros.
3. Se ofrece la oportunidad de configurar una vez terminado el diseño algún aspecto específico, como colores, tipografía e imágenes, entre otros.
4. Se ofrece ayuda técnica: dependiendo del proveedor ésta será ofrecida mediante e-mail, chat en vivo, foros o incluso por teléfono.

1.4 Lenguajes y tecnologías

1.4.1 HTML 5

HTML siglas de *HyperText Markup Language* («lenguaje de marcado de hipertexto»), desarrollado por el *World Wide Web Consortium (W3C)*, es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y

el contenido de las páginas en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta cierto punto, la apariencia de un documento, y puede incluir scripts (por ejemplo, escritos en JavaScript), los cuales pueden afectar el comportamiento de navegadores web y otros procesadores de HTML. (GALEANO, Germán, SÁNCHEZ, DÍAZ .2009)

1.4.2 Lenguaje de programación PHP5.5

PHP (acrónimo de "PHP: *Hypertext Preprocessor*") es un lenguaje interpretado ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente, se ejecuta en un servidor web tomando el código en PHP como su entrada y creando páginas web como salida. A esto se añaden valores como el hecho de ser un proyecto de código abierto, gratuito y multiplataforma. PHP es libre por lo que se presenta como una alternativa de fácil acceso para todos y permite técnicas de Programación Orientada a Objetos. Además, posee una extensa biblioteca nativa de funciones, así como una amplia documentación oficial. (John, COGGESHALL. 2005)

1.4.3 JavaScript 1.8

Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la de los lenguajes Java y C. Es un lenguaje orientado a objetos puesto que dispone de herencia, la cual se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. JavaScript se ejecuta en el cliente al mismo tiempo que las sentencias van descargándose junto con el código HTML. (SÁNCHEZ MAZA, Miguel Ángel. 2006)

1.4.4 CSS 3

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores web. La idea que se encuentra detrás del desarrollo de

CSS es separar la estructura de un documento de su presentación. (VAN LANCKER, Luc. 2007)

Entre las ventajas de usar CSS o (u otro lenguaje de estilo) se pueden citar

Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

1.4.5 SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella.

Características generales del SQL

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones en estos últimos. Es un lenguaje declarativo que especifica qué es lo que se quiere y no cómo conseguirlo. Gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, permite una alta productividad en codificación y la orientación a objetos. (CHARTE OJEDA, Francisco.2009)

1.4.6 Servidor web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse

algún protocolo. Generalmente, para estas comunicaciones se utiliza el protocolo HTTP perteneciente a la capa de aplicación del modelo de Interconexión de Sistemas Abiertos (OSI). (Universidad Santos Tomas.2013)

1.4.7 Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado o UML (siglas de *Unified Modeling Language*) es un lenguaje para especificar, construir, visualizar y documentar los artefactos (información que se utiliza o produce mediante un proceso de software). Este lenguaje de modelado no es una guía para realizar el análisis y diseño orientado a objeto, es decir, no es un proceso, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos informáticos, de arquitectura o de cualquier otra rama. (RMAN, Craig.1999)

1.5 Herramientas

1.5.1 Entorno de Desarrollo Integrado

NetBeans 7.3.1

Licencia del producto: Doble licencia; *Common Development and Distribution License* (CDDL) y *GNU General Public License version 2 with Classpath exception* (GPL2). NetBeans es un proyecto exitoso de código abierto con una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto NetBeans en junio 2000 y continúa siendo su patrocinador principal. Hoy en día hay disponibles dos productos: el Entorno de Desarrollo Integrado NetBeans y la Plataforma NetBeans. NetBeans es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además, un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso. (Oracle Corporation.2011)

1.5.2 Gestor de bases de datos

PostgreSQL

PostgreSQL es un sistema gestor de base de datos relacional libre y orientado a objetos, publicado bajo la licencia BSD. Se destaca por ejecutar consultas complejas y sobre vistas, subconsultas y joins de gran tamaño. Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Posee una amplia variedad de tipos nativos (números de precisión arbitraria, texto de largo ilimitado, figuras geométricas). Tiene gran soporte para vistas, procedimientos ubicados en el servidor, transacciones, almacenamiento de objetos de tamaño considerable y características orientadas a objetos (Pecos, 2008).

Se estará haciendo uso de PostgreSQL en su versión 9.2.

1.5.3 Ingeniería del Software asistida por computadoras

Nombre de la herramienta: Visual Paradigm8.0 o superior.

Visual Paradigm es una herramienta de modelado profesional que hace uso del Lenguaje Unificado de Modelado (de sus siglas en inglés UML).

Una característica esencial de Visual Paradigm es que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción y despliegue. Visual Paradigm ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (Visual Paradigm for UML.2011)

1.6 Metodologías de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Pueden ser comparadas con un plan de contingencias en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quiénes deben participar en el desarrollo de las actividades y qué papel deben tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Durante la selección de la metodología a utilizar para la documentación del portal web solo se tuvieron en cuenta las metodologías ágiles debido a que el equipo de desarrollo es pequeño y dispone de un período de tiempo corto.

1.6.1 Open Unified Process (OpenUP)

Es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. OpenUp es ágil, ya que permite realizar la documentación del proyecto con rapidez. (Rodríguez, Ana P., Polanco, Josué y Hernández, Darwin. 2013)

Características de OpenUp

1. Preserva la esencia del *Rational Unified Process* (RUP):

- Desarrollo iterativo e incremental.
- Desarrollo dirigido por Casos de Uso.

2. Centrado en la Arquitectura.

3. La mayoría de las características opcionales de RUP se excluyeron y otras se unificaron para simplificar el proceso: solo lo fundamental está incluido, sin dejar de ser completo y extensible

4. Está pensado para proyectos de 3 a 6 personas, 3 a 6 meses de esfuerzo de programación.

5. Es un proceso iterativo para el desarrollo de software.

- Mínimo: solo incluye el contenido del proceso fundamental.
- Completo: puede ser manifestado como proceso entero para construir un sistema.
- Extensible: puede ser utilizado como base para agregar o para adaptar más procesos.

(Rodríguez, Ana P., Polanco, Josué y Hernández, Darwin. 2013)

Ventajas de OpenUp

1. Es Apropiado para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.

2. Permite detectar errores tempranos a través de un ciclo iterativo.
3. Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
4. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas

Se decide utilizar OpenUp debido a que es una metodología de desarrollo de software de código abierto diseñado para pequeños equipos organizados quienes quieren tomar una aproximación ágil del desarrollo.

Es un proceso iterativo e incremental que es mínimo, completo y extensible. Los Practicantes de desarrollo de software (desarrolladores, administradores de proyectos, analistas y probadores) trabajan juntos como un equipo de proyecto. No define un modelo de negocio ni de dominio necesario. Permite detectar errores tempranos a través de un ciclo iterativo. Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

(Rodríguez, Ana P., Polanco, Josué y Hernández, Darwin. 2013)

1.7 Conclusiones parciales

Después de realizado un estudio de los principales conceptos que guiaron esta investigación se arribaron a las siguientes conclusiones:

El estudio realizado de los sistemas homólogos permitió identificar la elección de los principales contenidos que se desean promover en el portal y las principales funcionalidades que debería presentar la propuesta de la solución.

De igual forma se analizaron las herramientas y tecnologías a utilizar, profundizando así los conocimientos necesarios para el desarrollo de la propuesta de solución al problema planteado.

Capítulo 2: Descripción y análisis de la propuesta de solución

2.1 Introducción

En este capítulo se presenta la propuesta de solución al problema planteado en el diseño teórico de la investigación. Para ello se desarrolla el modelado del dominio y se describen los conceptos fundamentales asociados a este. Se especifican además los usuarios que tendrán acceso a la misma, los requerimientos funcionales y no funcionales y se muestran las técnicas utilizadas para su obtención. Por último, se realiza la descripción de la arquitectura, presentando luego el modelo de datos y la distribución física o diagrama de despliegue de dicha propuesta.

2.2 Propuesta de solución

A partir del estudio exhaustivo a los generadores de páginas webs, se evidenció que los mismos poseen características comunes en cuanto a secciones y servicios. Esto hizo posible la elección de los principales contenidos que se desean promover en el portal.

En la sección FAQ: son preguntas frecuentes como traducción literal. Las preguntas frecuentes, son aquellas que se hace la mayoría de los usuarios, son las preguntas más comunes que pueden surgir, en este caso, cuando interactúa con el sistema.

En la sección Contáctenos: estarán las diferentes formas de contactar con los administradores, para cualquier inquietud o recomendación.

