



Facultad 1

Título: Portal Web de la Facultad 1 de la Universidad de las Ciencias Informáticas.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor:

Wendy Rodríguez Muñoz

Tutores:

MSc. Aylín Estrada Velazco

Ing. Goar Espinosa Marrero

Ing. Javier Anias Santos

Consultante

Ing. Yurelkis de los Ángeles Riopredre

Ing. Iván Romay Aragón

La Habana, Cuba

Junio 2015

Declaración de autoría

Declaro ser la única autora de la presente tesis y 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 _____.

Wendy Rodríguez Muñoz

Ing. Javier Anias Santos

Ing. Goar Espinosa Marrero

MsC. Aylin Estrada Velazco



“En la tierra hacen falta personas que trabajen más y critiquen menos, que construyan más y destruyan menos, que prometan menos y resuelvan más, que esperen recibir menos y dar más, que digan mejor ahora que mañana.”
Ernesto Che Guevara

Agradecimientos

A mis padres por estar siempre que los necesitaba, por apoyarme cada día, por impulsarme a ser una mejor persona, por decirme: tú si puedes, cuando yo misma había perdido la fe en mí. Por todo el amor que me dan. A ustedes mis papitos les doy las gracias. No habrá día que no recuerde que soy quien soy porque tengo unos padres como ustedes, y que he llegado hasta aquí, gracias al esfuerzo que han realizado. A mi tití, por apoyarme, por reírse de mis destrozos, por confiar en mí. Ojalá sea ese pilar que te impulse a ser alguien mejor y la segunda ingeniera de la familia.

A mi novio Iván por apoyarme y aguantarme en los momentos más difíciles, por ayudarme cuando más lo necesitaba. Machi a ti, muchísimas gracias. Te amo.

A Mimi por confiar y creer en mí, por apoyarme y sobre todo por consentirme.

A Katy por ser alguien muy importante en mi vida, mi hermana.

A mis abuelos, a mis tíos y a mis primos por confiar en mí.

A Leo por estar cuando lo necesitaba, por ayudarme. Sin ti esto no hubiese sido posible.

A mis compañeros de aula durante cuatro años, especialmente a Lorena, Marycarmen, Antonio, Yoel, Marlon y Lisandra. Gracias por tantos momentos felices.

A Adrián, Gilberto, Nodelvis y Ramón por ayudarme cuando los necesité, y por sacarme de situaciones de apuro.

A Yure por ser mi mentora, mi amiga. Por ayudarme aun cuando estaba lejos.

A todos aquellos que contribuyeron a mi formación profesional.

A mis tutores que con tanta dedicación y empeño hicieron posible la realización de esta tesis.

A la Revolución, a Fidel y a la UCI por darme la oportunidad de realizar este sueño.

Y los que me faltaron de todo corazón:

¡Muchas Gracias!

Dedicatoria

Dedico el presente Trabajo de Diploma:

A mis padres por ser mi guía, mi ejemplo. Por enseñarme que nada en la vida es fácil que hay que luchar hasta el final por muy turbulento que parezca el camino.

A mi hermana que la amo. Titi faltas tú.

El portal de la Facultad 1, más conocido como el portal de los zorros, ha sido desde sus inicios el soporte fundamental para la divulgación de la información asociada a las áreas y procesos de la facultad. Sin embargo hoy día, la información no presenta un correcto flujo, debido a la carencia de funcionalidades que posibiliten una adecuada gestión, moderación y aprobación de los contenidos.

El presente trabajo de diploma que lleva por título: Portal Web de la Facultad 1 de la Universidad de las Ciencias Informáticas, tiene como objetivo desarrollar un portal web que contribuya a la gestión y difusión de la información asociadas a las áreas y procesos de la Facultad 1. El proceso de desarrollo del software se guía por la metodología de desarrollo SXP. Para la implementación de la propuesta de solución se utiliza el sistema de gestión de contenidos Drupal, PostgreSQL como sistema gestor de base de datos y NGINX como servidor web. Para verificar el correcto funcionamiento de la aplicación se aplican pruebas de usabilidad, pruebas de aceptación, pruebas de seguridad y pruebas de carga y estrés. Como resultado se obtiene un producto capaz de ofrecer a los usuarios mayores facilidades en cuanto a usabilidad, que incorpora además principios del diseño web adaptativo.

Palabras claves: aplicación, diseño web adaptativo, metodología, portal web.

1.1	Introducción.....	5
1.2	Categorías conceptuales asociadas al objeto de estudio.....	5
1.2.1	Aplicaciones web	5
1.2.2	Portales Web	5
1.2.3	Web 2.0	6
1.2.4	Usabilidad	7
1.2.5	Arquitectura de Información	8
1.2.6	Diseño web adaptativo	9
1.3	Entorno actual de los portales web universitarios	10
1.3.1	Portales Web de la Universidad de las Ciencias Informáticas	11
1.3.2	Conclusiones del estudio de homólogos	13
1.4	Metodología de desarrollo de software	13
1.4.1	Tipos de Metodologías	14
1.4.2	Fundamentación de la metodología utilizada	16
1.5	Sistemas de gestión de contenidos	17
1.6	Sistema gestor de bases de datos.....	21
1.7	Tendencias y tecnologías de desarrollo.....	23
1.7.1	Herramienta CASE.....	23
1.7.2	Lenguajes de programación.....	23
1.7.3	AJAX.....	25
1.7.4	Entorno integrado de desarrollo	25
1.7.5	Servidor de aplicaciones web.....	27
1.8	Herramientas de pruebas de software	28
1.9	Conclusiones parciales.....	29
Capítulo 2:	Características y Arquitectura del Portal Web de la Facultad 1.....	30
2.1	Introducción del capítulo.....	30
2.2	Descripción de la propuesta de solución	30

2.2.1 Usuarios y permisos.....	30
2.3 Levantamiento de requisitos.....	31
2.3.1 Requisitos funcionales del sistema	31
2.3.2 Requisitos no funcionales	33
2.4 Historias de usuarios	35
2.5 Descripción de la Arquitectura de software y los patrones de diseño.....	37
2.5.1 Arquitectura de software	37
2.5.2 Patrones de diseño	38
2.6 Diagrama de clases del diseño web	41
2.7 Diagrama de paquetes	46
2.8 Modelo de despliegue	47
2.9 Conclusiones parciales.....	48
Capítulo 3: Programación y Evaluación de la aplicación web para el Portal Web de la Facultad 1	49
3.1 Introducción.....	49
3.2 Diagrama de componentes.....	49
3.3 Estándares de codificación.....	50
3.4 Plan de iteraciones.....	54
3.5 Tareas de ingeniería.....	56
3.6 Pantalla de la aplicación	57
3.7 Validación de la propuesta de solución.....	58
3.7.1 Pruebas de aceptación	58
3.7.2 Pruebas de usabilidad.....	60
3.7.3 Pruebas de carga.....	61
3.7.4 Pruebas de estrés (stress)	61
3.7.5 Pruebas de seguridad	62
3.8 Conclusiones del capítulo	62
Conclusiones generales.....	64
Recomendaciones	64
Referencias Bibliográficas.....	65

Bibliografía Consultada.....	68
Glosario de Términos.....	69

Tablas

Tabla 1. Listado de Requisitos Funcionales (RF)	33
Tabla 2. Resumen de la etapa de captura de los RF	33
Tabla 3. Historia de usuario Autenticar usuario	35
Tabla 4. Historia de usuario Gestionar usuario.....	36
Tabla 5. Plan de iteraciones	55
Tabla 6. Tarea de ingeniería.....	56
Tabla 7. Tarea de ingeniería.....	56
Tabla 8. Caso de prueba Autenticar usuario.....	59

Figuras

Figura 1. Uso de los portales web de las facultades	11
Figura 2. Fases y flujo de trabajo de SXP.....	15
Figura 3. Fases y flujo de trabajo de Open UP	16
Figura 4. Límites de PostgreSQL	22
Figura 5. Estructura de la aplicación.....	37
Figura 6. Diagrama de clases del diseño del CMS Drupal.....	42
Figura 7. Paquete Gestionar Artículo.....	44
Figura 8. Paquete Gestionar Efeméride	45
Figura 9. Paquete Gestionar Aviso	46
Figura 10. Diagrama de paquetes	47
Figura 11. Modelo de despliegue	48
Figura 12. Diagrama de componentes.....	49
Figura 13. Resultados de las pruebas de funcionalidad.....	59
Figura 14. Resultados de las pruebas de usabilidad	60

El desarrollo que ha alcanzado la Informática como ciencia encargada de la gestión y control de la información, a partir de sus procesos de automatización, ha desencadenado el desarrollo de importantes tecnologías que tributan al progreso de la humanidad. Estas tecnologías se presentan cada vez más como una necesidad en el contexto social, donde los rápidos cambios han aumentado las demandas de uso de sistemas web de alto nivel, convirtiéndolos en una exigencia permanente. (Chinea y Rodríguez, 2009) Es por ello, que adquieren una mayor relevancia los tópicos relacionados con la calidad en el proceso de desarrollo de software, especialmente en el de aplicaciones y portales web.

El surgimiento y rápido desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), el incremento de su uso en empresas e instituciones y el fuerte crecimiento de Internet, tienen un importante impacto en el desarrollo económico y social de las naciones hoy día. En Cuba el uso de las tecnologías web no ha alcanzado el auge que presenta en otros países, aun así se evidencia en los últimos años un aumento considerable. Uno de los pilares fundamentales en el desarrollo de productos de este tipo en nuestro país es el Centro de Ideoinformática (CIDI) que se encuentra ubicado en la Universidad de las Ciencias Informáticas (UCI).

La UCI es una universidad atípica, donde mantener un adecuado nivel informacional es un elemento primordial en el logro de sus objetivos para con la sociedad. Con dicho fin se han trazado disímiles estrategias, que incluyen la creación de un canal televisivo, así como el desarrollo de soluciones informáticas que contribuyan a la divulgación de la información asociada a la institución.

Una de estas soluciones informáticas lo constituye el portal web de la Facultad 1, más conocido como el portal de los zorros (accesible desde la dirección <http://zorros.uci.cu>), el cual ha sido desde sus inicios el soporte fundamental para la divulgación de la información asociada a sus áreas y procesos. Sin embargo, la carencia de funcionalidades que permitan la gestión, moderación y aprobación de los contenidos, ha traído consigo que las áreas de la facultad¹ no puedan actualizar la información contenida en el portal.

¹ Entiéndase por áreas de la facultad a los Vicedecanatos de Administración, Formación, Investigación y Postgrado, Extensión Universitaria y Departamento de Tecnología.

Las áreas Administración y Tecnología no cuentan con un espacio en el portal para publicar la información referida a los procesos que realizan, trayendo como consecuencia que dicha información no sea accesible desde cualquier ubicación de la universidad y que en ocasiones no llegue a todos los usuarios en el momento requerido.

La carencia de metadatos en los contenidos como: fuente y fecha de publicación, contribuyen a que el usuario no tenga todos los elementos necesarios para alcanzar el nivel de información que requiere. La falta de información respecto al diseño, arquitectura y desarrollo del portal es un elemento que genera polémicas, pues el desconocimiento de estos impide su comprensión y con ello, la posibilidad de ser actualizado.

Por otra parte, el uso cada vez más amplio de tabletas y dispositivos móviles para navegar por la web ha generado la necesidad de que un portal web deba ser visible de forma óptima en cualquiera de estas tecnologías. El portal de los zorros no fue desarrollado bajo los principios del diseño web adaptativo, por lo que los usuarios que acceden a través de estos dispositivos no pueden apreciar el producto con la calidad visual requerida.

Teniendo en cuenta la problemática anteriormente planteada se identifica como **problema científico**: ¿Cómo contribuir a la gestión y difusión de la información asociada a las áreas y procesos de la Facultad 1?

Por tanto, el **objeto de estudio** está centrado en el proceso de desarrollo de portales web. El **campo de acción** queda enmarcado en el proceso de desarrollo de portales web utilizando sistemas de gestión de contenidos.

Objetivo general: Desarrollar un portal web que contribuya a la gestión y difusión de la información asociada a las áreas y procesos de la Facultad 1.

Objetivos específicos:

- ✓ Elaborar el marco teórico de la investigación mediante el estudio del estado del arte de sistemas web homólogos, herramientas y tecnologías más factibles para el desarrollo de la aplicación.
- ✓ Realizar la implementación del portal web, a partir del levantamiento de requisitos, análisis y diseño del mismo.

- ✓ Evaluar el correcto funcionamiento de la solución propuesta.

Idea a defender: El desarrollo del nuevo portal de la Facultad 1 contribuirá a la gestión y difusión de la información asociada a sus áreas y procesos.

Tareas de investigación

1. Estudio del estado del arte sobre las tendencias actuales en el desarrollo de portales web para conocer el estado real de su evolución en la actualidad.
2. Análisis de las principales herramientas, metodologías y tecnologías de desarrollo de portales web para seleccionar las más adecuadas al desarrollo de la solución propuesta.
3. Definición de los requisitos funcionales y no funcionales para el desarrollo del portal web de la Facultad 1.
4. Realización del análisis y diseño de la propuesta de solución para identificar los componentes de software que intervienen en el desarrollo de la misma.
5. Implementación de la propuesta de solución para contribuir a la gestión y difusión de la información asociada a las áreas y procesos de la Facultad 1.
6. Evaluación de la propuesta de solución para garantizar su correcto funcionamiento.

Métodos Teóricos

1. **Analítico-Sintético:** Se emplea para analizar la información encontrada acerca de los rasgos distintivos de los portales web, específicamente de los portales web universitarios. Además, permite recolectar elementos generales y específicos para establecer una relación entre los diferentes criterios que pueden existir y tributen al desarrollo de una correcta solución para la elaboración del portal web de la Facultad 1.
2. **Análisis Histórico-Lógico:** Se utiliza para analizar elementos que tributen a la correcta organización y estructuración de la información del sistema y el estado del arte relacionado con las tendencias en el desarrollo de portales web en la Universidad de las Ciencias Informáticas.
3. **Modelado:** Permite modelar el diseño de las clases y el proceso a seguir para desarrollar una nueva solución para el portal web de la Facultad 1, creando una abstracción con el objetivo de explicar la realidad.

Métodos Empíricos:

- 1. Entrevista:** Permite identificar las principales funcionalidades a tener en cuenta en el desarrollo de la propuesta de solución, así como los contenidos a mostrar.
- 2. Observación:** Se emplea en la adquisición de nuevas ideas con el objetivo de establecer métricas sobre la realización del diseño web para el portal web de la Facultad 1.
- 3. Encuesta:** Permite identificar los problemas existentes en el portal de la Facultad 1, así como las principales necesidades de sus usuarios.

Este trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica

En este capítulo se plasman los elementos teóricos que soportan la investigación, así como los principales métodos y herramientas que intervienen en el desarrollo del portal web de la Facultad 1.

Capítulo 2: Características y Arquitectura del portal web de la Facultad 1

En este capítulo se determinan las características del sistema, se definen los requisitos funcionales y no funcionales que debe cumplir y se describen a grandes rasgos, las características de la solución propuesta mediante los más diversos artefactos que especifica el proceso de desarrollo de software utilizado.

Capítulo 3: Programación y Evaluación de la aplicación web para el portal web de la Facultad 1

En este capítulo se detallan los temas referentes a la implementación de la solución y se realizan las pruebas a la aplicación con el objetivo de evaluar su correcto funcionamiento.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se precisan un conjunto de aspectos teóricos que facilitan el mejor entendimiento de la investigación. Se caracterizan los elementos fundamentales que poseen los portales web universitarios a través de la realización de un estudio de sistemas homólogos y se analizan las herramientas, tecnologías, lenguajes de modelado y metodologías de desarrollo de software para seleccionar las que serán utilizadas en la investigación.

1.2 Categorías conceptuales asociadas al objeto de estudio

Para lograr una mejor comprensión de la presente investigación se recogen a continuación un conjunto de conceptos asociados al objeto de estudio.

1.2.1 Aplicaciones web

El término de aplicación web es muy utilizado en los ambientes de desarrollo de productos de este tipo, sin embargo, aunque existen diferentes percepciones acerca de su definición, se pueden observar elementos comunes en dichas conceptualizaciones.

