

Universidad de las Ciencias Informáticas

Facultad 1



Título: Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas

Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor:

Luis Ernesto Zorrilla Pila

Tutores:

MSc. Laritza Magdalena Martínez Negrín

Ing. Mayra Ortiz Labrada

Ing. Lisbet Quesada Esteban

La Habana, Abril de 2016

“Año 58 de la Revolución”



*Sólo existe un sentimiento mayor que el amor hacia la libertad, que es el odio a
quien te la quita
Ernesto Che Guevara de la Serna.*

Declaración de autoría

Declaro ser el autor de la presente tesis Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas, reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Luis Ernesto Zorrilla Pila

Mcs. Laritza Magdalena Martínez Negrín

Ing. Mayra Ortiz Labrada

Ing. Lisbet Quesada Esteban

Datos del Contacto

Autor:

Luis Ernesto Zorrilla Pila

Universidad de las Ciencias Informáticas (UCI)

La Habana, Cuba

e-mail: lezorrilla@estudiantes.uci.cu

Tutores:

Mcs. Laritza Magdalena Martínez Negrín

Universidad de las Ciencias Informáticas (UCI)

La Habana, Cuba

e-mail: laritza@uci.cu

Ing. Mayra Ortiz Labrada

Universidad de las Ciencias Informáticas (UCI)

La Habana, Cuba

e-mail: molabrada@uci.cu

Ing. Lisbet Quezada Esteban

Universidad de las Ciencias Informáticas (UCI)

La Habana, Cuba

e-mail: lquezada@uci.cu

Dedicatoria

A mis padres por su entrega incondicional.

A toda la familia por su apoyo en todo momento.

A todos los profesores que han contribuido con mi formación.

Resumen

En el mundo existen disímiles portales *web* dedicados a difundir la información que generan los Observatorios Tecnológicos. Cada uno de estos portales satisface las necesidades de sus instituciones. En este sentido la Universidad de las Ciencias Informáticas (UCI), la cual ya contaba desde hace algunos años con un observatorio tecnológico decidió crear una nueva versión del portal *web* del mismo.

La tecnológica con que se desarrolló el antiguo portal ha quedado obsoleta, dificultando la toma de decisiones en la universidad, se realizó una nueva arquitectura de información con el objetivo de mejorar la ya existente.

La presente solución se encargará de reemplazar el actual portal que se encarga de difundir la información que genera el Observatorio Tecnológico de la Universidad de las Ciencias Informáticas. Para su implementación se tuvieron en cuenta las características comunes identificadas en los estudios de herramientas homólogas realizados y las solicitadas directamente por los clientes.

Finalmente se muestran las pruebas realizadas a la aplicación con el objetivo de verificar el cumplimiento de los objetivos trazados en la investigación, así como comprobar la funcionalidad de la aplicación informática.

Palabras clave: Observatorio tecnológico, vigilancia tecnológica, CMS Drupal, portal *web*.

Índice

Introducción	1
CAPÍTULO 1. Fundamentación Teórica.....	6
1.1. Conceptos fundamentales	6
1.2. Estudio de herramientas homólogas.....	7
1.2.1. Portal del Observatorio Tecnológico TIC del Ecuador	7
1.2.2. Portal del Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela	7
1.2.3. Portal del Observatorio Tecnológico TicEdu de Colombia	8
1.2.4. Portal del Observatorio Tecnológico CUJAE	8
1.2.5. Portal del Observatorio Tecnológico UPR	8
1.2.6. Portal del Observatorio de Ciencia y Tecnología InSTEC.....	8
1.2.7. Tendencias actuales de los portales web para la vigilancia tecnológica	10
1.3. Sistemas de Gestión de Contenidos (CMS).....	11
1.3.1. WordPress	11
1.3.2. Joomla	12
1.3.3. Drupal	12
1.3.4. Comparación entre los Sistemas de Gestión de Contenidos	13
1.4. Lenguajes empleados.....	14
1.4.1. Lenguaje de Marcado de Hipertexto (HTML 5).....	14
1.4.2. Hojas de Estilos en Cascada (CSS 3)	14
1.4.3. PHP 5.5.9.....	15
1.4.4. Lenguaje Unificado de Modelado (UML)	15
1.4.5. JavaScript	15
1.5. Entorno de Desarrollo Integrado (IDE).....	16
1.5.1. NetBeans 8.0	16

1.6.	Sistemas Gestores de Base de Datos	16
1.6.1.	PostgreSQL 9.4.....	17
1.6.2.	MySQL	17
1.6.3.	Comparación entre los Sistemas Gestores de Base de Datos.....	17
1.7.	Metodología a utilizar	18
1.7.1.	Variación AUP-UCI.....	18
1.8.	Herramientas utilizadas	19
1.8.1.	Visual Paradigm for UML 8.0.....	19
1.8.2.	Pgadmin III	19
1.8.3.	Servidor web Apache2.2.8	20
1.8.4.	Acunetix Web Vulnerability Scanner.....	20
1.8.5.	Apache Jmeter	20
1.9.	Conclusiones parciales	21

CAPÍTULO 2. Análisis, diseño e implementación del Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas..... 22

2.1.	Propuesta del sistema	22
2.2.	Modelo del Dominio	22
2.2.1.	Descripción de Clases del Modelo de Dominio.....	23
2.3.	Levantamiento de requisitos	24
2.3.1.	Requisitos Funcionales	24
2.3.2.	Requisitos no Funcionales	32
2.4.	Historias de usuarios.	34
2.5.	Estilos arquitectónicos	37
2.6.	Patrones de diseño	38
2.7.	Diagrama de clases del Diseño	39
2.8.	Modelo de Despliegue	40

2.9. Conclusiones Parciales.....	42
Capítulo 3. Validación del Portal <i>web</i> del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.	43
3.1. Diagrama de Componentes	43
3.2. Estándar de codificación	44
3.3. Pruebas del sistema	47
3.3.1. Pruebas Funcionales.....	47
3.3.2. Pruebas de Carga y Estrés	49
3.3.3. Pruebas de seguridad	50
3.3.4. Pruebas de usabilidad.....	52
3.4. Conclusiones Parciales.....	54
Conclusiones	55
Bibliografía.....	57
Anexos.....	61

Índice de tablas

Tabla 1: Comparación entre portales.	9
Tabla 2: Requisitos funcionales.	25
Tabla 3: Historia de Usuario Autenticar Usuario.....	34
Tabla 4: Historia de Usuario Insertar Noticia.	35
Tabla 5: Caso de prueba: Autenticar Usuario.....	48
Tabla 6: Descripción de las variables de Autenticar usuario	48
Tabla 7: Resultados de las pruebas de Usabilidad.....	53

Índice de figuras

Figura 1: Diagrama del modelo de dominio.....	23
Figura 2: Arquitectura N-Capas de Drupal	38
Figura 3: Diagrama de clases del diseño para Drupal.....	40
Figura 4: Diagrama del modelo de despliegue.....	41
Figura 5 Diagrama de Componentes	44
Figura 6 Ejemplo de Identación.....	45
Figura 7 Ejemplo de etiqueta de apertura	45
Figura 8 Ejemplo de uso de comillas.....	46
Figura 9 Ejemplo de uso de punto y coma en código PHP.....	46
Figura 10 Ejemplo de estructuras de control.....	47
Figura 11 Reporte resumen	50
Figura 12 Resultados de las pruebas de seguridad	51

Introducción

Con el desarrollo de las nuevas Tecnologías de la Información y las Comunicaciones (TIC), se ha producido una profunda revolución tecnológica. Un salto trascendental lo constituyó el surgimiento de Internet como sistema mundial de redes, donde un usuario, desde una computadora, puede acceder a la información de otra y tener comunicación con personas desde cualquier lugar del mundo. La red de redes pone a disposición de todos sus usuarios una serie de funcionalidades básicas que abren múltiples posibilidades de superación personal, profesional y de entretenimiento.

A diario, son millones los usuarios que acceden a internet en busca de información, utilizando de forma variada las potencialidades que brinda. En este sentido, los portales *web* se han convertido en una fortaleza y han pasado de ser desarrollados de forma estática a dinámica. Es decir, de estar enfocados principalmente en mostrar información permanente y contruidos por hipervínculos o enlaces (*links*) pasaron a propiciar servicios informativos e interactivos de forma eficiente.

El desarrollo de Internet y la *web* trajo consigo que se generara gran cantidad de información, por lo cual se ha planteado la necesidad de incorporar distintos elementos diferenciadores que permitan anticipar los cambios tecnológicos. En este sentido, la vigilancia tecnológica (VT) es un proceso organizado, selectivo y permanente de captar información del exterior y de la propia organización sobre ciencia y tecnología. Esta información es seleccionada, analizada y difundida para convertirla en conocimiento y utilizarla en la toma de decisiones con menor riesgo. Además, permite conocer la competencia, cambios de las tecnologías y en los mercados próximos al entorno en el que se trabaja.

La VT es una de las funciones que realiza un Observatorio Tecnológico (OT). De forma general, un OT es una herramienta que reconoce cambios en las fuentes de información que procesa, gestiona y observa. Por lo tanto, puede avisar de ciertas variaciones o diferencias en los parámetros que evalúa, generando información valiosa con un alto nivel de importancia para el cliente. Es también un espacio de captura, análisis y difusión de la información, es decir, constituye una ventana editorial del grupo u organización que le da vida, respecto al tema que se analiza.

En América Latina existe un gran número de observatorios tecnológicos, países como Venezuela, México, Colombia, Chile, Argentina, Ecuador y Brasil, son los más destacados en este ámbito. Tocan temas como el

empleo, la educación, servicios a la producción, construcción, *software*, medio ambiente, entre otros. El procesamiento de toda esta información permite entender mejor el entorno y reflexionar acerca de las estrategias a tomar. Una tendencia de los OT es que cada uno cuenta con un portal *web* que les permite difundir la información generada.

La UCI cuenta con el Centro de Información Científico Técnica que comenzó a brindar servicios de Vigilancia Tecnológica a partir del año 2006. Desde sus inicios contó con el portal *web* Vigitec, que le servía de soporte como una plataforma para divulgar estos servicios. En la actualidad la herramienta que se utiliza para gestionar todo el proceso de Vigilancia Tecnológica y la tecnología con que se desarrolló ha quedado obsoleta, dificultando la toma de decisiones en la universidad.

La solución actual presenta deficiencias en la difusión de sus listas de productos, temáticas de investigación, boletines, informes y alertas de vigilancia tecnológica por canales de sindicación de contenidos, lo que conlleva a que el portal tenga poca publicidad en la universidad y que los estudios y análisis realizados por el OT no sean del conocimiento de la comunidad UCI. Además, no dispone del proceso de solicitudes, que es un servicio demandado por los usuarios, por lo que el portal no tiene la forma de interactuar con los mismos, desaprovechando las potencialidades de la *web* 2.0, tampoco dispone de los enlaces a las redes sociales y no presenta un mapa de sitio para que los usuarios realicen la exploración en el portal de forma rápida.

A partir de la situación problemática descrita se puede identificar el siguiente **problema de la investigación**: ¿Cómo difundir la labor del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas para contribuir a la toma de decisiones en la universidad?

Por lo que el **objeto de estudio** se centrará en: el proceso de gestión de los portales *web* que se encargan de difundir la información de los OT, enmarcando el **campo de acción** en: el proceso de difusión de la información generada por el OT de la UCI.

En el contexto investigativo se define como **objetivo general**: Desarrollar un portal *web* que difunda la información procesada en el Observatorio Tecnológico de la UCI contribuyendo a la toma de decisiones de la dirección de la Universidad.

Para llevar a cabo esta propuesta de solución, se definen los siguientes **objetivos específicos**:

1. Elaborar el diseño teórico y metodológico de la investigación.
2. Identificar las herramientas, tecnologías y metodología de desarrollo a utilizar.
3. Diseñar e implementar el portal *web* del Observatorio Tecnológico de la UCI.
4. Realizar las pruebas que validen el funcionamiento del portal *web* del Observatorio Tecnológico de la UCI.

Durante el desarrollo de la investigación se propone como **idea a defender**:

Con la creación de un portal *web* que gestione la información procesada en el Observatorio Tecnológico de la UCI se contribuirá a su divulgación y a la toma de decisiones en la universidad.

Para darle cumplimiento al objetivo general planteado se realizarán las siguientes **tareas de investigación**:

1. Estudio de los referentes teóricos fundamentales que sustentan la investigación relacionados con el desarrollo de portales para los Observatorios Tecnológicos.
2. Selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo del portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.
3. Modelado de las funcionalidades del portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.
4. Implementación de las funcionalidades del portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.
5. Validación de las funcionalidades del portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas mediante las pruebas de carga y estrés, pruebas funcionales, pruebas de seguridad y pruebas de usabilidad.

Los **métodos científicos** que se utilizan en la investigación son:

Analítico – Sintético: se utiliza para analizar la información recopilada a través de los diferentes medios bibliográficos que pudieran contribuir a desarrollar mejor el diseño del sistema, y de esta forma aplicar estos conocimientos en la práctica para así adquirir una mayor preparación sobre el tema en cuestión, además se

utiliza en el análisis de las herramientas para el desarrollo de la solución informática propuesta, teniendo en cuenta sus características, ventajas y desventajas.

Entrevista: se manifiesta a través de la realización de entrevistas a los clientes, además es necesario el intercambio con el personal capacitado para obtener información referente al tema, criterios y corregir malas concepciones para un mayor entendimiento del problema a resolver y determinar las necesidades del cliente.

