Guía técnica para la elaboración de sitios web PUCP

- Preliminar -

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Dirección de Comunicación Institucional

2017

Introducción

El siguiente documento es una guía técnica dirigida a diseñadores y programadores que actualmente se encuentren elaborando un sitio web de la Universidad. Esta guía, se pone a disposición con el objetivo de mejorar la accesibilidad y seguridad de la información publicado en todo sitio web que sea publicado en los servidores de la Universidad.

Le agradeceremos leer con atención este documento y seguir las indicaciones cuidadosamente. Así estará aportando a mejorar los servicios de comunicación de la Universidad.

Si tuviera alguna consulta acerca de la actualización de este documento o algún comentario al respecto, escríbanos a través del Portal de Comunicación Institucional o llámenos al 6262000 anexo 3911.

1. ¿Qué administrador de contenidos (CMS) debo usar?

La tendencia que existe actualmente de usar este tipo de administrador de contenidos (Open Source Drupal y Wordpress) responde a la calidad del servicio y soporte brindado por la comunidad de desarrolladores en el ámbito internacional. Actualmente estos dos gestores de contenidos han sido premiados con el Packt Open Source Awards y poseen soporte permanente en caso se identifique un problema que comprometa la seguridad de los contenidos del usuario.

Por tales motivos, se recomienda el uso de los dos gestores de contenidos mencionados siempre que se tomen en cuenta los siguientes lineamientos:

- El proveedor debe evaluar qué CMS va a utilizar dependiendo de la envergadura y complejidad del sitio web. Se recomienda usar Wordpress para páginas web de eventos y unidades que no representen grandes lectura de información, y Drupal para proyectos mucho más grandes que requieran un sistema de almacenamiento y relación de datos más complejos. Es importante indicar que estos dos CMS son los únicos que brindan soporte para las aplicaciones de integración de la Universidad.

- Toda página web diseñada en cualquiera de estos dos CMS se deberá desarrollar sin alterar el núcleo del sistema (*core*). Por ningún motivo deberá ser vulnerado ("roto") ya que esto acarreará problemas de actualización de versiones y podría comprometer la seguridad del sitio web.

El objetivo es que el proveedor de la PUCP se especialice en el CMS que se decida utilizar de tal forma que el dominio de la herramienta permita crear componentes particulares para cada sistema, que se basen en los propios productos, recursos y servicios de la unidad. Asimismo, se espera que permita concebir diseños flexibles que puedan ser actualizados en próximas versiones sin necesidad de una reestructuración completa.

2. Consideraciones respecto a la elaboración del *template*

Wordpress

- El *template* debe ser creado a medida, no se debe editar las plantillas que vienen por defecto para Wordpress.
- Colocar todas las funciones en el archivo **function.php** del *template*
- Usar el archivo **404.php** para personalizar la página **Error 404**
- Desactivar las revisiones de las páginas
- No dejar espacios en blanco después del *tag* de cierre de **php**
- Procurar usar el editor de texto enriquecido que viene por defecto en Wordpress y no *plugins* que cambien los botones del editor
- Usar **style-editor.css** para personalizar los estilos del editor de texto enriquecido.
- Funciones de PHP para ejecución del sistemas, están deshabilitados en el servidor, funciones como: *exec*(), *system*(), *passthru*(), etc.
- Funciones de PHP de lectura externa de URL, están deshabilitados en el servidor, funciones como *file_get_contents*() y *file_put_contents*()

Drupal

- No modificar los **themes core** de Drupal
- Debe usarse el *theme* **Zen versión 6.x-2.0**
- El *theme* debe instalarse en la **ruta sites/nombre_del_sitio/themes**, donde **nombre_del_sitio** es la carpeta del sitio web que se está desarrollando.
- No modificar los módulos a utilizar. Si se desea hacerlo, crear un módulo auxiliar que modifique el original.
- Los módulos a usar se instalarán en la ruta **sites/all/modules**
- Usar siempre la última versión del módulo a utilizar
- Documentar la relación de módulos extras a utilizar indicando sus respectivas versiones

