

Personalización de la presentación de Revistas Institucionales basadas en OJS 2.4.2

Felix Ivan Romero Rodríguez ^{1*}, Yelena Islen San Juan ²

¹ Centro de Geoinformática y Señales Digitales. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

² Instituto de Información Científica y Tecnológica (IDICT). Calle 18 A entre 41 y 47, Reparto Miramar, Playa, ciudad de La Habana, Cuba. Código postal: 11300

* Autor para correspondencia: firomero@uci.cu

Resumen

La divulgación del contenido científico es una tarea que se dan muchos profesionales consagrados a la ciencia con el fin de hacer conocer sus resultados. De manera generalizada el proceso de divulgar se realiza a través de conferencias, eventos y la publicación de artículos científicos. Una de las maneras de publicar los artículos científicos es a través de revistas institucionales, las cuales implementan procesos editoriales según su definición. Numerosas revistas implementan sus esquemas de arbitraje sobre aplicaciones web ya desarrolladas con este fin, tal es el caso del Open Journal System. El Open Journal System, es una aplicación con la finalidad de servir como enlace para publicar los resultados del conocimiento científico. Sin embargo cuando la revista es configurada y puesta en funcionamiento, es común que no se encuentre totalmente adaptada a la identidad de la institución, ni se encuentre bajo un diseño adaptable que permite visualizar correctamente los contenidos desde dispositivos diferentes de un ordenador. El presente trabajo traza como metas algunas prácticas para personalizar el OJS, teniendo en cuenta su presentación. Para ello se utiliza bibliotecas de presentación como jQuery, marco de trabajo Bootstrap y como lenguaje base PHP. Como resultado se muestran casos de éxito de revistas institucionales personalizadas.

Palabras clave: personalización, ojs, revistas, publicación

Introducción

El avance científico en determinada área, sólo es perceptible los que trabajan en ella y las entes que interactúan con los nuevos descubrimientos. Por ello una manera de divulgar los conocimientos adquiridos y el resultado de las investigaciones se aloja en las publicaciones científicas. Para alojar las publicaciones existen diversas herramientas, como las base de datos, bibliotecas, y las revistas científicas (Biblioteca Universidad de Alcalá 2014). La revista científica se expone como un punto de acceso a los resultados científicos en diversas áreas. Con el objetivo de mantener revistas institucionales, han surgido herramientas que siguen esta pauta tale es el caso del Open Journal System (Smecher 2008). Este sistema, gestiona los procesos de arbitrajes en la publicación de artículos científicos. Debido a las funcionalidades que posee es ampliamente utilizado. Sin embargo, instituciones no tienen en cuenta la manera en que se visualizan los datos en esta revista, ya que mantienen su configuración al mínimo. Siendo así que no aprovechan las nuevas técnicas como el ajuste de la identidad de las instituciones, la arquitectura de la información, la adaptabilidad según el dispositivo de acceso y todo ello influye negativo en la experiencia de usuario. El presente trabajo propone diferentes niveles de configuración de la interfaz de revistas institucionales basadas en OJS 2.4.2, con el objetivo de incidir de manera positiva en la experiencia de usuario que interactúan con usuarios de revistas institucionales desplegadas en este sistema.

Materiales y métodos o Metodología computacional

Modelo Vista Controlador

Los patrones arquitectónicos definen la estructura base de un sistema, así como la forma en que se comunica con otros patrones a distintos niveles. El modelo vista controlador, denominado MVC, es un patrón arquitectónico perteneciente al estilo de llamada y retorno. Describe tres capas lógicas independientes que se intercomunican, las cuales contienen la interacción con la persistencia de los datos, el caso del modelo. La presentación de los datos, la vista; y finalmente la comunicación entre las funcionalidades, que sería la responsabilidad del controlador. De manera general el patrón MVC recrea el escenario de cuando usuario accede a una funcionalidad el controlador captura dicha petición, realiza una llamada al modelo solicitando datos para luego enviar la información obtenida a la vista. El OJS implementa el MVC de manera similar a la definición de este modelo, donde el modelo es representado por la capa de abstracción de datos que presenta la manera en que se interactúa con los datos. La capa de abstracción es conocida como Data Access Object, con la responsabilidad de persistir cada objeto que se genera en el modelo.

