



Universidad de las Ciencias Informáticas

Facultad 1

Perfil de instalación de Drupal 7 para el Departamento de Servicios Informáticos para Internet

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

Autor

Leonar Pina Legón

Tutores

MSc. Miguel Jaeger Rodríguez Lazo

Ing. Liuba Torres Almira

Ing. Adrián Garbey Fonseca

La Habana, Cuba

Junio de 2011



“La adquisición de cualquier conocimiento es siempre útil al intelecto, que sabrá descartar lo malo y conservar lo bueno.”

Leonardo Da Vinci.

AGRADECIMIENTOS |

Agradezco:

A mis padres, por ser las personas más especiales de mi vida, mis ejemplos y por dar todo lo que pueden.

A mis tías y tíos, por apoyarme en cada decisión que tomo, por su dedicación y todo su cariño, los quiero mucho.

A mi familia, por amarme y apoyarme en todo momento.

A mis amigos, en especial a las 4 personas que siempre discutíamos, pero estuvieron desde el día número 1, a Grillo que siempre será la hermana que nunca tuve, a Paquete de Pelly donde siempre va a tener un lugar especial en mi familia, a Pipa por tenerme presente en todo momento y el Noni por ser un verdadero amigo en todos los momentos siempre serás mi hermano, nunca se me olvidarán, siempre van a tener un significado en mi vida

A Lyen, Dayani, Yoan, Leo y Franklin por brindarme su amistad en todo momento.

A los amigos que, aunque empezó tarde la amistad sé que va a estar ahí para toda la vida como Yisel, Neivis, Victor, Neobel, Arlenis, Luisi.

A mis compañeros del grupo nuevo y viejo, por formar parte de esta aventura, no los voy a olvidar.

A mis amigos de Sancti Spiritus Ana Miriam, Ernesto y Yasiel, mis hermanos del alma.

A Lala y su mamá Isabel por incluirme en su familia.

AGRADECIMIENTOS |

A Yoyi por cuidarme como si fuera otro de sus hijos, así como agradecerle a su hija Liliana.

A Martha y Elba por cuidarme todos estos tiempos.

A unas personas que me ayudaron y me hicieron este año lo mejor posible a Yanet, Raúl, José, Dairon y todos los del apartamento

A mis tutores, por su apoyo y ayuda brindada confiando en mí en todo momento.

A los profesores, por su tiempo, y todas sus enseñanzas.

A todas las personas que hicieron posible este sueño.

Gracias

Dedicatoria

A mis padres por ser mis fanáticos # 1.

*Por confiar en mí y
apoyarme siempre para que este sueño se hiciera realidad.*

DECLARACIÓN DE AUTORÍA |

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro de Ideoinformática (CIDI) de la Universidad de las Ciencias Informáticas (UCI) que haga uso del mismo en su beneficio.

Para que así conste se firma el presente a los ____ días del mes de junio de 2016.

Firma del Autor

Leonar Pina Legón

Firma del Tutor

Ing. Adrián Garbey Fonseca

Firma del Tutor

Ing. Liuba Torres Almira

Firma del Tutor

MSc. Miguel Jaeger Rodriguez Lazo

Resumen

El presente trabajo de diploma consiste en el desarrollo de un perfil de instalación para el Departamento de Servicios Informáticos para Internet (SENIT) con el objetivo de incluir características y componentes necesarios para el desarrollo personalizado de portales web. Para la implementación de la propuesta de solución se utilizó Drupal 7 como Sistema Gestor de Contenidos y para guiar este proceso la metodología de desarrollo de software AUP-UCI. Se empleó como Entorno de Desarrollo Integrado NetBeans 8.0, Lenguaje de Marcado HTML 5, Hoja de Estilo en Cascada en su versión 3, PHP 5.5.9 como lenguajes de programación y como herramienta para las pruebas de seguridad WebSecurify 0.8. El perfil de instalación cuenta con funcionalidades que permiten la gestión de evento, noticias, productos, galería de imágenes y vídeos entre otros. Como resultado, la solución permite el ahorro en tiempo y esfuerzo del trabajo de los especialistas; disminuyendo los costos de desarrollo de los portales. Para verificar el cumplimiento de las funcionalidades propuestas se realizaron pruebas que permitieron detectar errores que, al ser corregidos, evidenciaron que una vez instalado el perfil, los portales web a desarrollar son una solución funcional.

Palabras clave: desarrollo, Drupal, instalación, perfil, portales.

Introducción	1
Capítulo 1: Marco teórico-conceptual para el desarrollo del perfil de instalación de Drupal 7 para el departamento de SENIT	5
1.1. Introducción	5
1.2. Portal web.....	5
1.3. Herramientas, tecnologías y metodología	6
1.4. Sistema de gestión de contenidos (CMS)	12
1.5. Conclusiones parciales	20
Capítulo 2: Propuesta de solución y características del perfil de instalación de Drupal 7 para el departamento de SENIT	21
2.1. Introducción	21
2.2. Propuesta de solución	21
2.3. Modelo de dominio	24
2.4. Captura de requisitos.....	25
2.5. Patrones de diseño.....	33
2.6. Arquitectura del perfil de instalación	37
2.7. Conclusiones parciales	39
Capítulo 3: Implementación y prueba del perfil de instalación de Drupal 7 para el departamento de SENIT	41
3.1. Introducción	41
3.2. Diagrama de componentes	41
3.3. Estándares de codificación	42
3.4. Pruebas de software	45
3.5. Conclusiones parciales	54
Conclusiones Generales	55
Recomendaciones	56
Referencias.....	57
Bibliografía consultada.....	61

Introducción

En la actualidad con el amplio desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), es mayor el número de empresas que tienen la necesidad de introducir innovaciones tecnológicas para ser más eficientes y lograr un adecuado nivel de gestión de sus negocios en los escenarios en que se desarrollan, pues los diferentes clientes o usuarios son cada vez más exigentes, sobre todo, en términos de flexibilidad, rapidez y calidad, de manera que, estos procesos tecnológicos constituyen el eje central de las actividades de las diferentes organizaciones (Iberestudio, 2010).

Un ejemplo de este tipo de tecnología lo constituyen los portales web que además son capaces de ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y servicios relacionados a un mismo tema. En una empresa, un portal web, significa un cambio en cuanto a la forma de socializar información y darse a conocer. Por lo que están conscientes que para introducirse en el mercado y poder incrementar su competitividad actualmente tiene que desarrollarse en un entorno diferente, que es mucho más exigente, muy dinámico y sobre todo eficiente. Los portales son un elemento fuerte para las empresas ya que se fundamenta en el análisis continuo para poder ser un mercado competitivo.

Cuba no ha estado ajena al desarrollo ocurrido a nivel mundial, es por ello que se han realizado nuevos proyectos en diferentes esferas de la sociedad, los cuales han propiciado la creación de diversas instituciones que se dediquen al desarrollo de soluciones informáticas. Un ejemplo actual de estas transformaciones en la rama educacional lo constituye la apertura de la Universidad de las Ciencias Informáticas (UCI). En la cual se encuentra un equipo de especialistas en desarrollo web del departamento de servicios informáticos para internet (SENIT) en el centro de Ideoinformática (CIDI), el cual tiene como objetivo el desarrollo de portales para las organizaciones, instituciones o empresas que la requieran.

Por tanto, los principales esfuerzos deben ser enfocados en definir cuáles son esos procesos claves y qué servicios tiene disponible, por lo que, los ingenieros de software son los mayores responsables de una correcta identificación de estos. Una vez definidos estos servicios, se comienza un trabajo de análisis, diseño e implementación de una posible solución del portal web, en muchos casos demorando meses para su despliegue. Las demoras de diferentes procesos asociados al desarrollo de estos portales generan dificultades en cuanto a la planificación, el análisis, y tiempo de entrega de una solución, debido

principalmente a que se dispone de poco personal capacitado para atender la gran demanda de solicitudes. Los especialistas muchas veces experimentan una excesiva carga de trabajo, además disponen de poco personal para acometer el desarrollo.

Los portales que se desarrollan integran una serie de servicios comunes para todas las empresas, entre los que destacan: gestión de eventos, avisos, noticias, galería de imágenes y efemérides. Debido a la re-implementación de estas funcionalidades en todos los portales que se desarrollan, el trabajo de los especialistas se hace mucho más complejo a la hora de desarrollar varias soluciones; se dificulta el proceso de liberación por calidad incrementando el tiempo de desarrollo, lo que trae consigo que muchas veces no se cumple con el tiempo previsto para su despliegue.

Por todo lo antes descrito, se plantea el siguiente **problema de investigación**: ¿Cómo facilitar el proceso de desarrollo de portales web en el departamento SENIT?

Para la presente investigación se plantea como **objeto de estudio**: el proceso de desarrollo de perfiles de instalación para el CMS¹ Drupal 7. Se define como **campo de acción**: el proceso de desarrollo de un perfil de instalación de portales web para SENIT sobre el CMS Drupal 7.

Se establece como **objetivo general**: desarrollar un perfil de instalación para el departamento SENIT que incluya características y componentes necesarios para el desarrollo personalizado de portales web.

Para el cumplimiento del objetivo general se definen los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico-conceptual de la investigación.
- ✓ Identificar servicios y funcionalidades comunes en portales web.
- ✓ Implementar módulos y funcionalidades necesarias para el perfil de instalación.
- ✓ Validar la solución desarrolladas a través de pruebas de *software*.

¹ *Content Management System*, que se traduce directamente al español como Sistema Gestor de Contenidos. Como su propio nombre indica, es un sistema que nos permite gestionar contenidos (Casas Cervero, 2011).

Para el desarrollo del presente trabajo se plantea la siguiente **idea a defender**: con la implementación del perfil de instalación SENIT debe disponer de un punto de partida para la creación de portales web y se facilitará el proceso de desarrollo en el departamento SENIT.

Para el desarrollo de la investigación se utilizan los siguientes métodos:

Métodos teóricos:

- **Análisis Histórico-Lógico**

Este método permite estudiar el avance y las novedades que ha tenido el proceso de desarrollo de módulos y perfiles de instalación a través de las diferentes versiones del CMS Drupal, y su aplicación en distintos sistemas web.

- **Inductivo-Deductivo**

Permite arribar a conclusiones lógicas a partir de los conocimientos adquiridos, además este se utilizó para el planteamiento de la idea a defender, el objetivo general, y de algunos requisitos principales para portales web.

- **Analítico-Sintético**

Este método se utiliza para seleccionar y resumir varios de los elementos más importantes relacionados con módulos y perfiles de instalación, además se detallará toda la información relacionada para el modelado de negocio y desarrollo de portales web de SENIT.

- **Modelado**

Este método se utiliza con el objetivo de materializar la representación abstracta de determinadas características del sistema mediante la construcción de diagramas y modelos a lo largo del desarrollo de la investigación.

Métodos empíricos

Entrevista

Se utilizará este método para entrevistar a los especialistas y obtener información acerca de las características, cualidades y requisitos funcionales que debe tener el perfil de instalación SENIT para portales. Además, su uso constituye un medio para recopilar información necesaria y obtener un

conocimiento cualitativo de las características del portal y a su vez para el posterior desarrollo de la investigación.

El presente documento está estructurado por los siguientes capítulos:

Capítulo 1: Marco teórico-conceptual para el desarrollo del perfil de instalación de Drupal 7 para el departamento SENIT

Se tratan los aspectos teóricos que apoyan la investigación, se presentan conceptos fundamentales, se reflejan las herramientas y tecnologías a utilizar en el desarrollo de la aplicación, así como un estudio de varios perfiles de instalación y conceptos asociados. Se detallan, además, elementos del CMS Drupal y su integración con perfiles de instalación como uno de sus componentes para agilizar los trabajos de desarrollo web.

Capítulo 2: Propuesta de solución y características del perfil de instalación en Drupal 7 para el departamento SENIT

Se expone la solución para agilizar el desarrollo de portales web para SENIT, se detallan las características propias del perfil de instalación y se describe el flujo de los procesos involucrados en la solución. Además, se explican las actividades de ingeniería y gestión de *software* llevadas a cabo durante el desarrollo del perfil.

Capítulo 3: Implementación y prueba del perfil de instalación en Drupal 7 para el departamento SENIT

Se especifican los estándares de codificación utilizados. Se presenta, la manera en que se llevó a cabo la informatización de los servicios identificados como básicos para un portal. Se realizan las pruebas necesarias para comprobar su correcto funcionamiento del perfil de instalación.

Capítulo 1: Marco teórico-conceptual para el desarrollo del perfil de instalación de Drupal 7 para el departamento de SENIT

1.1. Introducción

En este capítulo se citan las principales definiciones de interés asociadas al dominio del problema y que se hace necesario para entender el desarrollo de la investigación. Se realiza un estudio del estado del arte para lograr un acercamiento a las clasificaciones de portales web, desarrollos ágiles, tecnologías y herramientas en el desarrollo del perfil de instalación de portales web para SENIT.