El enlace Diseñe su página web: le solicita al usuario información necesaria para generar, el portal, como la categoría a la que va pertenecer teniendo en cuenta su objetivo principal, el tema, diseño de la portada, contenido que se desea promover, y los datos básicos como el nombre del sitio, el correo y el usuario del administrador.

2.3 Modelo de dominio

El objetivo del modelado del dominio es capturar, comprender y describir las clases más importantes dentro del contexto del sistema. El Modelo del Dominio, ayuda a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. En la figura 1 se muestra el Diagrama del modelo del Dominio.

2.3.1 Diagrama de dominio.

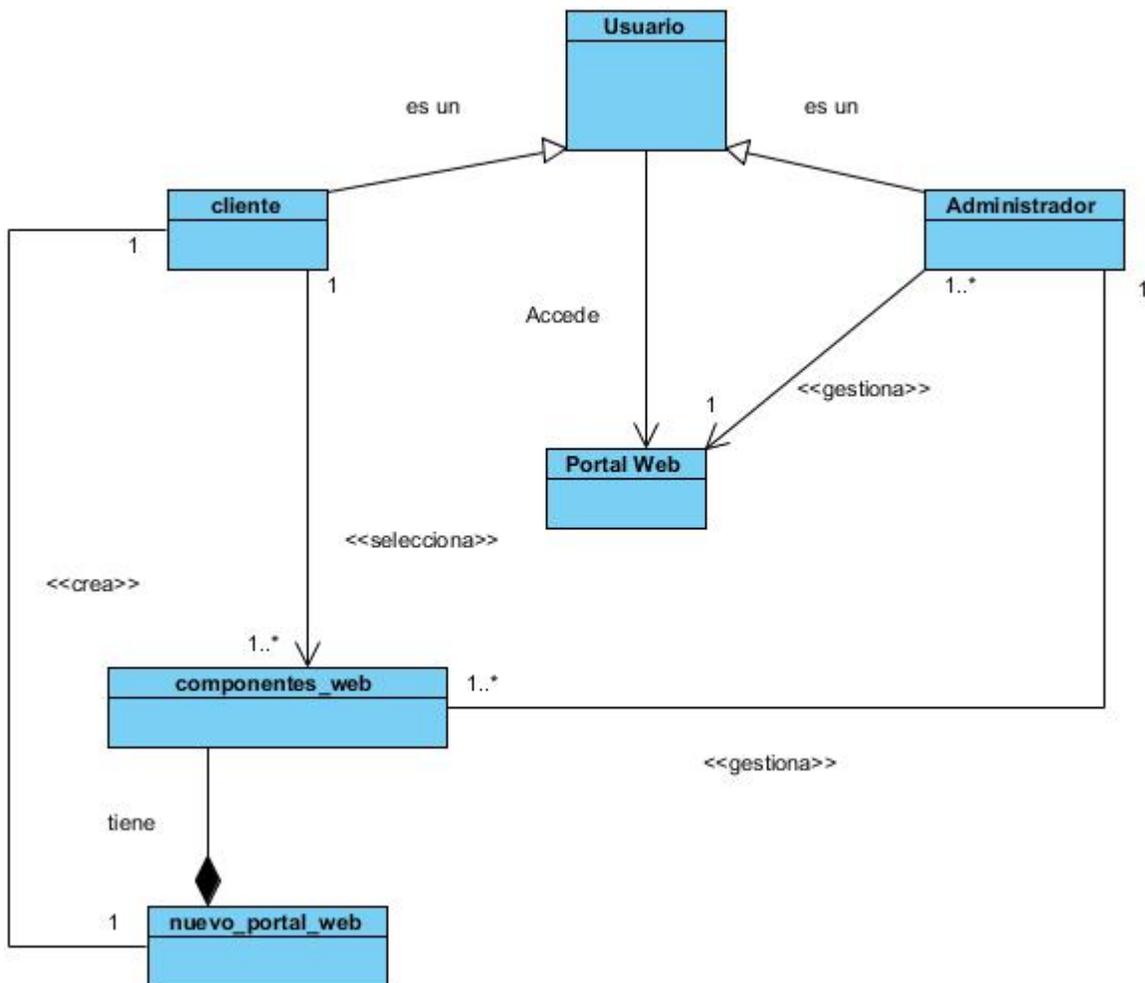


Ilustración 1 Modelo de dominio

Portal Web: Contiene los diseños de los diferentes componente web predefinidos.

Diseños componentes web: Son los diferentes componentes predefinidos los cuales serán escogidos por el usuario cliente para la confección de su portal web.

Nuevo portal web: Es el nuevo portal web que será exportado, el cual contiene los diseños de los componentes web seleccionado por el usuario cliente.

Usuario: representa a todas las personas autenticadas en el sistema.

Administrador: usuario con privilegios de administración.

2.3.2 Descripción del modelo de dominio.

En sentido general los usuarios acceden al portal web utilizando su nombre de usuario y contraseña, donde este usuario ya puede ser un administrador o un cliente. El administrador es el responsable de definir los permisos de acceso de los usuarios del sistema. Su principal tarea es mantener el portal en funcionamiento. Los clientes tienen la opción de seleccionar los distintos componentes web disponibles para la confección del portal web a exportar. Estos componentes son añadidos por el administrador. El producto final será el portal web compuesto por los componentes de portales web seleccionados por el usuario cliente.

2.4 Requisitos Funcionales

Los requisitos funcionales son condiciones o capacidades que el sistema debe cumplir. Además son el punto de partida para identificar qué debe hacer el sistema. Deben ser comprendidos por los desarrolladores, así como por los usuarios y clientes.

A continuación se muestran los requisitos funcionales de la propuesta de solución.

No.	Nombre
RF.1	Autenticar usuario
RF.2	Agregar Categoría
RF.3	Editar Categoría
RF.4	Eliminar Categoría
RF.5	Mostrar Categoría
RF.6	Agregar tipo de contenido
RF.7	Editar tipo de contenido
RF.8	Eliminar tipo de contenido
RF.9	Mostrar tipo de contenido
RF.10	Agregar portada
RF.11	Editar portada
RF.12	Eliminar portada
RF.13	Mostrar portada
RF.14	Agregar tema
RF.15	Editar Tema
RF.16	Eliminar tema

RF.17	Mostrar tema
RF.18	Agregar dependencia
RF.19	Editar dependencia
RF.20	Eliminar dependencia
RF.21	Mostrar dependencias
RF.22	Seleccionar Categoría
RF.23	Seleccionar Tema
RF.24	Seleccionar Portada
RF.25	Datos del sitio
RF.26	Seleccionar Tipos de contenidos
RF.27	Mostrar datos
RF.28	Descargar Portal

2.5 Requisitos no funcionales

Apariencia o Interfaz externa:

- **RNF 1:** El sistema estará optimizado para una resolución de 1024x768.
- **RNF 2:** La interfaz contará con menús desplegables para agilizar y acelerar su utilización

Usabilidad:

- **RNF 3:** Se proporcionará una interfaz de manejo cómoda, que posibilite una rápida adaptación, tanto para los usuarios con conocimientos avanzados en informática como los usuarios más inexpertos.

Eficiencia:

- **RNF 4:** El sistema debe ser capaz de responder con rapidez a las peticiones de los usuarios, demorando como promedio en una transición, de menos de 60 segundos aproximadamente.

Fiabilidad:

- **RNF 5:** Debe registrar cualquier fallo de su funcionamiento para poder ser solucionado por los administradores.

- **RNF 6:** Los errores no deben estar por encima de los 2 errores/MLC26. En caso de que ocurran fallas, los errores deben mostrarse sin detalles de información que puedan comprometer integridad de y seguridad del mismo.

Seguridad:

- **RNF 7:** El acceso será controlado con nombres de usuario y contraseñas.
- **RNF 8:** Los mensajes de error mostrados a los usuarios deben ser genéricos sin dar detalles de información, para no comprometer la seguridad e integridad de los datos.

Soporte:

- **RNF 9:** Debe dar la posibilidad de ser mejorado y de incorporarle nuevos servicios en caso de necesitarlos.