Observación: la observación consiste en saber seleccionar aquello que se quiere analizar, permite comprender cómo se desarrolla la creación de un portal web.

Modelación: empleado para la elaboración de los diagramas del lenguaje de modelado y en la representación de las características y las relaciones entre los objetos de la solución.

Resultados esperados:

Una vez concluido el desarrollo del presente Trabajo de Diploma se espera obtener el portal web del Observatorio Tecnológico de la UCI. El mismo, al ser publicado, deberá responder a las necesidades del grupo de especialistas que harán uso del mismo para apoyar la toma de decisiones de los usuarios de la red y propiciando el entorno de trabajo colaborativo entre las unidades de observación de la Universidad de las Ciencias Informáticas y la Red de Observatorios Universitarios.

La investigación consta de: resumen, introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y finalmente los anexos.

Capítulo 1: “Fundamentación Teórica”, se hace un análisis del estado actual del tema y se incluyen las definiciones fundamentales referentes al proceso de Vigilancia Tecnológica y Observatorios Tecnológicos. También se muestra el resultado del estudio acerca de las tecnologías, las herramientas a utilizar y la metodología de desarrollo de *software* que será empleada para dar cumplimiento al objetivo propuesto.

Capítulo 2: “Análisis, diseño e implementación del Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas”, se realiza una descripción de la construcción de la propuesta de solución a través de diversos artefactos. Además, se define todo lo relacionado a la implementación, se

especifica también cómo quedará desplegado el sistema.

Capítulo 3: “Validación del Portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas”. Se describen los resultados obtenidos en las pruebas realizadas, con el objetivo de comprobar el correcto funcionamiento del portal, además de los estándares de codificación.

CAPÍTULO 1. Fundamentación Teórica

En este capítulo se hace un análisis del estado actual del tema y se incluyen las definiciones fundamentales. También se realiza el estudio de algunos portales homólogos, se exponen las tecnologías más frecuentes en la implementación de este tipo de sistemas y se definen las características que debe tener, teniendo en cuenta los objetivos del mismo, los usuarios finales y sus expectativas.

1.1. Conceptos fundamentales

Observatorio tecnológico

Según (Pérez Acosta, y otros, 2014) un Observatorio Tecnológico es una herramienta que apoya la Vigilancia Tecnológica. Es un sistema de alerta para identificar y recopilar aquellos datos e informaciones que pueden ser fuente de amenaza u oportunidad.

El autor selecciona el concepto de Observatorio Tecnológico dado por (Marcial, 2009): se define como un centro de pensamiento que sale al encuentro de la realidad múltiple y compleja, con una actitud abierta ante el conocimiento y sensible en el análisis. Se dedica al estudio, reflexión y divulgación del conocimiento sobre la realidad del fenómeno observado o campo de observación y su perfil es el de un centro humanista, científico y cultural. Por lo tanto, se concibe como un organismo autónomo, independiente, pluralista, con capacidad crítica, que contribuye a mayor racionalidad en el debate.

Vigilancia Tecnológica

La Vigilancia Tecnológica es un proceso organizado, selectivo y permanente, de captar información del exterior y de la propia organización sobre ciencia y tecnología, seleccionarla, analizarla, difundirla y comunicarla, para convertirla en conocimiento para tomar decisiones con menor riesgo y poder anticiparse a los cambios tecnológicos. (Romero, y otros, 2013)

Portal web

Portal es un término, sinónimo de puente, para referirse a un sitio *web* que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al *World Wide Web*. Los portales tienen gran reconocimiento en Internet por el poder de influencia que poseen sobre grandes comunidades. Son empleados para localizar la información y de ahí comenzar alguna actividad en Internet. Un portal es más

bien una plataforma de despegue para la navegación en la *web* que ofrece al usuario, de forma fácil e integrada, el acceso a una serie de recursos y servicios relacionados a un mismo tema (García-Gómez 2001).

Teniendo en cuenta el estudio realizado, el autor de la presente investigación define un portal *web* como un sitio que, dentro de la *web*, ofrece de forma íntegra, una variedad de servicios al usuario, basándose en una misma temática y de forma organizada.

1.2. Estudio de herramientas homólogas

En el presente epígrafe se realiza un estudio de los portales que se encargan de difundir la información que generan algunos Observatorios Tecnológicos de Cuba y América Latina en general para identificar sus características, con el objetivo de conocer qué elementos son comunes y debería contener un portal de este tipo.

1.2.1. Portal del Observatorio Tecnológico TIC del Ecuador

El Observatorio TIC del Ecuador, es una Unidad Técnica de gestión de información, encargada de: integrar, consolidar, procesar, difundir y utilizar los datos para realizar estudios y análisis que son puestos a disposición de los ciudadanos. Además, debe contribuir en la generación de Políticas Públicas basadas en información real y dinámica, realizar el seguimiento de los resultados y el impacto de su emisión. (Ecuador, 2016)

1.2.2. Portal del Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela

El Observatorio Nacional de Ciencia, Tecnología e Innovación (ONCTI) es una institución adscrita al Ministerio del Poder Popular para la Educación Universitaria, Ciencia y Tecnología (MPPEUCT) de la República Bolivariana de Venezuela. Su función principal es recopilar, sistematizar, categorizar, analizar e interpretar información con el fin de contribuir en la definición de las políticas públicas que promuevan y fortalezcan el desarrollo científico-tecnológico, impactando económica y socialmente sobre la soberanía de la nación. Tiene como misión generar productos de información veraces, confiables, oportunos y de calidad en materia de ciencia, tecnología e innovación aplicada a diversas áreas, para satisfacer la demanda que surge de las instituciones públicas y del poder popular, a fin de promover y fortalecer el desarrollo científico

y tecnológico, con impacto en lo político, productivo, ambiental y social, que contribuya con la soberanía nacional. (Observatorio Nacional de Ciencia, 2016)

1.2.3. Portal del Observatorio Tecnológico TicEdu de Colombia

El portal educativo de Tecnologías de la Información y Comunicación para la educación, TicEdu, es un espacio en el cual estudiantes, profesores y comunidad en general pueden conocer y utilizar las diferentes herramientas tecnológicas integradas mediante las TICs en los procesos educativos. TicEdu tiene como fundamento el importante desarrollo que han experimentado en los últimos años las tecnologías de la información y las comunicaciones (TIC) propiciando una serie de demandas sociales que obligan a un replanteamiento de algunos de los principios relativos a los procesos de enseñanza - aprendizaje. (Colombia, 2016)

1.2.4. Portal del Observatorio Tecnológico CUJAE

El Observatorio Tecnológico de la CUJAE constituye una fuente de información condensada que permite la creación de capacidades estratégicas de Vigilancia Tecnológica. Es una herramienta indispensable para la toma de decisiones estratégicas, tácticas y operativas en el ámbito universitario. La comunidad universitaria tiene un papel protagónico en los productos y servicios que se obtienen, logrando que la producción científica esté reflejada como resultado de la Vigilancia Tecnológica (Naranjo, 2010).

1.2.5. Portal del Observatorio Tecnológico UPR

En el OT de la Universidad de Pinar del Río (UPR) se ofrece información sobre los desarrollos tecnológicos endógenos de los países, incluyendo a Cuba, así como resultados que constituyen tecnologías duras para introducir en el mercado internacional. Ofrece además información relevante para apoyar la definición de estrategias y políticas en ciencia y tecnología, la proyección de programas y proyectos de investigación, el licenciamiento o transferencia de tecnologías entre otros aspectos. Todo lo cual se puede realizar por regiones, temas de investigación, etc (Pérez, 2005).

1.2.6. Portal del Observatorio de Ciencia y Tecnología InSTEC

El Observatorio Infotecnológico es una unidad encargada de gestionar el conocimiento del InSTEC a través de la inteligencia y la vigilancia del entorno científico y tecnológico en las temáticas vinculadas con las líneas

de investigación y docentes que corresponden al perfil curricular de las especialidades que se imparten. Permite generar nuevos conocimientos y ofrecerlos a la comunidad de profesores y de alumnos. Además, permite establecer vínculos con otras organizaciones con el fin de compartir y recibir información necesaria para la toma de decisiones. (Suárez, 1995).

A continuación, se muestra una tabla comparativa realizada por el autor de esta investigación, en la cual se comparan los observatorios tecnológicos antes señalados. En esta se toman como base cuatro variables teniendo en cuenta lo común entre estos portales, además del criterio del cliente.

Tabla 1: Comparación entre portales.

Portales web	Variables		
	Redes Sociales	Buscador	Boletín
Observatorio Tecnológico CUJAE	si	si	si
Observatorio Tecnológico UPR	no	si	si
Observatorios de Ciencia y Tecnología InSTEC	no	si	si
Observatorio Tecnológico TIC del Ecuador	si	si	si
Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela	no	no	si

Observatorio Tecnológico TicEdu de Colombia	no	si	si
--	----	----	----

Terminado el estudio de varios de los portales *web* de los observatorios de Cuba y América Latina en general, se debe mencionar que cada uno de ellos, debido a que pertenecen a sectores o instituciones distintas, tienen sus características que los hacen diferentes.

En el caso de los sistemas nacionales estudiados, pese a que todos pertenecen a la Red de Observatorios Universitarios (ROU), cada uno mantiene su identidad y muestra sus propios productos de vigilancia tecnológica, de manera que cada uno es único. Motivo por lo que se hace necesario crear un portal *web* que responda a las necesidades de información del OT de la UCI en lugar de tratar de adaptar o integrarse a uno de los existentes.

Se debe destacar que estos portales presentan funcionalidades en común que se tendrán en cuenta para la implementación del sitio, además se corregirán elementos que no todos presentan dentro de la tabla comparativa y que son de importancia para la divulgación de la información.

1.2.7. Tendencias actuales de los portales web para la vigilancia tecnológica

Luego de realizado el estudio relacionado con tecnología, productos y servicios que ofrecen los diferentes portales *web* que se encargan de la divulgación de los resultados de la vigilancia tecnológica, se obtuvieron tendencias reflejadas en cada uno de ellos, las que se tendrán presentes en el desarrollo del portal *web* para el Observatorio Tecnológico de la UCI, entre las cuales se encuentran:

Buscador: permite a los usuarios de forma rápida y ágil encontrar los contenidos y las noticias publicadas en el portal.

Boletines: permite a los usuarios suscribirse y obtener la información generada por el observatorio.

Descargas de archivos: permite a los usuarios la descarga de los contenidos publicados en el portal.

El portal que se realizará para el Observatorio Tecnológico de la UCI debe contar con los enlaces para las redes sociales porque es una de las variables fundamentales para el posicionamiento *web*.

Otros temas que se tendrán en cuenta durante el desarrollo del portal y que son buenas técnicas para el posicionamiento *web* son el número de enlaces internos, la velocidad de carga de la página, el uso correcto de los metadatos, la utilización de descripción y título en las imágenes, utilizar un diseño adaptativo, entre otros.

1.3. Sistemas de Gestión de Contenidos (CMS)

Los sistemas de gestión de contenidos o CMS, por sus siglas en inglés, *Content Management System*, permiten la creación y administración de contenidos por parte de los usuarios principalmente en páginas *web*, permitiendo manejar de manera independiente el contenido y el diseño. Son de fácil edición y administración, donde los usuarios participan de forma interactiva y colaborativa en la creación de un producto de interés para un grupo, institución o comunidad de práctica (Vidal Ledo, y otros, 2008).

Además, un CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página *web*. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza la base de datos, incluyendo nueva información o editando la existente (Hurtado Rivas, 2015).

A continuación, se hace referencia a algunos de los CMS utilizados actualmente con el objetivo de fundamentar la elección del que se empleará en el desarrollo del portal del OT de la UCI.

1.3.1. WordPress

WordPress nació del deseo de un elegante y bien estructurado sistema de publicación personal basado en PHP y MySQL, bajo la Licencia Pública General en su versión 2 (GPLv2 o posterior). Es un producto maduro y estable. En principio, está configurado para usar una bitácora o *weblog* por sitio o instalación, pero también es posible tener varios blogs con diferentes o una única base de datos (Wordpress, 2005).

WordPress se centra en la facilidad de uso, solo se necesitan conocimientos básicos de Internet y acceso al mismo para poder descargar *plugins* para una fácil expansión del sistema además de los añadidos. Esta, aunque es una ventaja puede constituir su principal debilidad ya que a veces el desarrollador necesita

implementar una determinada funcionalidad y no lo puede hacer porque existen normas que restringen la modificación de código o no existen *plugins* que permitan la realización de dicha funcionalidad. Una modificación en lugares donde no se debe, lleva a que el sistema quede inservible de forma definitiva. Además, tiene como inconveniente que es solo indicado para páginas *web* sencillas y blogs, y cuando se tiene un portal mixto es necesario separar las carpetas de blogs de las páginas porque suelen aparecer conflictos si no se realiza esta acción (Wordpress, 2005).

1.3.2. Joomla

Joomla es un CMS que permite construir sitios *web* y aplicaciones en línea de gran alcance. Posee una gran facilidad de uso y extensibilidad. Joomla cuenta con características que permiten el correcto funcionamiento de los portales, entre ellas la separación de contenidos y el diseño en el Lenguaje Extensible de Marcado de Hipertexto (XHTML) y Hoja de Estilo en Cascada (CSS), lo que permite modificar de manera sencilla el estilo del portal. Soporta *plugins* y permite la creación de páginas estáticas y dinámicas (Joomla Group, 2005).

