



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

IMPLEMENTACIÓN DE UN SISTEMA ADMINISTRADOR DE CONTENIDOS WEB

TRABAJO DE TESIS
PARA OBTENER EL GRADO DE

INGENIERA EN REDES

PRESENTA
MARÍA ROSALINA EK DZIB

DIRECTORA DE TESIS
MTI. MELISSA BLANQUETO ESTRADA

ASESORES
M.S.I LAURA YÉSICA DÁVALOS CASTILLA

M. INGENIERO SERGIO ALBERTO SOLÍS SOSA
M. INGENIERO DIMITR VENIAMIN CABAÑAS VICTORIA

DR. JAIRES SILVERIO ORTEGÓN AGUILAR
SERVICIOS DE TITULACIONES



CHETUMAL QUINTANA ROO, MÉXICO, NOVIEMBRE DE 2017



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

**TRABAJO DE TESIS ELABORADO BAJO SUPERVISIÓN DEL
COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO
PARCIAL PARA OBTENER EL GRADO DE:**

INGENIERA EN REDES

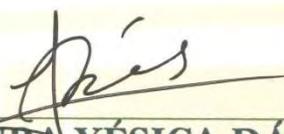
COMITÉ DE TRABAJO DE TESIS

DIRECTORA:



M.TI. MELISSA BLANQUETO ESTRADA

ASESORA:

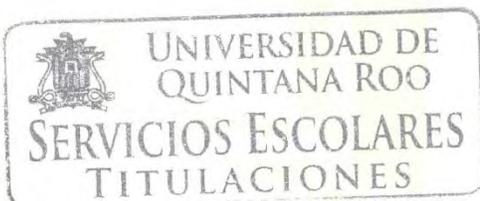


M.S.I LAURA YÉSICA DÁVALOS CASTILLA

ASESOR:



L.I. SERGIO ALBERTO SOLÍS SOSA



CONTENIDO

CAPÍTULO I

INTRODUCCIÓN	1
ANTECEDENTES	2
DESCRIPCIÓN DEL PROBLEMA	4
OBJETIVOS	6
<i>OBJETIVO GENERAL</i>	6
<i>OBJETIVOS ESPECÍFICOS</i>	6
JUSTIFICACIÓN	7
ALCANCES.....	8
<i>LIMITACIONES</i>	8

CAPÍTULO II

SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS (CMS).....	9
<i>HISTORIA</i>	9
<i>CONCEPTO DE CMS</i>	11
<i>CARACTERÍSTICAS DESEABLES EN UN ADMINISTRADOR DE CONTENIDOS</i>	12
<i>CARACTERÍSTICAS DE FUNCIONAMIENTO</i>	12
<i>NIVELES DE ORGANIZACIÓN</i>	14
PHP HYPERTEXT PRE-PROCESOR	15
CARACTERÍSTICAS FUNDAMENTALES DEL LENGUAJE	16

CAPÍTULO III

MARCO CONTEXTUAL.....	18
-----------------------	----

CAPÍTULO IV



METODOLOGÍA DE INVESTIGACIÓN.....	21
<i>TÉCNICAS DE INVESTIGACIÓN</i>	22
<i>INSTRUMENTOS DE INVESTIGACIÓN</i>	22
<i>OBJETIVOS CUALITATIVOS</i>	22
<i>ALCANCE DE LA INVESTIGACIÓN</i>	23
METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE.....	23
CICLO DE VIDA DEL SOFTWARE	23
<i>MODELO DE ENTREGA POR ETAPAS</i>	24

CAPITULO V

DEFINICIÓN DEL PROBLEMA	26
ANÁLISIS DE REQUERIMIENTOS	29
<i>ARQUITECTURA DEL SISTEMA</i>	30
<i>HARDWARE Y SOFTWARE</i>	31
<i>ESTRUCTURA DE ARCHIVOS</i>	32
<i>COMPONENTES DEL CMS</i>	35
DISEÑO	37
<i>MAPA DE NAVEGACIÓN</i>	38
<i>DISEÑO DE INTERFACES Y DEFINICIÓN DEL ESTILO</i>	40
<i>MÓDULO ADMINISTRADOR DE PÁGINAS</i>	46
<i>MÓDULO DE MENÚS</i>	49
<i>MÓDULO DE ARCHIVOS</i>	52
<i>MÓDULO DE AVISOS, EVENTOS, EVENTOS EXTERNOS, CONVOCATORIAS Y NOTICIAS</i>	54
<i>FRONT-END</i>	57
<i>ESTRUCTURA DE LA BASE DE DATOS</i>	60



CODIFICACIÓN	63
<i>TIPOS DE USUARIO</i>	65
<i>INICIO</i>	66
<i>CONEXIONES</i>	67
<i>PANEL DE CONTROL</i>	70
<i>MÓDULO PÁGINAS</i>	71
<i>EDITOR DE PÁGINAS</i>	73
<i>MÓDULO MENÚ DE NAVEGACIÓN</i>	76
<i>EDITOR GRUPOS</i>	78
<i>EDITOR OPCIONES</i>	79
<i>MÓDULO AVISOS, ANUNCIOS EXTERNOS, EVENTOS, CONVOCATORIAS</i>	81
PRUEBA	89
ENTREGA.....	92
CONCLUSIONES	93
BIBLIOGRAFÍA	95



TABLA DE ILUSTRACIONES

Figura 1 Editor WYSIWYG	13
Figura 2 Niveles de organización de las páginas en un CMS	14
Figura 3 Tipos de CMS	15
Figura 4 Soporte a enlaces locales y remotos.....	19
Figura 5 Mapa organizacional del DCT	20
Figura 6 Arquitectura Cliente-Servidor	31
Figura 7 Características Hardware	31
Figura 8 Características Software	32
Figura 9 Estructura de Archivos	34
Figura 10 Componentes del CMS	36
Figura 11 Mapa de Navegación	38
Figura 12 Layout del sistema	41
Figura 13 Datos de Acceso	45
Figura 14 Menú del sistema	45
Figura 15 Administrador de páginas.....	47
Figura 16 Editor de páginas	48
Figura 17 Editor de grupos.....	50
Figura 18 Editor de opciones.....	51
Figura 19 Panel de administración de archivos.....	53
Figura 20 Panel general de los módulos de publicación	55
Figura 21 Panel de publicaciones no vigentes	55



Figura 22 Vista previa de la publicación	56
Figura 23 Navegación del front-end	58
Figura 24 Área de contenidos del front-end	59
Figura 25 Diagrama Entidad-Relación	62
Figura 26 Codificación.....	63
Figura 27 Formulario de Inicio.....	66
Figura 28 parametros.php	67
Figura 29 Archivo lib.php.....	68
Figura 30 Archivo valida.php	69
Figura 31 Formulario inicio.....	69
Figura 32 Panel inicio.....	70
Figura 33 Archivo class.php	72
Figura 34 Menú modulo páginas.....	73
Figura 35 Editor páginas	74
Figura 36 Barra de herramientas TinyCME.....	75
Figura 37 Archivo paginas_editor.php.....	75
Figura 38 Modulo menú	76
Figura 39 Editor menú.....	77
Figura 40 Ejemplo sección encabezado.....	77
Figura 41 editor_grupos.php	78
Figura 42 Datos del link.....	79
Figura 44 Datos del link.....	80
Figura 45 Modulo eventos	81
Figura 46 bandeja.php	82

Figura 47 Editor eventos	83
Figura 48 Archivos del evento	84
Figura 49 Archivo ftpfunc.php.....	85
Figura 50 Banner del evento	86
Figura 51 Archivo sube_ficha.php	87
Figura 52 Integración de los módulos en el Front-End.....	88
Figura 53 Apartado Admisiones	90
Figura 54 Web personal	91



AGRADECIMIENTOS

En primer lugar quiero agradecer a Dios y a la vida por las oportunidades brindadas, pero sobre todo por los obstáculos que se interpusieron en mi camino y me ayudaron a crecer y ser mejor persona. Fue un camino difícil, pero nunca imposible, ahora puedo decir que al fin ¡lo logre!

Gracias a cada uno de los profesores que a lo largo de mi desarrollo profesional compartieron su conocimiento y me motivaron a seguir adelante con mis objetivos, sin ustedes no hubiera logrado culminar mis estudios.

En especial quiero agradecer al Lic. Sergio Solís por enseñarme el lenguaje de programación PHP desde cero, por su paciencia y dedicación durante el desarrollo del sistema.

A mis asesores y directora de tesis, por brindarme su apoyo durante todo el proceso, al MTI Vladimir Cabañas por animarme y apoyarme con todos los trámites, ¡muchas gracias!, paso mucho tiempo, ya era justo y necesario concluir con esta etapa.

Por ultimo quiero agradecer a una persona especial que tuve la dicha de conocer en el cuarto semestre de la carrera, al Ing. José Chan por ser mi compañero, amigo y hermano, llegaste en el momento indicado, ni antes, ni después, justo cuando necesite el apoyo para seguir adelante, estuviste ahí para motivarme a cumplir esta meta, ¡gracias!, y como siempre he dicho juntos en las buenas, malas y peores, somos un gran equipo.

A todos ustedes, muchas gracias, es la primera de muchas metas trazadas.



DEDICATORIA

A mi madre, sin ti, no lo hubiera logrado, eres pieza fundamental en mi vida, gracias por estar ahí, en mis desvelos, en los exámenes departamentales, cada vez que sentía que no iba a lograrlo, estuviste ahí, para abrazarme y aconsejarme, te dedico este trabajo, fruto de mi esfuerzo y lucha constante por permanecer en la universidad, no hay mejor herencia que el conocimiento, y tú me diste la oportunidad de adquirirlo.

A mis hermanas, tanto trabajo y dedicación tiene su recompensa, por ustedes sigo adelante para demostrarles que si yo pude ustedes también, siempre tenemos la oportunidad de cambiar la historia y ser mejores cada día, hay dos personitas que seguirán nuestros pasos, de nosotros depende el ejemplo que queremos darles.

Gracias por estar conmigo y apoyarme siempre, las quiero mucho.



RESUMEN

El presente trabajo de tesis describe la implementación de un Sistema Administrador de Contenidos Web, como solución de mejora en los tiempos de respuesta durante el proceso de creación de sitios y la administración de volúmenes considerables de información, esto como consecuencia a la creciente demanda de contenidos en el portal Web de la Universidad de Quintana Roo.

Para la implementación del sistema se empleó una metodología de investigación con enfoque cualitativo descrita en el Capítulo IV, el cual permitió obtener aspectos generales del comportamiento de los usuarios que intervienen en este proceso, y los posibles escenarios de trabajo. Esta información sirvió como base para determinar los requerimientos y objetivos que el sistema debe cumplir.

Ante la situación planteada, se realizó un análisis de los diferentes software existente en el mercado que pudieran cubrir las necesidades del usuario, ninguna de las opciones analizadas cumplía con todas las características establecidas, por esta razón, se decidió desarrollar un sistema desde cero, que permita realizar cambios constantes sin la afectación del contenido.

En el Capítulo V se describe el ciclo de vida del software implementado, el modelo de entrega por etapas, lo que permitió implementar por separado cada uno de los módulos del CMS, obteniendo un sistema altamente fiable, debido a que por cada módulo que se realizó se llevó a cabo la codificación, depuración, prueba y entrega.

En conclusión el contar con un sistema que administre los contenidos y publicaciones del portal universitario, reduce considerablemente los tiempos de desarrollo y ejecución de nuevos módulos, ya que se conoce el funcionamiento interno del sistema, la estructura y distribución de cada complemento, por tal motivo es posible crear nuevas funcionalidades y adaptar el sistema a distintas situaciones o entornos de trabajo.



CAPÍTULO I

INTRODUCCIÓN



INTRODUCCIÓN

La creación de un sitio Web ha sido y siempre será uno de los temas de mayor relevancia que se ha generado a través de los años desde la invención de Internet. Desarrollar un sitio Web puede ser un trabajo complejo, sino se cuentan con los conocimientos necesarios y con herramientas de apoyo que agilicen su elaboración.

En épocas pasadas, estas herramientas eran simples editores de texto que permitían generar sitios con información estática, que no se actualizaban frecuentemente y que proporcionaban una interacción limitada con los usuarios; de igual forma no existía el soporte para la incorporación de base de datos, foros y consultas online.

Debido al crecimiento exponencial de sitios Web y la gran demanda de actualizaciones de la información de las diversas páginas alojadas en Internet, surgió la necesidad de desarrollar aplicaciones que facilitaran su administración y mantenimiento, teniendo como resultado la implementación del primer Sistema Administrador de Contenidos (CMS, por sus siglas en ingles), a finales de los años noventa. Es así como surge la aparición de PHPNuke como una aplicación basada en Web, que popularizó el uso de estos sistemas en las comunidades de usuarios en Internet. Actualmente con el desarrollo de las tecnologías, se ha logrado la evolución de las aplicaciones de gestión de contenidos, que permiten incorporar estructuras completas de páginas dinámicas y otras funciones complementarias, enfocadas a la administración y actualización de manera interactiva en sitios Web.

ANTECEDENTES

Es complicado establecer un punto de partida donde las universidades hayan puesto interés e iniciado su participación en el desarrollo e implementación de CMS. La historia de los Sistemas de Administración de Contenidos, parte con la del desarrollo de lenguajes de programación y la presentación de información en páginas Web.

La existencia actual de estos sistemas construidos bajo las necesidades universitarias, tales como creación de portales Web, blogs, foros, plataformas de E-learning, administración de noticias, entre otros, se debe al desarrollo progresivo de las áreas tecnológicas y la necesidad de divulgación de hechos u acontecimientos académicos relevantes.

Entre algunas de las instituciones que han implementado este tipo de herramienta, se destacan las siguientes: Universidad Autónoma de México (UNAM), Universidad Autónoma de San Luis Potosí (UASLP) implementando plataformas alternativas para E-learning como Moodle, Universidad de Guadalajara con la adaptación de Drupal, entre otros.

Cabe resaltar que no únicamente en México se ha llevado a cabo estos estudios, las universidades de Latinoamérica han tenido mayor participación en la implementación de Administradores de Contenidos, así mismo las universidades españolas, que tras analizar las soluciones existentes en el mercado tanto como software propietario y código libre, optaron por desarrollar Sistemas de administración desde cero, tal es el caso de "Vuala", CMS desarrollado de manera interna por la universidad de Alicante, el cual permite la creación de contenido Web bajo los estándares de W3C. (García, 2008)

Existen algunas investigaciones, que hacen referencia al desarrollo de CMS en entornos universitarios, como los mencionados con anterioridad, lo cual proporciona un soporte sólido para la recopilación de información necesaria como objeto de referencia, en investigaciones posteriores.

Por otro lado, la Universidad de Quintana Roo cuenta con una versión de prueba de un Sistema Administrador de Contenidos, este CMS fue liberado a principios de enero de 2011 a cargo del área de Desarrollo Web del Departamento de Cómputo y Telemática, este sistema fue desarrollado desde cero para adaptarlo en su totalidad a las necesidades requeridas y al diseño existente del portal Web universitario.

Antes del desarrollo de este CMS, la administración del portal se realizaba a través de Dreamweaver, herramienta que permitía al Webmaster actualizar el contenido mediante la descarga de los archivos HTML por FTP, debido al crecimiento de los sitios Web universitarios y la rigurosa administración que representaba mantener actualizada toda la información, se optó la idea de utilizar alguna herramienta Web que permitiera facilitar esta labor.

Dentro de la parte operativa del CMS podemos encontrar los siguientes módulos de administración:

- Control Menús y Submenús.
- Administrador de páginas.
- Gestor de Archivos.
- Gestor Aviso, Eventos y Convocatorias.
- Visor de Noticias.

Cabe mencionar que el desarrollo de este sistema no se llevó a cabo bajo algún tipo de metodología o planeación, cada módulo fue realizado para la satisfacción de necesidades que surgieron en su momento. Esta herramienta provee una mejora en las publicaciones Web de la institución, pero a su vez refleja la falta de planeación y orden metodológico en el desarrollo del CMS universitario, provocando deficiencias en los módulos, y la falta de organización de los archivos de publicación.

DESCRIPCIÓN DEL PROBLEMA

Uno de los grandes pasos dados durante el desarrollo de Internet, fue plasmar información en páginas Web y mantenerla accesible desde cualquier parte del mundo. Esta idea sirvió como base para la formulación de la Web que hoy en día conocemos y la creación de aplicaciones que permiten satisfacer una gran cantidad de necesidades de la vida cotidiana de las personas.

Particularmente en el ámbito académico la Web desempeña un papel importante en la divulgación de conocimiento, implementación de tecnologías que permiten la integración de herramientas orientadas a la educación y docencia, publicación de acontecimientos académicos relevantes, y difusión planes de estudio.

Por tal motivo las universidades enfrentan con mayor seriedad el desarrollo y la actualización de información contenida en sus sitios Web, puesto que deben seguirse lineamientos internos y a su vez mantener el prestigio institucional como centros educativos de calidad y excelencia, comprometidos con la formación académica de los estudiantes.

Ahora bien, la problemática existente en la Universidad de Quintana Roo, radica en la administración y mantenimiento de los contenidos en su portal Web institucional, sitio en el cual se realizan diversas publicaciones, dirigidas a la comunidad universitaria y al público en general. Estas publicaciones deben de cumplir los siguientes lineamientos:

- La información contenida en la página de la universitaria debe ser de carácter institucional, o donde se note la participación académica de la universidad.
- Las solicitudes de publicación únicamente podrán realizarlas personal docente o administrativo.
- Toda publicación y/o actualización debe de realizarse de forma inmediata.
- Los banners y carteles a publicar requieren tamaños específicos.

- Todos los contenidos y el diseño de los sitios Web deben de tener un formato estándar acorde al diseño institucional aprobado en el Manual de Identidad Gráfica.

Las características anteriores influyen en los tiempos de respuesta de las publicaciones. La atención también se ve afectada por la falta de administración de volúmenes considerables de información, así como del número de solicitudes existentes, y el tiempo empleado en la elaboración de banners y carteles publicitarios.

Por otra parte, el control de los archivos es ineficiente como resultado de la escasa depuración de los mismos, ya que se encuentran alojados en el servidor Web por tiempo indefinido, almacenando información innecesaria en el disco duro.

Lo anterior refleja, las deficiencias en la administración en general de los sitios Web institucionales y la falta de una herramienta que permita agilizar el proceso de publicación de forma dinámica, con tiempos de respuesta mucho más cortos, donde se lleve a cabo una administración minuciosa de archivos, y sea posible jerarquizar la información.

Para la administración eficiente del sitio, se planteó como propuesta de solución la implementación de un Sistema de Administración de Contenidos (CMS) que dé solución a los aspectos mencionados con anterioridad. Por consiguiente, durante la investigación se pretende responder la siguiente cuestión:

¿El desarrollo y uso de un Sistema de Administración de Contenidos Web, agilizaría el procedimiento de publicación y administración de contenidos en los sitios Web institucionales?

OBJETIVOS

OBJETIVO GENERAL

Implementar un Sistema de Administración de Contenidos (CMS) que permita agilizar el proceso de publicación y administración del portal institucional de la Universidad de Quintana Roo, cumpliendo con los lineamientos de diseño institucional de los sitios Web universitarios.

OBJETIVOS ESPECÍFICOS

- Identificar las características que debe cubrir el nuevo Sistema Administrador de Contenidos (CMS).
- Definir qué factores intervienen durante el proceso de publicación.
- Conceptualizar las necesidades a resolver, en soluciones contundentes que permitan cubrir los requerimientos específicos.
- Diseñar un esquema de integración de módulos a programar, que permitan administrar el contenido publicado en el sitio Web institucional.
- Desarrollar módulos que conforman el entorno del CMS universitario.
- Configurar y adaptar el servidor Web para el manejo del CMS.
- Realizar pruebas de funcionalidad y desempeño del CMS.

JUSTIFICACIÓN

La Universidad de Quintana Roo cuenta con un portal Web institucional integrado por 34 sub-sitios, pertenecientes a las múltiples Divisiones Académicas y Departamentos. Este sitio Web se mantiene en constante cambio, ya que en él se realizan en promedio 20 publicaciones semanales, provenientes de diversas áreas de trabajo, clasificando las solicitudes dependiendo al tipo de publicación a la que pertenecen: noticias, avisos, eventos, convocatorias, entre otros. De igual forma se reciben solicitudes para la actualización de las mismas y la creación de nuevos sitios Web.

Debido a la creciente demanda de publicaciones y la gran cantidad de solicitudes para el desarrollo de nuevos sitios Web, así como el laborioso trabajo de administración que estos representan, se plantea el desarrollo de un administrador de contenido Web, que permita crear sitios de manera simplificada.

La implementación de esta herramienta, proveerá la mejora en el proceso de publicación. En lugar de elaborar página por página, se desarrollará un sistema que permita crear páginas Web siguiendo una plantilla de diseño predefinida, esto otorgará plena libertad al usuario de poder modificar el contenido, sin alterar el diseño y sin tener conocimientos en estructuras HTML o conocimientos de programación.

Adicionalmente este sistema informático permitirá el control de usuarios que participen en el mantenimiento del sitio, así como la posibilidad de crecer a medida que el administrador lo requiera, proporcionando escalabilidad.

Por otra parte, el acceso al sistema podrá realizarse de forma interna, a través de la intranet, o de forma remota a través de cualquier otro sitio que cuente con acceso de Internet.

El desarrollo de este sistema está dirigido a los administradores de los sitios Web de la universidad; quienes son responsables de la información publicada, y a quienes les será de gran utilidad esta herramienta.