2.6 Diagrama de casos de uso del sistema y especificación de casos de uso.

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. (Rivera, 2014)

2.6.1 Diagrama de casos de uso del sistema.

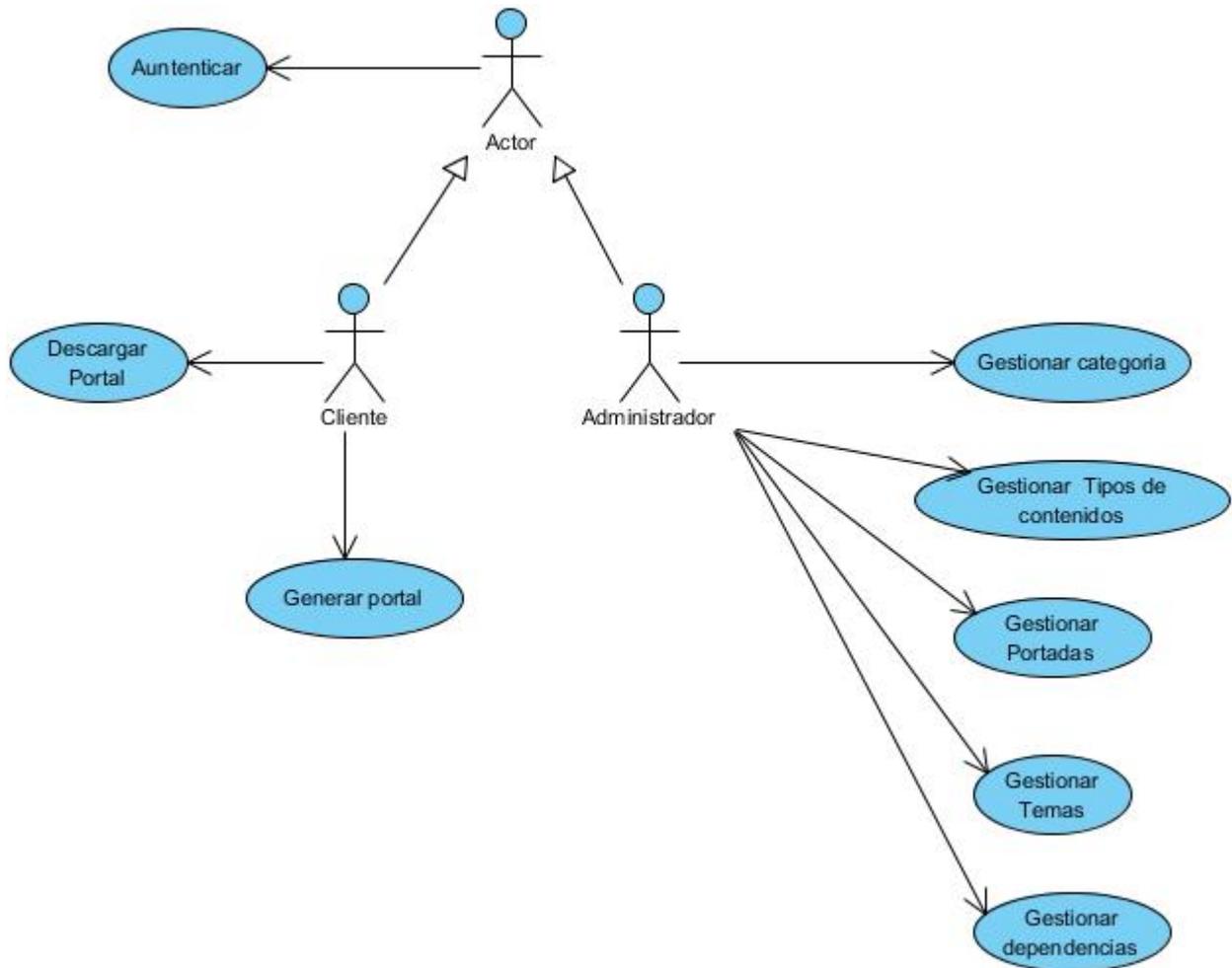


Ilustración 2 Diagrama de casos de uso del sistema.

2.6.2 Descripción de los casos de usos.

Tabla 1 CU#1 Gestionar categoría

Objetivo	Permite al usuario agregar, editar, eliminar el tipo de contenido categoría
Actores	Administrador

Resumen		
Complejidad	Baja	
Prioridad	Alta	
Precondiciones	<i>El usuario debe estar registrado.</i>	
Postcondiciones	Los datos introducidos o modificados por el usuario deben quedar registrados en el sistema.	
Flujo de eventos		
Flujo básico <Gestionar categoría>		
	Actor	Sistema
1.	El usuario selecciona la opción contenido que se encuentra en la barra de administración.	Se muestra un panel con las opciones de agregar, editar o eliminar.
2.	El usuario selecciona la opción deseada.	
Sección: "agregar categoría"		
1.	El usuario selecciona la opción agregar contenido que se encuentra en la barra de administración en el menú desplegable del enlace contenido.	Se muestra un panel con todos los tipos de contenidos existentes en el sistema.
2.	El usuario selecciona el enlace Categoría.	Se muestra primeramente dos campos de texto donde el usuario escribe el nombre del nuevo tipo de contenido y la descripción de este, a continuación se encuentra un campo para subir la imagen de la categoría".
3.	El usuario introduce los datos requeridos	
Flujo Alternativo		

Actor		Sistema
		En caso de que los datos introducidos por el usuario sean incorrectos, se muestra un mensaje de error con los datos incorrectos y se vuelve al paso # 3.
Sección: “editar categoría”		
1.	El usuario selecciona la opción de editar la categoría que desee editar.	Se muestra una nueva ventana con los datos a modificar
2.	El usuario modifica los datos.	
Flujo Alternativo		
Actor		Sistema
1		En caso de que los datos introducidos por el usuario sean incorrectos, se muestra un mensaje de error con los datos incorrectos y se vuelve al paso # 2.
Sección: “Eliminar categoría”		
1.	El usuario selecciona la opción de eliminar la categoría que desea eliminar.	La categoría es eliminada del sistema y muestra un mensaje que confirma que el tipo de contenido ha sido eliminado.
Relaciones	CU incluidos	
	CU extendidos	
Requisitos no funcionales		

2.7 Descripción de estilos arquitectónicos y patrones de diseño.

2.7.1 Estilo arquitectónico

Al ser utilizado el CMS Drupal para el desarrollo del portal, la arquitectura y los patrones son heredados del mismo. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos(*modules*), bloques y menús (*blocks & menus*), permisos de usuario y plantillas (*templates*). (cursosdrupal.com, 2015)

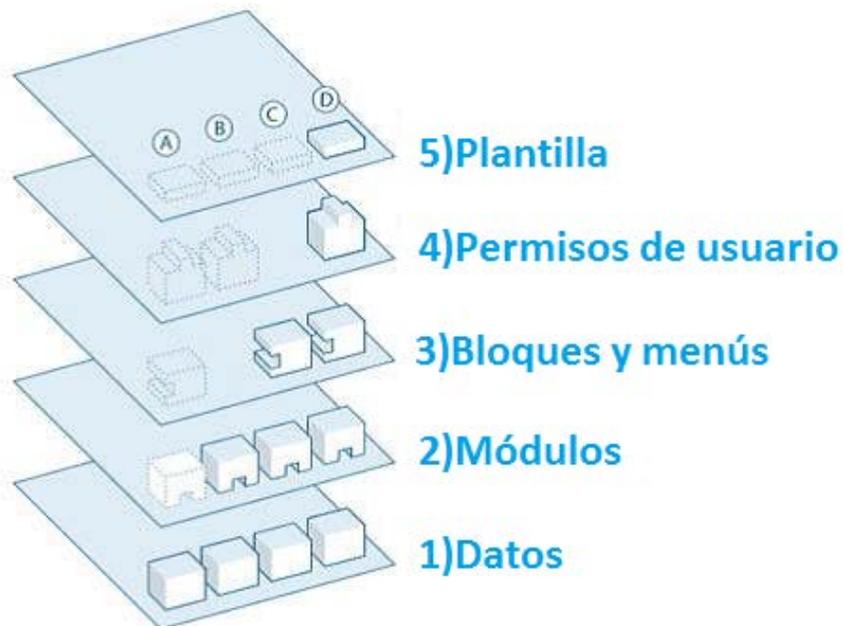


Ilustración 3 Estructura de Drupal

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. (cursosdrupal.com, 2015)

Módulos: Los Modules son los elementos que operan sobre los Nodes y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio web. (cursosdrupal.com, 2015)

Bloques y Menús: Estos permiten estructurar y organizar los contenidos en la página web. (cursosdrupal.com, 2015)

Usuarios y permisos: Actualmente, la mayor parte de sitios web son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permiten especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. (cursosdrupal.com, 2015)

Tema: 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. (cursosdrupal.com, 2015)

2.7.2 Patrones de diseños

Singleton: Este patrón está diseñado para que solo permita que se haga una instancia de un elemento cuando este no existe. Dentro del núcleo de Drupal se utiliza este patrón de diseño pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos elementos, pues Drupal solo crea una instancia de un nodo, cuando este nodo no existe. (drupal.org, 2013).