Permite desarrollar sitios *web* eficientes e interactivos además de crear, modificar o eliminar contenido de un sitio *web* de manera sencilla a través de un Panel de Administración. Es un software de código abierto, desarrollado en PHP y liberado bajo la Licencia Pública General de GNU. Este administrador de contenidos puede utilizarse en una computadora local, en una Intranet o a través de Internet y requiere para su funcionamiento una base de datos creada con un gestor MySQL, así como de un servidor HTTP Apache. Dentro de sus principales particularidades se encuentran la generación de código HTML bien formado, gestión de blogs, vistas de impresión de artículos, flash con noticias, foros, encuestas, calendarios, búsquedas integradas al sitio y soporte multi-idioma (Joomla Group, 2005).

1.3.3. Drupal

Es la solución para desarrollar aplicaciones *web* sofisticadas, complejas y robustas. Una de sus características fundamentales es que presenta un alto grado de escalabilidad, lo que lo hace ideal tanto para un simple sitio *web* personal como para proyectos mayores tales como portales *web*. Cuenta además con un sistema de “Búsqueda” que permite consultar todo el contenido que ha sido indexado anteriormente. Dispone de variados módulos para mejorar sus prestaciones y funcionalidades en un tiempo mínimo para un equipo pequeño de trabajo (Drupal, 2001).

Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión de determinado tema. Los comentarios también funcionan como entidades por lo que es posible gestionar sus campos y presentación igual que con el resto de las entidades del sistema. Incluye un módulo que permite a los administradores y/o usuarios crear encuestas *on-line* totalmente configurables. Incorpora también foros de discusión para crear sitios comunitarios fuertes y dinámicos (Drupal, 2001).

1.3.4. Comparación entre los Sistemas de Gestión de Contenidos

Las características que tienen en común Drupal, Joomla y WordPress son que los tres son gestores de contenidos, están escritos en PHP y son de código abierto. Los 3 CMS son modulares y están compuestos por un núcleo y módulos básicos de gestión de contenido, catálogo de módulos, aplicaciones, *plugins*, para numerosas presentaciones y temas gráficos.

Aun cuando estos CMS poseen puntos de convergencia, cada uno está integrado por aspectos que los hacen diferentes y que permiten definir un criterio de selección en la presente investigación. En el caso de Joomla y WordPress vienen más prefabricados y al hacer la instalación del paquete (núcleo + módulos básicos del CMS) se tienen más prestaciones de *frontend* que en Drupal; dichas prestaciones son tangibles para el cliente final quién verá un resultado bastante parecido a un sitio *web* al concluir la instalación. En cambio, Drupal es multipropósito, en el paquete no incluye tantos módulos para proyectos *web* estándar porque con él se termina haciendo tanto una *web* presencial, como una tienda *online*.

Los módulos en Drupal son más independientes, es decir, si se quiere montar, por ejemplo, un catálogo de productos utilizando Drupal se cuenta con diversos módulos para realizarlo, uno encargado de cargar las imágenes, uno que permite crear distintos campos, uno que permite mostrar distintas vistas del producto. En WordPress o Joomla es distinto, con un solo *plugin* se obtiene un catálogo, que puede ser configurado y modificado un poco, pero es más difícil de adaptar a casos de necesidades específicas.

Por otro lado, el mayor grado de prefabricación de WordPress y Joomla hace que sean dos plataformas menos flexibles y escalables, pero excelentes para una *web* estándar o un sistema de *blogs* en el caso de WordPress. Por el contrario, Drupal viene menos prefabricado, y requiere de un mayor trabajo para llegar a perfilar una *web* estándar, pero a partir de este punto será más útil que Joomla o WordPress.

Se decide utilizar el CMS Drupal en su versión 7.42 debido a las facilidades que brinda para el desarrollo de un portal. Además, posee un alto rendimiento, escalabilidad, un buen gestor de contenido, sin dejar de mencionar el tema de la seguridad, ya que las comunidades de desarrollo que posee están en constante evolución y detectan los problemas de seguridad que puedan aparecer en las versiones del CMS.

1.4. Lenguajes empleados

Los lenguajes de programación *web* han ido surgiendo según las necesidades de las plataformas, intentando facilitar el trabajo a los desarrolladores de aplicaciones. En el presente epígrafe se hace un análisis de los principales lenguajes de programación que se emplean en el mundo actualmente para el trabajo con las aplicaciones *web*.

1.4.1. Lenguaje de Marcado de Hipertexto (HTML 5)

HTML es un lenguaje de marcación de hipertexto que se diseñó en 1989 para crear páginas *web*. Desde su creación ha ido evolucionando de acuerdo a diversas necesidades que se orientan, sobre todo, a mejorar el procesamiento de la información, y así fue como aparecieron varias revisiones (HTML 2, HTML 3.2, HTML 4, HTML 4.01) que ampliaron y depuraron este lenguaje. La especificación de HTML 5 no se adscribe a una sintaxis o a la otra, sino que admite ambas: HTML y XHTML. De esta manera, los creadores de contenido pueden escoger entre un enfoque práctico, aunque poco riguroso (sintaxis HTML) y una visión académica y estricta (sintaxis XHTML). Con el tiempo, el Consorcio de la Gran Red Mundial (W3C) ha acabado aceptando que HTML y XHTML sean recomendaciones paralelas que pueden coexistir. HTML 5 no solo define cómo se deben analizar los documentos, sino también cómo se deben interpretar si no son válidos o si están mal formados. Actualmente, los navegadores corrigen los errores de sintaxis de distintas maneras, de modo que a los fabricantes les resulta más práctico. HTML 5 trata de poner fin a esa necesidad de ingeniería inversa de los navegadores, que compiten por definir cómo se deben subsanar los errores (Franganillo, 2011).

1.4.2. Hojas de Estilos en Cascada (CSS 3)

Cascading Style Sheets (CSS, por sus siglas en inglés): es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es una forma de separar los contenidos y su presentación, es de invaluable ayuda a la hora de crear páginas

web de estructura compleja (Eguíluz Pérez, 2008).

1.4.3. PHP 5.5.9

PHP es un lenguaje de programación de uso general de código del lado del servidor; originalmente diseñado para el desarrollo *web* de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procesara los datos. El código es interpretado por un servidor *web* con un módulo de procesador de PHP que genera la página *web* resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores *web* al igual que en casi todos los sistemas operativos y plataformas sin ningún costo (García, 2000).

1.4.4. Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo (Larman, 1999).

1.4.5. JavaScript

JavaScript es un lenguaje compacto y basado en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet. En una aplicación cliente para un navegador, las sentencias JavaScript pueden reconocer y responder a eventos generados por el usuario, como hacer *click* con el *mouse*, introducir información en formularios y navegación de documento a documento. Por ejemplo, se puede

escribir una función JavaScript que verifique que la información ingresada por el usuario sea correcta sin que haya transmisión de datos por la red.

Con JavaScript es posible validar los elementos antes de que el usuario se los envíe al servidor. De esta forma se reduce la cantidad de transacciones que se efectúan a través del protocolo http y las posibilidades de que se genere un error durante la inserción de datos. JavaScript también puede leer y escribir *cookies*, una operación que hasta ahora únicamente podía desarrollar el servidor *web* (Rodríguez R, 1998).

1.5. Entorno de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (Oracle Corporation, 2000).

1.5.1. NetBeans 8.0

NetBeans es un IDE de desarrollo utilizado por los programadores. Su licencia permite construir tanto aplicaciones *Open Source* como comerciales. Los proyectos desarrollados poseen lanzadores (*launchers*) para cada plataforma. Hace fuerte hincapié sobre la construcción del *software* de forma modular, módulo sobre módulo, ofreciendo implementados los mecanismos de descubrimiento de nuevos módulos (y de actualizaciones de los existentes) desde repositorios remotos, resolución de dependencias, activación/desactivación de módulos en caliente y comunicación entre los mismos. (Netbeans, 2016)

Cuenta con módulos que hacen extensible su uso hacia otros lenguajes de programación como PHP, siendo compatible con su versión 5.3. Tiene una creciente comunidad de usuarios, pues permite la depuración y ejecución de programas escritos.

Teniendo en cuenta que se utilizará el CMS Drupal para el desarrollo del Portal del OT, es importante destacar que permite la creación de módulos para este brindando opciones de auto completamiento.

1.6. Sistemas Gestores de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los

datos a distintos niveles de abstracción y los manipula, garantizando la seguridad e integridad de los mismos (Codd, 1970).

1.6.1. PostgreSQL 9.4

Es un sistema gestor de bases de datos relacionales de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y la corrección de los mismos. Es multiplataforma, ejecutándose en los principales sistemas operativos. Soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeos. Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede acomodar. Por sus características, garantiza la integridad de los datos, la velocidad de acceso y consultas a la base de datos. Incorpora una estructura de datos *array* y permite la declaración de funciones propias. Permite la gestión de diferentes usuarios, además de los permisos asignados a cada uno (Development Group, 2012).

1.6.2. MySQL

Es un gestor de base de datos sencillo de usar y rápido. También es uno de los más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración. MySQL ha demostrado que puede competir con los grandes nombres del mundo de la gestión de bases de datos. Lo que durante un tiempo se consideró como una sencilla aplicación para su uso en sitios *web*, se ha convertido en la actualidad en una solución viable y de misión crítica para la administración de datos. Ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL supera desde hace tiempo a muchas soluciones comerciales en velocidad y dispone de un sistema de permisos elegante y potente (Gilfillan, 2003).

1.6.3. Comparación entre los Sistemas Gestores de Base de Datos

La única analogía existente entre MySQL y PostgreSQL es que ambos son gestores de base de datos de código abierto. Sin embargo, aún ahí la semejanza es pequeña. MySQL está distribuido bajo una licencia "*Copyleft*" más restrictiva que obliga efectivamente a los vendedores de *software* propietario a liberar su código o adquirir una licencia propietaria de una sola entidad comercial y dueña de derechos de autor. En

contraste, PostgreSQL está distribuido libremente. Estos tienen perspectivas y técnicas radicalmente diferentes, y cada una está informada por un conjunto completamente diferente de preocupaciones. (Contreras, 2013)

PostgreSQL funciona más rápido en bases de datos grandes, ya sea en sistemas *web* o de escritorio. La velocidad de respuesta que ofrece PostgreSQL con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, algo que resulta loable.

MySQL se ha enfocado tradicionalmente en aplicaciones *web* de lectura, usualmente escritas en PHP, donde la principal preocupación es la optimización de consultas sencillas.

Teniendo en cuenta las características de los gestores de bases de datos antes mencionados se escoge PostgreSQL porque se ha enfocado tradicionalmente en la fiabilidad, integridad de datos y características integradas enfocadas al desarrollador. Además, tiene un planificador de consultas extremadamente sofisticado, que es capaz de unir cantidades relativamente grandes de tablas eficientemente.

1.7. Metodología a utilizar

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de *software*. Pueden ser comparadas con un plan de contingencia en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quiénes deben participar en el desarrollo de las actividades y qué papel deben tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (MERCHÁN, y otros, 2015).

Durante la selección de la metodología a utilizar para la documentación del portal *web* solo se tuvieron en cuenta las ágiles debido a que es un proyecto pequeño y dispone de un período de tiempo corto, siendo estas características importantes a tener en cuenta en este tipo de metodologías.

1.7.1. Variación AUP-UCI

No existe una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto exigiéndose así que el proceso sea configurable. En la Universidad de las Ciencias Informáticas se decidió hacer una variación de la metodología del Proceso Unificado Ágil (AUP, por

sus siglas en inglés *Agile Unified Process*), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la universidad.

Esta metodología consta de tres fases: Inicio, Ejecución y Cierre. En Inicio se lleva a cabo la planeación del proyecto. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. La fase de Ejecución se divide en las disciplinas Requisitos, Análisis y Diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. En la última fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (Universidad de las Ciencias Informáticas, 2013).

1.8. Herramientas utilizadas

En el presente epígrafe se mencionan y describen las herramientas a utilizar durante el desarrollo del *portal web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.

1.8.1. Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE que permite diseñar el sistema con todo tipo de diagramas con el Lenguaje unificado de modelado (UML). También es compatible con la gestión intensiva de casos de uso, requerimientos y diseño de base de datos. Con Visual Paradigm, el equipo de desarrollo de *software* puede realizar análisis y diseño de sistemas con facilidad. Visual Paradigm proporciona una *suite* de productos premiados que facilita a las organizaciones diseñar visualmente y de forma esquemática, integrar y desplegar sus aplicaciones de misión crítica de la empresa y sus bases de datos subyacentes (Rondón, y otros, 2011).

1.8.2. Pgadmin III

Es una aplicación gráfica para administrar el gestor de bases de datos PostgreSQL. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La aplicación

también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar *scripts* programados. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets y puede encriptarse mediante SSL para mayor seguridad (Padrón Ortiz, 2013).

1.8.3. Servidor web Apache2.2.8

El servidor Apache es un servidor *web* HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la *Apache Software Foundation*. Es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Apache es una aplicación que permite montar un servidor *web* en cualquier equipo y casi cualquier sistema operativo (Kabir, 2003).

Entre sus características se destacan:

- Multiplataforma: tiene la capacidad de funcionar en más de un sistema operativo con similares características y sin que su funcionalidad varíe en exceso.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con las API de programación de módulos.
- Extensible: por su propiedad de ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (Kabir, 2003).

1.8.4. Acunetix Web Vulnerability Scanner

Es una herramienta capaz de escanear sitios *web* en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en Internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor. (Román Muñoz, y otros, 2014)

1.8.5. Apache Jmeter

JMeter es una herramienta de pruebas cuyas funcionalidades se pueden resumir en tres:

- Diseñar un plan de pruebas, esto es, generar un fichero. *jmx*.