Una aplicación web es un sistema de hipermedia donde los recursos se encuentran vinculados unos a otros, por lo que debe verse como un sistema de nodos interconectados a través de vínculos. Estos vínculos proporcionan la forma para navegar entre los recursos de la aplicación. (Martínez et al., 2013)

Según Ávila (2014) una aplicación web se define como un programa o conjunto de programas informáticos que ayudan a los usuarios de una computadora a procesar una tarea específica. Una aplicación web está compuesta por una interfaz de usuario y es accesible a través de un navegador web. (Barrientos, 2014)

1.2.2 Portales 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*. Son sitios que los usuarios tienden a visitar como páginas de inicio. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades. La idea es emplear estos portales para localizar la información deseada. (Ramírez y Sanfrán, 2010)

1.2.2.1 Tipos de portales web en función de los objetivos del usuario

Por el vertiginoso crecimiento de internet y debido al gran número de usuarios que lo visitan asiduamente, se ha hecho necesaria la clasificación de los portales web de acuerdo a la audiencia a la que están dirigidos y las temáticas que abordan. A partir del estudio de las principales clasificaciones de los portales web, la autora de la investigación destaca como tipos de portales web en función de los objetivos del usuario, los que se muestran a continuación:

❖ Portales Horizontales o Generales

Su objetivo primordial es llegar al mayor número posible de personas con un amplio rango de informaciones y servicios. Ejemplo de portales de esta categoría son AOL (*América Online LLC*), *AltaVista*, *Lycos*, *Yahoo*, *MSN*, entre otros portales, como los sitios de prensa y subastas.

❖ Portales Verticales o Especializados

Son sitios web agrupados por áreas o sectores de interés o servicios específicos. Centran su funcionamiento en contenidos especializados dirigidos a determinado público para ofrecer información de temas afines, como puede ser un portal de música, deporte o finanzas personales.

❖ Portales Diagonales o Híbridos

Son aquellos portales que desean mezclar características de los dos anteriores: llegar a un amplio espectro de usuarios, pero conservando una temática específica ofreciendo contenidos sobre el tema de diversas naturalezas, alcances y variedades. Normalmente integran ya elementos provenientes de las redes sociales.

1.2.3 Web 2.0

En la actualidad se observa una gran diferencia respecto a las tendencias en el desarrollo de portales web y las posibilidades que ofrecen a los usuarios, dada por la incorporación de elementos dinámicos como consecuencia del surgimiento de la Web 2.0.

La Web 2.0 o Web Social es una denominación de origen que se refiere a la segunda generación en la historia de los sitios web. Su denominador común es que están basados en el modelo de una comunidad de usuarios. Abarca una amplia variedad de redes sociales, *blogs*, *wikis* y servicios multimedia interconectados, cuyo propósito es el intercambio ágil de información entre los usuarios y la colaboración

en la producción de contenidos. Todos estos sitios utilizan la inteligencia colectiva para proporcionar servicios interactivos en la red, donde el usuario tiene control para publicar sus datos y compartirlos con los demás. (intef, 2013)

Para garantizar un alto nivel de audiencia en los portales web no basta con la publicación de elementos dinámicos y de interés para los usuarios. En el proceso de desarrollo de portales web la usabilidad y la arquitectura de información juegan un papel primordial, pues inciden directamente en la calidad y por tanto en la aceptación del producto final por parte de los usuarios.

1.2.4 Usabilidad

De acuerdo con la norma ISO 9241, la usabilidad es definida como el rango en el cual el producto puede ser usado por un grupo de usuarios específicos. La usabilidad es la percepción de qué tan consistente, organizada, fácil de usar e intuitivo es el proceso de completar una tarea en particular dentro de un sistema. (Baeza-Yates, Loaiza y Martín, 2004)

En el estándar ISO/IEC 25010 dentro del cual se identifican características de la calidad del software entre las que se encuentra la usabilidad, ésta se define como: “la capacidad de un producto de software para ser entendido, aprendido, utilizado y atractivo hacia el usuario”.

Una buena usabilidad es aquella que pasa totalmente desapercibida porque el usuario puede entender sin esfuerzo, qué puede hacer, a dónde puede ir y qué le ofrece la página que consulta. Esta sencillez se traduce directamente en velocidad de interacción, que es un factor primordial que los usuarios demandan de un sitio web. (Leyva et al., 2011)

Atributos de usabilidad (Baeza-Yates, Loaiza y Martín, 2004)

- **Facilidad de aprendizaje:** Principio que hace referencia a la necesidad de minimizar el tiempo necesario que requiere un usuario, desde el no conocimiento de la aplicación hasta su uso productivo. El sistema debe ser sencillo de aprender, de tal manera que el usuario pueda comenzar rápidamente a usarlo.
- **Eficiencia:** Debe ser eficiente en su uso, de forma que al aprender a usarlo el usuario tenga un nivel de productividad alto.

- **Retención sobre el tiempo:** Su uso será fácil de recordar. Si un usuario deja de utilizarlo por un tiempo, retomar un nivel de entendimiento aceptable (dónde estaba y qué estaba haciendo) una vez que vuelve a usar la aplicación debe ser una tarea sencilla.
- **Tasas de error por parte del usuario:** ¿Qué tan frecuente el usuario comete errores y cuál es la gravedad de éstos? Se debe procurar, obviamente, que el sistema tenga una baja tasa de errores, de manera que se cometa la menor cantidad de equivocaciones posible.
- **Satisfacción subjetiva:** ¿Qué tan placentera es la utilización del sistema para los usuarios? ¿Se sienten subjetivamente satisfechos al utilizar el sistema? ¿Les gusta?

1.2.5 Arquitectura de Información

Steve Toub limita el concepto a: “la AI es definida como el arte y la ciencia de organizar espacios de información con el fin de ayudar a los usuarios a satisfacer sus necesidades de información. La actividad de organizar comprende la estructuración, clasificación y rotulado de los contenidos del sitio web”. (Toub, 2000)

Por otra parte Louis Rosenfeld y Peter Morville ofrecen una visión más contemporánea del concepto, enfocando a la Arquitectura de Información como “El arte y la ciencia de estructurar y clasificar sitios web e intranets con el fin de ayudar a los usuarios a encontrar y manejar la información”. (Rosenfeld y Morville, 2002) A pesar de la imponente evolución del término y su utilización, dichos conceptos guardan una estrecha relación, todas enfocadas en función de un factor primordial: el usuario y su rol protagónico en la interacción con el producto.

Componentes de la AI

La AI se puede separar en cuatro componentes: la organización de la información, diseño de los sistemas de navegación, elaboración de los sistemas de rotulado y sistemas de búsqueda. Cada uno de estos componentes cumple un papel fundamental en la arquitectura general del sitio web, y la deficiencia de uno de ellos puede ocasionar grandes problemas de usabilidad. (Rosenfeld y Morville, 2002)

- **Organización de la información:** Creación de estructuras de organización de información (alfabéticas, geográficas, funcional, por perfiles, metafóricas, temáticas, híbridas) y definición de las estructuras de acceso que se van a utilizar (jerárquicas, bases de datos).

- **Diseño de sistemas de navegación:** A partir de la organización de la información se crean los sistemas de navegación (globales o locales) que orientan y permiten al usuario desplazarse entre los contenidos del sitio o aplicación web, creando distintos tipos de sistemas de navegación.
- **Elaboración de los sistemas de rotulación:** Se utilizan para representar la información a través de un lenguaje adecuado para la web.
- **Sistemas de búsqueda:** Son elementos que complementan los sistemas de navegación, permitiendo la localización de la información en sitios web de gran tamaño.

Beneficios que brinda la AI (Quiala, 2014)

- Generar estructuras que soporten el cambio y el crecimiento en el tiempo.
- Asegurar la consistencia y localización de la información.
- Crear sistemas de navegación intuitivos

La AI es un elemento indispensable en cada una de las etapas de elaboración de aplicaciones web. El buen diseño de esta, contribuye a la realización de un sistema accesible y usable, fortalece la fidelización del usuario con la aplicación y favorece la creación de un producto ampliamente aceptable. (Zulueta et al., 2012)

1.2.6 Diseño web adaptativo

Debido a la proliferación de dispositivos móviles (teléfonos inteligentes y tabletas) en el mercado actual, existe una mayor diversidad de resoluciones de pantalla. Este panorama conlleva a la creación de sitios web inteligentes que se adapten a los nuevos dispositivos y la estructura de cada uno de ellos.

El término diseño web adaptativo se refiere a las técnicas de diseño y desarrollo web utilizadas para que su sitio web pueda ser visualizado de forma correcta, legible y ordenada desde cualquier dispositivo ya sea móvil o de escritorio. (Web2movil, 2013) La idea principal de esta técnica no es realizar gran cantidad de diseños para cada tipo de dispositivo, sino ser más flexible, hacer un único diseño que sepa adaptarse a las necesidades del dispositivo en cuestión, siendo transparente para el cliente esta transformación y haciendo que la usabilidad de la aplicación sea realmente cómoda para el usuario. (Vega, 2013)

1.2.6.1 Bootstrap

Bootstrap es un marco de trabajo para un desarrollo web más rápido y fácil. *Bootstrap* incluye plantillas de diseño HTML y CSS basadas en tipografía, formas, botones, cuadros, navegación, modales, carruseles de imágenes y muchas otras, así como *plugins* de *JavaScript* opcionales. (w3schools.com, 2015)

Características

Bootstrap es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles). *Bootstrap* es de código abierto y está disponible en GitHub.

El empleo de *Bootstrap framework* aporta facilidad de uso, agilidad de desarrollo, reutilización del código aportando un sencillo mantenimiento y aplicaciones ejecutables en cualquier navegador. (Vega, 2013)

1.3 Entorno actual de los portales web universitarios

Los portales web universitarios constituyen en la actualidad una poderosa herramienta, enfocadas a mantener elevados niveles de información dirigidas a un numeroso grupo de usuarios en la red. A partir de los problemas y necesidades encontradas en el portal web de la Facultad 1, se decide realizar un estudio de homólogos con el objetivo de detectar elementos que favorezcan al desarrollo del nuevo portal. Las soluciones seleccionadas para el estudio serán evaluadas mediante un conjunto de criterios, siendo de ellos los más representativos en el análisis:

1. Información de la organización (Fundación, matrícula, claustro y principales logros)
2. Funcionalidades
3. Arquitectura de Información
4. Adaptabilidad a dispositivos móviles
5. Uso de elementos gráficos y de sonido
6. Información de contacto

1.3.1 Portales Web de la Universidad de las Ciencias Informáticas

La Universidad de las Ciencias Informáticas es hoy, junto a la Universidad de Artemisa, el Centro de Estudios Superiores (CES) más joven del país. La concepción y puesta en marcha de esta institución se desarrolló de forma distinta a como habían sido concebidas las otras 63 universidades cubanas y esto le confiere especiales características que la distinguen en el concierto de la educación superior cubana.

En la actualidad la universidad se encuentra estructurada en siete facultades, las cuales, poseen una poderosa herramienta para la divulgación del quehacer cotidiano de la institución; un portal web que brinda entre otras informaciones datos referentes a las facultades, estudiantes, profesores y trabajadores en general.

De los seis portales web de las facultades que se encuentran funcionando en la actualidad, se decide analizar en el estudio los tres portales más visitados en correspondencia con el marcador de visitas asociado a cada sitio.

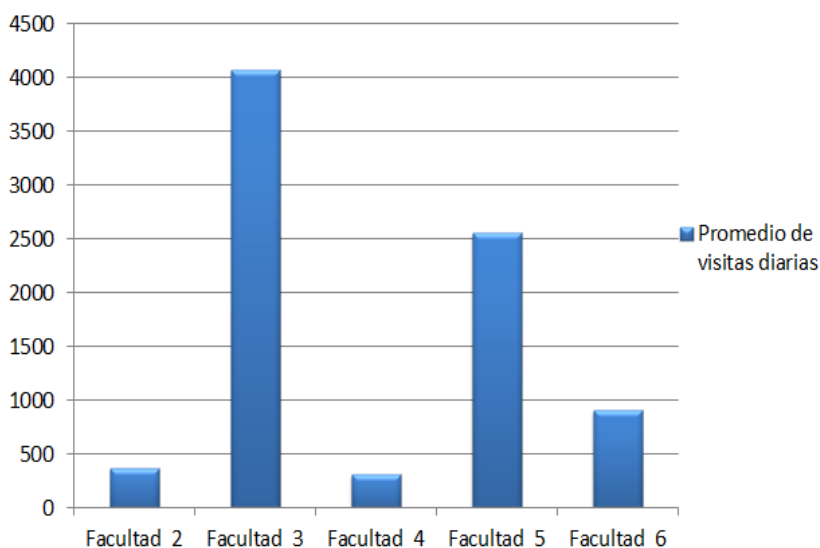


Figura 1. Uso de los portales web de las facultades

Portal web de la Facultad 3 (Dragones)

El portal web de la Facultad 3 (dragones.uci.cu) fue desarrollado mediante el Sistema Gestor de Contenidos *Joomla* en su versión 2.5. En este se puede encontrar una amplia gama de información del acontecer universitario, nacional e internacional, proporcionando a los usuarios una vía para mantenerse

constantemente informados. Investigación y Postgrado, Avisos y Clasificados, Facultad y Noticias son algunas de las secciones que se pueden observar en el sitio. La presencia de columnas temáticas como Píxeles, Atlético, Cuban3ando, Farandul3ando, AR3ir forma parte de la clave del éxito alcanzado por este portal en la comunidad de la universidad, pues tienen como punto de partida temas que resultan de interés para sus usuarios. Además de lo antes planteado se puede concluir que:

- No muestra información referente a la facultad.
- Posee escasa documentación asociada a las esferas de la facultad.
- El diseño del portal no permite que sea visualizado correctamente en los dispositivos móviles.
- Hace uso de elementos gráficos y de sonido.
- No brinda información de contacto.

Portal web de la Facultad 5 (Grundys)

El portal web de la Facultad 5 (facultad5.uci.cu) fue desarrollado mediante el Sistema Gestor de Contenidos Joomla. En la portada podremos encontrar secciones como Menú del Comedor, Actividades, Video Portada, Noticias, Fuera de Juego. En el primer nivel de navegación está presente la información relacionada con las áreas Producción, Investigación y Extensión, así como de las diferentes esferas de la Facultad. Además de ello es válido destacar que:

- Muestra información referente a la facultad.
- El diseño del portal permite que sea visualizado correctamente en los dispositivos móviles.
- Se hace uso de elementos gráficos y de sonido.
- Los contenidos se encuentran bien estructurados
- Brinda poca información de contacto.

Portal web de la facultad 6 (Gladiadores)

El portal web de la Facultad 6 (facultad6.uci.cu) fue desarrollado mediante el Sistema Gestor de Contenidos Joomla. El portal posee una interfaz sencilla, proporciona al usuario amplios caudales de información acerca de la facultad, el acontecer nacional e internacional. Dentro de las secciones que se pueden encontrar están: Horario, Docencia, Producción, Investigación y Postgrado, Organizaciones, Extensión, Noticias, Avisos, Menú del Comedor, Encuesta, Fotos en Portada, Algunas de sus características son:

- No muestra información referente a la facultad.
- El diseño del portal no permite que sea visualizado correctamente en los dispositivos móviles.
- Se hace uso de elementos gráficos y de sonido.
- Brinda información de contacto.

1.3.2 Conclusiones del estudio de homólogos

Después de realizado el estudio del entorno actual de los portales web universitarios se puede concluir que, a pesar de que las soluciones estudiadas presentan algunas dificultades, deben ser tenidas en cuenta sus principales potencialidades para el desarrollo del portal de la Facultad 1. A continuación se resumen los principales elementos detectados.

- Publicación de noticias referentes al acontecer universitario e investigativo de la institución.
- Presencia de elementos de interacción con el usuario.
- Información de contacto.
- Publicación de secciones que resulten de interés para la comunidad universitaria.
- La AI deberá estar enfocada a facilitar un acceso intuitivo a los contenidos.
- Por la importancia que se le confiere al auge y evolución de los dispositivos móviles, se decide incorporar el uso del diseño web adaptativo.

Después del análisis de los elementos teóricos es preciso definir las herramientas involucradas en el proceso de desarrollo de la propuesta de solución. Para ello se realiza un estudio minucioso de las metodologías y herramientas para la elección de las que más se adecuen a la construcción de la aplicación.

1.4 Metodología de desarrollo de software

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. (Méndez, 2010)

1.4.1 Tipos de Metodologías

Las metodologías tradicionales centran su atención en el control del proceso de desarrollo, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben generar y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo.

Otra de las posibles opciones es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. (Canós, Letelier y Penadés, 2010)

Por lo antes expuesto, y teniendo en cuenta las características del software a desarrollar, se valora la necesidad de sólo hacer un estudio de las metodologías ágiles, centrando especial atención en las metodologías de desarrollo SXP y Open UP.

Metodología ágil para el desarrollo de software (SXP)

SXP es un híbrido cubano de metodologías ágiles, que ofrece una estrategia tecnológica a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de desarrollo de software para el mejoramiento de la actividad productiva, fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil.

