MANUAL TÉCNICO WICAWEB

Manejador de Contenidos

Tabla de contenido

INTRODUCCIÓN

WICAWB es un sistema de gestión de contenidos que permite crear, editar, gestionar y publicar objetos en diferentes formatos; además de la creación y administración de los portales en donde se van a publicar dichos contenidos. El sistema permite manejar de manera independiente el contenido y el diseño.

Todo el sistema permite la administración y edición de sus contenidos, módulos, etc. de la manera más fácil. Además permite crear portales muy personalizados y adaptados a la necesidad del usuario.

CONSIDERACIONES GENERALES

- Se incluyó la estructura CSS (Framework) llamada Bootstrap utilizada para manejar la gran mayoría de estilos de la parte administrativa, además existen otros estilos que son manejadas directamente desde archivos separados y personalizados. En la parte visual del sitio, es decir, el FRONT- END también se hace uso de esta estructura para mantener un estándar entre estilos.

- Con el objetivo de disminuir la complejidad al momento de expandir el sistema, se creó un modelo genérico que permite leer las tablas de la base de datos y recuperar la información necesaria, además tiene gestiona el almacenamiento de información.

- Se han establecido funciones y lineamientos básicos que permitirán la inclusión automática de hojas de estilos (css) o archivos javascript (js), a través de “helpers” del layout. El objetivo de estas funcionalidades es evitar la inclusión de estos archivos de forma manual en las vistas y evitar la inclusión de archivos de todos los otros controladores que no van a ser necesarios para el correcto funcionamiento de un controlador en específico. Esto permite una correcta interacción entre todas las partes del sistema.

- Se ha desarrollado también la metodología y lineamientos bajo los cuales se subirán imágenes y archivos adjuntos desde los distintos formularios al sistema, para ello se estableció un formato para el nombramiento de los mismos así como una estructura de almacenamiento en la carpeta UPLOADS. Dentro de este punto se destaca que todas los archivos adjuntos o archivos de imagen que formen parte de un contenido o de una sección, se almacenarán en una carpeta específica llamada CONTENT y ahí se crearán sub carpetas con el año y el mes correspondientes a la fecha en la que se subió dicho archivo.

- Se realizó la implementación de la técnica de desarrollo web conocida como AJAX (acrónimo de Asynchronous JavaScript And XML). Esta técnica de desarrollo permite cambiar ciertas partes de las páginas web de manera asincrónica, es decir, algunos componentes se cargan en segundo plano dando la posibilidad de realizar cambios en las páginas sin necesidad de recargarlas completamente. Al usar esta técnica de desarrollo se aumenta la velocidad, interactividad y usabilidad en la aplicación.

- Se agregó un editor de texto llamado CKEditor para que el usuario tenga la posibilidad de editar o de agregar diversos estilos a los textos que va a ingresar.

- Se agregó el plug-in Highslide para que al hacer clic sobre una foto en el listado de contenidos se agrande a su tamaño real. Al hacer clic en los otros tipos de contenido, se mostrará la pantalla de edición de ese contenido.

RECOMENDACIONES DE SEGURIDAD

La gran mayoría de los agujeros de seguridad en las aplicaciones Web son causadas por:

• No comprobar correctamente los datos alimentados en la aplicación.

• No verificar la codificación de datos en la salida de la aplicación.

A continuación se describe brevemente técnicas de manipulación de datos en los sitios para vulnerar seguridades.

Cross-site scripting (XSS) attack. Este agujero de seguridad es causado por el hecho que la entrada del usuario (la cadena de consulta) se envía directamente al navegador sin verificar o codificar los datos suministrados.

Inyección SQL. Dejar de escapar caracteres especiales es otra causa común de vulnerabilidades de inyección SQL. Ataques de inyección SQL, siempre se debe comprobar o filtrar la entrada del usuario antes de hacer nada con ella, para manipular con una base de datos.

CSRF (cross-site request forgery) attacks. Script que hace petición a otro sitio web.

- Utiliza los privilegios de usuario
- Presentaciones de formulario que no provienen de ellos.