- Ejecutar un plan de pruebas.
- Ver de distintas formas los resultados de la ejecución de un plan de pruebas (vía *listeners*).

Para diseñar un plan de pruebas, JMeter dispone de una interfaz GUI a modo de diseñador, en la que el probador puede ir agregando componentes de manera visual, y ejecutar los componentes agregados, viendo el resultado. Una vez finalizado el diseño del plan de pruebas, la herramienta permite grabar este como un fichero. *jmx*.

La propia herramienta permite ejecutar un fichero *.jmx* previamente generado, mediante línea de comandos o la propia interfaz GUI. La ejecución de un fichero *.jmx* realiza peticiones a la aplicación objetivo a testear (JMeter dispone de la posibilidad de generar muchos tipos de peticiones: HTTP, FTP, LDAP, que se especifican al generar el fichero *.jmx*,). Para cada petición ejecutada, JMeter recopila ciertos datos. Además, en el fichero *.jmx* se puede especificar el número de usuarios de cada tipo que ejecuta las peticiones contra la aplicación, es decir, el *.jmx* simula una o más comunidades de usuarios trabajando contra la aplicación objetivo.

1.9. Conclusiones parciales

La realización del estudio de diferentes sistemas homólogos permitió identificar las tendencias actuales en el desarrollo de portales *web*, a partir de esto se caracterizaron de las funcionalidades y servicios a incluir en el portal del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.

Para el desarrollo de la solución se utilizará:

- El CMS Drupal en su versión 7.42 como sistema gestor de contenidos.
- Apache en su versión 2.2 como servidor web.
- Netbeans en su versión 8.0 como sistema de desarrollo.
- PostgreSQL en su versión 9.4 como sistema gestor de Base de datos.
- Apache Jmeter y Acunetix como herramientas de prueba.

CAPÍTULO 2. Análisis, diseño e implementación del Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas

En este capítulo se realiza una breve reseña de las características del sistema, especificando además los requisitos funcionales y no funcionales con los que deberá contar el portal web. Haciendo uso del lenguaje de modelado UML y la herramienta Visual Paradigm se elaborarán diagramas que ayuda a la comprensión de la estructura de la aplicación. Se definen también la arquitectura y el diseño del sistema.

2.1. Propuesta del sistema

Con la finalidad de darle solución al problema planteado en esta investigación se desarrollará un sistema que remplazará el actual portal que se encarga de difundir la información que genera el OT de la UCI. En este sistema se reorganizarán las funcionalidades contenidas en el anterior y se le agregarán las nuevas que surgieron a partir de los estudios de homólogos realizados.

Este portal prestará servicios en dependencia del tipo de usuario autenticado, permitiendo la seguridad de los datos. Además, brindará servicios como las alertas, los enlaces a las redes sociales y una selección de noticias que se encontrarán en la página principal.

2.2. Modelo del Dominio

El modelo del dominio representa las clases y relaciones más significativas a tener en cuenta en el desarrollo del sistema, se definen las entidades, atributos y relaciones existentes entre los objetos, representados mediante clases. Este modelado facilita la comprensión del problema ya que describe y limita el alcance del dominio.

A continuación, se muestra el diagrama del modelo de dominio realizador por el autor utilizando la herramienta de modelado Visual Paradigm.

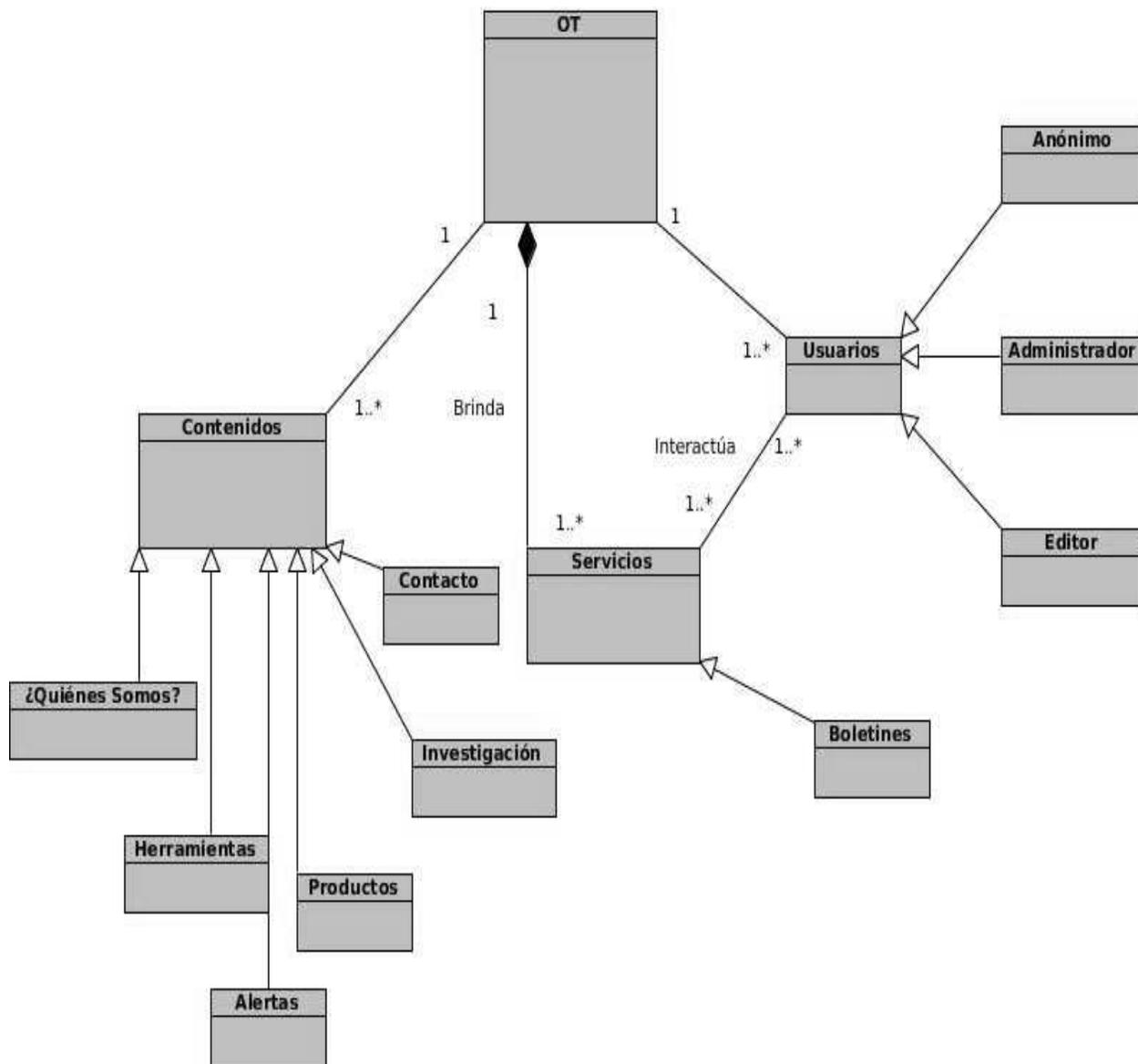


Figura 1: Diagrama del modelo de dominio.

2.2.1. Descripción de Clases del Modelo de Dominio

OT: representa el portal *web* que brinda a los usuarios un conjunto de contenidos los cuales puede consultar.

Usuarios: personas que interactúan con el portal *web*.

Anónimo: usuario sin identificación previa al que solo se le permite la visualización de los contenidos.

Administrador: usuario encargado de la gestión del portal *web*.

Editor: usuario encargado de publicar la información y los servicios que brinda el sitio *web*.

Servicios: conjunto de funcionalidades con el objetivo de responder a las necesidades de los usuarios y de la organización.

Contenidos: toda la información que se muestra en el portal.

2.3. Levantamiento de requisitos

Como paso fundamental en el proceso de desarrollo de la solución propuesta se realiza el levantamiento de requisitos.

El *Institute of Electrical and Electronics Engineers* (IEEE)¹ define un requisito como:

1. Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.
2. Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal.

Se pueden clasificar en funcionales y no funcionales y en ambos casos estos deben ser:

- Especificados por escrito. Como todo contrato o acuerdo entre dos partes.
- Posibles de probar o verificar.
- Descritos como una característica del sistema a entregar.

2.3.1. Requisitos Funcionales

Los requisitos funcionales están referidos a las funciones que debe cumplir el sistema y pueden entenderse como capacidades o condiciones que el sistema debe poseer.

A continuación, se enumeran los requisitos funcionales para la confección del portal *web* del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas mediante una tabla realizada por el autor.

¹El glosario estándar IEEE de la terminología de Ingeniería de Software identifica los términos actualmente en uso en el campo de la ingeniería de software y la definición estándar de los términos establecidos.

Tabla 2: Requisitos funcionales.

No	Nombre	Descripción	Complejidad
RF1	Autenticar Usuario	El sistema permite la autenticación de los usuarios para acceder a las funcionalidades, según el nivel de acceso establecido por el usuario Administrador. Los campos para autenticarse son: -Nombre del usuario* -Contraseña*	Baja
RF2	Insertar Usuario	El sistema permite que los usuarios con el rol de Administrador agreguen nuevos usuarios en el sistema y les asignen los privilegios, según el nivel de accesibilidad que tenga su rol. Los campos para insertar un nuevo usuario son: -Nombre de usuario* -Dirección de correo electrónico* -Contraseña* -Confirmar contraseña* -Estado -Roles	Media
RF3	Eliminar Usuario	El sistema permite a los usuarios con el rol de Administrador eliminar un usuario.	Media
RF4	Modificar Usuario	El sistema permite a los usuarios con el rol de Administrador, modificar la información relacionada con los usuarios existentes.	Media
RF5	Mostrar Usuarios	El sistema permite a los usuarios con el rol de Administrador, listar todos los usuarios existentes.	Media

RF6	Insertar Noticia	El sistema permite a los usuarios con el rol de Editor, crear nuevas noticias con los campos: -Título* -Imagen -Cuerpo*	Media
RF7	Modificar Noticia	El sistema permite a los usuarios con el rol de Editor, modificar la información de una noticia.	Media
RF8	Eliminar Noticia	El sistema permite a los usuarios con el rol de Editor, eliminar una noticia.	Media
RF9	Mostrar Noticia	El sistema permite a los usuarios con el rol de Editor, publicar las noticias, seis de ellas serán mostradas en la pantalla principal del portal. Si el usuario desea acceder a la noticia ampliada deberá dar clic encima del título o en el enlace Leer más y podrá visualizar toda la información.	Media
RF10	Insertar Alerta	El sistema permite a los usuarios con el rol de Editor, crear nuevas alertas con los campos: -Nombre* -Línea de Investigación Asociada* -Titular -Fuente -Cuerpo*	Media
RF11	Modificar Alerta	El sistema permite a los usuarios con el rol de Editor, modificar la información de una alerta.	Media
RF12	Eliminar Alerta	El sistema permite a los usuarios con el rol de Editor, eliminar una alerta.	Media

RF13	Mostrar Alerta	El sistema permite a los usuarios con el rol de Editor, publicar las alertas, tres de ellas serán mostradas en la pantalla principal del portal.	Media
RF14	Insertar Centro Productivo	El sistema permite a los usuarios con el rol de Administrador, crear nuevos centros productivos con los campos. -Nombre* -Cuerpo* -Línea de investigación asociada	Baja
RF15	Modificar Centro Productivo	El sistema permite a los usuarios con el rol de Administrador, modificar la información de un centro productivo.	Baja
RF16	Eliminar Centro Productivo	El sistema permite a los usuarios con el rol de Administrador, eliminar un centro productivo.	Baja
RF17	Mostrar Centro Productivo	El sistema permite a los usuarios con el rol de Administrador, publicar los centros productivos.	Baja
RF18	Insertar Línea de Investigación	El sistema permite a los usuarios con el rol de Administrador, crear nuevas líneas de investigación.	Baja
RF19	Modificar Línea de Investigación	El sistema permite a los usuarios con el rol de Administrador, modificar la información de una línea de investigación.	Baja
RF20	Eliminar Línea de Investigación	El sistema permite a los usuarios con el rol de Administrador, eliminar una línea de investigación.	Baja
RF21	Mostrar Línea de	El sistema permite a los usuarios con el rol	Baja

	Investigación	de Administrador, publicar las líneas de investigación.	
RF22	Insertar Grupo de Investigación	El sistema permite a los usuarios con el rol de Administrador, crear un nuevo grupo de investigación.	Baja
RF23	Modificar Grupo de Investigación	El sistema permite a los usuarios con el rol de Administrador, modificar la información de un grupo de investigación.	Baja
RF24	Eliminar Grupo de Investigación	El sistema permite a los usuarios con el rol de Administrador, eliminar un grupo de investigación.	Baja
RF25	Mostrar Grupo de Investigación	El sistema permite a los usuarios con el rol de Administrador, publicar los grupos de investigación.	Baja
RF26	Insertar Enlace de Interés	El sistema permite a los usuarios con el rol de Administrador, crear nuevos enlaces de interés con los siguientes campos. -Título -URL -Imagen	Baja
RF27	Modificar Enlace de Interés	El sistema permite a los usuarios con el rol de Administrador, modificar un enlace de interés.	Baja
RF28	Eliminar Enlace de Interés	El sistema permite a los usuarios con el rol de Administrador, eliminar un enlace de interés.	Baja
RF29	Mostrar Enlace de Interés	El sistema permite a los usuarios con el rol de Administrador, mostrar los enlaces de interés.	Baja