Consta de 4 fases principales: **Planificación-Definición** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; **Desarrollo**, es donde se

realiza la implementación del sistema hasta que esté listo para ser entregado; **Entrega**, puesta en marcha; y por último **Mantenimiento**, donde se realiza el soporte para el cliente.



Figura 2. Fases y flujo de trabajo de SXP

De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, priorización de la Lista de Reserva del Producto, definición de Historias de Usuario, diseño, implementación, planificación de las iteraciones y actividades que se van a realizar para obtener el producto y pruebas; además de las tareas necesarias para realizar las investigaciones y documentarlas en el proyecto. (Romero, Puente y Abad, 2010)

Open Up

Open Up es una metodología de desarrollo de software de código abierto, diseñado para pequeños equipos organizados, tomando una aproximación ágil del desarrollo. Open UP abraza una filosofía pragmática y ágil que se centra en la naturaleza colaborativa de desarrollo de software. Es un proceso iterativo Mínimo, Completo y Extensible que puede utilizarse tal cual o ampliarse para tratar una gran variedad de tipos de proyecto. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Está organizada dentro de cuatro áreas principales de contenido: Comunicación Y Colaboración, Intención, Solución y Administración.

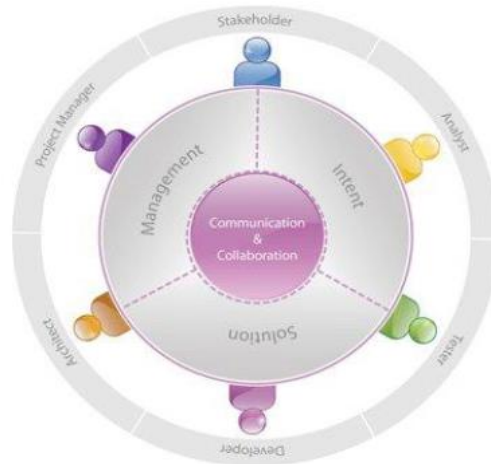


Figura 3. Fases y flujo de trabajo de Open UP

Principios básicos de Open UP

- **Colaboración:** Para alinear los intereses y un entendimiento compartido se deben desarrollar prácticas colaborativas que fomenten un ambiente saludable.
- **Enfoque:** Articular la arquitectura para facilitar la colaboración técnica, reducir los riesgos, minimizar excesos y trabajo extra. Enunciar las decisiones técnicas esenciales a través de una arquitectura creciente.
- **Evolución:** Dividir el proyecto en iteraciones cortas, enmarcadas en el tiempo para demostrar valor incremental, reducir riesgos, demostrar resultados y obtener retroalimentación temprana y continua de los clientes.
- **Balance:** Para confrontar las prioridades (necesidades y costos técnicos). Desarrollar una solución que maximice los beneficios para los interesados y cumpla con las restricciones planteadas en el proyecto. (Hernández, 2011)

1.4.2 Fundamentación de la metodología utilizada

De acuerdo con las características de las metodologías antes citadas, se decide utilizar SXP para el desarrollo del portal web de la Facultad 1 por las siguientes razones:

1. Metodología de desarrollo de software de código abierto, diseñado para pequeños equipos organizados que quieren tomar una aproximación ágil del desarrollo.
2. Se valora la colaboración y el aporte de los implicados sobre los entregables y las formalidades innecesarias.
3. Permite detectar errores tempranos a través de un ciclo iterativo.
4. Por ser una metodología ágil, tiene un enfoque centrado al cliente y con iteraciones cortas.

Después de determinar la metodología que guiará el proceso de desarrollo del software es necesario realizar un estudio de las herramientas a utilizar para el desarrollo de la propuesta de solución.

1.5 Sistemas de gestión de contenidos

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

Funcionalidades típicas de un CMS:

- ❖ Administración de la estructura del portal: módulos, menús, diseño, configuración general, etc.
- ❖ Administración del contenido: distintos tipos de contenidos, gestión y publicación de contenidos, etc.
- ❖ Administración de usuarios: políticas de gestión de usuarios y de acceso a los contenidos mediante roles y permisos, etc.
- ❖ Informes y gestión del portal: errores, estadísticas de acceso, etc.

De los CMS genéricos, algunos de los más utilizados en la actualidad son TYPO3, Joomla y Drupal. Todos ellos fueron publicados a principios de este siglo y tienen en común que están desarrollados en PHP (*Hypertext Pre-processor*) y MySQL y que se distribuyen como software libre. Cada uno de ellos cuenta con su propia comunidad de usuarios y desarrolladores que contribuyen al desarrollo del proyecto, ya sea trabajando en la mejora del software o aportando nuevos módulos para incrementar o mejorar sus funcionalidades. (Gil, 2012)

Drupal

Drupal es un Sistema de gestión de contenidos (CMS) que se distribuye como software libre bajo la licencia GNU GPL (*Licencia Pública General*) versión 2 o superior. Drupal puede ser modificado y distribuido libremente, pero siempre se debe hacer bajo la misma licencia.

El software está desarrollado con el lenguaje de programación PHP y utiliza una base de datos MySQL. Está maquetado con hojas de estilo CSS (*Cascading Style Sheets*), con lo que es posible construir sitios web totalmente accesibles. Hoy día casi cualquier proveedor de alojamiento (*hosting*) dispone de las características mínimas requeridas por Drupal para su instalación y correcto funcionamiento, aunque es recomendable consultar al proveedor previamente. (Gil, 2012)

Funcionalidades

- **Personalización:** Un robusto entorno de personalización está en el núcleo de Drupal. Tanto el contenido y la presentación puede ser individualizado basado en preferencias definidas por el usuario.
- **Sistema basado en permiso por roles:** Los administradores de Drupal no tienen que tediosamente configurar permisos para cada usuario. En lugar de ello, pueden asignar permisos a grupos de usuarios y luego especificar los roles y los permisos para las funciones.
- **Autenticación de usuario:** Los usuarios pueden registrar y autenticar localmente o mediante una fuente externa de autenticación, como *Jabber*, *Blogger*, *LiveJournal* u otro sitio de Drupal. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP (*Lightweight Directory Access Protocol*, en español Protocolo Ligero de Acceso a Directorios).
- **Encuestas:** Drupal viene con un módulo de encuesta que permite a los administradores y / o usuarios crear encuestas y mostrar información sobre diversas páginas.

Ventajas

- Totalmente configurable, de tal manera que el administrador de un sitio puede activar o desactivar diferentes características y establecer configuraciones que cambien su aspecto y funcionalidad.
- Sistema de privilegios que hacen posible que diferentes tipos de usuario – por ejemplo, miembros, compañeros y administrativos puedan ver y hacer cosas diferentes en el sitio.

- Está diseñado para ser fácilmente extensible a través de “módulos” bloques de código que proveen funcionalidad extra o mejoras. Algunos vienen con todas las instalaciones de Drupal (módulos del “núcleo”), mientras que otros pueden ser instalados y descargados individualmente del sitio web de Drupal (módulos “contribuidos”).
- El aspecto básico de un sitio Drupal puede ser cambiado usando diferentes “temas”. Al igual que con los módulos, existen temas del núcleo y temas contribuidos.
- Soporta varios lenguajes de base de datos (PostgreSQL, MySQL).
- Trabaja sobre servidores Apache e *Internet Information Server*.
- Cuenta con una gran comunidad de usuarios y desarrolladores.
- Tiene un potente motor de plantilla.
- Grandes capacidades SEO (*Optimizador de motores de búsqueda*,): Drupal fue diseñado desde la base para ser amigable con los motores de búsqueda.
- Estabilidad.
- Amigable con el desarrollador.

Desventajas

- Gran curva de aprendizaje.
- Necesita invertir tiempo para desarrollar todas sus posibilidades.

Joomla

Es un sistema de gestión de contenidos gratuito para la creación de sitios web. Es un proyecto de software libre que, al igual que la mayoría de estos proyectos, está en constante movimiento. Para desarrollar sitios sobre Joomla es preciso poseer conocimientos sobre diseño web y programación, pero para administrar sitios web basados en Joomla, solo se necesita una mínima capacitación sobre el manejo del programa. (Graf, 2011)

Características

- Organización del sitio web.
- Publicación de contenidos.
- Escalabilidad e implementación de nuevas funcionalidades.
- Administración de usuario.

- Diseño y aspecto estético del sitio.
- Disposición de módulos modificable.
- Encuestas.
- Estadísticas de visitas. (Ramírez y Sanfrán, 2010)

Ventajas

- Amigable con el usuario.
- Fuerte comunidad de desarrolladores.
- Variedad de extensiones.
- Grandes capacidades de gestión de contenidos.
- Fácil de instalar y configurar con sencillos clic.

Desventajas

- Necesita de una línea de aprendizaje bien definida.
- Falta de capacidades *SEO* (Optimizador de motores de búsqueda, *SEO* por sus siglas en inglés, de *Search Engine Optimization*).
- Soporte *ACL* (Listas de Control de Acceso) limitado.
- Permite crear solo un sitio por instalación.
- No se le pueden integrar otros *script* al sitio.
- Los módulos adicionales no son gratis.
- Solo conoce un lenguaje de base de datos (MySQL).
- Solo trabaja sobre servidores Apache.

WordPress

A inicios de su surgimiento estuvo enfocado a la creación de *blogs* (sitios web periódicamente actualizados), mientras que, hoy día es usado para el desarrollo de una multitud de sitios web. Está desarrollado en PHP y MySQL, bajo la licencia GPL (Licencia Pública General) y código modificable. Entre sus principales características se encuentra su licencia, el extenso repositorio de *plugins* y su facilidad de uso. Su estructura está basada en la arquitectura tres capas, compuesto por las capas de la base de datos, la base de programación y la base de diseño.

Otro motivo al que se le atribuye su éxito es la enorme comunidad de desarrolladores y diseñadores que atrae. WordPress presta especial atención a la estética, estándares web y usabilidad, sin embargo posee vulnerabilidades con las inyecciones SQL. (Sánchez, 2010)

Funcionalidades

- Garantiza la posibilidad de actualizar automáticamente el sistema.
- Permite ordenar artículos y páginas estáticas en categorías, subcategorías y etiquetas.
- Permite comentarios y herramientas de comunicación entre blogs.
- Permite subir y gestionar adjuntos y archivos multimedia.

Desventajas

- Seguridad: Como es el CMS líder, con millones de actualizaciones, WordPress es a menudo la diana de *hackers*.
- Incompatibilidad con *plugins* antiguos.
- Capacidad de gestión de contenidos limitada

Fundamentación del sistema de gestión de contenidos utilizado

Después de haber realizado un estudio de los Sistemas de Gestión de Contenidos más utilizados por las comunidades de desarrollo de portales web a nivel mundial, se decide utilizar Drupal como CMS para el desarrollo del portal web de la Facultad 1. Drupal presenta una gran flexibilidad para el desarrollo de aplicaciones web, soporta amplia variedad de estructuras web, es ampliamente adaptable a cualquier necesidad, ofrece gran seguridad y posee entre otras de sus ventajas, un robusto sistema de ayuda online.

Es importante destacar que el desarrollo de portales web en la UCI en su mayoría se encuentra guiado por el uso de Drupal, por lo que el conocimiento y la cantidad de documentación del mismo son superiores con respecto a otros CMS. Además, al contar con una amplia comunidad de desarrolladores hace que el trabajo esté sometido a una menor probabilidad de ocurrencia de errores.

1.6 Sistema gestor de bases de datos

PostgreSQL

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

PostgreSQL utiliza el modelo cliente-servidor y usa multiprocesos en vez de multihilo para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando. (Martínez, 2010)

Características fundamentales de PostgreSQL

- Es una base de datos 100% ACID.
- Juegos de caracteres internacionales.
- Múltiples métodos de autenticación.
- Actualización in-situ integrada (pg_upgrade).
- Completa documentación.
- Licencia BSB.
- Multiplataforma.

Algunos límites de PostgreSQL son:

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Figura 4. Límites de PostgreSQL

MySQL

Es un software libre, licenciado bajo la licencia GPL de GNU. Es un sistema de administración de bases de datos para bases de datos relacionales. Su diseño multihilo le permite soportar una gran carga de

forma muy eficiente. Fue escrito en C y C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más empleados como PHP, Perl y Java, y su adaptación a distintos sistemas operativos. Es un sistema gestor de bases de datos muy rápido, fiable y fácil de usar.

MySQL presenta como desventaja que una vez eliminada una tabla, no se eliminan automáticamente los privilegios concebidos sobre la misma. Debe usarse explícitamente un comando *REVOKE* para quitar los privilegios de una tabla, acción que resulta tediosa para los usuarios con permiso de administración para la base de datos. (Oracle corporation, 2014)

Fundamentación del sistema gestor de bases de datos utilizado

Teniendo en cuenta los elementos de los sistemas gestores de bases de datos antes tratados y las características de la aplicación a desarrollar se decide emplear del SGBD PostgreSQL en su versión 9.2. PostgreSQL es el sistema de gestión de bases de datos de código abierto más potente del mercado y provee al sistema de una mayor estabilidad.

1.7 Tendencias y tecnologías de desarrollo

1.7.1 Herramienta CASE

Visual Paradigm es una herramienta UML (Lenguaje Unificado de Modelado) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar, actualizar y compatible entre ediciones. (Betancourt, 2010)

1.7.2 Lenguajes de programación

1.7.2.1 Lenguajes del lado del cliente

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se

pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (Pérez, 2014)

Protocolo de transferencia de hipertexto (HTML)

Una definición simple de HTML indica que “HTML es lo que se utiliza para crear todas las páginas web de Internet”. Más concretamente, HTML es el lenguaje con que se “escriben” las páginas web. El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado W3C (*World Wide Web Consortium*). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de la misma manera en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como “un lenguaje reconocido universalmente y que permite publicar información de forma global”. Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas: buscadores, tiendas online, banca electrónica, etc. (Pérez, Javier Eguiluz, 2014b)

Cascading Style Sheets (CSS)

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML (*eXtensible HyperText Markup Language*). CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas. El lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc. (Pérez, Javier Eguiluz, 2014a)

1.7.2.2 Lenguaje del lado del servidor

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Se conecta a servidores de bases de datos tales como MySQL, Oracle, Informix, PostgreSQL, entre otros. Debido a su amplia distribución, PHP está perfectamente soportado por una gran comunidad de desarrolladores. Es un lenguaje de programación de alto nivel, totalmente compatible con los modernos métodos orientados a objetos, prácticas y principios. (BAKKEN et al., 2003)

1.7.3 AJAX

AJAX por sus siglas en inglés de *Asynchoronous JavaScript And XML*. AJAX no es un lenguaje de programación sino un conjunto de tecnologías (*HTML-JavaScript-CSS-DHTML-PHP*) que permite la creación de páginas web más interactivas. La característica fundamental de AJAX es permitir actualizar parte de la página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar se puede enviar información hacia el servidor. (Pérez, Javier Eguiluz, 2014a)

1.7.4 Entorno integrado de desarrollo

NetBeans

Es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como para aplicaciones web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP, Groovy, C/C++ y HTML5. Además posee una muy importante característica, es multiplataforma. (Goovy, 2014)

Características principales

- ✓ Suele dar soporte a casi todas las novedades en el lenguaje Java.
- ✓ Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos marcos de trabajo.
- ✓ Buen editor de código multilenguaje, con habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de la ubicación de la clase actual.
- ✓ Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistente de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario.
- ✓ Herramienta para el depurado de errores.
- ✓ Optimización de código.
- ✓ Acceso a bases de datos.

Geany

Es un editor de texto usando el kit de herramientas GTK2 con las características básicas de un entorno de desarrollo integrado. Fue desarrollado para proporcionar un pequeño y rápido IDE, que tiene sólo unas pocas dependencias de otros paquetes. Es compatible con muchos tipos de archivos. Está disponible para distintos sistemas operativos, como GNU/Linux, *Mac OS X*, BSD, Solaris y *Microsoft Windows*. Es

distribuido como software libre bajo la Licencia Pública General de GNU. Algunas de sus principales funcionalidades son:

- Autocompletado.
- Código de navegación.
- Cierre automático de etiquetas XML y HTML.
- Muchos tipos de archivos soportados tales como C, Java, PHP, Python, Perl y Pascal.
- Fácil gestión de proyectos.
- Soporte para *plugins*.

Notepad ++

Notepad++ es un editor de texto de código fuente libre con soporte para varios lenguajes de programación. Se distribuye bajo los términos de la Licencia Pública General de GNU. Aunque su soporte nativo es *Microsoft Windows* puede utilizarse en GNU/Linux mediante *Wine*.

