

Proyecto “Seguimiento y promoción para la actualización del sitio Web del Servicio Meteorológico Nacional” TH1114.2 Informe final

Coordinación de Hidrología
Subcoordinación de Hidrometeorología

Julio Sergio Santana
Ricardo Prieto González
María Cristina García Vega

1 ÍNDICE

1	ÍNDICE	1
2	RESUMEN EJECUTIVO.....	3
3	OBJETIVOS.....	3
4	ANTECEDENTES.....	4
5	METODOLOGÍA.....	4
6	RESULTADOS.....	5
7	CONCLUSIONES	5
8	AGRADECIMIENTOS	5
9	ANEXO 1: Mapa Interactivo para la Navegación en el Nuevo Sitio <i>Web</i> del Servicio Meteorológico Nacional	6
10	ANEXO 2: Análisis de los Resultados de la Encuesta de Evaluación de la Integración de Nuevos Servicios en el Servidor Web del SMN.....	8
1.	Antecedentes.....	9
2.	Planteamiento de la Encuesta.....	9
3.	La Base de Datos	11
4.	Presentación de la Encuesta.....	15
5.	La Manipulación de la Base de Datos.....	17
6.	Procedimiento para el Análisis de la Encuesta.....	35
7.	Resumen del Análisis de la Encuesta	38
11	ANEXO 3: Prototipo de Migración del Portal Galileo a un Sistema de Gestión de Contenidos41	
	Objetivo.....	44
	¿Qué es un Sistema de Gestión de Contenidos (CMS)?	44

Instalación del ambiente LAMP (Linux, Apache, MySQL y PHP) en Ubuntu.	44
Instalación de Joomla! (versión 1.5.20).	46
Iniciando Joomla!	53
Elementos básicos de Joomla!	53
Plantillas	53
Componentes	53
Módulos	54
Plugins	54
Usuarios de Joomla!	54
Tipos de usuarios	54
Usuarios del Sitio (Front-end)	55
Usuarios del Administrador (Back-end)	55
Registro	55
Funciones de Gestión desde el Front-end	56
Gestión del Contenido	56
Funciones de Gestión desde el Back-end	56
Acceso a la administración	56
Gestión de archivos Multimedia	58
Jerarquía de Contenido en Joomla!	59
Requisitos del Sistema	59
Diseño y desarrollo de la nueva plantilla para el portal Galileo.	60
Seguridad	61
Algunas consideraciones generales serian	62
12 ANEXO 4: Colección y Análisis de Información del Uso del Sitio a lo largo del año.	64

2 RESUMEN EJECUTIVO

A principios del 2010 el Servicio Meteorológico Nacional (SMN), encargó al Instituto Mexicano de Tecnología del Agua (IMTA) la reestructuración y rediseño de su sitio Web. Este proyecto se llevó a cabo en conjunto con la Coordinación de Comunicación, Participación e Información.

Debido al pago tardío del proyecto, los recursos derivados de ese proyecto y que correspondieron a nuestra Coordinación se manejaron a través del presente proyecto en 2011 (TH 1114.2).

Entre las actividades desarrolladas por nuestra Coordinación en el referido proyecto estuvieron:

- Mapas de la estructura del sitio Web del SMN, antes y después de la reestructuración y rediseño del sitio.
- Encuesta para detectar nuevos servicios a integrar en el sitio.

Para darle continuidad a esos trabajos, el presente proyecto se enfocó principalmente a desarrollar un análisis de la encuesta aplicada, con el fin de generar una propuesta de los nuevos productos y servicios que pueden ser incorporados al sitio Web del SMN, y a agregar el mapa *interactivo* de la estructura del nuevo sitio Web del SMN como una alternativa de navegación en el sitio.

3 OBJETIVOS

Dar atención al sitio Web del Servicio Meteorológico Nacional (SMN) para verificar su adecuado funcionamiento.

Realizar promoción con el personal del SMN, a fin de continuar con las mejoras del sitio. Incorporar encuesta de evaluación de la integración de nuevos servicios en el sitio.

Incorporar mapa gráfico interactivo del sitio en el sitio.

4 ANTECEDENTES

A principios de 2010 el Servicio Meteorológico Nacional (SMN), encargó al Instituto Mexicano de Tecnología del Agua (IMTA) la reestructuración y rediseño de su sitio Web. El planteamiento de los trabajos acordados para ese proyecto, incluyó, por una parte, la incorporación de los contenidos actuales bajo el control de un sistema administrador de contenidos y, por otra, el análisis crítico de la situación del sitio al momento de iniciar los trabajos, así como, una propuesta para la mejora, tanto en contenidos, como en la estructura y funcionalidad del sitio.

El proyecto en cuestión (CP 1035.3) se concluyó con éxito y tuvo como resultado la implantación de un nuevo sitio Web, más acorde con las necesidades del SMN y con las expectativas de los usuarios del sitio.

Ese proyecto, sin embargo, además de la implantación del sitio Web, incluyó la generación de un conjunto de recomendaciones para encaminar el sitio a una mejora sustancial que tendría lugar en una etapa posterior.

El proyecto, objeto del presente documento, tiene como propósito dar seguimiento al buen funcionamiento del sitio y llevar a efecto la promoción encaminada a concretar las recomendaciones mencionadas anteriormente.

5 METODOLOGÍA

1. Instalación de la encuesta de evaluación de la integración de nuevos servicios para el sitio Web del SMN.
2. Análisis de los resultados colectados en la encuesta, para seleccionar los servicios factibles de ser integrados al sitio.
3. Realización de visitas promocionales periódicas, para mostrar en el sitio, las bondades de los nuevos servicios.
4. Instalación del mapa gráfico interactivo del sitio.
5. Recolección de información de uso del sitio y análisis.

6 RESULTADOS

A continuación se listan de manera breve los principales resultados del presente proyecto.

1. Se entregó a los administradores del sitio *Web* del SMN, para su instalación como una alternativa de navegación en dicho sitio, el mapa conceptual interactivo del mismo. Este mapa se muestra en el Anexo 1 del presente documento.
2. Se realizó un análisis de la encuesta para la integración de nuevos servicios al sitio *Web* del SMN. Los resultados de este análisis, y que constituye el reporte de la promoción para la implantación de las recomendaciones, se muestran en el Anexo 2 del presente documento.
3. Uno de los propósitos de este proyecto fue mostrar la bondad de los nuevos servicios a integrar al sitio *Web* del SMN. Dado que no es posible alterar abruptamente el funcionamiento de un sitio Web en operación, como lo es el sitio del SMN, se requiere establecer otro sitio como laboratorio o plataforma para sustentar dichos servicios. Por ello se planteó utilizar un apartado en el sitio *Web* de la Subcoordinación de Hidrometeorología, el portal Galileo con esa finalidad. Encaminado a esto, se desarrolló el marco para migrar el actual sitio al sistema de gestión de contenidos Joomla, que es la plataforma en la que opera el sitio *Web* del SMN. Este trabajo se reporta en detalle en el Anexo 3 del presente documento.
4. A lo largo del año se desarrolló un monitoreo del sitio *Web* del SMN. La información correspondiente a ese monitoreo se analizó y en el Anexo 4 del presente documento se muestra un resumen gráfico de dicho análisis.

7 CONCLUSIONES

Los resultados de la encuesta que se desarrolló y aplicó en el marco del presente proyecto, arrojaron como resultado un conjunto de servicios y funcionalidades que sería deseable incorporar al sitio *Web* del SMN. Como los recursos que el SMN tiene destinados a ese propósito son limitados, en el resultado de la encuesta también se ha establecido un orden de prioridades para la incorporación paulatina de dichos servicios y funcionalidades.

8 AGRADECIMIENTOS

Se agradece el apoyo del Ing. Sinué Benjamín Bravo Ramírez en el proyecto, particularmente en lo que se refiere a la elaboración de un prototipo para migrar el portal Galileo a un sistema gestor de contenidos.

9 ANEXO 1: Mapa Interactivo para la Navegación en el Nuevo Sitio *Web* del Servicio Meteorológico Nacional

10 ANEXO 2: Análisis de los Resultados de la Encuesta de Evaluación de la Integración de Nuevos Servicios en el Servidor Web del SMN

Análisis de los Resultados de la Encuesta de Evaluación de la Integración de Nuevos Servicios en el Servidor *Web* del SMN

1. Antecedentes

A principios del 2010 el Servicio Meteorológico Nacional (SMN), encargó al Instituto Mexicano de Tecnología del Agua (IMTA) la reestructuración y rediseño de su sitio Web. Entre otras actividades, este trabajo incluía la elaboración y lanzamiento de una encuesta para la evaluación de la integración de nuevos servicios al sitio Web del SMN.

En el presente documento se hace un análisis de la encuesta y de sus resultados, con el fin de determinar la importancia de los conceptos considerados y la factibilidad de incorporarlos como parte de los nuevos servicios del Sitio Web del SMN.

2. Planteamiento de la Encuesta

Como punto de partida de la encuesta, se hizo una investigación de los conceptos o servicios que otros sitios enfocados a la misma materia, ofrecen.

Por otra parte, se consideró que los usuarios del sitio Web del SMN podrían ser considerados dentro de dos categorías, a saber: los *usuarios comunes* y los *expertos*, que son aquellos usuarios versados tanto en la materia, como en la estructura y políticas internas del SMN.

De este modo, para cada uno de los conceptos identificados, que fueron 27, se elaboró una entrada de la encuesta como sigue:

1. Envío de alertas nacionales y/o regionales por correo electrónico.
2. Descarga de documentos.
3. Personalización de servicios gratuitos mediante registro.
4. Secciones especiales para actividades humanas.
5. Búsqueda por código postal, localidad o centro de reunión.
6. Muestras de vídeos e imágenes enviados por los usuarios con actividades afectadas o beneficiadas por el clima (Web 2.0).
7. Noticias enviadas por los usuarios (Web 2.0).
8. Noticias y recomendaciones clasificadas por regiones geográficas.
9. Noticias internacionales incluidas en el sitio.
10. Herramientas informáticas, relacionadas con la meteorología, descargables para usuarios.
11. Versiones del sitio adaptadas para dispositivos móviles.
12. Unidades en Sistema Métrico Decimal y Sistema Inglés de Medidas.
13. Recarga y actualización automática de la página cada x minutos.
14. Tabla de pronósticos del estado del tiempo.
15. Pronóstico: Día anterior (datos pronosticados con antelación y datos realmente observados).
16. Pronóstico: Por hora durante el día de hoy.
17. Pronóstico: Día de mañana.
18. Pronóstico: Hasta 7 días (incluido hoy).
19. Pronóstico: Mes actual.
20. Mapa de máximos y mínimos observados.
21. Biblioteca y Librería.
22. Blogs (Web 2.0).
23. Nube de etiquetas (Web 2.0).
24. Listado de últimas ciudades consultadas (Web 2.0).
25. Inclusión de un widget en la página de los visitantes.

26. Aprendizaje sobre meteorología.

27. Investigación y publicaciones.

Se planteó entonces cuestionar a cada *usuario común* mediante el despliegado del concepto y la pregunta, correspondiente al *primer rubro*, que se muestra a continuación y a la que se podía responder con una de las cuatro opciones que se ofrecerían igualmente en el despliegado.

¿Considera que el sitio Web del SMN debería incluir siguiente concepto:

<aquí se inserta el concepto y su definición>?

- *Sí*
- *Es indiferente si se incluye o no*
- *No*
- *No sé*

A los *usuarios expertos*, además de lo anterior se plantea cuestionarles, mediante el *segundo rubro*, acerca de la factibilidad de incluir el concepto, de la siguiente manera:

De acuerdo con los recursos del SMN ¿cree usted que es posible incluir este concepto?

- *Sí, es posible*
- *No. No es posible*
- *No sé*

Dado que la definición textual de los conceptos puede no ser suficiente para su completa comprensión por parte de los usuarios, la interfaz de la encuesta ofrece la posibilidad de tener acceso a un ejemplo gráfico que ilustra cómo se exhibe el concepto en alguno de los sitios que ofrecen el servicio correspondiente a dicho concepto.

3. La Base de Datos

La implementación práctica de la encuesta se hizo mediante una base de datos, que mantiene tanto la información de presentación de la encuesta, como la colectada por el resultado de aplicación de la encuesta. Esta base de datos se desarrolló en la plataforma MySQL.