De manera que cada clase del modelo tiene asociada una clase DAO, la cual se encarga de interactuar con la base de datos. Por ejemplo la clase que Artículo que se implementa en el fichero Article.inc.php tiene como manejador a la clase ArticleDAO que se implementa en el fichero Article.DAO.inc.php y es esta la responsable de interactuar entre el modelo y los registros en la base de datos (Smecher 2008). Para manejar las peticiones realizadas por los usuarios, se implementan los controladores. Estas clases son identificables porque en su nombre tiene el sufijo Handler, debido a que heredan de la clase Handler y a su vez hereda de la clase PKPHandler. Los controladores según su escenario reciben la petición, la procesan con los resultados del modelo en caso de que apliquen y realizan una llamada a la plantilla responsable de presentar los datos.

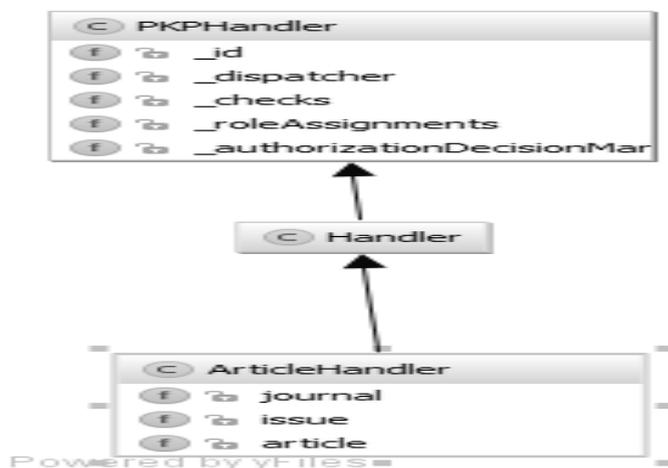


Figura. 1 Controladores en el OJS

Finalmente el manejador de plantillas invoca al motor de plantillas instalado en el OJS y le envía los datos a representar en las páginas generadas. Un motor de plantillas es un conjunto de clases que tienen como finalidad preparar en eterno donde se mostrarán los datos. Permite asegurar la escalabilidad de los sistemas y es utilizado de manera general para implementar la capa de la vista en sistemas web basados en el patrón arquitectónico MVC. Normalmente puede permitir herencia para definir bloques de información y obtener un esqueleto base de cómo se visualizarían los datos (Carlos Hernan Aguilar Hurtado 2014). OJS como motor de plantillas utiliza Smarty (Ohrt,

Zmievski, Ramírez 2010). Smarty es un motor de plantillas PHP, que se encarga de compilar las representaciones de plantillas escritas para visualizarlas a través de la creación de páginas PHP.

Luego una vez analizado el funcionamiento básico del OJS, su arquitectura quedaría representada como muestra la figura 1