ALCANCES

El implementar un nuevo sistema de administración de contenidos (CMS) permitirá establecer mecanismos y procedimientos que proporcionen mejoras en el proceso de creación y manipulación de sitios Web, en el cual se define una estructura jerárquica que suministra el control de gran cantidad de páginas, accesibilidad desde cualquier sitio remoto que cuente con acceso a Internet, escalabilidad, mejora en la navegación de los sitios agrupando menús de primer y segundo nivel, control del estado de las publicaciones de acuerdo al estatus y periodo de vigencia, reducción de la duplicidad de información, seguridad del sitio estableciendo perfiles y permisos que lleven el control de los usuarios que participan en la administración y de igual forma las acciones que pueden realizar, consistencia visual homogénea aplicando un estilo para todas las páginas, y una misma estructura de elementos a través de patrones de diseño, reutilización de páginas y publicaciones al contar con un historial que facilite la recuperación de los datos generales de cualquier evento vigente o caducado , en síntesis el sistema debe poder ampliarse y adaptarse de acuerdo a las necesidades futuras.

LIMITACIONES

Principales limitaciones existentes en el desarrollo del sistema se detallan a continuación:

- El apego estricto a los lineamientos de diseño.
- La constante evolución de los lenguajes de programación, lo que provocaría que el sistema se vuelva obsoleto en cierto periodo de tiempo.
- Incompatibilidad con nuevas versiones de los navegadores.

CAPÍTULO II

MARCO TEÓRICO

SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS (CMS)

En términos generales, los sistemas de administración de contenidos (Content Management System o CMS) surgen como respuesta a las necesidades de una evolución tecnológica constante y del aprovechamiento de la misma. Convirtiéndose en los últimos años en herramientas esenciales orientadas a la creación y edición de información Web.

HISTORIA

Antes de la aparición de los administradores de contenido, la publicación de información en la Web se realizaba de forma lenta, laboriosa y poco ágil. Los propietarios de diversas páginas se veían en la necesidad de aprender a utilizar un programa de edición Web como Dreamweaver, FrontPage, entre otros, para elaborar manualmente cada uno de los archivos HTML necesarios, y conformar el contenido de los portales Web, esta información era presentada de forma estática, es decir que no contaba con efectos o funcionalidades más allá de texto y enlaces a otras páginas o archivos, dadas las condiciones anteriores la modificación de información y la manipulación de archivos, se convirtió en un trabajo complicado que únicamente podría realizarse por profesionistas informáticos, por lo tanto Internet alojaba contenido desactualizado y poco atractivo para los visitantes. (InfoAL, 2009)

Los primeros CMS fueron desarrollados a principios de los años 90, por organizaciones que necesitaban administrar publicaciones a gran escala, como noticias, artículos, documentos Web, y una diversidad de información. De esta manera se fueron ideando sistemas dinámicos y poco complejos en su utilización, que permitían realizar esta tarea de forma rápida y concisa.

En 1995, el sitio de noticias tecnológicas CNET presentó su primer sistema de administración de documentos y publicación, creando Vignette una de las empresas pioneras en el desarrollo de CMS.

Con la aparición de PHP como lenguaje de programación, el desarrollo de CMS dio un giro importante, debido a que la siguiente generación de CMS adoptó este lenguaje como estándar de programación, cambiando la forma de funcionamiento de estos sistemas, permitiendo la creación de páginas Web dinámicas, y la compatibilidad con la mayoría de sistemas operativos.

PHPNuke, fue el resultado de la implementación de PHP y SQL en su arquitectura interna, categorizado como un CMS impulsor de la Web 2.0, orientado a portales Web, fue lanzado a finales de los 90 popularizando el uso de los CMS, por ser un sistema rápido y eficaz para la creación de comunidades de usuarios Online. (Cuerda, 2004)

En el 2001 Michel Valdrighi desarrolló B2/Cafelog, sistema que estableció las bases para el desarrollo de Wordpress en el 2003, dedicado a la creación de blogs, incorporó funcionalidades imprescindibles en el desarrollo de CMS; como las plantillas intercambiables, compatibilidad entre sistemas, y APIs exclusivas para programadores. (InfoAL, 2009)

A su vez Mambo dio lugar a una de las comunidades de colaboración más grandes dedicadas al desarrollo de nuevos entornos y funcionalidades, galardonado como el mejor CMS de código abierto, en el 2005 restringió parte de su código, lo cual provocó la separación de gran parte de su comunidad abriendo paso a Joomla, quien integró la funcionalidad de las hojas de estilo CCS, gestión de versiones imprimibles de páginas y la incorporación de Flash. (InfoAL, 2009)

Con el desarrollo de estas herramientas, la expansión de las páginas Web ha sido de forma exponencial, ya que la administración de sus contenidos, se realizan a través de entornos amigables y fáciles de usar. Lo que permite desarrollar portales Web a medida de las necesidades de los usuarios y de acuerdo a la evolución de los lenguajes de programación.

Por lo tanto, los CMS se han convertido en parte fundamental en el mantenimiento de portales Web, en consecuencia, el interés en su estudio y desarrollo radica en la posibilidad de construir soluciones específicas, teniendo como referencia las

implementaciones de los CMS actuales y la colaboración de múltiples comunidades de desarrolladores.

CONCEPTO DE CMS

Un CMS soporta la creación, gestión, distribución y publicación de la información. Cubre el ciclo de vida completo de las páginas, contenidos y archivos alojados en un sitio Web, desde la provisión de simples herramientas para crear los contenidos, pasando por la publicación y, finalmente, el archivo de los mismos. También proporciona capacidades para gestionar la estructura del sitio Web, la apariencia de las páginas publicadas y el estilo de navegación que se ofrece a los usuarios. (Robertson, 2003)

Estos sistemas, que en ocasiones se ven como simples herramientas de publicación digital, cuenta con una arquitectura interna modular, que permite la jerarquización de los contenidos, además es posible la separación de la información y el diseño del sitio lo que proporciona mayor control de las modificaciones a realizarse, en las cuales se lleva a cabo la comprobación de usuarios permitidos para el mantenimiento del portal Web.

Otra de las características particulares de los sistemas CMS es la accesibilidad, debido a que la información se encuentra de forma centralizada y es operado a través de navegadores. El acceso a los contenidos se puede realizar desde cualquier computadora, ya sea a través de una Intranet o de Internet. Esto significa que el contenido no tiene que ser transferido de un lugar a otro para llevar a cabo el mantenimiento del mismo.

Un aspecto sobresaliente de los sistemas de administración de contenido es la creación de páginas interactivas, es decir que se generan según las peticiones de los usuarios, no se encuentran alojadas en el servidor como las páginas estáticas, sino que se realiza una conexión a una base de datos presentando en el navegador el resultado de la petición.

CARACTERÍSTICAS DESEABLES EN UN ADMINISTRADOR DE CONTENIDOS

De acuerdo a lo descrito con anterioridad, según Javier Albusac (2005) los CMS deben cumplir con una serie de condiciones mínimas para tener un cierto grado de calidad, las cuales se describen a continuación:

- Disponibilidad de la información. Tener acceso a ella siempre que sea necesario.
- Poder obtener la información de una forma selectiva (utilización de filtros).
- El tiempo de respuesta del sistema no debe ser alto.
- La información puede ser presentada de diferentes formas y en distintos medios.
- Es deseable una respuesta “inteligente” por parte del CMS a la hora de seleccionar la información. No debe mostrar información que el usuario no esté buscando.
- Facilidad de uso (usabilidad).

CARACTERÍSTICAS DE FUNCIONAMIENTO

James Robertson (2003) propone una división de la funcionalidad de los sistemas de gestión de contenidos en cuatro categorías: Creación de Contenido, Administración de Contenido, Publicación y Presentación.

CREACIÓN DE CONTENIDO

La creación de contenido en un CMS es de manera sencilla y rápida, ya que se maneja el lenguaje HTML para la creación de páginas sin ningún tipo de dificultad. Esto se debe a la incorporación de herramientas que permiten a los usuarios editar sin la necesidad de tener conocimientos previos en lenguajes de programación o etiquetas HTML.

Los CMS cuentan con el editor un editor de texto WYSIWYG (*WhatYouSee, IsWhatYouGet*), estos editores presentan una apariencia similar al procesador de texto Word, lo que proporciona al usuario un entorno de trabajo sencillo, gracias a la familiarización con las funcionalidades de la hoja de texto, en el que el usuario ve el resultado final mientras escribe. (Cuerda, 2004)

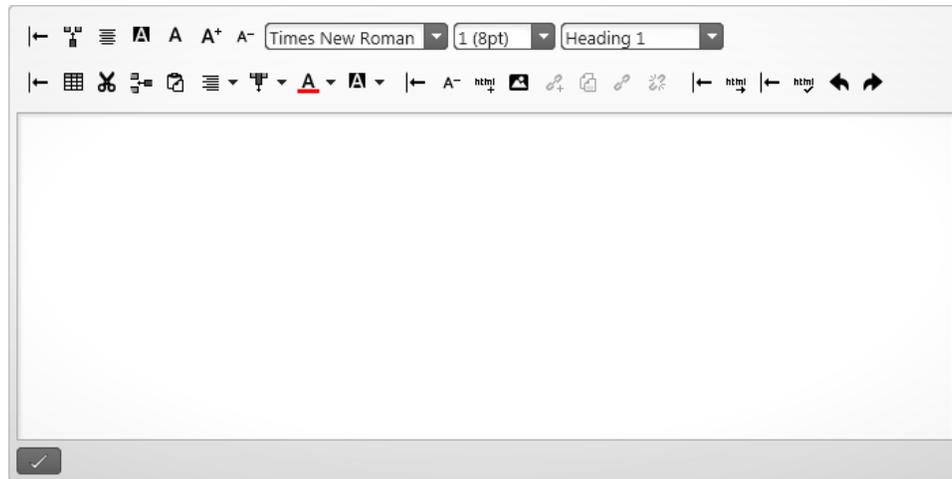


Figura 1 Editor WYSIWYG

ADMINISTRACIÓN DE CONTENIDO

Cuando la página se crea, es almacenada en una base de datos. Es decir, no se genera un archivo HTML, lo que significa, que el contenido generado en el textarea del editor WYSIWYG es enviado a un registro en la base de datos. Lo que permite llevar a cabo una serie de funciones descritas a continuación:

- Seguimiento de las actualizaciones de cada página.
- Administración de usuarios que participan en la modificación de los sitios.
- Determinación de permisos de lectura y escritura.
- La integración de toda la información.

PUBLICACIÓN

Una página aprobada se publica automáticamente cuando inicia el periodo de vigencia, y al momento de caducar se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la Web o para la sección

concreta donde está situada, de forma que el resultado final es un sitio Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y diseño permite que se pueda modificar el aspecto visual de un sitio Web sin afectar a los documentos ya creados. (CMS-Hispano, 2004)

PRESENTACIÓN

Los administradores de contenidos proporcionan y facilitan el soporte a muchos navegadores, ya que la presentación de la información se lleva a cabo de forma dinámica, es decir son el resultado de las peticiones de los usuarios, por lo tanto los contenidos no son archivos HTML alojados en el servidor, sino que son la composición de una base de datos.

NIVELES DE ORGANIZACIÓN

Otra parte fundamental en la estructura de un CMS, es la organización de los contenidos, el esquema de navegación permite al usuario ordenar de manera jerárquica las dependencias de cada una de las páginas pertenecientes al sitio Web. Estas dependencias se generan de forma automáticamente, permitiendo ubicar los niveles de organización.

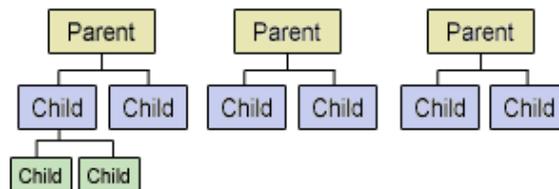


Figura 2 Niveles de organización de las páginas en un CMS

Las páginas pueden ubicarse en varios niveles, el primer nivel es el “padre”, a partir de este nivel se generan las dependencias para las páginas posteriores. Cada página puede tener varios hijos, que a su vez pueden ser los padres de otros hijos. Esta organización facilita y agiliza al máximo la búsqueda de información, al mismo tiempo, permite las actualizaciones futuras de la Web, sin afectar drásticamente el contenido de otras páginas.

TIPOS DE CMS

De acuerdo a la funcionalidad de cada administrador de contenidos, se puede clasificar en las siguientes categorías:

Portales: Creación de sitios web de uso general.
Ejemplo Joomla.

Blogs: Creación de bitácoras, ejemplo Blogger, y en un principio Wordpress

E-Commerce: Tiendas online, sitios dedicados a comercio electrónico. Magento y PrestaShop.

E-learning: plataformas desarrolladas para crear ambientes de aprendizaje personalizados. Ejemplo Moodle.



Figura 3 Tipos de CMS

PHP HYPERTEXT PRE-PROCESOR

PHP (acrónimo de PHP. Hypertext Pre-Procesor) es un lenguaje de programación, relativamente nuevo (su antecesor, PHP/FI, data de finales de 1994), concebido principalmente como herramienta para el desarrollo de aplicaciones Web. PHP permite diseñar paginas dinámicas de servidor, es decir, generar paginas bajo petición capaces de responder de manera inteligente a las demandas del cliente y permite la automatización de gran cantidad de tareas. (Gutiérrez Rodríguez & Bravo García, 2005)

En otras palabras, las páginas dinámicas se ejecutan en el servidor Web, justo antes de ser enviadas a través de Internet al cliente. Pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente, lo cual significa, que el cliente solamente recibe una página con el código HTML resultante de la ejecución PHP.

CARACTERÍSTICAS FUNDAMENTALES DEL LENGUAJE

PHP es un lenguaje para la creación de sitios Web del que se pueden destacar las siguientes características:

- Es un potente y robusto lenguaje de programación embebido en documentos HTML.
- Dispone de extensiones para la conexión con la gran mayoría de los sistemas de gestión de base de datos para el almacenamiento de información permanente en el servidor (Oracle, Postgres, Informix, MySQL, SQL Server...).
- Proporciona soporte a múltiples protocolos de comunicaciones en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.). (Gil Rubio, Alonso Villaverde, & Yagüe Panadero, 2006).

MYSQL

Es un sistema de gestión de datos relacional que utiliza el lenguaje SQL (Structured Query Language), multihilo y multiusuario, SBD de código abierto lanzado en 1995 y más tarde adquirido por la empresa MySQL AB en enero de 2008, una subsidiaria de Sun Microsystems.

MySQL está desarrollado en su mayor parte en código ANSI C y es muy utilizado en aplicaciones web, como phpBB, en plataformas (Linux, Windows, Apache, PHP, Perl) y por herramientas de seguimiento de errores como Bugzilla. (Arías, 2014)

AJAX

JavaScript Asíncrono y XML (AJAX) no es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Esto incluye: HTML o XHTML, CSS, JavaScript, DOM, XML, XSLT, y el objeto XMLHttpRequest. (Condor Tinoco & Soria Solís). Cuando estas tecnologías se combinan en un modelo AJAX, es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. Esto crea aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

CSS

Las hojas de estilo en cascada (Cascading Stylesheets) ofrecen propiedades para ampliar el lenguaje HTML en la representación visual de las páginas Web. El lenguaje CSS, definido por primera vez en el año 1996, es el más conocido y utilizado para definir propiedades de formato de los diferentes elementos HTML. (Schulz, 2009)

Proporciona una manera fácil y eficaz de definir el diseño visual de las páginas Web, mejorando la apariencia de los elementos que integran. Al estar todas las instrucciones de diseño y formato en una hoja de estilo, facilita las actualizaciones de los documentos HTML, debido a que las modificaciones se pueden realizar de forma centralizada.

Separar el contenido de la presentación es la base fundamental para conseguir páginas web más accesibles y con menos problemas de administración. Las hojas de estilo constan de una serie de instrucciones que definen como cómo se ha de presentar determinados elementos de la página.

CAPÍTULO III

MARCO CONTEXTUAL

MARCO CONTEXTUAL

La Universidad de Quintana Roo (UQROO), se fundó mediante el decreto de creación publicado el 31 de Mayo de 1991, como un centro académico de Nivel Superior, enfocado a impartir educación en los niveles de licenciatura, estudios de posgrado, cursos de actualización y especialización bajo diferentes modalidades de enseñanza para formar profesionistas, profesores e investigadores que requiere el Estado de Quintana Roo, la región y el país. (Universidad de Quintana Roo)

La institución cuenta con un sitio Web que funge como herramienta importante de divulgación del quehacer académico y un medio dinámico que contribuye a la comunicación interna entre los tres campus universitarios, creada el 29 de Marzo de 1997, en su primera versión el sitio Web de la universidad constaba solamente con documentos de hipertexto que tenían referencia a información básica de la institución.

Para el año 2005 se realizó el cambio más importante y notorio en el portal universitario, al incorporar páginas dinámicas, con diseños agradables y la publicación de nuevas secciones de información.

En donde se destacó la participación y compromiso del Departamento de Cómputo y Telemática, al aplicar una nueva estrategia de navegación flexible que incorporara el contenido de nuevos sitios Web universitarios.

Actualmente la universidad cuenta con el Centro de Tecnologías de la Información y Comunicación quien alberga dentro de sus instalaciones dos áreas de trabajo; “el Departamento de Sistemas” y “el Departamento de Cómputo y Telemática”. El CTIC (por su abreviación en siglas), se encuentra ubicado en el edificio “O” en la unidad académica Chetumal.

El Departamento de Cómputo y Telemática (DCT) tiene la responsabilidad de estar a la vanguardia en las tecnologías de información y telecomunicaciones para

brindar un servicio acorde a las crecientes necesidades de la comunidad universitaria. (Departamento de Computo y Telematica, 2010)

Actualmente brinda conectividad a las tres unidades académicas con las que cuenta la universidad: Unidad Chetumal, Unidad Cozumel, Unidad Playa del Carmen y la División de Ciencias de la Salud, proporciona acceso a la red inalámbrica a más 3000 alumnos y da soporte a 1500 nodos de red. En la siguiente figura se puede observar el mapa de soporte a enlaces remotos y locales.

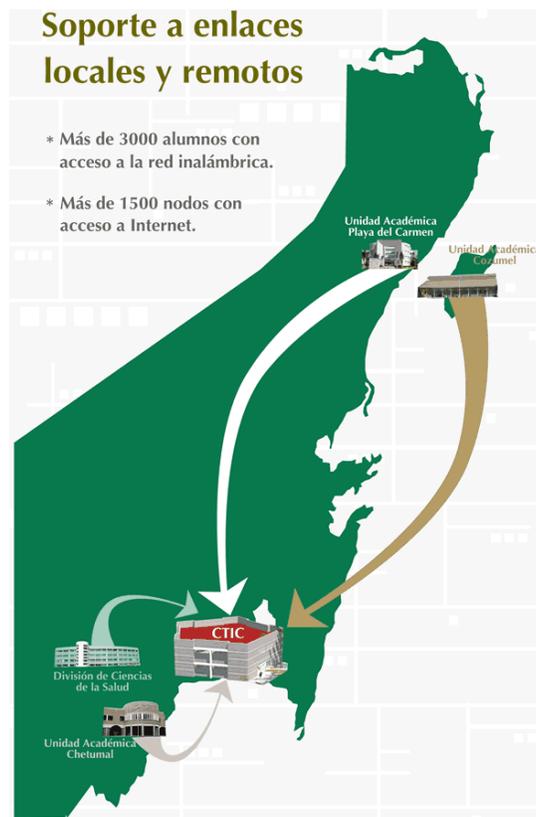


Figura 4 Soporte a enlaces locales y remotos.

Fuente: *ti.uqroo.mx*

A continuación podemos observar el mapa de organización del DCT y las áreas lo conforman:

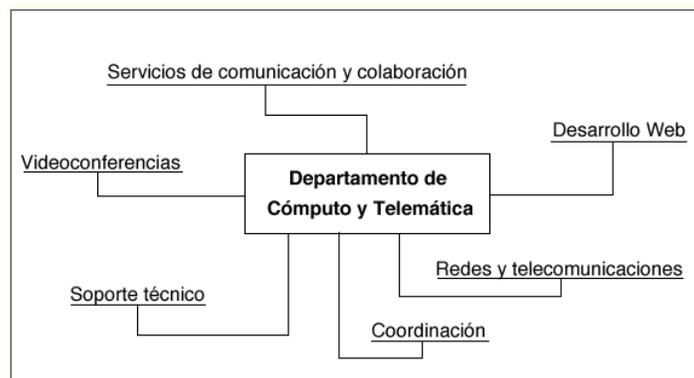


Figura 5 Mapa organizacional del DCT

El área de desarrollo Web es el encargado de administrar los contenidos que se publican en la página de la institución, todas las solicitudes son registradas a través del sistema de Atención a Usuarios (SAU), por medio del cual se lleva el control de los mismos, y en donde se agenda para su pronta atención.

Específicamente en esta área se llevará a cabo la presente investigación, dado a las características existentes, y debido a que se lleva el 98% del control del contenido publicado en la página universitaria y en la cual existe la problemática referente al proceso de publicación.

El desarrollo de un CMS servirá para mejorar el proceso de publicación y proveer al departamento, de un sistema informático que permita administrar con mayor rapidez todas las solicitudes que llegan al sistema SAU, mejorar el servicio que ofrece y a su vez, influir en los indicadores de calidad, en el cual participa.