La base de datos consta de cinco tablas relacionadas de la manera que se muestra en la Fig. 1. En seguida se describe cada una de estas tablas:

- **Encuesta.** Esta tabla constituye el núcleo de la base de datos. En ella se concentran las respuestas de todos los encuestados a cada uno de los 27 conceptos considerados. Sus campos son los siguientes:
 - **id.** Este campo es la llave primaria de la tabla y se genera automáticamente cada vez que se crea un nuevo registro de la tabla.
 - **ip.** Este campo permite identificar, genéricamente, al usuario encuestado y será únicamente la dirección IP de la máquina en la que se conteste la encuesta. Desde el punto de vista de las relaciones entre tablas, se trata de un índice a la tabla “Encuestado”, que se describe más adelante.
 - **id_concpt.** Este campo permite identificar cada uno de los 27 conceptos considerados en la encuesta. Se trata de un índice a la tabla de conceptos: “Concpt”.
 - **calif_quiere.** Esta es la calificación o respuesta otorgada por el usuario encuestado al *primer rubro* del concepto indicado por el campo anterior. Se trata de un índice al campo “valor” de la tabla “Opciones”, que se describe más adelante.
 - **calif_puede.** Esta es la calificación o respuesta otorgada por el usuario encuestado al *segundo rubro* del concepto indicado por el campo anterior. Se trata de un índice al campo “valor” de la tabla “Opciones”, que se describe más adelante. En caso de que la IP del usuario no esté calificada como de un usuario entrevistado “experto”, este valor será NULO.
- **Encuestado.** Esta tabla es para distinguir cada uno de los usuarios encuestados. Ya que no se registra ninguna información personal de los usuarios encuestados, el único medio para distinguirlos es la dirección IP de la máquina de la cual responden a la encuesta. Así, sus campos son:
 - **ip.** Dirección IP de la máquina de la que se responde a la encuesta. Este campo además constituye la llave primaria de la tabla.
 - **id_categoria.** Identifica la categoría del usuario encuestado en cuestión. Este campo es un índice a la tabla “Categoría”.
- **Categoría.** Esta tabla contiene las categorías de los encuestados, que en este caso son únicamente dos, a saber: usuario común (*Público general*) y usuario experto (*Especialistas*). Los campos de la tabla son como sigue:

- **id_categoria.** El identificador de la categoría y llave primaria de la tabla.
- **descripcion.** Se trata de un texto descriptivo de la categoría, en este caso son dos posibles: *Público general y Especialistas*.
- **texto.** Es un texto a desplegar para introducir el rubro del concepto. Este texto solamente se usa en la categoría de Especialistas, dado que el texto para introducir el primer rubro se despliega para todos los usuarios y es igual para todos los conceptos. Básicamente es algo parecido a: *“De acuerdo con los recursos del SMN, ¿cree usted que es posible incluir este concepto?”*.
- **Concpt.** Esta tabla contiene la descripción de cada uno de los 27 conceptos considerados. Sus campos son los que siguen:
 - **id_concpt.** Es el identificador numérico del concepto y además la llave primaria de la tabla.
 - **concepto.** El concepto del que se trata en este registro. Es un título descriptivo del concepto.
 - **descripcion.** Este campo contiene una descripción textual del concepto. Sirve para ampliar de manera inteligible el título del campo anterior.
 - **figura.** La dirección (URL) de una figura explicativa del concepto. En un directorio (folder) particular, se mantiene un conjunto de figuras explicativas de cada uno de los conceptos. Cada una de estas figuras ha sido tomada de alguno de los sitios que presentan el concepto en cuestión. El sistema habilita el acceso a ellas para ilustrar el concepto.

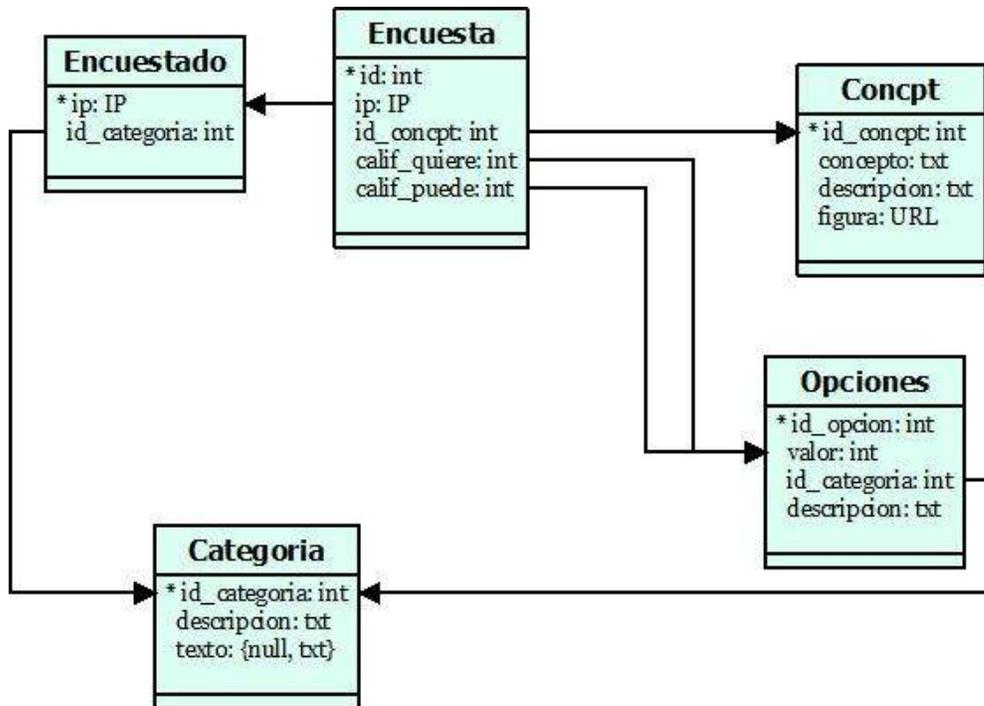


Figura 1. Estructura de la Base de Datos

- **Opciones.** Esta tabla contiene las distintas opciones que se podrán elegir como respuesta para cada uno de los conceptos y en cada uno de sus rubros. Sus campos son como sigue:
 - **id_opcion.** Identificador de la opción y llave primaria de la tabla.
 - **valor.** Este es un entero que identifica la opción. El campo id_opcion es secuencial, para todas las opciones, mientras que este campo le da un valor entero a la opción pero dentro de un rubro, que de alguna manera califica la opción elegida por el encuestado.
 - **id_categoria.** Este campo permite identificar la categoría o, igualmente el rubro al que pertenece la opción. También se trata de un índice a la tabla de categorías, "Categoría".
 - **descripcion.** Este campo contiene el texto correspondiente a la opción. Este mismo texto es el que le será desplegado al encuestado para que pueda seleccionar entre las diferentes opciones mostradas para un concepto y rubro determinados.

4. Presentación de la Encuesta

Para presentar la encuesta a los distintos encuestados, se desarrolló un sistema en el lenguaje PHP, que se encarga de consultar la base de datos y presentar cada uno de los conceptos, tal como se muestra en la Fig. 2. Como ilustración, en esta figura se despliegan solamente las entradas de la encuesta correspondientes a los primeros dos conceptos de los 27 considerados.

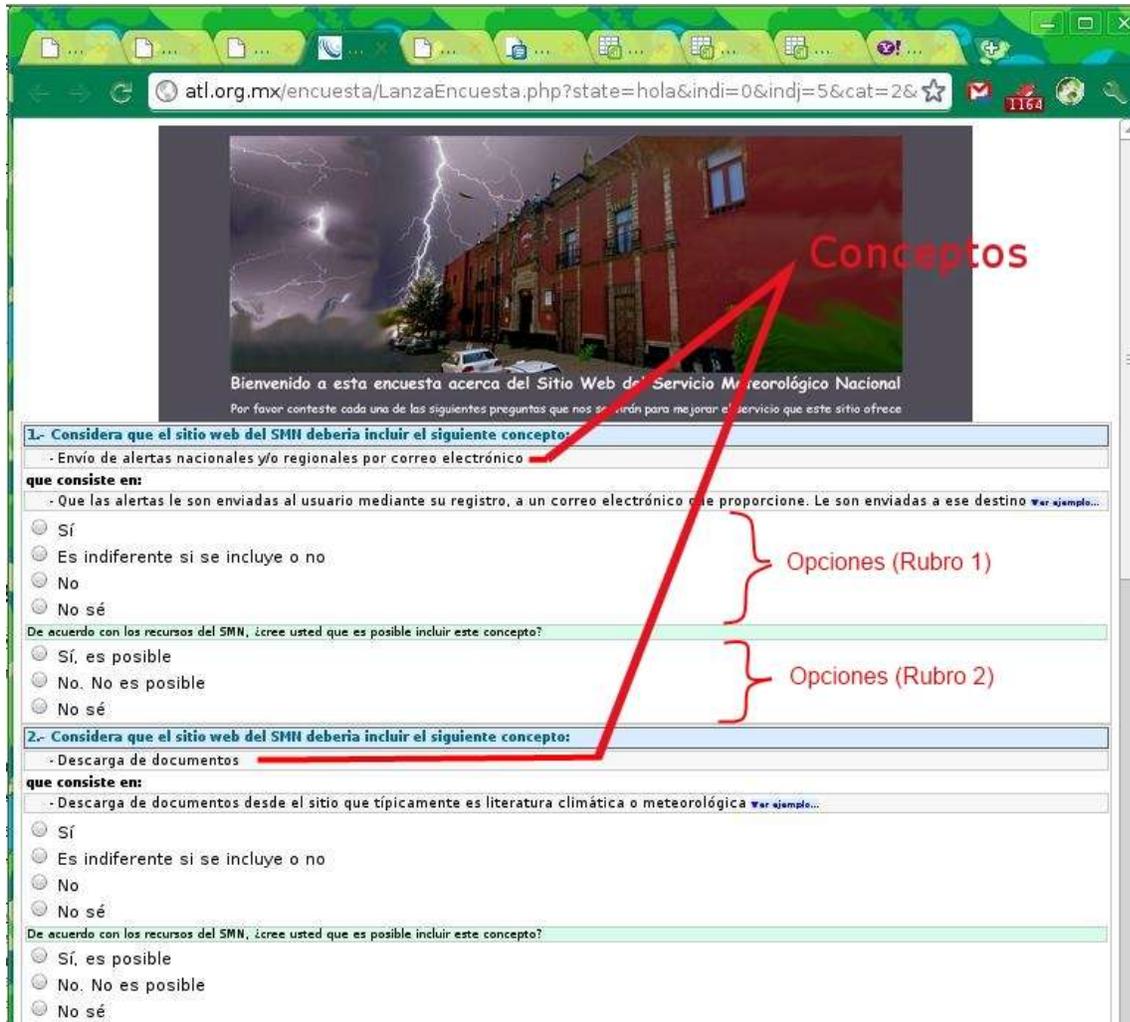
Para cada concepto se muestra, además de éste, el texto correspondiente a su definición, contenida en el campo “descripcion”, una liga (hyperlink) a la ilustración, señalada por el URL dado en el campo “figura”, e introducida aquí mediante el texto “ver ejemplo”, y las opciones para cada uno de los dos rubros. Las opciones para el segundo rubro solamente se desplegarán si el encuestado es un *usuario experto*.

A la apariencia que se muestra en la Fig. 2, se llega mediante una serie de procedimientos, programados en el lenguaje PHP, que consultan y actualizan la base de datos, cuya estructura se ha mostrado en la Fig. 1. Estos procedimientos se lanzan desde una página Web, en cuya llamada se pasarán también algunos parámetros necesarios para el control de los procedimientos. Dado que la encuesta está formada por un número de conceptos que puede ser incómodo de desplegarse en una sola página *Web*, la encuesta estará dada por una serie de páginas *Web*, que muestran un cierto número de conceptos cada una. A continuación se muestra la forma como se invoca inicialmente la primera página de la encuesta y se describen, asimismo, los parámetros:

<http://URL-sitio/LanzaEncuesta.php?state=hola&indi=0&indj=n&cat=k>

Aquí, la URL-sitio es la dirección *Web* numérica o simbólica del servidor que aloja los procedimientos de la encuesta, por ejemplo: atl.imta.gob.mx. LanzaEncuesta.php es el procedimiento PHP inicial de la encuesta; state es una variable de control, para la cual el valor hola indica que es el punto de partida de la encuesta; indi es una variable de control mediante la cual se obtiene el índice inicial del primer concepto a desplegar en la página de la encuesta que se muestra, en el caso inicial su valor es 0, lo que estaría indicando el índice del primer concepto, es decir, 1; indj es el índice del concepto final a desplegar en la página de la encuesta que se esté mostrando, y además, la diferencia entre indj e indi indica el número de conceptos a desplegar por página, en este caso, n ; la variable de control cat indica la categoría de la persona encuestada, k , en este caso. Aparte de estas variables de

control, se puede, opcionalmente, dar una variable más, test, para correr la encuesta en forma de prueba, esto es, se correrá toda la interfaz de usuario sin hacer actualizaciones a la base de datos.