Para proteger los sitios web de las técnicas de vulnerabilidad de datos, se recomienda:

- Enviar los datos de forma única en el formulario.
- Sólo aceptar el envío de datos del formulario si es desde este mismo o de una dirección conocida.

El código de WicaWeb ha sido analizado bajo la versión free de Acunetix, una herramienta para medir las vulnerabilidades de seguridad web de las aplicaciones. Esta versión de Acunetix midió los ataques XSS y no encontró ningún agujero para que se pueda dar dicho ataque.

ANÁLISIS DE RENDIMIENTO

Las pruebas de rendimiento realizadas a sistemas web son utilizadas para determinar la velocidad y eficiencia de los mismos. Las pruebas pueden ser llevadas a cabo a través de herramientas que proveen pruebas de estrés, que permiten determinar la estabilidad del sistema.

Para analizar el tiempo de respuesta del servidor se utilizó la herramienta de código abierto Jmeter, implementada en Java y que permite también medir el comportamiento funcional y el rendimiento.

Se ha analizado las páginas de login y la principal del front end, utilizando Agreggate Graph, que permite visualizar los resultados en una tabla y genera una distribución de los parámetros de medición a nivel gráfico. Sus indicadores son:

- URL : etiqueta de la muestra
- #Muestras: cantidad de Thread utilizados para la URL.
- Media: tiempo promedio en milisegundos para un conjunto de resultados.
- Mediana: valor en tiempo del percentil 50.
- Línea de 90%: máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevo más tiempo.
- Min: tiempo mínimo de la muestra de una determinada URL.
- Max: tiempo máximo de la muestra de una determinada URL.
- %Error: porcentaje de requerimientos con errores.
- Rendimiento: rendimiento medido en los requerimiento por segundo / minuto / hora.
- KB/sec: rendimiento medido en Kbytes por segundo.

Las pruebas realizadas consistieron en definir 50 y 100 threads cada uno, los cuales simulan 50 y 100 accesos de usuarios respectivamente. Se definieron una lista de enlaces a los que se simuló el acceso aleatorio y a partir de ahí, se recolectaron los datos necesarios para su interpretación.

Prueba con 50 usuarios:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Core - Login	50	249	248	384	77	533	0.00%	37.1/sec	59.0
Default - Ho...	50	170	159	271	72	284	0.00%	37.3/sec	59.2
TOTAL	100	209	183	354	72	533	0.00%	70.5/sec	111.9

Como se puede observar en la imagen anterior, el tiempo promedio para acceder a la página de login en el core es de 2,49 segundos y 1,70 segundos en el front end, realizándose un total de 50 peticiones de usuarios tanto en core como front.

El tiempo total utilizado para los 50 threads se puede calcular con la siguiente fórmula:

Tiempo Total = #Muestras * Media

Tiempo Total Core = 50 * 249 = 12450 milisegundos

Tiempo Total Front = 50 * 170 = 8500 milisegundos

El tiempo promedio total requerido por cada thread, se puede calcular de la siguiente manera:

((Tiempo Total /1000))/cantidad de Thread

Tiempo medio por Thread en Core = ((12450/1000))/50 = 0.249 segundos

Tiempo medio por Thread en Front = ((8500/1000))/50 = 0.17 segundos

Prueba con 100 usuarios:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Core - Login	100	846	999	1151	62	1254	0,00%	47,6/sec	73,5
Default - Home	100	750	768	971	115	1129	0,00%	38,3/sec	59,1
TOTAL	200	798	829	1126	62	1254	0,00%	73,7/sec	113,9

Como se puede observar en la imagen anterior, el tiempo promedio para acceder a la página de login en el core es de 8,46 segundos y 7.50 segundos en el front end, realizándose un total de 100 peticiones de usuarios tanto en core como front.