Este sistema proveerá al Webmaster de una herramienta que pueda ser distribuida en otras áreas pertenecientes a la universidad, que requieran de la administración de sus respectivos sitios Web, como por ejemplo el COLEST, área de Becas, Admisiones, entre otros. Al igual que promueva la participación de los alumnos de servicio social en la publicación de avisos, eventos y convocatorias bajo la supervisión del Webmaster.

CAPÍTULO IV

METODOLOGÍA

METODOLOGÍA DE INVESTIGACIÓN

La investigación previa al desarrollo de software, determina los aspectos necesarios que el nuevo sistema debe de cubrir, para un óptimo desempeño. La recolección de información se llevara a cabo bajo una metodología de investigación, el modelo que se adopta es el de Sampieri (2006) debido a que cuenta con un enfoque bastante amplio y flexible que permite adaptar de manera general los requerimientos y alcances del estudio, cubriendo puntos esenciales que guían la elaboración de la investigación.

Las investigaciones generalmente tienen dos propósitos fundamentales, el primero es generar un nuevo conocimiento o contribuir en nuevos enfoques, el segundo es proveer herramientas que resuelvan un problema en específico, este último es el enfoque de la presente investigación.

Basados en la metodología de investigación descrita por Sampieri (2006) se puede deducir que la investigación a realizar cuenta con un enfoque cualitativo, es decir, que permite obtener conocimientos a partir del análisis de la información, no recolectando datos para aprobar teorías, sino analizando el comportamiento de los mismos en escenarios diferentes, y así obtener pautas para el desarrollo de soluciones específicas.

El estado del conocimiento del tema determino que la investigación es de tipo Exploratoria, como consecuencia de la poca existencia de información referente al tema de investigación, durante la revisión de la literatura se encontraron estudios realizados a cerca de la implementación de Sistemas de Administración de Contenidos, enfocados a las áreas de E-learning, Repositorios de Archivos, Foros y blogs. Utilizando en su mayoría CMS comerciales, como Joomla, Drupal etc. Estas ideas e investigaciones están poco relacionadas con el problema de estudio. Lo que proporciono la pauta para la determinación anterior.

TÉCNICAS DE INVESTIGACIÓN

Tomando en cuenta las necesidades de la investigación se determinó que el estudio de campo es el más adecuado, debido a que se realiza directamente en el medio donde se presenta la problemática.

Las técnicas de investigación a utilizar son las Entrevistas, las cuales serán aplicadas al personal de la Universidad de Quintana Roo, que tenga contacto directo con la administración de los diversos sitios Web de la institución, lo que permitirá determinar los requerimientos previos para el nuevo sistema. Y analizar en los distintos departamentos los aspectos que afectan al proceso de publicación.

INSTRUMENTOS DE INVESTIGACIÓN

Los instrumentos de investigación a utilizar, fueron seleccionados de acuerdo al estudio no experimental, ya que estos permiten observar las muestras ante situaciones existentes, para después analizar su comportamiento.

Con la implementación de las técnicas y herramientas de estudio anteriores, se pretende analizar los siguientes objetivos cualitativos, que darán pauta en aspectos como de usabilidad y accesibilidad para el desarrollo del sistema.

OBJETIVOS CUALITATIVOS

- Evaluar y describir el proceso de publicación
 - Análisis descriptivo de las actividades a realizar para la publicación.
 - Describir las experiencias y percepciones de los usuarios.
 - Evaluar los tiempos de respuesta.
 - Análisis de la interacción que lleva a cabo el usuario con ciertos módulos del sistema.
- Detectar aspectos que brinden una mejora en el proceso de publicación
 - Descripción de las necesidades, preferencias y expectativas de los usuarios.
- Evaluar la estética en general del sistema.
 - Análisis la interfaz de usuario.
 - Analizar sugerencias en cuestiones de diseño.

ALCANCE DE LA INVESTIGACIÓN

El propósito de la investigación cualitativa-exploratoria es describir los aspectos internos y externos que afectan el proceso de publicación, analizando la relación e interacción entre variables, para plantear una estrategia que permita desarrollar un sistema a la media de las necesidades de los usuarios, y proveer una base sólida para futuras investigaciones.

METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta sistematización nos indica como dividiremos un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas.

CICLO DE VIDA DEL SOFTWARE

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de un sistema, determinando el orden de las etapas involucradas, influye tanto en el éxito del proyecto como en cualquier otra decisión que se tome. Elegir un modelo de ciclo de vida apropiado orienta el proyecto hacia el cumplimiento de los objetivos. (Verzal, 2011)

El modelo de ciclo de vida elegido es el de *Modelo de Entrega por Etapas*, por varios motivos que se señalan a continuación:

- Es un modelo en el que el software se divide en pequeñas partes, lo que permite implementar por separado módulos del sistema.
- Genera un sistema altamente fiable, debido a que por cada módulo que se realice se lleva a cabo la codificación, depuración, prueba y la entrega.

MODELO DE ENTREGA POR ETAPAS

Este modelo cuenta con 3 etapas principales, pero solo la última etapa se repite por cada módulo que se desea implementar..

1. **Definición del problema:** Durante esta etapa se detallan todos los aspectos relacionados con el problema, las principales causas y los factores que intervienen.
2. **Análisis de requerimientos:** En esta etapa se define con claridad las necesidades de los usuarios, se determina que es lo que hace falta para solucionar el problema, se estudian las características a implementar en el nuevo sistema.
3. **Diseño Global:** Es el diseño de la solución del problema planteado inicialmente, teniendo como base los requerimientos establecidos, en la etapa anterior.
 - 3.1. **Codificación:** En esta parte del ciclo de vida, es donde se construye el software, se lleva a cabo la programación de las funcionalidades de cada módulo.
 - 3.2. **Depuración:** Es el proceso en el cual se identifican y corrigen errores.
 - 3.3. **Prueba:** en esta etapa se el modulo se emplea de manera experimental, para determinar si cumple con los requerimientos planteados, se emplean evaluaciones de calidad para verificar si el sistema cumple con las expectativas correctas.
 - 3.4. **Entrega:** Contando con todos los aspectos anteriores debidamente organizados y teniendo en cuenta la operación del proceso y el resultado de cada uno de sus pasos elaboramos conclusiones y resolvemos los estados sucesivos de desarrollo para la solución del problema que se definió en el comienzo del modelo.

Los módulos a desarrollados para el Sistema de Administración de Contenidos fueron los siguientes:

- Módulo para el Control Menús y Submenús.
- Módulo para la Creación y mantenimiento de páginas.
- Módulo para la administración de Archivos.
- Módulo para la Creación y mantenimiento de Avisos
- Módulo para la Creación y mantenimiento de Eventos Próximos
- Módulo para la Creación y mantenimiento de Convocatorias.
- Módulo para la Creación y mantenimiento de Noticias.

CAPITULO V

DESARROLLO DEL PROYECTO

DEFINICIÓN DEL PROBLEMA

La administración de un portal Web es una actividad complicada y laboriosa. Implica actualizar, eliminar y crear grandes volúmenes de contenidos, como páginas Web, archivos, perfiles de usuarios, entre otras.

Existe en el mercado una amplia variedad de aplicaciones que permiten solucionar algunos de los problemas de administración antes mencionados, pero en su gran mayoría se ha detectado ciertas dificultades al momento de distribuir los contenidos en las plantillas de diseño predefinidas por la herramienta, por tal motivo se solicitó el desarrollo un sistema que permita generar un sitio desde la estructura de los archivos, hasta la manipulación dinámica y sencilla del contenido de las páginas, con el propósito de reducir el tiempo de construcción y mantenimiento del portal.

Para el diseño e implementación del Sistema Administrador de Contenidos, se analizó la herramienta de publicación que utiliza el Departamento de Cómputo y Telemática, con la finalidad de observar su funcionamiento, determinar las deficiencias del sistema y conocer el entorno de trabajo en el cual se desarrollará el administrador de contenidos.

Con el objeto de recopilar información detallada se realizó una reunión con el Webmáster encargado del área de desarrollo Web, para obtener los aspectos que intervienen en el proceso de publicación y las dificultades que se presentan en la administración del portal universitario.

Con base en la recolección de la información anterior, se detalla la problemática encontrada:

- Todas las solicitudes de actualización de los contenidos publicados tanto en el portal institucional como en los diversos sitios anexos al mismo, se concentran en el área de desarrollo Web. Esta área es la encargada de dar solución a las peticiones, el alto índice de solicitudes provoca demoras en el tiempo de publicación y descontentos en los usuarios.

- El área de desarrollo Web está integrado por el Webmáster y alumnos de servicio social, debido a la magnitud de información existente es imposible poder administrar todos los sitios, es por ello que algunos departamentos y/o divisiones cuentan con administradores externos que implementan las páginas con diseños diferentes sin tomar en cuenta el manual de identidad gráfica, lo que provoca falta de homogeneidad visual en las mismas.
- El portal universitario posee cuentas en redes sociales como Twitter y Facebook en las cuales se publican de forma manual parte de los avisos, eventos y convocatorias contenidas en la página oficial, aún no se cuenta con un responsable específico, por tal motivo el administrador plantea que será necesario reflejar todas las publicaciones vigentes de manera automática en todas las redes sociales.

Siguiendo con la definición del problema, el análisis del sistema que el administrador Web utiliza para la publicación, en donde se hizo mención de

- La herramienta de publicación utilizada no cuenta con un agrupamiento óptimo de la información referente a las páginas, lo que provoca que se generen listas de visualización demasiado largas imposibilitando una navegación eficiente para el usuario, teniendo como referencia que el portal cuenta con aproximadamente 326 páginas por el momento.
- Si se desea eliminar una página es necesario recargar de nuevo el visualizador para verificar si se realizó de forma correcta la eliminación, por lo tanto se realiza una consulta al servidor para listar nuevamente las páginas existentes.
- No se cuenta con una vista previa de las páginas al momento de editarlas, es necesario conocer el "Id" de la página para verificar en el navegador las modificaciones que se realizaron.

- El portal universitario no cuenta con URLs fáciles de recordar por los usuarios, es necesario saber el “id” de la página para poder acceder al contenido o por el contrario navegar en el sitio hasta encontrar la información requerida.
- En el apartado de menús y archivos se listan todas las opciones contenidas en cada grupo de menús, no existe un orden o agrupación, es complicado para el usuario común la creación de un nuevo menú de navegación debido a la gran cantidad de opciones que se visualizan.
- Para la creación de un grupo es necesario saber el “id” del último grupo, debido a que no se manejan nombres que los identifiquen.
- Existen segmentos de código que son posibles de visualizar por el usuario en las interfaces.

ANÁLISIS DE REQUERIMIENTOS

Los requerimientos del sistema se determinaron con base a la problemática existente y las deficiencias que presenta el sistema actual.

El sistema a desarrollar debe tener la capacidad de:

- Ser intuitivo, fácil de entender y usar.
- Crear, actualizar y eliminar páginas dinámicas de manera fácil, sin la necesidad de tener conocimientos avanzados de programación.
- Controlar la publicación de páginas de forma inmediata.
- Organizar eficientemente los contenidos de los sitios en secciones permitiendo crear una estructura ordenada y sencilla.
- Proporcionar independencia entre el contenido y la presentación, de manera que un cambio de diseño no implique la revisión de todas las páginas para su adaptación.
- Delegar roles de usuarios para determinar qué acciones podrán realizar sobre los contenidos, debe ser posible crear nuevos roles y actualizar o eliminar los existentes.
- Administrar secciones de eventos, noticias, convocatorias, avisos.
- Publicar de forma automática en las Redes sociales, algunos de los contenidos existentes en el portal.
- Conformar las URLs con palabras relacionadas con el contenido de la página y fáciles de recordar (URLs amigables).
- Permitir el acceso de los usuarios a través de Internet.
- Permitir la autenticación de usuarios.
- Gestión de archivos.
- Administrar menús y submenús para presentar al usuario una forma más sencilla de crear y modificar menús.
- Presentar la información de forma ordenada, para facilitar la visualización.
- Centralizar el contenido de los sitios.

ARQUITECTURA DEL SISTEMA

Siguiendo con el ciclo de vida del software y de acuerdo a la información recolectada, se determinó que para el óptimo funcionamiento del sistema administrador de contenidos es necesario integrarlo bajo la arquitectura cliente-servidor.

El principal motivo por el que se aplica esta arquitectura, es el brindar una interacción ágil y eficiente del usuario con el sistema, de igual forma permite el acceso por usuarios multiplataforma a través del soporte de distintos navegadores Web.

CLIENTE

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de información, por lo tanto las interfaces se desarrollan sobre plataformas que permitan construir interfaces gráficas de usuario (GUI), es decir lo que compone visualmente al sitio o sistema.

SERVIDOR

El servidor es el encargado de atender las solicitudes de múltiples clientes, es el hospedaje físico donde se almacenan los datos que conforman el sitio Web y la interfaz. En conjunto con la base de datos organiza y gestiona los datos que componen los contenidos.

FUNCIONAMIENTO

En esta arquitectura el cliente realiza una petición al servidor a través de un navegador Web, el navegador envía la petición al servidor utilizando el protocolo HTTP O HTTPS según sea el caso. La solicitud llega al servidor Web correspondiente a través de la red, el servidor procesa la petición, si detecta que se trata de un documento con código PHP es enviado al intérprete PHP que reside y se ejecuta en el servidor Web, de esta forma el cliente lo único que recibe

es el resultado generado de forma dinámica a través de la consulta, seguidamente esta información se devuelve al servidor para que este a su vez lo transfiera al cliente que hizo la petición. Por último el cliente visualiza la página o interfaz generada en el navegador.



Figura 6 Arquitectura Cliente-Servidor

HARDWARE Y SOFTWARE

En este apartado se detalla la arquitectura del servidor que se utilizó para integrar el sistema desarrollado. En la siguiente figura se puede observar las características de los componentes de hardware.



Figura 7 Características Hardware

El servidor cuenta con dos procesadores SUNW UltraSPARC de 1165 MHz de procesamiento, 4 memorias RAM de 4 Gb cada una, haciendo un total de 16 Gb de memoria, un disco duro de 146 Gb.

En aspectos de software el servidor funciona bajo el sistema operativo SunOs, versión del kernel 5.10. Utilizando Apache 2.2.6 como servidor Web encargado de aceptar las peticiones de los usuarios. Para almacenar el contenido de las páginas se utilizó MySQL 5.0.45 como base de datos. La parte dinámica de las páginas fue a través del intérprete PHP en su versión 5.2.4.



Figura 8 Características Software

ESTRUCTURA DE ARCHIVOS

El sistema administrador de contenidos cuenta con una estructura de archivos independiente, que permite la incorporación de módulos que pueden o no incluirse en el sistema, es decir, que no afectan el funcionamiento del mismo ya que sólo se agregan características adicionales.

A continuación se describen las principales carpetas y archivos de configuración que conforman la arquitectura de archivos del CMS:

- **/LDAP.** En esta carpeta se almacenan los datos de configuración que permiten administrar directorios, es decir, acceder a bases de información de usuarios.

- **/adminfile:** En esta carpeta se almacena el administrador de archivos, incluye clases, jscripts y archivos de configuración, que permiten subir, eliminar archivos en el servidor, además de convertirlos a formato ZIP.
- **/editor:** Esta carpeta contiene el editor de páginas TinyMCE, que permite convertir la información escrita en el Textarea (contenido de la página) a formato HTML. Esta carpeta incluye los estilos y scripts del editor.
- **/WEB:** Esta carpeta incluye los archivos de configuración del administrador de páginas.
- **/Noticias:** Contiene los archivos de configuración del módulo de noticias.
- **/Avisos:** Contiene los archivos de configuración del módulo de avisos.
- **/Eventos:** Contiene los archivos de configuración del módulo de eventos.
- **/Convocatorias:** Contiene los archivos de configuración del módulo de convocatorias.
- **/img:** Contiene las imágenes utilizadas en la plantilla de diseño y de la interfaz del sistema.
- **index.php:** archivo que contiene la plantilla de diseño del sitio Web, y las llamadas PHP a la base de datos.
- **valida.php:** Archivo que realiza la validación de los usuarios.
- **menu.php:** Archivo que contiene la interfaz principal del sistema, incluye el acceso a cada uno de los módulos que lo conforman.
- **session.php:** Archivo que crea la sesión del usuario, si no se crea la sesión se deniega el acceso al sistema.

- **lib.php:** Archivo que realiza la conexión al servidor y selecciona la base de datos donde se guardara el contenido del sitio.
- **fin.php:** Archivo que termina la sesión del usuario.
- **parametros.php:** Archivo que contiene la dirección del servidor LDAP, URL del sitio y Path del servidor, datos que son consultados por config.php del adminfile.

Estructura de archivos



Figura 9 Estructura de Archivos

COMPONENTES DEL CMS

Núcleo: Parte funcional del administrador de contenidos, necesaria para su funcionamiento y para la incorporación del resto de componentes de la arquitectura.

Módulos: Agrega funcionalidades adicionales al CMS. El sistema cuenta con los siguientes módulos: **Avisos, Eventos, Convocatorias y Noticias.**

Cada módulo permite publicar su contenido en el sitio Web estableciendo la fecha de publicación y caducidad, por otra parte estos módulos se encuentran ligados con el modulo RSS, lo que significa que se tiene con la posibilidad de difundir el contenido a través de RSS y redes sociales como Facebook y Twitter.

Plantilla: Define el diseño específico del sitio Web, la plantilla puede ser modificada sin alterar los contenidos, no es necesario realizar la modificación página por página, al momento de cambiar o editar el diseño este se adapta a todas las páginas existentes, lo que proporciona homogeneidad visual.

Backend: Es el área privada del sitio donde se edita, administra, categoriza y ordena los contenidos, como textos e imágenes, en el cual solo tienen acceso los usuarios administradores que cuenten con usuario y contraseña válidos.

Las modificaciones realizadas en el back-end se verán reflejadas de manera inmediata en el front-end al momento que el administrador guarde o active el contenido.

Frontend: Es el sitio final con el que interactúan los usuarios finales, se compone de páginas, menús, galerías y textos.

Base de Datos: Organiza y gestiona los datos que componen los contenidos Web.

Web Server: Es el hospedaje físico donde se almacenan las estructuras HTML que componen el sitio Web, las plantillas de diseño, archivos etc.

En la siguiente figura podemos observar los componentes del sistema administrador de contenidos.

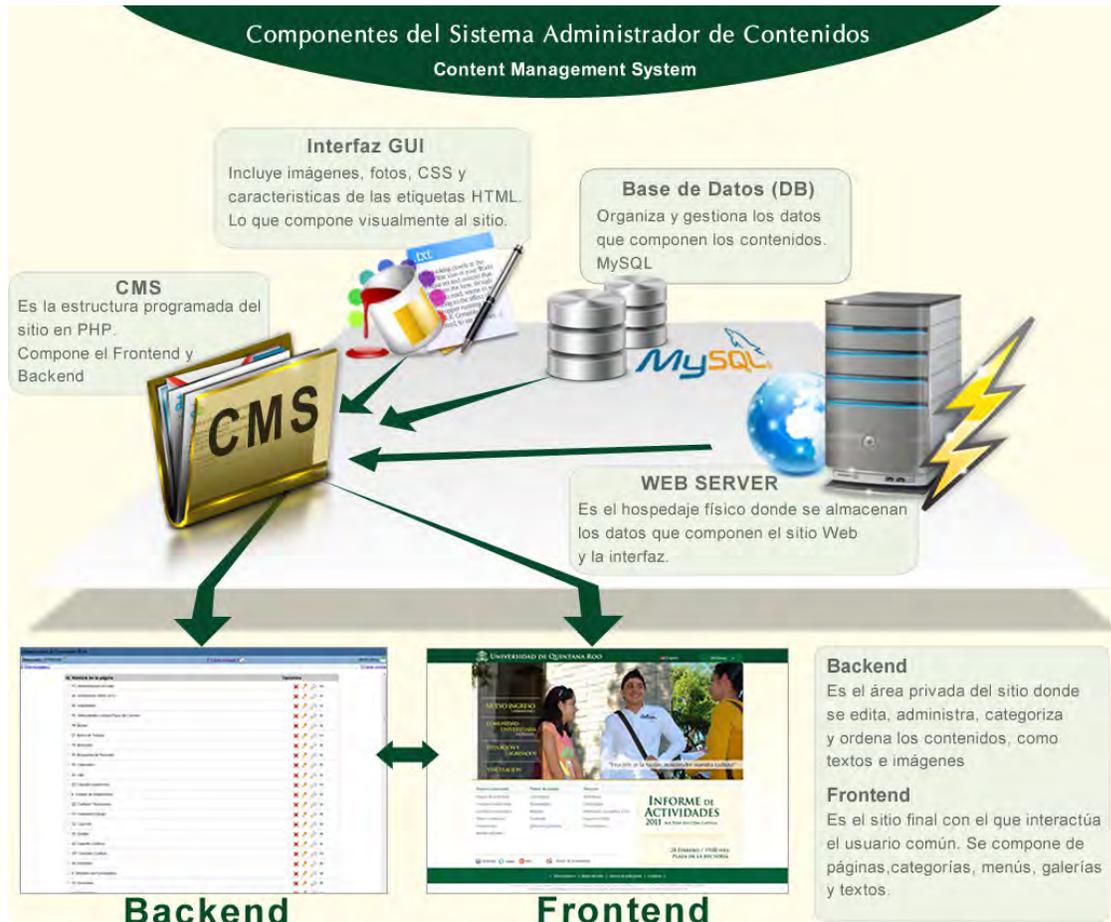


Figura 10 Componentes del CMS

DISEÑO

Cuando hacemos mención de sistemas basados en entornos Web nos referimos a aquellos que funcionan en Internet o intranets bajo las características de los navegadores, es decir que emplean un conjunto de elementos del navegador que permiten al usuario realizar acciones sobre el sistema que está utilizando.