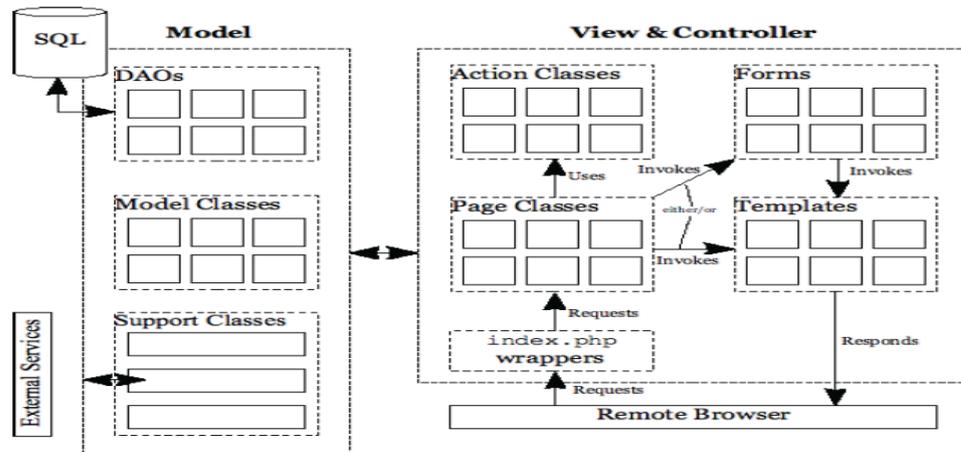


Figura. 2 MVC implementado OJS(Smecher 2008)

Presentación de Interfaces de Usuario Web.

Existen disímiles herramientas destinadas a la presentación de interfaces web en las aplicaciones. Ejemplo de ellas se ven mostrados en bibliotecas JavaScript y marcos de trabajo CSS, que facilitan la adecuación de identidad, manejo de la arquitectura de la información en las aplicaciones web, adaptando la experiencia del usuario. Tal es el caso de jQuery y jQuery UI (Ramos Velazco Adrianet 2014), son bibliotecas de clases que permiten la interacción con los elementos que describen de las páginas web. Apoyando estas herramientas se encuentra el concepto de diseño adaptable, lo que permite que la visualización de la web sea independiente del dispositivo, ya que enmarca numerosas reglas para ajustarse según la resolución o relación de aspecto objetivo. El diseño adaptable o responsive web design responde a un conjunto de prácticas que pretenden alcanzar este fin partiendo de estar centrado en la experiencia del usuario (Alexander Bernal 2011). Herramientas que logran este modelo de desarrollo de interfaces web, se encuentra Bootstrap, un marco de trabajo de JavaScript y CSS desarrollado por Twitter orientado al diseño adaptable de páginas y aplicaciones web (Alvaro Fontela Sanchez 2013). Conjuntamente con todo ello es necesario tener en cuenta la Arquitectura de la información, disciplina que orienta la usabilidad de las aplicaciones y sitios. Permite desarrollar basado en la experiencia del usuario las interfaces.

La Arquitectura de la información es necesaria tenerla en cuenta en el desarrollo de plataformas actuales sobre todo debido al crecimiento exponencial de la información que se genera en internet y la manera de acceder a estos contenidos generados (Gutiérrez, Santiago 2008).

Administración de OJS

Como parte del proceso de instalación y configuración de la plataforma existen opciones que permiten personalizar la presentación de la revista. Por el sistema de administración es posible cargar archivos de estilos propios, modificar las cabeceras de información, agregar imágenes a cada número de la tabla de contenido de las revistas.

Adicionar imágenes, opciones de visualización y de componentes, de manera que de forma básica el mismo proceso de configuración provee de funcionalidades mínimas con este fin como se muestra en la figura 3:



Figura. 3 Personalización básica OJS(Links, News, Blogs 2006)

Resultados y discusión

Personalizar el OJS, se puede realizar desde diferentes niveles, de manera que adaptar este sistema puede realizarse escaladamente, de manera que se convierte en un proceso estructurado. Primeramente en un primer nivel sería modificar las clases de estilo que define el OJS para personalizar sus páginas. Es recomendable no modificar los archivos CSS que vienen por defecto en la instalación del OJS, de manera que si se actualiza a versiones superiores éstos no se sobre escriban y se pierda la información editada. Para ello primeramente se crea un nuevo archivo CSS y ahí se redefinen las clases que propone por defecto el sistema. Por ejemplo para las tablas, en el archivo common.css existe la clase denominada:

```
table.tocArticle {
  width: 100%;
}
```

Modificar dicha clase sería agregar al final de la definición de estilos en el archivo lib/pkp/templates/common/header.tpl el nuevo archivo que contenga la definición de la clase con las nuevas propiedades ajustadas. De esta manera proceder con cualquier otra propiedad que se quiera ajustar.

```
<link rel="stylesheet" href="{baseUrl}/styles/extends.css" type="text/css" />
```