1.2. Portal web

Unos de los tipos de clasificaciones para los sistemas web, son los llamados portales web, definiéndose como un sitio web que sirve como pasarela a internet (Vega Lebrún, 2007). El concepto de portal web se ha hecho muy ambiguo en la web en los últimos años con el desarrollo acelerado de las tecnologías al tornarse un poco dudoso su clasificación, ya que muchos sitios brindan gran cantidad de servicios y recursos para el usuario, aún sin estar definidos conceptualmente como portales (Definición ABC, 2007).

Se podría ofrecer un concepto bastante básico diciendo que es un punto de entrada a internet donde se organizan sus contenidos, ayudando al usuario y concentrando servicios y productos de forma que le permitan realizar cuanto necesite hacer en la red a diario, o al menos que pueda encontrar allí todo cuanto utiliza cotidianamente sin necesidad de salir de dicho sitio. Un portal es, en todos los casos, un sitio web, una página, pero no viceversa; de igual forma ni cualquier página ni cualquier sitio web serían un portal (García Gómez, 2013).

Los portales ofrecen de forma integrada, una amplia variedad de servicios y recursos al usuario. Por lo general, los portales ofrecen noticias actualizadas al instante, buscadores, foros para compartir opiniones, chat, juegos online, tiendas virtuales para realizar compras electrónicas, servicio de correo electrónico, entre

otros servicios, (...) intentan ofrecer un amplio abanico de opciones, de forma tal que puedan convertirse en la puerta de ingreso del internauta a la web. Se supone que el usuario tendrá al portal como su página de inicio y utiliza todos sus servicios antes de comenzar a navegar en otros sitios más específicos (Jimenez, 2015).

1.3. Herramientas, tecnologías y metodología

En este epígrafe se presentan las tecnologías, herramientas y metodología empleadas. Se justifica la selección de las técnicas de modelado y lenguajes para el desarrollo del perfil de instalación para portales web para SENIT utilizando Drupal 7. Esta selección se realizó teniendo en cuenta los avances actuales de la web, así como las alternativas (sistemas de gestión de contenido, HTML, CSS, entre otros) que hacen más eficiente el funcionamiento del CMS.

Lenguajes de programación

Lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. El programador es el encargado de utilizar un lenguaje de programación para crear un conjunto de instrucciones que, al final, constituirá un programa o subprograma informático. Los lenguajes de programación pueden clasificarse según el paradigma que usan en: procedimentales, orientados a objetos, funcionales, lógicos, híbridos, etc. (Alegsa, 2010).

HTML 5

HTML² es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque sí le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados (Lubbers, 2005).

Se trata de un formato abierto que surgió a partir de las etiquetas SGML (*Standard Generalized Markup Language*), concepto traducido generalmente como “Estándar de Lenguaje de Marcado Generalizado” y

² Lenguaje de Marcas de Hipertexto.

que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se emplea al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto.

CSS 3

CSS (*Cascading Style Sheets*) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML³. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (Mcfarland, 2012).

JQuery 1.8

JQuery es un marco de trabajo para el lenguaje Javascript (abreviado comúnmente "JS"), este lenguaje además de ofrecer una infraestructura con la que se tiene mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente con jQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. (Matthew, 2011 pág. 42)

PHP 5.5.9

PHP (acrónimo de "PHP: *Hypertext Preprocessor*") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor donde se puede procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir *cookies* (Bakken, 2001).

Entorno de desarrollo

Netbeans 8.0

Se seleccionó como entorno de desarrollo NetBeans en su versión 8.0 por sus ventajas y características. Es libre y gratuito no cuenta con restricciones de uso, su interfaz es amigable e intuitiva, posee además

³ Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos (W3C, 2014).

todas las herramientas para crear aplicaciones profesionales ya sean de escritorio, web y móviles. Es multiplataforma y posee una creciente comunidad de desarrolladores, trabajando constantemente para su mantenimiento y mejora (NetBeans IDE, 2015).

Herramientas CASE

Lenguaje Unificado de Modelado.

El lenguaje unificado de modelado (por sus siglas en inglés UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura las decisiones y los conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas. UML incluye conceptos semánticos, notación, y principios generales (Jacobson, y otros, 2011).

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión (Hernández Orallo, 2012). UML es además un método formal de modelado, lo que aporta las siguientes ventajas:
 - ✓ Mayor rigor en la especificación.
 - ✓ Permite realizar una verificación y validación del modelo realizado.
 - ✓ Se pueden automatizar determinados procesos y permite generar códigos a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto (Hernández Orallo, 2012).

Visual Paradigm 8.0

Es una herramienta CASE (*Computer Aided Software Engineering*). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Se caracteriza por soporte de UML (*Unified Modeling Language*) versión 2.1, además de un diseño centrado en casos de uso y enfocado al negocio que generan un *software* de mayor calidad (Madroñero Casadomé, 2015).

Proceso de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. Se van indicando todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben tener. Además, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (Barzanalla, 2011).

Existe una gran variedad de metodologías para la creación del software, las que se clasifican en dos grandes grupos:

- ✓ Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas Metodologías pesadas.
- ✓ Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas Metodologías ligeras/ágiles (Carrillo Pérez, y otros, 2008).

Para la realización del perfil de instalación de portales web utilizando Drupal 7 se usa el siguiente proceso de desarrollo de software definido por la UCI basado en la unión del CMMI-DEV v 1.3, la metodología “Proceso Unificado Ágil” (AUP).

AUP

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de *Rational*, también es un marco de trabajo extensible.

Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental (Flores, 2014). Las disciplinas que rigen esta metodología se conforman:

1. **Modelo:** Entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
2. **Implementación:** Transformar el modelo(s) en código ejecutable y realizar un nivel básico de pruebas individuales.
3. **Prueba:** Realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificar que se cumplan los requisitos.
4. **Despliegue:** Realizar un plan para la presentación del sistema y ejecutarlo para hacer que el sistema se encuentre a disposición de los usuarios finales.
5. **Gestión de Configuración:** Realizar la gestión de acceso a artefactos de su proyecto. Esto incluye no sólo el seguimiento de las versiones del artefacto en el tiempo, sino también el control y la gestión de cambios para ellos.
6. **Gestión del Proyecto:** Dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de los riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc.), y coordinar con las personas para garantizar que se entrega a tiempo y dentro del presupuesto.
7. **Ambiente:** Apoyar el resto de los esfuerzos por garantizar que el proceso adecuado, la orientación (normas y directrices), y herramientas (*hardware*, *software*, etc.) están disponibles para el equipo según cuando ellos lo necesiten.

Es notable que las disciplinas han cambiado; en primer lugar, la disciplina "Modelo" abarca el Modelado de Negocios, Requisitos y Análisis y Diseño del RUP⁴. El Modelo es una parte importante del AUP, como se puede ver, no domina el proceso. En segundo lugar, las disciplinas Configuración y Gestión del Cambio es ahora la Gestión de Configuración, en su desarrollo ágil de Gestión del Cambio son parte de los esfuerzos de gestión de requerimientos, que forma parte de la disciplina del Modelo (Karen, 2008).

Variación de AUP en la UCI

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del *software* que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el Modelo CMMI-DEV v1.3. El cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (UCI, 2014).

CMMI (Modelo de Madurez de Capacidad Integrado)



Figura 1: Logo del modelo CMMI (Nazcasof, 2008)

Las siglas de CMMI responden a *Capability Maturity Model Integration*, en español Integración de Modelos de Madurez de las Capacidades. Siendo un poco más claros, CMMI es un conjunto de modelos basados en

⁴ Proceso Racional Unificado.

las mejores prácticas en la gestión de los procesos, desarrollados a través de un proyecto conjunto en el que participaron el SEI (*Software Engineering Institute*), el gobierno estadounidense y algunos miembros de la industria. Dichos modelos establecen cinco niveles de ‘madurez’ de las organizaciones en función de si tienen o no una serie de características que detalla cada modelo. Las organizaciones pueden ser evaluadas y, en función de dicha evaluación, se las puede otorgar un nivel de madurez del 1 al 5 (Clikea, 2015).

Es decir, a través de CMMI, se puede saber el grado de ‘madurez’ de los procesos que tiene una organización, de acuerdo a un modelo de buenas prácticas. En principio, CMMI estaba orientado exclusivamente al desarrollo de software, pero se ha ido generalizando hasta finalmente derivar en los 3 modelos que conforman el conjunto:

- Desarrollo de productos y servicios (CMMI-DEV)
- Establecimiento y gestión de servicios (CMMI-SVC)
- Adquisición de productos y servicios (CMMI-ACQ)

En el desarrollo del perfil de instalación se tiene en cuenta la representación por etapas, debido a que este tipo de implementación está definido por la Universidad de las Ciencias Informáticas (UCI), por lo que el departamento lo adquiere para el desarrollo de sus proyectos.

1.4. Sistema de gestión de contenidos (CMS)

Un sistema de gestión de contenidos (CMS), es un software que permite crear una estructura base para la creación y administración de contenidos, principalmente de páginas web. Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan los contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web (Gil, 2012).

Características de los CMS

- ✓ Independencia del contenido respecto a la presentación.

- ✓ Facilita futuros cambios de diseño en la página web.
- ✓ Posibilita publicar un mismo contenido a través de varios canales a la vez.
- ✓ Organización de la información en tipos de contenido.
- ✓ Tratamiento individualizado de cada tipo de contenido.
- ✓ Flujo de edición y publicación.
- ✓ Creación y publicación de contenidos.
- ✓ Publicación y des-publicación programadas. Edición de contenidos supervisados por más de un usuario mediante flujos de trabajo (González Rodríguez, 2013).
- ✓ Existen CMS especializado según el tipo de gestión y contenidos que maneja, entre los que se encuentran:
 - ✓ CMS para sitios web
 - ✓ CMS para foros
 - ✓ CMS para blogs
 - ✓ CMS para galerías
 - ✓ CMS para wikis
 - ✓ CMS para comercio electrónico

El término *Content Management System* (CMS, siglas a las que responden en inglés los sistemas de gestión de contenidos) fue originalmente usado para la publicación de sitios web. Conforme en el mercado han evolucionado los productos marcados como CMS, también se ha ampliado el panorama con respecto al significado del término. Actualmente para referirse al fenómeno también se utiliza *Web Content Management* (WCM), es decir, sistema de gestión de contenidos web, dado el nivel de aplicabilidad y dependencia de este contexto (Rosell León, 2011).

Otras referencias lo definen como una aplicación informática usada para crear, editar, gestionar y publicar contenido digital en diversos formatos que genera páginas dinámicas interactuando con el servidor para

generar la página web bajo petición del usuario, con el formato predefinido y el contenido extraído de la base de datos del servidor (Díaz González, 2011).

De manera general se puede decir que un CMS es una herramienta que permite a los usuarios modificar, administrar, manejar y publicar de forma rápida y segura el contenido de una página web. Da la posibilidad de actualizar, tanto el contenido como el diseño del sitio web en forma regular sin necesidad de tener los conocimientos técnicos requeridos, además controlar que sólo las personas autorizadas accedan al sitio, mejorando de esta forma la calidad y eficiencia del mismo. Gracias a que posee una estructura modular, permite personalizar el desenvolvimiento de nuevos módulos específicamente para atender a las necesidades tanto de una empresa como de un usuario en particular (González Rodríguez, 2006).

Los Sistemas de Gestión de Contenido trabajan en conjunto con una base de datos que es utilizada para almacenar documentos, texto e imágenes, lo cual significa que el contenido puede ser reutilizado en muchos lugares dentro del sitio web. Estos generan automáticamente las páginas nuevas y sub-niveles de navegación, lo cual proporciona una gran interactividad. Además, posibilitan la aplicación de un mismo estilo a su entorno visual y una misma estructura mediante patrones de páginas, ofreciendo así una consistencia de la web.

Dentro de este tipo de CMS se encuentran Drupal, Joomla, TYPO3, PHP-Nuke, presentándose Drupal en su versión 7 como la mejor alternativa para el desarrollo del sistema por la gran variedad de beneficios y características que reúne para la realización de aplicaciones web.

Drupal

Es un Sistema de Administración de Contenido libre, bajo los términos de la Licencia Pública General (GPL), que se destaca por la calidad de su código y por la forma dinámica de generar las páginas. Su diseño es utilizado fundamentalmente para construir y gestionar un gran número de aplicaciones web. Extiende sus funcionalidades básicas mediante la implementación de módulos. El código fuente de los Sistemas de Administración de Contenido es abierto, lo cual posibilita que los desarrolladores puedan hacer modificaciones y mejoras al código, haciéndolo cada vez más potente. Drupal se caracteriza además por ser un potente gestor de contenido, que se encarga de formar grandes comunidades de usuarios registrados que colaboran en un fin común (Gil, 2012).

Ventajas del CMS Drupal frente a otros CMS

- ✓ Organización del contenido

Tiene una gran capacidad para almacenar, organizar y gestionar grandes volúmenes de contenido.