El tiempo total utilizado para los 100 threads se puede calcular con la siguiente fórmula:

Tiempo Total = #Muestras * Media

Tiempo Total Core = 100 * 846 = 84600 milisegundos

Tiempo Total Front = 100 * 750 = 75000 milisegundos

El tiempo promedio total requerido por cada thread, se puede calcular de la siguiente manera:
((Tiempo Total /1000))/cantidad de Thread

Tiempo medio por Thread en Core = ((84600/1000))/100 = 0.846 segundos

Tiempo medio por Thread en Front = ((75000/1000))/100 = 0.750 segundos

Para estas pruebas de rendimiento, se ha descargado el proyecto wicaweb en una máquina con las siguientes especificaciones técnicas:

Procesador: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz 2.00GHz

Memoria RAM: 8Gb

Sistema Operativo: Windows 7 64bits

Disco Duro: 600Gb

MODULO INSTALLER

Instalador

El módulo instalador está ubicado en /application/modules/installer, y sirve para instalar Wicaweb desde cero y paso a paso de forma gráfica. Sin embargo para que funcione correctamente necesita los siguientes prerrequisitos, expuestos también al usuario en el instalador:

- Instalación de PHP
- Instalación de MySQL
- Permisos de lectura y escritura (777) en los siguientes archivos:
    - application.ini (Ubicado en /application/configs/application.ini)
- Permisos de lectura y escritura para las siguientes carpetas (777):
    - /public
    - /public/uploads
    - /public/uploads/tmp
    - /public/uploads/cache
    - /public/uploads/website
    - /public/uploads/template_img
    - /public/uploads/content
    - /public/website
    - /public/content
    - /public/images
    - /public/images/captcha
    - /public/js
    - /public/js/modules/
    - /public/css
    - /public/css/modules/
    - /public/images/controlPanel
    - /application/modules/core/layouts/scripts/partials
    - /public/uploads/
    - /public/uploads/tmp
    - /application/modules/ (Para módulos externos)

Acerca de la creación de la base de datos

El instalador ofrece dos opciones para la ejecución de scripts de la base de datos. La una no crea la base de datos, sino solo ingresa las tablas y los datos necesarios. Previamente esta base de datos necesita ser creada en un servidor de base de datos MySQL.

La otra opción en cambio crea la base de datos de forma transparente en el servidor MySQL, por lo que necesita obligatoriamente ser un usuario “root” o poseer todos los privilegios para esta operación. Ambas ejecutan los scripts ubicados en */application/modules/installer/sql_scripts*. Estos scripts pueden ser manipulados para realizar otros ingresos desde el inicio, sin embargo no podrán ser cambiados de nombre.

Acerca de base de datos remota

Si se desea realizar una conexión remota a un servidor de base de datos MySQL, previamente en el servidor externo se deberá asignar un nuevo usuario con acceso de cualquier servidor (%), además de todos los permisos y su clave correspondiente. Posteriormente desde el cliente, en el instalador en el campo Host (Host Base de datos) debe colocar la dirección IP del servidor externo (Ej 192.168.0.120) y el nombre de usuario y contraseña creados previamente.

MÓDULO CORE

Login

El módulo login, realizado con la librería *Zend_Auth* provista por *Zend Framework*, permite el ingreso de usuarios almacenados en base de datos a la administración del sistema; los usuarios del sistema se encuentran almacenados en la tabla *wc_user.*

Paralelamente con el proceso de login de usuario, el sistema controla el acceso a determinados módulos del sistema, basándose en los roles asignados a cada perfil almacenado, proceso que implica el uso de las siguientes tablas del sistema: *wc_module, wc_module_action, wc_module_action_profile, wc_profile, wc_section_profile, wc_user, wc_website_profile*.

Panel de Control

El panel de control permite visualizar de forma dinámica y gráfica los módulos de Wicaweb, y extrae su información de la tabla *'wc_modules'.* A destacar el campo *'wc_module.action'* que almacena el path de la acción a donde redirigirá cada módulo, y también el campo *'wc_module.partial'* que almacenará el nombre del partial con extensión .phtml que servirá para renderizar el contenido, en caso de que sea módulo externo.
Paralelamente se controla y muestra sólo los módulos asignados a cada usuario y perfil, información que se encuentra almacenada en la tabla *'wc_module_action'* y *'wc_module_action_profile'* y también los módulos que pueden ser accedidos en un determinado portal, información que se encuentra en *'wc_website_profile'.*

Barra de Navegación

La barra de navegación permite visualizar los módulos existentes en Wicaweb y acceder a ellos desde cualquier parte de Wicaweb. Al igual que panel de control, extrae la información de la tabla *'wc_modules'*, controla los módulos a los que puede acceder cada perfil y cada usuario analizando las tablas *'wc_module_action'* , *'wc_module_action_profile', 'wc_website_profile'* y los muestra en forma de menú.