Observer: Este patrón es usado en Drupal para controlar la forma en que se realizan las interacciones entre nodos, pues cuando se hace una modificación en un nodo, Drupal notifica de este cambio a todos los nodos que se vean afectados por él. (drupal.org, 2013).

Command: Drupal utiliza este patrón para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro, junto con los argumentos y así agilizar la ejecución de ciertas tareas en el sistema. (drupal.org, 2013).

Chain of Responsibility: El sistema de menús de Drupal sigue este patrón. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud y si el usuario tiene acceso a los recursos solicitados. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa a otro. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado. (drupal.org, 2013).

Bridge: La capa de abstracción de bases de datos de Drupal se aplica de una forma similar a este patrón. Los módulos son escritos en una forma que es independiente del sistema que se está utilizando en la base de datos, y de esta forma, se puede añadir soporte para distintas bases de datos sin tener que modificar el código de un módulo. (drupal.org, 2013).

Decorator: Este patrón se usa en Drupal, por ejemplo, de la siguiente manera: un nodo necesita tener la opción de adjuntar ficheros y para esto, Drupal se vale del módulo *upload*, que brinda a los nodos la posibilidad de adjuntar, en vez de implementar esta funcionalidad en cada nodo. (drupal.org, 2013).

2.8 Modelo de despliegue

Este modelo da una medida de la tecnología necesaria para el correcto funcionamiento del módulo a desarrollar, propone la distribución física de los elementos que lo conforman ya que representa cómo estarán distribuidos y cómo se satisfacen los requerimientos no funcionales de *hardware* y *software*.



Ilustración 4 Diagrama de distribución física

2.8.1 Descripción de elementos e interfaces de comunicación

- **PC Cliente:** Desde aquí el usuario, ejecutando un navegador web podrá acceder al sistema alojado en el servidor de aplicaciones. La máquina cliente necesita disponer de muy pocas prestaciones puesto que solo necesita un navegador web para poder acceder al sistema y realizar las operaciones necesarias.
- **Servidor aplicaciones:** En este nodo es donde se ejecuta la aplicación. Gracias a la ejecución constante de un servicio de Apache o semejante recibe y procesa las peticiones del nodo cliente, contestando un HTML una vez completada la ejecución.

- **Servidor de Base de Datos:** Este nodo contiene un servidor de base de datos, el cuál almacena la información relacionada con el sistema.
- **<<HTTP>>:** El Protocolo Seguro de Transferencia de Hipertexto es un protocolo de red basado en HTTP por lo que está orientado a transacciones, sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor. La principal diferencia entre ellos es que este está destinado a la transferencia segura de datos de hipertexto, en otras palabras, es la versión segura de HTTP.
- **<<TCP/IP>>:** La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permite la transmisión de datos entre redes de computadoras. Denominada en muchas ocasiones conjunto de protocolos TCP/IP en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (*TCP*) y Protocolo de Internet (*IP*), que fueron los dos primeros en definirse, y que son los más utilizados de la familia. El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes Sistemas Operativos sobre redes de área local (LAN) y área extensa (WAN).

2.9 Diagrama de clases del diseño web

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases y las interfaces en una aplicación. Normalmente, contiene la siguiente información (Visconti, 2010): clases, asociaciones y atributos, interfaces, con sus operaciones y constantes, métodos, información sobre los tipos de los atributos, navegabilidad y dependencias.

Como se utiliza el CMS Drupal es necesario analizar su estructura y funcionamiento, para tener el conocimiento de cómo se confeccionan las clases del diseño web.