- ✓ Arquitectura potente, flexible y accesible

El sistema Drupal está concebido de base con una arquitectura que le otorga una gran flexibilidad: manejo análogo de diferentes tipos de contenido, categorización del contenido por taxonomías, código HTML generado de gran limpieza y accesibilidad (Pyme Networks, 2014).

- ✓ Capacidad para extender las funcionalidades

Drupal cuenta con numerosos módulos que extienden las funcionalidades básicas del sistema. En esto puede ser comparable a otros tipos de CMS, la diferencia con Drupal es la capacidad de integración entre todos los módulos para construir soluciones sólidas y escalables.

- ✓ Comunidad y modelo de desarrollo

Cada vez más usuarios, desarrolladores y empresas eligen Drupal como su CMS, la comunidad de colaboradores que aportan su esfuerzo en mejorar la solución es enorme y muy bien organizada, con gran espíritu de pertenencia al grupo, y compromiso con el proyecto. Todo ello es importante porque garantiza que el sistema no va a desaparecer, y que va a seguir creciendo y adaptándose a las nuevas necesidades que vayan apareciendo.

- ✓ Permisos

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

- ✓ Rendimiento

Drupal es sistema de caché avanzado, replicación de base de datos, balanceo de carga, mecanismos de control de congestión configurable para habilitar o deshabilitar módulos (Maestro, 2015).

Además de brindar la posibilidad de construir un portal web con facilidades para realizar flujos y funcionalidades, CMS Drupal es altamente adaptable a cualquier necesidad del cliente y principalmente por la manera de gestionar de forma individual los contenidos y las presentaciones, además de diferenciar usuarios atendiendo a sus privilegios en la realización de tareas o actividades dentro del sistema para la gestión de la información.

1.4.1. Perfil de instalación

Durante el proceso de instalación de Drupal este ejecuta disímiles tareas o instrucciones que permiten la activación de los módulos, temas, opciones y configuraciones que garantizan el correcto funcionamiento del CMS. Hay determinados módulos y configuraciones que siempre se utilizan y se tienen que instalar una y otra vez, haciendo que el inicio de cada proyecto sea tedioso y repetitivo. Dichas tareas o instrucciones son definidas en una distribución: “perfil de instalación”.

Un perfil de instalación es un conjunto de instrucciones que permitirá definir qué módulos y opciones se quieren activar durante el proceso de instalación de Drupal. De esta forma se pueden crear portales web sin necesidad de empezar de cero, ganando en tiempo y productividad. Los perfiles de instalación permiten crear sitios temáticos. Por ejemplo, se pueden encontrar perfiles de instalación específicos para implementar un periódico digital, una web corporativa, un aula virtual, una tienda online, etc. (Rodríguez, 2012).

Drupal en sus últimas distribuciones trae consigo por defecto de forma personalizada dos perfiles distintos de instalación llamados Estándar y Mínimo como se aprecia en la figura 2.

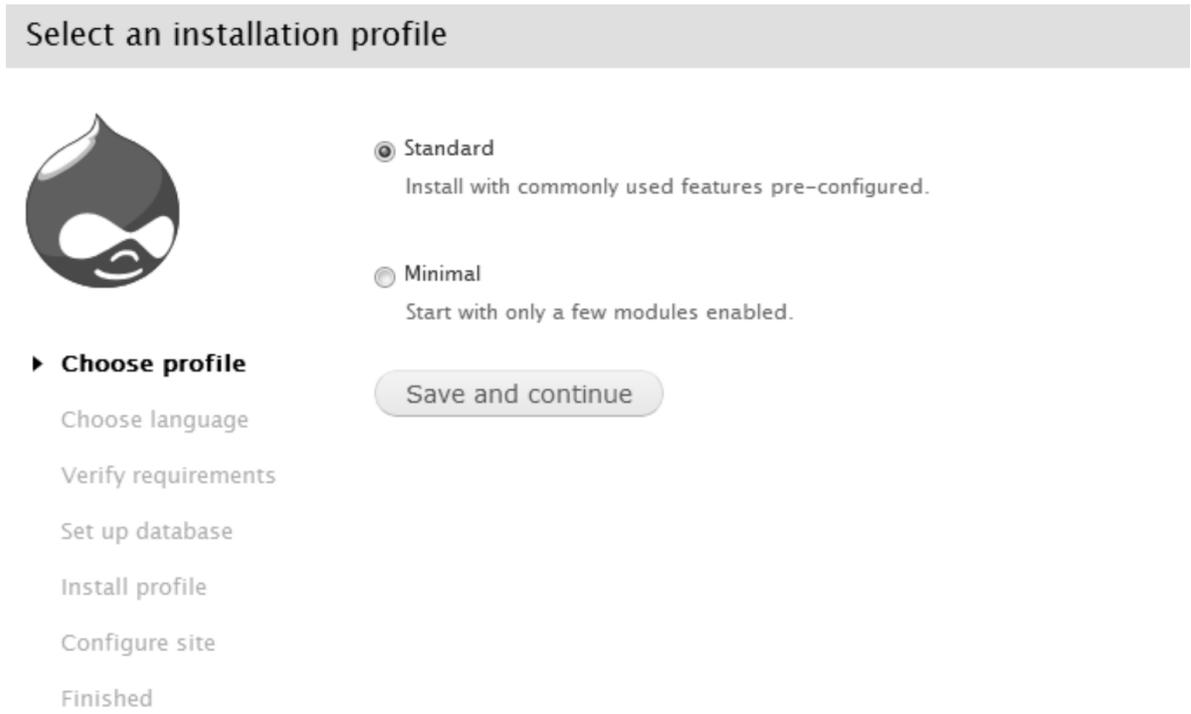


Figura 2: Primer paso de la instalación de Drupal 7. Selección del perfil a instalar (Gil, 2012)

Además, incluye un tercer perfil, llamado *Testing* (Prueba), definido como oculto y no se muestra como seleccionable durante el proceso de instalación de Drupal.

1.4.2. Perfiles de distribución contribuidos

Actualmente en el sitio colaborativo de Drupal se ponen a disposición algunos perfiles que son implementados siguiendo su arquitectura y estándares de codificación. Algunos de estos son:

Corporate site:

Es una distribución que incluye un buen número de módulos útiles para crear un portal corporativo. También incluye algunas “*Features*” (Características) para añadir funcionalidades adicionales.

Open Publish:

Es un perfil de instalación orientado a la creación de un periódico digital, fue diseñado para satisfacer las necesidades de medios como periódicos masivos o sitios de noticias. Implementa 15 módulos propios creados usando el módulo “*features*” con características específicas para el desarrollo de un periódico digital, instala además alrededor de 50 módulos contribuidos permitiendo agregar nuevas funcionalidades al sitio, crea 7 tipos de contenidos y define un sistema de taxonomía, posee temas adaptados con administración de interfaz elegante e íconos informativos.

Commerce Kickstart:

Es un perfil de instalación orientado al comercio electrónico, instala alrededor de 60 módulos adicionales, implementa además alrededor de 20 módulos propios creados usando el módulo “*features*” con características específicas para el perfil, define tareas adicionales durante el proceso de instalación, instala varios tipos de contenidos, instala contenido de ejemplo para visualizar en el sitio y define sistema de taxonomía. Posee temas adaptados con administración de interfaz elegante.

Panopoly:

Es una poderosa distribución de base de Drupal impulsado por un montón de herramientas y paneles. La distribución está diseñada para ser a la vez una base general para la creación de sitios y un marco de base sobre el cual construir otras distribuciones de Drupal.

Open Outreach:

Es una manera rápida y de bajo costo para las organizaciones no lucrativas y las organizaciones de la comunidad para poner en marcha el uso de Drupal con las herramientas de la web que necesitan para la participación pública eficaz.

Open Folio:

Es una distribución para los fotógrafos, pintores, u otros artistas visuales que quieran crear una cartera web de su trabajo. Fue creado con la simplicidad en mente.

Open Deals Drupal Distribution:

Es un perfil de instalación orientado a ofrecer cupones de descuento, instala alrededor de 30 módulos adicionales, implementa además 10 módulos propios creados usando el módulo “*features*” con características específicas para el perfil, instala tipos de contenidos y define sistema de taxonomía. Posee un tema gráfico adaptado para el sitio.

1.4.3 Resultados del estudio realizado a perfiles de instalación

Los perfiles en su proceso de instalación instalan varios módulos y herramientas de Drupal que son de ayuda para el desarrollo de cualquier sitio web, instalan módulos para la visualización de contenido (*Views*, *Panels*), módulos para el soporte de imágenes (*ImageAPI*, *ImageField*, *ImageChace*), módulos para enriquecer los textos (*CKeditor*), entre otros módulos necesitados para un desarrollo web en particular.

Diseñados para el trabajo y la gestión en un área o solución específica, los perfiles estudiados hacen uso de muchas funcionalidades a considerar, tienen características y opciones útiles, como la creación de blogs, calendarios, gestión y visualización de noticias, que sirven como modelo y guía para el desarrollo del perfil de instalación de portales web, son sistemas bien definidos y acoplados, lo que los hace un poco complicados de configurar y adaptar a una solución personalizada.

Se elige comenzar a generar un nuevo perfil de instalación para portales web para SENIT empleando Drupal 7, teniendo en cuenta los estándares, patrones y arquitectura de los sistemas homólogos, dando respuesta a todas las necesidades y deseos del cliente. Para comenzar el desarrollo de este perfil de instalación se identificaron elementos, características y requisitos entre los que destacan:

- ✓ La instalación de módulos de ayuda para los desarrolladores y otros módulos adicionales.
- ✓ Crear tipos de contenidos y roles asociados.
- ✓ Seguir estándares de codificación y arquitectura de Drupal.

Se tiene que tener claro la siguiente estructura de carpetas y archivos necesarios:

- **.info:** se define la información básica de la discreción al igual que hacemos con los módulos y temas.
- **.profile:** se implementan las tareas necesarias que se deben llevar a cabo durante la instalación al igual que en el `.module` o `template.php`.

- **.install:** se utiliza para implementar el hook_intall durante la instalación.
- **/modules:** en esta carpeta se añaden los módulos necesarios para la distribución.
- **/themes:** en esta carpeta se añade el tema o temas que se usan en la instalación.
- **/translation:** en esta carpeta se añaden los idiomas, es decir, los archivos .po que se quieren para la instalación.

Una vez implementados estos archivos, se tiene el perfil listo para ser distribuido e instalado.

1.5. Conclusiones parciales

Como parte del desarrollo de este capítulo se arriban a las siguientes conclusiones parciales:

- ✓ El estudio de los referentes teóricos y metodológicos, así como la aplicación de los métodos y los procesos durante la investigación permitió justificar la necesidad de implementar un perfil de instalación para SENIT que cumpla con las expectativas de los desarrolladores.
- ✓ El estudio de varios perfiles de instalación posibilitó un mejor entendimiento acerca del funcionamiento de estos sistemas, y la identificación de elementos comunes que pueden ser incorporados al que se desea crear.
- ✓ El estudio de las herramientas y tecnologías permitió seleccionar las más adecuadas para el desarrollo del perfil de instalación, partiendo de los requerimientos definidos en el levantamiento de información.

Capítulo 2: Propuesta de solución y características del perfil de instalación de Drupal 7 para el departamento de SENIT

2.1. Introducción

En el presente capítulo se realiza una descripción de la propuesta que se plantea para agilizar el desarrollo de portales web. También el levantamiento de los requisitos funcionales y no funcionales que tiene el perfil de instalación para portales web utilizando Drupal 7, así como su diseño y arquitectura con el objetivo de dar una solución al problema existente.

2.2. Propuesta de solución

Se propone la implementación de un perfil de instalación para el CMS Drupal 7, que integre los servicios necesarios que son utilizados durante el desarrollo de portales web. Todo será el resultado de haber tenido en cuenta las necesidades del departamento SENIT, el objetivo general trazado al inicio de la investigación y los estudios realizados a diferentes perfiles de instalación. Como parte de la solución se propone que al inicio del proceso de instalación de Drupal se muestre la opción de seleccionar un nuevo perfil de instalación, donde Drupal muestra la variante de seleccionar la instalación que se desee realizar como se ilustra en la Figura 3.

Se propone la automatización de los servicios a través de módulos y configuraciones personalizadas para Drupal 7, mediante la automatización de procesos o tareas como: creación de tipos de contenidos, creación y activación de menús y bloques, definición de condiciones o reglas que los servicios deben cumplir. El perfil también instala módulos contribuidos por la comunidad de Drupal, necesarios para facilitar el desarrollo de sitios web y mostrar un funcionamiento más completo en los portales.

