



Scientia Et Technica

ISSN: 0122-1701

scientia@utp.edu.co

Universidad Tecnológica de Pereira
Colombia

Gómez Codutti, Ana; Mariño, Sonia I.; Alfonzo, Pedro L.
Una propuesta integradora de Mantenimiento Correctivo aplicada al Diseño Web
Adaptativo y Accesibilidad Web
Scientia Et Technica, vol. 21, núm. 1, marzo, 2016, pp. 51-59
Universidad Tecnológica de Pereira
Pereira, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=84950584008>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

Una propuesta integradora de Mantenimiento Correctivo aplicada al Diseño Web Adaptativo y Accesibilidad Web

A corrective maintenance integrated proposal applied to Adaptive Web Design and Web Accessibility

Ana Gómez Codutti, Sonia I. Mariño, Pedro L. Alfonzo

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad Nacional del Nordeste. 9 de Julio 1449. 3400 Corrientes. Argentina.
simarinio@yahoo.com, plalfonzo@hotmail.com

Resumen— El incremento de las TIC ha promovido el desarrollo de herramientas informáticas libres y gratuitas, como los Content Management System o Sistemas de Gestión de Contenidos que facilitan la generación eficaz de contenidos digitales. Éstos deben ser evaluados previos a su utilización, a los efectos de producir un producto accesible por todos los usuarios. Se presenta una experiencia de Mantenimiento Correctivo del software aplicada a la Accesibilidad Web y el Diseño Web Adaptativo. La incorporación de estas pautas en el desarrollo de software, permiten impulsar la generación de productos tecnológicos de calidad y fomentan la inclusión digital de los individuos independientemente de sus capacidades.

Palabras clave— Mantenimiento Correctivo, accesibilidad web, Diseño Web Adaptativo, Sistemas Gestores de Contenidos.

Abstract— The increase of ICT has promoted free software such as Content Management System (CMS) that facilitate the efficient generation of digital content. CMS should be evaluated before to be to use, in order to generate accessible product. The work presents a corrective maintenance application implemented through Web Accessibility and Adaptive Web Design. The incorporation of this guidelines in software development, guarantee the quality of this products and promotes digital inclusion of individuals regardless of their abilities.

Key Word — Corrective Maintenance Software, Web Accessibility, Adaptive Web Design, Content Management Systems.

I. INTRODUCCIÓN

En la actualidad, puede observarse cómo los países industrializados dependen de sistemas software complejos, ya que sus distintos procesos de fabricación industrial y distribución se encuentran completamente informatizados [1].

La Ingeniería del Software (IS) trata “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, es decir, la aplicación de la Ingeniería al Software” [2]. En [3] se afirma que “se necesita disciplina, pero también adaptabilidad y agilidad”. Estos aspectos se vinculan con calidad en el software, que según el autor, cualquier enfoque de ingeniería debe basarse en un compromiso organizacional con ésta, para asegurar su mejora continua del software.

El Proceso de Software está constituido por el conjunto de actividades y resultados asociados que producen un producto software [1], quien afirma que existen cuatro actividades fundamentales de proceso: Especificación del software, Desarrollo del software, Validación del software y Evolución del software. Esta última incluye el proceso de mantenimiento.

Siguiendo lo expuesto en [4], el mantenimiento del software (MS) “se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente”. En [1] se considera al mantenimiento como “el proceso general de cambiar un sistema después de que éste ha sido entregado”. Si bien esta última definición señala que dichas actividades comienzan después que el producto se encuentra en funcionamiento, usualmente, algunas pueden comenzar antes de su entrega.

Los cuatro aspectos principales sobre los cuales se centra el mantenimiento son [5]:

- Mantener el control sobre las funciones cotidianas del sistema.
- Mantener el control sobre la modificación del sistema.
- Perfeccionar las funciones aceptables existentes.

- Prevenir que el rendimiento del sistema se degrada a niveles inaceptables.

En [6] se distinguen cuatro categorías de mantenimiento:

- Mantenimiento Correctivo: se realizan cambios en el software para corregir los defectos;
- Mantenimiento Adaptativo: se modifica el software para acomodarlo a los cambios de su entorno externo;
- Mantenimiento Perfectivo: se conduce al software más allá de sus requerimientos funcionales originales;
- Mantenimiento Preventivo: también llamado “reingeniería del software”, aplica cambios en el software para asegurar su corrección, adaptación y mejora más fácilmente.

Actualmente, existe un gran interés en la incorporación de pautas de calidad en el desarrollo de software, que impulsen la interacción y fomenten la inclusión digital de los individuos independientemente de sus capacidades. De esto se trata la accesibilidad web (AW), la cual se ha convertido en varios países del mundo en una preocupación.

En [7] se entiende a la accesibilidad como el acceso a la información contenida en los sitios web sin limitación alguna por cuestiones de deficiencia, minusvalía o tecnología utilizada sin que interfieran causas como: i) Problemas de oído, visión, movilidad; ii) Dificultades de lectura o comprensión cognitiva; iii) Imposibilidad de utilización del teclado o el ratón; iv) Lector de sólo texto, pantalla pequeña o conexión lenta.