RF30	Insertar Boletín del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas	El sistema permite a los usuarios con el rol de Editor, crear nuevos boletines con los siguientes campos. -Título* -Fecha -Archivo* -Categoría*	Alta
RF31	Eliminar Boletín del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas	El sistema permite a los usuarios con el rol de Administrador, eliminar un boletín.	Alta
RF33	Mostrar Boletín del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas	El sistema permite a los usuarios con el rol de Administrador, publicar los boletines.	Alta
RF34	Suscribir usuarios al Boletín del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas	El sistema debe permitir al usuario realizar la suscripción al boletín. Debe ingresar: - dirección de correo*	Alta
RF35	Insertar Herramienta	El sistema permite a los usuarios con el rol de Editor, crear nuevas herramientas.	Media
RF36	Modificar Herramienta	El sistema permite a los usuarios con el rol de Editor, modificar la información de una herramienta.	Media

RF37	Eliminar Herramienta	El sistema permite a los usuarios con el rol de Editor, eliminar una herramienta.	Media
RF38	Mostrar Herramienta	El sistema permite a los usuarios con el rol de Editor, publicar las herramientas.	Media
RF39	Insertar información sobre Quiénes somos	El sistema debe permitir insertar información sobre Quiénes somos con los siguientes campos: -Título* -Cuerpo*	Media
RF40	Modificar información sobre Quiénes somos	El sistema debe permitir editar la información sobre Quiénes somos.	Media
RF42	Mostrar información sobre Quiénes somos	El sistema debe permitir publicar la información sobre Quiénes somos.	Media
RF43	Mostrar información de contacto	El sistema debe permitir a los usuarios anónimos contactar a los administradores del sitio mediante un formulario con los siguientes datos: -Nombre* -Dirección de correo electrónico* -Mensaje*	Media
RF44	Mostrar estadísticas de visitas	El sistema debe mostrar las estadísticas del portal en cuanto a cantidad de visitas.	Media
RF45	Realizar búsqueda simple	El sistema debe permitir realizar búsquedas simples sobre los contenidos existentes.	Alta
RF46	Realizar búsqueda	El sistema debe permitir realizar búsquedas	Alta

	avanzada	avanzadas sobre los contenidos existentes.	
RF47	Insertar Producto	El sistema debe permitir insertar un producto.	Media
RF49	Eliminar Producto	El sistema debe permitir eliminar un producto.	Media
RF50	Modificar Producto	El sistema debe permitir editar la información de un producto.	Media
RF50	Mostrar Producto	El sistema debe permitir publicar un producto.	Media
RF51	Mostrar enlace a las redes sociales	El sistema debe mostrar los enlaces a las redes sociales	Media
RF52	Mostrar RSS	El sistema debe permitir que los usuarios soliciten por RSS los temas de su interés.	Media
RF53	Realizar comentarios en las noticias	Los usuarios autenticados en el portal podrán realizar comentarios sobre las noticias publicadas.	Media
RF54	Aprobar comentarios	El administrador del portal podrá aprobar los comentarios realizados por los usuarios.	Media
RF55	Insertar Boletín de los Centros	El sistema debe permitir al usuario con el rol de editor insertar un boletín de un centro productivo con los siguientes campos: *Título Fecha *Archivo	Alta
RF56	Eliminar Boletín de los Centros	El sistema de permitir a los usuarios con rol editor eliminar un boletín de los centros.	Alta
RF57	Modificar Boletín de	El sistema de permitir a los usuarios con rol	Alta

	los Centros	editor modificar un boletín de los centros.	
RF58	Mostrar Boletín de los Centros	El sistema de permitir a los usuarios con rol administrador publicar un boletín de los centros.	Alta

2.3.2. Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe tener. Se agrupan en diversas categorías con el propósito de que el producto sea atractivo, usable, rápido o confiable.

Usabilidad

RNF 1. El sistema mostrará un mapa del sitio para garantizarle al usuario facilidad de navegación dentro del sistema.

RNF 2. Diseño limpio y claro que facilite la navegación.

RNF 3. Se debe lograr una estructura simple y lo más intuitiva posible de la arquitectura de información.

Seguridad

RNF 4. El sistema debe permitir usuarios con diferentes niveles de acceso.

RNF 5. Se asignarán los permisos de acceso, escritura y lectura en dependencia del rol que desempeñe cada usuario del sistema.

Soporte

RNF 6. El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarios.

Diseño e Implementación

RNF 7. Lenguaje de programación PHP 5 o superior.

RNF 8. Lenguaje de marcas de hipertexto (HTML) en su versión 5.

Disponibilidad

RNF 9. Se garantizará la disponibilidad de la información en el sistema durante todo el día.

Interfaz

RNF 10. La aplicación debe ser visible en navegadores como *Internet Explorer 9* o superior y *Mozilla Firefox 25* o superior.

Licencia

RNF 11. Uso de la licencia *PHP License*.

RNF 12. Uso de la licencia *ApacheSoftware*.

RNF 13. Uso de la licencia *BSD* de *PostgreSQL*.

RNF 14. Uso de la licencia *GNU/GPL* para el CMS *Drupal*.

Estándares Aplicables

RNF 15. Hojas de estilos en cascada (*CSS*).

Hardware

Cliente:

RNF 16. 1 GB de memoria RAM.

RNF 17. 60 GB de disco duro.

RNF 18. Procesador *Pentium IV*.

Servidor web *Apache*:

RNF 19. 4 GB de memoria RAM.

RNF 20. 60 GB de disco duro.

RNF 21. *Intel Core 2 Duo*.

Servidor base de datos:

RNF 22. 4 GB de memoria RAM

RNF 23. 500 GB de disco duro

RNF 24. *Intel Core 2 Duo*

2.4. Historias de usuario

Las historias de usuario son la forma en que se especifican los requisitos del sistema, según la metodología de desarrollo utilizada. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento estas pueden romperse, remplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. En la claridad de su descripción se encuentra el éxito del proyecto por lo cual el contenido que ellas abarcan debe ser concreto y sencillo.

En la historia de usuario se considera:

La prioridad del negocio:

- Alta: cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- Media: cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- Baja: cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

Tabla 3: Historia de Usuario Autenticar Usuario.

Número: RF1	Nombre del requisito: Autenticar usuario		
Programador: Luis Ernesto Zorrilla Pila		Iteración Asignada: Primera Iteración	
Prioridad: Media		Tiempo Estimado: 0.13 días	
Riesgo en Desarrollo: Ninguno		Tiempo Real: 0.25 días	
<p>Descripción: Ofrece al administrador o al usuario editor la posibilidad de autenticarse en el sistema para una mayor seguridad de la información, 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:			

1. Si el usuario introduce la información de forma correcta, el sistema carga la página principal dejando que el usuario acceda, según sus permisos.
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 llenarse.

Prototipo de interfaz:

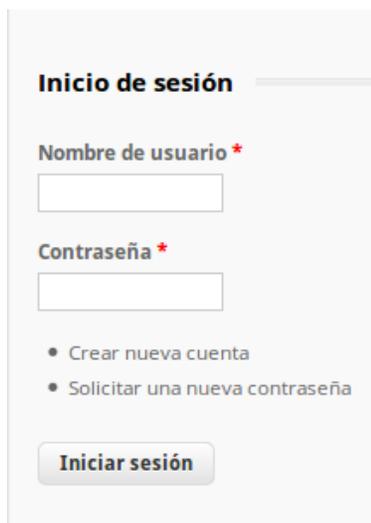


Tabla 4: Historia de Usuario Insertar Noticia.

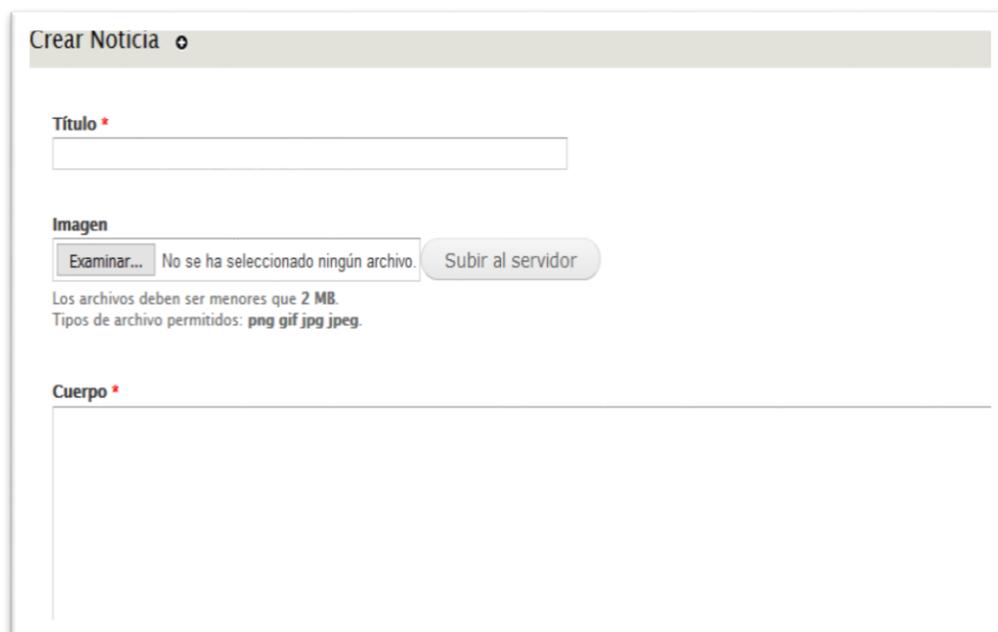
Número: RF6	Nombre del requisito: Insertar Noticia		
Programador: Luis Ernesto Zorrilla Pila	Iteración Asignada: Primera Iteración		
Prioridad: Media	Tiempo Estimado: 0.38 días		
Riesgo en Desarrollo: Ninguno	Tiempo Real: 0.25 días		
<p>Descripción: Los usuarios que tengan rol editor pueden insertar noticias en el sistema, para ello deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> • Título (Obligatorio. Campo de texto. Long. máxima 255. Permite todos los caracteres). 			

- **Imagen** (Opcional. Campo de tipo archivo. Tamaño máximo 2 MB. Permite solo los archivos con las extensiones png, gif, jpg, jpeg).
- **Cuerpo** (Obligatorio. Campo de texto. Permite todos los caracteres).

Observaciones:

1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente la noticia.
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.

Prototipo de interfaz:



Crear Noticia o

Título *

Imagen

Examinar... No se ha seleccionado ningún archivo. Subir al servidor

Los archivos deben ser menores que 2 MB.
Tipos de archivo permitidos: png gif jpg jpeg.

Cuerpo *

2.5. Estilos arquitectónicos

Para el desarrollo del portal se utiliza el CMS Drupal por lo que los patrones y la arquitectura son heredados del mismo. Muchas son las opiniones con respecto al estilo arquitectónico que presenta Drupal.

La afirmación de que Drupal presenta una arquitectura de n-capas, viene dada por la combinación de dos elementos fundamentales. El primero, que esta arquitectura es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completa. El segundo, que la característica fundamental del CMS Drupal es su adaptabilidad mediante la adición de módulos, que son al final porciones de la aplicación que se modifican, sin tener que realizar cambios en el resto de la aplicación, además de dividir su contenido en una serie de elementos básicos: nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks & menus*), permisos de usuarios (*userpermissions*) y plantillas (*templates*).

Los nodos son los elementos básicos en que Drupal almacena la información y los contenidos. Así a medida que el sitio *web* crece, lo va haciendo el número de nodos los cuales van formando un “depósito de nodos” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este “depósito de nodos”.

Los módulos son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio *web*.

La siguiente capa de Drupal la constituyen los “*blocks & menus*”. Estos permiten estructurar y organizar los contenidos en el sitio *web*. Es decir, que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos.

La capa sucesiva es la de control de usuarios y permisos. Actualmente, la mayor parte de sitios *web* son multiusuarios, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permite especificar qué tareas pueden realizar y a qué contenidos puede acceder cada tipo de usuario. Es decir, que las operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal.

La última capa, es la capa de “*templates*” y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño o apariencia del sitio *web* sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que se quiere es renovar la apariencia.

A continuación, se muestra una imagen de la arquitectura tomada de <http://www.cursosdrupal.com/content/arquitectura>.

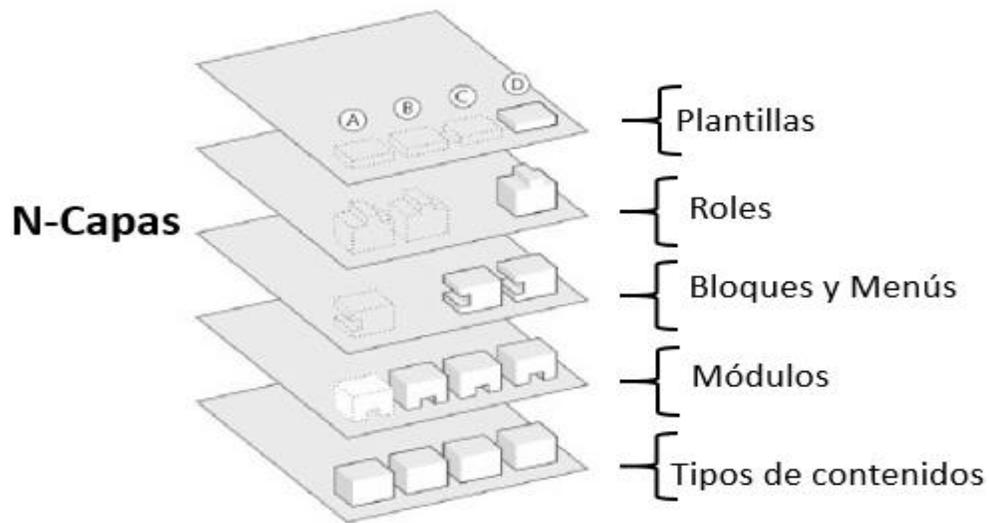


Figura 2: Arquitectura N-Capas de Drupal

2.6. Patrones de diseño

De los patrones de diseño que presenta Drupal, para el desarrollo del portal se emplean los siguientes:

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

Singleton (Instancia única): Este patrón está diseñado para que solo permita que se haga una instancia de un elemento cuando este no existe. Dentro del núcleo (*core*) de Drupal se utiliza este patrón de diseño

pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos elementos, pues Drupal solo crea una instancia de un nodo, cuando este nodo no existe.