The screenshot shows a web browser window displaying a survey page. The URL is `atl.org.mx/encuesta/LanzaEncuesta.php?state=hola&indi=0&indj=5&cat=2&`. The page features a header image of a street scene with a lightning bolt striking a building. Below the image, there is a welcome message: "Bienvenido a esta encuesta acerca del Sitio Web del Servicio Meteorológico Nacional". The main content consists of two survey questions, each with a list of radio button options. Red annotations are present: a red arrow points from the word "Conceptos" to the first question, and another red arrow points from "Conceptos" to the second question. Brackets on the right side of the options list for each question are labeled "Opciones (Rubro 1)" and "Opciones (Rubro 2)".

Conceptos

Bienvenido a esta encuesta acerca del Sitio Web del Servicio Meteorológico Nacional
Por favor conteste cada una de las siguientes preguntas que nos servirán para mejorar el servicio que este sitio ofrece

1.- Considera que el sitio web del SMN debería incluir el siguiente concepto:

- Envío de alertas nacionales y/o regionales por correo electrónico

que consiste en:

- Que las alertas le son enviadas al usuario mediante su registro, a un correo electrónico que proporcione. Le son enviadas a ese destino [Ver ejemplo...](#)

Sí

Es indiferente si se incluye o no

No

No sé

De acuerdo con los recursos del SMN, ¿cree usted que es posible incluir este concepto?

Sí, es posible

No, No es posible

No sé

2.- Considera que el sitio web del SMN debería incluir el siguiente concepto:

- Descarga de documentos

que consiste en:

- Descarga de documentos desde el sitio que típicamente es literatura climática o meteorológica [Ver ejemplo...](#)

Sí

Es indiferente si se incluye o no

No

No sé

De acuerdo con los recursos del SMN, ¿cree usted que es posible incluir este concepto?

Sí, es posible

No, No es posible

No sé

Opciones (Rubro 1)

Opciones (Rubro 2)

Figura 2. Presentación de los conceptos de la encuesta a los encuestados

5. La Manipulación de la Base de Datos

La médula de este sistema está la consulta y las actualizaciones a la base de datos, que se dan como resultado de la aplicación de la encuesta. La Fig. 3 muestra la arquitectura del sistema que implementa este requerimiento. La idea general es que, mediante consultas (*queries*) a la base de datos se presentan al usuario (encuestado), por medio de una página *Web*, una a una las preguntas correspondientes a cada uno de los 27 conceptos de la encuesta; éste elige su respuesta mediante la misma página, y el sistema se encarga de traducir esta respuesta en un *query* de modificación a la base de datos.

El lenguaje PHP se eligió como el fundamento programático del sistema. Una de las cualidades por las que se hizo esta elección, es que provee de una interfaz muy cómoda entre el servidor *Web - Apache*, en este caso– y el manejador de la base de datos (*MySQL*). En este sentido, PHP provee de varios mecanismos para la consulta y la modificación de la base de datos. En el caso del presente sistema se eligió la biblioteca ADODB que asimila los protocolos manejados para bases de datos en los sistemas operativos MS Windows.

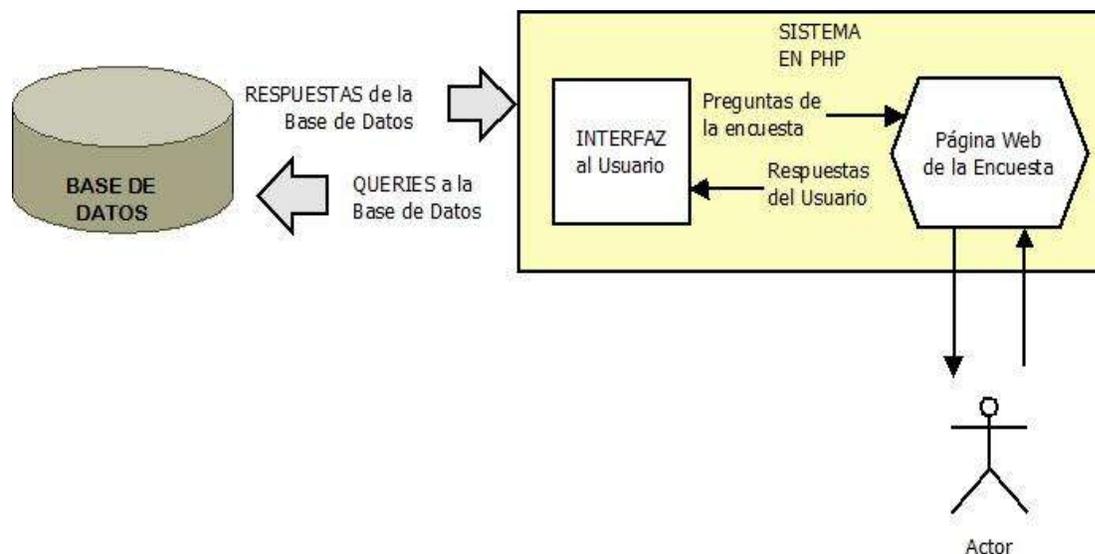


Figura 3. Arquitectura del sistema de encuesta

Para tener acceso a la base de datos mediante los protocolos establecidos por la biblioteca ADODB, se elaboraron dos funciones en PHP que abren y cierran la base de datos como se muestra a continuación:

```
//INICIO:JSS: Funciones para apertura y cerrado de la base de datos:
// Para abrir y cerrar la base de datos:

function &E_DB_open(){ // Abre la base de datos
// Como resultado de la función entregaré la conexión también
global $host;
global $user;
global $pass;
global $base;
global $conexion;

$conexion =& ADONewConnection("mysql"); // Selección de manejador
if($conexion->PConnect($host, $user, $pass, $base)) // Estableciendo conexión
// fue correcto, por lo tanto:
return $conexion;

// No se abrió correctamente la BD, por tanto mensaje y salir c/false
echo "Falló la conexión\n <BR />";
return null;
}

function E_DB_close()
{ // Cierra la base de datos
global $conexion;
$conexion->Close();
}
```

Aquí las variables \$host, \$user, \$pass y \$base son variables globales en las que se han establecido los parámetros de acceso a la base de datos de la encuesta. Internamente, para PHP la base de datos se identifica exclusivamente por medio de la variable global \$conexion. En otras partes del código, como se verá más adelante, se mantiene el acceso a la base de datos mediante los mecanismos provistos por la biblioteca ADODB.

La interacción con el usuario se modelará mediante un autómata de estados finitos, tal como se muestra en la Fig. 4. El autómata consta de tres estados, cada uno de los cuales representa una página *Web*, a la que se llega mediante las acciones etiquetadas en las aristas del diagrama como: “hola”, “proceso” y “adios”.

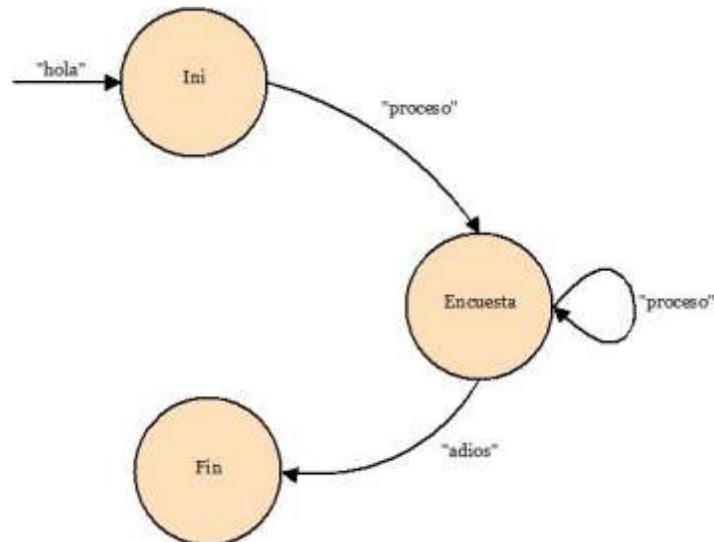


Figura 4. Autómata de estados finitos del proceso de la encuesta

Los estados del sistema son como sigue:

- **Ini:** Este es el estado inicial. A éste se ingresa indicando en la dirección de la página, en la variable state el valor hola:

<http://URL-sitio/LanzaEncuesta.php?state=hola&indi=0&indj=n&cat=k>

Este estado representa el despliegue de una portada de la encuesta y las preguntas correspondientes a los primeros n conceptos de la encuesta, en una típica forma de HTML. La forma contiene al final un botón para habilitar la captura de los datos ingresados en ella. Una vez que se colectan las respuestas proporcionadas por el usuario encuestado a cada una de las preguntas desplegadas en la página, el sistema vuelve a ser automáticamente invocado para pasar al siguiente estado, pero ahora con:

<http://URL-sitio/LanzaEncuesta.php?state=proceso&indi=i&indj=j&cat=k>

y ésta es la forma como se transfiere el control al estado o página siguiente.

- **Encuesta:** Este es el estado de producción de la encuesta y a él se ingresa indicando en la dirección de la página, en la variable state el valor proceso, como se vio anteriormente. En

este estado se despliegan otros n conceptos de la encuesta, se colectan las respuestas proporcionadas por el usuario encuestado a cada una de las preguntas desplegadas en la página y se vuelve a invocar el sistema. En este caso, el sistema lleva una cuenta de los conceptos que se han desplegado en total, de tal modo que, o se invoca para seguir emitiendo páginas de “proceso”, esto es, permanece en el mismo estado, o lo hace para concluir la encuesta. Esta situación se ilustra en la Fig. 4, mediante las dos aristas que salen del estado “Encuesta”. Nótese ahí, que la arista etiquetada como “proceso” obliga a permanecer en el mismo estado, mientras que la etiquetada como “adiós” indica la transición al estado final “Fin”. En este caso, la invocación del sistema se hace automáticamente con:

<http://URL-sitio/LanzaEncuesta.php?state=adios&indi=i&indj=j&cat=k>

- **Fin:** Este es el estado final del sistema. Se llega a él cuando se han concluido la encuesta y consiste simplemente en un mensaje de agradecimiento al usuario encuestado por su participación en la encuesta.

Este sencillo autómata se implementa en PHP con un código como el siguiente:

```
switch ($_GET["state"]) {  
  case "hola":  
    saluda_hija_no_seas_grosera();  
    prepara_comida();  
    break;  
  case "proceso":  
    come_pues();  
    prepara_comida();  
    break;  
  case "adios":  
    come_pues();  
    ahora_despidete();  
    break;  
  default: // Este es importante tenerlo para detectar posibles errores  
    echo "Desconozco ese estado: $_GET['state']\n <br/>";  
}
```

Donde `$_GET["state"]` es una variable, parte de un arreglo que contiene las variables que se proveen en la invocación de una página *Web*. En este caso, esa variable contiene el texto que identifica el estado del autómata al que se desea ingresar.

Las actividades que realiza el autómata se agrupan en cuatro procedimientos que se combinan de distinta manera de acuerdo con el estado. Ellos son: `saluda_hija_no_seas_grosera()`, `prepara_comida()`, `come_pues()` y `ahora_despidete()`. En seguida se describe cada uno de estos procedimientos:

- **saluda_hija_no_seas_grosera()**. Esta función se encarga de algunas inicializaciones y le da la bienvenida al usuario encuestado. Su código, en el cual se insertan comentarios explicativos pertinentes, es como sigue:

```
function saluda_hija_no_seas_grosera() {
    global $conexion; // Conexión a la base de datos
    global $nConcepts; // Número de conceptos en la encuesta
    global $testing; // Si es caso de prueba; $testing=0, indica que no es prueba
    global $IP; // La IP de la máquina del encuestado
    global $cat; // La categoría del encuestado $cat=2 es experto

    // El siguiente código en HTML es para desplegar en la página una imagen de
    // bienvenida a la encuesta, con una breve instrucción para contestar la
    // encuesta.
    ?>
    <CENTER>
    <IMG src="images/SMNLighting1.jpg"
        alt="Edificio del SMN entre rayos"
        width=800
        height=292 >
    </CENTER>
    <?

    // En el siguiente código se arma un query a la base de datos, y se ejecuta
    // mediante los procedimientos provistos por ADODB.
    // El query se arma en un string de caracteres, y es una expresión en el
    // lenguaje para consultas a bases de datos SQL. Los queries de consulta,
    // generalmente toman como argumento una tabla que es directamente una
    // tabla de la base de datos o proveniente de otro query de consulta, ya que en
```

```
// general, éstos también arrojan como resultado otra tabla, con varios campos
// o columnas y varios registros o renglones.
// En este caso, el query es muy simple y consiste en
// contar el número de registros en la tabla concpt, que es igual al número de
// conceptos de la encuesta.
$query = "SELECT COUNT(*) FROM concpt";
$rs = $conexion->Execute($query); // Se ejecuta el query
$nConcepts = $rs->fields[0]; // Resultado del query
// Note que la variable $nConcepts es global; además es una que se irá pasando
// entre estado y estado del autómata.
```

```
// Si no se está en pruebas, se actualiza la base de datos con la
// información del encuestado
switch ($testing) {
  case 2:
    // En este caso no se hará absolutamente nada.
    // Es simplemente un caso para que algunos funcionarios
    // chequen las preguntas de la encuesta.
    break;
  case 1:
    // Cosas que se quieran hacer o desplegar en el caso
    // de prueba 1
    break;
  case 0: // El caso 0 es también el default
  default:
    // Aquí no es prueba, por tanto se actualiza
    // la base de datos:
    actualiza_encuestado($IP, $cat); // Ver código abajo
}
} // FIN de la función
```

actualiza_encuestado(\$IP, \$cat) Es una función de servicio que se encarga de actualizar la información del usuario encuestado en la base de datos, en cualquiera de dos casos:

- (1) De plano no se encontraba antes de la Base de Datos.
- (2) Si se encontraba, pero su categoría ahora es superior a la que tiene registrada en la Base de Datos.