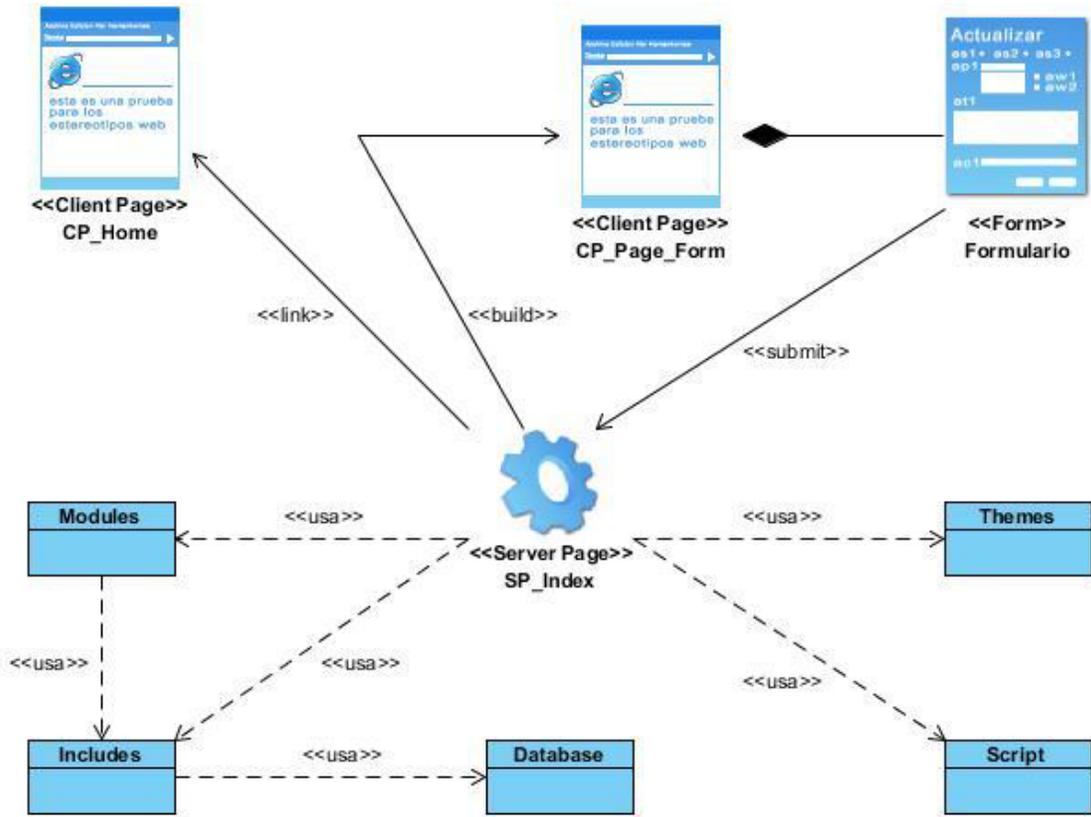


Ilustración 5 Diagrama de clases del diseño web de Drupal

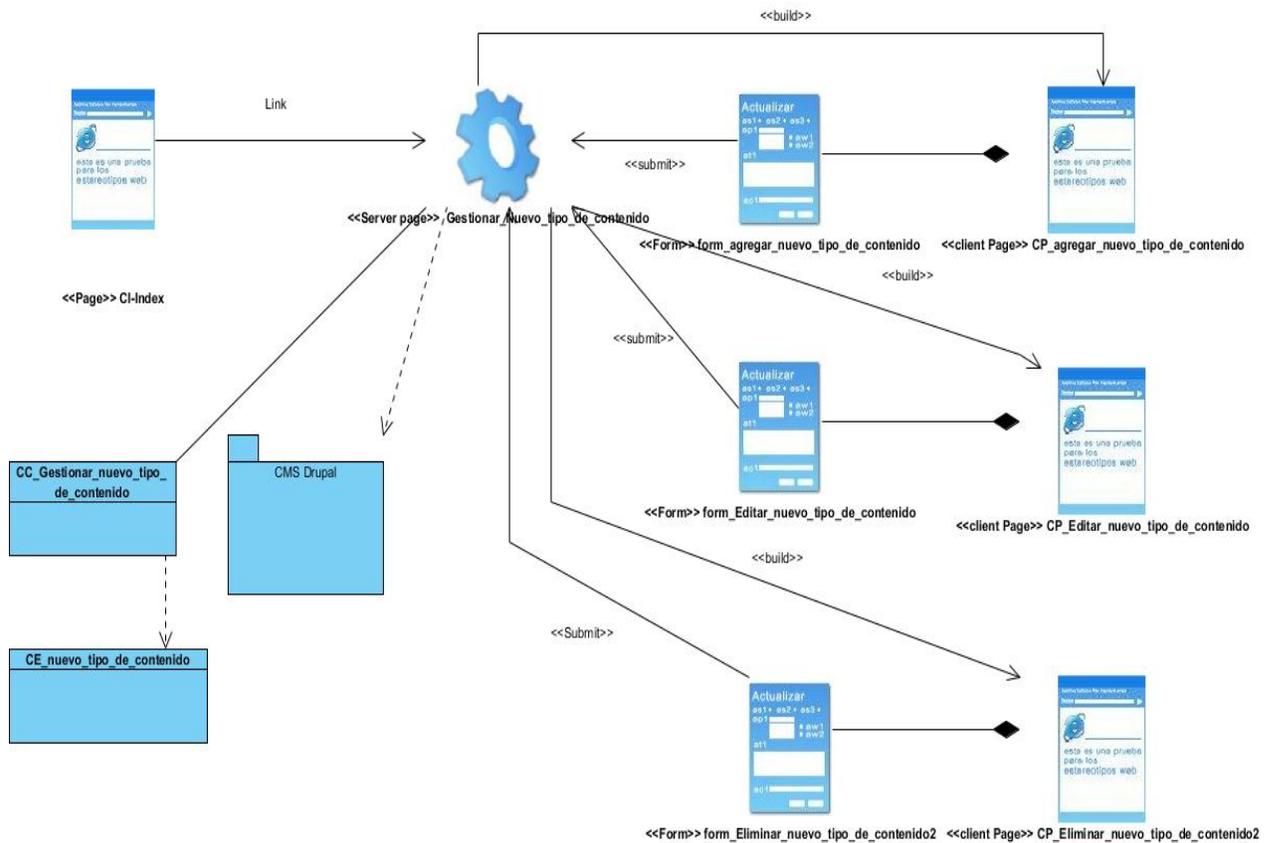


Ilustración 6 Diagrama de clase del diseño para el CU Gestionar tipo de contenido

2.9.1 Estructura y funcionamiento de Drupal

Para un mejor entendimiento, los usuarios finales que deben interactuar con la aplicación, se ve necesario que tengan un conocimiento sobre la estructura y funcionamiento de Drupal. La distribución que constituye básicamente el CMS Drupal contiene 4 paquetes principales: *Themes*, *Includes*, *Modules* y *Scripts*, que contiene a su vez gran cantidad de archivos. Para una mayor comprensión de los diagramas de diseño web se modela primeramente el diagrama general de Drupal:

Includes: contiene un conjunto de librerías en forma de archivos PHP con extensión .inc, que incluyen funciones comunes del sistema indispensables para su funcionamiento, como son las conexiones a la base de datos.

Modules: contiene todos los módulos del núcleo que permiten las distintas funcionalidades del CMS, cada uno en su carpeta correspondiente. Nunca se deben modificar directamente los módulos, ni subir módulos adicionales a esta carpeta.

Los módulos adicionales se deben subir a ***/sites/all/modules*** o ***/sites/default/modules***.

Scripts: contiene utilidades adicionales que no utiliza Drupal directamente, pero que podemos utilizar desde la línea de comandos de *shell*. Por ejemplo, el script `password-hash.sh` permite obtener una contraseña codificada a partir de la contraseña original (en texto plano).

Themes: contiene los temas que vienen con la distribución de Drupal. Los temas nuevos que quieran añadir, crear o modificar, se deben subir a ***/sites/all/themes*** o ***/sites/default/themes***.

Index: El archivo ***index.php*** es la puerta de entrada al sistema. Cuando carga una página del sitio web se está siempre haciendo una llamada a ***index.php***.

El procesamiento de la página comienza en el script ***index.php***, que llama a la función ***drupal_bootstrap()*** y pone en marcha distintas fases del proceso de arranque del sistema (bootstrap) (Gil, 2012).

2.10 Diagrama de secuencia

El diagrama de secuencia de un sistema es una representación que muestra los eventos generados por actores externos, su orden y los eventos internos del sistema (sophia.javeriana.edu.co, 2012)

El siguiente Diagrama de Secuencia corresponde al requisito de Agregar Nuevo tipo de contenido.

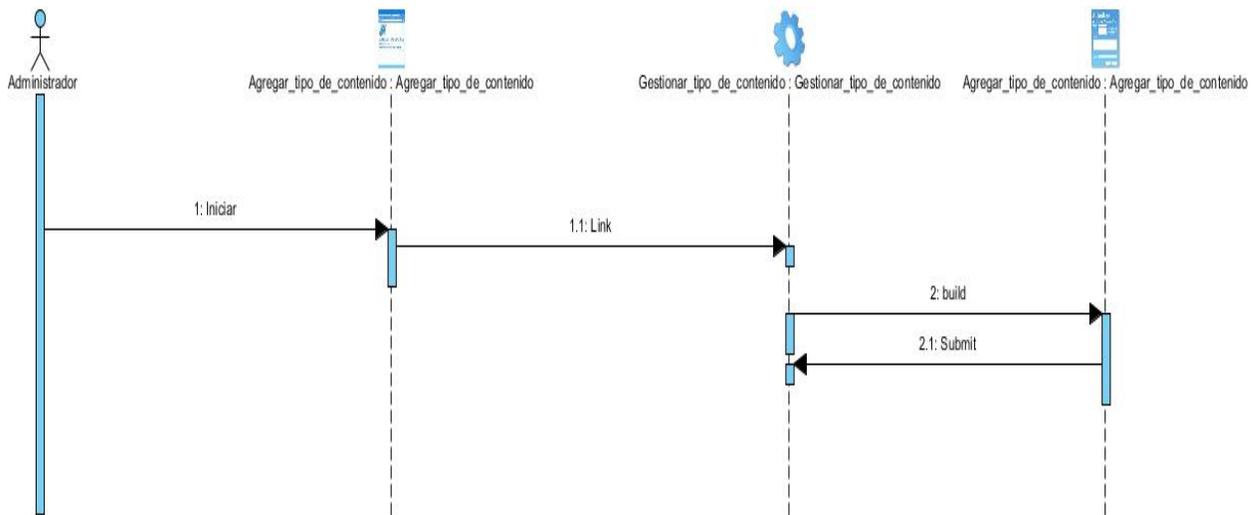


Ilustración 7 Diagrama de secuencia Agregar Nuevo tipo de contenido.

2.11 Conclusiones parciales

En el presente capítulo han sido descritas las características del portal. Se describieron los elementos significativos de la arquitectura de software y de información, así como los diferentes requisitos. Estas acciones arrojan las siguientes conclusiones:

Con la caracterización realizada al sistema, se identificaron los requisitos funcionales y no funcionales de la aplicación, los que aclararon cuáles serán las funcionalidades que este tendrá.

Se determinó la forma en que serán agrupados los contenidos a mostrar en la aplicación, y cómo se organizarán. De esta manera, se obtuvo como resultado una vista general del sistema para definir la estructura de los elementos del portal web, y los parámetros a tener en cuenta para el análisis con vistas a su implementación.

El análisis permitió refinar la estructura del sistema y profundizar en el análisis de este, mediante los diagramas realizados. El estilo arquitectónico de Drupal constituye un elemento importante, pues la calidad de la arquitectura define en qué medida se cumplirá o no con los indicadores de calidad del software y juega un papel fundamental para guiar el desarrollo.

Capítulo 3: Implementación y Prueba del Sistema

Introducción

En el presente capítulo se abordarán todos los temas relacionados con las etapas de implementación y prueba, haciendo uso del diagrama de componentes. Además se mostrará una descripción de las pruebas seleccionadas para validar la implementación realizada.