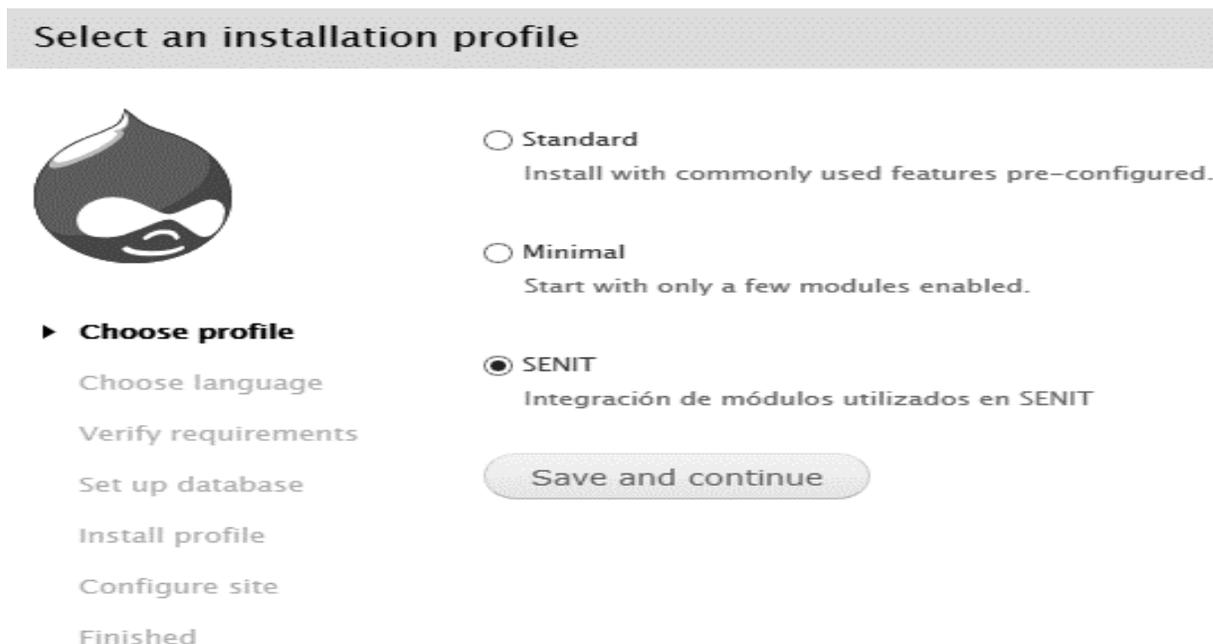


Figura 3: Selección del perfil de instalación a instalar (Gil, 2012).

Estudio de los portales web desarrollados en el centro

Se realizó un estudio de los diferentes portales web desarrollados en SENIT, de los cuales hay algunos en proceso de implementación. Se pudo definir los módulos más repetitivos para la realización de estos portales para utilizar en el perfil de instalación a implementar. Para el análisis de la información, se utilizó la entrevista a uno de los jefes de proyecto del departamento SENIT, logrando con ello identificar las necesidades y requisitos de la misma. Se analizaron diferentes homólogos y antecedentes, a través de los cuales se definieron los contenidos en la propuesta actual.

1. **Barriga Verde:** El sistema surge por la necesidad del medio de prensa de República Dominicana Barriga Verde de contar con un portal web que respondiera a las tendencias y tecnologías actuales de los portales web a nivel mundial. Con dicho sistema se les da solución a problemas tales como la simplificación del proceso de actualización del portal actual y la interacción entre los usuarios de este.

2. **DESEQUIP**: El proyecto “Portal Web de la Empresa DESEQUIP”, comprende la creación de un portal destinado a publicar los contenidos y servicios cuya divulgación en la red de redes sean de interés para esta institución. El desarrollo del portal abarca la implementación de las funcionalidades identificadas, aplicando el diseño que sea aprobado por la empresa y se validará la calidad del sistema, obteniéndose finalmente un portal ajustado a las necesidades de DESEQUIP. El objetivo general es proveer a la empresa DESEQUIP de un portal web de avanzada para la publicación de los contenidos de interés en Internet.
3. **OAHCE**: La Oficina del Asuntos Históricos está en proceso de informatización, debido a ello se solicita una Intranet, a partir de la cual se puedan gestionar los diferentes documentos de interés de la misma, por ejemplo: boletines, revistas, mapas, investigaciones, entre otros. Además, debe brindar diferentes directorios: de personas, telefónico, de correo electrónico. En general el sistema debe facilitar el acceso a la información que manejan los trabajadores de la oficina, agilizando los procesos dentro de la misma. El sistema debe tener un diseño sencillo, agradable y funcional, el cual no requiera de altos conocimientos de informática para lograr una interacción efectiva del usuario.
4. **RED-DEES**: La Red Internacional de Dirección Estratégica en la Educación Superior (RED-DEES), es una red académica constituida por Instituciones de Educación Superior (IES) y redes universitarias de países de diversas áreas geográficas, orientada al desarrollo de variadas temáticas relacionadas con ese perfil. Su misión es contribuir a la promoción, estimulación y fomento de la investigación, del postgrado y de otras actividades académicas sobre temas y experiencias de interés común en el área de la Dirección Estratégica Universitaria (DEU), mediante un mecanismo de diálogo, intercambio y cooperación multilateral entre sus miembros, con el fin de elevar la calidad y pertinencia de la gestión institucional y responder favorablemente a las necesidades de la sociedad.
5. **Intranet-Imeco**: La Empresa Exportadora e Importadora de la Construcción-IMECO cuya misión fundamental es ejecutar operaciones de Comercio Exterior relacionadas con la exportación e importación y comercialización de equipos y bienes de consumo, según Nomenclatura aprobada por

el Ministerio de Comercio Exterior y la Inversión Extranjera, ofrece servicios de Consultoría en gestión de compra-venta en el país y en el exterior. Dicha entidad se ha planteado como necesidad la informatización de una Intranet corporativa con el objetivo de brindar información actualizada sobre la entidad y garantizar la interactividad y rápida interconexión de los usuarios locales.

Resultados obtenidos

De cada uno de estos portales web desarrollados por SENIT se tomaron los elementos comunes para la implementación del sistema a desarrollar concluyendo con los siguientes servicios:

- ✓ Buscadores.
- ✓ Galería de vídeos.
- ✓ Galería de imágenes.
- ✓ Mapa del Sitio.
- ✓ Redes Sociales.
- ✓ Gestión de noticias.
- ✓ Efemérides.
- ✓ Gestión de eventos.
- ✓ Promoción.
- ✓ Gestión de avisos.
- ✓ Estadísticas.
- ✓ Boletines.
- ✓ Enlace de interés.
- ✓ Datos de contactos.
- ✓ Preguntas Frecuentes.
- ✓ Traducción.

2.3. Modelo de dominio

Según *Craig Larman*, un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidades. Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales (término utilizado en la primera edición del libro de Larman), modelo de objetos del dominio y modelos de objetos de análisis (Larman, y otros, 2003).

El modelo conceptual de la problemática a resolver se muestra en la Figura 4.

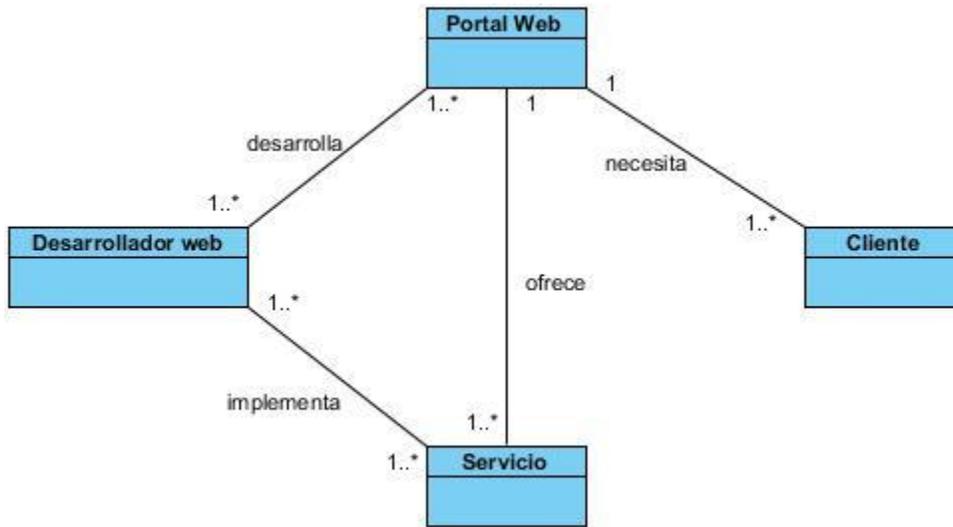


Figura 4: Modelo de dominio.

Clases que conforman el modelo de dominio

Tabla 1. Descripción de las entidades que interactúan.

Entidad	Descripción
<p>Desarrollador web</p>	<p>Los desarrolladores web son los encargados de desarrollar los portales web, enfocados en implementar las características y servicios que ofrece el portal.</p>

Portales web	Los portales web integran servicios que se brindan al usuario final. Es por ello su necesidad debido a que diversas entidades necesitan en varias ocasiones contar con un portal.
Cliente	Forma genérica de enunciar cualquier cliente que solicite la creación de un portal web.
Servicio	Los servicios son ofrecidos en el portal web al usuario, los desarrolladores son los encargados de llevar a cabo su implementación al utilizar los módulos correspondientes.

2.4. Captura de requisitos

La captura de requisitos es una actividad muy importante mediante la cual el equipo de desarrollo extrae de cualquier fuente la información disponible, para poder gestionar las necesidades que debe cubrir dicho sistema de forma estructurada. Proporciona un punto de partida para controlar actividades específicas y evita rechazos finales del usuario debido a que obliga a los mismos a considerar sus requerimientos cuidadosamente. Por la complejidad que todo esto puede implicar, los especialistas en estos temas han trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa (A. Bollati, 2013).

2.4.1. Requisitos funcionales

Los requisitos funcionales son una categoría importante de los requisitos reales, donde se describen lo que el sistema o el software debe hacer. Llamados a veces conductistas o requisitos operacionales porque especifican las entradas (los estímulos) para el sistema, las salidas (respuestas) del sistema, y las relaciones

conductistas entre ellos. Esto se refiere a una colección global de las características de un sistema y las capacidades que hará disponible para los usuarios. Provee un análisis detallado de los datos que el sistema manipulará. Puede incluir una definición detallada de las interfaces de usuario del sistema (Young, 2004).

Lista de reserva del producto

El propósito de la definición de requisitos es especificar las condiciones o capacidades con que el sistema debe contar y las restricciones bajo las cuales debe operar, logrando un acuerdo entre el equipo de desarrollo y el cliente y especificando las necesidades reales de forma que satisfaga sus expectativas. Se dividen en dos grupos: los requisitos funcionales y los requisitos no funcionales (Jacobson, y otros).

A continuación, se presentan los requisitos funcionales que se implementan en el perfil de instalación para portales web utilizando Drupal 7.

- | | |
|---|--------------------------------|
| 1. Adicionar galería de imágenes | 17. Adicionar galería de video |
| 2. Modificar galería de imágenes | 18. Modificar galería de video |
| 3. Eliminar galería de imágenes | 19. Eliminar galería de video |
| 4. Mostrar galería de imágenes. | 20. Mostrar galería de video |
| 5. Mostrar mapa del sitio | 21. Adicionar noticia |
| 6. Mostrar estadísticas de visitas al sitio | 22. Modificar noticia |
| 7. Autenticar usuario | 23. Eliminar noticia |
| 8. Crear usuario | 24. Mostrar noticia |
| 9. Eliminar usuario | 25. Adicionar promoción |
| 10. Modificar usuario | 26. Modificar promoción |
| 11. Mostrar usuario | 27. Eliminar promoción |
| 12. Mostrar efemérides | 28. Mostrar promoción |
| 13. Crear efemérides | 29. Adicionar eventos |
| 14. Eliminar efemérides | 30. Modificar eventos |
| 15. Modificar efemérides | 31. Eliminar eventos |
| 16. Importar efemérides | 32. Mostrar eventos |

- | | |
|---------------------------------|---|
| 33. Adicionar enlace de interés | 47. Modificar avisos |
| 34. Modificar enlace de interés | 48. Eliminar avisos |
| 35. Eliminar enlace de interés | 49. Mostrar avisos |
| 36. Mostrar enlace de interés | 50. Adicionar preguntas frecuentes |
| 37. Adicionar redes sociales | 51. Modificar preguntas frecuentes |
| 38. Modificar redes sociales | 52. Eliminar preguntas frecuentes |
| 39. Eliminar redes sociales | 53. Mostrar preguntas frecuentes |
| 40. Mostrar redes sociales | 54. Adicionar datos de contacto |
| 41. Adicionar boletín | 55. Modificar datos de contacto |
| 42. Modificar boletín | 56. Eliminar datos de contacto |
| 43. Eliminar boletín | 57. Mostrar datos de contacto |
| 44. Mostrar boletín | 58. Traducir contenidos |
| 45. Enviar boletín | 59. Seleccionar idioma (inglés o español) |
| 46. Adicionar avisos | |

Historias de usuarios

En la metodología AUP, las historias de usuarios son las que describen los requisitos del sistema. Se realizan en dependencia de los requisitos funcionales descritos en la Lista de Reserva de Producto (LRP). El contenido que ellas abarcan debe ser concreto y sencillo. El éxito de un proyecto radica en las historias de usuario. Este artefacto sirve como guía para las futuras pruebas de aceptación. Aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas, estas se redactan desde la perspectiva del cliente. Se describen con un lenguaje claro y natural para que puedan ser comprendidas por el resto del equipo que las vaya a utilizar. A continuación, se presentan ejemplos de estas historias de usuario, el resto de las historias de usuarios se encuentran en el Anexo.