Un segundo nivel sería cómo hacer que la presentación de la revista responda a un diseño adaptativo, para ello una práctica es utilizando las propiedades mediaquery y screen de CSS, lo cual permite establecer diferentes valores a los elementos tomando como punto de partida la resolución y relación de aspecto de la pantalla en que se visualiza la información. Para ello se procede de la misma forma anterior, se crea un fichero donde se plasmen un espectro de esquemas de visualización y se van tratando los elementos según se requieran:

```
/* Desktops and laptops ----- */
@media only screen
and (min-width : 1224px) {
/* Styles */
```

```
.header-img {
height: 170px;
position: relative;
text-align: center;
text-shadow: 0 1px 3px rgba(0, 0, 0, 0.4), 0 0 30px rgba(0, 0, 0, 0.075);
width: 1280px;
}
```

```
}
```

Sin embargo, esta vía requiere un tener amplio conocimientos, sobre todo para la disposición de los elementos según el dispositivo en que se visualice. Como ventaja permite ser específico con cada uno de los dispositivos objetivos, dígame ordenadores, laptops, tabletas, móviles. Entonces, ¿cómo proceder de manera genérica a un diseño adaptable? Responder a esta pregunta sería utilizar marcos de trabajo que ya tengan en cuenta técnicas de diseño web adaptable como lo es Bootstrap. Para adicionar es necesario incluir los archivos del marco de trabajo en la carga de la página, ello se realiza en el fichero header.tpl, del cual se explicó anteriormente, es este fichero donde se deben realizar todas las inclusiones de fuentes externas.

```
<link rel="stylesheet" href="{baseUrl}/styles/bootstrap.css" type="text/css" />
<link rel="stylesheet" href="{baseUrl}/styles/bootstrap-responsive.css" type="text/css" />
```

```
<script type="text/javascript" src="{baseUrl}/js/jquery.js"></script>
<script type="text/javascript" src="{baseUrl}/js/jquery-ui.js"></script>
<script type="text/javascript" src="{baseUrl}/js/bootstrap.js"></script>
```

Es necesario señalar que conjuntamente con Bootstrap es necesario incluir el jQuery, ya que Bootstrap depende de dicha biblioteca de clases para funcionar. Una vez adicionado el marco de trabajo se procede a trabajar a modificar el cuerpo de la página para aprovechar las propiedades que provee este marco de trabajo CSS en ese sentido, dígame el modelo de cajas que propone, mediante el cual simula la página como toda una rejilla, adicionando filas y columnas a través de las clases row, span y sus variantes.

```
<div class="row-fluid">
  <div class="span6">
    <h4 class="tocSectionTitle">Cartas al Director</h4>
```

Luego como un tercer nivel quedaría controlar la disposición de los elementos para garantizar la usabilidad de la revista con ello influir en la Arquitectura de la Información. Para lograr este nivel de configuración es necesario apoyarse en el motor de plantillas que propone el OJS, el Smarty. Tomando como punto de partida la definición que tiene los elementos de la página, éstos se pueden desagregar en bloques. Cada bloque puede ser definido en archivo y luego ser incluido según como se disponga, de manera que puede incluir toda una lógica la presentación de los datos que tenga incluido. {include file="common/nenuArriba.tpl"}

De igual manera se puede trabajar con los datos que sean proporcionados por el controlador, los cuales pueden ser tratados en dependencia de la interfaz lógica que implementen. Las interfaces que pueden ser tratadas en la plantilla son ItemIterator, ArrayItemIterator, DAOResultFactoryDRowIterator y VirtualArrayIterator(Smecher 2008).