Su código es como sigue:

```
function actualiza_encuestado ($ip, $cat) {
    global $conexion; // Variable que liga con la Base de Datos

    // Se arma query para checar si ya está esa IP en la tabla "Encuestado" de la
    // Base de Datos. En este caso el query es para seleccionar de la tabla todos los
    // registros en los que su campo 'ip' coincida con el valor de la variable $ip. Por
    // supuesto que la consulta arrojará un solo registro, en caso de que exista.
    $query = "SELECT * FROM encuestado WHERE ip='$ip'";
    $rs = $conexion->Execute($query); // Se ejecuta el query
    $query = "";

    // Se prepara el query de inserción. Nótese que hay dos casos, a saber, la IP ya
    // se encontraba registrada y sólo se actualiza el registro en caso de que haya
    // cambiado la categoría del mismo o se trata de una nueva IP que se tiene que
    // dar de alta. Otra nota importante es que en esta etapa se arma el query, pero
    // su ejecución se pospone para la siguiente etapa.
    if (!$rs->EOF) { // Sí, está en la BD
        // sólo nos interesa actualizar en el caso de que la categoría de entrada
        // ahora es mayor
        if ($cat > $rs->fields[1]) {
            $query = "REPLACE INTO encuestado (ip,id_categoria) VALUES ('$ip','$cat')";
        }
    }
    else { // No hay tal registro en la base de datos, por tanto, se prepara el query
        // para la inserción del registro:
        $query = "INSERT INTO encuestado (ip,id_categoria) VALUES ('$ip','$cat')";
    }

    // Si se ha generado o creado el query en el código anterior, se procede a
    // ejecutarlo:
    if ($query != "") { // Hay o no hay query?
        if ($conexion->Execute($query) === false) {
            // Hubo un error; por tanto se notifica
            echo "ERROR en la insercion en la BD \n <br/>";
            return;
        }
    }
}
```

```
}  
} // FIN de la función
```

- **prepara_comida()**. Esta función consiste en la preparación de la Forma que se le presentará al usuario para que la llene. A continuación se muestra su código con comentarios explicativos donde es pertinente:

```
function prepara_comida() {  
    global $indi; // Índice inicial del concepto a exhibir - 1  
    global $indj; // Índice final del concepto a exhibir  
    global $cat; // Categoría del usuario encuestado  
    global $nConcepts; // Número total de conceptos en la base de datos  
    global $testing; // Indica si se trata de un caso de prueba  
  
    // En seguida se calculan los índices inicial y final de los conceptos de la encuesta  
    // que se mostrarán en la página Web. Dado que el número de conceptos podría  
    // provocar una página muy cargada, se decidió partir la encuesta en bloques  
    // que contienen un determinado número de conceptos, e ir las exhibiendo y  
    // colectando las respuestas una secuencia. El índice final va truncado al número  
    // total de conceptos contenidos en la encuesta, así:  
    $final = $indj < $nConcepts ? $indj : $nConcepts;  
    // Por razones de cálculo, el índice inicial ($inicio) es $indi incrementado en 1  
    $inicio = $indi + 1;  
    // Calculamos para la nueva acción al hacer Submit a la Forma que se genera:  
    $i = $indj; // a partir de aquí se empezará  
    $j = 2*$indj - $indi; // este es el nuevo marcador final  
  
    // Inicializamos $action con las cosas que no varían. Esto es, el siguiente bloque  
    // de conceptos, indicado por $i, $j, que ahora pasarán a ser  
    // los nuevos $indi, $indj:  
    $action =  
    "LanzaEncuesta.php?test=$testing&indi=$i&indj=$j&cat=$cat&nConcepts=$nConcepts&";  
  
    // Y ahora le agregamos lo que cambia. Esto básicamente es la determinación  
    // del siguiente estado al que se pasará en la siguiente llamada, que solamente  
    // depende de si se ha llegado al final de la encuesta.  
    $action .= ($final == $nConcepts) // esto indica que aquí terminaríamos  
    ?
```

```
        "state=adios"  
        :  
        "state=proceso";  
// y ahora simplemente se pasa a generar el tramo de la encuesta  
// correspondiente con:  
generaConceptos($inicio, $final, $cat, $action);  
  
} // FIN de la función
```

generaConceptos(\$inicio, \$final, \$cat, \$action) Es una función de servicio que consulta en la base de datos todos los conceptos comprendidos entre los índices \$inicio y \$final. Para cada uno de estos conceptos genera la pregunta correspondiente de acuerdo con la categoría (\$cat) de la encuesta, y la presenta en la página Web correspondiente al bloque que se está mostrando. Su código es como sigue:

```
function generaConceptos($idIni, $idFin, $idCat, $action) {  
    global $conexion; // Variable que liga con la Base de Datos  
  
    // En seguida se arma el query para la base de datos. Éste consiste en una  
    // expresión en el lenguaje de consulta a bases de datos SQL que ordena buscar  
    // todos los registros de la tabla 'concept' cuyo campo 'id_concept' se encuentre  
    // entre los valores dados por las variables $idIni e $idFin. Se indica también que  
    // los registros de la tabla resultante deberán estar ordenados de acuerdo con el  
    // valor del campo 'id_concept'.  
    $query =  
    "SELECT * FROM concept WHERE id_concept".  
    "BETWEEN $idIni AND $idFin ORDER BY id_concept";  
  
    // Ahora se procede a ejecutar el query:  
    $rs = $conexion->Execute($query);  
  
    // En seguida se "escupe" el código HTML correspondiente a una típica forma  
    // de entrada de HTML. En este caso, se usará el método POST, y la acción  
    // asociada, contenida en el argumento de la función $action, es la que se ha  
    // calculado antes en la función prepara_comida(), que llama a esta función.  
    // Este encabezado de la forma establece, además, que al enviar la forma se  
    // chequen las entradas proporcionadas por el usuario mediante el llamado a la  
    // función de JavaScript validarEntradas(this).
```

```
// -----INICIO de encabezado de forma HTML----
?>
<form name="form1"
  method="post"
  action="<?=$action;?>"
  onsubmit="return validarEntradas(this);">
<?
// -----FIN de encabezado de forma HTML----

// La variable $rs trae el resulting-set del query de SQL que se ejecutó
// anteriormente. Éste consiste en una tabla, cada uno de cuyos registros o
// renglones corresponde con cada uno de los conceptos de la encuesta. Aquí
// se hace, entonces, un ciclo que toma uno a uno esos conceptos y los pasa a la
// función generaConcepto($rs, $idCat), que se encarga de generar el código
// HTML de la forma correspondiente a un solo concepto individual.
while (!$rs->EOF) {
  generaConcepto($rs, $idCat);
  $rs->MoveNext();
} /* END-while (!$rs->.. */

// Ahora, al final de la forma se incluye en HTML un botón (botón de Submit)
// que llevará la etiqueta "Enviar" y cuya función será enviar la forma una vez
// que se ha llenado, y aquí concluye la forma HTML:
?>
<input type="submit" value="Enviar" />
</form>
<?

} // FIN de la función
```

validarEntradas(this) Es una función en el lenguaje JavaScript, cuyo propósito es verificar que los datos introducidos por el usuario encuestado estén completos. Dado que este lenguaje es orientado a objetos, el único argumento de la función es el objeto correspondiente a la forma que se está procesando. La Encuesta se ha diseñado de tal manera que las respuestas a cada pregunta se dan en un arreglo de "*radio buttons*", así que la tarea de la función consiste en checar que en todos los grupos de *radio buttons* esté uno checado. Aquí se muestra el código de esta función:

```
function validarEntradas(form) {
    var stkmsj = ""; // Este mensaje contendrá los grupos
        // que no hayan sido checados
    var ya_revisado = ""; // Registro del último grupo
        // revisado, inicialmente vacío
    var notifica = false; // si se notifica o no la
        // omisión de un grupo
    var revisando = ""; // El grupo que se está revisando

    // Resulta que el indexado como un arreglo asociativo
    // para form.elements no funciona para la porquería de
    // ChafaSoft (Microsoft) IE, se cambiará a indexado numérico. Pero
    // se dejará comentado, con "// [X]IE >>", el código original que sí
    // funcionaba en todos los otros browsers.

    // [X]IE >>for (var x in form.elements)
    for (var i = 0; i < form.elements.length; i++) {
        // Sólo interesan elementos de tipo radio
        // [X]IE >>if (form.elements[x].type == "radio") {
        if (form.elements[i].type == "radio") {
            // [X]IE >>revisando = form.elements[x].name; // el nombre del grupo
            revisando = form.elements[i].name; // el nombre del grupo
            if (ya_revisado != revisando) { // ¿es diferente del ya revisado?
                // Este es el inicio de un grupo de radio buttons nuevo:

                if (notifica) {
                    // En todos los elementos del grupo anterior no se ha checado
                    // ninguno de los radio buttons. En este caso se introduce el nombre
                    // del grupo en cuestion en el string que formará parte del mensaje de
                    // diagnóstico:
                    stkmsj += ya_revisado + " "; // Se agrega
                }
                // El elemento -radio button-- que se está revisano no está checado;
                // eso se notifica en la variable 'notifica':
                // [X]IE >>notifica = !(form.elements[x].checked);
                notifica = !(form.elements[i].checked);
            }
        }
    }
}
```

```
else {  
    // Aquí se continúa sobre el mismo grupo de radio buttons; esto  
    // significa que la variable 'notifica' depende de lo encontrado antes y  
    // del valor del elemento revisado actualmente (chechado o no chechado):  
    // [X]IE >>notifica = (form.elements[x].checked) ? false : notifica;  
    notifica = (form.elements[i].checked) ? false : notifica;  
}  
// Se mantiene registrado el grupo que ya se terminó de revisar, para la  
// siguiente vuelta del ciclo:  
ya_revisado = revisando;  
} // END if (...type == "radio")  
} // END for  
  
if (notifica)  
    stkmsj += ya_revisado; // En caso de notificación, se agrega último grupo  
  
if (stkmsj == "") {  
    // No se detectó ningún problema, por tanto, la función regresa true  
    return true;  
}  
  
// Al menos hay un grupo para el que no se ha checado uno de sus radio  
// buttons. Por tanto, se manda mensaje con grupos no checados. Esto es,  
// preguntas de la encuesta no contestadas:  
alert (  
    "Le ha faltado contestar las preguntas: \n" + legible(stkmsj.mtrim()));  
  
// Y el valor de la función en este caso es falso.  
return false;  
} // FIN de la función (JavaScript)
```

generaConcepto(\$rs, \$idCat) Es una función en el lenguaje PHP, cuyo propósito es tomar un solo registro del resulting set (\$rs) de la base de datos, que corresponde a un concepto de la encuesta, y generar el desplegado correspondiente (HTML) en la página Web que se está construyendo. Aquí se muestra el código de esta función:

```
function generaConcepto($rs, $idCat) {  
    // El $rs es el resulting set correspondiente a uno de los conceptos en  
    // la base de datos. Para ese concepto particular, se generan cada una de las
```



```
<?// <-Cierra HTML y abre PHP
// El siguiente código PHP, inserta el valor de $rs->fields[2] en el código
// HTML. Este campo del resulting set ($rs) contiene la descripción
// textual del concepto.
=$rs->fields[2];
// En seguida se averigua si viene el campo correspondiente a una figura
// ilustrativa para el concepto:
if(($rs->fields[3] != null) && ($rs->fields[3] != "null") &&
    strlen(trim($rs->fields[3])) > 0) {
// Sí viene figura, por tanto se incluye un link que le permita al usuario
// acceder a la figura, si así lo desea, el siguiente código es HTML (note
// que se inserta desde PHP el valor de $rs->fields[3], que es el URL de la
// figura) :
?>
<span class="img_bd">
<a href="show_image.php?img=<?=$rs->fields[3];?>"
    target="_blank" class="lbOn">
<strong>Ver ejemplo...</strong></a>
</span> <?
} ?>
</td>
</tr>
<? // <-Cierra HTML y abre PHP
// Ya que la categoría 2 incluye a la 1 se deben mostrar para ese caso
// las opciones de las 2 categorías.
for ($id=1; $id <= $idCat; $id++) { ?>
<tr>
<td> <?
// Y aquí se generan una a una las opciones correspondientes al concepto
// que se está manejando en esta función:
generaOpciones($id,$idConcpt); ?>
</td>
</tr> <?
} ?>
</table> <?
} // FIN de la función
```