Tabla 2. Historia de usuario para el requisito funcional “Crear Usuario”.

Historia de Usuario

Número: HU_8	Nombre Historia de Usuario: Crear Usuario.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Leonar Pina Legón	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1/2
Riesgo en Desarrollo: Gespro/Plan de Riesgo	Puntos Reales:
<p>Descripción: Los usuarios que tengan rol webmaster pueden crear nuevos usuarios en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre de usuario (Obligatorio. Campo de texto. Long min 5, máximo 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9). Debe comenzar con (a-z, A-Z)). • Dirección de correo electrónico (Obligatorio. Campo de texto. Long máximo 254 caracteres. Permite direcciones de correo estructuralmente válidas (Estructura: usuario@subdominios.dominio). • Contraseña (Obligatorio. Campo de texto. Long min 6, máximo 128 caracteres. Permite los caracteres a-z, A-Z, 0-9 y signos de puntuación) • Confirmar contraseña (Obligatorio. Campo de texto. Long min 6, máximo 128 caracteres. Permite los caracteres a-z, A-Z, 0-9 y signos de puntuación) • Estado (Opcional. Campo de selección. Valores (Bloqueado, Activo). Valor por defecto (Activo)). • Roles (Opcional. Campo de selección múltiple. Valores (webmaster, editor)). • Notificar al usuario acerca de su nueva cuenta (Opcional. Campo de selección. Valor por defecto (Desmarcado)). 	
Observaciones:	

1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el usuario.
2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.
3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.
4. Si el usuario ya existe, el sistema emite un mensaje notificando que el usuario ya está en el sistema.

Tabla 3. Historia de usuario para el requisito funcional “Autenticar usuario”.

Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: Autenticar usuario.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Leonar Pina Legón	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1/2
Riesgo en Desarrollo: Gespro/Plan de Riesgo.	Puntos Reales:
<p>Descripción: Los usuarios con rol webmaster o editor que se hayan registrado con anterioridad en el sistema se pueden autenticar, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> •Nombre de usuario (Obligatorio. Campo de texto. Permite todos los caracteres). •Contraseña (Obligatorio. Campo de texto. Permite todos los caracteres). 	

Observaciones: Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha autenticado satisfactoriamente el usuario.

1. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.
2. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.

Tabla 4. Historia de usuario para el requisito funcional “Importar efeméride”.

Historia de Usuario	
Número: HU_16	Nombre Historia de Usuario: Importar efemérides.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Leonar Pina Legón	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1/7
Riesgo en Desarrollo: Gespro/Plan de Riesgo.	Puntos Reales:
Descripción: El sistema permite que los usuarios rol webmaster o editor que accedan al sistema importen la información de los contenidos en formato .scv	
Observaciones: No se aplica	

Tabla 5. Historia de usuario para el requisito funcional “Mostrar Boletín”.

Historia de Usuario	
Número: HU_48	Nombre Historia de Usuario: Mostrar Boletín.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Leonar Pina Legón	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1/7
Riesgo en Desarrollo: Gespro/Plan de Riesgo.	Puntos Reales:
Descripción: Los usuarios que tengan rol webmaster o editor pueden visualizar los boletines existentes en el sistema. Los usuarios anónimos pueden visualizarlos en el sistema.	
Observaciones: No se aplica	

2.4.2. Requisitos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (Scribd, 2015).

A continuación, se presentan las características y propiedades que el perfil de instalación debe tener para un mejor funcionamiento. Entre ellos se encuentran los siguientes:

Requisitos para las restricciones del diseño

- El perfil de instalación tiene que ser usado con Drupal 7, como Sistema Gestor de Contenidos (CMS). El perfil tiene que ser desarrollado o configurado con PHP como lenguaje de programación del lado del servidor, en su versión 5.5.9.

Requisitos de usabilidad

- El perfil de instalación puede ser usado de forma fácil por cualquier desarrollador de portales, ya que hace uso del mismo tema de administración que trae Drupal por defecto, siguiendo de esta forma un diseño bastante usable y legible.

Requisitos legales

- Al igual que las distintas distribuciones de Drupal, el perfil se distribuye como software libre bajo licencia GNU GPL (General Public License).

Requisitos de mantenimiento

- Se debe mejorar y actualizar el perfil frecuentemente e ir incorporando servicios en caso de que sea necesario.

Requisitos de hardware**✓ Mínimos**

- Se debe garantizar para el desarrollo del proyecto por parte del servidor una memoria RAM de 1 GB y por parte del cliente una RAM de 512 MB.

✓ Recomendados

- Se recomienda para el servidor una memoria RAM de 2 GB o más y para las computadoras clientes una RAM de 1 GB en adelante.

2.5. Patrones de diseño

Brad Appleton define un patrón de diseño de la siguiente manera: “Un patrón es una semilla de conocimiento, la cual tiene un nombre y transporta la esencia de una solución probada a un problema recurrente dentro de cierto contexto en medio de intereses en competencia” (Appleton, 2014). Dicho de otro modo, un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico y en medio de “fuerzas” que pueden tener un impacto en la manera sencilla en que se aplica y utiliza el patrón.

A pesar de que Drupal no hace uso completo de las características de programación orientada a objetos (POO) nativos de PHP, el código base de Drupal y la interfaz de programación de aplicaciones (API) reflejan algunos principios que se encuentran en la programación y diseño orientado a objetos por lo que el CMS puede ser evaluado teniendo en cuenta principios de diseño orientado a objetos (Marimón, 2012).

Este sistema se encuentra desarrollado en la plataforma Drupal por lo que su propuesta de solución responde a dicho diseño, además de seguir la misma concepción de la POO. Por la tanto se pueden identificar los siguientes patrones de diseño presentes en el perfil de instalación para portales web.

***Singleton* (Instancia única)**

El patrón *Singleton* garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia (Microsoft, 2012).

El funcionamiento de este patrón es muy sencillo y podría reducirse a los siguientes conceptos:

1. Ocultar el constructor de la clase *Singleton*, para que los clientes no puedan crear instancias.
2. Declarar en la clase *Singleton* una variable miembro privada que contenga la referencia a la instancia única que se quiere gestionar.
3. Proveer en la clase *Singleton* una función o propiedad que brinde acceso a la única instancia gestionada por el *Singleton*. Los clientes acceden a la instancia a través de esta función o propiedad.

En Drupal si los módulos se toman como objetos, se sigue el patrón *singleton*. En general, estos objetos no encapsulan datos; lo que separa a un módulo de otro es el conjunto de funciones que contiene, por lo que debe ser considerado como una clase con una instancia *singleton*. Se puede representar a los módulos creados a partir del módulo “*Features*”, como el conjunto de funciones encargadas de la creación de algunos de los tipos de contenidos y la manipulación de los datos introducidos almacenados en la base de datos, realizando la interacción con estos a través de la misma instancia del módulo.

Decorator (Decorador)

Permite añadir responsabilidades extra que se pueden añadir y quitar a objetos concretos de manera dinámica y transparente, esto es, sin afectar a otros objetos. Además, proporciona una alternativa flexible a la herencia para extender funcionalidad. Se aplica cuando la herencia sea impracticable, porque implique crear múltiples subclases para todas las combinaciones posibles (Guerra Sánchez, 2008).

La evidencia de este patrón se encuentra en la definición de *hooks* por parte de los módulos del núcleo de Drupal y otros módulos contribuidos, que posibilitan que otros puedan extender el comportamiento de dichas funciones. Los *hooks* son funciones que permiten la comunicación con todo el núcleo de Drupal, pueden ser eventos u operaciones con campos. De esta forma se brinda la flexibilidad de que nuevos módulos puedan modificar el comportamiento del núcleo en cuanto al tratamiento de los datos y en cada uno de los eventos del sistema. En el desarrollo de la solución propuesta se emplearon *hooks* como el *hook_node_load ()* y *hook_node_save ()*.

Observer (Observador)

Define una dependencia de uno a muchos entre objetos de forma que, cuando un objeto cambia de estado, se notifica a los objetos dependientes para que se actualicen automáticamente. Se usa cuando una abstracción tiene dos aspectos, y uno depende del otro. Encapsular los aspectos en objetos distintos permite cambiarlos y reutilizarlos. Además, cuando cambiar un objeto implica cambiar otros, pero no sabemos exactamente cuántos hay que cambiar y cuándo un objeto debe ser capaz de notificar algo a otros sin hacer suposiciones sobre quiénes son dichos objetos. Esto es, cuando se quiere bajo acoplamiento (Guerra Sánchez, 2008).

Este patrón es un fenómeno generalizado en todo Drupal. Cuando se realiza una modificación a un vocabulario en el sistema de taxonomía de Drupal, el gancho `hook_taxonomy_vocabulary_update ()` se llama en todos los módulos que lo implementan. Al implementar el hook, los módulos se registran como observadores del objeto vocabulario.

Bridge (Puente)

Este patrón se usa para desacoplar una abstracción de su implementación. Así mismo la herencia liga una implementación a una abstracción dificultando la modificación, extensión y reutilización (Puigserver, 2008).

Aplicabilidad:

- Si se quiere evitar un enlace permanente entre abstracción e implementador (Permite cambiar la implementación en tiempo de ejecución.)
- Si se quiere evitar la proliferación de clases heredadas como se ha visto en la motivación, es indicativo de la necesidad de separar abstracción/implementación.
- Si se quiere compartir una misma implementación entre varios clientes.

En Drupal la capa de abstracción de la base de datos es implementada de una manera similar, imitando el patrón *Bridge*, no realizando llamadas directas a la base de datos, sino que se hacen a través de funciones genéricas definidas por la capa de abstracción, funcionando ésta como puente. Estas funciones serán siempre iguales, independientemente del gestor de bases de datos utilizado. En el perfil puede apreciarse este patrón también en la abstracción de los módulos con respecto al diseño, pues puede cambiarse la estructura del perfil o los tipos de contenido sin afectar al módulo.

La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. Los módulos necesitan ser programados independientes del gestor de bases de datos que se esté usando. Se utiliza en el hook `simplenews_node_insert ()` que permite conectarse a la base de datos y realizar consultas abstrayéndose del Sistema Gestor de Base Datos que se emplee para manejar los datos almacenados.

Chain of Responsibility (Cadena de responsabilidades)

Evita que el receptor se apodere por completo de la petición, y da oportunidades a otros receptores para contestar la petición.

Se usa dicho patrón cuando:

- Hay más de un objeto que pueden manejar una petición, y el manejador no se conoce a prioridad, sino que debería determinarse automáticamente.
- Se quiere enviar una petición a un objeto entre varios sin especificar explícitamente el receptor.

El conjunto de objetos que pueden tratar una petición debería ser especificado dinámicamente (Hernández, 2008).

El sistema de menús de Drupal sigue el patrón *Chain of Responsibility*. En cada solicitud de página, el sistema de menú determina si hay un módulo para manejar la petición, si el usuario tiene acceso al recurso solicitado, y la función será llamada para hacer el trabajo. Para hacer esto, un mensaje se pasa al elemento de menú correspondiente a la trayectoria de la solicitud. Si el elemento de menú no puede atender la solicitud, que se transmite por la cadena, este continúa hasta que un módulo controla la solicitud, un módulo niega el acceso al usuario, o la cadena está agotada. Todas las peticiones de rutas definidas por los módulos creados evidencian este patrón. Un ejemplo en el perfil puede verse cuando se selecciona el vínculo "Buscar". El sistema detecta el módulo de búsqueda que puede encargarse de la petición, verifica si éste tiene permisos para usar esta función y entonces permite acceder a su contenido.

Command (Comando)

Intención del patrón:

- Encapsular una solicitud como un objeto, lo que permite parametrizar clientes con diferentes solicitudes, cola o registro de solicitudes, y soportar operaciones que se pueden deshacer.
- Promover la "invocación de un método en un objeto" a ser un objeto "completo" (Barrionuevo, 2008).

Muchos de los hook de Drupal utilizan el patrón de comandos para reducir el número de funciones que son necesarias para poner en práctica, pasando la operación como un parámetro junto con los argumentos. De

hecho, el hook system en sí utiliza este patrón, por lo que los módulos no tienen que definir cada hook, sino sólo los que cuidan a implementar. En el desarrollo de la solución propuesta se evidencia este patrón ya que los módulos creados no tienen que definir cada hook, sino los que se vayan a aplicar.