Finalmente, queda trabajar con los mensajes, cabeceras e información que se muestra en la aplicación. Inicialmente se piensa en escribir directamente el texto en las páginas de la aplicación, pero ¿Que sucede cuando se quiere mostrar un mensaje diferente e incluso cambiarlos de idioma? El OJS propone una herramienta de internacionaliza-

ción con este fin (Smecher 2008). La internacionalización se encuentra definida en archivos XML en el directorio local y pueden ser invocadas desde las plantillas de la siguiente forma:

```
<h4>{translate key="announcement.announcementsHome"}</h4>
```

Utilizarla se reduce solamente a definir en los archivos de traducciones las nuevas claves de los mensajes y su contenido. En el caso de que se esté trabajando directamente sobre archivos JavaScript, se recomienda crear archivos JSON que contengan objetos con los diálogos y que sean cargados según el idioma que se encuentre activo.

Una vez que una revista institucional haya pasado por procesos específicos de configuración, se mostrará desde el punto de vista de usuario más apegada a la identidad, tendrá portabilidad y contendrá influencia positiva sobre la experiencia de usuario. Tal es el caso de la Revista MediSur y la Revista Ciencia de la Información las cuales se han adaptado a la institución que las aloja, ambas basadas en OJS, como se muestra en la siguiente figura:



Figura. 5 Revista Ciencia de la Información (<http://cinfo.idict.cu/index.php/cinfo>)

Conclusiones

Open Journal System, es un sistema de Código Abierto que se posiciona como herramienta para dar a conocer el conocimiento científico. Esta herramienta es ampliamente utilizada y conlleva una iniciativa para las comunidades de investigadores (Edgar, Willinsky 2010). Es por ello que a través de este trabajo se proponen técnicas que permiten acentuar la identidad de la presentación de la interfaz de usuario, incidir de manera positiva en la experiencia de usuario y permitir adaptabilidad ante los diferentes entornos de acceso. Se muestran casos de instituciones que han seguido diferentes niveles de configuración según los pasos explicados obteniendo un resultado tangible en la presentación de las revistas institucionales.

Referencias

- ALEXANDER BERNAL, 2011, Adinteractive | Agencia SEO, SEM, Desarrollo Web y Marketing Digital. [online]. 2011. [Accessed 21 November 2014]. Available from: <http://www.adinteractive.co/web/disenio-web-adaptable-practico>
- ALVARO FONTELA SANCHEZ, 2013, ¿Que es Bootstrap? [online]. 2013. [Accessed 21 November 2014]. Available from: <http://openwebcms.es/2013/que-es-bootstrap/>
- BAHIT, Eugenia, 2011, POO y MVC en PHP El paradigma de la Programación Orientada a Objetos en. 2011. P. 66.

- BIBLIOTECA UNIVERSIDAD DE ALCALÁ, 2014, Revistas científicas. 2014.
- CARLOS HERNAN AGUILAR HURTADO, 2014, Trabajar o no con motor de plantillas. [online]. 2014. [Accessed 21 November 2014]. Available from: <http://www.ingenieroweb.com.co/trabajar-o-no-con-motor-de-plantillas>
- EDGAR, Brian D and WILLINSKY, John, 2010, A Survey of the Scholarly Journals Using Open Journal Systems. 2010. Vol. 1, p. 1–22.
- GUTIÉRREZ, Claudia and SANTIAGO, Keplermedia S A, 2008, Diseño web y arquitectura de información Web Design and Information Architecture for 2 . 0 Sites. Cuadernos de Información. 2008. P. 58–65.
- LINKS, Adding Navigation, NEWS, Providing and BLOGS, Integrating, 2006, Customizing Your OJS Journal. Public Knowledge. 2006. P. 1–10.
- OHRT, Monte, ZMIEVSKI, Andrei and RAMÍREZ, Mario, 2010, Smarty Manual. New Digital Group. 2010. P. 244.
- RAMOS VELAZCO ADRIANET, Romero Rodríguez Felix Ivan (tutor), 2014, UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS TÍTULO : Componente de recuperación y difusión de noticias de. University of Information Sciences.
- SMECHER, Alec, 2008, OJS Technical Reference. 2008. Vol. 4, p. 52.



www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x