generaOpciones(\$id,\$idConcpt) Es una función en el lenguaje PHP, cuyo propósito es tomar cada una de las opciones del concepto \$idConcpt en la base de datos, y generar el desplegado

correspondiente (HTML) en la página Web que se está construyendo. Aquí se muestra el código de esta función:

```
function generaOpciones($idCat,$num) {
    global $conexion; // Conexión a la base de datos--
        // aquí se entra con conexión ya establecida
    // La operación se tiene que hacer en 2 queries, una para el texto descriptivo
    // correspondiente a la categoría (que puede venir o no venir), y otra para cada
    // una de las opciones:
    $query_res =
        "SELECT texto as texto FROM categoria WHERE id_categoria = $idCat";
    // Se ejecuta el query:
    $rs_res = $conexion->Execute($query_res);
    if(!$rs_res->EOF) { // Aquí entramos si viene el texto:
        if(strlen(trim($rs_res->fields["texto"])) > 0) {
            // Se "escape" el texto al HTML que se está generando:
            echo "<div class='texto_categoria'>".$rs_res->fields["texto"]."</div>";
        }
    }

    // Ahora el query para extraer las opciones:

    $query_res =
        "SELECT * FROM opciones WHERE id_categoria = $idCat ORDER BY valor DESC";
    $rs_res = $conexion->Execute($query_res); // <-Se ejecuta query
    // Notemos, en lo que sigue, que el nombre de cada grupo de radios, se
    // construye con el texto "res" seguido del número de concepto, luego un guión
    // y la categoría P.ej. res3-1 corresponde a la pregunta 3 y la categoría 1.
    while(!$rs_res->EOF) { ?>
        <label>
        <input type="radio"
            name
            <? // <-Cierra HTML y abre PHP
            // Aquí se arma el nombre, como se indicó arriba, y más abajo se
            // inserta su valor (value), como campo 1 del resulting set
            // ($rs_res->fields[1]), que es el valor numérico de la opción
            // correspondiente :
            ="\"res${num}-${idCat}\"";
```

```
?>
value="<?=$rs_res->fields[1]; ?>"
/>
<span><? <-Cierra HTML y abre PHP
// El texto de la etiqueta, es el texto de la opción, que a su vez es el valor del
// resulting set en su campo 3:
    =$rs_res->fields[3];?></span>
</label> <br /> <?
$rs_res->MoveNext(); // Nos movemos al siguiente registro que contiene la
    // siguiente opción.
} // END While
} // FIN de la función
```

- **come_pues()**. Esta función consiste en consumir lo que se ha preparado con anterioridad. Es decir, esta función se encarga de interpretar las respuestas a la encuesta entregadas por el usuario y verter ese contenido de regreso a la base de datos. A continuación se muestra su código con comentarios explicativos donde es pertinente:

```
function come_pues() {
// Esta función "come" la información que ha sido vertida en la forma
// previamente, esto es, la registra en la base de datos.
// Pero, primero obtenemos de la URL algunas variables que faltan y que se han
// enviado de estados anteriores.
global $nConcepts; // Núm. de conceptos en la base de datos
global $testing; // Se trata de una prueba o ya se actualiza la BD ?
global $IP;
global $cat;
$nConcepts = $_GET["nConcepts"];
switch ($testing) {
case 2:
    // En este caso no se hará absolutamente nada.
    // Es simplemente un caso para que algunos funcionarios
    // chequen las preguntas de la encuesta.
    break;
case 1:
    // Y para el caso de prueba sólo mostraremos el
    // contenido de la variable $_POST y otros datos
```

```
echo "Lo sig. se insertaría en IP:$IP, cat:$cat <br/>";  
echo '<pre>';  
// Se despliega el contenido de $_POST con los caracteres especiales  
// convertidos a entidades HTML:  
echo htmlspecialchars(print_r($_POST, true));  
echo '</pre>';  
break;  
case 0:  
default:  
    // Mediante la función descifra_post, se descifra lo que hay en la variable  
    // $_POST, que es lo que el usuario ha llenado en la forma provista, y se  
    // registra en la base de datos:  
    descifra_post($IP);  
}  
} // FIN de la función
```

descifra_post(\$IP) Es una función en el lenguaje PHP, cuyo propósito es interpretar el contenido de lo que se ha recuperado de la forma llenada por el encuestado y que viene guardado en la variable superglobal \$_POST. El formato de cada uno de los índices (\$key) de \$_POST es: res<n-concpt>-<cat>. Por ejemplo, res25-2, que estaría indicando el número de concepto 25 y la categoría 2. Aquí se muestra el código de esta función:

```
function descifra_post($ip) {  
    // Recordemos que la IP de la máquina, ahora en el argumento $IP de la función,  
    // es la que identificará al usuario encuestado.  
    //-----  
    global $conexion; // Variable global que identifica la base de datos.  
  
    // La variable superglobal $_POST, es un arreglo de relaciones ($key, $val),  
    // donde $key se “desarma” de acuerdo con lo dicho antes, y $val es el valor  
    // numérico de la opción seleccionada por el usuario. El ‘foreach’ siguiente, es  
    // para recorrer y desmenuzar todo ese arreglo:  
    foreach ($_POST as $key => $val) { // Para cada pareja ($key,$val)  
        // Se elimina el "res" del string -> p.ej. quedaría: 25-2  
        $y = substr($key, 3);  
        // Ahora se separa mediante "-" -> p.ej. $a[0]==25, $a[1]==2  
        $a = preg_split ("/-/", $y);  
        // Sólo por claridad se hacen los siguientes ‘sinónimos’ de variables:
```

```
$concepto = & $a[0]; // El número de concepto
$nivel = & $a[1]; // Este es también la categoría

// En un solo registro vienen dos respuestas del usuario para cada concepto:
// 1) calif_quiere, 2) calif_puede. Si el usuario no es experto, la calif_puede
// quedará nula, en caso contrario, otro de los pares ($key,$val) contendrá la
// respuesta, y será necesario actualizar el registro. De este modo,
// ahora sí, preparamos el query
switch ($nivel) { // O sea la categoría
  case 1:
    $query = "REPLACE INTO encuesta
              (ip, id_concpt, calif_quiere, calif_puede)
              VALUES ('$ip', $concepto, $val, null)";
    break;
  case 2:
    $query = "UPDATE encuesta SET calif_puede=$val
              WHERE ip='$ip' AND id_concpt=$concepto";
    break;
  default:
    echo "categoria desconocida \n <br/>";
    return;
}
//
// Se ejecuta el query:
if ($conexion->Execute($query) === false) {
  // Hubo un error:
  echo "ERROR en la actualizacion de la BD \n <br/>";
  return;
}
}
} // FIN de la función
```

- **ahora_despidete ()**. Esta función consiste solamente en el envío de un mensaje de despedida y agradecimiento. Su código es como sigue:

```
function ahora_despidete() {
  // Sólo un mensajito de despedida; para ello se sale del modo PHP y se entra en
  // el modo HTML:
  ?>
  <CENTER>
```

```
<IMG src="images/SMNLighting2.jpg"  
  alt="Edificio del SMN entre rayos"  
  width=800  
  height=496 >  
</CENTER>  
<?  
} // FIN de la función
```

6. Procedimiento para el Análisis de la Encuesta

Los datos crudos, resultado de la encuesta, contenidos en la base de datos requieren de un procesamiento que los haga más inteligibles. Existen varias formas para desarrollarlo, desde la consulta y programación con un lenguaje como PHP, hasta las consultas directas mediante *queries* en el lenguaje de consulta de bases de datos SQL. En el presente caso se optó por esta segunda opción, que resultó ser la más compacta. En seguida se da una breve descripción de este procedimiento.

Cuando se ejecutan *queries* de consulta a una base de datos mediante SQL, lo primero que se debe tener en cuenta es que la base de datos está compuesta por un conjunto de tablas (relaciones), y a su vez, los resultados de los *queries* serán también tablas.

Los resultados de la encuesta se encuentran básicamente en la tabla Encuesta. En la Fig. 1 y en la sección 3 de este documento, se puede ver que esta tabla tiene los siguientes campos: id, ip, id_concpt, calif_quiere, calif_puede. Para un ip dado (usuario encuestado dado), habrá tantos registros como conceptos hay en la base de datos y las columnas (campos) calif_quiere y calif_puede, contendrán los valores que el usuario hubiere otorgado a esos conceptos en esos rubros. Esto se muestra en la Fig. 5, donde por claridad se ha omitido el campo id, y en vez de los valores numéricos de 'ip' y 'calificación' se ha puesto una representación simbólica.

Lo que básicamente interesa saber es cuántos usuarios han contestado 'Sí', 'No', 'Indiferente', o 'No sé', para un concepto particular. Así, por ejemplo, el código en SQL para encontrar cuántos han contestado Sí para el concepto 2 es el siguiente:

```
(SELECT count(*)
FROM `encuesta` WHERE `id_concpt` = 2 AND `calif_quiere` = 3) AS Si
```

Aquí, la función 'count(*)' de SQL, se encarga de entregar el número de registros encontrados en vez de un listado o tabla de ellos; el valor 3, especificado para 'calif_quiere' es el que corresponde a la respuesta 'Sí' por parte del usuario. Si este procedimiento se aplica a lo que se ve en lo que se puede ver de la tabla mostrada en la Fig. 5, el resultado que se obtendría sería 2.

ip	id_concpt	calif_quiere	calif_puede
U1	1	Sí	No sé
U1	2	Sí	No
U1	3	No	No
⋮	⋮	⋮	⋮
U2	1	Sí	null
U2	2	No	null
U2	3	Indif	null
⋮	⋮		
U3	1	Sí	Sí
U3	2	Sí	Sí
U3	3	Sí	No sé
⋮	⋮	⋮	⋮

Figura 5. Una tabla de encuesta típica.

La cuestión es si esto se puede hacer en su conjunto para cada una de las distintas calificaciones posibles, para obtener como resultado una tabla como la del ejemplo que se muestra en la Fig. 6. Tomando en cuenta además que los primeros 32 registros de la tabla 'encuesta' se deben omitir, dado que fueron registros para un caso de prueba. La obtención de una tabla semejante para nuestro caso se hace mediante la operación JOIN de SQL, de acuerdo con el código que se muestra en seguida:

```
SELECT @ic:=t.id_concpt AS id_concpt,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_quiere` = 3) AS Si,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_quiere` = 2) AS Indif,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_quiere` = 1) AS No,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_quiere` = 0) AS NoSe
FROM `concpt` AS t
```

Y el código para 'calif_puede' es semejante:

```
SELECT @ic:=t.id_concpt AS id_concpt,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_puede` = 2) AS SiPuede,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_puede` = 1) AS NoPuede,
  (SELECT count(*) FROM `encuesta`
   WHERE `id` > 32 AND `id_concpt` = @ic AND `calif_puede` = 0) AS NoSe
FROM `concpt` AS t
```

id_concpt	Sí	Indif	No	No_sé
1	30	5	8	0
2	14	6	15	8
3	22	3	15	3
4	38	0	5	0
5	20	2	20	1
6	25	5	10	3
7	43	0	0	0
8	0	3	40	0
9	39	1	1	2
10	20	20	2	1
11	23	10	9	1
⋮	⋮	⋮	⋮	⋮

Figura 6. Ejemplo de respuestas, por concepto (calif_quiere).