2.6. Arquitectura del perfil de instalación

Al igual que con los patrones de diseño, el perfil de instalación debido a que está hecho para Drupal, se ajusta también a su estructura, la arquitectura que emplea el perfil hereda de este CMS, siguiendo de esta forma un modelo de programación estructurada y una arquitectura modular.

Drupal no considera los sitios web como un conjunto de páginas interrelacionadas entre sí, sino que estructura el contenido en una serie de elementos básicos como nodos, módulos, bloques, menús, permisos de usuario y plantillas.

Para generar una página, los módulos de Drupal obtienen todos los datos necesarios de una base de datos, se les aplica un tema visual, y se envía de vuelta al navegador del cliente.

Arquitectura modular

La lógica de Drupal está programada en PHP, siguiendo un modelo de programación estructurada, y que hace uso de un sistema de bases de datos relacional. El núcleo del sistema se complementa con algunos módulos obligatorios y otros opcionales, que vienen dentro de la distribución de Drupal. El sistema incluye un conjunto de herramientas y pautas que se deben seguir para desarrollar e integrar nuevas funcionalidades a través de módulos adicionales.

El sistema Drupal se encarga de ejecutar los módulos solo cuando se necesitan, y para ello es necesario que cada módulo tenga un medio de comunicarle qué hace y cuándo debe hacerse. Esto se realiza mediante un conjunto de funciones llamadas hooks o ganchos. Cuando Drupal atiende una determinada solicitud, examina los módulos activados en el sitio buscando funciones cuyos nombres se ajustan a determinados hooks. Los hooks van a permitir incorporar simples y complejas funcionalidades al sistema Drupal.

Además de los módulos, Drupal está integrado también por temas. Un tema es un conjunto de archivos que

permiten cambiar el aspecto gráfico del sitio, estos archivos definen la capa de presentación del sitio. Con los temas, Drupal separa los contenidos de la presentación o diseño gráfico con que se muestran. Cada elemento es tratado de forma independiente, aplicándole su propia plantilla.

En Drupal, un nodo, un bloque o un mensaje del foro son tratados por separado a la hora de devolver el HTML resultante. La función *theme* () es la encargada de aplicar las plantillas para cada uno de estos elementos, devolviendo el código HTML correspondiente. Por último, el sistema completa la salida con cada HTML individual, dando como resultado la página que se está cargando en cada momento.

Tanto módulos como temas pueden implementar un conjunto de funciones PHP que permiten intervenir en el sistema y modificar la salida final HTML. Cuando se desea mostrar un elemento por pantalla, el sistema de temas entra en juego, procesando el elemento y devolviendo una salida HTML final, que será la que se muestre en el navegador.

El perfil de instalación de portales es, por tanto, un sistema con una arquitectura modular que permite ampliar sus funcionalidades a través de métodos de desarrollo e integración de nuevos módulos, consiguiendo de esta forma incorporar nuevas características funcionales a un portal.

En la Figura 5 obtenida del libro de Rodríguez, F. G. (2012) se muestran de forma esquemática los elementos que conforman un sistema Drupal.

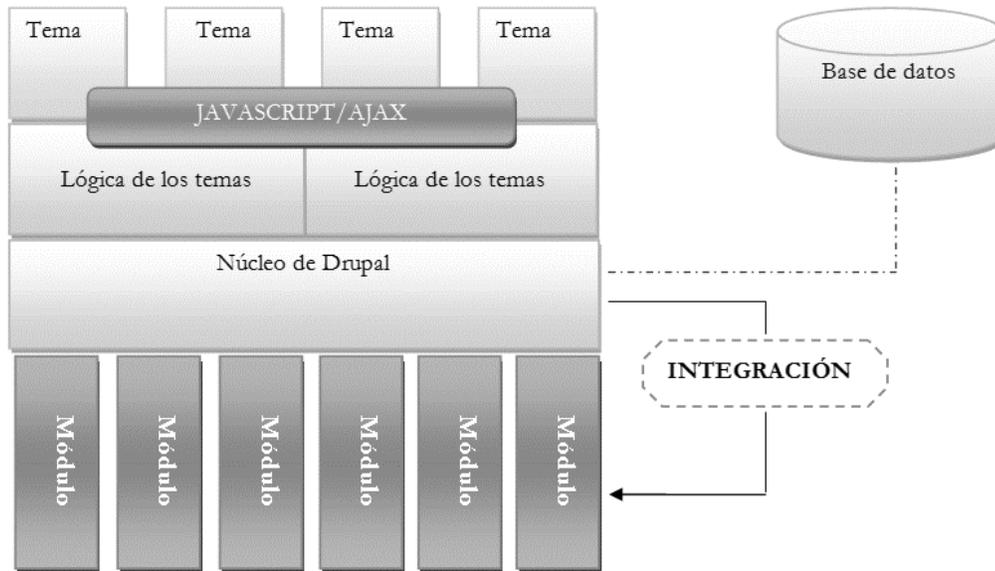


Figura 5: Arquitectura de Drupal (Drupal, 2013).

Por consiguiente, el perfil de instalación para portales web utilizando Drupal 7 es un sistema con arquitectura modular siguiendo un modelo de programación estructurada, ya que incluye un conjunto de herramientas y pautas a seguir para desarrollar e integrar nuevas funcionalidades a través de módulos adicionales que a su vez permiten incorporar nuevas características funcionales al perfil.

2.7. Conclusiones parciales

Como parte del desarrollo de este capítulo se arriban las siguientes conclusiones parciales:

- ✓ El modelo conceptual permitió de forma precisa cómo se relacionan las clases que conforman el modelo del dominio.
- ✓ La lista de reserva del producto obtenida a partir del proceso de identificación de los requisitos, sirvió de guía para desarrollar las distintas funcionalidades de la aplicación.
- ✓ El desarrollo de la arquitectura modular y los patrones de diseño con que cuenta Drupal brindó la

flexibilidad y adaptabilidad presentes en el sistema ante posibles cambios en la implementación de las funcionalidades.

Capítulo 3: Implementación y prueba del perfil de instalación de Drupal 7 para el departamento de SENIT

3.1. Introducción

En el siguiente capítulo se presentará el diagrama de componentes y la definición de los estándares de codificación empleados durante la implementación del perfil de instalación, por último, se especificarán y ejecutarán las pruebas para validar la solución propuesta y su correcto funcionamiento.

3.2. Diagrama de componentes

Los diagramas de componentes son utilizados para estructurar el modelo de la implementación. Permiten modelar una vista estática del sistema, muestran la organización y las dependencias lógicas entre un conjunto de componentes del *software*, que pueden ser librerías, binarios, ejecutables y códigos fuentes (Jacobson, y otros, 2011) .

A continuación, se describen los componentes de la solución propuesta, así como su interacción con otros componentes:

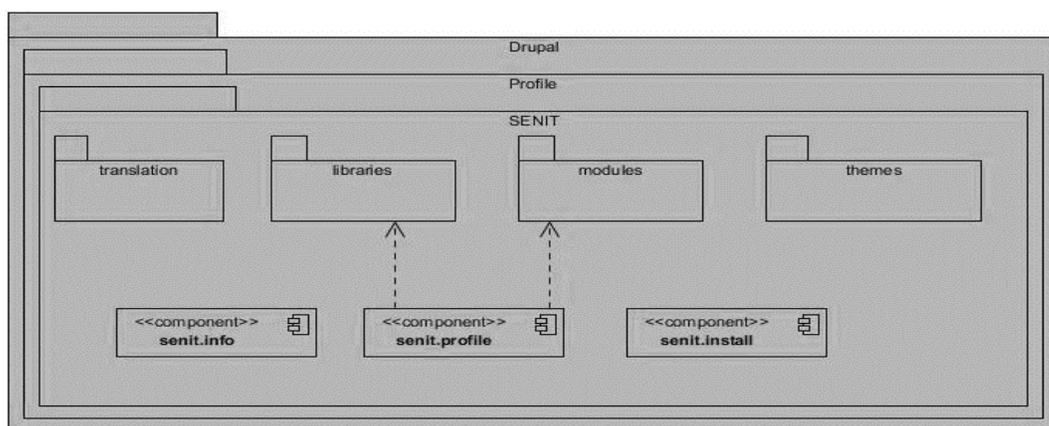


Figura 5. Diagrama de componentes.

Seguidamente se presenta una descripción de los archivos y carpetas que componen el perfil de instalación.

senit.info: es el archivo donde se define el perfil de instalación. El contenido es similar al archivo .info de módulos y temas, además de contener información básica sobre el perfil (nombre, descripción, versión de Drupal, etc.), así como dependencias con otros módulos.

senit.profile: es el archivo donde se implementan las tareas que se llevarán a cabo durante la instalación. Es equivalente al archivo “.module” de los módulos, y en él se pueden usar la mayoría de funciones disponibles de la API de Drupal.

senit.install: permite implementar **hook_install ()** para insertar contenido en la base de datos durante la instalación. Permite activar módulos propios del perfil de instalación para portales web que estarán disponibles en el sitio después de la instalación.

translations: en esta carpeta se añaden las traducciones del núcleo, necesarias para que el proceso de instalación se realice en otro idioma (por ejemplo en español).

libraries: es una carpeta donde se ubican las librerías a usar, que son necesarias para extender las funcionalidades en el sitio.

modules: en esta carpeta se añaden los módulos que estarán disponibles durante la instalación. Generalmente sólo se usa esta carpeta para añadir módulos creados específicamente para la distribución. El resto de módulos requeridos serán registrados como dependencias, de forma que se soliciten durante la instalación.

themes: en esta carpeta se pueden añadir temas que estarán disponibles durante la instalación. Generalmente sólo se usa esta carpeta para añadir temas creados específicamente para la distribución.

3.3. Estándares de codificación

Drupal cuenta con estándares de codificación que permiten lograr una implementación limpia y organizada siendo esta entendible para cualquier desarrollador que utilice el código desarrollado (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 la indentación debe ser de 2 espacios.

Etiquetas de apertura y cierre de PHP: el código PHP se debe escribir siempre utilizando las etiquetas `<?php y?>` y en ningún caso la versión corta `<? y?>`.

```
<?php print $user_picture; ?>
```

Figura 6: Etiqueta 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. Los operadores unarios como ++, -- no deben tener separación.

```
$meses = $i+1;
```

Figura 7: Operadores

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. También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple.

```
echo '<pre>';  
print_r($node->field_categoria['und'][0]['taxonomy_term']->name);  
echo '</pre>';
```

Figura 8: Uso de comilla

Estructuras de control: con respecto a las estructuras de control, se debe tener en cuenta las siguientes normas:

- ✓ Debe haber un espacio entre el comando que define la estructura (if, while, for) y el paréntesis de apertura permitiendo no confundir las estructuras de control con la nomenclatura de las funciones.

- ✓ La llave de apertura ({) se sitúa 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.
- ✓ Las estructuras else y elseif se deben escribir en la línea siguiente al cierre de la sentencia anterior.

```
for($i=0; $i<12;$i++){
  if($meses[$i+1] == $mes_letra){
    $meses = $i+1;
  }
}
```

Figura 9: Estructura 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.

```
function efemerides_uninstall()
```