Al desarrollar este tipo de sistemas es necesario considerar los elementos de identificación, navegación, contenidos y de acción en las interfaces, todos ellos deben estar preparados para ofrecer servicios determinados al usuario, con el fin de cumplir con los objetivos establecidos por el sistema.

Durante la etapa de diseño se planteó los prototipos de las interfaces, con el objetivo de determinar las propiedades básicas de cada módulo, las acciones que el usuario puede realizar dentro del sistema y los procesos que intervienen en la interacción del usuario y el administrador. En base a lo anterior, se estableció el entorno visual del CMS, la estructura de la base de datos y las características de acceso que cada módulo debe presentar, estas propiedades se tomaran en cuenta para la codificación del administrador en un lenguaje de programación específico.

A continuación se describe cada prototipo que conformará el aspecto visual del back-end.

- Mapa de navegación
- Diseño visual y definición del estilo
- Diseño de la interfaz de usuario
 - Layout
 - Elementos de navegación, información, interacción y apoyo.
- Back-end
 - Wireframes
- Front-end
 - Plantilla
 - Estructura de navegación

- Área de contenidos
- Estructura de la base de datos
 - Modelo E-R

MAPA DE NAVEGACIÓN

El mapa de navegación se desarrolló con el propósito de mostrar la estructura del sistema y el flujo de navegación, por lo tanto se establecieron las bases para la organización de los accesos y las interfaces que permitirán al usuario gestionar la administración del sitio Web y la visualización de los contenidos de forma estructurada y organizada.

A continuación se presenta el esquema jerárquico de los componentes del CMS: parámetros de configuración, archivos de sesión y validación de usuarios, panel de administración o menú principal y módulos que conforman el sistema. En la Figura 11 se puede observar las rutas o caminos de acceso a los contenidos, es decir, las interfaces y archivos que participaran en la interacción con el administrador.

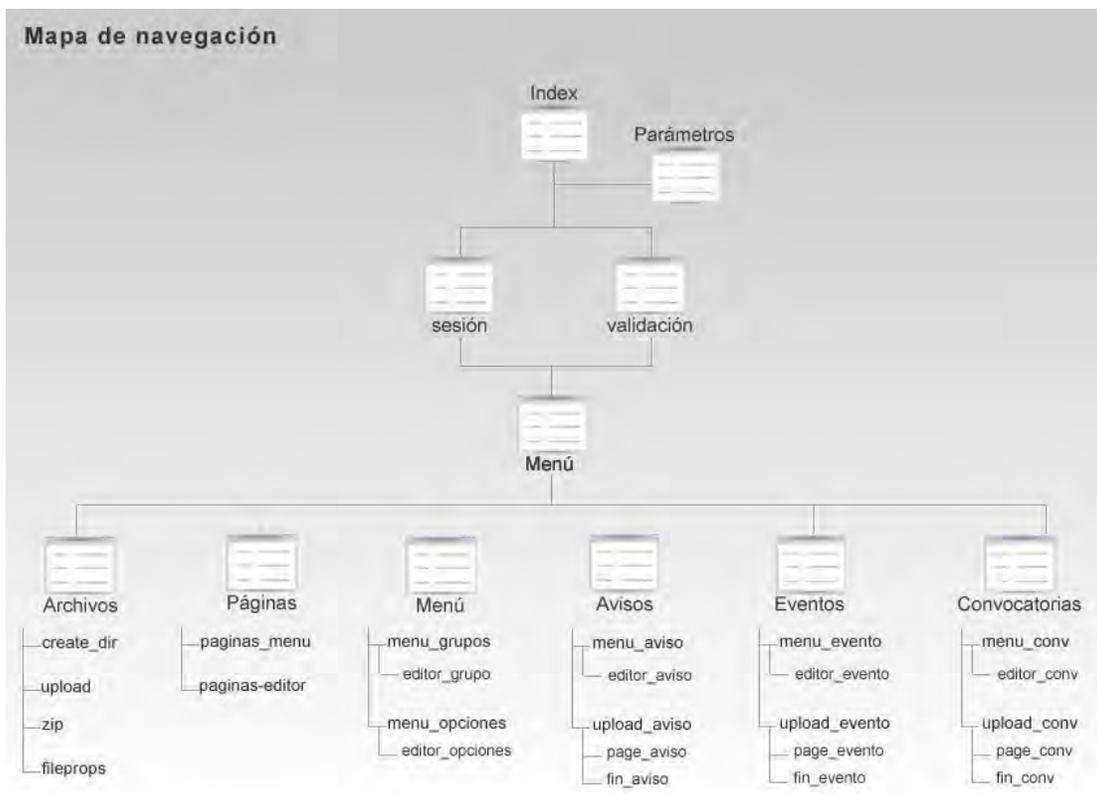


Figura 11 Mapa de Navegación

El esquema presenta la secuencia de interfaces que proporcionan dinamismo y funcionalidad a cada módulo, dando la pauta para elaborar escenarios de comportamiento del usuario y anticipar posibles errores en la organización de la información a presentar.

Para acceder al back-end se requiere de una validación, por lo que es necesario definir una página de inicio a la que nombraremos **index**, esta recibirá los datos del usuario para su autenticación, durante este proceso intervienen tres componentes fundamentales: parámetros, validación y sesión que interactúan entre sí para validar y cargar la configuración global del CMS.

Una vez realizado este paso se presenta el menú principal que debe contener links de acceso a las extensiones del sistema, al acceder se presentará un panel de administración que muestre las opciones que el usuario puede utilizar para la creación y modificación del contenido, de esta forma se guía al usuario hacia las ramificaciones o secuencias de interfaces que le permitirán realizar cada acción.

La construcción del mapa de navegación fue un punto importante ya que en base a su desarrollo se pudo conocer con exactitud cuántas interfaces se van a utilizar y cómo se va a desplazar el usuario en un futuro a través del sistema.

DISEÑO DE INTERFACES Y DEFINICIÓN DEL ESTILO

El diseño de interfaces para la Web es diferente del diseño tradicional, principalmente por el entorno en el que los usuarios acceden al sistema, es decir, cada plataforma interpreta la tipografía, estilos y proporciones de las pantallas de diferente forma, de acuerdo con lo anterior es necesario definir criterios en el diseño de las interfaces de usuario, para que al menos sino se visualiza la información de igual manera, sea de forma consistente en todas las plataformas.

Por consecuencia se desarrolló un layout principal para las interfaces del Back-end, donde se definen los componentes que integran la estructura de la interfaz prototipo de acuerdo a las siguientes características:

- Elementos de interacción:
 - Debe permitir mostrar, obtener y validar datos, al mismo tiempo filtrar e interpretar las acciones que se puede realizar en cada sección, con el fin de permitir el acceso únicamente a las operaciones necesarias de cada módulo.
 - Cada elemento captura un evento y llama a la función de control para indicar a la interfaz qué modo de visualización de información debe presentarse.
 - Es necesario ubicar cada elemento en secciones visibles para el usuario, de tal forma que la navegación a través del sistema sea de forma rápida y fácil de realizar.
- Presentación de la información
 - Debe existir secuencia entre la información y las acciones que podrán realizarse, de esa forma el usuario se familiarizará con cada elemento de la interfaz.
 - Por otro lado es indispensable que se presente rasgos generales del módulo, es decir, visualizar una descripción breve o resumen de la información en la interfaz principal de cada módulo.

Como resultado se obtuvo un bosquejo gráfico en el cual se estableció las áreas relevantes del layout (estructura) del sistema, cada elemento integrado dentro de la interfaz fue pensado para causar un efecto en el usuario y definido con un propósito en particular. Como podemos observar en la Figura 12, cada elemento permite responder a ciertas necesidades y preferencias de los usuarios, para proporcionar descripciones breves de la información correspondiente a cada módulo.

El layout se conforma por elementos de navegación, información, interacción y apoyo, que se describe a continuación:

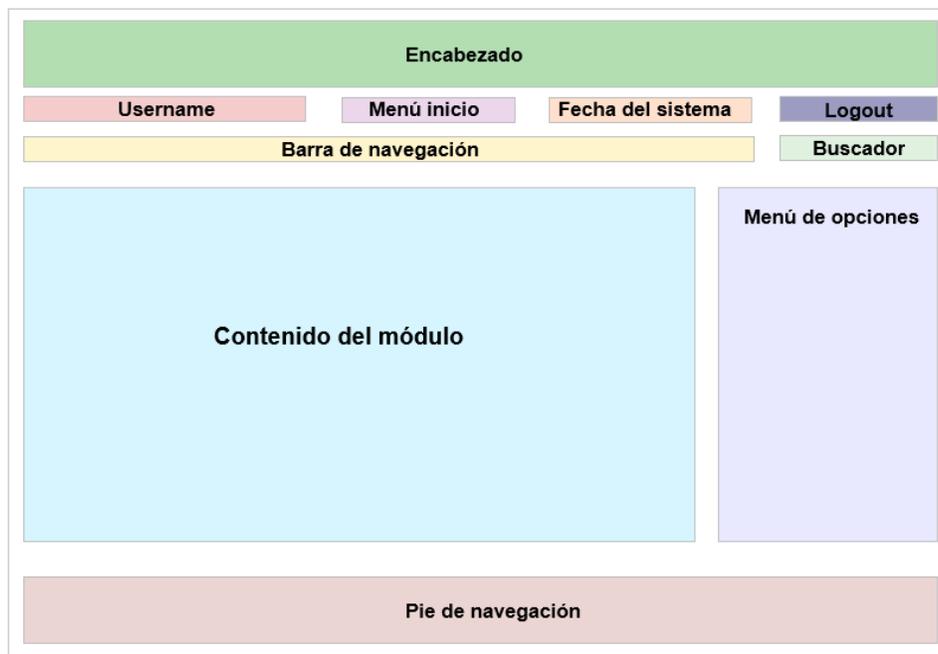


Figura 12 Layout del sistema

Elementos de navegación

Estos componentes se encuentran ubicados en la parte inferior de los elementos de apoyo, son reutilizados en todas las interfaces con el objetivo de conservar la consistencia visual del sistema, además permite al usuario acceder al panel de control principal y al buscador de forma rápida.

- **Menú inicio:** Esta opción tiene la finalidad de regresar al usuario al panel de administración a través de un acceso directo.
- **Barra de navegación:** Zona donde se ubican funciones primordiales del sistema, por ejemplo: crear nueva página, grupo, opción, etc.
- **Buscador:** Sección que permite relacionar palabras claves proporcionadas por el usuario con el contenido de la información presentada.
- **Pie de navegación:** Sección inferior del sistema, conformada por la paginación del contenido, con el propósito de exhibir información faltante en el caso de no ser presentada en una sola pantalla.

Elementos de información

Corresponde a la sección del layout que se encuentra orientado a la entrega de información.

- **Contenido:** Apartado donde se presenta la información de cada módulo, cambia constantemente dependiendo de la operación realizada por el usuario, en esta sección se podrá visualizar aspectos generales de las páginas, grupos de menús, archivos, etc.

Elementos de interacción

Permiten al usuario la realización de las actividades de administración del sitio.

- **Menú de opciones:** Menú lateral del sistema, tiene la finalidad indicar las opciones de administración de los contenidos de cada módulo, tales como eliminar, modificar, vista previa, activar o desactivar, entre otros.

Elementos de apoyo

- **Logout:** Permite al usuario cerrar la sesión.
- **Encabezado:** Sección de la interfaz de usuario ubicado en la parte superior, constituido por el nombre y logo del sistema. Cabe mencionar que

el tamaño del encabezado disminuye considerablemente en las interfaces secundarias, esto con el propósito de optimizar el área de contenido, debido a que están orientadas a presentar información detallada en cada módulo.

- **Username:** En esta área se muestra el nombre del usuario que ha iniciado sesión.
- **Logout:** Permite al usuario cerrar la sesión.

Diseñar el layout permitió resaltar y organizar los aspectos comunes que pueden ser utilizados en todas las pantallas que conforman el CMS, de esta forma se cumple con el objetivo de proporcionar homogeneidad a las interfaces, y conseguir que los elementos de trabajo establecidos cumplan con las actividades que el usuario espera realizar en el administrador.

BACK-END DEL SISTEMA

El back-end es el área privada del administrador de contenidos, se encuentra dividido en módulos que permiten controlar y gestionar todas las propiedades del CMS, estos módulos son extensiones del administrador que permiten añadir complementos en diferentes zonas de la plantilla del sitio Web, cada extensión funciona de forma independiente, por lo tanto es posible añadir o eliminar complementos sin alterar el funcionamiento del mismo, esto se debe a que el back-end interactúa directamente con la base de datos, cada extensión realiza una serie de consultas y el layout determina en que parte será presentada la información solicitada.

Cada uno de los módulos agrupa una serie de tareas que al llevarse a cabo permite cumplir los objetivos establecidos en las etapas anteriores. El hecho de que cada una de las extensiones funcione de manera independiente proporciona flexibilidad al administrador ya que es posible adaptarlo según las necesidades requeridas únicamente re-estructurando la plantilla del back-end.

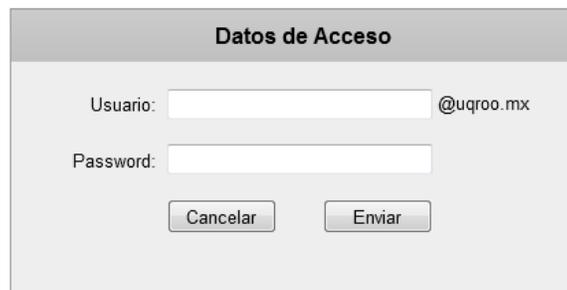
WIREFRAMES

Para establecer el contenido del back-end se realizó una serie de diseños simplificados conocidos como Wireframes, estos detallan la estructura y los elementos relevantes que se mostraran en pantalla y a su vez contribuyen de modo eficaz en la toma de decisiones relevantes acerca del contenido y la interacción, el hecho de utilizar wireframes en el diseño, permitió llevar a cabo pruebas de las interfaces y determinar si cumplen con los objetivos que se esperan conseguir al momento de su desarrollo.

A continuación se describe las funcionalidades que se debe de proveer al usuario final y los wireframes que se desarrollaron para establecer la interacción del sistema:

INDEX

Se crearon wireframes para las secciones más importantes del sistema, en el index o página principal del administrador, se requiere la autenticación del usuario (ver Figura 13) para acceder al sistema es necesario proporcionar los datos de acceso: cuenta de correo electrónico institucional y password, de igual forma es necesario estar registrado como usuario administrador, sino se lleva a cabo la validación es imposible acceder al CMS.



The image shows a wireframe for a login form titled "Datos de Acceso". It features two input fields: "Usuario:" followed by a text box and "@uqroo.mx", and "Password:" followed by a text box. Below the fields are two buttons: "Cancelar" and "Enviar".

Figura 13 Datos de Acceso

MENÚ PRINCIPAL

El menú principal permite agrupar en una interfaz los accesos a todas las funcionalidades útiles del CMS para la administración periódica del sitio. Si son agregadas nuevas extensiones al administrador, el panel se expande para incorporar un link de acceso al nuevo módulo.



Figura 14 Menú del sistema

MÓDULO ADMINISTRADOR DE PÁGINAS

El objetivo principal de esta sección es proporcionar al usuario un panel de administración que facilite la creación y mantenimiento de páginas Web, este módulo debe proporcionar las siguientes funcionalidades al usuario:

- Crear, eliminar, modificar páginas.
- Proporcionar el código fuente del contenido en formato HTML para usuarios familiarizados con este lenguaje de programación.
- Vista previa del contenido de las páginas.
- Realizar búsquedas en el listado de las páginas.
- Activar y desactivar la visualización de la página en el Front-end del sitio Web.
- Incorporación de estilos de fuentes.
- Agregar menú de navegación.
- Insertar hipervínculos, imágenes, tablas.
- Establecer dependencias entre páginas.
- Agregar palabras claves.
- Insertar imagen de fondo y encabezado.
- Jerarquizar la información presentada.
- Validación LDAP en páginas.
- Creación automática de indicadores uniformes de recursos (URI).
- Optimización de las páginas para motores de búsqueda.

PANEL DE ADMINISTRACIÓN DEL MÓDULO DE PÁGINAS

En base a lo anterior se desarrolló el wireframe del administrador de páginas, en la Figura 15 se puede observar un panorama general de las páginas que conforman el sitio Web, la información a visualizar es el ID y nombre, se pretende que al dar clic sobre el signo [+] se despliegue una lista de páginas dependientes, esto con la finalidad de agrupar y reducir el contenido a presentar en la interfaz, cada una de las páginas contara con opciones de eliminar, modificar, vista previa y activar/desactivar.

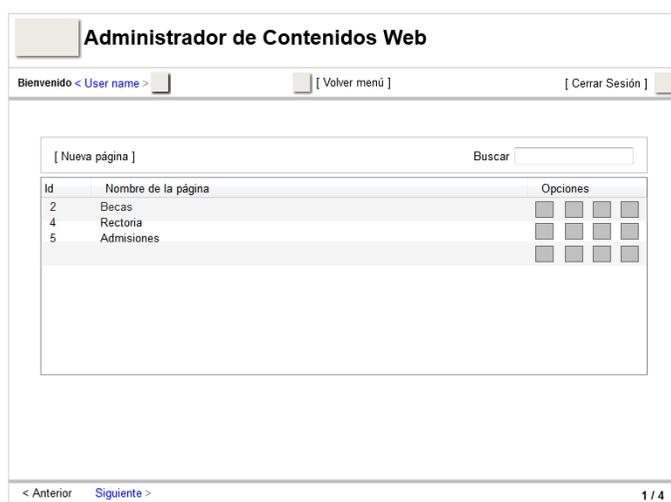


Figura 15 Administrador de páginas

EDITOR DE PÁGINAS

Al momento de crear o modificar una página el panel de administración se dirige al editor (Ver Figura 16), en esta sección se presenta la configuración detallada de la página, las primeras cuatro opciones se establecieron con la finalidad de identificar la página dentro del administrador, por otra parte agregan propiedades en los metadatos con el objetivo de facilitar a los motores de búsqueda, información sobre el sitio.

- **Nombre corto:** conforma el URI de la página, se utiliza para añadir el TITLE del html, es decir, definir el título de la página en la barra de navegador,

además de establecer el nombre que aparece en la lista de páginas dentro del back-end.

- **Nombre Largo:** Nombre que se visualiza en el encabezado del Front-end .

En las opciones avanzadas encontramos las siguientes:

- **Descripción:** Resumen general de la página, es utilizado por el meta description del Front-end para describir el contenido o tema de la página en los motores de búsqueda.
- **Palabras claves:** Usado para construir el meta keywords de la página, con el objetivo de optimizar el posicionamiento en buscadores.
- **Dependencia:** Define dependencias entre páginas anidadas, conformar la URI.
- **Encabezado:** Es un elemento opcional donde se puede colocar una imagen como encabezado en lugar del nombre largo.

Con la finalidad de proveer un entorno fácil de usar, se anexa una barra de herramientas en la interfaz del editor, esta barra debe contener opciones similares a un procesador de texto que permita cambiar el tamaño y estilo de las fuentes, insertar imágenes, crear tablas, insertar o editar hipervínculos, etc.

Las herramientas que proporciona deben facilitar al usuario las actividades de administración sin dificultades. Por último en la parte central de la interfaz encontramos el editor HTML, esta región es el lienzo de la página, el objetivo principal de esta área es agregar o visualizar el contenido de la página Web.



Figura 16 Editor de páginas

MÓDULO DE MENÚS

Este módulo debe permitir la creación e incorporación de menús en las páginas Web, así mismo proporcionar la capacidad de configurar cada menú de forma independiente de tal manera que sea posible agregar opciones diferentes a cada grupo.

Funcionalidades

- Crear, editar y eliminar grupos de menús.
- Crear, editar y eliminar opciones pertenecientes a grupos.
- Permitir cambiar la posición de cada opción.
- Las opciones podrán abrir:
 - Archivos
 - Páginas
 - URLs
 - Grupos de menús

Los menús se componen de opciones, que son enlaces a páginas, que pueden estar dentro o fuera del sitio. Estos elementos se muestran ordenados o anidados, de tal forma que se tiene la posibilidad de construir menús con diferentes niveles de profundidad, cada opción puede estar ligado a: páginas Web, archivos, URLs o a un grupo de menú diferente.

Cabe mencionar que el panel de administración de menús y de opciones es similar al panel del administrador de páginas, en ella se debe mostrar una lista con los grupos de menús creados, y por separado una lista de opciones del menú donde se ordenen y activen los elementos ya creados.

EDITOR DE GRUPOS

En la Figura 17 observamos el editor de grupos, en este wireframe se define aspectos específicos del menú, entre ellos la denominación, que indica el nombre o identificador del grupo dentro del CMS. Este identificador será usado por el sistema a nivel interno, cabe mencionar que no es el título o nombre del menú que se mostrará en el sitio. El encabezado por el contrario es el nombre del sitio que utiliza el menú o al que pertenece, se encuentra definido por una imagen o por un espacio en blanco, el encabezado interactúa directamente con el administrador de archivos ya que en esta sección es donde se podrá almacenar o subir la imagen al servidor.