3.1 Diagrama de componente

Es utilizado para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

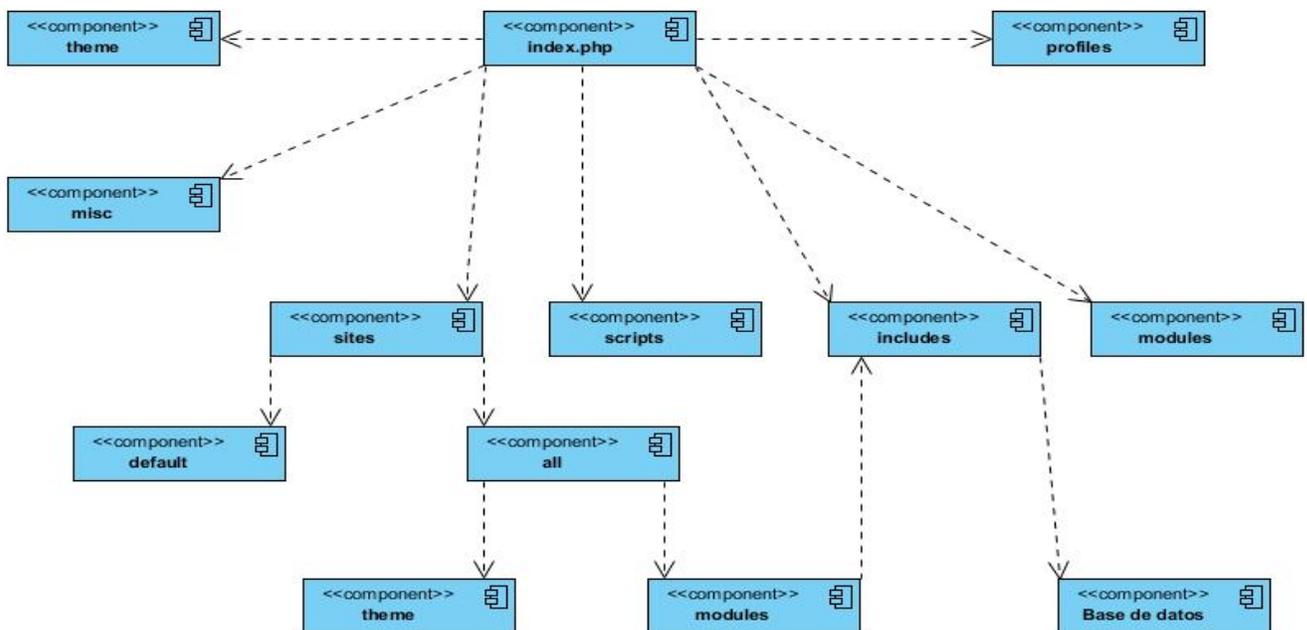


Ilustración 8 Diagrama de componentes

index.php: Este componente es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS

Drupal.themes: En este componente se incluyen los temas que vienen con la distribución de Drupal.

misc: Este componente incluye archivos *javascript* e imágenes requeridas por el sistema.

includes: Este componente tiene un conjunto de ficheros indispensables para el funcionamiento del CMS Drupal.

scripts: Contiene utilidades adicionales que no utiliza Drupal directamente, pero que se pueden utilizar desde la línea de comandos de *shell*.

modules: En este componente se encuentran los módulos básicos del *core* de Drupal.

profiles: Este componente contiene los perfiles de instalación de Drupal.

sites: La carpeta *sites* contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal, colocados en *sites/all/modules*. Además los temas adicionales instalados o creados, en *sites/all/themes*. La carpeta incluye tras la instalación el archivo de configuración del sitio (*settings.php*).

3.2 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica de programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente alta calidad y no contenga errores. Drupal le brinda a sus desarrolladores una serie de normas o pautas para fomentar el código de una forma común para todos. A continuación se describen los estándares de codificación que establece Drupal y que fueron utilizados en la implementación de los módulos.

Para la declaración del código se tuvieron en cuenta los siguientes estándares (Gil, 2012):

Indentación:

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos. En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea.

```

<?php
function formulario_permission() {
return array(
'access formulario content' => array(
'title' => t('Access content for the Current posts module')),);
}

function formulario_menu(){
$items['site_generation'] = array(
'title' => t('Generar portal'),
'page callback' => 'drupal_get_form',
'page arguments' => array('formulario_generador_portal'),
'access arguments' => array('access formulario content'),
'description' => t(''),
'type' => MENU_NORMAL_ITEM,
'menu_name' => 'navigation',
'weight' => 1,);
}

```

Ilustración 9 Estándares de codificación: indentación

Etiquetas de apertura y cierre de PHP:

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas <?php y ?>, y en ningún caso la versión corta <? y ?>. En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos .module y .inc. Esta convención evita que puedan quedarse olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre ?>), que serían identificados como salida HTML y podrían provocar un error muy típico, *"Cannot modify header information - headers already sent by..."*.

Por tanto, la etiqueta de cierre final del archivo (?>) es opcional en Drupal.

No hay que confundir esto con el uso normal del lenguaje PHP en archivos que también contienen HTML (como por ejemplo los archivos de plantilla .tpl.php), donde cada fragmento de PHP debe llevar sus correspondientes etiquetas de apertura y cierre, para diferenciarlo del código HTML.

```

<?php
function formulario_permission() {
    return array(
        'access formulario content' => array(
            'title' => t('Access content for the Current posts module')),);
    }
}

```

Ilustración 10 Estándares de codificación: etiquetas de apertura y cierre de PHP

Operadores:

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, `$numero = 3`, en lugar de `$numero=3`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `>`, `<`, `.` (Concatenación de cadenas), `.=`, `+=`, `-=`, etc.

Los operadores unarios como `++`, `--` no deben tener separación. Por ejemplo, `$numero++`.

Uso de comillas:

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

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, `"<h1>${title}</h1>"`. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

Uso de punto y coma (;) en código PHP:

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo `<?php print $title ?>`. En Drupal es **siempre** obligatorio: `<?php print $title; ?>`.

- Correcto: `<?php print $title; ?>`

- Incorrecto: `<?php print $title ?>`

Estructuras de control:

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:

- Debe haber un espacio entre el comando que define la estructura (*if*, *while*, *for*, etc.) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones, como verá más adelante.

- La llave de apertura { se situará en la misma línea que la definición de la estructura, separada por un espacio.

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

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

```

function valores_default($campo) {
    if (empty( $_GET[$campo])) {
        if ($campo == 'modulos') {
            return array();
        }
        else {
            return null;
        }
    }
    else {
        return $_GET[$campo];
    }
}

```

Ilustración 11 Estándares de codificación: estructuras de control

Funciones:

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones.

En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. Ejemplo:

function forum_help(\$path, \$arg) {

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra en el siguiente ejemplo:

\$field = field_info_instance('node', 'taxonomy_forums', \$node->type);

Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque:

```
$numero1 = foo($a, $type);  
$primer_valor = foo2($b);  
$i = foo3();
```

Arrays:

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

Cuando la línea de declaración del *array* supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```
$vector1 = array(1, 2, 'clave' => 'valor');  
$vector2 = array(  
    'forum' => 'foro1',  
    'template' => 'forums',  
    'arguments' => array('tid' => NULL, 'topics' => NULL),  
    'size' => 128,  
);
```

Nombres de archivos:

Los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tendrás extensión `.txt` y el nombre es mayúsculas. Por ejemplo, `README.txt`, `INSTALL.txt`, etc.

Comentar el código:

En este apartado se debe diferenciar entre los comentarios para aclarar determinados fragmentos de código, que se insertan en cualquier punto del mismo, y los **comentarios de documentación**.

Los comentarios de documentación suelen escribirse al principio de un archivo o de cada función y se utilizan para generar documentación de ayuda a través de aplicaciones que extraen la información a partir de las etiquetas empleadas.

En el primero de los casos se pueden utilizar las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (por ejemplo `TRUE` o `FALSE`).

3.3 Pruebas

Las pruebas son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software. En este proceso se ejecutan pruebas dirigidas al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. (Oré B, 2009)

3.3.1 Pruebas funcionales

Las pruebas funcionales están centradas en comprobar que las funcionalidades descritas en el documento de requisitos del sistema se cumplen con la implementación realizada. A este tipo de pruebas también se les denomina pruebas de comportamiento o de caja negra, debido a que los analistas enfocan su atención a las respuestas del sistema de acuerdo a los datos de entrada y sus resultados en los datos de salida, los cuales se definen generalmente en los casos de prueba que se crean antes del inicio de las pruebas. (Oré B, 2009)

A continuación se muestra el diseño de casos de pruebas correspondiente a la funcionalidad Autenticar usuario. Los casos de pruebas restantes se encuentran en los anexos

Autenticar usuario

Tabla 2 Diseño del caso de prueba Gestionar categoría

Escenario	Descripción	Nombre de usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario correctamente	El sistema le permite al usuario autenticarse correctamente	Se escribe el Nombre de usuario del portal	Se escribe la Contraseña asignada a ese	El sistema permite el acceso del usuario a la sección.	1. El usuario accede al portal, en la parte superior del navegador donde se escribe la url, escribir a continuación de la dirección del sitio <i>user</i> por ejemplo "

			nombre de usuario		<p>http://10.53.7.239/OPCI/user", luego escribe su usuario y contraseña y da clic en la opción Iniciar sesión.</p> <p>2. El sistema verifica que los datos ingresados estén correctos y los campos obligatorios completos.</p> <p>3. El sistema muestra la página principal del portal y las opciones correspondientes.</p>
EC 1.2 Autenticar usuario incorrectamente	El usuario se autentica incorrectamente	Se ingresa un Nombre de usuario que no pertenece al portal, o se deja el campo Nombre de usuario vacío	Se ingresa la Contraseña incorrecta, o se deja el campo Contraseña vacío	El sistema no permite el acceso del usuario a la sección, muestra los mensajes de error correspondientes y una vista nueva con las opciones de Acceder . En caso de que los campos estén vacíos el mensaje para cada campo es: "El campo X es obligatorio" "En caso que los datos sean incorrectos el	3. El sistema muestra nuevamente el formulario para el ingreso de usuario y contraseña. 4. El usuario ingresa los datos correctos y da clic en el botón Iniciar sesión . 5. El sistema verifica que los datos ingresados estén correctos y los campos obligatorios completos. 6. El sistema muestra la página principal del portal y las opciones correspondientes.

				mensaje es: Lo sentimos. No reconocemos el nombre de usuario o la contraseña. ¿Olvidó su contraseña?	
--	--	--	--	--	--

Descripción de las variables.

Tabla 3 Descripción de las variables del diseño del caso de prueba Gestionar categoría

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre de usuario	Campo de texto	No	El campo Nombre de usuario solo acepta letras.
2	Contraseña	Campo de texto	No	La contraseña puede estar compuesto por cualquier carácter alfanumérico: letras, números, caracteres extraños, mayúsculos o minúsculos.

3.3.2 Pruebas de carga

Las pruebas de carga permiten la simulación del acceso de muchos usuarios a un servidor al mismo tiempo, posibilitando observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. La carga pueden ser el número de usuarios concurrentes que se espera que utilicen la aplicación, y que realizan durante el tiempo en que dura la carga un número específico de transacciones. Este tipo de prueba facilita la monitorización del servidor y la base de datos para encontrar posibles cuellos de botella en la aplicación, y puede mostrar los tiempos de respuesta de todas las transacciones importantes (msdn.microsoft.com, 2013).

3.3.3 Pruebas de estrés

Las pruebas de estrés posibilitan la obtención de datos sobre la carga del sistema. Tiene como objetivo generar cargas en el sistema hasta hacerlo inutilizable, para centrarse en verificar la calidad de los mensajes de error o establecer alertas para anticipar los fallos. Las pruebas de estrés son uno de los últimos tipos de pruebas que se deben efectuar, debido a que tienen un carácter poco realista ya que podría darse el caso que nunca se diera en la vida real la situación de carga simulada (Fraile, 2011).

3.4.4 Pruebas de seguridad

Las pruebas de seguridad permiten realizar una evaluación de los sistemas desde el punto de vista externo y sin conocimiento previo del mismo. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación. Además buscan medir la confidencialidad, integridad y disponibilidad de los datos, partiendo de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Una vez terminadas las pruebas es posible medir y cuantificar los riesgos a los cuales se ven expuestos aplicativos de la infraestructura interna y externa (vyvquality.com, 2012).

3.5 Resultados de las pruebas

3.5.1 Pruebas Funcionales

En la primera iteración se detectaron 12 no conformidades, de las cuales 6 no conformidades fueron de validación, 3 no conformidades de error de idioma y 3 no conformidades de Redacción-CP. En la segunda iteración 3 no conformidades de Recomendación. Todas las no conformidades encontradas fueron resueltas por el equipo de desarrollo como se muestra en la ilustración 9.

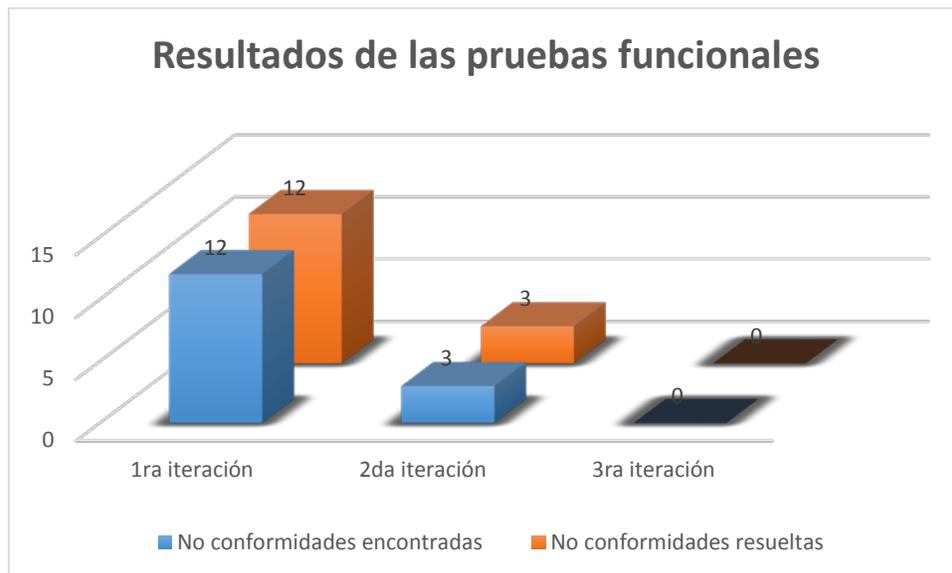


Ilustración 12 Resultados de las pruebas Funcionales

3.5.2 Resultados de las pruebas de carga y estrés

Las pruebas de carga y estrés se desarrollaron con la herramienta Apache JMeter en su versión 2.12, JMeter es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web. JMeter permite realizar pruebas Web clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y WebServices (en Beta). Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Además activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario.

Para la realización de las pruebas fue necesario tener en cuenta las condiciones del escenario, tanto del hardware como software, donde se encuentra el sistema; para obtener una correcta información del comportamiento y resultados en general. Por tanto se hizo necesario simular las pruebas en un escenario con las características siguientes:

Hardware:

- Microprocesador Intel Core i5-3210M a 2.50GHz.
- Memoria RAM 8GB.

Software:

- Tipo de Servidor Web: Apache v2.2.21.
 - Máximo de hilos concurrentes (simulación de usuarios): 30.
- Servidor de BD: PostgreSQL 9.2.
- Sistema operativo Windows 8