Por lo que, la accesibilidad mejora el acceso a la web en general, siendo de interés también para organizaciones y personas sin discapacidad.

Como se mencionó en trabajos previos [8] [9] [10] [11] [12], la AW ([7] [13]) conforma un criterio de calidad del software, habiendo numerosos métodos y herramientas utilizadas en los procesos de diseño y desarrollo del software.

Por otra parte, el incremento de las TIC ha impactado en la disseminación de las herramientas informáticas libres y gratuitas, que conducen a una mayor eficacia en los procesos de gestión de contenidos digitales ([14], [15], [16]). Tal es el caso de los Content Management System o Sistemas de Gestión de Contenidos (CMS), como [17], [18], [19], entre otros.

Por otra parte, desde hace varios años se incrementó significativamente el uso de diferentes dispositivos portátiles, tales como smartphones o tablets. En este sentido, es necesario que las páginas web se adapten al dispositivo desde las que se accede para mejorar su visualización. El Diseño Web Adaptativo o Responsive Web Design (RWD) tiene como objetivo adaptar las páginas web a las características de resolución de cada dispositivo.

A partir de los conceptos vertidos, en este trabajo se aborda la propuesta de un modelo de proceso software, aplicado al MS,

centrado en la AW aplicado a un CMS de amplia difusión en el mercado.

II. METODOLOGÍA

Para ilustrar el Mantenimiento Correctivo en un CMS de libre distribución, se optó como objeto de estudio la Accesibilidad Web y el Diseño Web Adaptable. A continuación se exhibe el método asociado que constó de las siguientes etapas:

Etapa 1. Se relevaron proyectos similares vinculados al estudio de la AW, centrándose en continuar la evaluación en CMS [20], [21], [12].

Etapa 2. Se optó por Drupal [17] como CMS, ampliamente utilizado en la implementación de plataformas web.

Etapa 3. Se determinaron los criterios establecidos por las pautas WCAG 2.0 [22] para el Nivel “A”.

Etapa 4. Se llevó a cabo acciones concretas de AW. Esta etapa se compone de las siguientes iteraciones:

- Primera iteración.
 - ✓ Medición de la accesibilidad de la página principal del sistema desarrollado, implementado con la plantilla que responde al diseño RWD que permite visualizar las páginas web en diferentes dispositivos, por medio del validador TAW dado que automatiza la valoración de las pautas WAI 2.0
 - ✓ Sistematización y procesamiento de los datos.
- Segunda iteración.
 - ✓ Análisis y corrección de los errores detectados automáticamente.
- Tercera iteración.
 - ✓ Comprobación de la accesibilidad del sitio tras efectuar correcciones de los problemas detectados automáticamente por el validador TAW, con miras a concluir el estudio.
 - ✓ Comparación de los resultados obtenidos en los incrementos.
 - ✓ Planteamiento de futuras correcciones atendiendo a aquellos errores que requieren revisión manual.

III. RESULTADOS

En esta sección se presentan aspectos referentes al Mantenimiento Correctivo observados en el sistema elaborado, particularmente aquellos vinculados a la accesibilidad del mismo, atendiendo a que la AW es uno de los criterios de la calidad del software. En este sentido, [21] afirma que en los últimos años se han llevado a cabo numerosas acciones centradas en promover la sanción y aplicación de legislaciones concernientes a la accesibilidad web (AW). Con relación a esto, se menciona al W3C (Consorcio World Wide Web) [23], reflejada en su Iniciativa para la Accesibilidad a la Web (WAI o Web Accessibility Initiative) [13], cuyo objetivo es definir las pautas que

faciliten el acceso a los contenidos Web, por parte de las personas independientemente de sus capacidades.

El mantenimiento abordado en este trabajo se dividió en tres incrementos/iteraciones. En cada uno de ellos, la medición de la página principal del sistema se llevó a cabo mediante el validador automático TAW [24], de acuerdo a las pautas definidas por la WCAG 2.0 [22], en el CMS Drupal [17]. TAW [24] permite al usuario personalizar el examen, seleccionando el nivel de análisis requerido (A, AA o AAA) e indicar en la revisión las diferentes tecnologías soportadas como ser HTML, CSS o JavaScript. El software generó en cada caso, como resultado de la comprobación, un informe HTML conformado por las siguientes pestañas:

- Resumen: que expone de un modo general los problemas encontrados (detectados de forma automática), las advertencias (que deben revisarse manualmente) y los puntos no verificados (que requieren una comprobación completamente manual).
- Vista Marcada: que ilustra gráficamente sobre la página web las incidencias encontradas.
- Detalle: que indica los números de línea de código correspondientes a las incidencias detectadas, enlazando además las técnicas relacionadas, situadas en el sitio del W3C.
- Listado: que muestra un cuadro resumen, con los resultados obtenidos en cada una de las pautas y criterios de cada principio.