El último elemento es el pie de página, se ubica al final de todas las opciones del menú, está conformado por una página secundaria inactiva, es decir, que la información a visualizarse es el contenido de una página que no será visible en el Front-end o en el mapa del sitio, pero si se encontrara listada en el panel de administración del módulo de páginas.



The screenshot displays the 'Administrador de Contenidos Web' interface. At the top, there is a header with the title 'Administrador de Contenidos Web' and a navigation bar containing 'Bienvenido < User name >', '[Volver menú]', and '[Cerrar Sesión]'. Below the header, there is a secondary navigation bar with links for '[Lista de Grupos]', '[Crear grupo]', '[Crear Opción]', and a search field labeled 'Buscar'. The main content area is titled 'Editor de grupos' and contains three input fields: 'Denominación', 'Encabezado' (with a file upload icon), and 'Pie de página' (with a dropdown menu set to 'Ninguno'). To the right of these fields are three buttons: 'Cancelar', 'Guardar', and 'Guardar y Salir'.

Figura 17 Editor de grupos

EDITOR DE OPCIONES

Con la finalidad de cubrir los requerimientos del módulo de submenú se planteó el prototipo “editor de opciones” que podemos observar en la Figura 18, el cual se divide en dos secciones: Datos generales que contiene el nombre de la opción (etiqueta), el tipo (opción o separador) puede ser un título que aparecerá en el menú o un enlace, por último se encuentra el elemento para seleccionar el grupo al que va a pertenecer la opción.

La segunda sección se conforma por los datos de link (archivo, página, URL o menú), en esta sección la opción creada puede estar ligada a un archivo, una página Web, un URL o un menú creado con anterioridad este último proporciona los niveles de profundidad del menú.



The screenshot shows the 'Editor de opciones' form within the 'Administrador de Contenidos Web' application. The interface includes a header with the application name, a user login area, and navigation links. The main form area contains a navigation bar with links for '[Lista de Grupos]', '[Crear grupo]', and '[Crear Opción]', along with a search field. The 'Editor de opciones' form itself has the following fields:

- Etiqueta:** A text input field.
- Tipo:** A dropdown menu currently set to 'Opción'.
- Destino:** A dropdown menu with the text 'Seleccionar destino'.
- Archivo:** A radio button option with a file selection icon.
- Página:** A radio button option with a dropdown menu 'Seleccionar página'.
- Menú:** A radio button option with a dropdown menu 'Seleccionar menú'.
- URL:** A radio button option.
- Grupo:** A dropdown menu with the text 'Seleccionar grupo'.

At the bottom of the form are three buttons: 'Cancelar', 'Guardar', and 'Guardar y salir'.

Figura 18 Editor de opciones

MÓDULO DE ARCHIVOS

El módulo de archivos tiene como objetivo principal llevar a cabo la administración de directorios y archivos del sitio Web, debe poner a disposición los recursos del sistema para su interacción con los demás módulos.

Funcionalidades

- Colocar archivos en el servidor.
- Fácil administración de los archivos y directorios.
- Crear directorios.
- Copiar, mover, eliminar, renombrar archivos.
- Visualización previa de archivos con formato jpg, gif o html.
- Listar las propiedades de los recursos.
- Asignación de almacenamiento físico.
- Archivos ZIP.

Panel de administración del módulo de archivos

El panel de administración de archivos se divide en tres secciones, en la parte superior se colocó una barra de herramientas para proporcionar al usuario opciones para la manipulación archivos y directorios, en el área central se debe listar la información general de los recursos alojados en el servidor y los que al usuario puede acceder si cuenta con permisos suficientes, por último en la zona lateral derecha se puede observar las propiedades avanzadas, tales como tamaño del recurso, fecha de creación y/o última modificación, y las características (permisos) de acceso.



Figura 19 Panel de administración de archivos

MÓDULO DE AVISOS, EVENTOS, EVENTOS EXTERNOS, CONVOCATORIAS Y NOTICIAS

Funcionalidades

- Crear y/o modificar publicaciones.
- Establecer periodos de vigencia para cada publicación.
- Crear páginas (opcional) por cada publicación realizada.
- Adjuntar archivos.
- Almacenar en una carpeta especifica los archivos adjuntos.
- Generar una lista con publicaciones caducadas.

Durante el diseño de las extensiones del sistema se identificó características o comportamientos comunes en relación a las funcionalidades y aspectos generales de los módulos de avisos, eventos, eventos externos, convocatorias y noticias, por lo tanto se generó un panel de administración con características similares para cada módulo.

Panel de administración

Con base a las características anteriores se desarrolló el siguiente wireframe en el cual se presenta un panel de administración con el resumen general del módulo, en donde se podrá encontrar el Id, denominación, fecha de publicación y vencimiento, el correo del responsable de la información a publicar, el nickname del usuario que realizo el servicio y por último el estatus de la publicación (activo, inactivo, nuevo), estas opciones son de gran importancia, debido a que dependiendo al estatus en el que se encuentre la publicación, se visualiza en el Front-end , por ejemplo si la publicación se encuentra inactiva, no es posible apreciarla en el sitio Web ya que puede estar siendo modificada, el estatus nuevo significa que la publicación es reciente y es posible visualizarla en el sitio.

En la Figura 20 podemos observar en la parte inferior la barra de opciones, en cada módulo se podrá crear y modificar las publicaciones, los botones de subir y

bajar sirven para ordenar la posición de los mismos, cabe mencionar que de acuerdo a la forma en que se encuentren ordenas será como se visualizará en el Front-end .

#	Denominación	Fecha publicación	Fecha vencimiento	Solicitó	Publicó	Status
593	Becas PRONABES 2012-2013 .	2012-07-25	2012-09-07		Rosa	Activo
592	Programa nacional de inglés para todos	2012-07-25	2012-09-10	dramirez	Rosa	Activo
591	Inglés para Todos .	2012-07-25	2012-09-10	rolando	Rosa	Activo
580	Programa de Servicio Social UQROO-PE	2012-06-05	2012-09-14	dafne	Rosa	Activo
578	I Coloquio de Investigación de la DCSEA	2012-06-05	2012-08-31	jlesparza	Rosa	Activo
565	Programa de Apoyo a la Titulación .	2012-05-09	2012-10-19	magnolia	Rosa	Activo
556	Maestría en Economía del Sector Público	2012-04-26	2012-08-31	ladame	Rosa	Activo
537	XXII Coloquio Mexicano de Economía Mat	2012-03-09	2012-08-24	smonroy	Rosa	Activo
524	Examen General al Egreso .	2012-03-06	2012-12-14	rosescalante	Rosa	Activo
428	Elecciones Cozumel .	2011-10-11	2012-10-25	rebeca	Rosa	Activo

Figura 20 Panel general de los módulos de publicación

La opción “caducados” proporciona una lista de las publicaciones no vigentes, es decir que su periodo de publicación ha concluido, en la Figura 21 se muestra el aspecto visual de esta interfaz, el cual cuenta con características similares del panel de administración principal, pero sin la barra de opciones inferiores.

#	Denominación	Fecha publicación	Fecha vencimiento	Solicitó	Publicó	Status
593	Becas PRONABES 2012-2013 .	2012-07-25	2012-09-07		Rosa	Activo
592	Programa nacional de inglés para todos	2012-07-25	2012-09-10	dramirez	Rosa	Activo
591	Inglés para Todos .	2012-07-25	2012-09-10	rolando	Rosa	Activo
580	Programa de Servicio Social UQROO-PE	2012-06-05	2012-09-14	dafne	Rosa	Activo
578	I Coloquio de Investigación de la DCSEA	2012-06-05	2012-08-31	jlesparza	Rosa	Activo
565	Programa de Apoyo a la Titulación .	2012-05-09	2012-10-19	magnolia	Rosa	Activo
556	Maestría en Economía del Sector Público	2012-04-26	2012-08-31	ladame	Rosa	Activo
537	XXII Coloquio Mexicano de Economía Mat	2012-03-09	2012-08-24	smonroy	Rosa	Activo
524	Examen General al Egreso .	2012-03-06	2012-12-14	rosescalante	Rosa	Activo
428	Elecciones Cozumel .	2011-10-11	2012-10-25	rebeca	Rosa	Activo

Figura 21 Panel de publicaciones no vigentes

Para editar la publicación es necesario marcar como seleccionado la casilla del evento y posteriormente dar doble clic para ir a la vista previa del evento (Ver Figura 22), se desarrolló este wireframe con el objetivo de permitir al administrador visualizar el aspecto que tendrá la publicación, antes de ser colocado en el sitio Web, cuenta con dos opciones: “modificar” el cual nos dirige a la configuración global del evento y la opción “volver a bandeja” que es un acceso directo al panel de administración.



Figura 22 Vista previa de la publicación

FRONT-END

El Front-end conforma la parte pública del CMS, es decir el lugar donde se visualiza toda la información generada y administrada por el back-end, es decir, el contenido perteneciente a las páginas del portal universitario que puede ser consultado por cualquier usuario en la Web. La forma en que se mostrarán dichos contenidos está determinada por el diseño de la plantilla que apliquemos al sistema.

Plantilla

Una plantilla es el elemento responsable de definir la estética del sitio Web. En ella se establece cómo ha de mostrarse la información que cada módulo del sistema provee. En otras palabras, tiene la función de separar la estructura del contenido, esta propiedad permite modificar el diseño sin afectar el contenido que presenta, simplemente se adapta a la nueva presentación.

Estructura de navegación

Como parte del diseño del Front-end se desarrolló un conjunto de elementos que se encuentran presentes en cada una de las pantallas, que permiten al usuario navegar por las diferentes secciones del Portal Web. La estructura de navegación cuenta con los siguientes elementos que se describen a continuación:

- *Navegación global:* se conforma por una zona en la plantilla que debe permitir el acceso a las secciones o categorías en las que se encuentra dividida la información del portal universitario, se ubica en la parte superior y contiene las siguientes opciones: Nuevo ingreso, comunidad universitaria, titulación y egresados, vinculación, sistemas y finalmente debe de contar con la opción de inicio, para acceder al index del portal.
- *Navegación local:* Son elementos agrupados que constituyen el menú principal de la página o de cada sitio, realizar acciones relativas al contenido o sección al que pertenecen.

- *Navegación contextual*: Permite navegar desde el contenido mostrado en pantalla a contenidos relacionados, de igual forma en esta área se puede presentar el contenido de las páginas.
- *Navegación por cortesía*: Región inferior de la platilla del Front-end que se encarga de completar la información mostrada en las zonas superiores de la navegación.
- *Breadcrumbs o "migas de pan"*: Son elementos utilizados para informar al usuario en que parte del sitio Web se encuentra, se construye con las dependencias establecidas en el módulo de páginas del back-end, así mismo indican un posible camino desde la página de inicio hasta la página actual por donde el usuario se puede desplazar para encontrar información general.

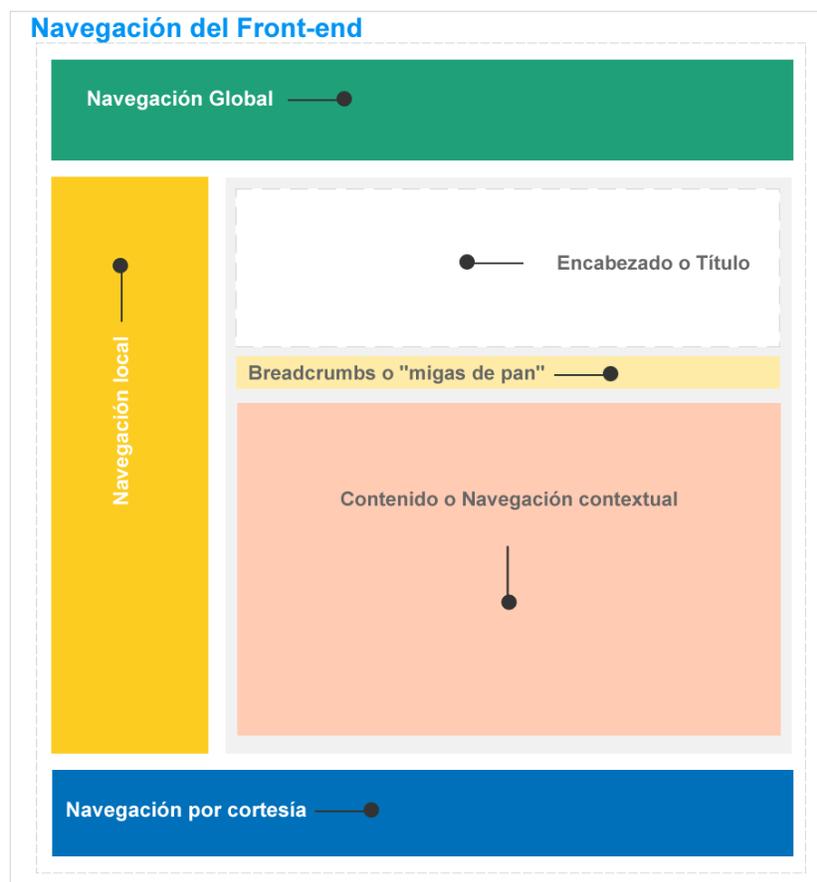


Figura 23 Navegación del front-end

Área de contenidos

Para determinar el área de contenidos se tomó en cuenta la estructura de navegación anterior, para poder definir los marcadores de posición donde se sitúan los componentes que cada módulo ofrece, como podemos ver en la Figura 24 se encuentra señalado las áreas donde se ubica los contenidos creados en cada módulo, cabe mencionar que al modificar o crear plantillas podemos agregar o no complementos, en otras palabras, se puede modificar los marcadores sin alterar el funcionamiento del CMS.

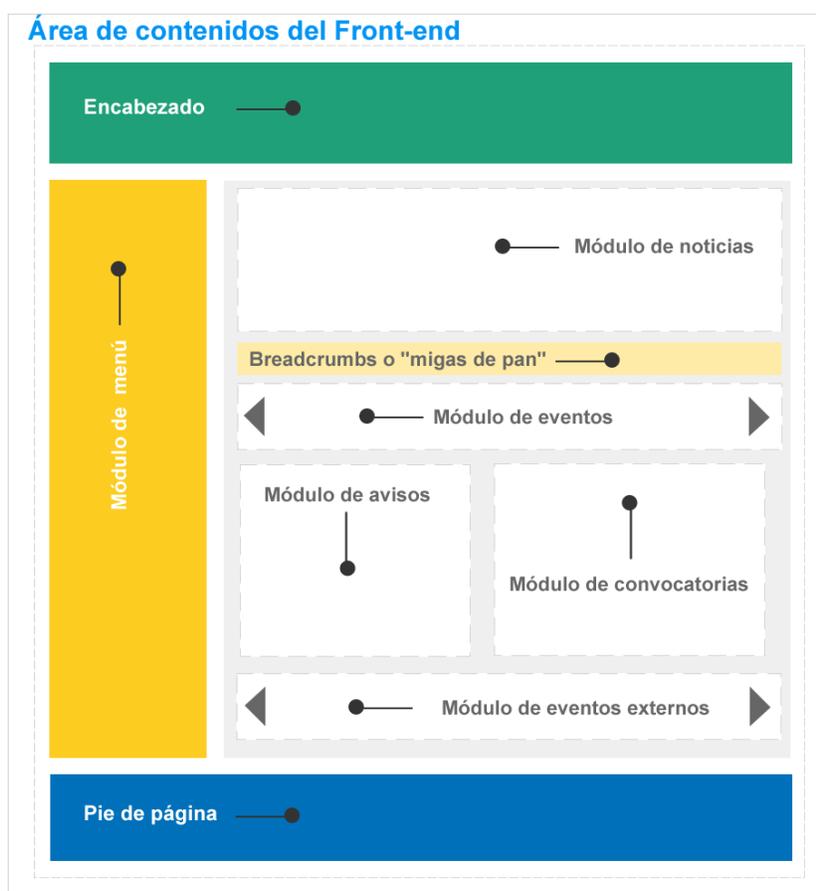


Figura 24 Área de contenidos del front-end

ESTRUCTURA DE LA BASE DE DATOS

La estructura de la base de datos (BD) juega un papel fundamental para el administrador de contenidos, debido a que la mayoría de la información presentada tanto en el Front-end y back-end se realiza a través de consultas a la base de datos, es por esta razón que el sistema debe permitir manejar de manera independiente el contenido y el diseño, es decir, que en cualquier momento se puede realizar un diseño distinto al sitio sin tener que darle formato a todo el contenido de las páginas.

La finalidad de almacenar la información del sistema en una base de datos, es mantener los datos organizados para una mejor manipulación de ellos, llevar a cabo respaldos de una forma rápida y segura. Es por ello que se debe diseñar correctamente la BD para permitir el acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados con anterioridad.

En la Figura 25 se presenta el modelo lógico de la base de datos que soportará al sistema administrador de contenidos, este modelo ha sido diseñado tomando como referencia el desarrollo de los wireframes anteriores, los cuales permitieron establecer las tablas y campos necesarios para almacenar la información de cada módulo y las especificaciones necesarias para la construcción de la misma.

A continuación se presenta una descripción de las tablas que se crearon para almacenar los de datos del CMS.

Tabla 1 Descripción de tablas del CMS

	Nombre de la Tabla	Descripción
	paginas	Almacena el contenido y la información general de las páginas Web creadas.
	Etiquetas2	Las opciones generadas para cada grupo de menú se encuentran almacenadas en esta tabla.
	grupos	Almacena los datos de los grupos creados con el administrador de submenús.

	publica_eventos	La información referente a los eventos a publicar se encuentra almacenada en esta tabla.
	publica_pag_eventos	Contiene los datos generales y el contenido de las páginas de los eventos publicados.
	Notas	Almacena el resumen y los contenidos de las notas creadas en el módulo de noticias.
	Calificaciones	Guarda el historial de calificaciones y comentarios de cada noticia publicada en el sitio de sala de prensa.
	Log	El registro de las acciones realizadas en el módulo de noticias se encuentra almacenado en esta tabla.
	avs_avisos	Almacena los datos generales de los avisos publicados.
	avs_etiquetas	Guarda la información de acceso a los recursos del aviso, como dirección de páginas Web, archivos o URL.
	avs_paginas	Almacena los datos generales y el contenido de las páginas pertenecientes a los avisos publicados.
	conv_convocatorias	La información general de las convocatorias tales como: Fecha de publicación, el estatus y la descripción se encuentra almacenado en esta tabla.
	conv_paginas	Guarda el contenido HTML de las páginas de cada convocatoria.
	publica_eventos_externos	Registra la información general de los eventos externos.
	publica_pag_eventos_ext	Almacena los datos generales y el contenido de las páginas de los eventos externos publicados.

Existen diferentes modelos de organización de base de datos, para el CMS se empleó el modelo Entidad-Relación (E-R), es uno de los diagramas que más se utiliza actualmente, y proporciona una noción diagramática muy comprensiva que ayuda a reflejar en un modelo conceptual la estructura de la BD.

La creación de un diseño esquemático del sistema, tiene como objetivo revisar el funcionamiento deseado e identificar duplicidad de información y evitar conflictos entre llaves primarias.

Podemos apreciar en la Figura 25, se creó diagramas de las tablas descritas anteriormente, se dividieron en secciones según el módulo al que pertenecen para tener un orden específico, en cada tabla se estableció una llave primaria (Primary Key o PK) o identificador (id) cuyos valores identifican de forma única cada fila de la tabla. Las llaves externas (Foreign Key o FK) se usan para establecer un vínculo entre los datos de dos tablas esto se da cuando las columnas de una tabla hacen referencia a las columnas de otra que contiene el valor de la clave principal.