Basado en un potente componente de edición llamado *Scintilla*, *Notepad++* está escrito en C++ y usa directamente la API de Win32 y STL, lo que asegura una gran velocidad de ejecución y un menor tamaño del programa. Mediante la optimización de tantas rutinas como sea posible sin perder la interfaz amigable con el usuario. Algunas de sus principales características son:

- Resaltado de sintaxis y envoltura de sintaxis
- WYSIWYG
- Resaltado de sintaxis definido por el usuario
- Auto-completado
- Multi-documento
- Multi-vista
- Soporte para la búsqueda y substitución de expresiones regulares
- Soporte completo para arrastrar y soltar
- Posición dinámica de vistas
- Auto-detección de estado del fichero
- Aumentar y reducir el zoom
- Entorno multi-lenguaje soportado
- Favoritos

- Resaltado de paréntesis y sangría (Fabri, 2012)

Fundamentación del IDE de desarrollo seleccionado

Teniendo en cuenta los elementos antes citados y la ventaja de que no se requiere un potente Entorno de Desarrollo Integrado (IDE) para el desarrollo del sistema, se propone la utilización de la herramienta *Notepad++* en su versión 6.5.5.

1.7.5 Servidor de aplicaciones web

Un servidor web es un programa que permite crear un servidor HTTP en el ordenador. Con el modelo cliente-servidor y el protocolo de transferencia de hipertexto de Internet, proporciona a los usuarios archivos que conforman las páginas web (figuras, formularios, botones, etc.). Está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. (Márquez, Sampedro y Vargas, 2012)

Servidor web Apache

Es un servidor de red para el protocolo HTTP, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Apache se distribuye como software libre de código abierto, modular multiplataforma, extensible, popular (fácil de conseguir ayuda/soporte) y gratuito. Su licencia es una descendiente del tipo BSD que permite el uso comercial y no comercial de Apache. (Márquez, Sampedro y Vargas 2012)

Características de Apache

- Apache es un servidor altamente configurable. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otro elemento importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Permite la creación de ficheros de logs a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

Servidor web NGINX

Es un servidor HTTP de código abierto, distribuido bajo la licencia LSB y de proxy inverso de alto rendimiento, además funciona como servidor proxy para IMAP/POP3/SMTP. Fue desarrollado por Igor Sysoev para uno de los sitios más visitados en Rusia, *Rambler*. NGINX es muy conocido por su estabilidad, sus características, configuración simple y su bajo consumo de recursos, lo último como consecuencia de manejar requerimientos basados en eventos, a diferencia de Apache que lo hace basado en procesos y que permite asegurar un funcionamiento óptimo bajo mucha carga. (Lucas, 2011)

Algunas de las principales características de NGINX son:

- Capacidad de manejar más de 10 000 conexiones simultáneas con bajo uso de memoria.
- Balanceo de carga.
- Tolerancia a fallos.
- Soporte TLS/SSL.
- Autenticación de acceso.
- Compresión y descompresión gzip.
- Reescritura de URL.
- Limitaciones de conexiones concurrentes y respuestas.
- Manejo de ancho de banda.
- Proxy IMAP, POP3, SMTP.
- Procesamiento de datos XSLT.

Fundamentación del Servidor Web seleccionado

Después del análisis detallado de las características de los servidores web antes mencionados se determina emplear en el desarrollo de la aplicación el servidor NGINX, teniendo en cuenta como elementos primordiales el número de conexiones que puede manejar, el consumo de recursos, el balanceo de la carga y el manejo de ancho de banda.

1.8 Herramientas de pruebas de software

Apache JMeter

JMeter es una aplicación de escritorio de código abierto para realizar pruebas funcionales de software y medir el rendimiento. Inicialmente se diseñó para pruebas de aplicaciones web, aunque en versiones posteriores, ha aumentado su funcionalidad a otro tipo de pruebas.

Funcionalidad

- Probar el rendimiento de recursos.

JMeter se puede utilizar para probar el rendimiento tanto de recursos estáticos como dinámicos: archivos, *Servlets*, *scripts* de *Perl*, objetos de Java, bases de datos y consultas.

- Pruebas de rendimiento de servidores.

JMeter permite realizar pruebas de rendimiento de distintos tipos de servidores: HTTP, HTTPS, SOAP, bases de datos mediante JDBC, JMS, POP3 e IMAP y servidores FTP.

- Construcción y ejecución de un plan de pruebas

Un plan de pruebas describe los pasos que ejecutará JMeter para realizar las diferentes pruebas.

- Simulación

JMeter permite simular una carga pesada en un servidor, una red o un objeto, y así poder probar su resistencia o analizar su rendimiento global en distintos tipos de carga. Además se puede realizar un análisis gráfico de rendimiento o probar el comportamiento de un objeto, secuencia de comandos o servidor, bajo carga simultánea. (bilib, 2012)

1.9 Conclusiones parciales

- ✓ El análisis crítico de las teorías asociadas a la investigación arrojó como resultado que, existe una correspondencia entre las principales potencialidades que tienen los portales web estudiados y las dificultades que presenta el portal web de la Facultad 1.
- ✓ El estudio de las tecnologías adecuadas para el desarrollo de portales web determinó que para el desarrollo de la investigación se debe emplear la metodología de desarrollo SXP, los lenguajes de programación CSS, HTML, JavaScript y PHP, el gestor de bases de datos PostgreSQL, el IDE de desarrollo *Notepad++* y como herramienta de modelado *Visual Paradigm*.
- ✓ El estudio de los sistemas de gestión de contenidos más utilizados en la actualidad arrojó que para el desarrollo del portal web se debe utilizar el CMS Drupal.

Capítulo 2: Características y Arquitectura del Portal Web de la Facultad 1.

2.1 Introducción del capítulo

En el presente capítulo se caracteriza la propuesta de solución, a partir de su funcionamiento, teniendo en cuenta los estilos arquitectónicos y los patrones de diseños aplicables al sistema de gestión de contenidos seleccionado: Drupal. Se muestra a través del modelo de dominio, los principales objetos que se manejan en el contexto del sistema y se identifican los requisitos funcionales y no funcionales que deben estar presentes en la propuesta de solución.

2.2 Descripción de la propuesta de solución

De acuerdo con los resultados arrojados por el estudio de sistemas homólogos, el portal web cuenta con varias secciones, en las que se presentan los contenidos de forma organizada mediante elementos de menús. En la portada se muestra una vista de la noticia más relevante, acompañada de un bloque con las últimas noticias publicadas. Se presentan además, secciones como “**Menú del comedor**”, “**Columna temática**”, “**Cumples de hoy**” y “**Avisos**”, presentes en la versión actual del portal. Se incorporan las secciones “**El Decano informa**”, espacio en el cual el usuario puede encontrar orientaciones asociadas a eventos y actividades de diversa índole, así como las secciones “**Receta de la abuela**” y “**A donde voy**”.

Se habilita la opción de comentarios, garantizando con ello la interoperabilidad entre los usuarios y la aplicación, aprovechando este elemento para la retroalimentación del portal. Los usuarios podrán suscribirse al boletín y participar en las encuestas publicadas.

2.2.1 Usuarios y permisos

Para garantizar la seguridad e integridad de la información se establece un mecanismo de control de acceso a datos, donde los privilegios de cada usuario sobre los contenidos del portal, se encuentran en correspondencia con los roles y permisos asignados por el usuario administrador.

- **Usuario anónimo:** Usuario que navega por el portal sin estar autenticado, solo tiene permiso de lectura en el portal.
- **Usuario autenticado:** Tendrá permiso para la publicación de comentarios y para la participación en encuestas publicadas.

- **Administrador:** Es el responsable de definir los permisos de acceso a los usuarios del sistema. Su principal tarea es mantener el portal actualizado, es decir, gestionar los servicios que va a brindar.
- **Colaborador:** Usuario con permiso para la creación de artículos, en correspondencia con el nivel de accesibilidad asignado.
- **Editor:** Es el encargado de la moderación y aprobación de los contenidos.

En caso de ser necesario, el usuario administrador podrá crear nuevos roles y asignarle permisos sobre los contenidos del portal.

2.3 Levantamiento de requisitos

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema. (Pressman, 2001)

Se recomienda establecer diferentes niveles de especificación del sistema, los que deben estar en correspondencia con los lectores a los que va dirigido, para una mayor comprensión. En correspondencia a lo antes planteado, en la presente investigación se tratarán en la descripción de requisitos solo dos niveles, requisitos funcionales y no funcionales, detallados a continuación.

2.3.1 Requisitos funcionales del sistema

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle de los posibles usuarios del sistema y del enfoque general tomado por la organización al redactar requerimientos. (Pressman, 2001)

En el proceso de levantamiento de requisitos para el desarrollo del portal web de la Facultad 1, fueron detectados un total de 71 requisitos funcionales, 26 de ellos de prioridad muy alta para el cliente y 20 de prioridad alta. A continuación se presenta un resumen de la etapa de captura de requisitos.

Listado de Requisitos Funcionales (RF)

<p>RF1 Autenticar usuario</p> <p>RF2 Insertar usuario</p> <p>RF3 Editar usuario</p> <p>RF4 Eliminar usuario</p> <p>RF5 Listar usuarios</p> <p>RF6 Crear página</p> <p>RF7 Editar página</p> <p>RF8 Mostrar página</p> <p>RF9 Eliminar página</p> <p>RF10 Listar páginas</p> <p>RF11 Crear contenido artículo</p> <p>RF12 Editar contenido artículo</p> <p>RF13 Mostrar contenido artículo</p> <p>RF14 Eliminar contenido artículo</p> <p>RF15 Listar contenido artículo</p> <p>RF16 Crear rol</p> <p>RF17 Editar rol</p> <p>RF18 Mostrar rol</p> <p>RF19 Eliminar rol</p> <p>RF20 Listar roles</p> <p>RF21 Adaptar visibilidad del portal para las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.</p> <p>RF22 Crear tipo de contenido avisos</p> <p>RF23 Editar tipo de contenido avisos</p> <p>RF24 Mostrar tipo de contenido avisos</p> <p>RF25 Eliminar tipo de contenido avisos</p> <p>RF26 Listar tipo de contenido avisos</p> <p>RF27 Crear contenido galería de imágenes</p> <p>RF28 Editar contenido galería de imágenes</p> <p>RF29 Mostrar contenido galería de imágenes</p>	<p>RF34 Mostrar contenido video</p> <p>RF35 Eliminar contenido video</p> <p>RF36 Listar contenido video</p> <p>RF37 Crear banner promocional</p> <p>RF38 Editar banner promocional</p> <p>RF39 Mostrar banner promocional</p> <p>RF40 Eliminar banner promocional</p> <p>RF41 Listar contenido banner promocional</p> <p>RF42 Crear tipo de contenido menú del día</p> <p>RF43 Editar tipo de contenido menú del día</p> <p>RF44 Mostrar tipo de contenido menú del día</p> <p>RF45 Eliminar tipo de contenido menú del día</p> <p>RF46 Listar tipo de contenido menú del día</p> <p>RF47 Realizar búsquedas simples</p> <p>RF48 Realizar búsqueda avanzada</p> <p>RF49 Realizar comentarios</p> <p>RF50 Crear tipo de contenido encuesta</p> <p>RF51 Editar tipo de contenido encuesta</p> <p>RF52 Mostrar tipo de contenido encuesta</p> <p>RF53 Eliminar tipo de contenido encuesta</p> <p>RF54 Listar tipo de contenido encuesta</p> <p>RF55 Mostrar resultados de la encuesta</p> <p>RF56 Crear tipo de contenido efeméride</p> <p>RF57 Editar tipo de contenido efeméride</p> <p>RF58 Mostrar tipo de contenido efeméride</p> <p>RF59 Eliminar tipo de contenido efeméride</p> <p>RF60 Listar tipo de contenido efeméride</p> <p>RF61 Crear contenido boletín</p> <p>RF62 Editar contenido boletín</p> <p>RF63 Mostrar contenido boletín</p> <p>RF64 Eliminar contenido boletín</p> <p>RF65 Listar tipo de contenido boletín</p>
---	--

RF30 Eliminar contenido Galería de imágenes	RF66 Suscribirse usuario al boletín
RF31 Listar contenido Galería de imágenes	RF67 Mostrar enlaces a sitios de interés
RF32 Crear contenido video	RF68 Listar pie de página
RF33 Editar contenido video	RF69 Registrar votos de un contenido en la web
	RF70 Mostrar estadísticas
	RF71 Mostrar paginado

Tabla 1. Listado de Requisitos Funcionales (RF)

Prioridad para el cliente	Cantidad de Requisitos Funcionales
Muy Alta	26
Alta	20
Media	22
Baja	3
Total	71

Tabla 2. Resumen de la etapa de captura de los RF

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad y surgen de las necesidades del usuarios, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas software o hardware, o a factores externos como regulaciones de seguridad o legislaciones sobre privacidad. (Pressman, 2001)

➤ Apariencia o Interfaz externa

RNF 1: La interfaz gráfica de la aplicación debe concebirse con un ambiente amigable, sencillo y de navegación fácil para el usuario.

RNF 2: La interfaz contará con menús desplegables para agilizar y acelerar su utilización.

➤ Usabilidad

RNF 3: El Sistema debe permitir al usuario una fácil interacción con el mismo.

RNF 4: El sistema debe tener buena visibilidad en los principales navegadores web.

➤ **Confiabilidad**

RNF 5: En caso de que el sistema presente alguna falla, los errores deben mostrar la menor cantidad de detalles posible, de forma tal, que se evite dar información que comprometa la seguridad e integridad del sistema. Sólo se mostrarán detalles ampliados del error a usuarios con privilegios de administración.

RNF 6: El sistema establecerá mecanismos para garantizar la confiabilidad e integridad de la información ante posibles accesos no autorizados.

RNF 7: Los ficheros públicos estarán separados de los ficheros privados, creando una ruta del sistema de archivos donde se almacenarán los ficheros privados.

➤ **Seguridad**

RNF 8: Se podrá acceder a las páginas de administración del sitio web a través del protocolo HTTPS, y a las páginas de usuarios por el protocolo HTTP.

RNF 9: Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando, usuario, contraseña y niveles de accesos para cada usuario, de manera que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

RNF 10: Se usarán mecanismos de encriptación de la contraseña que por cuestiones de seguridad no deben estar en texto plano, por lo que las contraseñas en la base de datos se almacenarán en forma encriptada.

➤ **Soporte**

Para el correcto funcionamiento del portal web es necesaria la utilización de las siguientes herramientas:

RNF 11: Servidor web NGINX.

RNF 12: Gestor de base de datos PostgreSQL en su versión 9.2 o superior.

RNF 13: Lenguaje de programación PHP 5.3.4 y como sistema de gestión de contenidos Drupal 7.26.

RNF 14: Navegador web: Mozilla Firefox 2.0 o superior.

➤ **Dispositivos móviles**

RNF 15: El Sistema podrá ser visualizado en dispositivos para las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.

➤ **Requisitos de licencia**

RNF 16: Uso de licencia GNU/GPL versión 2 para el CMS Drupal.

RNF 17: Uso de licencia BSD de PostgreSQL.

RNF 18: Uso de licencia PHP *License*.

➤ **Hardware**

RNF 22: El servidor de aplicaciones web y de base de Datos deben poseer como mínimo un CPU Core2Duo a 2.20 GHz.

RNF 23: El servidor de base de Datos debe poseer una capacidad mínima de 320 GB.

RNF 24: El servidor de aplicaciones web debe poseer una capacidad mínima de 120 GB.

RNF 25: Los servidores proxy, web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.

2.4 Historias de usuarios

Unos de los artefactos más importantes generados por la metodología de desarrollo de software SXP son las historias de usuario, las cuales son utilizadas para la especificación de requisitos funcionales del sistema. Durante el análisis en la etapa de planificación – definición fueron definidas 37 historias de usuarios, en correspondencia con las necesidades del cliente. A continuación se muestra la descripción de las historias de usuarios más importantes.