3. Consideraciones respecto al maquetado e integración de la plantilla

- La plantilla debe estar en formato *XHTML 1.0 Transitional o HTML5*
- Usar framework *Bootstrap* para uso del sitio en dispositivos móviles (*responsive)*
- No usar condicionales html para cada tipo de navegador
- En caso de usar *jQuery*, utilizar el que viene por defecto con *Wordpress* o enlazar el jquery.js invocándolo desde el repositorio de Google.
- En caso de usar *Google Fonts*, invocarlas desde el repositorio de Google Fonts.
- En el html no usar rutas absolutas escritas directamente en el código ni rutas relativas simples. Se debe procurar usar la función *bloginfo()* del *Wordpress* para obtener la ruta de la carpeta del template.
- Las ID y *Class* para estilos personalizados no pueden ir en el contenido editable, solo en los contenedores. Por ese motivo, se debe procurar usar estilos anidados.
- Es necesario validar el código html de los errores críticos (XHTML y CSS), como mínimo: HTTP://validator.w3.org, http://jigsaw.w3.org/css-validator/
- Las imágenes que formen parte del portal deben ser optimizadas con la finalidad de disminuir el tiempo de carga por página.

4. Consideraciones que debe tener la administración respecto al armado del sitio

- No colocar todos los tipos de páginas como *posts*, sino crear *Custom Post Type* por cada tipo diferente de página, por ejemplo: eventos, boletines, cursos, profesores, etc.
- No exceder el uso de las categorías para asignar propiedades a las páginas creadas o filtradas, sino crear taxonomías personalizadas
- Procurar usar los *CustomFields* de Wordpress para campos personalizados en cada página
- Usar la funcionalidad Menú de la administración para crear menús administrables

5. Consideraciones respecto al uso de componentes y *plugins*

- Es necesario verificar la confiabilidad del sitio desde donde se descargan los plugins necesarios, por tanto se debe utilizar los que han sido validados por la Universidad y que

se muestran en la siguiente lista http://test.pucp.edu.pe/plugins/

- Toda página web debe tener asociada, a la sección Contáctenos, un formulario que registre información bajo los siguientes campos:

    - Nombres y apellidos
    - Correo electrónico
    - Asunto
    - Mensaje
    - Código **captcha**

    El *plugin* que se deberá usar es el **contact-form-7,** que permitirá habilitar un formulario personalizado según las necesidades de cada unidad.

- Evitar el uso de plugins que permitan el uso de php en el editor de contenidos, en widgets o en campos personalizados
- Evitar el uso de plugins para compartir en redes sociales. En ese caso, se recomienda usar botones propios de cada red social
- Evitar el uso plugins para guardar cache de las páginas internas
- Para compartir por correo, solo usar el API del pluginAddthis, no instalar el plugin
- Listado de plugins válidos para diversas funcionalidades que requiera la página:

    - **advanced-custom-fields**: para crear campos personalizados o **CustomFields**
    - **cms-tree-page-view**: vista mejorada de listado de páginas en formato de árbol
    - **contact-form-7**: para crear formularios personalizados y administrables
    - **custom-post-type-ui**: sirven para administrar los **Custom Post Type**
    - **really-simple-captcha**: para crear **captcha** para formularios
    - **redirection**: para crear redirecciones personalizadas
    - **tinymce-advanced**: usar solo en caso se requiera el manejo de tablas con el editor de contenido
    - **wordpress-seo**: para mejorar el SEO de la página y crear mapa de sitios
    - **wp-cumulus**: para crear nubes de etiquetas animadas
    - **wp-paginate**: para crear una paginación personalizada
    - **wp-print**: para crear versiones para imprimir cada página

    Para ver la lista completa de *plugins* validados por la Universidad, ingrese al siguiente enlace: http://test.pucp.edu.pe/plugins/.