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

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

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

2.7. Diagrama de clases del Diseño

El diagrama de clase se elabora para tener en cuenta los detalles concretos de la implementación del sistema. Es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones. (Oliveros, y otros, 2015)

La representación de los paquetes estará conformada por los módulos de Drupal.

Themes: Incluye las distintas plantillas, cuando se desee incluir un nuevo diseño, se copia la plantilla dentro de esta carpeta.

Includes: Contiene un conjunto de ficheros indispensables para el correcto funcionamiento del CMS, como, por ejemplo, la capa de abstracción o conexión que provee las funcionalidades de acceso a la base de datos de Drupal.

Modules: Incluye todos los módulos, que permiten las distintas funcionalidades del CMS, cuando se desee incorporar un nuevo módulo solo se tiene que copiar en dicha carpeta.

Misc: Contiene un conjunto de ficheros indispensables para el funcionamiento, fundamentalmente orientados al aspecto visual, CSS y JavaScript.

A continuación, se muestra el diagrama de clases del diseño de Drupal realizado por el autor mediante la herramienta de modelado Visual Paradigm.

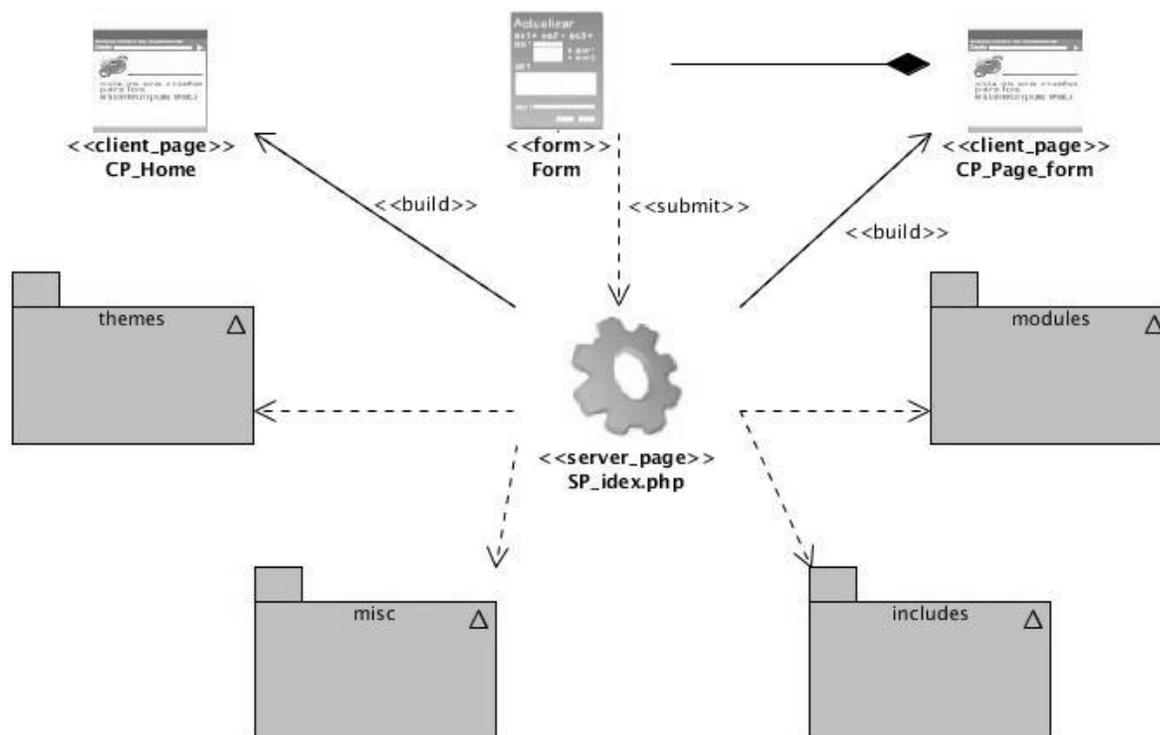


Figura 3: Diagrama de clases del diseño para Drupal.

2.8. Modelo de Despliegue

El diagrama de despliegue es utilizado para visualizar la distribución de los componentes de *software* en los nodos físicos. La relación entre los nodos es denominada como protocolos de comunicación.

El portal *web* necesita para su aplicación satisfactoria un servidor de Base de Datos (BD), un servidor *web* donde estarán alojadas las aplicaciones incluyendo el portal *web* del OT y una PC cliente para que el usuario pueda acceder a los recursos del sitio.

- **PC Cliente:** Es la computadora mediante la cual los usuarios podrán acceder al portal. Esta PC se conecta vía HTTPS con el servidor *web* a través del puerto 443.

- **PC Servidor web:** Realiza la función de intermediario entre las PC Clientes que realizan sus peticiones y el servidor de BD donde se encuentran las informaciones. Se comunica con el servidor de BD mediante el protocolo TCP/IP por el puerto 5432.
- **Servidor de base de datos:** Contiene todas las bases de datos de las funcionalidades del portal web del OT.
- **Servidor de correo:** Se comunica por el protocolo SMTP, para el envío de correos y boletines.

A continuación, se muestra el diagrama de despliegue del portal web del Observatorio Tecnológico de la Universidad de las Ciencias informática, realizado por el autor en la herramienta Visual Paradigm.

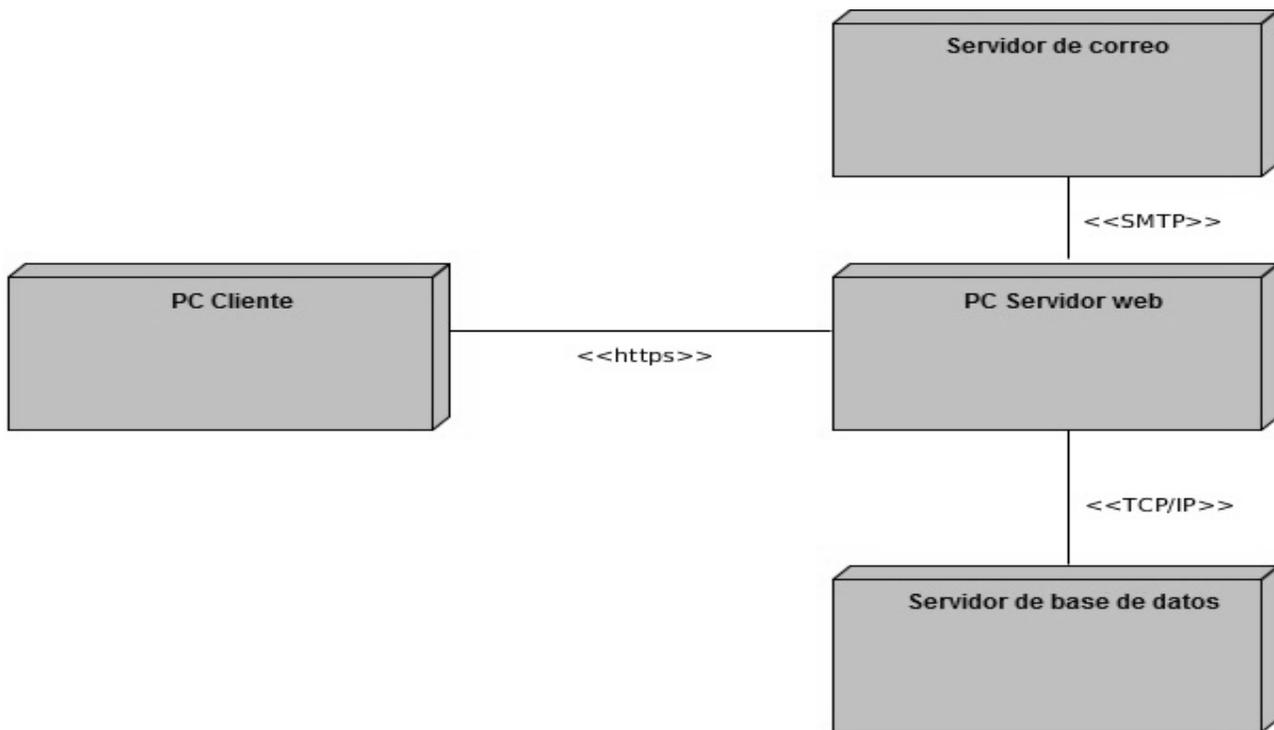


Figura 4: Diagrama del modelo de despliegue.

2.9. Conclusiones Parciales

Se realizó la descripción de la propuesta de solución logrando una mayor comprensión del sistema a implementar.

Se identificaron 58 requisitos funcionales a partir de las necesidades del cliente y fueron descritos en las historias de usuario correspondientes.

Se modelaron los diagramas de clases del diseño y de despliegue para guiar la construcción del portal *web* del OT de la UCI.

Se realizó una descripción de la arquitectura de Drupal, así como los patrones que implementa para un mejor entendimiento de su uso y funcionamiento

Capítulo 3. Validación del Portal web del Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.

En el presente capítulo se define el diagrama de componentes para mostrar la estructura en capas de la aplicación. Se detalla la estrategia de pruebas a seguir y los resultados de las mismas una vez aplicadas al sistema, contribuyendo a mejorar la calidad, usabilidad e identificar fallos en la aplicación.

3.1. Diagrama de Componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema (Casasola, 2016).

Descripción de los componentes más relevantes:

- **Paquete *Impact theme*:** Agrupa los archivos asociados al tema utilizado por el portal.
- **Paquete *js*:** Contiene todos los archivos con código javascript existentes en el tema.
- **Paquete *templates*:** Contiene todos los archivos tpl del tema.
- ***style.css*:** Contiene todo el código css del tema, código utilizado para las vistas del portal.
- ***impact_theme.info*:** Contiene la información del tema, es decir que versión de Drupal puede utilizarlo, la cantidad de regiones editables del tema, entre otros.

A continuación, se muestra una parte del diagrama de componentes del portal web del Observatorio Tecnológico de la Universidad de las Ciencias informática, realizado por el autor en la herramienta Visual Paradigm.

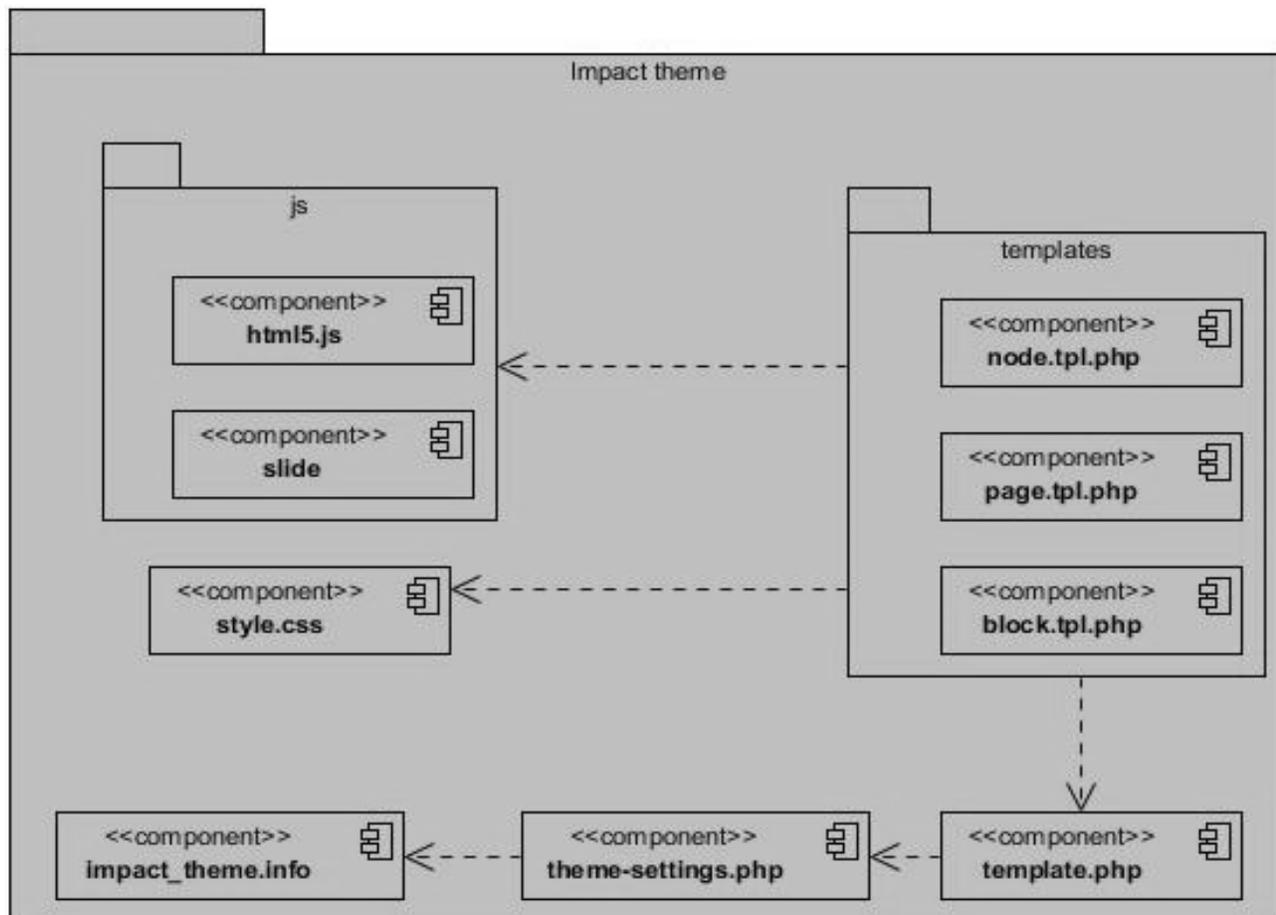


Figura 5 Diagrama de Componentes

3.2. Estándar de codificación

Un estándar de codificación es un conjunto de reglas que se siguen para la escritura del código fuente, de tal manera que a otros programadores se les facilite entender el código.