Historia de usuario	
Número: 1	Nombre Historia de Usuario: <i>Autenticar usuario</i>
Modificación de Historia de Usuario Número: <i>ninguna</i>	
Usuario: <i>Wendy Rodriguez Muñoz</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Muy alta</i>	Puntos Estimados: 2/5
Riesgo en Desarrollo: <i>Medio</i>	Puntos Reales: 2/5
Descripción: <i>El sistema permitirá autenticarse para acceder a las funcionalidades del sistema, según el nivel de acceso establecido para cada usuario.</i>	
Observaciones: <i>Para acceder a la aplicación el usuario debe introducir los siguientes datos:</i> <ul style="list-style-type: none">• <i>Usuario (obligatorio, campo de texto)</i>• <i>Contraseña (obligatorio, campo de texto)</i>	

Tabla 3. Historia de usuario Autenticar usuario

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: <i>Gestionar usuario</i>
Modificación de Historia de Usuario Número: <i>ninguno</i>	
Usuario: <i>Wendy Rodriguez Muñoz</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Muy Alta</i>	Puntos Estimados: 3/5
Riesgo en Desarrollo: <i>Medio</i>	Puntos Reales: 3/5
<p>Descripción: <i>Permite adicionar, eliminar, editar y mostrar la información referente a un usuario.</i></p> <p>Observaciones: <i>En el caso insertar usuario el sistema le permitirá a los usuarios con el rol Administrador agregar nuevos usuarios al sistema y asignar privilegios, según el nivel de accesibilidad que tenga su rol. Para ello se debe introducir los siguientes datos:</i></p> <ul style="list-style-type: none"> • <i>Nombre de usuario (obligatorio, campo de texto)</i> • <i>Dirección de correo electrónico (obligatorio, campo de texto)</i> • <i>Contraseña (obligatorio, campo de texto)</i> • <i>Confirmar contraseña (obligatorio, campo de texto)</i> • <i>Estado (obligatorio, campo de selección)</i> • <i>Roles (obligatorio, campo de selección)</i> • <i>Opciones de idioma (obligatorio, campo de selección)</i> <p><i>En el caso eliminar usuario el sistema permitirá a los usuarios con el rol de administrador eliminar un usuario en el sistema. El sistema mostrará las opciones:</i></p> <ul style="list-style-type: none"> • <i>Eliminar (se elimina el contenido de la aplicación)</i> • <i>Cancelar (se deshace la operación)</i> <p><i>En el caso editar usuario, el sistema permitirá a los usuarios con el rol Administrador modificar la información relacionada con el usuario en el sistema. El sistema mostrará los datos:</i></p> <ul style="list-style-type: none"> • <i>Nombre de usuario (obligatorio, campo de texto)</i> • <i>Dirección de correo electrónico (obligatorio, campo de texto)</i> • <i>Contraseña (obligatorio, campo de texto)</i> • <i>Confirmar contraseña (obligatorio, campo de texto)</i> • <i>Estado (obligatorio, campo de selección)</i> • <i>Roles (obligatorio, campo de selección)</i> • <i>Opciones de idioma (obligatorio, campo de selección)</i> 	

Tabla 4. Historia de usuario Gestionar usuario

2.5 Descripción de la Arquitectura de software y los patrones de diseño

La arquitectura es un elemento primordial en el éxito o el fracaso de un proyecto, proporciona una visión global del sistema a construir, es una vista estructural de alto nivel que define el estilo arquitectónico o su combinación para la solución de un determinado problema.

2.5.1 Arquitectura de software

Al ser utilizado el CMS Drupal para el desarrollo del portal web, la arquitectura y los patrones son heredados del mismo. La reconocida flexibilidad y facilidad de los portales web creados con el CMS Drupal, se basa en la abstracción y organización en capas que se aplica al contenido. La arquitectura de Drupal es un tema que genera hoy día una enorme polémica, algunos desarrolladores afirman la idea de que el CMS presenta una arquitectura modelo-vista-controlador mientras otros establecen la hipótesis de arquitectura modular.

La autora de la presente investigación defiende la idea de que Drupal presenta una arquitectura n-capas, siendo lo primero que, esta arquitectura es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo mediante capas y permitir intercambiar porciones de la aplicación sin necesidad de modificarla completamente, y el segundo, 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: datos(nodos), módulos, bloques y menús, permisos de usuarios y plantillas.

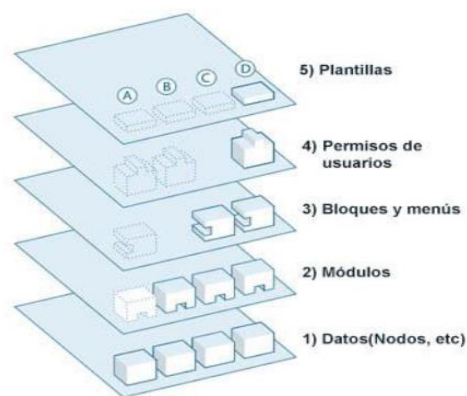


Figura 5. Estructura de la aplicación

Datos: Son los elementos básicos en que Drupal almacena la información, los contenidos. Así, a medida que el sitio web crece, los nodos aumentan proporcionalmente, formando un “depósito de nodos” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal está formada por este “depósito de nodos”.

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. Esta es conocida como la segunda capa de la arquitectura del CMS.

Bloques y Menús: La siguiente capa se encuentra estructurada por los bloques y menús. Estos permiten estructurar y organizar los contenidos de las páginas 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.

Permisos de usuarios: Actualmente la mayor capa de los sitios web son multiusuarios, por lo que la seguridad y el control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta funcionalidad Drupal dispone de un registro de usuarios y de roles que especifican las acciones que pueden desarrollar los mismos y los contenidos a los que tendrán acceso. 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.

Plantillas: Esta es la última capa y es la que establece la apariencia gráfica o estilo de la información que se les muestra a los usuarios. 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 cambiar la apariencia. (Gil, 2012)

2.5.2 Patrones de diseño

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos. El concepto de patrones de diseño fue el resultado de un trabajo realizado por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, publicado en 1995 en un libro titulado: “Patrones de diseño: Elementos de Software orientado a objetos reutilizables” en el que se esbozaban 23 patrones de diseño.

Drupal utiliza patrones de diseño propiamente orientados a objetos como los patrones *Gang of Four (GoF)* y *Grasp (General Responsibility Assignment Software Patterns)*, permitiendo la construcción de un diseño elegante y robusto. Los patrones *GoF* son clasificados según el propósito para el cual han sido definidos. Los patrones generales de software para asignación de responsabilidades *GRASP* proporcionan principios generales para asignar responsabilidades a las clases. (Ibáñez, 2014) Entre los patrones usados por Drupal se pueden citar los siguientes:

Patrones GoF

Patrones de creación: Solucionan problemas de creación de instancias. Nos ayudan a encapsular y a abstraer dicha creación.

- **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 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.

En la aplicación puede evidenciarse este patrón cuando el editor crea los tipos de contenido Artículos, Avisos y Efemérides. Primeramente verifica que no existan en la base de datos y luego los crea.

Patrones de comportamiento: Solucionan problemas respecto a la interacción y a las responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

- **Cadena de responsabilidades:** Evita acoplar el emisor de una petición a su receptor, dando a más de un objeto la capacidad de responder a la petición. El sistema de menús de Drupal sigue el patrón de cadenas de responsabilidades. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados, y qué función se llama para hacer el trabajo.
- **Patrón de reflexión:** Describe el comportamiento de todo sitio basado en Drupal. Gracias a este patrón el sistema se convierte en una aplicación muy adaptable a diferentes entornos puesto que permiten prácticamente la modificación de todo su comportamiento a través de los módulos instalables, sin necesidad de modificar el núcleo.

- **Observador:** Este patrón es usado en Drupal para controlar la forma en que se utilizan 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.

EL uso del patrón Observador se evidencia en la aplicación cuando se modifica el tipo de contenido “Artículos”, el sistema informa a todas sus dependencias de la modificación realizada, como por ejemplo la vista de los artículos con categoría “Cobertura”.

Patrones estructurales: Solucionan problemas de composición (agregación) de clases y objetos.

- **Puentes:** Desacopla una abstracción de su implementación, de modo que ambas puedan ser variadas de forma independiente, sin modificar una a la otra.

En 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 esté empleando en las bases de datos, garantizando que se pueda añadir soporte para diferentes bases de datos sin tener que modificar el código de un módulo.

En la aplicación puede evidenciarse el uso de este patrón, en la abstracción de los módulos con respecto al diseño, pues puede cambiarse la estructura de la aplicación web o los tipos de contenidos sin que para ello se afecte el módulo. Un ejemplo de ello es la eliminación del tipo de contenido “**Recetas de la abuela**”, al realizar esta operación no provoca que el módulo “CKEditor”, utilizado en el diseño de este tipo de contenido, cese su funcionamiento en otro tipo de contenido donde se emplee como “**Artículos**”.

- **Decorador:** Asigna responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Permite no tener que crear clases que hereden de la primera incorporando nuevas funcionalidades, sino otras que la implementan y se asocian a ella.

Este patrón se usa en Drupal de la siguiente manera: un nodo necesita tener la opción de adjuntar archivos 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.

Patrones GRASP

- **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- **Controlador:** Sirve como intermediario entre determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Un controlador es un objeto de interfaz no destinado al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación.

Un ejemplo de la utilización del patrón controlador es cuando el administrador elige la opción “Crear banner promocional”. Cuando el usuario rellena los campos y oprime el botón “Guardar”, está generando un evento sistémico que indica que el tipo de contenido ha sido creado.

2.6 Diagrama de clases del diseño web

Al ser Drupal el sistema de gestión de contenidos utilizado en el desarrollo de la propuesta de solución, es necesario analizar su estructura y funcionamiento, para tener el conocimiento necesario de cómo se confeccionan las clases del diseño web. Drupal depende de un conjunto de clases o módulos que le posibilita al usuario final interactuar con el software, estos se encuentran estructurados de acuerdo a sus funcionalidades.

Este CMS se encuentra estructurado mediante 4 paquetes principales: *Themes*, *Includes*, *Modules* y *Scripts*, que a su vez están conformados por diversos archivos. Con el fin de lograr una mejor comprensión de la aplicación, se modela a continuación el diagrama de clases del diseño web.

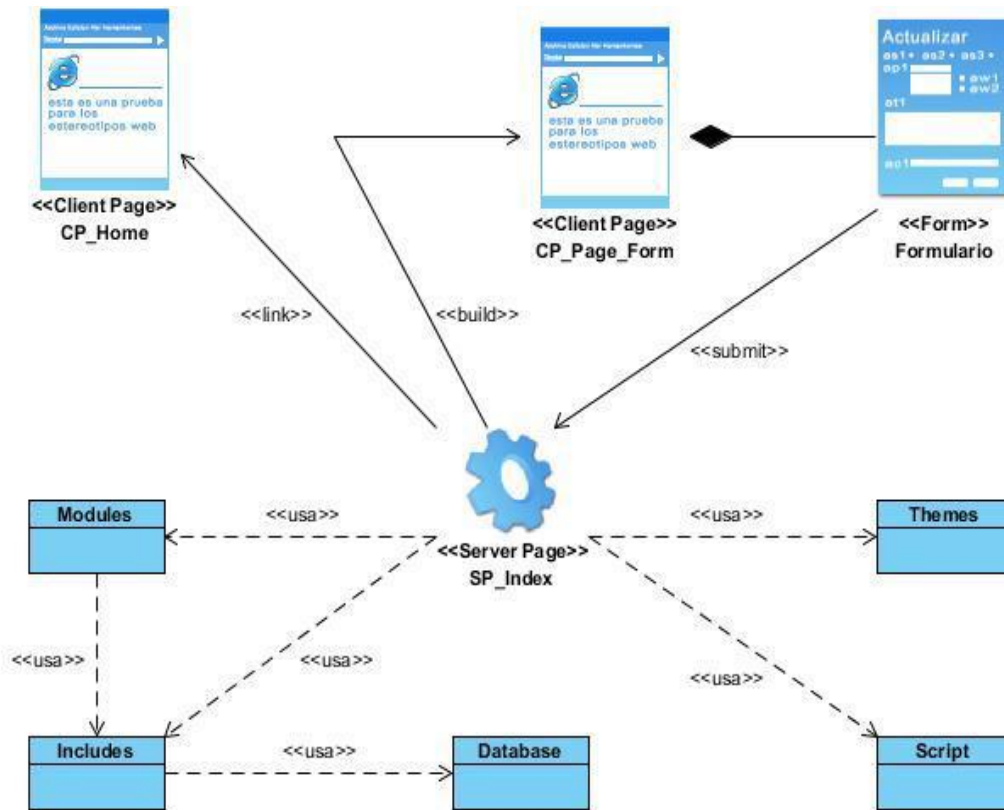


Figura 6. Diagrama de clases del diseño del CMS Drupal

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

Modules: contiene todos los módulos del núcleo que permiten la integración de disímiles funcionalidades. No es recomendable modificar directamente estos módulos o subir módulos adicionales a esta carpeta. Los módulos adicionales se deber guardar en la carpeta /sites/all/modules o /sites/default/modules, definida en el archivo de instalación del CMS.

Themes: Contiene los temas que vienen con la distribución del CMS. Al igual que los módulos no es recomendable modificar directamente estos temas o subir temas adicionales a esta carpeta. Los

módulos adicionales se deber guardar en la carpeta /sites/all/themes o /sites/default/themes, definida en el archivo de instalación del CMS.

Scripts: Contiene utilidades adicionales que no utiliza Drupal directamente, pero que se puede utilizar desde la línea de comandos del Shell.

Index: El archivo index.php es la puerta de entrada al sistema. Se ejecuta siempre que se cargue una página del portal web.

El CMS Drupal contiene una única página servidora, que a través de funciones o ganchos (*hook*) se conectan a la base de datos mediante *Includes*, para mostrar la información requerida.

Después de elaborado el diagrama general de clases del diseño por paquetes, es necesario modelar el diagrama de clases del diseño web para cada subpaquete. Entre las principales funcionalidades que debe tener implementadas el software se encuentran: gestionar contenido artículo, gestionar contenido aviso y gestionar contenido efeméride. Forman parte de la gestión de este tipo de contenido las funciones para insertar, modificar, mostrar y eliminar. A continuación se muestran los diagramas de clases del diseño web correspondientes a estas funcionalidades.

Diagrama de clases del diseño web del paquete Gestionar Artículo

Para insertar un artículo la página servidora SP_GestionarArticulo construye la página cliente CP_InsertarArticulo, la cual está compuesta por el formulario frm_InsertarArticulo que permite el ingreso de los datos requeridos en la inserción del artículo y que va a ser ejecutado por la página servidora. Para modificar un artículo la página servidora crea la página cliente CP_ModificarArticulo que contiene el formulario frm_ModificarArticulo donde se modifican los datos y que será posteriormente ejecutado por la página servidora SP_GestionarArticulo. Para acceder a los datos se hace a través de la clase controladora CC_GestionarArticulo. En el caso de eliminar un artículo se accede al formulario frm_EliminarArticulo contenido en la página cliente CP_EliminarArticulo construida por la página servidora. Para mostrar los artículos la página servidora construye la página cliente CP_MostrarArticulo que permite la visualización del contenido.