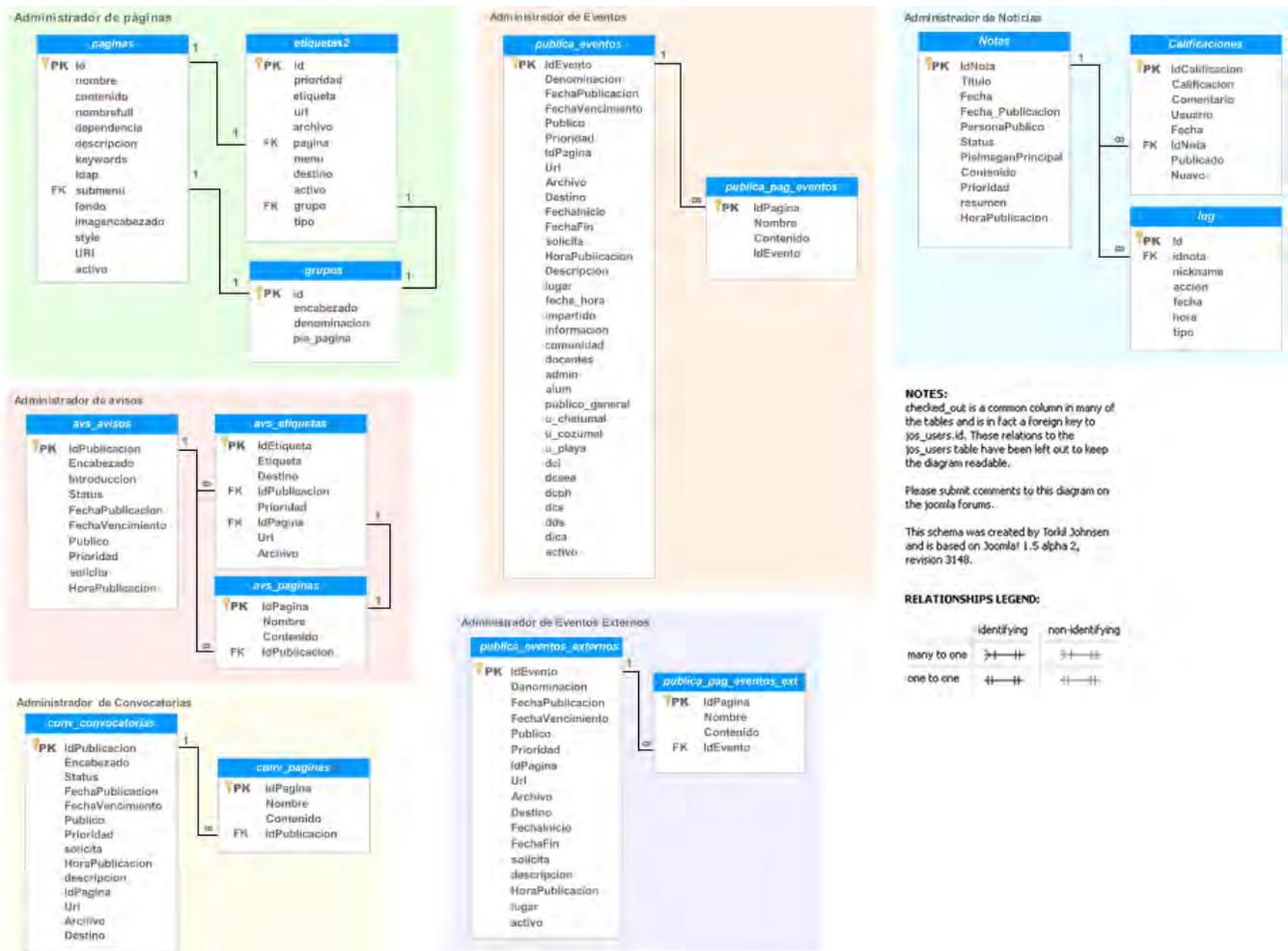


Figura 25 Diagrama Entidad-Relación

Por ejemplo, la tabla **paginas** de la base de datos “uqroo2” tiene un vínculo a la tabla **etiquetas2** porque existe una relación lógica entre las lista de menús y el submenú que la pagina despliega en el Front-end.

CODIFICACIÓN

Después de haber concluido con las consideraciones respecto al diseño de interfaces y la estructura de la base de datos, se procedió con la etapa de codificación del sistema, según lo establecido en el ciclo de vida del software en la metodología del proyecto.

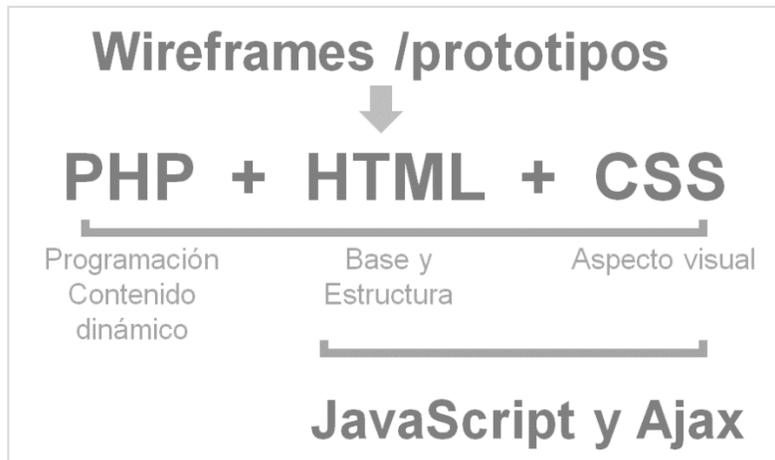


Figura 26 Codificación

Esta etapa es de suma importancia ya que el propósito principal es satisfacer los objetivos y requerimientos descritos en el análisis y diseño del sistema, tal como se citó en las etapas anteriores. Para realizar el proceso de codificación intervinieron las siguientes herramientas: PHP como lenguaje de programación, MySQL base de datos, PHPMyAdmin administrador de la base de datos, Dreamweaver editor de etiquetas HTML, CSS apariencia visual, Javascript y AJAX para realizar interacciones en segundo plano al servidor.

La codificación del back-end se realiza por módulos independientes, lo cual permite encapsular las funcionalidades y relaciones que existen entre cada elemento, de esta manera se delimita su comportamiento y la interacción con algún otro modulo, para no afectar el funcionamiento general de todo el sistema y permitir en un futuro incorporar nuevas funciones.

DREAMWEAVER

Entre las herramientas utilizadas para la codificación del CMS se encuentra Dreamweaver, que proporciona una interfaz visual intuitiva para la creación y edición de sitios Web basados en HTML. Específicamente se utilizó la versión CS5 en el cual se estableció una conexión FTP (File Transfer Protocol) entre la PC local y el servidor remoto, para transferir y alojar los archivos del sistema al servidor Web, de igual forma realizar pruebas y depuración del código fuente.

Dreamweaver permitió visualizar de forma gráfica la apariencia aproximada de la interfaz Web antes de alojarlo en el servidor, manipular las propiedades de las etiquetas HTML, y a su vez incorporar bloques de código PHP en la estructura HTML, con la finalidad de realizar consultas SQL y validación de variables recibidas a través de los formularios.

PHP

Se optó el lenguaje de programación PHP para la codificación del CMS, debido a que es un lenguaje de programación interpretado, diseñado para la creación de páginas Web dinámicas, sus características han ido creciendo hasta convertirse en un lenguaje completo, capaz de manejar entornos que integran volúmenes de información bastante amplios alojados en bases de datos, multiplataforma, compatible con la mayoría de los sistemas de base de datos en especial con MySQL, permite incrustar código HTML en la estructura de los programas, es decir, PHP puede crear HTML y viceversa.

PHPMysqlAdmin

La gestión de la base de datos se realizó a través de la aplicación PHPMyAdmin, programa de software libre escrito y ejecutado en PHP, el cual se aloja en el servidor MySQL, esta herramienta proporciona una interfaz Web amigable que permite acceder a las tablas y registros de la BD para su modificación, por otra

parte, crea y administra usuarios y privilegios, lo que provee seguridad en el acceso de la base de datos.

TIPOS DE USUARIO

Una parte relevante dentro del desarrollo de un sistema, son las cuestiones referentes a la seguridad de la información que interviene durante el uso del CMS, tanto en el control de usuarios, despliegue de información, y delimitación de privilegios, para ello se optó definir tres tipos de usuarios, “root” , “admin” y “user” responsables de la manipulación del front-end y back-end.

El usuario “**root**”, es utilizado de manera interna, debido a que cuenta con permisos suficientes para establecer conexiones con el servidor, crear y editar la base de datos, incorporar nuevos usuarios y modificación del código fuente del sistema interno y del portal universitario.

Por el contrario, el perfil “**admin**” hacen referencia a los nicknames de los administradores que serán registrados para la publicación de contenidos en los múltiples sitios Web universitarios, por lo tanto, no cuenta con acceso a ningún fragmento de código, ya que solo interactúan con las interfaces gráficas del CMS, las cuales generan únicamente el contenido de las páginas Web. Para poder agregar nuevos administradores es necesario que cuenten con correo electrónico institucional, debido a que este les provee el nickname de autenticación que será utilizado posteriormente para la validación con el servidor LDAP.

Los “**users**” acceden a la parte del front-end, es decir, únicamente navegan a través de las páginas del sitio Web. Desplegando información de acuerdo a la petición que realicen al servidor. Cuentan con el permiso de solo lectura, imposibilitando cualquier otra acción en la parte del front-end.

Estas restricciones proporcionan un medio para asegurar la integridad y consistencia de los datos almacenados en la BD, llevar un control de usuarios administradores y delimitar los perfiles de usuario.

INICIO

En base a las consideraciones anteriores, se procede con la codificación del CMS, como primer punto se presenta en la Figura 27 el formulario de autenticación para el acceso al sistema, en el cual se solicita el nickname y password, datos que serán enviados al servidor para identificar los usuarios registrados en la base de datos y en el servidor LDAP, esta restricción provee el control de la información que será publicada y agregada a la base de datos.



The image shows a login interface for the 'Administrador de Contenidos Web' of the 'Universidad de Quintana Roo'. At the top, there is a green header with the university's logo and name on the left, and the title 'Administrador de Contenidos Web' on the right. Below the header is a white login form titled 'Datos de Acceso'. The form contains two input fields: 'Usuario:' with a placeholder '@uqroo.mx' and 'Password:'. At the bottom of the form are two buttons: 'Cancelar' and 'Enviar'.

Figura 27 Formulario de Inicio

Para llevar a cabo este proceso es necesario la intervención de los siguientes archivos: ***parametros.php***, ***lib.php*** y ***valida.php***, los cuales se encargan de establecer la conexión y la autenticación. Como medida de seguridad el acceso al back-end del sistema se lleva a cabo a través del protocolo HTTPS (HyperText Transfer Protocol Secure), el cual nos permite proteger la integridad y la confidencialidad de la comunicación, HTTPS crea un canal cifrado SSL/TLS entre el servidor y el navegador del cliente, con la finalidad de aumentar la seguridad en el tráfico de información.

CONEXIONES

La configuración requerida para realizar la conexión del CMS, se encuentra en el archivo ***parametros.php*** que puede apreciarse en la Figura 28, en ella se observa la estructura del archivo, contiene un conjunto de variables que almacena la información necesaria para establecer los vínculos al servidor tales como, dirección IP del servidor LDAP ***\$dir_server***, directorio para el alojamiento de los archivos del CMS ***\$PATH_SERVIDOR***, usuario y contraseña para la autenticación con el servidor Web y por último el nombre de la base de datos ***\$BASE_DATOS***.

```
parametros.php
1  <?php
2  //Datos conexion servidor LDAP
3  $dir_server = "****.*.*.*.*** 389"; //Direccion IP servidor LDAP
4  $base_dn = "ou=people,o=uqroo.mx,o=isp"; // Base DN
5  //Configuracion del sitio
6  $URL_SITIO="http://www.uqroo.mx";
7  $PATH_SERVIDOR='/opt/sitios/nuevo/';
8  //Datos conexion servidor web
9  $SERVIDOR = "localhost";
10 $USUARIO = "****";
11 $PASSWORD = "*****";
12 $BASE_DATOS = "uqroo";
13 ?>
```

Figura 28 **parametros.php**

El contar con archivos de configuración como ***parametros.php*** permite en un futuro poder implementar un archivo autoejecutable para la instalación del CMS, es decir, que la configuración interna se realice a través de interfaces gráficas para su fácil comprensión por el usuario. Cabe mencionar que el usuario *root* es el único que puede realizar modificaciones en los datos almacenados en ***parametros.php***.

Después de determinar los parámetros de configuración global, se realiza la conexión con el servidor donde se aloja la base de datos, para ello se desarrolló el archivo ***lib.php***, donde se invoca la función ***mysql_connect*** en el cual se envía el nombre del servidor, usuario y contraseña definidos anteriormente en “parametros.php”, esta función devuelve *true* si fue satisfactoria la conexión, o *false* si se produjo algún error, este resultado se asigna a la variable ***\$Link***.

Como podemos apreciar en la Figura 29 una vez establecida la conexión, se indica la base de datos a utilizar, mediante la función *mysql_select_db* enviando el nombre de la base de datos y el valor obtenido de la variable \$Link.

```
lib.php
1  <?php
2  function conectaBD(){
3  include("parametros.php");
4  //Conexion servidor Web
5  $Link      =      @mysql_connect($SERVIDOR,$USUARIO,$PASSWORD)      or      die
6  (mysql_error()." Error al conectar con el servidor");
7  mysql_query("SET NAMES 'utf8'", $Link);
8  //Conexion base de datos
9      @mysql_select_db($BASE_DATOS,$Link) or die ("Error en la conexión
10 con la Base de datos");
11 return $Link;
12 }
13 ?>
```

Figura 29 Archivo lib.php

Si la conexión no se estableció, se recibe una notificación de error **“Error en la conexión con la base de datos”**.

Para continuar con la fase de autenticación se definió **valida.php**, como se había mencionado con anterioridad, es necesario que el administrador del sitio Web, cuente con correo electrónico institucional para la validación a través del servidor LDAP que se encarga de almacenar un directorio en el cual se contiene la información relevante de los usuarios. Prácticamente el funcionamiento básico de LDAP es realizar una conexión con el servidor Web, autenticarse, realizar búsquedas en el directorio, interpretar el resultado y cerrar la conexión.

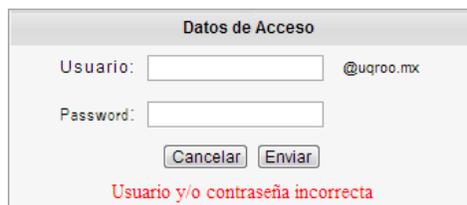
En la línea 10 del fragmento de código de valida.php se realiza la comprobación de las variables con el objetivo de no enviar valores nulos al servidor LDAP, si la condición se cumple se procesa el siguiente bloque de sentencias, en donde se envía los valores obtenidos del formulario de autenticación para verificar el registro en el servidor.

Valida.php

```
1  <?php
2      //Inicio de sesion
3      session_start();
4      include "parametros.php";
5      include "ldap/class.AuthLdap.php";
6      include "lib.php";
7      //Variables de usuario para validacion LDAP
8      $LDAP_Username = $_POST["usuario"];
9      $LDAP_Password = $_POST["pass"];
10     if(isset($_POST["enviarformbitacora"]) and $LDAP_Username!=" " and
11     $LDAP_Password!=""){
12         //variables de validacion LDAP
13         $ldap = new AuthLdap();
14         $server[0] = $dir_server;
15         $ldap->server = $server;
16         $ldap->dn = $base_dn;
17         //Establecer conexion servidor LDAP
18         if($ldap->connect() {
19             if($ldap->checkPass($LDAP_Username,$LDAP_Password)&&
20             !empty($LDAP_Username)){
21                 $Link = conectaBD();
22                 //validar usuario registrado
23                 $resultado = mysql_query("SELECT * FROM usuarios WHERE
24 Nickname= '$LDAP_Username'", $Link);
25                 $num_r=mysql_num_rows($resultado);
26                 $rowData = mysql_fetch_row($resultado);
27
28         <?php
29         //Cerrar conexion servidor LDAP
30         $ldap->close();
31     }else{
32         header("Location:index.php");
33     }
34     ?>
```

Figura 30 Archivo valida.php

Posteriormente si los parámetros de conexión fueron correctos se establece un enlace con el servidor, en el cual se efectúa una segunda validación, es decir, si el usuario se encontró registrado en el servidor LDAP, se procede a consultar la tabla “usuarios” de la base de datos. Si alguna de las dos validaciones devuelve un valor “False” como respuesta, no se permite el acceso al sistema, enviando al usuario un mensaje de error.



Datos de Acceso

Usuario: @uqroo.mx

Password:

Usuario y/o contraseña incorrecta

Figura 31 Formulario inicio

PANEL DE CONTROL

Si el usuario fue autenticado, se permite el acceso al back-end mostrando el panel de control, donde se encuentran los accesos directos a los módulos que conforman el CMS, tales como la creación y administración de páginas Web, la administración de eventos, avisos, etc. De esta forma se agrupa en una sola interfaz, todos los accesos de las múltiples funcionalidades del sistema que se consideran de suma importancia para la administración periódica del sitio.

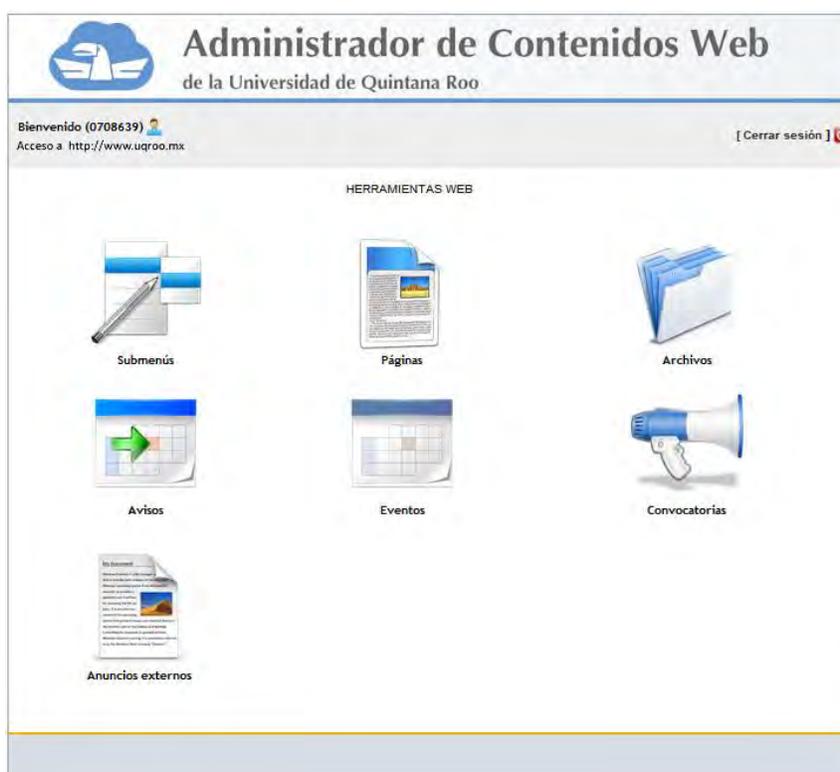


Figura 32 Panel inicio

Cabe mencionar que las opciones del sistema varía en función a los módulos activados en el CMS, es decir si el sitio Web solo requiere la administración de páginas se activa las opciones relacionadas con la publicación de contenido, tales como los módulos de “paginas”, “submenú” y “archivos”, sin que el sistema presente algún tipo de inestabilidad, ya que cada módulo es independiente.

FRAME

La interfaz principal de cada módulo se distribuye en “marcos” o cuadros que dividen la ventana en dos o más secciones diferentes, de modo que cada una recibe una estructura distinta, teniendo como resultado un contenedor principal que almacena las secciones de encabezado y contenido.

En la parte superior del contenedor se ubica la división denominada encabezado, en el cual se visualiza el archivo **encabezado.php** que contiene la barra de navegación general, el Nickname del usuario y el nombre del sitio web que se está administrando, en el apartado de contenido se despliega la información de cada módulo, es decir, el resultado de las consultas SQL realizadas por el usuario.

MODULO PÁGINAS

Cumpliendo con los objetivos específicos del sistema y en base a los wireframes desarrollados en la etapa del diseño, se procedió con la codificación del módulo administrador de las páginas Web, que permitirá construir la información del portal universitario, es decir el front end.

Se determinó la creación de los archivos **paginas_menu.php** el cual presenta un listado de las páginas que el usuario *admin* crea y edita, este archivo trabaja en conjunto con archivo **class.php** que es el encargado de realizar las consultas SQL a la base de datos, para extraer los datos identificadores las paginas, por ejemplo id, nombre y las paginas dependientes (hijos), como podemos observar en la Figura 33 se crea una clase denominada **web** extrae todas las paginas cuya dependencia sea igual al id de la página padre, y el método **verificaHijos** obtiene las paginas dependientes, es decir las páginas pueden ubicarse en varios niveles, el primer nivel es el “padre”, a partir de este nivel se generan las dependencias para las páginas posteriores.

Cada página puede tener varios hijos, que a su vez pueden ser los padres de otros hijos. Esta organización facilita y agiliza al máximo la búsqueda de

información, al mismo tiempo, permite las actualizaciones futuras de la Web, sin afectar drásticamente el contenido de otras páginas.

```
Class.php
1  <?php
2  class web{
3      var $link;
4      function getPaginas($id){
5          return($this->execSQL("SELECT * FROM paginas WHERE dependencia=$id
6  ORDER BY nombre "));
7      }
8  }
9  function verificaHijos($id){
10     $resultado = $this->execSQL("SELECT count(id) as num FROM paginas
11 WHERE dependencia=$id");
12     $row = mysql_fetch_assoc($resultado);
13     return($row["num"]);
14 }
15
16     return($resultado);
17 }
18 }
19 ?>
```

Figura 33 Archivo class.php

Las funcionalidades que ofrece este apartado podemos encontrar Eliminar, Editar, Vista Previa, Activar/Desactivar.

Eliminar: Esta opción le permite al administrador eliminar la página y sus dependencias de forma permanente, es decir, el registro se elimina directamente de la base de datos, no es posible recuperar la información.

Editar: Este apartado otorga el acceso a los datos generales y contenido de la página, redirecciona al apartado ***paginas_editor***.

Vista Previa: La opción vista previa, genera una nueva pestaña en el navegador, en el cual se visualiza la presentación final del contenido.

Activar / Desactivar: Esta opción permite retirar el contenido del sitio Web de manera temporal, sin la necesidad de eliminar la página y sus dependencias. Con un solo clic el usuario puede activar o desactivar información del sitio, de manera oportuna.