La comunidad de Drupal ha considerado necesario la introducción de una serie de estándares que faciliten la legibilidad y estructuración del desarrollo de código para Drupal. A continuación, se mencionan algunos de los estándares de codificación en Drupal (Andalucía, 2016).

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.

```
188     <?php if (!empty($tabs)): ?>
189         <?php print render($tabs); ?>
190     <?php endif; ?>
```

Figura 6 Ejemplo de Indentación

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

```
<?php
/**
 * Implements hook_html_head_alter().
 * This will overwrite the default meta character type tag with HTML5 ver
 */
function impact_theme_html_head_alter(&$head_elements) {
    $head_elements['system_meta_content_type']['#attributes'] = array(
        'charset' => 'utf-8'
    );
}
```

Figura 7 Ejemplo de etiqueta de apertura

Uso de comillas: Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\$title</h1>".

```
310 <div class="grande">
311     <div class="imagen"></div>
312     <div class="titulo"><?php print $titulo; ?></div>
313     <div class="descripcion"><?php print $descripcion; ?></div>
314 </div>
```

Figura 8 Ejemplo de uso de comillas

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

```
290 <div class="imagen"></div>
291 <div class="titulo"><?php print $titulo; ?></div>
292 <div class="descripcion"><?php print $descripcion; ?></div>
```

Figura 9 Ejemplo de uso de punto y coma en código PHP

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

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

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

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

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

```
277 <?php if (!empty($arreglo_noticias)) { /* noticias */
278     ?>
279     <div class="mceItemTable contenedor" border="0" align="center">
280         <div class="header-block" style="width: 100%; text-align: center"><?php echo t('Noticias '); ?></div>
281
282
283     <?php
284     foreach ($arreglo_noticias as $noticia) {
285         $titulo = l($noticia->title, $base_url . "/node/" . $noticia->nid);
286         $descripcion = $noticia->field_desc_resu['und'][0]['value'];
287         $src = file_create_url($noticia->field_imagen['und'][0]['uri']);
288         ?>
289         <div class="grande">
290             <div class="imagen"></div>
291             <div class="titulo"><?php print $titulo; ?></div>
292             <div class="descripcion"><?php print $descripcion; ?></div>
293         </div>
294     <?php } ?>
```

Figura 10 Ejemplo de estructuras de control

3.3. Pruebas del sistema

Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un producto de *software* antes de su puesta en marcha. Básicamente, es una fase en el desarrollo de *software* que consiste en probar las aplicaciones construidas. Existen varios tipos de pruebas que suelen ser utilizadas por los equipos de probadores. Las pruebas de caja negra y de caja blanca se encuentran en función de lo que se conoce. Por otra parte, las unitarias, de integración, aceptación, funcionales y de rendimiento se encuentran en dependencia de lo que se prueba. (Pressman, 2005).

Pruebas de Caja Negra: Son aquellas orientadas a los requisitos funcionales, buscan que se haya ingresado toda clase de entrada y que la salida obtenida es igual a la esperada.

Pruebas de Caja Blanca: Consisten en realizar pruebas para verificar que líneas específicas de código funcionan tal como está definido.

3.3.1. Pruebas Funcionales

Se denominan pruebas funcionales a las pruebas de *software* que tienen como objetivo fundamental asegurar el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados (Natalia Juristo, 2005).

Las metas de estas pruebas son:

- Verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio.
- Verificar la apropiada aceptación de datos.

Para la realización de estas pruebas se utilizó la técnica de caja negra y a continuación se muestran algunos casos de pruebas basados en requisitos que son de importancia para el cliente.

Tabla 5: Caso de prueba: Autenticar Usuario

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Autenticar usuario correctamente	El usuario introduce los datos (usuario y contraseña) correctos	V	V	El sistema verifica que los datos estén correctos de ser así, mostrará la sesión definida para este rol.	El usuario selecciona el botón "Iniciar" que se encuentra en la derecha de la pantalla principal, luego selecciona el campo Usuario y Contraseña, introduce los datos y selecciona el botón "iniciar sesión"
		prueba	prueba		
EC 1.2 Autenticar usuario incorrectamente	El usuario introduce el dato usuario incorrectamente	I	V	El sistema verifica que los datos estén correctos de no ser así, se muestra un mensaje de error.	El usuario selecciona el campo Usuario y Contraseña introduce los datos y selecciona el botón "iniciar sesión"
		pepe			

Tabla 6: Descripción de las variables de Autenticar usuario

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Usuario	texto	no	En este campo se le permite al usuario poner su nombre de usuario
2	Contraseña	texto	no	En este campo se le permite al usuario poner su contraseña

Las pruebas funcionales realizadas al portal se ejecutaron en 3 iteraciones, en la primera iteración se obtuvieron un total de 23 no conformidades (NC), de las cuales 3 fueron de funcionalidad, 8 de idioma, 7 de ortografía y las 5 restantes de validación; quedó pendiente el tema del envío del boletín al correo del usuario. Durante la segunda iteración se verificó que las NC de la primera fueran resueltas correctamente y surgieron 7 nuevas NC, relacionadas a errores de idioma, ortografía y funcionalidades, las mismas fueron resueltas en su totalidad. Luego se realizó la tercera iteración donde se verificó que las no conformidades de la segunda iteración fueran corregidas, en esta iteración no surgieron no conformidades.

3.3.2. Pruebas de Carga y Estrés

Estas pruebas se diseñan para enfrentar a los sistemas a situaciones anormales, es decir, ejecutar el sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales. Igualmente busca validar el correcto funcionamiento del sistema bajo las condiciones de carga normales para la operación.

Para la ejecución de las pruebas se utilizó una computadora con las características:

- Microprocesador Intel Corei3.
- Memoria RAM de 4 GB.
- Disco duro de 500 GB.
- Sistema Operativo Xubuntu 14.10.

La prueba de carga y estrés realizada a través de la herramienta Apache Jmeter, brinda los resultados del rendimiento del sistema. Para ello se realizó una prueba donde para 40 usuarios conectados concurrentemente en un intervalo de 1 segundo, se obtuvo lo mostrado a continuación en el reporte resumen generado por la herramienta utilizada.

Reporte resumen

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de By...
localhost/pot	40	26724	12414	35985	6767,20	0,00%	1,1/sec	22,60	21395,0
noticias	40	6130	660	8532	2121,71	0,00%	1,3/sec	22,96	18775,0
quiénes so...	40	5501	652	7792	1608,67	0,00%	1,2/sec	20,92	18531,0
Total	120	12785	652	35985	10716,51	0,00%	2,5/sec	47,79	19567,0

Figura 11 Reporte resumen

Los resultados de las pruebas de carga y estrés se consideran satisfactorios, debido a que los tiempos de respuesta del servidor ante la interacción de 40 usuarios concurrentes se encuentra en el rango de tiempo de 1 a 3 segundos. La propuesta de solución generó un total de 47.69 kb/seg de transferencia de datos para 40 usuarios, con un rendimiento de 2.5 seg. Con ello queda demostrado que la propuesta de solución es estable, ya que se mantuvo prestando servicios todo el tiempo.

3.3.3. Pruebas de seguridad

Las pruebas de seguridad consisten en revisar las aplicaciones en busca de vulnerabilidades. Este proceso se realiza bajo diferentes fases, tales como: el diseño de amenazas de acuerdo al tipo de aplicación, la ejecución de las pruebas, el análisis de los resultados y, posteriormente, la entrega de los hallazgos encontrados junto con el impacto generado por cada vulnerabilidad (ARBOLEDA, y otros, 2013).

Para la ejecución de esta prueba se utilizó la herramienta Acunetix la cual se encarga de señalar las vulnerabilidades del portal web Observatorio Tecnológico de la Universidad de las Ciencias Informáticas.

A continuación, se muestra la imagen generada por la herramienta Acunetix mostrando el resultado de las pruebas de seguridad.

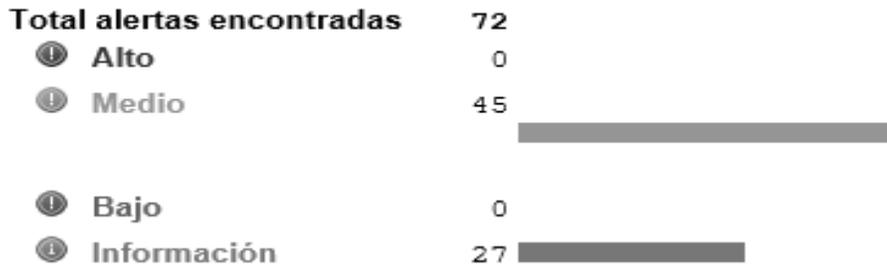


Figura 12 Resultados de las pruebas de seguridad

Una vez terminadas las pruebas de seguridad, surgieron un total de 72 alertas. Las alertas se clasificaron en:

1. Formulario HTML sin protección *Cross-site request forgery*(CSRF).

Descripción: es una clase de ataque que afecta a las aplicaciones basadas en *web*. El ataque funciona mediante la inclusión de un enlace o secuencia de comandos en una página que tiene acceso a un sitio al que se conoce el usuario (o se supone) que se han autenticado. Esta vulnerabilidad también es conocida por otros nombres como sección de manejo y ataque de un *click*.

Corrección: Para darle solución a esta alerta se utilizó un módulo llamado *securelogin* que se encarga de proveerle seguridad al portal y a su vez a los formularios

2. Tipo de entrada de contraseña con autocompletar habilitado.

Descripción: cuando se introduce un nuevo nombre y contraseña en un formulario y se envía el formulario, el navegador le pregunta si la contraseña debe ser guardada. A partir de entonces cuando se muestra el formulario, el nombre y la contraseña se rellenan de forma automática o se completan como se introduce el nombre. Un atacante con acceso local podría obtener la contraseña de texto plano de la caché del navegador.

Corrección: La función de la contraseña de autocompletar debe ser desactivada. Para desactivar la función de autocompletar, es posible utilizar un código similar al siguiente:

```
<INPUT TYPE="password" AUTOCOMPLETE="off">
```

3.3.4. Pruebas de usabilidad

Las pruebas de usabilidad son enfocadas a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva, entre otros. A la pregunta del cómo garantizar la efectividad, eficiencia y satisfacción de aplicaciones que corren bajo un entorno *web*, una respuesta que sin dudas puede resultar acertada es la usabilidad. Evaluar la usabilidad de los sitios *web* se ha convertido en un proceso crítico; sobre todo en los últimos años como consecuencia de la mayor oferta de productos y servicios en línea (Baeza, 2003).

Para la realización de las pruebas de usabilidad se utilizó una lista de chequeo, que se puede consultar en el [Anexo 1](#), que divide las preguntas a realizar en las siguientes categorías:

- Visibilidad del sistema.
- Lenguaje común entre sistema y usuario.
- Libertad y control por parte del usuario.
- Consistencia y estándares.
- Estética y diseño minimalista.
- Prevención de errores.
- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores.
- Ayuda y documentación.

Se realizó una iteración de las pruebas que arrojó un total de 8 NC de las que 2 no procedieron debido a que no se encontraban entre los requerimientos del cliente y las otras se dividieron de la siguiente manera:

Tabla 7: Resultados de las pruebas de Usabilidad

Categoría	Cantidad	No conformidad
Prevención de errores	1	No se dan indicaciones para completar campos problemáticos
Estética y diseño minimalista	1	Uso de colores no moderado.
Consistencia y estándares	1	No se añaden descripciones a las imágenes.
Lenguaje común entre sistema y usuarios	1	No utiliza los conceptos establecidos para las funciones estándar ("buscar" para las búsquedas, etc.)
Visibilidad del sistema	2	Cada pantalla no empieza con un título que describe su contenido
		Se muestran errores ortográficos

Una vez terminadas las pruebas de usabilidad y después de haber arreglado las no conformidades surgidas a partir de estas, se llega a la conclusión que respecto a la lista de chequeo utilizada el sitio es usable 100%.

3.4. Conclusiones Parciales

Se implementaron los requisitos identificados y fueron verificados mediante pruebas de usabilidad corrigiendo las NC detectadas.

Se realizaron pruebas de seguridad y se corrigieron los 2 tipos de alertas surgidas.

Se realizaron 3 iteraciones de pruebas de funcionalidad, solucionando las NC surgidas en cada iteración.