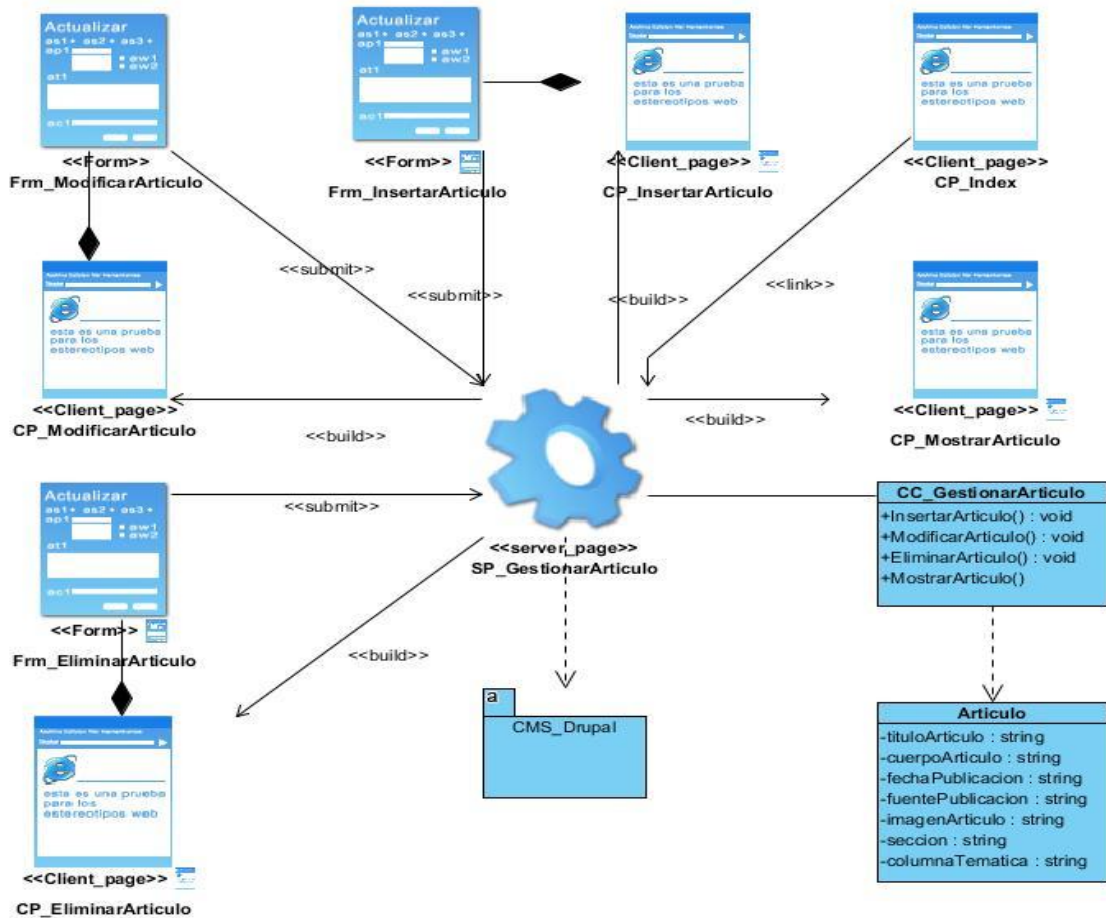


Figura 7. Paquete Gestionar Artículo

Diagrama de clases del diseño web del paquete Gestionar Efeméride

Para insertar una efeméride la página servidora SP_GestionarEfeméride construye la página cliente CP_InsertarEfeméride, la cual está compuesta por el formulario Frm_InsertarEfeméride que permite el ingreso de los datos requeridos en la inserción de la efeméride y que va a ser ejecutado por la página servidora. Para modificar una efeméride la página servidora crea la página cliente CP_ModificarEfeméride que contiene el formulario Frm_ModificarEfeméride donde se modifican los datos y que será posteriormente ejecutado por la página servidora SP_GestionarEfeméride. Para acceder a los datos se hace a través de la clase controladora CC_GestionarEfeméride. En el caso de eliminar una efeméride se accede al formulario Frm_EliminarArticulo contenido en la página cliente CP_EliminarEfeméride construida

por la página servidora. Para mostrar las efemérides la página servidora construye la página cliente CP_MostrarEfemeride que permite la visualización del contenido.

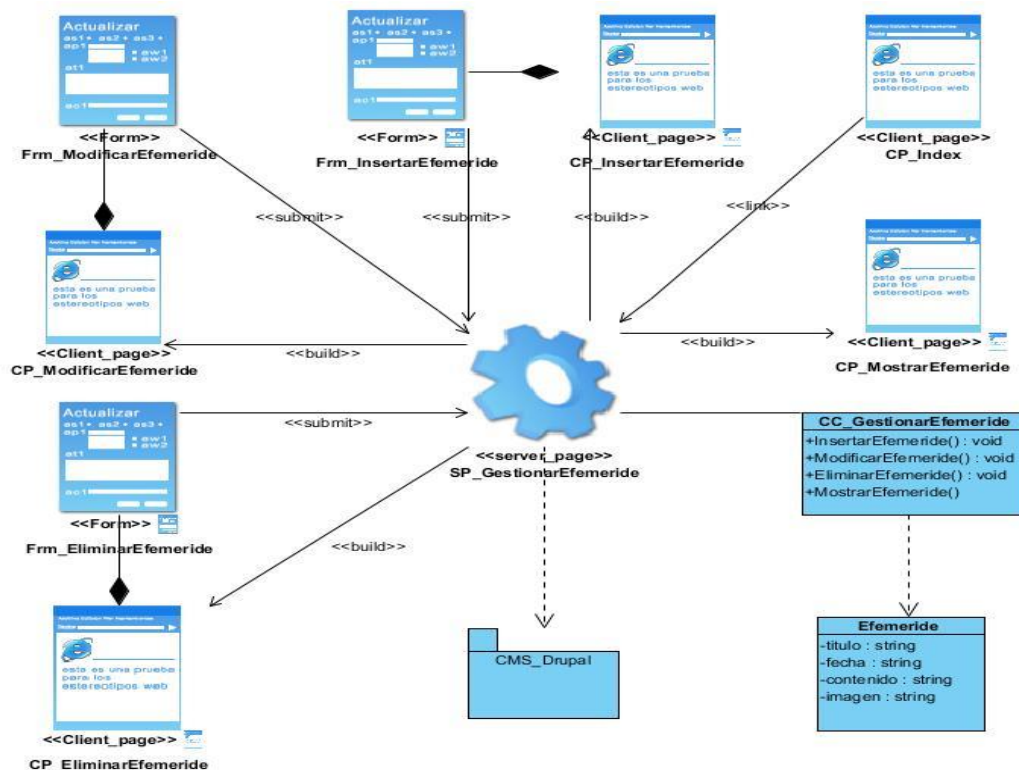


Figura 8. Paquete Gestionar Efemeride

Diagrama de clases del diseño web del paquete Gestionar Aviso

Para insertar un aviso la página servidora SP_GestionarAviso construye la página cliente CP_InsertarAviso, la cual está compuesta por el formulario Frm_InsertarAviso que permite el ingreso de los datos requeridos en la inserción del aviso y que va a ser ejecutado por la página servidora. Para modificar un aviso la página servidora crea la página cliente CP_ModificarAviso que contiene el formulario Frm_ModificarAviso donde se modifican los datos y que será posteriormente ejecutado por la página servidora SP_GestionarAviso. Para acceder a los datos se hace a través de la clase controladora CC_GestionarAviso. En el caso de eliminar un aviso se accede al formulario Frm_EliminarAviso contenido en la página cliente CP_EliminarAviso construida por la página servidora. Para mostrar los avisos la página servidora construye la página cliente CP_MostrarAviso que permite la visualización del contenido.

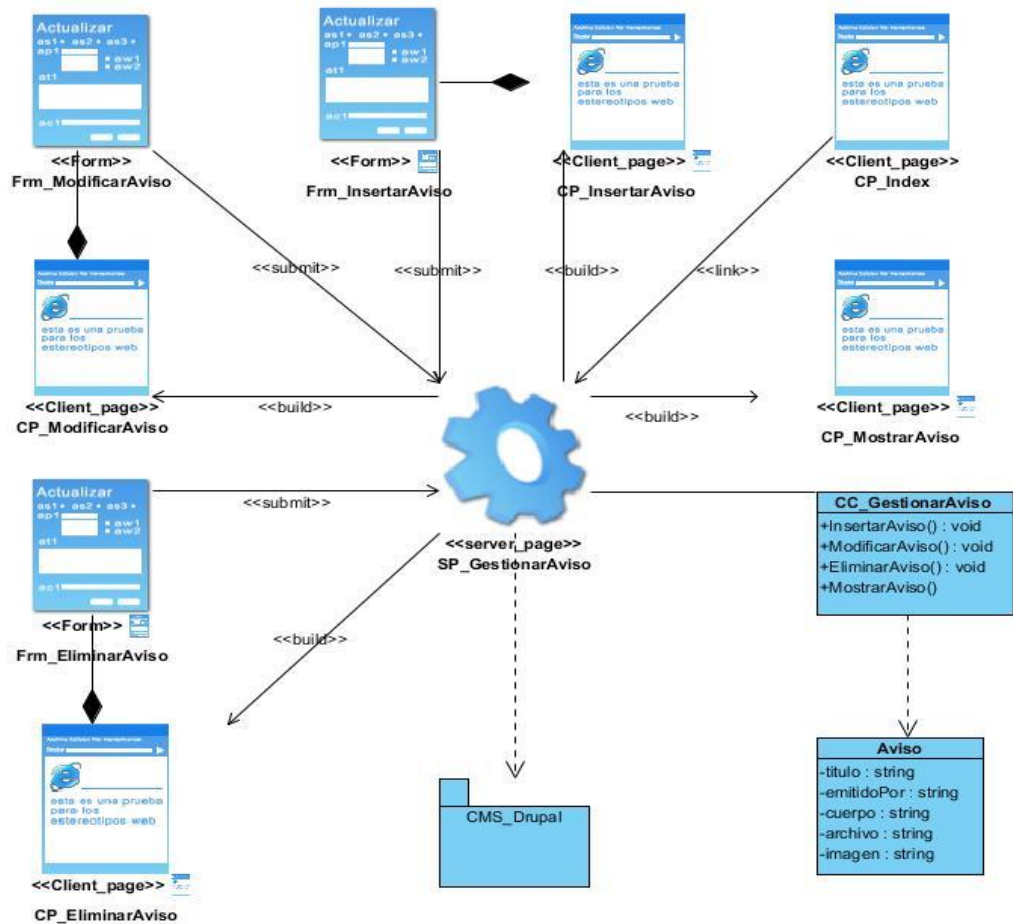


Figura 9. Paquete Gestionar Aviso

2.7 Diagrama de paquetes

Con el objetivo de alcanzar un mayor dominio sobre el funcionamiento del portal, se hace necesario describir su estructura. A continuación se muestra cómo se encuentra conformado arquitectónicamente el Portal Web de la Facultad 1, para ello se representan los elementos correspondientes a cada capa de la aplicación.

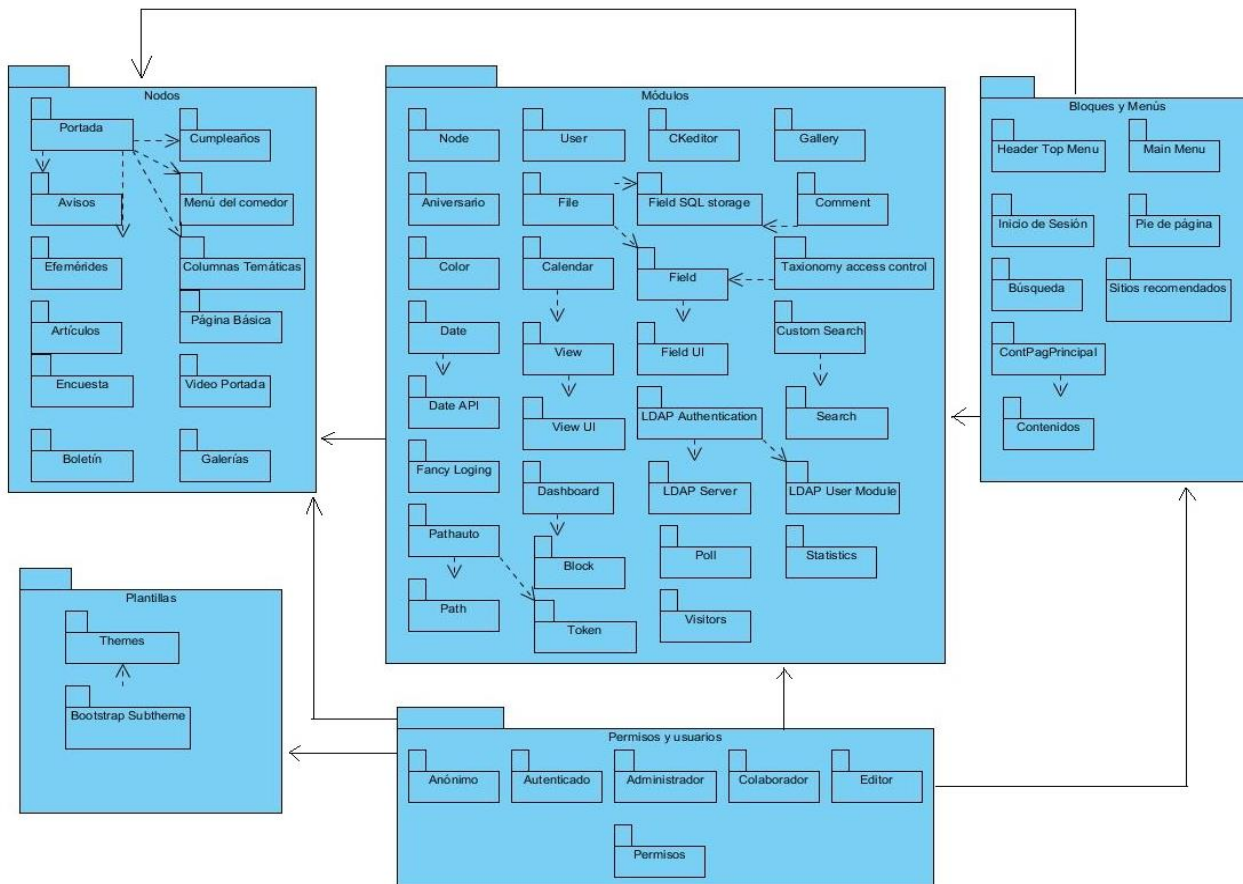


Figura 10. Diagrama de paquetes

2.8 Modelo de despliegue

El diagrama de despliegue es uno de los tipos de diagramas que aparecen cuando se modelan los aspectos físicos de los sistemas orientados a objetos. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Son utilizados para modelar la vista de despliegue estática de un sistema. La siguiente figura muestra el diagrama de despliegue correspondiente a la propuesta de solución.

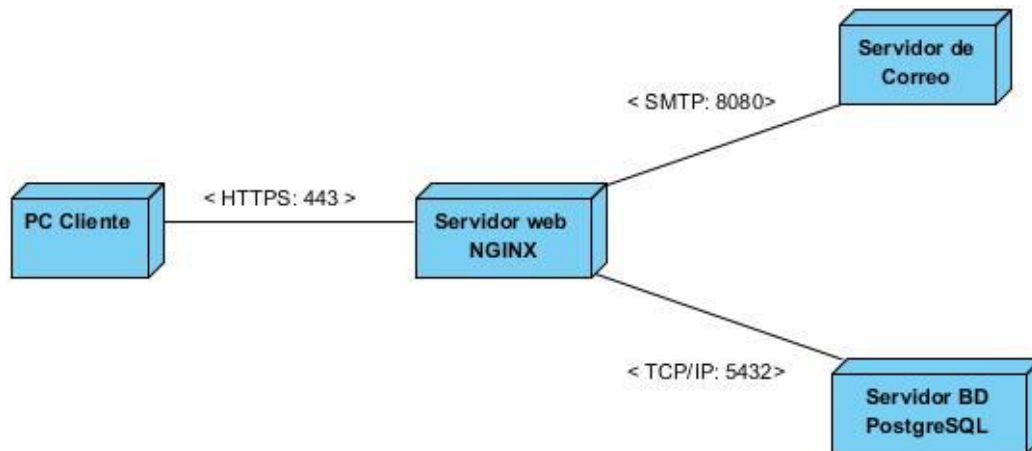


Figura 11. Modelo de despliegue

Descripción de los elementos de interface y comunicación

<<HTTPS>>: Protocolo para establecer la conexión segura entre la PC cliente y el servidor de aplicaciones a través del puerto.

<<TCP/IP>>: Protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de base de datos a través del puerto definido para el gestor de base de datos PostgreSQL: 5432. La conexión entre estos servidores permitirá ejecutar un conjunto de órdenes y obtener rápidamente respuesta a las mismas.

<<SMTP>>: Protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de correo. Esta conexión permitirá la autenticación en el sistema a través de las cuentas de usuarios, así como el envío de boletines.

2.9 Conclusiones parciales

- La caracterización realizada permitió describir los elementos más significativos de la arquitectura del sistema, así como la definición de los requisitos funcionales y no funcionales, a través de los que se establecerán las principales funcionalidades del sistema.
- Las actividades contempladas en la etapa de planificación-definición, posibilitaron dar una mayor descripción de los elementos del sistema, permitiendo una mejor comprensión para su implementación.

Capítulo 3: Programación y Evaluación de la aplicación web para el Portal Web de la Facultad 1

3.1 Introducción

En el capítulo se presentará la definición de los estándares de codificación que debe tener en cuenta el desarrollador durante la etapa de implementación del software, así como el plan de iteraciones y las tareas de ingeniería por historia de usuario. Además se mostrará una descripción de las pruebas empleadas en la validación de la herramienta desarrollada, presentándose a su vez los resultados obtenidos durante dicho proceso.