6. Consideraciones respecto a la seguridad del sitio

Este apartado busca brindar las recomendaciones o consideraciones más importantes que todo proveedor debe tener presente ante cualquier desarrollo de aplicaciones web. Teniendo en cuenta que todo sitio web será desarrollado en un CMS Open Source, es recomendable identificar si ya fue cubierto el tema de la seguridad con algún *plugin*. Si no se dispone del componente necesario, es recomendable que el proveedor desarrolle uno específico con el fin de cubrir este requerimiento y que sea instalado sin alterar el *core* del CMS utilizado.

6.1 Validación de entradas y salidas de información: el proceso de entrada y salida de información es el principal mecanismo del que dispone un atacante para enviar o recibir código malicioso contra el sistema. Por tanto, siempre debe verificarse que cualquier dato entrante o saliente es el apropiado y debe estar en el formato que se espera. Las características de estos datos deben estar predefinidas y deben verificarse en todas las ocasiones.

Algunas recomendaciones para la **validación de las entradas**:

- Tener mecanismos de validación de datos
- Validar todas las entradas que pueden ser modificadas por un usuario malicioso: cabeceras HTTP, input fields, hidden fields, drop down lists, etc.
- Comprobar las longitudes de todas las entradas
- Validar todos los campos, cookies, http headers/bodies y form fields
- Formatear los datos convenientemente y asegurarse de que solo contengan caracteres conocidos como buenos
- Validar los datos en el servidor
- Asegurarse de que no hay "puertas traseras" en el modelo de validación
- Validar la extensión, contenidos y tamaños de los elementos a publicar de acuerdo con las limitaciones del servidor web, ya sea imágenes, documentos, videos, audios, etc.

Importante: cualquier entrada externa, sea cual sea, debe ser examinada y validada.

Algunas recomendaciones para la **validación de las salidas**:

En primer lugar, se debe ofrecer la mínima información ante una situación de error o una validación negativa. Por este motivo, los mecanismos de seguridad deben diseñarse de tal manera que faciliten la mínima información posible y para que, cuando se haya denegado una operación, cualquier otra sea igualmente denegada.

6.2 Gestión de errores / fuga de información: durante el proceso de desarrollo de una aplicación, el reporte de errores puede llegar a ser muy útil; pero una vez que se pasa la aplicación al entorno de producción, es necesario esconder todo tipo de reporte, ya que podría facilitar el trabajo a posibles atacantes.

Si deseamos evitar que se muestren los mensajes de error predefinidos, podemos modificar el reporte de errores en el *script* de **PHP** mediante la función **error_reporting(0)**

Algunas recomendaciones para la gestión de errores o fuga de información:

- Asegurarse de que todas las llamadas a métodos o funciones que devuelven un valor tienen su control de errores. Asimismo que sea posible comprobar el valor devuelto.
- Asegurarse de gestionar adecuadamente las excepciones y los errores en la aplicación (dependiendo de la aplicación se debe activar un *Preloading* que indique al usuario que la aplicación está cargando).

- Asegurarse de que el usuario no perciba errores del sistema
- Asegurarse de que la aplicación falla de un modo seguro
- Asegurarse de liberar los recursos en caso de error.
- Controlar el detalle de errores a través de un log protegido y no en la misma pantalla de navegación
- Personalizar los errores de páginas no encontradas *(404 Not Found).* En esos casos, se recomienda utilizar el siguiente mensaje:

> **Página no encontrada**
>
> ¿Busca algo en la Pontificia Universidad Católica del Perú?
>
> La página solicitada puede no estar disponible, haber cambiado de dirección (URL) o no existir. Disculpe por las molestias. Con frecuencia es debido a algún error al escribir la dirección de la página (URL). Compruébela de nuevo para ver si es correcta.
>
> Gracias por visitar la Pontificia Universidad Católica del Perú.

6.3 Protección contra vulnerabilidades: verificar que el sitio web no es vulnerable a los siguientes ataques comunes:

- SQL Injection: es una técnica que se utiliza para introducir o modificar las llamadas a una base de datos, para ello se aprovecha de la falta de validación de las variables.