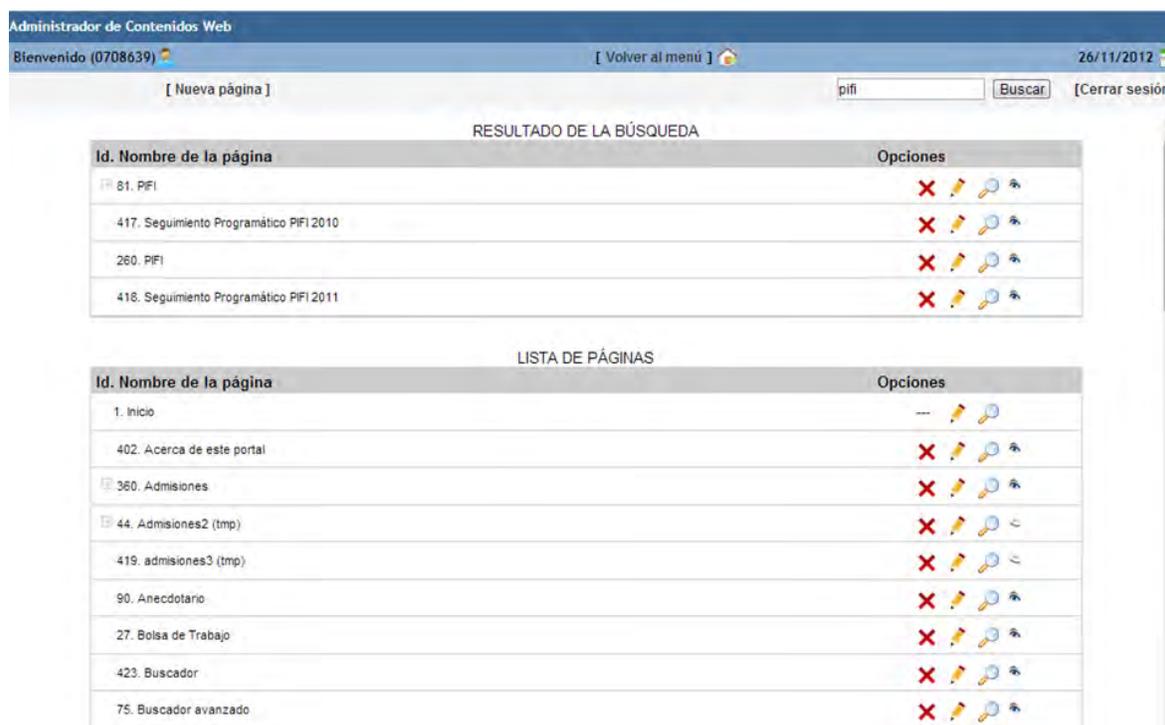


Figura 34 Menú modulo páginas

EDITOR DE PÁGINAS

Al momento de crear o modificar una página el panel de administración se dirige al editor de páginas (Ver **Error! Reference source not found.**), en esta sección se presenta la configuración detallada y el contenido de la página, entre las opciones principales se encuentran las siguientes:

- **Nombre corto:** conforma el URI de la página, se utiliza para añadir el TITLE del html, es decir, definir el título de la página en la barra de navegador, además de establecer el nombre que aparece en la lista de páginas dentro del back-end.
- **Nombre Largo:** Nombre que se visualiza en el encabezado del Front-end.

- **Descripción:** Resumen general de la página, es utilizado por el meta description del Front-end para describir el contenido o tema de la página en los motores de búsqueda.
- **Palabras claves:** Usado para construir el meta keywords de la página, con el objetivo de optimizar el posicionamiento en buscadores.
- **Dependencia:** Define dependencias entre páginas anidadas. El usuario genera la dependencia al seleccionar la página padre.

Estos datos permiten crear la página, sin necesidad de que el usuario inserte código html, la información que el formulario solicita, es incorporado en un plantilla html prediseñada, de tal forma que el contenido se genera de manera dinámica, de acuerdo a la solicitud del usuario general.

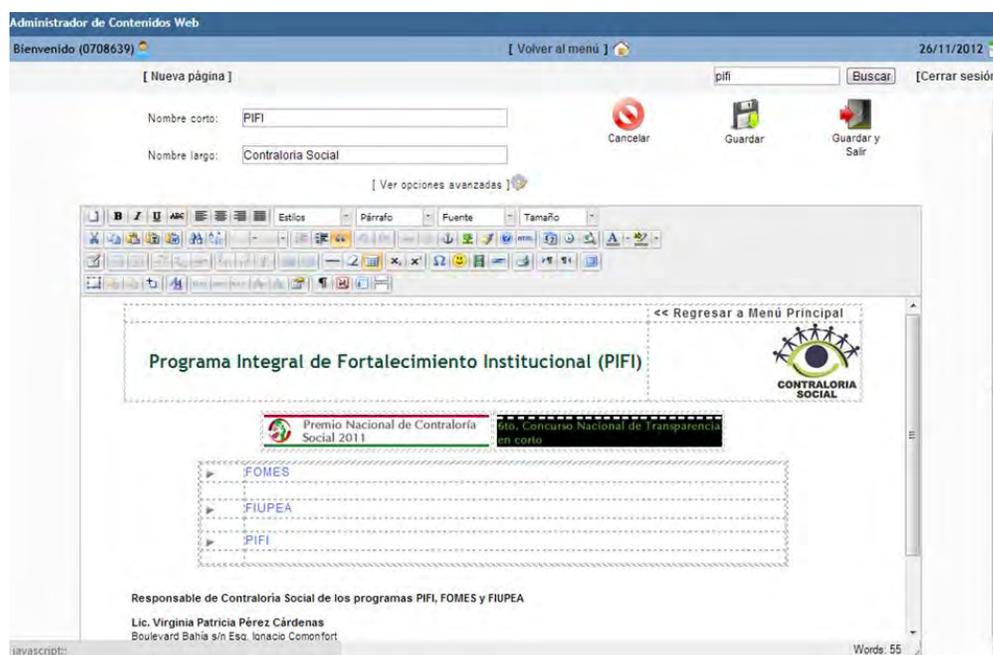


Figura 35 Editor páginas

Así mismo, se incluye dentro del módulo una herramienta que facilita la manipulación del contenido, proporciona opciones similares a un procesador de texto, el cual permite cambiar el tamaño y estilo de las fuentes, insertar imágenes, crear tablas, insertar o editar hipervínculos, etc. Se analizaron las características

básicas de algunos editores, se consideró incluir la herramienta tinyMCE , debido a que la distribución es de forma gratuita, es independiente a la plataforma, fácil de usar, permite editar la información en "modo gráfico" y proporciona la edición directa del código HTML generado para los usuarios avanzados.

La barra de herramientas que proporciona este TinyMCE es similar al editor de textos Word (ver Figura 36), por lo tanto, el usuario está familiarizado con el entorno de trabajo.



Figura 36 Barra de herramientas TinyCME

Para agregar el editor al módulo de páginas se insertó un bloque de código JavaScript (Ver Figura 37), en la parte del encabezado del archivo ***paginas_editor.php***, en el cual se detalla las propiedades básicas del editor, como el idioma, tema y caratula, de igual forma se determina que el elemento a utilizar para presentar el editor es un textarea, es decir, que todas las etiquetas textareas van a tener la apariencia del editor.

```
paginas_editor.php
1 <script src="../../editor/jscripts/tiny_mce/tiny_mce.js"></script>
2 <script type="text/javascript">
3 tinyMCE.init({
4 //Configuración general (Tema, caratula, lenguaje)
5     mode : "textareas",
6     theme : "advanced",
7     skin : "o2k7",
8     language : "es",
9 //Ubicación de la hoja de estilo
10 content_css : "../../editor/css_paginas/css/content.css",
11 //ubicación de los elementos que interactúan con el editor
12 external_image_list_url : "../../editor/css_paginas/lists/image_list.php",
```

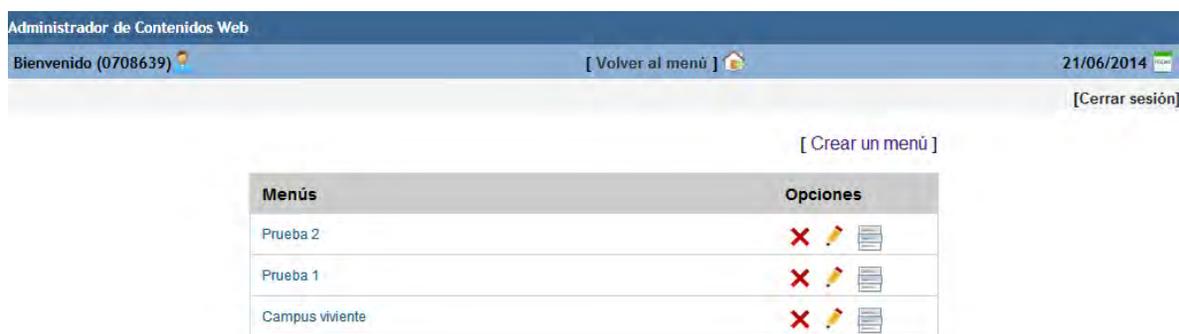
Figura 37 Archivo paginas_editor.php

MODULO MENÚ DE NAVEGACIÓN

Este módulo se desarrolló para la creación e incorporación de menús en las páginas Web, así mismo proporcionar la capacidad de configurar cada menú de forma independiente de tal manera que sea posible agregar opciones diferentes a cada grupo.

Estos elementos se muestran ordenados o anidados, de tal forma que se tiene la posibilidad de construir menús con diferentes niveles de profundidad, cada opción puede estar ligado a: páginas Web, archivos, URLs o a un grupo de menú diferente.

Cabe mencionar que el panel de administración de menús y de opciones, es similar al panel del administrador de páginas, en ella se visualiza una lista con los grupos de menús creados (Ver Figura 38), y por separado una lista de opciones del menú donde se ordenan y activan los elementos ya creados.

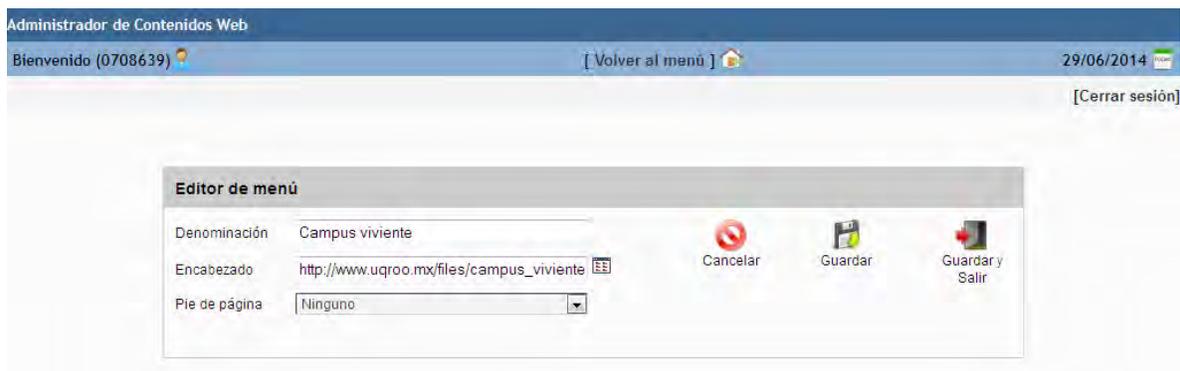


The screenshot shows the 'Administrador de Contenidos Web' interface. At the top, there is a header with the text 'Administrador de Contenidos Web', a user greeting 'Bienvenido (0708639)', a home button, a link '[Volver al menú]', the date '21/06/2014', and a session management button '[Cerrar sesión]'. Below the header, there is a link '[Crear un menú]'. The main content area features a table with two columns: 'Menús' and 'Opciones'.

Menús	Opciones
Prueba 2	  
Prueba 1	  
Campus viviente	  

Figura 38 Modulo menú

Para crear un menú nuevo, es necesario llenar el siguiente formulario Figura 39 donde se especifica el nombre del menú (denominación), esta denominación sirve como identificador en el módulo de páginas, para desplegar una lista de los menús disponibles que podrán ser añadidos en las páginas Web del sitio.



The screenshot shows a web administration interface titled 'Administrador de Contenidos Web'. At the top, there is a navigation bar with 'Bienvenido (0708639)', a home icon, '[Volver al menú]', and the date '29/06/2014'. A '[Cerrar sesión]' link is located in the top right corner. The main content area features a form titled 'Editor de menú' with the following fields: 'Denominación' (Campus viviente), 'Encabezado' (http://www.uqroo.mx/files/campus_viviente), and 'Pie de página' (Ninguno). To the right of the form are three buttons: 'Cancelar', 'Guardar', and 'Guardar y Salir'.

Figura 39 Editor menú

La sección de encabezado se enlaza con el módulo de archivos para seleccionar una imagen que se presentará como encabezado en la parte superior del menú como podemos apreciar en Figura 40 , esta opción se usa normalmente para anexar información adicional o presentación visual al menú, las dos últimas configuraciones son opcionales, debido a que no afecta el funcionamiento del mismo si se dejan los campos vacíos.



Figura 40 Ejemplo sección encabezado

Si un menú está disponible en el sitio, como es el caso, por defecto, de los enlaces primarios y secundarios, cualquier cambio que se realice sobre éste se reflejara de manera automática en todas las paginas donde se tenga asignado el ID del menú.

EDITOR GRUPOS

Después de las consideraciones anteriores fue necesario la creación del archivo **editor_grupos.php** (Ver Figura 41) quien es el encargado de realizar consultas a la base de datos del sistema para crear, actualizar y/o eliminar algún registro en específico.

```
editor_grupos.php
1  <?php
2  //Datos recibidos
3  $id=$_GET['id'];
4  $denominacion=$_GET['denominacion'];
5  $accion=$_GET['accion'];
6  $pie_pagina=$_GET['pie_pagina'];
7  $encabezado=$_GET['encabezado'];
8  if($accion=="guardar" or $accion=="salir"){
9      if($id!=""){
10         //Actualiza datos
11         $result1=mysql_query("UPDATE grupos SET
12         encabezado='$encabezado',denominacion = '$denominacion',
13         pie_pagina=$pie_pagina WHERE id=$id LIMIT 1",$Link) or die ("Error al
14         modificar el grupo ".mysql_error() );
15         }elseif($id==""){
16         //Guarda en la BD
17         $result2= mysql_query("SELECT max(id) AS maxid FROM grupos",$Link);
18         $row=mysql_fetch_array($result2);
19         $id= $row['maxid'] + 1;
20         $result2= mysql_query("INSERT INTO grupos
21         VALUES($id,'$encabezado','$denominacion','$pie_pagina)", $Link) or die ("Error
22         al crear el grupo ".mysql_error() );
23     }
24 }
25 ?>
```

Figura 41 editor_grupos.php

Para extraer los grupos de menús y opciones es necesario realizar una consulta a la base de datos, el archivo **menú_grupos.php** es el encargado de ejecutar esta acción y presentarla en conjunto con el archivo **bitácora_caratura.php**.

EDITOR OPCIONES

En la sección de submenú se lista las opciones dependientes de cada grupo, cada ítem se puede eliminar, editar, cambiar la posición en el submenú, únicamente moviendo las flechas de arriba hacia abajo, de igual forma se puede crear una opción y mantenerla oculta, es decir, no se puede visualizar en el sitio Web hasta que se modifique la propiedad de inactivo.

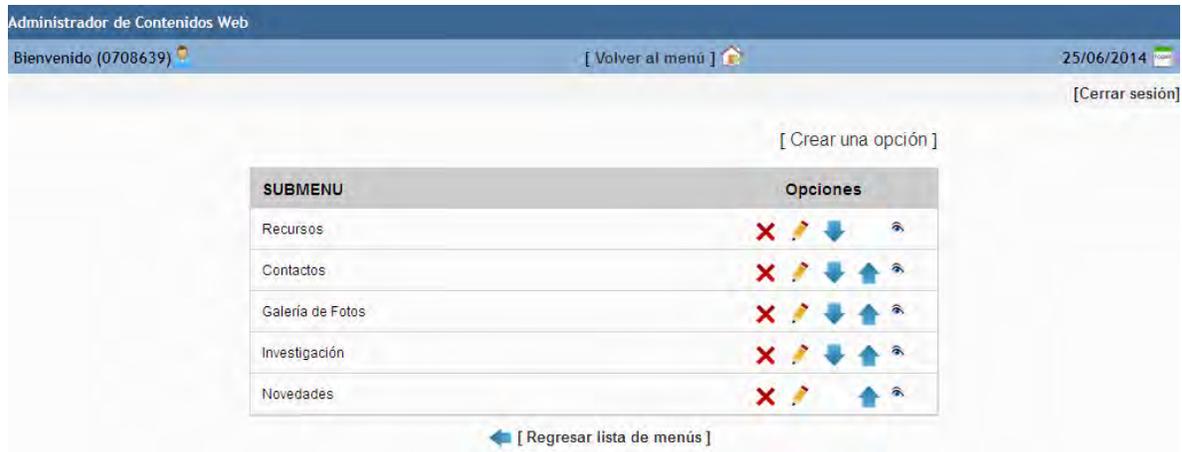


Figura 42 Datos del link

Para crear una opción, el sistema presenta un formulario donde se solicita al usuario las características generales del link, como por ejemplo, el nombre de la opción, el lugar donde la información se va a presentar, puede ser en la misma ventana o en una nueva, y el grupo al que pertenece.

En la sección *Abrir* se determina si la opción direcciona a un archivo, una página, un sitio externo o un menú. Únicamente es posible seleccionar una casilla de verificación.

Si se selecciona la opción de abrir archivo, es necesario cargar el recurso al servidor, para esto el usuario debe dar clic en el icono que aparece del lado derecho.

The screenshot shows a web application interface for managing content. At the top, there is a blue header with the text 'Administrador de Contenidos Web' and a user greeting 'Bienvenido (0708639)'. Navigation links include '[Volver al menú]' and a date '26/11/2012'. Below the header, there are three main navigation buttons: '[Lista de Grupos]', '[Crear Grupo]', and '[Crear Opción]'. The central focus is a modal window titled 'Datos del link'. This window contains several input fields: 'Etiqueta' with the value 'Doctorado en Geografía', 'Tipo' set to 'Opción', 'Destino' set to 'Misma ventana', and 'Grupo' set to '51 : 2012_carreras_doctorados'. Under the 'Abrir' section, there are four radio buttons: 'Archivo', 'Página' (which is selected), 'Uri', and 'Menú'. Each radio button is associated with a text input field. The 'Página' field contains '393 : Doctorados', and the 'Menú' field contains '1 : Nuestra universidad'. At the bottom of the modal, there are three buttons: 'Cancelar', 'Guardar', and 'Guardar y salir'.

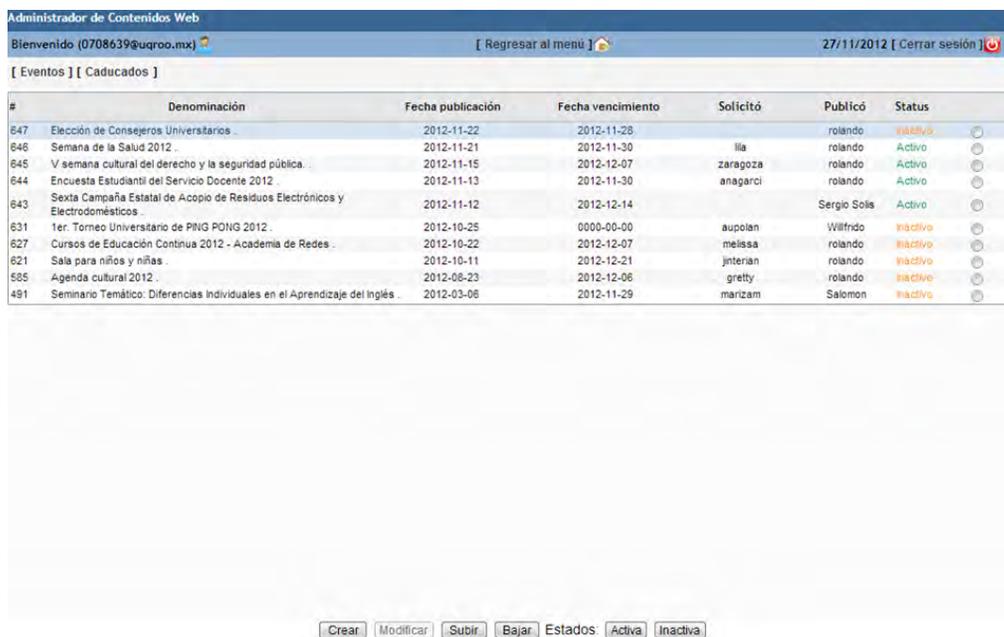
Figura 43 Datos del link

Se concluye el proceso al momento de guardar la opción, este elemento queda habilitado de manera automática, es posible modificar el estatus en el panel general

MODULO AVISOS, ANUNCIOS EXTERNOS, EVENTOS, CONVOCATORIAS

Durante el diseño de las extensiones del sistema se identificó características o comportamientos comunes en relación a las funcionalidades y aspectos generales de los módulos de avisos, eventos, eventos externos, convocatorias y noticias, por lo tanto, se codificó un panel de administración con características similares para cada módulo.

Al acceder al menú principal se presenta un resumen general de las publicaciones recientes, donde el usuario puede observar el estatus de los mismos y administrar según sea el caso. De acuerdo a los puntos que se establecieron en los requerimientos, es necesario que estas publicaciones se realicen en el menor tiempo posible, estas características se pueden observar en el panel principal (Ver Figura 45) de este módulo, con solo dos pasos el usuario puede retirar de manera inmediata la publicación seleccionada, en la parte inferior del panel se incluyeron dos botones que administran el estado de las publicaciones.