3.2 Diagrama de componentes

Los diagramas de componentes muestran los elementos de diseño de un sistema de software. Permiten visualizar con mayor facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de interfaces. (Microsoft Developer Network, 2014)

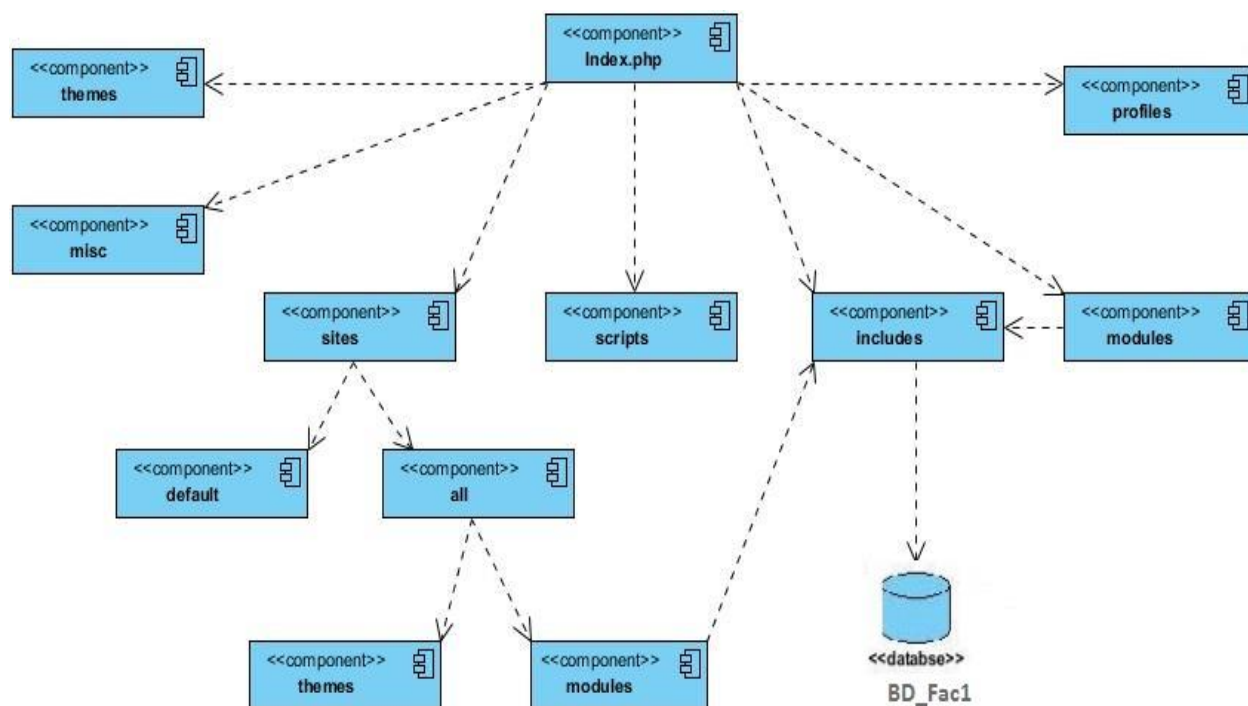


Figura 12. Diagrama de componentes

A continuación se describen los componentes presentes en el diagrama:

- ***Index.php***: Este componente es el punto de inicio de la aplicación. A partir de esta entrada se solicitan los diferentes módulos del sistema de gestión de contenidos.
- ***Themes***: En este componente se incluyen los temas que vienen con la distribución de Drupal.
- ***Misc***: Incluye los archivos JavaScript e imágenes requeridas por el sistema.
- ***Includes***: Contiene los ficheros indispensables para el funcionamiento de Drupal.
- ***Script***: Proporciona utilidades adicionales que no emplea Drupal directamente, pero que pueden ser utilizadas a través de las líneas de comandos de Shell.
- ***Modules***: En este componente se encuentran los módulos contenidos en el core de Drupal.
- ***Profiles***: Contiene los perfiles de instalación de Drupal.
- ***Sites***: Se encuentran elementos extras y modificaciones que se añaden al sistema original.

3.3 Estándares de codificación

Los estándares de codificación constituyen un factor primordial en el desarrollo de proyectos. Garantizan que el código obtenido sea fácil de leer, entendido y modificado independientemente de quien haya sido el desarrollador del producto. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores. Drupal proporciona a sus desarrolladores un conjunto de normas para fomentar el código de una forma uniforme para todos. (Gil, 2012) A continuación se detallan los estándares de codificación utilizados en la implementación de los módulos.

Indentación

La Identación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza Identación para anidar elementos. En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además no se debe dejar espacios en blanco al final de cada línea. En el siguiente ejemplo se muestra un fragmento de código con las identaciones realizadas, de 2 espacios cada una, y los saltos de línea o *Enter* al final de cada línea (sin dejar espacios).

```

<?php
function forum_block_view($delta = '') {
  $query = db_select('forum_index', 'f');
  $query->fields('f');
  $query->addTag('node_access');
  switch ($delta) {
    case 'active':
      $title = t('Active forum topics');
      $query->orderBy('f.last_comment_timestamp', 'DESC');
      $query->range(0, variable_get('forum_block_num_active', '5'));
      break;
    case 'new':
      $title = t('New forum topics');
      $query->orderBy('f.created', 'DESC');
      $query->range(0, variable_get('forum_block_num_new', '5'));
      break;
  }
  $block['subject'] = $title;
  // Cache based on the altered query.
  $block['content'] = drupal_render_cache_by_query($query,
    'forum_block_view');
  $block['content']['#access'] = user_access('access content');
  return $block;
}
?>

```

Etiquetas de apertura y cierre

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<?` y `?>`. En general se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos `.module` y `.inc`. Esta convención evita que 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 ...”*. Por tanto, la etiqueta de cierre final del archivo (`?>`) es opcional en Drupal. A continuación se muestra un ejemplo de la utilización de las etiquetas apertura y cierre, contenido en el archivo `page.tpl.php`.

```

<div class="col-md-2">
  <!-- #header-top-right -->
  <div class="clearfix" id="header-top-right">
    <div class="header-top-area">
      <?php print render($page['header_top_right']); ?>
    </div>
  </div>
  <!-- EOF:#header-top-right -->
</div>

```

Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, \$numero = 3, en el lugar de \$numero=3. Esto se aplica a operadores como +, -, *, /, =, !=, >, <, . (Concatenación de cadenas), .=, +=, -=, etc. Los operadores unarios como ++, -- no deben tener separación. Por ejemplo \$numero++.

Uso de comillas

Se pueden utilizar tanto las comillas simples como la ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\${title}</h1>". También se recomienda el uso de las comillas dobles cuando el texto puede incluir alguna comilla simple.

Uso de punto y coma (;) en código PHP

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

- Correcto: <?php print \$title; ?>.

- Incorrecto: <?php print \$title ?>.

Estructuras de control

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

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

- ✓ 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 { } aun en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).
- ✓ Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

Funciones

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

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

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra a continuación:

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

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

```
$numero1 = foo($a, $type);
```

```
$primer_valor = foo2($b);
```

```
$i = foo3();
```

Arrays

Los valores dentro de un *array* (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del *array* supera los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```

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

```

Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúscula. La única excepción son los archivos de documentación, que tendrán extensión .txt y el nombre en mayúscula. Por ejemplo README.txt.

Comentar el código

Para la realización de comentarios suelen emplear */** para comentarios en varias líneas y *//* para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, esta deberá escribirse en mayúscula (por ejemplo TRUE o FALSE).