Portales

Portales permite la administración de portales alojados en Wicaweb. Para ello accede, guarda y actualiza en la tabla '*wc_websites*'. Los formularios son generados con la librería de Zend *Zend_Form,* que permite mediante *'Core_Form_Website_Website'* la reutilización de campos en varias vistas, en este caso en *'new.phtml', 'edit.phtml'* e *'index.phtml'* así como la carga de imagenes hacia el servidor facilmente.

CMS

El CMS del sistema se encarga netamente del manejo de contenidos de los portales creados en el sistema que van a ser renderizados en el front-end del sitio.
Comprende el manejo y creación de secciones, subsecciones, artículos y contenidos relacionados a una sección creada anteriormente, dicho manejo se realiza mediante el uso de un árbol de navegación dinámico que se explicará posteriormente.

Para brindar una mayor facilidad de uso a los usuarios, el CMS del sistema se encuentra desarrollado completamente con funciones ajax de la librería *Jquery,* con las cuales se llega a realizar hasta tres niveles de carga ajax para lograr la visualización de la administración de contenidos.

Listado de Secciones

En la vista de detalle de la sección se puede trabajar en la disposición visual de los contenidos en el front-end, así como en el orden de las secciones. Cuando se selecciona la raíz de las secciones se podrá ordenar entre las diferentes secciones, mientras si se entra al detalle de una sección, subsección; se podrá ordenar sus contenidos y artículos.

Barra de herramientas

La barra de herramientas que tenemos en CMS está directamente relacionada con el árbol dinámico, con la podremos insertar secciones, contenidos, artículos y vincular contenidos.
Esta barra de herramientas es una lista de ítems que se encuentra creada en el partial *sectionbar.phtm* del *core* del sistema, que se encuentra ubicado en *core/layouts/scripts/partials/,* cuya funcionalidad es controlada mediante javascript en el archivo *index.js* del módulo *section* ubicado en *public/js/modules/core/section/*.

Imágenes

Se agregó la librería phpThumb que permite la optimización de las imágenes utilizadas en la presentación de los contenidos o front-end. Esta librería maneja un cache propio donde se almacenan y se vuelven a generar las imágenes adaptadas al espacio que ocuparán en el front-end para tener una mejor optimización en el uso de las mismas. El cache impide a la librería generar nuevamente las imágenes, lo que hace es buscarlas en el cache. Esto permite que la impresión de las imágenes sea más rápida y eficiente.

Para integrar esta librería y sus funcionalidades al sistema, se creó un archivo PHP llamado PhpThumbHelper que contiene clases específicas que acceden a las funciones de la librería y que permiten generar las imágenes. Además desde aquí se maneja un cache propio (diferente al cache por defecto de la librería) donde se almacenan las imágenes y se basa en las opciones establecidas en el archivo de configuraciones de phpThumb.

Al ser un proyecto diseñado y codificado en Zend Framework, es necesario hacer cambios en el archivo Bootstrap así como crear un archivo de ayuda o Autoloader que permita cargar la librería para que ésta pueda ser incluida de manera "natural" por Zend Framework al inicializar la aplicación.

Una vez completado este proceso se continuó con la adaptación de la clase PhpThumbHelper, clase desarrollada por Mushoq, modificando de sintaxis y estructura de las llamadas o de las funciones para poder cumplir con los estándares de Zend Framework permitiendo así utilizar y llamar a las funciones desde diversas partes de la aplicación sin ningún problema.

Este archivo de ayuda o Helper está constituido por dos clases, la primera llamada *PhpThumbHelper* contiene las funciones e interacciones necesarias para poder crear las nuevas imágenes en base a los parámetros enviados por el usuario. La segunda clase se llama *imageRender*, la cual contiene solamente la función *cache_image.* Se debe llamar a esta función con los parámetros necesarios para la generación de la nueva imagen. Igualmente, desde esta función se establecen las rutas para grabar las imágenes en el cache personalizado.