7. Resumen del Análisis de la Encuesta

Una vez que se ha hecho el procesamiento indicado antes, con de la información cruda contenida en la Base de Datos de la encuesta, se obtiene una tabla como la mostrada en la Fig. 6. Una tabla de ese tipo ya es fácilmente manipulable por medio de un hoja de cálculo como Excel. De este modo, se obtuvieron los resultados que se muestran en la tabla de la Fig. 7 y en la gráfica de la Fig. 8, que se muestran a continuación:

id_concpt	Descripción	%QUIERE			
		Sí	Indif	No	No sé
1	[4] Envío de alertas nacionales y/o regionales por correo electrónico.	76%	16%	8%	0%
2	[5] Descarga de documentos.	88%	4%	8%	0%
3	[6] Personalización de servicios gratuitos mediante registro.	68%	12%	20%	0%
4	[8] Secciones especiales para actividades humanas.	56%	20%	20%	4%
5	[10] Búsqueda por código postal, localidad o centro de reunión.	72%	4%	20%	4%
6	[12] Muestras de videos e imágenes enviados por los usuarios con actividades afectadas o beneficiadas por el clima (Web 2.0).	56%	12%	20%	0%
7	[13] Noticias enviadas por los usuarios (Web 2.0).	52%	8%	24%	4%
8	[14] Noticias y recomendaciones clasificados por regiones geográficas.	76%	8%	4%	0%
9	[15] Noticias internacionales incluidas en el sitio.	56%	8%	24%	0%
10	[16] Herramientas informáticas, relacionadas con la meteorología, descargables para usuarios.	64%	4%	16%	4%
11	[18] Versiones del sitio adaptadas para dispositivos móviles.	72%	4%	8%	0%
12	[19] Unidades en Sistema Métrico Decimal y Sistema Inglés de Medidas.	52%	12%	20%	0%
13	[20] Recarga y actualización automática de la página cada x minutos.	72%	8%	4%	0%
14	[21] Tabla de pronósticos del estado del tiempo.	80%	4%	0%	0%
15	[22] Pronóstico: Día anterior (datos pronosticados con antelación y datos realmente observados).	68%	8%	4%	4%
16	[24] Pronóstico: Por hora durante el día de hoy.	56%	0%	16%	8%
17	[25] Pronóstico: Día de mañana.	76%	0%	4%	0%
18	[27] Pronóstico: Hasta a 7 días (incluido hoy).	72%	4%	4%	0%
19	[28] Pronóstico: Mes actual.	68%	8%	4%	0%
20	[30] Mapa de máximos y mínimos observados.	76%	4%	0%	0%
21	[34] Biblioteca y Librería.	60%	0%	12%	0%
22	[36] Blogs (Web 2.0).	40%	8%	24%	0%
23	[40] Nube de etiquetas (Web 2.0).	40%	8%	24%	0%
24	[41] Listado de últimas ciudades consultadas (Web 2.0).	44%	12%	16%	0%
25	[43] Inclusión de un widget en la página de los visitantes.	52%	4%	16%	0%
26	[44] Aprendizaje sobre meteorología.	60%	0%	8%	4%
27	[46] Investigación y publicaciones.	60%	0%	8%	4%

Figura 7. Tabla resumen de los resultados obtenidos en la encuesta

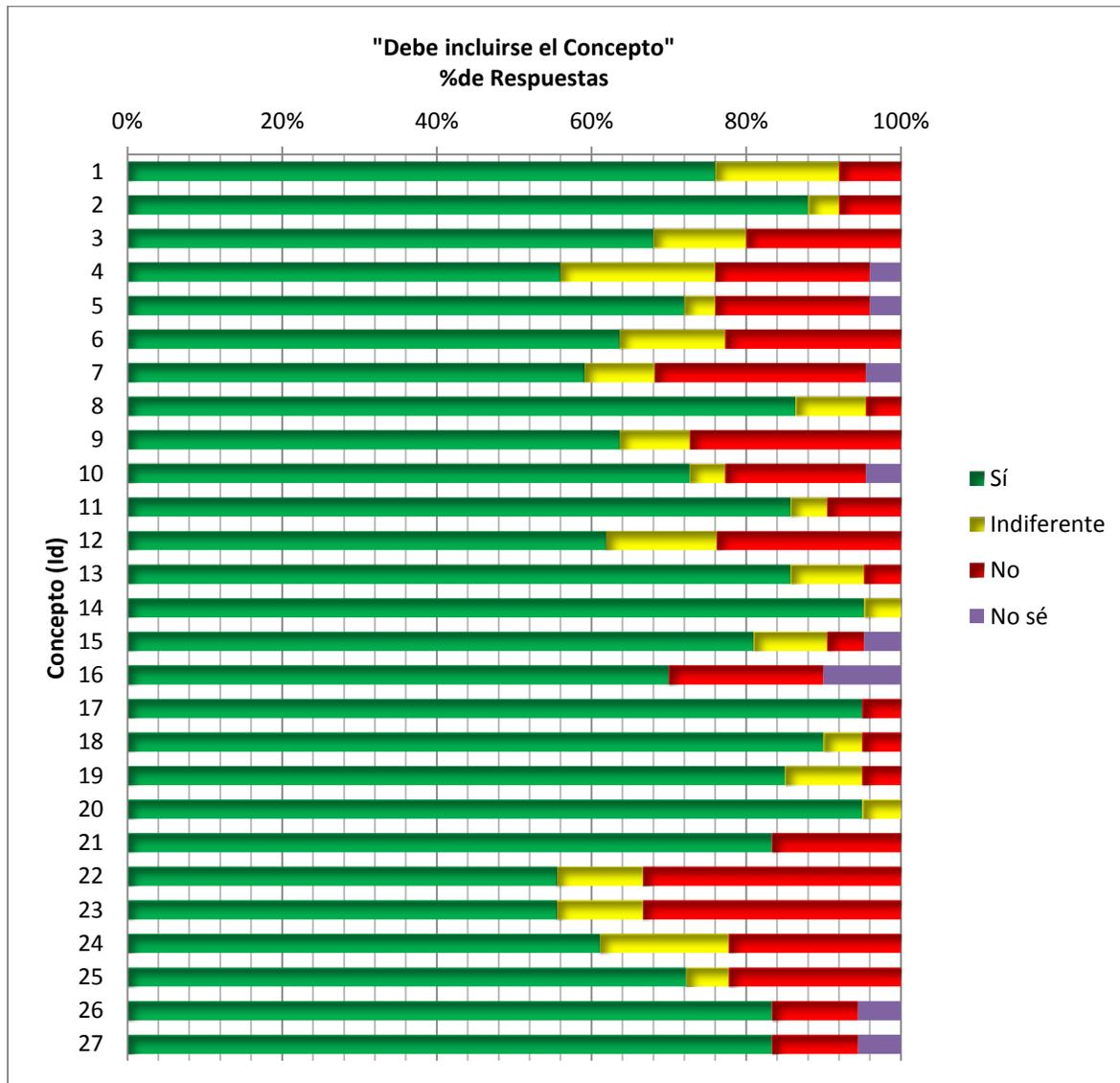


Figura 8. Gráfico con el resumen de respuestas en la encuesta

11 ANEXO 3: Prototipo de Migración del Portal Galileo a un Sistema de Gestión de Contenidos

Prototipo de Migración del Portal Galileo a un Sistema de Gestión de Contenidos

Informe final de trabajo



INFORME FINAL PROYECTO
TH 1114.2

SECRETARÍA DE
MEDIO AMBIENTE Y
RECURSOS NATURALES



Jiutepec, Morelos, 2011

Objetivo

Explorar el uso el Sistema de Gestión de Contenido **Joomla!** para facilitar el manejo de la información que será publicada en el portal *Galileo*.

Introducción

Joomla! es un Sistema de Gestión de Contenidos (CMS) reconocido mundialmente, que le ayuda a construir sitios web y otras aplicaciones en línea potentes. Lo mejor de todo, es que Joomla! es una solución de código abierto y está disponible libremente para cualquiera que desee utilizarlo.

¿Qué es un Sistema de Gestión de Contenidos (CMS)?

Un CMS, es un sistema de software para ordenador que permite organizar y facilitar la creación de documentos y otros contenidos de un modo cooperativo. Con frecuencia, un CMS es una aplicación web usada para gestionar sitios web y contenidos web. Joomla! para Usuarios Finales.

El paquete está diseñado para ser instalado fácilmente, no es necesario ser un programador. Una vez que se ha instalado y configurado Joomla!, resulta muy sencillo para usuarios sin conocimientos técnicos el añadir o editar contenido, subir imágenes, y gestionar los datos principales de su compañía u organización. Cualquiera con conocimientos sobre un procesador de palabras básico puede aprender fácilmente a gestionar un sitio Joomla!.

Mediante un navegador web será capaz de publicar fácilmente artículos o noticias, gestionar páginas de personal, listados de trabajo, imágenes de productos, y crear una cantidad ilimitada de secciones o páginas de contenido en su sitio web.

Instalación del ambiente LAMP (Linux, Apache, MySQL y PHP) en Ubuntu.

Para la instalación de **LAMP** el proceso fue el siguiente:

Dentro de una terminal ingresamos el siguiente comando:

```
sinue@servidor:~$ sudo apt-get install tasksel
```

Una vez hecho esto se procede a lanzar la aplicación desde la misma terminal.

```
sinue@servidor:~$ sudo tasksel
```

Seleccionamos la tarea **LAMP**, que se encargará automáticamente de instalarnos todos los paquetes y dependencias necesarias para la interacción entre las dependencias instaladas:

```
Package configuration
Software selection
You can choose to install one or more of the following
Choose software to install:
[ ] Cloud computing: Walrus storage service
[ ] Cloud computing: all-in-one cluster
[ ] Cloud computing: cluster controller
[ ] Cloud computing: node controller
[ ] Cloud computing: storage controller
[ ] Cloud computing: top-level cloud controller
[ ] DNS server
[ ] Edubuntu server
[ ] LAMP server
[ ] Mail server
[ ] OpenSSH server
[ ] PostgreSQL database
[*] Print server
[ ] Samba file server
[ ] Tomcat Java server
```

Figura 1: Instalación de LAMP

Después de esto, nos solicitará una contraseña para el administrador de la base de datos:

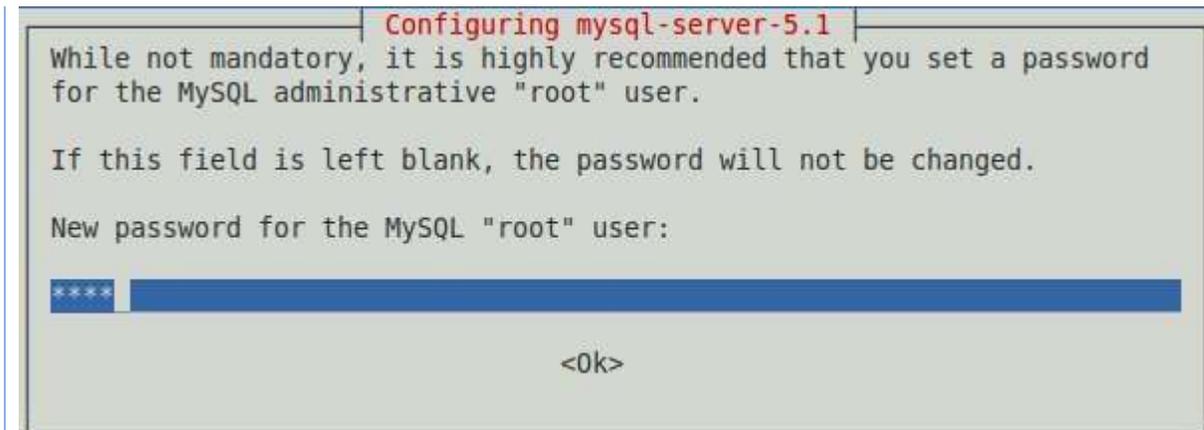


Figura 2: Configuración de la contraseña de MySQL.

Una vez terminada la instalación, vamos a comprobar que todo funciona como esperamos, así que crearemos un fichero *test.php* en el directorio */var/www/*, donde por defecto se encontrará nuestra web.

Cambiamos los permisos del directorio,

```
sinue@servidor:~$ cd /var/www/  
sinue@servidor:/var/www$ sudo chown -R user  
[sudo] password for user: *****
```

y creamos el fichero *test.php* incluyendo la siguiente línea,

```
sinue@servidor:/var/www$ nano test.php
```

Finalmente, reiniciamos el servidor Web

```
sinue@servidor:~$ sudo /etc/init.d/apache2 restart  
* Restarting web server apache2 ... waiting [ OK]
```

Instalación de Joomla! (versión 1.5.20).