Figura 10: Funciones

Arreglos: los valores dentro de un arreglo 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 arreglo 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 debe escribir, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```
121 |
122 | //cargando archivos js y css
123 | //drupal_add_js(drupal_get_path('module', 'efemerides').'/efemerides.js');
124 | //drupal_add_css(drupal_get_path('module', 'efemerides').'/efemerides.css');
125 |
126 | }
127 |
128 |
129 |
130 | /**
131 |  * Implements hook_theme()
132 |  * aplicandole el tpl del nodo al node efemerides
133 |  */
134 | function efemerides_theme($existing, $type, $theme, $path) {
135 |
136 |     $theme = array();
137 |     $theme['node__efemerides'] = array(
138 |         'render element' => 'content',
139 |         'base hook' => 'node',
140 |         'template' => 'node--efemerides',
141 |         'path' => drupal_get_path('module', 'efemerides'),
142 |     );
143 |
144 |
145 |     return $theme;
146 | }
147 |
148 |
149 | /**
150 |  * Implements hook_node_presave()
```

Figura 11: Arreglos

Constantes: los nombres de las constantes deben escribirse en mayúsculas, con guiones bajos para separar palabras. Al igual que ocurre con las funciones, los nombres de las constantes deben tener como prefijo el nombre del módulo (o tema) en el que se utilizan, para evitar errores de duplicidad de constantes. Este prefijo también se debe escribir en mayúsculas.

Variables globales: estas se deben declarar utilizando un guion bajo inicial, seguido del nombre del módulo o tema y otro guion bajo antes del nombre de la variable.

Nombres de archivos: los nombres de archivos deben escribirse siempre en minúsculas. La única excepción son los archivos de documentación, que tienen extensión .txt y el nombre es en mayúsculas.

URLs de ejemplo: por convención, siempre que haya que indicar una URL de ejemplo, se debe utilizar "example.com".

3.4. Pruebas de software

Las pruebas de *software* son un elemento crítico para la garantía de calidad del producto y representa una revisión final de las especificaciones, del diseño y de la codificación (S. Pressman, 2012).

Las pruebas de *software* son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón, se debe definir en el proceso de la ingeniería del *software* una plantilla para las pruebas del *software*: un conjunto de pasos en los que se puede situar los métodos específicos de diseño de casos de prueba. Las pruebas constituyen el último bastión desde el que se puede evaluar la calidad y de forma más pragmática descubrir los errores. La prueba del *software* es un elemento de un tema más amplio que, a menudo, es conocido como verificación y validación (S. Pressman, 2012).

3.4.1. Pruebas funcionales

Las pruebas funcionales tienen como objetivo asegurar que el *software* desarrollado cumpla con las especificaciones requeridas y eliminar los posibles defectos que este pudiera tener. Para la realización de estas pruebas se utilizó el método de caja negra las cuales son realizadas desde el punto de vista de interfaz de usuario, lo cual permite valorar el funcionamiento de la aplicación según la interacción del usuario con el portal.

CAPÍTULO III

Propuesta de solución y características del perfil de instalación de Drupal 7 para el departamento de SENIT

A continuación, se muestra el diseño de casos de pruebas correspondiente a la funcionalidad Autenticar Usuario. Los casos de pruebas restantes se pueden consultar en los anexos.

Condiciones de ejecución

El usuario debe estar autenticado en el sistema.

Tabla 6. Caso de prueba. Escenario Autenticar usuario

Escenario	Descripción	Nombre de usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario de forma correcta.	El sistema autentica al usuario de forma correcta.	V	V	El sistema autentica al usuario y le permite el acceso a las funcionalidades del sistema, según su rol.	El usuario accede a la página principal URL: URL/user. 1.- El sistema habilita un formulario para introducir la información. 2.-El usuario introduce la información y
		Editor	Editor		
		I	V		

CAPÍTULO III

Propuesta de solución y características del perfil de instalación de Drupal 7 para el departamento de SENIT

<p>EC1.2 Error al autenticarse un usuario.</p>	<p>El usuario inserta datos incorrectos por lo que el sistema no puede autenticarlo.</p>	<p>5458*-*</p>	<p>Editor</p>	<p>El sistema no autentica al usuario y emite el siguiente mensaje de error: “Lo sentimos. No reconocemos el nombre de usuario o la contraseña. ¿Olvidó su contraseña?”.</p>	<p>presiona el botón: Iniciar sesión.</p>
<p>EC 1.3 Dejar campos vacíos al autenticarse un usuario.</p>	<p>El usuario deja algunos o ambos, de los campos vacíos, por lo que el sistema no puede autenticarlo.</p>	<p>V</p>	<p>I</p>	<p>El sistema no autentica al usuario y emite el siguiente mensaje de error: “El campo Contraseña es obligatorio.”</p>	
		<p>Editor</p>	<p>vacío</p>		
		<p>I</p>	<p>V</p>	<p>El sistema no autentica al usuario y emite el siguiente mensaje de error: “El campo Nombre de usuario es obligatorio.”</p>	
		<p>Vacío</p>	<p>editor</p>		
<p>I</p>	<p>I</p>				

CAPÍTULO III

Propuesta de solución y características del perfil de instalación de Drupal 7 para el departamento de SENIT

		Vacío	vacío	El sistema no autentica al usuario y emite el siguiente mensaje de error: “El campo Nombre de usuario es obligatorio. El campo Contraseña es obligatorio.”	
EC 1.4 Cerrar sesión.	El usuario cierra sesión.	NA	NA	El sistema muestra la página principal del portal sin los privilegios del usuario autenticado.	Seleccionar la opción Cerrar sesión en la parte superior derecha del menú.

Descripción de las variables

Tabla 7. Descripción de las variables del caso de prueba Autenticar Usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre de usuario	campo de texto	No	Campo de longitud de hasta 60 caracteres, Se permiten caracteres a-z, A-Z, 0-9, espacios en blanco pero no uno al lado del otro. No se permiten signos de puntuación excepto los puntos, guiones, comillas

CAPÍTULO III

Propuesta de solución y características del perfil de
instalación de Drupal 7 para el departamento de SENIT

				simples (apóstrofes) y guiones bajos. No deben existir dos signos de puntuación consecutivos.
2	Contraseña	campo de texto	No	Admite todos los caracteres. Debe tener como mínimo 8 caracteres.

Resultados de las pruebas funcionales

Las pruebas funcionales aplicadas a la propuesta de solución fueron basadas en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional especificado. Para la realización de las pruebas se ejecutó cada caso de prueba usando datos válidos e inválidos, y se eligieron correctamente los valores de entrada abarcando el mayor número posible de combinaciones, sin que fuera muy elevado el número de casos de pruebas.

Durante la validación del perfil de instalación se realizaron pruebas de caja negra en 3 iteraciones, se obtuvieron los siguientes resultados:

- En la primera iteración se obtuvieron 15 no conformidades. Entre estas se encuentran la ausencia de campos en algunos tipos de contenidos creados, por ejemplo de texto y algunos formatos de presentación de contenidos erróneos. Estas no conformidades fueron corregidas.
- En la segunda iteración se detectaron 3 no conformidades. Entre estas se encuentran: Incorrecta validación para la entrada de datos incorrectos en algunos campos de entrada de datos. Dichas no conformidades también fueron rectificadas.
- En una tercera iteración el sistema estaba libre de no conformidades, cumpliendo con los requisitos funcionales especificados.

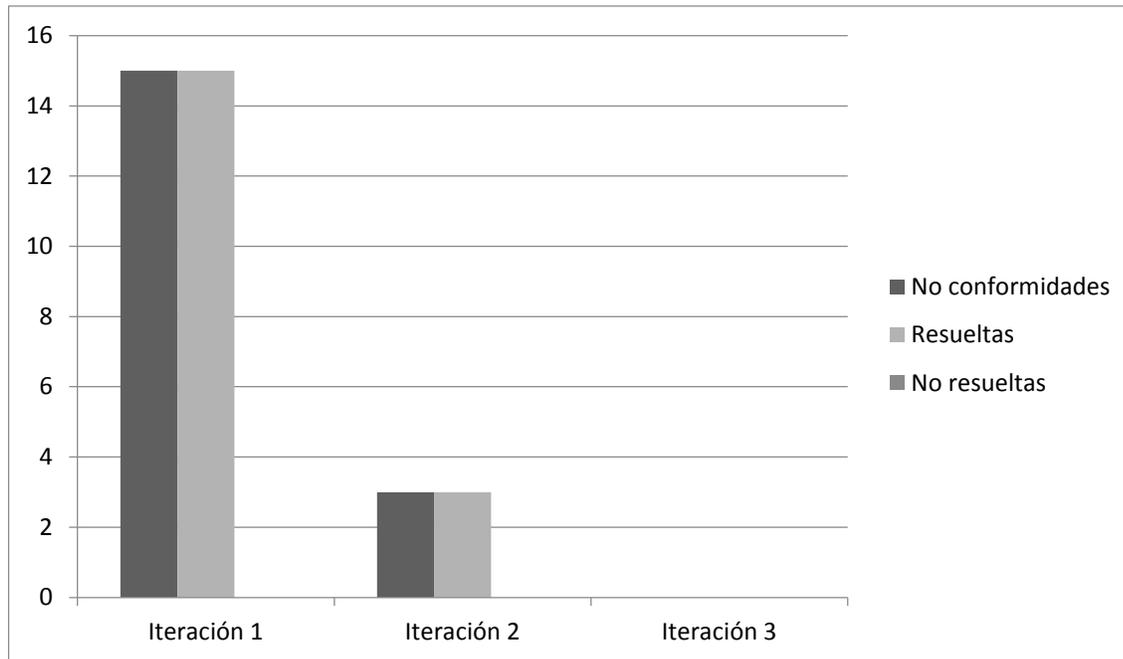


Figura 7: Resultados de las pruebas funcionales

A continuación, se muestran los tipos de errores encontrados con los porcentajes que representan.

Errores de idioma (35%), Interfaz de usuario (20%), Validación (15%), Ortografía (15%), Funcionalidad (10%), Opciones que no funcionan (5%).

3.4.2. 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 (vyyquality, 2013).

Resultados de las pruebas de seguridad

Para garantizar la seguridad del perfil de instalación después de instalada se realizan pruebas de seguridad mediante una lista de chequeo aplicable fundamentalmente a portales y aplicaciones web.

El objetivo general de esta lista es evaluar a través de indicadores la seguridad de las aplicaciones en un primer nivel (nivel 1), establecido por los especialistas del grupo de seguridad del Departamento de Pruebas de Software (DEPSW) de la UCI.

Luego de analizar el perfil de instalación después de ser instalada a través de las listas de chequeo, se detectaron inconvenientes asociados a los siguientes indicadores:

- ✓ Al copiar la URL del portal después de estar autenticado, cerrar el navegador y volver a abrirlo, al pegar la dirección copiada anteriormente nuevamente en el navegador, el usuario puede acceder al portal nuevamente.
- ✓ Al cerrar la sesión de un usuario y dar clic en el botón del navegador “Atrás” la aplicación vuelve entrar a la sesión autenticada.
- ✓ El campo usuario de la autenticación al sistema tiene el auto completamiento activado (guarda los usuarios que se autentican). Para ver esto se debe autenticar al sistema, luego salir y poner la primera letra del usuario autenticado para ver si muestra el nombre de usuario completo.

Se realizaron otras pruebas de seguridad haciendo uso de la herramienta Websecurify en su versión 0.8, la cual es un entorno integrado para pruebas de seguridad web, y se usan para mitigar riesgos relacionados con la existencia de vulnerabilidades en aplicaciones web, estas pruebas fueron realizadas en dos iteraciones.

En la primera iteración se detectaron a través de la herramienta empleada un total de 1 no conformidad que comprometen la seguridad de la aplicación, quedando evaluada de insatisfactoria la prueba. En la segunda iteración la prueba fue evaluada de satisfactoria, debido a que no se encontraron no conformidades que afecten la seguridad de la aplicación. A continuación, se muestran las vulnerabilidades identificadas con el uso de la herramienta seleccionada.

Tabla 8: Pruebas de seguridad. (Websecurify)

Vulnerabilidad	Descripción	Solución
Autocompletar Activado	El autocompletar siempre debe estar desactivado (autocomplete = "off"), especialmente en las formas que procesan datos sensibles, tales como formularios con campos de contraseña, ya que un atacante, si es capaz de acceder a la caché del navegador, podría obtener fácilmente la información almacenada en caché en texto plano.	Desactivar la función de autocompletar (autocomplete = "off") en formas que pueden contener datos sensibles.

3.5. Conclusiones parciales

Como parte del desarrollo de este capítulo se arribaron las siguientes conclusiones:

- ✓ La utilización de los estándares de codificación con que cuenta Drupal, permitió la obtención del código del sistema de manera limpia y organizada, siendo esta entendible para cualquier programador que utilice el código desarrollado.
- ✓ Las pruebas realizadas al perfil permitieron detectar los errores existentes para poder solucionarlos y propiciar un mejor funcionamiento del mismo.

Conclusiones Generales

El desarrollo de esta investigación ha posibilitado el cumplimiento de los objetivos y tareas propuestas, por lo que se arriban a las siguientes conclusiones

- ✓ La revisión bibliográfica y el estudio de sistemas homólogos permitió identificar las principales funcionalidades que debe contener la aplicación.
- ✓ La descripción de los requisitos funcionales y no funcionales aceleró la implementación del sistema, debido a que posibilitó la selección de módulos contribuidos como apoyo al desarrollo basándose en las características y requerimientos de la aplicación.