#	Denominación	Fecha publicación	Fecha vencimiento	Solicitó	Publicó	Status
647	Elección de Consejeros Universitarios	2012-11-22	2012-11-28		rolando	Inactivo
646	Semana de la Salud 2012	2012-11-21	2012-11-30	lla	rolando	Activo
645	V semana cultural del derecho y la seguridad pública	2012-11-15	2012-12-07	zaragoza	rolando	Activo
644	Encuesta Estudiantil del Servicio Docente 2012	2012-11-13	2012-11-30	anagarci	rolando	Activo
643	Sexta Campaña Estatal de Acopio de Residuos Electrónicos y Electrodomésticos	2012-11-12	2012-12-14		Sergio Solís	Activo
631	1er. Torneo Universitario de PING PONG 2012	2012-10-25	0000-00-00	aupolan	Wilfrido	Inactivo
627	Cursos de Educación Continua 2012 - Academia de Redes	2012-10-22	2012-12-07	meissa	rolando	Inactivo
621	Sala para niños y niñas	2012-10-11	2012-12-21	jnterian	rolando	Inactivo
585	Agenda cultural 2012	2012-08-23	2012-12-06	gretty	rolando	Inactivo
491	Seminario Temático: Diferencias Individuales en el Aprendizaje del Inglés	2012-03-06	2012-11-29	marizam	Salomon	Inactivo

[Crear] [Modificar] [Subir] [Bajar] Estados: [Activa] [Inactiva]

Figura 44 Modulo eventos

Como resultado se creó el archivo **bandeja.php** en el cual se realiza una serie de consultas que permiten desplegar en pantalla todos los anuncios que se encuentran vigentes, activos ó inactivos y el periodo de vigencia. Al momento de crear una publicación se toma la fecha actual del sistema que sirve como base para extraer las publicaciones que contengan la fecha de vencimiento mayor o igual a la fecha actual.

```
bandeja.php
1  <?php
2  $fechaActual = date("Y-m-d");
3  //Consulta eventos vigentes
4  $resultado = mysql_query("SELECT * FROM publica_eventos WHERE
5  FechaVencimiento>='$fechaActual' or FechaVencimiento='0000-00-00' ORDER BY
6  Prioridad DESC",$Link);
7  //Estatus de la publicacion
8  if($accion=="inactiva"){
9  $resEstatus = mysql_query("UPDATE publica_eventos SET activo ='0' WHERE
10 IdEvento='$IdEvento'", $Link) or die (mysql_error());
11 //activo
12 }else if($accion=="activa"){
13 $resEstatus = mysql_query("UPDATE publica_eventos SET activo='1' WHERE
14 IdEvento=".$IdEvento,$Link) or die (mysql_error());
15 }
16 ?>
```

Figura 45 bandeja.php

Para poder publicar el contenido de un aviso, evento, convocatoria, se requiere seguir 4 pasos básicos, que el panel editor de eventos proporciona al usuario a través de formularios.

El primer formulario recaba los datos generales de la publicación, los datos relevantes son los siguientes: el nombre del evento, lugar donde se llevara a cabo, descripción general, el periodo de vigencia.

Cabe mencionar que en apartado de vigencia el usuario determina la fecha de inicio en que la publicación se activara en el front-end y la fecha de retiro. Con estos datos le proporcionamos al CMS la capacidad de programar la publicación de manera automática.

Administrador de Contenidos Web

Bienvenido (0708639@uqroo.mx) [Regresar al menú] 27/11/2012 [Cerrar sesión]

[Eventos] [Caducados]

Datos Generales del Evento Paso 1 de 4

Denominación:

Lugar:

Hora:

Descripción:

Dirigido a:

- Comunidad Universitaria
- Docentes
- Administrativos
- Alumnos
- Público General

Impartido por:

Organizado por:

- Campus Chetumal
- Campus Cozumel
- Campus Playa del Carmen
- DCI
- DCSEA
- DCPH
- DCS
- DDS
- DICA

Más Información:

Periodo de vigencia del evento en Web:

Periodo de realización del evento:

Responsable: @uqroo.mx

Figura 46 Editor eventos

El segundo formulario se encarga de crear un directorio en el servidor donde se alojara los recursos que la publicación requiere, es decir los archivos pdf, Word, Excel, JPG que estarán vinculados a esta publicación, en caso contrario, únicamente se pasa al siguiente formulario.

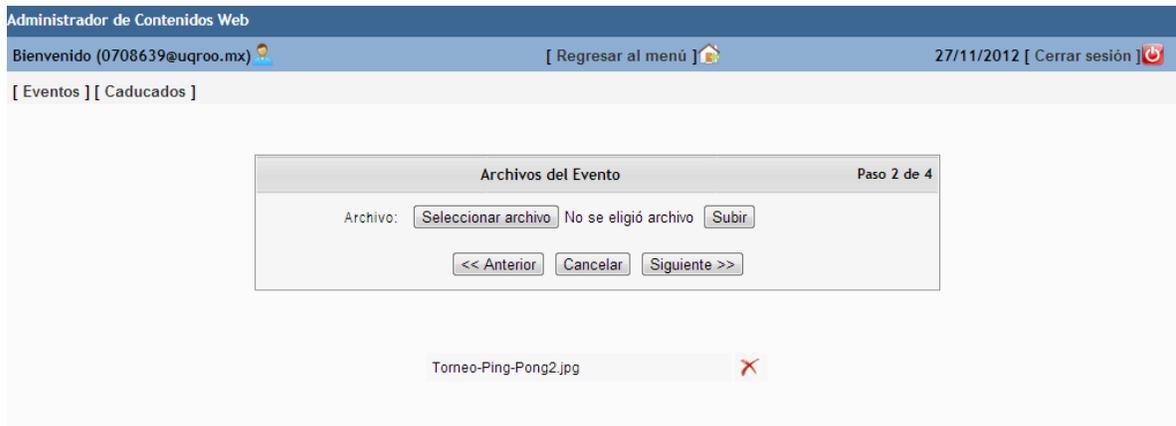


Figura 47 Archivos del evento

Este proceso se lleva a cabo por medio del archivo ***ftfunc.php***, el cual permite establecer una conexión con el servidor a través del protocolo de transferencia de archivos FTP, este protocolo se basa en la arquitectura cliente-servidor, el objetivo principal es el intercambio de archivos entre dos equipos, independientemente de la plataforma que cada uno utilice.

Para poder acceder al servidor es necesario contar con un perfil de usuario que cuente con los permisos escritura y descargar de archivos. Los datos de configuración del archivo ***ftfunc.php*** (Ver Figura 48) son los siguientes: Nombre o dirección IP, número de puerto, usuario, password, dirección donde se creara el directorio de la publicación.

La sentencia `ftp_mkdir` crea un directorio en el servidor donde se almacena los archivos de la publicación, cada directorio se identifica con el ID del evento, en caso de que exista la dirección, únicamente se anexan los nuevos recursos.

```
ftpfunc.php
1  <?php
2  //Datos del Servidor FTP
3  define("SERVER","xxx.xxx.xxx.xxx"); //IP o Nombre del Servidor
4  define("PORT",xxx); //Puerto
5  define("USER","xxx"); //Nombre de Usuario
6  define("PASSWORD","xxxxxx"); //Contraseña de acceso
7  define("PASV",false); //Activa modo pasivo
8  define("RUTASERVIDOR","/opt/sitios/nuevo/eventos/");
9
10 //FUNCION
11 function ConectarFTP($IdEvento){
12 //Permite conectarse al Servidor FTP
13 $id_ftp=ftp_connect(SERVER,PORT); //Obtiene un manejador del Servidor FTP
14 ftp_login($id_ftp,USER,PASSWORD); //Se loguea al Servidor FTP
15 ftp_pasv($id_ftp,MODOS); //Establece el modo de conexión
16 ftp_chdir($id_ftp,RUTASERVIDOR);
17 if (file_exists(RUTASERVIDOR.$IdEvento."/")){
18     ftp_chdir($id_ftp,RUTASERVIDOR.$IdEvento."/");
19     if(file_exists(RUTASERVIDOR.$IdEvento."/archivos/")){
20         ftp_chdir($id_ftp,RUTASERVIDOR.$IdEvento."/archivos/");
21     }else{
22         ftp_mkdir($id_ftp,RUTASERVIDOR.$IdEvento."/archivos");//crea
23         el directorio
24         ftp_chdir($id_ftp,RUTASERVIDOR.$IdEvento."/archivos/");
25     }
26 }else{
27     ftp_mkdir($id_ftp,RUTASERVIDOR.$IdEvento."/");//crea el directorio
28     ftp_chdir($id_ftp,RUTASERVIDOR.$IdEvento."/");
29     ftp_mkdir($id_ftp,RUTASERVIDOR.$IdEvento."/archivos");//crea el
30     directorio
31     ftp_chdir($id_ftp,RUTASERVIDOR.$IdEvento."/archivos/");
32 }
33 return $id_ftp;
34 }
```

Figura 48 Archivo `ftpfunc.php`

Finalmente, para concluir con la publicación del evento el sistema presenta al usuario un formulario que permite colocar un banner, cuando el usuario carga la imagen, se presenta la vista previa de la imagen, como podemos observar en Figura 50.



Figura 49 Banner del evento

Al formulario se le inserto el atributo **enctype="multipart/form-data"**, este atributo especifica cómo se deben codificar los datos del formulario al enviarlos al servidor. Este parámetro se requiere cuando utilizamos formularios que tienen un control de carga de archivos y aplican el método post.

Para complementar este procedimiento se codifico el archivo **sube_ficha.php** (Ver Figura 50) en el cual comprobamos que el archivo exista antes de cargar al servidor, con la variable `$_FILES` obtenemos el nombre del archivo temporal `$_FILES ["archivo"] ["tmp_name"]`, la carpeta temporal, el tamaño, el tipo de archivo y el error que se genera sino cumple con alguna de las validaciones.

```

sube_ficha.php
1  <?php
2  if(!empty($_FILES["archivo"])){
3      $file = $_FILES["archivo"]["tmp_name"];
4      $base_archivo = basename($_FILES["archivo"]["name"]);
5      $tamano = $_FILES["archivo"]["size"];
6      $tipo = $_FILES["archivo"]["type"];
7      $tamano=$tamano/1024;
8      $info=getimagesize($file);
9  if($tipo=='image/jpeg' or $tipo=='image/pjpeg'){
10     if(($tipo=='image/pjpeg' or $tipo=='image/jpeg')){
11         $img = imagecreatefromjpeg($file);
12         $thumb = imagecreatetruecolor(300,140);
13         imagecopyresized($thumb,$img,0,0,0,0,280,55,$info[0],$info[1]);

```


El resultado final va a depender, de la **plantilla predefinida**, en ella se define cómo se va a mostrar la información que se administra desde el Back-end, los módulos pueden situarse en diferentes posiciones dentro del sitio Web, en la Figura 53 podemos observar la posición que cada módulo ocupa, es necesario recalcar que cualquier modificación aplicada en la plantilla afecta directamente a la presentación, no al contenido.



Figura 51 Integración de los módulos en el Front-End

PRUEBA

Para dar continuidad con el ciclo de vida del software, se procedió a realizar el testing del sistema, como primer paso, se alojó el CMS en el servidor Web del Departamento de cómputo y telemática, seguidamente se ejecutó la validación de los usuarios, accediendo con cada uno de los perfiles registrados, en este proceso se analizó la integración de los diferentes componentes del back-end, en el cual, cada módulo debe cumplir con el objetivo para el que fue creado.

El back-end del sistema es la parte fundamental que permitirá al usuario administrador crear y manipular el contenido del portal web. Cumpliendo con los requerimientos establecidos durante el desarrollo del proyecto, se efectuó pruebas paralelas de cada módulo, de tal forma que el funcionamiento se realice de manera independiente.

Estas pruebas consistieron en crear un apartado “Admisiones” en el cual, se utilizó el módulo páginas, grupos-opciones, archivos, con el módulo de grupos, se creó el menú de navegación, y opciones secundarias, las cuales hacen referencia a otras páginas web u otros recursos, tal es el caso de opción “Convocatorias de admisión” que hace referencia a un archivo PDF que fue cargado al servidor por medio del módulo archivos. El contenido se generó a través del módulo páginas, con la ayuda del editor TinyMCE se personalizo la información, aplicando tipo de fuentes y distribución del contenido.

Una vez establecido el ambiente de trabajo del Back-end, se llevó a cabo el ingreso a la parte Front-end, con la finalidad de verificar que el trabajo realizado en el back-end sea visible y navegable en el portal universitario. Como podemos observar en la Figura 53 el contenido de la página se adaptó a la estructura de navegación definida en la etapa de diseño, la interacción de los módulos fue satisfactoria, ya que validaron cada uno de los elementos que conforman el apartado, desde las opciones del menú, la homogeneidad de las páginas y el acceso a los recursos del sitio.

Durante las pruebas se detectaron errores de tipo ortográfico dentro del contenido web, la solución de estos detalles se llevó a cabo corrigiendo el texto embebido en el apartado de admisiones. .



Figura 52 Apartado Admisiones

Seguidamente se realizaron pruebas de usabilidad en múltiples navegadores Web, entre los que destacan Internet Explorer, Chrome, Mozilla Firefox, esto se debe a que la mayoría de los administradores de la universidad cuentan con navegadores con versiones obsoletas, en especial Internet Explorer, ya que si el usuario final trabaja con un navegador incompatible puede verse afectada la experiencia con el CMS y el despliegue de la información en el sitio Web.

El sistema operó de manera idónea en los navegadores y el tiempo de carga (despliegue) de la información fue razonable, alrededor de 2-4 segundos.

WEB PERSONAL

Cabe mencionar que durante la etapa de prueba, se tuvo la oportunidad de configurar el CMS para la administración de las páginas personales de docentes y administrativos, en esta versión se incorporó únicamente el módulo de páginas, menú y archivos, esta adaptación ha tenido un funcionamiento óptimo, aún se está trabajando en el diseño de las plantillas del sitio y la posible incorporación de nuevas funcionalidades.



The image shows a screenshot of a personal website for Homero Toral Cruz. The website has a dark green header with the University of Quintana Roo logo and name. Below the header, there is a navigation menu on the left side with the following items: Home, Academic Background, Research Interests, Publications, Teaching, Students, Research Projects, and Editorial Services and Events. The main content area features a large portrait of Homero Toral Cruz, a man with short dark hair, wearing a brown jacket over a light-colored shirt. Below the portrait, the text reads: "Homero Toral Cruz, Ph.D. Assistant Professor Department of Sciences and Engineering University of Quintana Roo (UQROO) Boulevard Bahía s/n Esq. Ignacio Comonfort, Col. del Bosque, Chetumal, Quintana Roo, México Tel: +52-9838350300 Ext. 325, Fax: +52-9838329656 E-mails: htoral@uqroo.edu.mx and homerotorcruz@gmail.com".

Figura 53 Web personal

ENTREGA

Siguiendo la metodología para el desarrollo de software, y el ciclo de vida del sistema, llegamos a la etapa final, la entrega del producto, el resultado obtenido es un sistema administrador de contenidos CMS, diseñado para cubrir las necesidades específicas de los usuarios administradores del portal web de la universidad de Quintana Roo, el cual presenta un panel de administración sencillo y de fácil entendimiento.

Se entrega un sistema Web capaz de generar contenido interactivo, es decir que se despliega según las peticiones de los usuarios, este contenido no se encuentra alojado en el servidor como las páginas estáticas, sino que se realiza una conexión a una base de datos presentando en el navegador el resultado de la petición. Esta característica otorga al usuario la libertad de poder modificar el contenido, sin alterar el diseño y sin la necesidad de contar con conocimientos en estructuras HTML o de programación.

Las características principales del sistema desarrollado son las siguientes:

- ✓ Escalabilidad
- ✓ Control de usuarios
- ✓ Contenido centralizado
- ✓ Homogeneidad visual
- ✓ Aprobación del contenido
- ✓ Validación LDAP
- ✓ Historial de modificaciones
- ✓ Subida de Archivos
- ✓ URL amigables
- ✓ Programación automática de publicaciones

CONCLUSIONES

En la actualidad existen diversos CMS que permiten realizar varias de las funcionalidades con las que cuenta el sistema construido, pero por la difícil tarea que implica modificar y adaptar las plantillas de diseño, no es una opción viable para su aplicación en el portal universitario. Se realizó una comparación entre los CMS más utilizados en la categoría de administración de portales, y podemos concluir que el sistema implementado cuenta con la mayoría de las funcionalidades de los CMS más conocidos en el mercado como Joomla, Drupal y WordPress.

El aplicar una metodología de investigación nos permitió conocer a detalle, el proceso que se lleva a cabo para la divulgación del contenido. Y sobre todo las características que influyen de manera directa en el proceso de publicación, esto sirvió como base para detallar los requerimientos que el sistema debe cumplir.

El sistema cuenta con una estructura jerárquica que suministra el control de gran cantidad de páginas, permite la accesibilidad desde cualquier sitio remoto que tenga acceso a Internet, homogeneidad visual de páginas, control de usuarios, incorporación de URL amigables, posicionamiento Web y la libertad de modificar la plantilla del front-end sin necesidad de cambiar todo el contenido.

Los resultados obtenidos con la implementación del sistema fueron satisfactorios, debido a que se mejoró el tiempo de respuesta en el proceso de publicación y a su vez permitió que la administración del portal universitario se distribuya en las diferentes áreas administrativas de la universidad, es decir que no exclusivamente se aplica el sistema en el área de desarrollo Web.

El contar con un CMS propio, que administre los contenidos y publicaciones del portal universitario, reduce considerablemente los tiempos de desarrollo y ejecución de nuevos módulos, ya que se conoce el funcionamiento interno del

sistema, la estructura y distribución de cada complemento, por tal motivo es posible crear nuevas funcionalidades y adaptar el sistema a distintas situaciones o entornos de trabajo.

Finalmente podemos concluir que la aplicación de una metodología de investigación y de desarrollo de software nos permite llevar un proceso ordenado en todas las etapas, en las cuales obtenemos un panorama amplio de la problemática existente y los aspectos que influyen en el proceso. Las reseñas que obtenidas a lo largo del desarrollo facilitaron las mejoras del sistema, y permitieron entregar un sistema capaz de agilizar y automatizar el proceso de administración de los sitios web universitarios.

BIBLIOGRAFÍA

Arias, A. (2014). *Base de Datos con MySQL*. ITcampus Academy.

Briceño, L. E. (2005). *DELFO: Gestor de Contenidos Digitales de la Universidad Peruana de Ciencias Aplicadas*. Memoria, Universidad Peruana de Ciencias Aplicadas, Trujillo, Perú.

CMS-Hispano. (2004). *cms-hispano.org*. Recuperado el 1 de Noviembre de 2011, de <http://www.cms-hispano.org/index.php?s=content&p=cms>

Cobos, J. S. (5 de Mayo de 2007). *El Profesional de la Información*. Recuperado el 25 de Octubre de 2011, de www.elprofesionaldelainformacion.com

Condor Tinoco, E., & Soria Solís, I. *Programación Web con CSS, JavaScript, PHP y Ajax*. Universidad Nacional José María Arguedas.

Cuenca, Z. K. (2008). *SISTEMA PARA EL DESARROLLO Y MANTENIMIENTO DE PORTALES WEB*. Tesis, PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ, FACULTAD DE CIENCIAS E INGENIERIA, Lima, Perú.

Cuerda, X. G. (2004). Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. *mosaic "Tecnologías y comunicación multimedia"* .

Departamento de Computo y Telemática. (2010). *Departamento de Computo y Telemática*. Recuperado el 12 de Diciembre de 2011, de ti.queroo.mx

F. (s.f.).

García, J. C. (2008). Usabilidad, Accesibilidad y Gestores de Contenidos en el Plenario Sectorial CRUE_TIC. *Úsalo* .

Gil Rubio, F. J., Alonso Villaverde, S., & Yagüe Panadero, A. (2006). *Creación de sitios web con PHP 5*. España: McGraw-Hill.

González, Y. S. (Agosto de 2006). Sistema de Gestión de contenidos. En busca de una plataforma ideal. Cuba.

Gutiérrez Rodríguez, A., & Bravo García, G. (2005). *PHP 5 a través de ejemplos*. México: Alfaomega RA-MA.

InfoAL. (2009). Joomla-Breve historia de los gestores de contenidos. *InfoAL* .

Jiménez, J. A. (2005). *Sistema Gestor de Contenidos Web para el Aprendizaje de idiomas: BABEL*. Tesis, Universidad de Castilla-La Mancha, Escuela Superior de Informática, Departamento de Informática.

Mendoza, J. A. (4 de Octubre de 2004). *Milenium*. Recuperado el 8 de Noviembre de 2011, de <http://www.informaticamilenium.com.mx/paginas/mn/articulo98.htm>

Robertson, J. (3 de Junio de 2003). *Step two Designs*. Recuperado el 4 de Noviembre de 2011, de http://www.steptwo.com.au/papers/kmc_what/index.html

Sampieri, R. H. (2006). *Metodología de la Investigación*. MCGRAW-HILL.

Schulz, R. (2009). *Diseño web con CSS*. Barcelona: MARCOMBO, S.A.

Smile Open Source Solutions. *Gestión de contenidos, Soluciones abiertas*. Barcelona.

Tocora, P. C. (2008). *Implementación de un Content Management System (CMS)*.
Escuela de Ingeniería de sistemas y Telecomunicaciones, Universidad Sergio
Arboleda, Bogotá Colombia.

Tramullas, J. (2005). Herramientas de software libre para la gestión de contenidos.
HIPERTEXT.NET (3).

Universidad de Quintana Roo. (s.f.). *Portal Web Universitario*. Recuperado el 3 de
Noviembre de 2011, de www.uqroo.mx

Verzal, F. (2011). El ciclo de vida de un sistema de información.