Descargar la última versión estable de Joomla en español y lo colocamos dentro de una carpeta llamada "Joomla" en */var/www/*:

1.- Creando la carpeta:

```
# mkdir /var/www/Joomla
```

2.- Entrando al directorio:

```
# cd /var/www/Joomla:
```

3.- Descargamos y descomprimos la última versión de Joomla en español dentro de dicha carpeta:

```
# wget http://joomlancode.org/gf/download/frsrelease/12649/53618/joomla\_1.5.20-Spanish-pack\_completo.tar.gz
```

```
# tar zxvf Joomla_1.5.20-Spanish-pack_completo.tar.gz
```

Una vez hecho esto tenemos que crear una serie de variables que, aunque hay un método automático de permisos, ésta es la manera en que menos probabilidades hay de errores o malos funcionamientos. Dentro de `/var/www/joomla` tecleamos lo siguiente:

```
PLACES=' (esto nos arrojará un signo > en el que vamos a ir tecleando lo siguiente línea por línea)
>administrator/backups
>administrator/components
>administrator/modules
>administrator/templates
>cache
>components
>images
>images/banners
>images/stories
>language
>plugins
>media
```

```
>modules  
>templates' (este es un "" que indica que terminamos de ingresar variables)
```

Una vez que veamos nuestro prompt normal es necesario hacer los cambios de permisos necesarios para las variables que acabamos de crear, para eso tecleamos:

```
for i in $PLACES; do
```

Se nos muestra de nuevo el ">" y tecleamos lo siguiente:

```
sudo chown -R www-data:www-data $i
```

Y luego tecleamos:

```
done
```

y habremos terminado con los permisos de los directorios. Ahora toca el turno de los archivos lo cual hacemos de la siguiente manera:

```
# find . -type f -exec chmod 644 {} \;
```

Nota: Recordar que todo esto lo estamos haciendo dentro de /var/www/Joomla

Una vez ya concluidos estos sencillos pero sustanciosos pasos, procedemos a ingresar la siguiente dirección en el navegador de nuestra preferencia. <http://localhost/joomla>

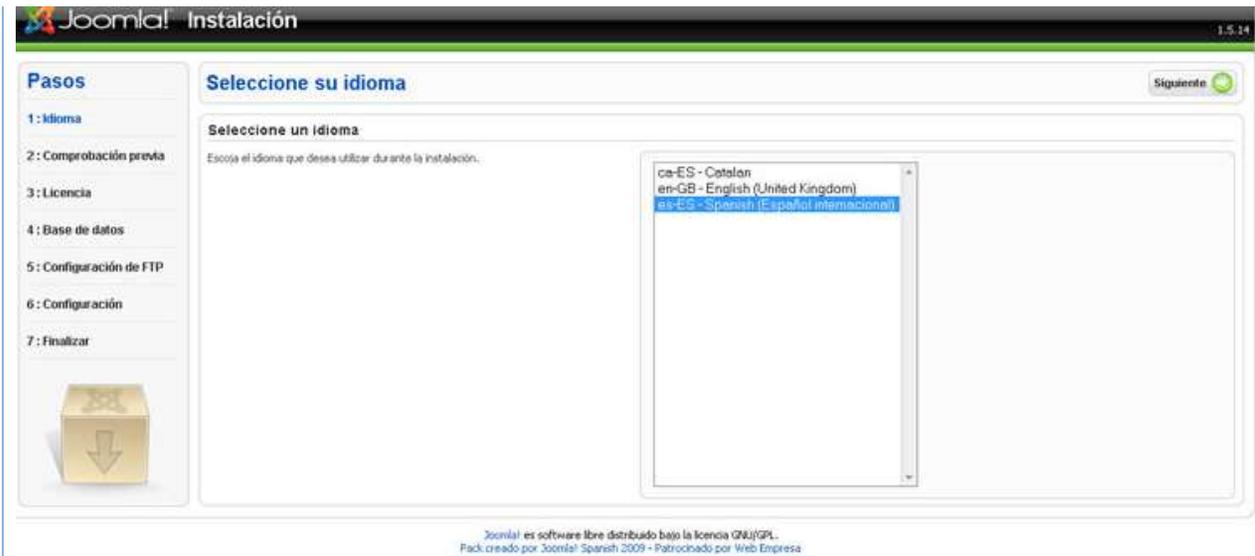


Figura 3: Pantalla inicial y paso No. 1 de la instalación de Joomla!

Seleccionamos el idioma a instalar, a continuación aparecerá una comprobación previa a la instalación, con ella podremos saber si contamos con los requisitos mínimos para llevar a cabo la instalación.



Figura 4: Paso No. 2, comprobación de los requisitos mínimos para la instalación.

El siguiente paso es el que nos encontramos en casi todas las cosas que instalamos: Aceptar la licencia.



Figura 5: Paso No. 3, aceptación de la licencia del software.

La configuración de la base de datos es muy sencilla, solo debemos especificar el nombre del servidor, Nombre de usuario y contraseña y el nombre de la base de datos.

Joomla! Instalación 1.5.14

Pasos

- Idioma
- Comprobación previa
- Licencia
- Base de datos**
- Configuración de FTP
- Configuración
- Finalizar

Configuración de la base de datos

Parámetros de la conexión:

Configurar Joomla! para que funcione en su servidor requiere de cuatro pasos sencillos...

Configuración básica

- Seleccionar el tipo de base de datos que utilizará en la lista desplegable (generalmente **mysql**).
- Escribir el nombre del servidor en el que se instalará Joomla!.
- Escribir el nombre de usuario de MySQL, la contraseña y el nombre de la base de datos que utilizará para Joomla!. Estos deben estar previamente configurados en su servidor.
- Configuración avanzada.

Si la base de datos contiene tablas de instalaciones anteriores de Joomla!, deberá indicar cómo proceder. Eliminarlas o hacer una copia de seguridad.

Escriba un prefijo con el que comenzarán los nombres de todas las tablas de esta instalación de Joomla!.

Configuración básica

Tipo de base de datos: **mysql** (Normalmente será MySQL)

Nombre del servidor: **localhost** (Normalmente localhost o un nombre de host provisto por su proveedor)

Nombre de usuario: **prueba** (Puede ser algo como root o un nombre de usuario, para la base de datos, asignado por su proveedor)

Contraseña: ********* (Por razones de seguridad el uso de una contraseña para la cuenta de la base de datos MySQL es altamente recomendado. Esta es la contraseña para acceder a su base de datos. Es posible que esta información sea predeterminada por su proveedor)

Nombre de la base de datos: **prueba_joomla** (Algunos hosts permiten solo una base de datos por cuenta. Si debe notarse más de un sitio Joomla! con una misma base de datos, puede modificar la opción de prefijo de tabla en la sección **Parámetros avanzados** para cada sitio instalado)

Configuración avanzada

Eliminar las tablas existentes

Hacer una copia de las tablas existentes.

Prefijo de las tablas: **JOS_** (No puede utilizarse "0a1_" ya que es una para la copia de seguridad de las tablas)

Joomla! es software libre distribuido bajo la licencia GNU/GPL.
Pack creado por Joomla! Spanish 2009 - Patrocinado por Web Empresa

Figura 6: Configuración de la base de datos y paso No. 4.

La configuración del FTP es un extra que va a depender de nuestras necesidades. En nuestro caso no se configuro.

Joomla! Instalación 1.5.14

Pasos

- Idioma
- Comprobación previa
- Licencia
- Base de datos
- Configuración de FTP**
- Configuración
- Finalizar

Configuración de FTP

Configuración de FTP:

Debido a posibles restricciones en los permisos en el sistema de archivos y a las restricciones de modo seguro (Safe Mode) de PHP en algunos servidores, para que todos los usuarios puedan utilizar los instaladores de Joomla! existe una capa FTP que gestiona las operaciones con los archivos.

Escriba el nombre de usuario FTP y la contraseña correspondiente, con acceso al directorio raíz de su sitio web Joomla!; esta será la cuenta FTP que gestionará las operaciones en el sistema de archivos cuando Joomla! requiera un acceso FTP para completar una tarea.

Por razones de seguridad, es mejor utilizar una cuenta de usuario FTP diferente de la cuenta principal, con acceso únicamente a la instalación de Joomla!.

Configuración básica

Sí

No

Habilitar la capa FTP para la gestión de archivos.

Nombre del usuario FTP:

Contraseña de FTP:

Ruta a la carpeta de Joomla! desde la raíz FTP:

Configuración avanzada

Servidor FTP: **127.0.0.1**

Puerto FTP: **21**

Guardar la contraseña FTP: Sí No

Joomla! es software libre distribuido bajo la licencia GNU/GPL.
Pack creado por Joomla! Spanish 2009 - Patrocinado por Web Empresa

Figura 7: Paso No. 5, configuración del servicio de FTP.

Se procede a la configuración del nombre de nuestro sitio y del correo electrónico del administrador. Nos aseguramos de hacer una instalación limpia, y no necesitaremos de “Instalar los datos de ejemplo predeterminados”.

Figura 8: Configuración principal del portal y paso No. 6.

Con esto hemos terminado la instalación de nuestro servidor Joomla con LAMP, para seguir con la siguiente ventana es necesario modificar los permisos del archivo *configuration.php* y eliminar el archivo *installation*:

```
# rm -R /var/www/Joomla/installation  
  
# chmod 644 /var/www/Joomla/configuration.php
```

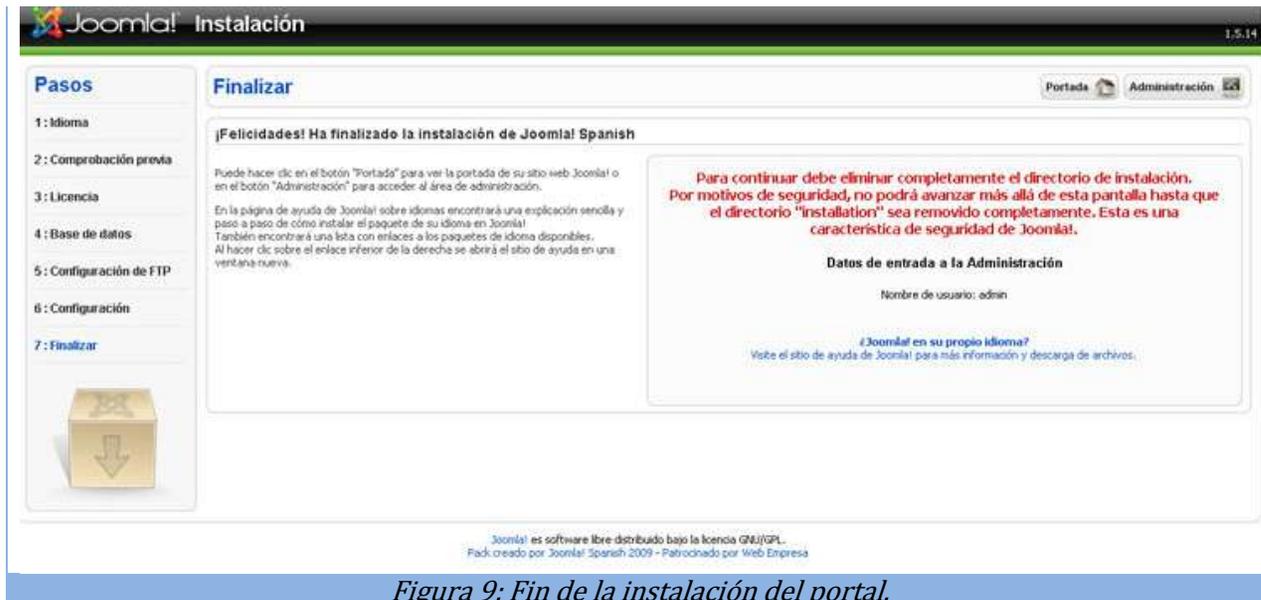


Figura 9: Fin de la instalación del portal.

Iniciando Joomla!

Elementos básicos de Joomla!

Plantillas

La plantilla (template) y sus archivos asociados proporcionan el 'aspecto visual y el manejo' del sitio web y se mantienen separados del contenido del sitio. Esta se almacena en una base de datos MySQL.

Los archivos de la plantilla se ubican en la carpeta 'templates' en la carpeta de instalación de Joomla!.

Componentes

Los Componentes son elementos del núcleo de Joomla! con una funcionalidad determinada y que se muestran en el cuerpo principal de la plantilla del sitio web. Dependiendo del diseño de la plantilla utilizada, suelen estar en el centro de la página web.

Módulos

Los Módulos amplían las posibilidades de Joomla! proporcionando nueva funcionalidad al software. Un Módulo es un pequeño artículo de contenido que puede mostrarse en cualquier parte que la plantilla lo permita. Los módulos son muy fáciles de instalar en el Administrador (Back-end).

Plugins

Los Plugins son extensiones que realizan dentro de Joomla! una amplia variedad de funciones relacionadas fundamentalmente con la autenticación de usuarios, el funcionamiento del buscador interno o con la edición de contenidos

Usuarios de Joomla!

Tipos de usuarios

Un Usuario de Joomla! tiene una serie de permisos predefinidos, es decir las tareas que puede realizar. Por eso, en el contexto de gestión de un sitio web, dependiendo del modo en que el propietario del sitio quiera delegar las responsabilidades, se definirán unos u otros tipos de usuarios. Joomla!.

- Invitados.
- Usuarios Registrados.

Los Invitados son sencillamente usuarios de Joomla! que han navegado hasta encontrar su sitio web. Dependiendo de cómo el administrador ha configurado el sitio, los invitados podrán navegar libremente por todo el contenido o tener restringido el acceso a cierto tipo de contenidos, reservados para usuarios registrados.

Los Usuarios Registrados están registrados en su sitio con un nombre de usuario y contraseña.

Este nombre de usuario y contraseña les permite acceder al área restringida del sitio, recibiendo privilegios especiales no disponibles para los invitados. Los usuarios registrados se dividen en dos grupos:

- Usuarios del Sitio (Front-end).
- Usuarios del Administrador (Back-end).