Conclusiones

Luego de culminada la presente investigación se puede arribar a las siguientes conclusiones:

El estudio de homólogos realizado permitió identificar tendencias comunes en los portales *web* de diferentes observatorios tecnológicos tales como: boletines, buscadores y enlaces a las redes sociales y se tomaron como funcionalidades que debe cumplir el portal *web* del OT de la UCI.

El portal *web* del OT de la UCI se montó sobre el CMS Drupal y su desarrollo fue guiado por la metodología AUP variación UCI.

Con la implementación del portal *web* del OT de la UCI se logró:

- Responder a las necesidades del grupo de especialistas del OT para apoyar a la toma de decisiones de los usuarios de la red.

- Propiciar un entorno de trabajo colaborativo entre las unidades de observación de la UCI y la ROU.

Se realizaron pruebas de usabilidad, carga y estrés, funcionales y de seguridad donde fueron corregidas las NC y vulnerabilidades detectadas, validando así, el correcto funcionamiento del portal *web* del OT de la UCI.

Recomendaciones

Implementar un módulo para la búsqueda y tratamiento de información utilizando técnicas de minería de datos para encaminar el trabajo de análisis en el OT.

Bibliografía

Joomla Group. 2005. Joomla. *Joomla*. [En línea] 17 de agosto de 2005. [Citado el: 2 de diciembre de 2015.] <https://www.joomla.org/joomla-in-your-language/5331-es-es-what-is-joomla-in-spanish.html>.

2007. Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela. *Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela*. [En línea] 2007. [Citado el: 2 de diciembre de 2015.] http://www.oncti.gob.ve/index.php?option=com_content&view=article&id=22&Itemid=34.

2012. Observatorio Tecnológico TicEdu de Colombia. *Observatorio Tecnológico TicEdu de Colombia*. [En línea] 15 de junio de 2012. [Citado el: 2 de diciembre de 2015.] <http://ticedu.com.co/index.php/quienes-somos>.

¿Qué son los observatorios y cuáles son sus funciones? **Marcial, Noel Angulo. 2009.** Distrito Federal : s.n., 2009.

Arias, Fidias G. 1999. *El proyecto de investigación*. 1999.

—. **2006.** *Mitos y errores en la elaboración de Tesis y Proyectos de Investigación*. 3ra. 2006.

Arias, Fidias G. 2012. *El Proyecto de Investigación. Introducción a la metodología científica*. 5ta. 2012.

Böck, Heiko. 2011. *The Definitive Guide to NetBeans™ Platform 7*. s.l. :Apress, 2011.

Boudeau, Tim. 2002. *NetBeans: the definitive guide*. s.l. : O'Reilly Media, 2002.

Codd, Edgar Frank. 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 1970, Vol. 13, 6 p. 377-387.

Corlosquet, Stéphane. 2009. *Produce and Consume Linked Data with Drupal*. Springer :s.n., 2009.

Creación del sitio web de la secretaria de educación de Risaralda e implementación de herramientas web 2.0. **Hurtado Rivas, Lorely Tatiana. 2015.** 2015.

Development Group, PostgreSQL. 2012. *Postgresql*. *Postgresql*. [En línea] 2012. [Citado el: 2 de diciembre de 2015.] <http://www.postgresql.org/about/>.

Diseño de la base de datos para sistemas de digitalización y gestión de medias. **Rondón, Y., Domígez, L. y Berenguer, A. 2011.** 15 p. 17-25, 2011, Vol. 8.

2001. Drupal. *Drupal*. [En línea] 1 de enero de 2001. [Citado el: 2 de diciembre de 2015.] <https://www.drupal.org/about>.

Flanagan, David. 2007. *avaScript. La Guía Definitiva*. 2007.

García, Esteban Trigos. 2000.*PHP 4*. s.l. : Anaya Multimedia-Anaya Interactiva, 2000.

Gauchat, Juan Diego. 2012.*El gran libro de HTML5, CSS3 y JavaScript*. s.l. :Marcombo, 2012.

Gilfillan, Ian. 2003.*MySQL*. 2003.

Gutiérrez, Juan Diego. 2004.*Desarrollo web con PHP 5 y MySQL*. 2004.

Herramientas para el trabajo colaborativo o sistema de gestión de contenidos. **Vidal Ledo, María y Concepción Báez, Carlos Mario. 2008**. 3, 2008, Vol. 22.

Html5: el nuevo estándar básico de la Web. **Franganillo, Jorge. 2011**. p 261-265, s.l. :AnuarioThinkEPI, 2011, Vol. 5.

Incorporación de Proactividad a los Agentes en un Observatorio Tecnológico. **FERNÁNDEZ, YahimaHadfeg, MORALES-AGUIAR, Nelson y MORENO-ESPINO, Maylin. 2014**. 13 p.72-79, 2014.

Introducción a CSS. **Eguíluz Pérez, Javier. 2008**. 2008.

Kabir, Mohammed J. 2003.*Servidor Apache 2*. 2003.

Larman, Craig. 1999.*UML y Patrones*. Pearson. 1999.

Los CMS como pieza fundamental en el despliegue de la Web semántica. **Pastor Sánchez, Juan Antonio. 2012**. p. 184-189, 2012, Vol. vol. 6. AnuarioThinkEPI.

Lubbers, Peter. 2011.*Pro HTML5 programming*. New York :Apress, 2011.

Manger, Jason J., Soto, Saúl Flores y Caballero, José Francisco Becerril. 1997.*Fundamentos de JavaScript*. s.l. : McGraw-Hill, 1997.

Matos, José Manuel Pardo. 2013.*Proceso de Pruebas de Aceptación de Software*. La Habana : s.n., 2013.

Medios de Comunicación, Publicidad y Adiciones. **García, Jose A del Castillo y Sánchez López, Carmen. 2009**. 978-84-414-2109-7, Madrid :Edaf, 2009.

2015.*Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.

Métodos Prospectivos Implementados en la Vigilancia Tecnológica y Prospectiva de Vehículos Eléctricos (Evs) y Tecnologías Periféricas en Colombia. **García, Javier Rosero, Díaz, Maria Victoria Victoria y Prias, Omar. 2015**. 2, 2015, Vol. 11.

Naranjo, Yanelis Jerez. 2010. Observatorio Tecnológico CUJAE. *Observatorio Tecnológico CUJAE*. [En línea] 2010. [Citado el: 2 de diciembre de 2015.] <http://observatorio.cujae.edu.cu/pages/view/43/%C2%BFquienes-somos>.

2000. NetBeans. *NetBeans*. [En línea] Sun Microsystems/Oracle Corporation, diciembre de 2000. [Citado el: 2 de diciembre de 2015.] <http://netbeans.org>.

2008. Observatorio Tecnológico TIC del Ecuador. *Observatorio Tecnológico TIC del Ecuador*. [En línea] 2008. [Citado el: 2 de diciembre de 2015.] <http://www.observatoriotic.mintel.gob.ec>.

Pérez, Maidelyn Díaz. 2005. Observatorio Tecnológico UPR. *Observatorio Tecnológico UPR*. [En línea] 2005. [Citado el: 2 de diciembre de 2015.] <http://observatorio.progintec.upr.edu.cu/>.

PGADMIN III: Administrador de Base de Datos Open Source PostgreSQL. **PadrónOrtiz, Fredy Marcelo. 2013.** 2013.

Pilgrim, Mark. 2010. *HTML5: up and running*. .s.l. : O'Reilly Media, 2010.

Plaza Chinchón, Gonzalo. 2006. SlideShare. *SlideShare*. [En línea] 4 de octubre de 2006. [Citado el: 2 de diciembre de 2015.] <http://es.slideshare.net/madgon/vigilancia-tecnologica-2739313>.

Rodríguez R, José Manue. 1998. El Lenguaje JavaScript. *El Lenguaje JavaScript*. [En línea] 15 de noviembre de 1998. [Citado el: 2 de diciembre de 2015.] <http://www.infor.uva.es/~jmrr/tgp/tgprecurso/intro1.htm>.

Schullz, Ralph G. 2008. *Diseño web con CSS*. s.l. :Marcombo, 2008.

Senn, James A. 1996. *Análisis y Diseño de sistemas*. México : s.n., 1996.

Sierra, María. 2008. *Trabajando con Visual Paradigm for UML*. 2008.

Suárez, Pedro Romero. 1995. Observatorios de Ciencia y Tecnología InSTEC. *Observatorios de Ciencia y Tecnología InSTEC*. [En línea] noviembre de 1995. [Citado el: 2 de diciembre de 2015.] <https://observatorio.instec.cu/index.php/acerca-de/quines-somos>.

Tomlinson, Todd y Vandyke, John. 2010. *Pro Drupal 7 Development*. s.l. :Apress, 2010.

2005. Wordpress.com. *Wordpress.com*. [En línea] 2005. [Citado el: 2 de diciembre de 2015.] <https://en.wordpress.com/about/>.

Colombia, Observatorio Tecnológico TicEdu de. 2016. Observatorio Tecnológico TicEdu de Colombia. *Observatorio Tecnológico TicEdu de Colombia*. [En línea] 12 de 5 de 2016. <http://ticedu.com.co/index.php/quienes-somos>.

Contreras, Juan Angel,. 2013. *Contenidos de las materias de Base de Datos en los Planes de Estudio Universitarios de Grado en Informática en el EEES. En 12 Conferencia Iberoamericana en Sistemas, Cibernética e Informática*. Florida : s.n., 2013.

Ecuador, Observatorio Tecnológico TIC del. 2016. Observatorio Tecnológico TIC del Ecuador. *Observatorio Tecnológico TIC del Ecuador*. [En línea] 12 de 5 de 2016. <http://www.observatoriotic.mintel.gob.ec/>.

Netbeans. 2016. Netbeans. *Netbeans*. [En línea] 12 de 5 de 2016. <http://netbeans.org>.

Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela. 2016. Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela. *Observatorio Nacional de Ciencia, Tecnología e Innovación de Venezuela*. [En línea] 2016. [Citado el: 12 de 5 de 2016.] http://www.oncti.gob.ve/index.php?option=com_content&view=article&id=22&Itemid=34.

Oliveros, Alejandro y Antonelli, Leandro. 2015. *Técnicas de elicitación de requerimientos*. En *XXI Congreso Argentino de Ciencias de la Computación*. Buenos Aires : s.n., 2015.

Román Muñoz, Fernando, Sabido Cortés, Iván Israel y GARCÍA VILLALBA, Luis Javier. 2014. *Capacidades de detección de las herramientas de análisis de vulnerabilidades en aplicaciones web*. 2014.

Romero, L. y Diez, E. 2013. *Sistema de Vigilancia Tecnológica con el uso de patrones y tecnologías de la web social*. La Habana, Cuba: Cujae, : Facultad de Ingeniería Industrial., 2013.

Anexos

Anexo 1. Estructura de la lista de chequeo

No	Indicador a evaluar	Evaluación	NP	Observación
Visibilidad del sistema				
1	¿La página refleja la identidad de la empresa logos, compañía...)?	si		
2	¿Cada pantalla empieza con un título que describe su contenido?	no		
3	¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	si		
4	¿Los enlaces del menú se resaltan cuando se seleccionan?	si		
5	¿Los iconos que aparecen se identifican claramente con lo que representan?	si		
6	¿El menú de navegación aparece en un lugar destacado?	si		
7	¿No utiliza más de siete opciones principales en el menú de navegación?	no		
8	¿El logo de la organización está ubicado en el mismo lugar en todas las	si		

	páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?			
9	¿Las etiquetas de las categorías describen con precisión la información de las mismas?	si		
10	¿Las imágenes se muestran con buena resolución?	si		
11	¿No se muestran errores ortográficos?	si		
Lenguaje común entre sistema y usuario				
12	¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)	no		
Libertad y control por parte del usuario				
13	¿Ofrece el contenido en otros formatos, como dispositivos móviles?	si		
14	¿Es posible imprimir la web sin perder información?	no	NP	El cliente no lo especificó
Consistencia y estándares				
15	¿Existe un cambio visible cuando el ratón apunta a algo clickeable (excluyendo los cambios de cursor)?	si		

16	¿Existe una clara distinción entre campos “requeridos” y “opcionales” en los formularios?	si		
17	¿Se mantiene una tipografía coherente en todo el sitio web?	si		
18	¿El link al mapa del sitio aparece en todas las páginas del sitio?	si		
19	¿Añade una descripción en las imágenes?	no		
20	¿Todos los estilos se han creado en hojas CSS?	si		
Estética y diseño minimalista				
21	¿El sitio puede ser usado sin desplazamiento horizontal?	si		
22	¿El uso de los colores es moderado?	no		
Prevención de errores				
23	¿Hay ausencia de enlaces rotos o que no lleven a ninguna página?	si		
24	¿Se dan indicaciones para completar campos problemáticos?	no		
25	¿Los botones de acción, (tales como “Enviar”) siempre son invocados por el usuario y no	si		

	automáticamente invocados por el sistema cuando el último campo de un formulario ha sido lleno?			
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores				
26	¿Ofrece área de Preguntas Frecuentes con datos de ayuda a usuarios?	no	NP	El cliente no lo incluyó en sus requerimientos.
Ayuda y documentación				
27	¿Existe un vínculo a los datos de contacto en un lugar bien visible en todas las páginas web del sitio?	si		
28	¿Incluye un vínculo a los datos legales en todas las páginas?	si		
29	¿Puede el usuario ponerse en contacto con el encargado del Sitio Web para hacer sugerencias o comentarios?	si		