- ✓ El análisis y la selección de las tecnologías adecuadas posibilitó que se implementara un sistema con una gestión fácil de datos y contenidos, capaz de satisfacer las necesidades del cliente.
- ✓ Se demuestra la factibilidad del desarrollo de la solución asegurando un funcionamiento completo y una mejor productividad en los procesos de ingeniería, gestión y desarrollo de sistemas.
- ✓ Las pruebas de software efectuadas a la aplicación permitieron identificar y corregir las no conformidades garantizando así la calidad del portal web desarrollado después de instalado el perfil.

Recomendaciones

- ✓ El análisis y la selección de las tecnologías adecuadas permitió que se implementara un sistema con una gestión fácil de datos y contenidos, capaz de satisfacer las necesidades del cliente.
- ✓ Se recomienda descargar la versión más reciente de los módulos usados en el perfil en el momento de realizar la instalación o después, para asegurarse que se están instalando las últimas versiones de dichos módulos.

Referencias

A. Bollati, Verónica. 2013. Proceso Unificado: Captura de Requisitos. [En línea] 2013. http://www.kybele.etsii.urjc.es/docencia/IS_LADE/2011-2012/Material/%5BIS-LADE-2011-12%5DT13%20-%20PU%20-%20Captura%20de%20requisitos.pdf.

Alegsa, Leandro. 2010. DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. [En línea] 12 de Mayo de 2010. <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php>.

Appleton, Brad. 2014. Everything you never wanted to know. [En línea] 10 de junio de 2014. <http://www.bradapp.com/docs/rec/stretching/>.

Bakken, S. S. and AULBACH. 2001. Manual de PHP. Grupo de documentación de PHP. [En línea] 2001. www.php.net.

Barrionuevo, Juan. 2008. Programación sólida: Patrones de diseño de comportamiento. [En línea] 2008. http://programacionsolida.com.ar/2012/04/patrones-de-diseno-de-comportamiento_23.html.

Barzanalla, Rafael. 2011. Informática Aplicada a la Gestión Pública. [En línea] 13 de octubre de 2011. <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>.

Carrillo Pérez, Isaías y Pérez González, Rodrigo. 2008. *Metodología de desarrollo de Software*. 2008.

Casas Cervero, Joan. 2011. *Content Management System*. Informática, Educación Superior. Barcelona : s.n., 2011.

Clikea. 2015. Qué es CMMI y para qué sirve. [En línea] 2015. <http://www.cantabriatic.com/que-es-cmmi-y-para-que-sirve>.

Definición ABC. 2007. Concepto en Definición ABC. [En línea] 2007. <http://www.definicionabc.com/tecnologia/portal.php>.

Díaz González, María Socorro. 2011. El gestor de contenidos en la Sociedad de la información. [En línea] 2011. <http://gredos.usal.es/jspui/handle/10366/82619>.

Drupal. 2013. Tutorial Drupal. [En línea] 2013. <http://www.cursosdrupal.com/content/arquitectura>.

BIBLIOGRAFÍA REFERENCIADA |

Flores, Lic. Ervin. 2014. METODOLOGIAS AGILES AUP. [En línea] 2014.
http://www.academia.edu/7894130/METODOLOGIAS_AGILES_AUP.

García Gómez, Juan Carlos. 2013. *Portales de internet: concepto, tipología básica y desarrollo.* 2013.

Gil, Fran. 2012 . *Curso de creación y gestión de portales web en Drupal 7. Nivel Avanzado.* 2012 .

—. **2012.** *Experto en Drupla 7.Nivel inicial.* s.l. : Forcontu, 2012.

Giugni O., Marylin. 2008. METODOLOGÍA PARA EL DESARROLLO DE PORTALES CENTRADA EN EL USUARIO:. [En línea] 2008.
<http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/854/2100>.

González Rodríguez, Leover Armando. 2006. Alternativas para el desarrollo de Aplicaciones Web. [En línea] 2006. <http://www.informatica-juridica.com/trabajos/trabajosVarios.asp>.

—. **2013.** Alternativas para el desarrollo de Aplicaciones Web. [En línea] enero de 2013. <http://www.informatica-juridica.com/trabajos/trabajosVarios.asp>..

Granal, Rafael. 2013. Guía para principiantes sobre optimización para motores de búsqueda. [En línea] febrero de 2013.
https://static.googleusercontent.com/media/www.google.com/es//intl/es/webmasters/docs/guia_optimizacion_motores_busqueda.pdf..

Guerra Sánchez, Esther. 2008. *Patrones de diseño: Patrón de comportamiento Decorator.* 2008.

Hernández Orallo, Enrique. 2012. Disca. [En línea] 2012.
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

Hernández, Alejandro Villén. 2008. *Patrón de Comportamiento.* 2008.

Iberestudio. 2010. ¿Qué son las TIC's? [En línea] Mayo de 2010.
<http://noticias.iberestudios.com/%C2%BFque-son-las-tic-y-para-que-sirven/>.

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

Jacobson, Ivar, Grady, Booch y Jame, Rumaugh. 2011. *El lenguaje unificado de modelado.* 2011.

BIBLIOGRAFÍA REFERENCIADA

- Jimenez, Leyda. 2015.** Portales. [En línea] 19 de Julio de 2015. <https://prezi.com/zacfmfjjsi93/portales/>.
- Karen. 2008.** APUNTES INFORMÁTICOS. [En línea] 2008. <https://avancesinformaticos.wordpress.com/tag/aup/>.
- Larman, Craig y Hall, Prentice. 2003.** *Modelo de dominio*. 2 edición. 2003.
- Lubbers, Peter, ALBERS, Brian and SALIM, Frank. 2005.** *Pro HTML5 Programming*. 2005.
- Madroñero Casadomé, Raúl Diez. 2015.** Visual Paradigm. [En línea] 2015. <http://es.scribd.com/doc/65619842/Visual-Paradigm>.
- Maestro. 2015.** Maestro de la Computacion. [En línea] Marzo de 2015. <https://www.maestrodelacomputacion.net/wordpress-joomla-o-drupal-cual-es-el-mejor-cms/>.
- Marimón, Diosbel Mezquí-a. 2012.** *Perfil de instalación de portales intranet para el Sistema Gestor de Contenido Drupal 7*. La Habana, Universidad de las Ciencias Informáticas : s.n., 2012.
- Matthew, David. 2011.** *Developing websites with jQuery mobile*. s.l. : Taylor & Francis, 2011.
- Mcfarland, Sawyer David. 2012.** *CSS3: The Missing Manual*. s.l. : O'Reilly Media, 2012.
- Microsoft. 2012.** El Patrón Singleton. [En línea] 2012. <https://msdn.microsoft.com/es-es/library/bb972272.aspx#EDAA>.
- Nazcasof. 2008.** Calidad y Software. [En línea] 2008. http://www.calidadyssoftware.com/otros/introduccion_cmml.php.
- NetBeans IDE. 2015.** NetBeans IDE. [En línea] Oracle, 2015. <https://netbeans.org/features/>.
- Puigserver, Biel Massot. 2008.** *Patrón State Bridge*. 2008.
- Pyme Networks. 2014.** pymenetwork. [En línea] 2014. <http://www.pymenetworks.es/blog/ventajas-y-desventajas-de-los-cms-iii-drupal>.
- Rodríguez, Gil Frank. 2012.** *Experto en Drupal 7 Nivel Avanzado*. s.l. : Forcontu S.L., 2012.
- Rosell León, Yorbelis. 2011.** *Sistemas gestores de contenidos: una mirada desde las ciencias de la información*. s.l. : ACIMED, 2011.

BIBLIOGRAFÍA REFERENCIADA

S. Pressman, Roger. 2012. *Ingeniería de software. Un enfoque práctico.* 2012.

Scribd. 2015. [En línea] 2015. <https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>.

UCI. 2014. *Metodología de desarrollo para la UCI.* Ingeniería de Software, Educación Superior. 2014.

Vega Lebrún, Carlos Arturo. 2007. Clasificación de Portales. [En línea] 2007. <http://www.eumed.net/tesis-doctorales/2007/cavl/clasificacion%20de%20portales.htm>.

Viteri, Sergio. 2009. Desarrollo web, desarrollo ágil. [En línea] 13 de octubre de 2009. <http://sergioviteri.com/2009/10/13/desarrollo-web-desarrollo-agil/>.

vyvquality. 2013. Pruebas de seguridad. [En línea] 2013. [Citado el: 6 de abril de 2016.] <http://www.vyvquality.com/w1/index.php/servicios/pruebas-de-seguridad.html>. ISO 9001:2008.

W3C. 2014. Guía brebe de XHTML. [En línea] 2014. [Citado el: 2016 de mayo de 22.] <http://www.w3c.es/Divulgacion/GuiasBreves/XHTML>.

WordPress. 2008. Definición de lenguaje de programación. [En línea] 2008. <http://definicion.de/lenguaje-de-programacion>.

Young, Ralph R. 2004. *The Requirements Engineering Handbook.* London : s.n., 2004. ISBN 1-58053-266-7.

Bibliografía consultada

1. Estudio de los sistemas de gestión de contenidos web. [En línea] 13 de julio de 2012. [Citado el: 10 de noviembre de 2014.] http://www.bilib.es/uploads/media/estudio_sistemas_gestion_contenidos_web_cms.pdf.
2. Miguel Angel Álvarez. Desarrollo web. Qué es un CMS. [En línea] 11 de noviembre de 2008. [Citado el: 11 de noviembre de 2014.] <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>.
3. Alvaro Fontela Sánchez. ¿Qué es Bootstrap? [En línea] Open Web CMS, 2013. [Citado el: 2014 de abril de 20.] <http://openwebcms.es/2013/que-es-bootstrap/>.
4. PostgreSQL. Comunidad de usuarios de PostgreSql [Consultado el: 20 marzo de 2012]. Disponible en: [En línea] [Citado el: 2015 de marzo de 20.] <http://www.postgresql.org/>.
5. PostgreSQL.About. [En línea] [Citado el: 2015 de febrero de 14.] <http://www.postgresql.org/about/>.
6. Eguiluz, Javier. Introducción a CSS. [En línea] [Citado el: 2 de diciembre de 2014.] http://www.jesusda.com/docs/ebooks/introduccion_css.pdf.
7. The Apache Software Foundation. [En línea] [Citado el: 15 de febrero de 2015.] <http://www.apache.org/>.
8. Visual Paradigm Internacional. Visual Paradigm for UML. [En línea] [Citado el: 2015 de marzo de 20.] <http://www.visual-paradigm.com/product/vpuml/>.
9. **Gil, Fran.** Experto en Drupal 7 Nivel avanzado. Curso de creación y gestión de portales web con con Drupal 7.
10. RODRÍGUEZ, Gil Frank. *Experto en Drupal 7 Nivel Avanzado* [online]. [no date]. ISBN 978-84-939410-5-5. Available from: <http://www.forcontu.com>

11. PETER LUBBERS, BRIAN ALBERS and FRANK SALIM. *Pro HTML5 Programming*. [no date]. ISBN 978-1-4302-2790-8 (Print) 978-1-4302-2791-5.
12. SÁNCHEZ. *Gestión del Evento Científico NeuroRehabana 2010*. [no date]. 005.12-San-G-2095-09
13. SOMMERVILLE, Ian and GALIPIENSO, María Isabel Alfonso. *Ingeniería del software*. Pearson Educación, [no date]. ISBN 9788478290741.
14. PRESSMAN, Roger S. *Ingeniería del software: un enfoque práctico*. Mikel Angoar, [no date].
15. MEZQUÍ-A MARIMÓN, Diosbel. *Perfil de instalación de portales Intranet para el Sistema Gestor de Contenidos Drupal 7*. La Habana : Universidad de las Ciencias Informát, [no date]. 005.12-Mez-P
16. ALVARADO OTERO, M^a Teresa. *Preparación y organización de eventos*. Barcelona : UOC, [no date].
17. SERRANO BARRERO, Yeleny Perdigón and CASTEL (eds.). *Proceso de desarrollo de aplicaciones Web sobre la plataforma Drupal*. 2010. 005.1-Ser-P-TD-0897-07
18. CHAFFER, J. *Drupal programming from an object-oriented perspective*. [online]. [no date]. Available from: www.drupal.org/node/547518.
19. RALPH R. YOUNG. *The Requirements Engineering Handbook*. London : ARTECH HOUSE, 2004. ISBN 1-58053-266-7.
20. BECK, K. *Una explicación de la programación extrema. Aceptar el cambio*. Pearson Education, 2010.
21. DOMÍ-NGUEZ RODRÍGUEZ, Ernesto Julio. *Solución informática para la gestión de información en eventos en el Sistema de Gestión Universitaria*. La Habana : Universidad de las Ciencias, [no date]. 005.43-Dom
22. BUTCHER, M. and DUNLAP, G. *Drupal 7 Module Development* [online]. 2010. ISBN 978-1-849511-16-2. Available from: <https://www.packtpub.com> Arredondo Morales, Perla Azucena, Hernández Torres, Martha Irene y Fabela Soto, Miguel Angel. *Servidores Web*. [En línea] Monografía 2009.

[Citado el: 2 de diciembre de 2014.] <http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml>.

23. Márquez Díaz, José, Sanpedro, Leonardo y Vargas, Félix. Instalación y configuración de Apache, un servidor Web gratis. Red de Revista Científica de América Latina y el Caribe, España y Portugal, 2002, No. 012, p. 10–23. [En línea] [Citado el: 6 de febrero de 2015.] <http://www.redalyc.org/pdf/852/85201202.pdf>.

24. Apache JMeter. Manual de usuario v1.2. [En línea] [Citado el: 29 de abril de 2015.] https://www.euskadi.eus/y7904/eu/contenidos/informacion/herramientas_ejie/eu_0213/adjuntos/JMeter.%20Manual%20de%20usuario%20v1.2.pdf.