Los resultado obtenidos fueron que para 10 usuarios concurrentes esperados la transferencia de datos alcanzó un total de 1429,50 kb/seg lo que incurrió en un rendimiento de 36,3/ seg. Se demuestra que la propuesta de solución es satisfactoria, ya que mantuvo prestando servicios todo el tiempo sin incurrir en fallos. Además se probó para un total de 10, 20 y 30 usuarios concurrentes, obteniendo los resultados que se muestran en la ilustración 10:

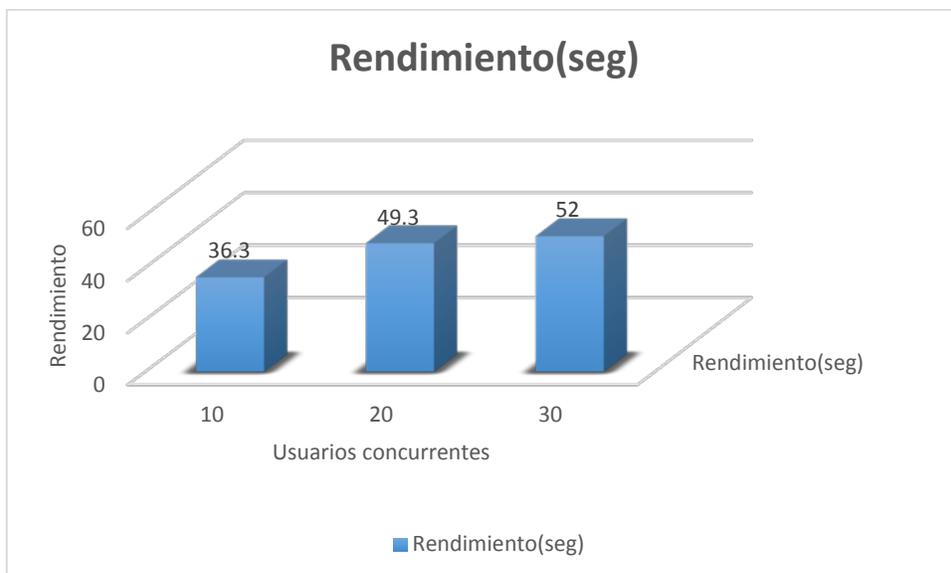


Ilustración 13 Resultados de las pruebas de carga y estrés

Resultados de la prueba de seguridad:

Se utilizó la herramienta websecurify scanner 0.9 para realizar las pruebas de nivel dos, que divide la criticidad de los errores en cuatro niveles (Alto, Medio, Bajo e Informativo). Se detectaron ocho no conformidades: dos de nivel medio (Envío de credenciales por protocolos inseguros y posible vulnerabilidad a un ataque de falsificación de petición de sitios cruzados), dos de nivel bajo y cuatro informativos, los cuales fueron comprobados manualmente y solucionados en la aplicación.

3.6 Conclusiones

- En este capítulo la confección del diagrama de componentes ofreció una vista arquitectónica de alto nivel para ayudar al equipo de desarrollo en la implementación.
- La realización de las pruebas de funcionalidad, seguridad y carga y estrés, encaminó la calidad de la aplicación posibilitando determinar y asegurar la calidad y robustez del portal.
- El estándar de codificación empleado facilitó la comprensión del código fuente por los miembros del equipo de desarrollo, permitiendo desarrollar el portal en un menor tiempo.

CONCLUSIONES GENERALES

Una vez realizada la fundamentación teórica que sustentó la presente investigación, definidas las características del generador de portales web y efectuado su desarrollo y validación, se obtuvieron resultados que permiten a los autores arribar a las siguientes conclusiones:

- El estudio realizado de las tendencias actuales de los generadores de portales web permitió identificar las principales funcionalidades.
- El análisis y la selección de las tecnologías y herramientas adecuadas hicieron que se implementara un sistema de fácil manejo con una gestión de contenidos y de datos segura.
- La metodología de desarrollo de software ágil OpenUp guio el ciclo de vida completo de la implementación del portal web, garantizando el óptimo desarrollo, logrando la documentación necesaria desde el comienzo del mismo.
- Con la implementación del generador de portales web se logró crear una herramienta que facilita el trabajo de los desarrolladores ya que permite generar un portal base, con los principales tipos de contenido y sus dependencias, según la categoría, logrando así que los desarrolladores se enfoquen en características específicas de cada portal ahorrando recursos al departamento SENIT del Centro de Ideoinformática (CIDI) de la Facultad 1, en la Universidad de las Ciencias Informáticas.

Recomendaciones

Permitir una mayor personalización de la portada de los sitios a generar

Incorporar nuevas categorías.

Incorporar nuevos temas.

Incorporar más tipos de contenidos para las categorías existentes.

Referencias bibliográficas:

CHARTE OJEDA, Francisco. SQL [en línea]. [s.l.]: Anaya Multimedia, 2009 [fecha de consulta: 27 Octubre 2011]. Disponible en:

<http://books.google.com.cu/books?id=TZJ5QgAACAAJ&dq=sql&hl=es&sa=X&ei=xvSzT8ScLImJ6AGzt7yuCQ&ved=0CDgQ6AEwAQ>.

cursosdrupal.com. Arquitectura de Drupal [en línea] [Fecha de consulta:24 de febrero 2015] Disponible en:

<http://www.cursosdrupal.com/content/arquitectura>

GALEANO GIL, Germán, SÁNCHEZ ALONSO, José Carlos y DÍAZ MÁRQUEZ, Pablo. *HTML: manual imprescindible* [en línea]. [s.l.]: Anaya Multimedia, 2009 [fecha de consulta: 1 de noviembre 2014]. Disponible en:

<http://books.google.com.cu/books?id=-WuJPgAACAAJ&dq=HTML&hl=es&sa=X&ei=8LqzT7u8LojBtqf4kIWeAw&ved=0CFkQ6AEwBzqK>.

COGGESHALL, John. *La biblia de PHP 5* [en línea]. [s.l.]: Anaya Multimedia, 2005.

[fecha de consulta: 24 Octubre 2014] Disponible en:

<http://books.google.com.cu/books?id=D3jkPAAACAAJ&dq=php+5&hl=es&sa=X&ei=x7KzT62BGITWtgfGhvixCA&ved=0CEQQ6AEwAjqK>.

D. Miguel Ángel Ladrón Jimenez. 2014. *Sistema operativo, búsqueda de la información: Internet y correo electrónico.* [fecha de consulta: 24 de octubre 2014].

drupal.org, 2013. Drupal 6/7 programación desde una perspectiva orientada a objetos.

[en línea] [fecha de consulta: 24 de febrero 2015] disponible en:

<https://www.drupal.org/node/547518>

Fraile, Luis . 2011. *Pruebas de rendimiento.* [En línea] 2011. [fecha de consulta: 11 de abril 2015] Disponible en: <http://www.globetesting.com/pruebas-de-rendimiento/>.

Gil, Fran. 2012. *Experto en Drupal 7 Nivel avanzado. Curso de creación y gestión de portales web con Drupal 7.*[En línea] 17 de Octubre de 2012.

García , Josep. ¿Qué es un creador de páginas web? [en línea] [fecha de consulta: 25 Octubre 2014]. Disponible en:

<http://www.websitetooltester.com/es/que-es-un-creador-de-paginas-web>.

LARMAN, Craig. UML y Patrones [en línea]. México D.F.: Prentice Hall, 1999 [fecha de consulta: 25 Octubre 2014]. Disponible en:

http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/UML_y_Patrones.

Oré B, Ing. Alexander. Pruebas Funcionales. 2009. [En línea] [fecha de consulta: 9 de Abril del 2015]. Disponible en:

http://www.calidadyssoftware.com/testing/pruebas_funcionales.php

Pecos, Daniel. 2008. *PostGreSQL vs. MySQL.* 2008.

Pgadmin. Introduction [en línea]. 2011 [fecha de consulta: 6 Noviembre 2014]. Disponible en: <http://www.pgadmin.org/>

Rivera Jony. Ingeniería del software. 2014. [En línea][Fecha de consulta: 25 de noviembre del 2014]. Disponible en:

<http://ingsofrivera.blogspot.com/2014/08/diagrama-de-componentes-casos-de-usos.html>
<http://ingsofrivera.blogspot.com/2014/08/diagrama-de-componentes-casos-de-usos.html>

Rodríguez , Ana P., Polanco, Josué y Hernández, Darwin. 2013.Open UP. 2013

[fecha de consulta 12 de diciembre 2014] disponible en:

<http://es.scribd.com/doc/37116717/Open-Up#scribd>

SÁNCHEZ MAZA, Miguel Ángel. Javascript [en línea]. [s.l.]: Editorial Innovación y Cualificación, 2006 [fecha de consulta: 24 Octubre 2014]. Disponible en:

http://books.google.com.cu/books?id=3x09sewjaHIC&printsec=frontcover&dq=javascript&hl=es&ei=wpC5TsrED62DsgLs9ZSjCA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCwQ6AEwAA#v=onepage&q&f=false.

Shawn Lawton, Henry . 2004. *Accesibilidad Web.*

[En línea] 2004. [fecha de consulta: 10 Abril 2015].Disponible en:

<http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

sophia.javeriana.edu.co [en línea]. [fecha de consulta: 24 febrero 2015]. Disponible en:

<http://sophia.javeriana.edu.co/~metorres/Diplomado/7%20-%20Curso%20Arquitecturas%20-%20Interaccion.pdf>

Universidad Santos Tomas [en línea].

Disponible en:

<http://www.ustamed.edu.co/sistemas/index.php/frentes/sistemas/soporte-a-servidores>.

VAN LANCKER, Luc. *CSS 1 y CSS 2.1: Hojas de estilo para enriquecer el código HTML* [en línea]. Barcelona: Ediciones ENI, 2007 [fecha de consulta: 24 Octubre 2014]. Disponible en:

http://books.google.com.cu/books?id=xBrgWWa31pcC&printsec=frontcover&dq=css&hl=es&sa=X&ei=n7WzT_SCDliXtweouY3NCA&ved=0CGYQ6AEwCQ#v=onepage&q=css&f=false.

Visconti, M. y Astudillo, H. 2010. *“Fundamentos de Ingeniería de Software”*. [En línea] 2010. [Citado el: 18 de Febrero de 2015.] Disponible en: [http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-](http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-DisenoOO.pdf)

[DisenoOO.pdf](http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-DisenoOO.pdf).

Visual Paradigm for UML [en línea]. 2011 [fecha de consulta: 6 Noviembre 2014].

Disponible en: <http://www.visual-paradigm.com/product/vpum/>

vyvquality.com. 2012. *Pruebas de seguridad*. [En línea] 2012. [Citado el: 11 de Abril de 2015.] Disponible en: <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-seguridad.html>