- HTML Injection: al igual que ocurre con el ataque anterior, se puede utilizar la inyección de código HTML.

- Cross-Site-Scripting (XSS): es una técnica que se utiliza para introducir código HTML o Java Script a través de formularios web.

6.4 Contar con mecanismos de autenticación

- Asegurarse de que todas las peticiones pasan por un formulario de autenticación y que este no se puede omitir (tener cuidado con las URL de acceso directo, por ese motivo es importante la autenticación).

- Verificar que todas las páginas cumplen el requisito de autenticación

- Asegurarse de que siempre que se pasen credenciales de autenticación (o cualquier información sensible) solo se aceptará la información vía HTTP POST y nunca con GET.

- Cualquier página para la que se descarte el mecanismo de autenticación debe ser revisada para asegurarse de que no tiene brechas de seguridad.

- Asegurarse de que las credenciales de autenticación no van en texto plano.

- Verificar que no hay “puertas traseras” en el código en producción

6.5 Estructura administrable e independiente: consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. El sistema debe permitir manejar el contenido y las plantillas de manera independiente, además de agregar editores y designar roles diferenciados en la administración del sitio.

6.6 Privilegios de archivos y carpetas: es necesario identificar las carpetas donde el sitio web realizará "subidas de archivos" para que estas reciban un trato diferente al momento de la publicación del sitio web, ya que será necesario brindar permisos sobre dichas carpetas además de registrarlas en el servidor donde será alojada la página web.

7 Consideraciones sobre las versiones de aplicativos en los servidores

Para que los desarrollos web elaborados por terceros sean compatibles con las herramientas de desarrollo utilizadas en la Universidad, se debe tener en cuenta las siguientes características:

- Manejador de base de datos: Mysql-server 5.5.x
- Lenguaje de programación de aplicaciones web: PHP 5.3.x
- Sistemas manejadores de contenidos (CMS – últimas versiones)
  Wordpress 4.x.x en español. Multisitio para los que cuentan con un dominio asociado a uno genérico, por ejemplo http://red.pucp.edu.pe/ridei
- Validación con el módulo de seguridad Mod-Security

Para consultar acerca de las versiones actualmente vigentes de estos servicios, visite la sección de preguntas frecuentes del Portal de Comunicación Institucional.

8 Consideraciones sobre uso del logotipo institucional

Con el fin de proyectar una imagen sólida y consistente de la Universidad es necesario tomar en consideración la guía que la misma establece como estándar visual para el uso del logotipo institucional de la Pontificia Universidad Católica del Perú en medios electrónicos.

Por tal motivo, la unidad debe descargar la última versión actualizada del **Manual de uso del logotipo institucional en productos digitales** para que se tomen en consideración todas las indicaciones allí establecidas. Este documento puede ser descargado del Portal de Comunicación Institucional.

9 Consideraciones sobre módulos institucionales

Todas las páginas web de unidades académicas deben contar con los módulos institucionales:

Módulos Agenda y *PuntoEdu* en pie de página

Estos módulos deben ser incluidos y presentados dentro de las propuestas de diseño que se envían a las unidades PUCP considerándolos como estándares institucionales que rigen para todo sitio web que pertenece al dominio **pucp.edu.pe**

La unidad debe solicitar con anticipación a la Dirección de Comunicación Institucional (DCI) el listado de noticias, eventos o videos que aplican a la unidad en mención.