Cuando se quiera utilizar phpThumb desde cualquier parte del sistema, se debe realizar una llamada a la función *cache_image* de la clase *imageRender* como se indica a continuación:

imageRender::cache_image(NOMBRE_ARCHIVO, PARÁMETROS)

Los PARÁMETROS deben ir en un arreglo cuya "key" o posición es el código del parámetro y el "value" es el valor que tiene ese parámetro, por ejemplo **array('width' =>'100')**

Barra de búsqueda Secciones o Contenidos

La búsqueda de contenidos se la puede realizar para encontrar un elemento específico, teniendo la posibilidad

de editar este si la búsqueda es efectiva. La funcionalidad del botón de búsqueda que dirije a la acción de búsqueda en el controlador de secciones o de contenidos, se encuentra localizado en el archivo index.js dentro de section.

Árbol de secciones

El árbol de secciones es incluido en el layout a través de un partial llamado sections.phtml del módulo core. Los datos que este leen son definidos a través de los controladores en la función init de los mismos. Complementariamente utiliza funciones javascript para controlar la apertura y cierre de los nodos. Estas funciones se encuentran en el archivo globalfunctions.js, localizado en la raíz de la carpeta js.

Nueva sección

En general todo el módulo CMS tiene como eje principal a las secciones, los archivos js que este contiene incluyen las llamadas a los contenidos. Se utilizaron llamadas AJAX para incluir la carga interna de los contenidos. Después cada funcionalidad propia del módulo que asocia es la lógica de programación del controlador, vistas, js.

Una sección incluye opciones básicas, pero también puede manejar otras opciones que corresponden a los parámetros de configuración del portal; como por ejemplo, la inclusión de un campo para agregar autores.

La sección puede incluir imágenes propias, las cuales son almacenadas en una carpeta secciones dentro de public/uploads.

Nuevo Artículo

El artículo está desarrollado como bajo la misma idea de cualquier otro módulo del CORE, es decir respetando las estructuras de vistas, js y css.

Nuevo Contenido

Nuevo contenido permite la creación de un nuevo contenido relacionado a una sección, existe la posibilidad de crear siete contenidos básicos entre los que se encuentran: texto, imagen, link, formulario, flash, video flash y carrusel, que se encuentran almacenados en la tabla *wc_content_type.*

La renderización de contenidos en el *core* del sistema se hace de manera dinámica utilizando datos de cada tipo de contenido almacenados en la base de datos en la tabla *wc_field,* en la cual se encuentran almacenados los campos que cada tipo de contenido necesita para poder ser almacenado y posteriormente renderizado en el *front-end* del sitio.
El contenido de cada uno de los campos de cada tipo de contenido es almacenado en la tabla *wc_content_field,* y el registro de cada contenido se lo hace en la tabla *wc_content,* que tiene datos básicos del contenido.

Para el caso particular del contenido formulario existe la tabla *wc_form_field* en la cual son almacenados los campos que pertenecen a cada uno de los formularios creados, dichos campos pueden ser de los siguientes tipos: campo de texto, área de texto, botón radio, lista desplegable, caja de verificación, comentario y archivo.

La renderización de contenidos en el *front-end* se la realiza con la ayuda de un *Zend_View_Helper_Abstract,* el cual se encarga de identificar cada uno de los contenidos y armar la estructura correspondiente con los datos almacenados en base, para luego imprimir en formato html que podrá ser visualizado en el *front-end*.

Para crear un nuevo tipo de contenido en el wicaWeb, en la base de datos se necesita tomar en cuenta lo siguiente:

- Ingresar manualmente un nuevo tipo de contenido en la tabla *wc_content_type.*
- Ingresar los campos que va utilizar el nuevo tipo de contenido en la tabla *wc_field.*

Vincular Contenido

Al acceder a esta opción, se puede vincular un contenido existente con otra sección. Tiene una búsqueda similar a la barra de búsqueda sin embargo se podría vincular un contenido a una o más secciones. No existe ningún control particular o programación adicional para esta funcionalidad que deba ser mencionado.