```

/*!
 * Bootstrap v3.3.2 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 *//*/ normalize.css v3.0.2 | MIT License | git.io/normalize */

```

3.4 Plan de iteraciones

El plan de entrega del producto se divide en cuatro iteraciones, teniendo como base la prioridad y complejidad de cada uno de los requisitos del sistema, partiendo de la definición de las historias de usuarios. Al concluir la última iteración, la aplicación se encuentra lista para su despliegue. A continuación se detallan los elementos analizados en cada iteración.

Iteración 1

En la primera iteración se implementan las historias de usuarios con mayor prioridad, obteniendo al final de la misma una primera versión de prueba en el proceso de desarrollo incremental y dotando al sistema de sus primeras funcionalidades.

Iteración 2

En la segunda iteración se realiza la implementación de los requisitos correspondientes a las historias de usuarios con prioridad alta para el cliente, además se corrigen los errores encontrados en las historias de usuarios desarrolladas durante la primera iteración. De esta forma se obtiene la segunda versión de pruebas del software.

Iteración 3

En la tercera iteración se realiza la implementación de los requisitos correspondientes a las historias de usuarios de prioridad media para el cliente. Se corrigen los errores encontrados en la versión anterior.

Iteración 4

En la tercera iteración, se desarrollan los requisitos de prioridad baja, contenidos en las restantes historias de usuarios. Al concluir se obtiene la versión 1.0 del producto final.

Duración de las iteraciones

Para lograr una correcta organización en el desarrollo de la aplicación, se hace necesaria la creación de un plan de duración de iteraciones. Este plan se confecciona con el objetivo de establecer un orden en el desarrollo de los requisitos de la aplicación, así como el tiempo destinado a cada una de las iteraciones.

Iteración	Descripción de la iteración	Orden de las HU a implementar	Duración total
1	Desarrollo de las historias de usuarios de prioridad muy alta.	1, 2, 3, 4, 5, 11,12, 21, 22, 25, 26	8 semanas
3	Desarrollo de las historias de usuarios de prioridad alta.	8, 9, 13, 14, 16, 17, 24, 32, 33	8 semanas
2	Desarrollo de las historias de usuarios de prioridad media	6, 7, 10, 15, 23, 27, 28, 29, 30, 31, 34, 35, 36, 37	5 semanas
3	Desarrollo de las historias de usuarios de prioridad baja	19, 18, 20	1 semana

Tabla 5. Plan de iteraciones

3.5 Tareas de ingeniería

A continuación se muestran varias de las tareas de ingeniería realizadas para desarrollar los requisitos de prioridad muy alta.

Tarea de Ingeniería	
Número tarea: 1	Número Historia de Usuario: 1
Nombre tarea: Instalación del módulo User	
Tipo de tarea: Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 6/10/14	Fecha Fin: 6/10/14
Programador responsable: Wendy Rodriguez Muñoz	
Descripción: Se utiliza el módulo User para dar el acceso a los usuarios a la aplicación.	

Tabla 6. Tarea de ingeniería

Tarea de Ingeniería	
Número tarea: 2	Número Historia de Usuario: 4, 7, 11, 12, 19, 22
Nombre tarea: Instalación y configuración del editor de texto Ckeditor.	
Tipo de tarea: Desarrollo	Puntos Estimados: 2 días
Fecha Inicio: 8/10/14	Fecha Fin: 9 /10/14
Programador responsable: Wendy Rodriguez Muñoz	
Descripción: Se instala y configura el editor de texto Ckeditor para generar páginas con contenidos ricos en estilo sin necesidad de conocer el lenguaje HTML.	

Tabla 7. Tarea de ingeniería

3.6 Pantalla de la aplicación

MI Facultad Formación Investigación y Postgrado Extensión Producción Administración Tecnología Organizaciones
Buscar



EL DECANO INFORMA COBERTURA CURIOSITOS ANDARES PROTAGONISTAS RECETAS DE LA ABUELA ¿A DÓNDE VOY?

Iniciar

Entérate / Conozca lo más relevante del acontecer nacional e internacional



Sed ut perspiciatis unde omnis iste natus error sit voluptatem sit voluptatem
Mar. 12/05/2015 - 18:24

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

[Leer Más](#)

Avisos

- i Reunión el próximo martes con los testistas
Fecha Dom, 10/05/2015 - 20:32
- i Reunión de la rectora con Sto año
Fecha Lun, 04/05/2015 - 18:27
- i Taller de Trabajo Político-Ideológico
Fecha Mar, 04/06/2013 - 19:43
- i Reunión con 3er año Lunes 22 8:30 AM Bajos del Docente
Fecha Lun, 22/04/2013 - 13:46

Ciencia y Tecnología / Infórmese sobre la actualidad tecnológica



Nave espacial rusa caerá en las próximas horas en la Tierra
Mar. 12/05/2015 - 18:24

Roscosmos, la Agencia Espacial rusa, ha anunciado que el carguero espacial ruso Progress M-27M alcanzará la atmósfera de la Tierra entre las 22-13...

[Leer Más](#)

Deportes / Infórmese sobre la actualidad deportiva



Kroos y Benzema podrían disputar semifinal Real Madrid-Juventus
2 semanas 2 días

El alemán Toni Kroos y el francés Karim Benzema, dos titulares indiscutidos del club español Real Madrid, entrenaron hoy con normalidad y apuntan a

[Leer Más](#)

Recomendamos

- Dragon3s
- Octavitos
- Grundys
- Gladiadores
- Humanos
- Android

Prensa Digital

- Periódico Meilla
- Cubadebate
- Granma
- Juventud Rebelde
- Ecured

Boletín de los Zorros

¡Manténgase informado de nuestras últimas novedades!

Visitantes

Total de visitas: 6

Copyright © Facultad 1 2015. Todos los derechos reservados.

3.7 Validación de la propuesta de solución

Los procesos de pruebas de software se destacan como el proceso de establecer datos de entrada al producto de software e inspeccionar los resultados obtenidos, permitiendo determinar si el software funciona realmente como se espera. Las pruebas de software responden fundamentalmente a dos interrogantes, ¿se ha obtenido un buen producto?, ¿se ha desarrollado de forma correcta? Este concepto da lugar al proceso de verificación y validación del software. (Zamora, 2011)

Según Zamora las pruebas de software se diferencian del resto de las actividades del proceso de desarrollo de software en que estas son un proceso destructivo. El objetivo de quien realiza las pruebas, es descubrir errores que no han sido detectados con anterioridad. Entre los principales tipos de pruebas realizadas a aplicaciones web se encuentran las pruebas de aceptación, las pruebas de usabilidad, las pruebas de seguridad y las pruebas de carga y estrés.

3.7.1 Pruebas de aceptación

Las pruebas de software son aquellos procedimientos que se realizan para verificar la calidad de un producto de software y pueden ser aplicadas periódicamente. Estas tienen como objetivo fundamental la identificación de posibles errores que se presenten en la aplicación. En la metodología SXP se realizan pruebas de aceptación, estas son las pruebas finales antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. (Quiala, 2014)

A continuación se describe el caso de prueba Autenticar Usuario:

Caso de prueba de aceptación	
Código de Caso de Prueba:	Nombre de Historia de Usuario: Autenticar usuario
Nombre de la persona que realiza la prueba: Wendy Rodriguez Muñoz	
Descripción de la prueba: Prueba a la funcionalidad Autenticar usuario	
Entrada / Pasos de ejecución: La entrada consta de la introducción de los siguientes campos en el formulario de inicio de sesión: ➤ nombre de usuario: wendy	

➤ contraseña: soporte

Luego de que el usuario seleccione el botón iniciar sesión, el sistema verifica en la base de datos que el usuario y la contraseña existan y sean correctos, para luego habilitar los permisos que tienen el usuario autenticado sobre la aplicación.

Resultado esperado: El usuario entra a su sesión.

Evaluación de la Prueba: Satisfactoria

Tabla 8. Caso de prueba Autenticar usuario

Resultados de las pruebas a las funcionalidades del software

Las pruebas funcionales aplicadas a la propuesta de solución fueron basadas en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional. En correspondencia con los artefactos generados por la metodología SXP, para la realización de las pruebas se empleó la Plantilla de Casos de Pruebas de Aceptación, en la que se describe el comportamiento de la aplicación ante determinado flujo de datos especificado por el probador.

En la primera iteración se detectan 28 no conformidades, de ellas 26 son de Error de idioma, 1 de Redacción y 1 de Validación. En la segunda iteración se encuentran un total de 7 no conformidades de Error de idioma. Como resultado final todas las no conformidades encontradas fueron resueltas, lo que demuestra que el software funciona correctamente.

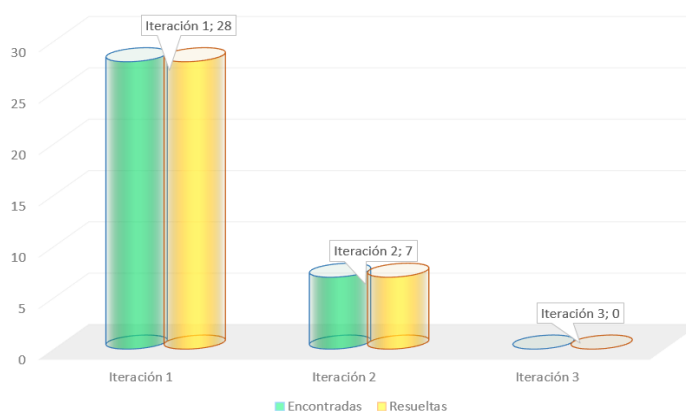


Figura 13. Resultados de las pruebas de funcionalidad

3.7.2 Pruebas de usabilidad

La meta de las pruebas de usabilidad es identificar y rectificar las deficiencias de usabilidad y errores existentes. Intenta asegurar que la aplicación sea fácil de aprender y de usar y que cumpla con las funcionalidades para la que fue diseñada. Aportan datos cuantitativos y cualitativos sobre usuarios reales que realizan tareas reales sobre el software. (Florian, Solarte y Reyes, 2010)

Para la realización de las pruebas de usabilidad a la propuesta de solución, se utilizó el documento rector para pruebas de este tipo que establece el Centro Nacional de Calidad de Software (CALISOFT), conocido como “Lista de Chequeo de Usabilidad de Sitios Web”. Esta cuenta con 110 indicadores separados en 13 elementos de usabilidad.

Resultados de las pruebas de usabilidad

Las pruebas fueron ejecutadas en dos iteraciones. En la primera iteración el resultado de la prueba fue regular, debido a que se encuentran 5 no conformidades, de ellas 2 críticas y 3 no críticas. En la segunda iteración se solucionan todas las no conformidades, resultando satisfactoria la prueba.

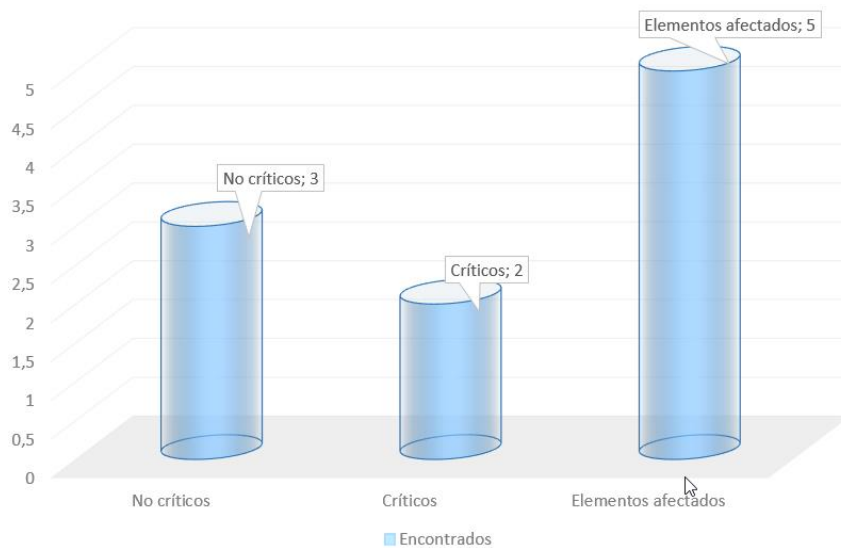


Figura 14. Resultados de las pruebas de usabilidad

3.7.3 Pruebas de carga

La carga de trabajo se refiere a la capacidad máxima que tiene un servidor web (hardware y software), para atender a un conjunto de usuarios de manera simultánea. Las pruebas de carga consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware es capaz de soportar la cantidad de visitas esperadas. (Zapata y Velazquez, 2011)

3.7.4 Pruebas de estrés (stress)

Estas pruebas son utilizadas normalmente para someter la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios superior al esperado. La concurrencia es uno de los factores que más afecta el desempeño, calidad y operabilidad de productos de software que funcionan en ambientes cliente-servidor, debido a que, comúnmente la carga transaccional del servidor (donde se ejecuta el software) es muy alta a consecuencia de las múltiples conexiones o peticiones de usuarios para realizar procesos en él.

El objetivo de las pruebas de estrés es intentar romper el sistema. La idea es encontrar las circunstancias bajo las cuales colapsará. Es importante porque puede revelar defectos en tiempo real, así como las áreas débiles donde el diseño defectuoso podría causar la indisponibilidad del servicio. (Zapata y Velázquez, 2011)

Resultados de las pruebas de carga y estrés

Para la realización de las pruebas de carga y estrés se empleó la herramienta Apache JMeter. El ambiente de prueba estuvo conformado por:

- Sistema Operativo: Windows 8.0
- Microprocesador: Inter(R) Core(TM) i5-3230M CPU @2.60GHz
- Memoria RAM: 6 GB
- Disco Duro: 750 GB

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 100 usuarios concurrentes se encuentra en el rango de

tiempo de 1 a 5 segundos. La propuesta de solución generó un total de 25576,07 kb/seg de transferencia de datos para 100 usuarios, con un rendimiento de 17.2 seg. Con ello queda demostrado que la propuesta de solución es estable, ya que se mantuvo prestando servicios todo el tiempo, sin incurrir en fallos

3.7.5 Pruebas de seguridad

Las pruebas de seguridad están diseñadas para probar las vulnerabilidades en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor de vuelta, y el ambiente del lado del servidor. Cada uno de estos dominios poder recibir ataques, y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo. (Pressman, 2001)

Resultados de las pruebas de seguridad

Las pruebas de seguridad se aplican en dos iteraciones, hasta el nivel 2 de los tres niveles establecidos por los especialistas del grupo de Seguridad del DEPSW de la Universidad de las Ciencias Informáticas. En el primer nivel se evalúan los elementos contenidos en la “Lista de Chequeo de Pruebas de Seguridad para estudiantes”. En el segundo nivel se desarrollaron pruebas mediante la herramienta *Acunetix Web Vulnerability Scanner 8.0*, la cual ha sido considerada como la herramienta pionera en la tecnología de análisis de seguridad.

En la primera iteración del nivel 1 se detectan 3 no conformidades no críticas, quedando evaluada de insatisfactoria la prueba. Las no conformidades encontradas se solucionan en su totalidad. Para la segunda y última iteración de prueba no se encuentran no conformidades, por lo que la prueba es evaluada de satisfactoria.

En la primera iteración que comprende el nivel 2 se detectan 115 no conformidades, de ellas 70 alertas de tipo alta, 3 alertas tipo baja y 42 de tipo informacionales. En la segunda iteración se solucionan todos los problemas encontrados que pudieran comprometer la seguridad e integridad de la aplicación.

3.8 Conclusiones del capítulo

- La confección de los diagramas de componentes y despliegue permitieron establecer una mayor comprensión de la arquitectura y funcionamiento de los elementos del sistema.

- Las pruebas realizadas al software permitieron determinar y corregir errores y fallos en el funcionamiento de la aplicación, lo que permite asegurar que la propuesta de solución es estable y segura.

Conclusiones generales

- El estudio de las tendencias actuales del desarrollo de portales web y la definición de las necesidades de los usuarios finales, permitieron identificar setenta y uno (71) requisitos funcionales a tener en cuenta en el desarrollo del portal web de la Facultad 1.
- La selección de las tecnologías necesarias permitió el desarrollo de todas las funcionalidades, obteniéndose como resultado el portal web de la Facultad 1.
- Las pruebas de software realizadas al sistema permitieron mejorar la calidad de la aplicación.

Recomendaciones

- Incorporar elementos de Accesibilidad al portal web de la Facultad 1.
- Mejorar el diseño web adaptativo

Referencias Bibliográficas

- ✓ BAEZA-YATES, R., LOAIZA, C.R. y MARTÍN, J.V. 2004. Arquitectura de la información y usabilidad en la web. , vol. 13, no. 3, pp. 168-178.
- ✓ BAKKEN, SAETHER, SCHMID, STIG y EGON, 2003. *Manual de PHP* [en línea]. 2003. S.I.: s.n. Disponible en: <http://www.php.net/docs.php>.
- ✓ BARRIENTOS, E.Á. 2014. Formación de usuarios de la información mediante aplicaciones Web 2.0. [en línea], no. 55. ISSN 1562-4730 (online). DOI 10.5195/biblios.2014.160. Disponible en: <http://biblios.pitt.edu/ojs/index.php/biblios/article/view/160/200>.
- ✓ BETANCOURT, D.B. 2010. *Análisis y diseño del portal "Calidad Facultad 9"*. Pregrado. La Habana, Cuba: Universidad de las Ciencias Informaticas.
- ✓ BILIB 2012. Análisis de Aplicación: Apache JMeter. *Centro de Apoyo Tecnológico a Ciudadanos y Empresas* [en línea]. Disponible en: <http://www.bilib.es/recursos/catalogo-de-aplicaciones/analisis/doc/analisis-de-aplicacion-apache-jmeter/>.
- ✓ CANÓS, J.H., LETELIER, P. y PENADÉS, M.C., 2010. *Métodologías Ágiles en el Desarrollo de Software*. 2010. S.I.: s.n.
- ✓ CHINEA, M.V. y RODRÍGUEZ, R.S. 2009. *Implementación del nuevo Portal de la Facultad 3*. [en línea]. Pregrado. Ciudad de La Habana, Cuba: Universidad de las Ciencias Informaticas. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2249_09.
- ✓ FABRI, F. 2012. El bloc de notas más completo y avanzado. [en línea]. Disponible en: <http://notepad.softonic.com/>.
- ✓ FLORIAN, B.E., SOLARTE, O. y REYES, J. 2010. Propuesta para incorporar evaluación y pruebas de usabilidad dentro de un proceso de desarrollo de software. [en línea], ISSN 1794-1237. Disponible en: http://www.scielo.org/co/scielo.php?pid=S1794-12372010000100010&script=sci_arttext.
- ✓ GIL, F. 2012. *Experto en Drupal. Curso de creación y gestión de portales web con Drupal 7*. [en línea]. 1.1. España: s.n. Disponible en: <https://www.forcontu.com/libros/experto-en-drupal-7-nivel-inicial>.
- ✓ GOOVY, M., 2014. *Netbeans IDE de desarrollo*. [en línea]. 2014. S.I.: s.n. Disponible en: http://www.genbeta.deu.com/herramienta_netbeans-1.
- ✓ GRAF, H. 2011. *Joomla: Guía para principiantes* [en línea]. S.I.: s.n. Disponible en: <http://cocoate.com/es/j17es>.

- ✓ HERNÁNDEZ, J.Z. 2011. *Análisis de los procesos de verificación y validación en las organizaciones software*. [en línea]. Madrid, España: s.n. Disponible en: <http://orff.uc3m.es/handle/10016/12880>.
- ✓ HERNÁNDEZ, Y.A., 2011. *CONFIGURACIÓN DE LA METODOLOGÍA OPENUPV1.0* [en línea]. 2011. S.l.: s.n. Disponible en: http://www.somosjovenes.cu/sites/default/files/cidi_configuracion_de_la_metodologia_open_up.pdf.
- ✓ IBÁÑEZ, L.B. 2014. *Desarrollo del portal web Observatorio Político Cubano en Internet*. Pregrado. La Habana, Cuba: Universidad de las Ciencias Informaticas.
- ✓ INTEF 2013. *Multimedia y Web 2.0*. [en línea], ISSN 978-84-369-5351-0. Disponible en: http://www.ite.educacion.es/formacion/materiales/155/cd/modulo_1_Iniciacionblog/concepto_de_web_20.html.
- ✓ LEYVA, I.P., GORRÍN, M.G., MARTÍNEZ, N.S. y ALARCÓN, D.M. 2011. *ASEGURANDO LA USABILIDAD EN EL PROCESO DE DESARROLLO DEL SOFTWARE*. , pp. 16.
- ✓ LUCAS, J., 2011. *Tutorial: Como instalar NGINX EN uBUNTU 14.04 lts*. mayo 2011. S.l.: s.n.
- ✓ MÁRQUEZ, J., SAMPEDRO, L. y VARGAS, F. 2012. *Instalación y configuración de Apache, un servidor web gratis*. , vol. 012, pp. 10-23.
- ✓ MARTÍNEZ, J.C.M., GUZMÁN, L.A., ALARCÓN, V.M.H. y GÓMEZ, C.M. 2013. *Diagramas de navegación en aplicaciones Web*. , vol. 10, no. 2, pp. 17. ISSN 1794-211X.
- ✓ MARTÍNEZ, R. 2010. *Sobre PostgreSQL* | www.postgresql.org.es. [en línea]. [Consulta: 28 mayo 2015]. Disponible en: http://www.postgresql.org.es/sobre_postgresql.
- ✓ MÉNDEZ, A.V., 2010. *Metodologías de desarrollo de Software*. diciembre 2010. S.l.: s.n.
- ✓ MICROSOFT DEVELOPER NETWORK 2014. *Diagramas de componentes de UML: Referencia*. [en línea]. Disponible en: <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
- ✓ ORACLE CORPORATION 2014. *Las 10 razones principales para usar Mysql como base de datos integrada*. [en línea]. Disponible en: <http://www.mysql.com>.
- ✓ PÉREZ, J.E., 2014a. *Introducción a AJAX* [en línea]. 2014. S.l.: s.n. Disponible en: <http://librosweb.es/libro/ajax/>.
- ✓ PÉREZ, J.E. 2014b. *Introducción a CSS* [en línea]. España: s.n. Disponible en: <http://librosweb.es/libro/css/>.
- ✓ PÉREZ, J.E. 2014. *Introducción a JavaScript* [en línea]. España: s.n. Disponible en: <http://librosweb.es/libro/javascript/>.

- ✓ PÉREZ, J.E. 2014c. *Introducción a XHTML* [en línea]. España: s.n. Disponible en: <http://librosweb.es/libro/xhtml/>.
- ✓ PRESSMAN, R.S. 2001. *Ingeniería de Software: Un enfoque práctico*. 5. S.l.: Mc Graw Hill. ISBN 0-07-285318-2.
- ✓ QUIALA, I.B.S. 2014. *Aplicación de soporte a la Metodología de desarrollo de distribuciones GNU/Linux, Nova-OpenUp*. Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.
- ✓ RAMÍREZ, A.M. y SANFRÁN, M.M. 2010. *Desarrollo del Portal Web de la Federación Cubana de Voleibol*. [en línea]. Pregrado. Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_02864_10.
- ✓ ROMERO, G.M.P., PUENTE, S.J.G.D.L. y ABAD, A.M. 2010. SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE. , pp. 12.
- ✓ ROSENFELD, L. y MORVILLE, P. 2002. Information Architecture for the World Wide Web. , no. 2. ISSN 0-596-00035-9. 2002.
- ✓ SÁNCHEZ, E., 2010. *WordPress. Guía para principiantes*. 2010. S.l.: s.n.
- ✓ TOUB, S. 2000. EVALUATING INFORMATION ARCHITECTURE. , pp. 27.
- ✓ VEGA, A.A. 2013. *Responsive Web Design: Interfaces Web Adaptables al dispositivo empleando HTML5 y CSS3*. [en línea]. Pregrado. España: UNIVERSIDAD DE ALCALÁ. Disponible en: <http://dspace.uah.es/dspace/bitstream/handle/10017/19972/Memoria.pdf?sequence=1>.
- ✓ W3SCHOOLS.COM 2015. Bootstrap 3 Tutorial. *Bootstrap 3 Tutorial* [en línea]. [Consulta: 26 mayo 2015]. Disponible en: <http://www.w3schools.com/bootstrap/default.asp>.
- ✓ WEB2MOVIL 2013. Web2movil | How to - Tutorial básico Responsive Design en español. [en línea]. [Consulta: 28 mayo 2015]. Disponible en: <http://www.web2movil.com/tutorial-basico-responsive-design/>.
- ✓ ZAPATA, C.M. y VELAZQUEZ, C. de J.C. 2011. Comparacion de las características de algunas herramientas de software para pruebas de carga. ,
- ✓ ZULUETA, D.P., ÁLVAREZ, A.R., LOMBIDA, Y.M. y PEREZ, D.E. 2012. Accesibilidad en sitios Web. , pp. 13. ISSN 978-959-286-019-3.

Bibliografía Consultada

1. ESCALONA, A.C. 2010. *Arquitectura de la Información del portal de la Facultad1*. Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.
2. FUNDORA, E.T. 2013. *Portal para el Modelo Cubano de Desarrollo de Aplicaciones Informáticas (MCDAI)*. Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.
3. LEON, R.A.H. y GONZALEZ, S.C 2011. *El proceso de Investigación Científica*. ISBN 978-959-16-1307-3.
4. ÑÚNEZ, S.A.P. 2012. *Uso de JAX en el desarrollo de aplicaciones web con Java y PHP*. , pp. 21.
5. SOMMERVILLE, I. 2005. *Ingeniería del Software*. 6. S.I. ISBN 84-7829-074-5.
6. TORRES, L.M.A. 2014. *Red Internacional de Dirección Estratégica en la Educación Superior*. Trabajo de diploma para optar por el título de ingeniero en Ciencias Informáticas. La Habana: Universidad de las Ciencias Informáticas.

Glosario de Términos

Módulo: Es un software que agrupa un conjunto de subprogramas y estructuras de datos.

Permisos: Son reglas asociadas a los objetos de un equipo o red, como archivos y carpetas. Los permisos determinan si se puede obtener acceso a un objeto y lo que se puede hacer con él.

Rol: Función que una persona desempeña en un lugar o situación determinados.

SEO: El posicionamiento web, también conocido como SEO, es el conjunto de técnicas que aplicadas a un sitio web mejoran su posicionamiento o visibilidad en los diferentes buscadores.