- Noticias de la unidad en *PuntoEdu*
- Eventos de la unidad en AgendaPUCP
- Lista de producción de videos (en caso la unidad cuente con una)

Redes sociales PUCP (íconos)
Facebook PUCP
http://www.facebook.com/pucp

Twitter PUCP
http://twitter.com/pucp/

YouTube
http://www.youtube.com/pucp

Servicios PUCP (enlaces)

Página principal PUCP
http://pucp.edu.pe/

PuntoEdu
http://puntoedu.pucp.edu.pe/

Intranet y correo
http://intranet.pucp.edu.pe/directorio-servicios.html

Campus Virtual
http://campusvirtual.pucp.edu.pe/

Biblioteca
http://biblioteca.pucp.edu.pe/

Firma o cierre de página

Las firmas de todas las páginas deberán guardar este formato con la salvedad del teléfono y

anexo, que son los elementos que se diferencian por cada unidad, así como el año, que debe ser editado desde la administración:

© [año] Pontificia Universidad Católica del Perú - Todos los derechos reservados
Av. Universitaria N°1801, San Miguel, Lima 32 - Perú | Teléfono: (511) 626-2000, [anexo]

10 Consideraciones sobre el seguimiento del sitio

Es indispensable que todo proveedor tenga en cuenta las siguientes consideraciones respecto al uso de la analítica y códigos de seguimiento en la página web que está desarrollando:

- Todas las analíticas estarán asociadas a las cuentas institucionales creadas en Google Analytics. Esta asignación la realiza la DCI al momento de la publicación de la página web.

- Dentro de los apartados de una web es necesario que las siguientes secciones cuenten con código de seguimiento que permita registrar el dato de descarga.

  Los datos resaltados son los elementos diferenciadores por cada sitio.

  Zona de descargas

  descarga aquí

  Donde:
  NombreDeDocumento =”Silabo-curso-Ciudadania-Intercultural.pdf”.

  Formulario de inscripción o contacto

  _gaq.push(['_trackEvent', 'inscripcion','nombrePrograma", 'aqui'])

  <input name="RegSolicitud" type="button" value="RegSolicitud" onclick="validar_formulario(document.forminscripcion); _gaq.push(['_trackEvent', 'inscripcion','nombrePrograma','aqui']);"/>

  Donde:
  Inscripcion =”Ficha de inscripcion”.
  NombrePrograma=” Primeras Jornadas de Derecho de Aguas”

  Nota: los nombres deberán ir sin tildes.

11 Lineamientos de accesibilidad web (según http://www.w3.org/TR/WCAG/)

- El tamaño del cuerpo de texto no debe ser menor de 12 píxeles. Se puede usar 11 píxeles solo para información complementaria, como barras laterales, los textos de fechas y la

sección Ver más.

- Se debe procurar que los colores tengan contraste entre sí para que todo el contenido sea legible en monitores de baja resolución y por usuarios que no puedan percibir diferencias de colores.

- La definición del color de los enlaces debe estar estandarizada en el contenido de todo el sitio. Los enlaces en títulos, cuerpo de texto y barras laterales o módulos deben comportarse de manera consistente en todas las vistas de la web.

- El usuario debe poder identificar con facilidad la navegación principal del sitio (menú principal) en relación con la secundaria.

12 Consideraciones para las diferentes etapas del proyecto

El proveedor deberá tener en cuenta que desde la definición de los términos del contrato y durante todo el proceso de desarrollo del sitio web, hasta su puesta en marcha, se realizarán diferentes etapas de revisión con el fin de corroborar que el trabajo está siguiendo correctamente las especificaciones que establece la Universidad:

Revisiones sobre la implementación

- **Al terminar el 100% del desarrollo del sitio web**

  La unidad debe enviar:

  - Ruta (**URL**) del desarrollo final del sitio web
  - Código fuente de todo el desarrollo
  - Lista de todos los *plugins* y/o componentes utilizados
  - *Template*
  - Estructura de la base de datos en formato **Mysql**
  - Datos de acceso como administrador del sitio web

Todos los entregables considerados para la revisión deberán ser enviados por la unidad a través de la generación de una solicitud de comunicación a través del Portal de Comunicación Institucional.

Una vez que el sitio web haya pasado por la validación técnica, la DCI elaborará un informe donde detallará los ajustes encontrados en la aplicación; de lo contrario, esta será publicada en la ruta oficial previamente definida.

Es importante indicar que la Universidad realizará cuando considere necesario una nueva revisión de la aplicación con la finalidad de garantizar la correcta funcionalidad del mismo.