Usuarios

Los usuarios tienen asociado un perfil, y son aquellos que van a interactuar con el Core del sistema; por tanto son creados desde la opción correspondiente del panel de control. Su información es almacenada en wc_user y complementariamente obtiene permisos de interacción. Se explica a continuación.

Perfiles

Los perfiles otorgan permisos al usuario y han sido concebidos para trabajar bajo acción por módulo y dentro de una sección. La programación de este módulo del Core es realizada a través de un wizard, con el cual se van completando los pasos que entregan la funcionalidad total del perfil.

El wizard básicamente presenta unas secciones y oculta otras, se ha implementado dichas funcionalidades en el new.js y edit.js

Archivos Externos

Archivos externos permite la carga de archivos sean css o js al sistema mediante dos *Zend_View_Helper_Abstract* llamados *ExternalCssHelper.php* y *ExternalJavascriptHelper.php*, ubicados en *application/modules/default/views/helpers/* siendo estos incluidos en el *front-end* del sitio con la finalidad de modificar estilos css o realizar acciones mediante javascript.

Plantillas

El sistema wicaWeb tiene 4 plantillas básicas por defecto, las cuales pueden ir siendo modificadas de acuerdo a las necesidades de cada portal, de igual manera existe la posibilidad de crear una plantilla totalmente nueva, subiendo un archivo basado en la estructura empleada en los templates base, teniendo también la posibilidad de subir archivos css, js e imágenes.

Los archivos css, js e imágenes así como también el archivo de la plantilla se subirán en las siguientes ubicaciones:

- css: public/css/templates/**nombre de la plantilla**/
- js: public/js/templates/**nombre de la plantilla**/
- imágenes: public/images/templates/**nombre de la plantilla**/
- archivo del template: application/modules/default/views/scripts/partials/

Diccionario

Diccionario permite la administración de diccionarios. Para ello accede, guarda y actualiza en la tabla '*wc_dictionary*' la información general acerca del diccionario y en la tabla '*wc_words*' las palabras que posee cada diccionario.
Diccionario es utilizado para filtrar palabras enviadas en los campos *text area* generados en contenidos tipo formulario, usualmente usados para comentarios o sugerencias. Por este motivo en la acción *sendformemailAction()* del *IndexController* del módulo *Default* se extrae las palabras y la información nuevamente de '*wc_words*' y '*wc_dictionary*', se compara con las palabras ingresadas en el campo tipo *text area* y se envía por ajax un mensaje de éxito o error hacia el js ubicado en */public/js/modules/default/index/index.js* que maneja esta información.

Módulos Externos

Módulos externos permite instalar un nuevo módulo externo y visualizar los módulos externos instalados anteriormente. Es importante recalcar que para instalar un módulo externo necesariamente se debió haber dado permisos de lectura y escritura (777) previamente a las siguientes carpetas:

- /public/js
- /public/js/modules/
- /public/css
- /public/css/modules/
- /public/images/controlPanel
- /application/modules/core/layouts/scripts/partials
- /public/uploads
- /public/uploads/tmp
- /application/modules/ (Para módulos externos)

Además se tiene que tener las siguientes consideraciones en el paquete de instalación a cargar:

- El paquete de instalación debe ser en formato.zip y tener el nombre del módulo a instalar. Ejm banners.zip
- El paquete de instalación debe contener dentro una carpeta con el mismo nombre del módulo a instalar y también con el mismo nombre del paquete de instalación. Ejm banners.
- La carpeta deberá tener dentro dos carpetas: una con el mismo nombre (ej. banners) que contendrá dentro lo necesario para su funcionamiento (controllers, forms, models, views y el bootstrap.php) y otra llamada public que contendrá dentro dos carpetas: css y js, que contendrán dentro cada una otra carpeta con el mismo nombre del módulo (Ej. banners). Dentro de esta irán los archivos css y js respectivamente necesarios para el módulo.
- Obligatoriamente se deberá incluir el archivo parcial (partial) .phtml para la renderización en /controller/views/scripts/partials/
- Se deberá copiar los helpers ubicados en /core/views/helpers/ a la misma ruta en el módulo externo

Así tendremos por ejemplo la estructura siguiente en el paquete de instalación banners:

- banners.zip
    - banners
        - banners
            - controllers
            - forms
            - models
            - views
                - helpers
                - CssHelper.php
                - FlashMessagges.php
                - FormLabelHelper.php
                - JavascriptHelper.php
                - PreviewHelper.php
                - scripts
                - banners
                - Vistas .phtml

index

- Vistas .phtml

partials

- banner.phtml
- Bootstrap.php

public

- css
    - banners

Archivos css

- js
    - banners

Archivos js

Render de contenidos en template frontal

El módulo default utiliza el controlador index y la vista del mismo nombre, desde el cual se llama a vistas auxiliares que son las encargadas de construir los contenidos html. La vista del index ubica los contenidos en la plantilla que se utiliza en el sistema para poder presentar cada una de las páginas.

Las vistas de módulos externos también son llamadas a través de partials auxiliares con los cuales se incluye en la respectiva área de la plantilla. Esta funcionalidad del manejo del nombre de archivos y la ubicación que estos demandan se detallan en la sección de módulos externos.

MÓDULO DEFAULT

Contador de impresiones

El contador de impresiones sirve para contar las veces que una determinada sección es accedida en el front-end de Wicaweb (Módulo default) siempre y cuando está opción esté habilitada en el portal. Para ello accede a la tabla *'wc_section_prints'* cada vez que se visualiza una sección en el front-end y registra el conteo en el controlador index ubicado en /modules/default/controllers/IndexController.php. Paralelamente en el back-end (Módulo core) extrae esta información y la presenta visualmente gracias al *sectiondetails.phtml* del core ubicado en *application/modules/core/views/scripts/section/section/sectiondetails.phtml.*

Login secciones privadas

Las secciones privadas son secciones particulares a las cuales pueden acceder determinados usuarios registrados en el portal, existe la posibilidad de crear una sección destinada únicamente a ser un login en el portal, para esto se debe seleccionar como template de la sección *login template,* con el cual automáticamente aparecerá el login en la sección con las opciones de registrar un nuevo usuario que se guardará en la tabla *wc_public_user,* de igual manera existe la opción de recuperación de contraseña.

Con la finalidad de tener un mayor control y seguridad, el registro y recuperación de contraseña se maneja mediante envío de correos electrónicos al usuario registrado.

Construcción de vistas de módulos externos

Las contenidos de los módulos externos, como es el caso de banners y productos, requieren la creación de un partial a través del cual se construye el html de la porción de página correspondiente. El instalador de módulo externo reconocerá la estructura durante la instalación.

El modelo del módulo externo debe incluir la función rendercontents() la cual devuelve los datos para ser enviados al partial.

MÓDULO DE BANNERS

Crear y Editar banners

El módulo Banners posee la administración de banners, y es la encargada de ingresarlos, eliminarlos y editarlos. Para ello tiene acceso a la tabla *'banner'* que posee la información de cada uno de ellos. Cuando se crea uno, también se afecta a la tabla *'wc_module_description'* que es la tabla que permite relacionar los módulos externos con el core y el default. A su vez se creerá registros en *'wc_section_module_area'* que permitirán asociar los contenidos con las áreas y secciones para su renderización en el front-end.
Las imágenes y archivos flash de un banner se ubican en /uploads/banners/ seguido del año y mes en el cual fueron creados. Por este motivo si es que no se dio permisos el momento de la instalación se deberá dar permisos de escritura y lectura a la carpeta public/uploads/.

MÓDULO DE PRODUCTOS

Crear y Editar productos

El módulo Productos posee la administración de productos y es el encargado de ingresar, eliminar y editar productos. Para ello tiene acceso a la tabla *'products'* que posee la información de cada uno de ellos. Cuando se crea uno, también se afecta a la tabla *'wc_module_description'* que es la tabla que permite relacionar los módulos externos con el core y el default. A su vez se creerá registros en *'wc_section_module_area'* que permitirán asociar los contenidos con las áreas y secciones para su renderización en el front-end.
Las imágenes y archivos flash de un banner se ubican en */uploads/products/* seguido del año y mes en el cual fueron creados. Por este motivo si es que no se dio permisos el momento de la instalación se deberá dar permisos de escritura y lectura a la carpeta */public/uploads/*.