Por ejemplo, un propietario puede tener a una única persona administrando todos los aspectos del sitio web. En este caso se crearía un usuario Súper-Administrador.

En otro caso, un propietario puede querer tener el control sobre la publicación del Contenido creado por otro y que las tareas de Administración sean responsabilidad de una persona diferente. En este caso, se crearía una cuenta de Autor para la creación de contenidos, una cuenta de Supervisor para la comprobación y publicación de los contenidos y una cuenta de Súper-Administrador para mantener el diseño de la Plantilla, la configuración de Componentes y Módulos del sitio.

Usuarios del Sitio (Front-end)

Los usuarios del Sitio (Front-end) disfrutan de ciertos derechos adicionales sobre los visitantes, entre los que se puede incluir la capacidad para crear y publicar contenido en el sitio web. Generalmente, nos referimos a estos usuarios como proveedores de contenido ya que su meta principal es la de proveer contenido al sitio web, no la de administrar el sitio o alterar su diseño.

Dentro de esta amplia clasificación de proveedores de contenido, existen cuatro niveles específicos, que pueden ser asignados por el administrador del sitio. Estos niveles son: Registrado (*Registered*), Autor (*Autor*), Editor (*Editor*) y Supervisor (*Publisher*).

El nivel por defecto de los nuevos usuarios es el de Usuario Registrado. Para que los usuarios del Sitio (Frontend) puedan acceder a otro tipo de nivel, es necesario que un Administrador o un Súper-Administrador cambie su perfil mediante el Panel de Administración (Backend).

Usuarios del Administrador (Back-end)

Los usuarios del Back-end: Mánager, Administrador y Súper-Administrador, habitualmente se conocen como Administradores del Sitio, pero también tienen acceso a la interfaz del Front-end. Como los usuarios del Front-end, los usuarios del Back-end tienen diferentes privilegios. El único usuario que existe después de una instalación de Joomla! es el Súper-Administrador. Esta es la cuenta 'admin' creada durante el proceso de instalación.

Registro

Además del Súper-Administrador (admin) creado por defecto en la instalación de Joomla!, existen dos formas para que los invitados pueden registrarse como miembros de un sitio web Joomla!:

- Pueden registrarse por sí mismos utilizando el enlace 'registro' del formulario de acceso (si está disponible).
- Un Administrador o un Súper-Administrador puede añadirlos directamente usando el Panel del Administrador (Back-end).

Funciones de Gestión desde el Front-end

Gestión del Contenido

En general, la gestión de contenido incluye las siguientes tres operaciones:

1. Envío de nuevo contenido al sistema (Artículo de Contenido).
2. Corrección y edición de ese contenido si es necesario.
3. Publicación del contenido.

En el esquema de gestión de contenidos de Joomla!, los Autores generalmente son los responsables de enviar el contenido, los Editores son los responsables de editar el contenido y los Supervisores son los responsables de publicar el contenido. No obstante, tanto los Editores como los Supervisores pueden enviar contenido y un Supervisor también puede editar. Para el objeto de este capítulo, asumiremos que los Autores son los creadores del contenido, los Editores los que hacen la edición y los Supervisores los que publican el contenido.

Funciones de Gestión desde el Back-end

Acceso a la administración

Para acceder a las funciones de la Administración (Back-end) introduzca la siguiente URL: <http://www.sudominio.com/administrator>. Aparecerá un el formulario de acceso. Introduzca su Nombre de Usuario y su Contraseña en los respectivos campos y pulse el botón *Acceder* para Entrar.

Acceso a la administración de Joomla!

Usa un nombre de usuario y contraseña válido para poder tener acceso a la administración.

[Regresar a la página de inicio](#)

Nombre de usuario

Contraseña

Idioma

Figura 10: Autenticación como usuario registrado.

Se mostrará el Panel de Control (Página de Inicio de la Administración). El panel de control está compuesto por:

- Opciones de Menú



Figura 11: Opciones del menú Administrador.

- Iconos de acceso rápido



Figura 12: Panel de opciones rápidas.

- Módulos del Administrador

▶ Usuarios identificados					
#	Nombre	Grupo	Cliente	Última vez Activo	Cerrar Sesión
1	admin	Super Administrator	administrator	0.0 horas	
▶ Popular					
Artículos populares			Creado	Impresiones	
▶ Artículos añadidos recientemente					
Últimos items			Creado	Autor	
Sin resultados					
▶ Menú de estadísticas					
Menú					# Ítems
mainmenu					1

Figura 13: Opciones de monitoreo.

Gestión de archivos Multimedia

Para poder gestionar los archivos multimedia (imágenes, vídeos y otro tipo de documentos) hay que tener privilegios de al menos de manager. Para acceder a la pantalla de gestión hay que pulsar

en el menú Sitio > Gestión de Archivos Multimedia o bien pulsar el icono de acceso rápido “Gestor Multimedia”.



Figura 14: Gestor Multimedia

Jerarquía de Contenido en Joomla!

La estructura del contenido en un sitio web Joomla! está organizada en un orden jerárquico en base a Secciones, Categorías y Artículos de Contenido. En la práctica, una Sección puede tener una o más Categorías y una Categoría puede tener uno o más Artículos de Contenido.

Requisitos del Sistema

Antes de descargar el software Joomla!, debe asegurarse que su servidor web, o su servicio de hosting, cumple los requisitos mínimos para utilizar Joomla!. Son los siguientes:

- PHP 4.2.x o superior - <http://www.php.net>
- MySQL 3.23.x o superior - <http://www.mysql.com>
- Apache 1.13.19 o superior - <http://www.apache.org>

Además debe comprobar que el módulo PHP tenga instalado el soporte para MySQL, XML y Zlib. Joomla! puede utilizarse con los principales navegadores web, incluyendo: Firefox, Internet Explorer (versión 5.5+) y Netscape. Estos navegadores se aprovechan de la interfaz administrativa de Joomla!.

Diseño y desarrollo de la nueva plantilla para el portal Galileo.

El diseño de la plantilla se asemejará lo más posible a una plantilla institucional, y para ejemplo, se decidió seguir como modelo el portal del **IMTA**.

Como primer elemento y prácticamente el principal, se agregó el logotipo actual del Portal de Galileo.



Figura 15: Banner principal del prototipo de la portal de Galileo.

Teniendo este elemento integrado, se agregaron los menús superior y principal, respectivamente, quedando de la siguiente manera.



Figura 16: Resultado final del Banner con los menús superior e inferior.

Siguiendo los elementos de la plantilla, agregamos el pie de página de “vivir mejor”.



Figura 17: Pie de página de “vivir mejor”.

Una vez hecho eso, se agregó texto al pie de página.



Figura 18: Pie de página final con leyenda del domicilio del IMTA.

Seguridad

Usar las “*disable_functions*” para prevenir el uso de algunas funciones PHP peligrosas. Por ejemplo:

```
disable_fuctions = show_source, exec, phpinfo
```

Usar “*open_basedir*”: Esto limitará qué archivos PHP pueden abrirse y cuáles no. Veamos un ejemplo:

```
open_basedir = /home/webguy/www/html
```

Deshabilitar “*register_globals*”: Joomla! te avisará cuando esta función no esté activada. La usaremos del siguiente modo:

```
register_globals = 0
```

Deshabilita "*allow_url_fopen*": Esta función se utiliza cuando creamos PHP wrappers para abrir URLs remotas. Para configurarlo tendremos que añadir:

```
allow_url_fopen = 0
```

Algunas consideraciones generales serian

- Cambie sus passwords regularmente y no use siempre los mismos. Utilice una combinación aleatoria de letras, números, o símbolos y evite usar nombres o palabras que puedan ser encontradas en un diccionario. Nunca utilice los nombres de sus parientes, mascotas, etc.
- Nunca dependa de los *backups* de otro. Hágase responsable personalmente de respaldar regularmente los archivos de su sitio y su base de datos.

HTTP Server (Apache, etc.)

- PHP, MySQL y muchos otros componentes base fueron originalmente diseñados para, y generalmente funcionan mejor en, servidores Apache. Evite usar otros servidores si es posible.
- * Utilice archivos *.htaccess* para bloquear intentos de *exploits*.
- Regularmente revise los registros de acceso en busca de actividad sospechosa. No confié en sumarios y gráficas. Revise los "*raw logs*" (registros en crudo) para detalles más reales.
- Configure los filtros de Apache *mod_security* y *mod_rewrite* para que bloqueen ataques PHP.

MySQL

- Asegúrese de que la cuenta MySQL de Joomla! está configurada con acceso limitado. Este consciente de que la instalación inicial de MySQL es insegura. Una cuidadosa configuración manual es requerida luego de la instalación.

PHP

- Antes que nada PHP 4 ya no es mantenido activamente, actualice su código PHP a PHP 5.
- Aplique todos los parches necesarios para PHP y para aplicaciones basadas en PHP.
- Se recomienda un frecuente escaneo en ámbitos donde un gran número de aplicaciones PHP están en uso.

Extensiones (Componentes, Módulos, y Bots) de Joomla!

- Remueva todas las extensiones Joomla! que requieran *register_globals* ON.
- Descargue extensiones solo de sitios de confianza. La definición oficial de "**sitio de confianza**" es aquel sitio en el que **USTED** confía.

- Antes de instalar extensiones de 3eros (*third party extensions*), revise: Lista oficial de extensiones de 3eros vulnerables!!!:
<http://forum.joomla.org/index.php/topic,79477.0.html>.
- Tenga cuidado! Las extensiones de 3eros vienen en todos los sabores, tamaños y antigüedad. Aunque existen los estándares de código Joomla! (*coding standards*), la extensiones listadas en el sitio oficial de Joomla! no son revisadas para ver si cumplen dichos estándares. Prueba todas las extensiones en un sitio de desarrollo antes de instalarlas en un sitio "real" sitio en producción.
- Respalde su sitio y la base de datos del mismo antes de instalar nuevas extensiones.
- Revise regularmente:
Cuestiones de seguridad de 3eros/No Joomla!:
<http://forum.joomla.org/index.php/board,296.0.html>
- Remueva cualquier extensión no usada, y revise doblemente que los directorios y archivos relacionados hayan sido borrados.

12 ANEXO 4: Colección y Análisis de Información del Uso del Sitio a lo largo del año.

A lo largo del año se monitoreó la actividad del sitio Web del SMN, ahora en su nueva versión y se analizaron estos resultados. En la siguiente figura se muestra el resumen de ese monitoreo.

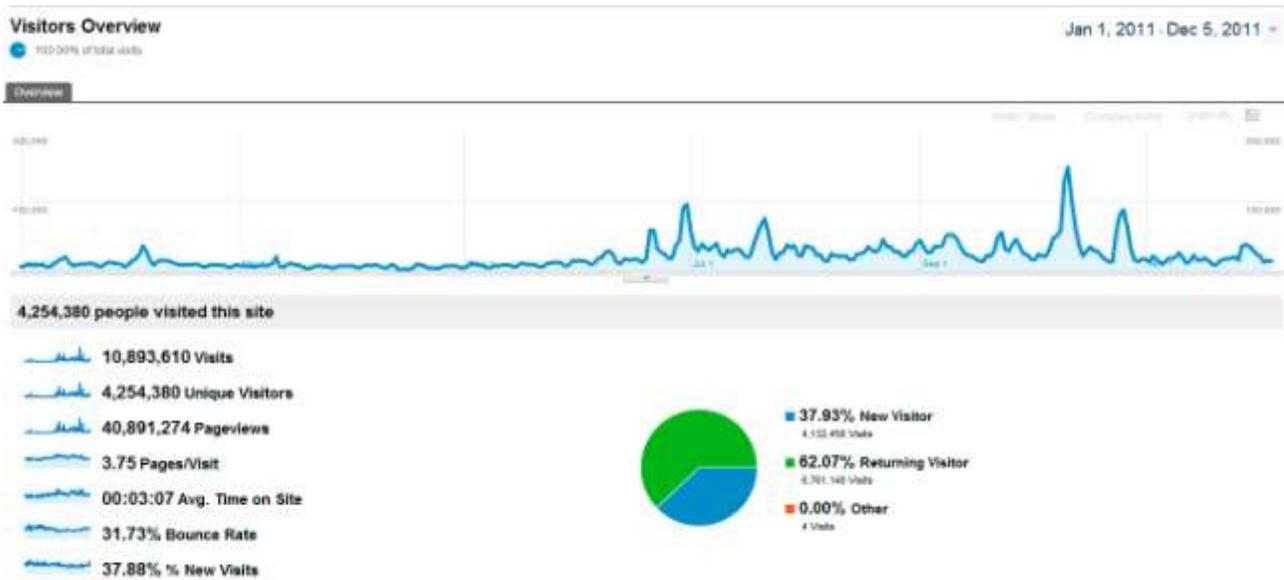


Figura 1. Resumen y Análisis de la actividad del Sitio *Web* del SMN