A. Primer incremento: Medición de la accesibilidad.

La Tabla 1, resume los resultados obtenidos tras evaluar la accesibilidad en la página principal del sitio implementado con Drupal [17], utilizando la plantilla “Software responsive theme” versión 7.x-1.0. Se describe, en la columna “Problemas”, la detección automática de errores de accesibilidad detectados por el validador. En la columna “Advertencias”, se indica la necesidad de realizar una revisión manual de los posibles problemas de accesibilidad y analizar su confirmación o descarte; y en la columna “No verificados”, se manifiesta la imposibilidad de validar o comprobar automáticamente, requiriéndose una completa revisión manual del sitio. En la columna “Principios”, se mencionan los establecidos por las WCAG 2.0.

Principios	Problemas	Advertencias	No verificados
	Son necesarias correcciones	Es necesario revisar manualmente	Comprobación completamente manual
Perceptible	1	19	3
Operable	0	16	6
Comprensible	0	2	2
Robusto	1	52	1

Tabla 1. Resumen de la evaluación de principios de AW. Primer Incremento.

La Tabla 1, refleja dos problemas automáticos. Con respecto al problema hallado en el Principio Perceptible, específicamente referente a la Pauta: Alternativas textuales, Criterio 1.1.1: Contenido no textual; se determinó la carencia del texto alternativo que debe acompañar a cada contenido no textual, para permitir que éste pueda ser transformado a otros formatos, según la deficiencia de las personas, tales como textos ampliados, braille, voz, entre otros [22].

En relación al problema relativo al Principio Robusto, el cual está relacionado a la Pauta: Compatible, Criterio 4.1.1: Procesamiento; hace alusión a que los contenidos implementados mediante lenguajes de marcas deben emplear una sintaxis válida de acuerdo a la gramática definida, por ejemplo: los elementos deben tener las etiquetas de apertura y cierre completas; deben estar anidados de acuerdo a sus especificaciones; no deben contener atributos duplicados y los identificadores deben ser únicos, excepto cuando las especificaciones lo permitan [22]. El no cumplimiento de esta recomendación influye en la efectividad del funcionamiento de los navegadores y la visualización correcta de las páginas.

B. Segundo incremento: Corrección de los errores detectados.

En este apartado se explica las correcciones pertinentes realizadas, luego del análisis realizado a los problemas detectados en el primer incremento.

Se reparó el error añadiendo el atributo “alt=...” y los datos correspondientes a los contenidos no textuales de la página. Visualizar en la Figura 1 (línea 85) el código insertado en la imagen existente en la página.

```

76 <div id="inner_header_wrapper">
77   <header id="header" role="banner">
78     <div class="top_left">
79       <?php if ($logo): ?>
80         <div id="logo">
81           <a href=<?php print $front_page; ?>" title=<?php print t('Home'); ?>">
82             <img src=<?php print $logo; ?>" alt="Logo FACENA." />
83           </a>
84         </div>
85       <?php endif; ?>
86     </div>
87   </header>
88 </div>
89

```

Figura 1. Código insertado referente a la imagen de la página.

Para verificar errores de sintaxis en el código HTML/XHTML, se requirió del uso de un validador complementario. Se optó por el validador de Markup Validation Service [25], detectándose errores y advertencias, relacionados a un incorrecto anidamiento de los elementos y existencia de elementos identificadores duplicados en el código. La Figura 2, refleja el resultado obtenido del análisis.

Se corrigieron los errores (ver Figura 3), accediendo al código fuente y eliminando los elementos identificadores duplicados relacionados al atributo “class” y realizando una disposición jerárquica correcta, de acuerdo a la gramática especificada de los elementos anidados.

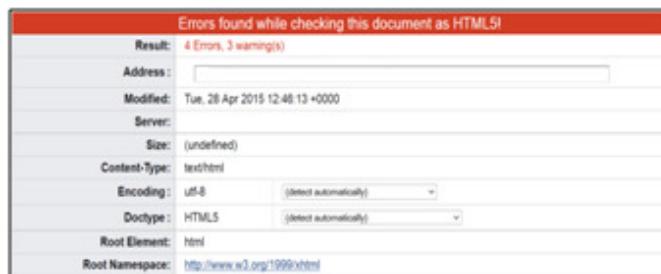


Figura 2. Resultado del primer análisis con Markup Validation Service.

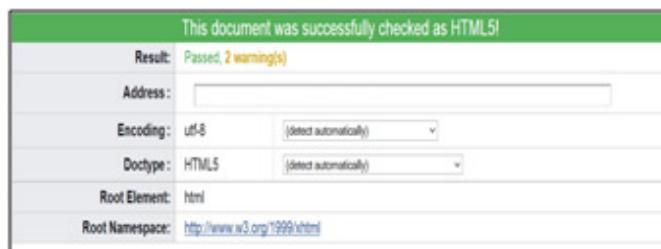


Figura 3. Resultado del segundo análisis con Markup Validation Service.

C. Tercer incremento: Comprobación de AW del sitio posterior a las modificaciones realizadas.

En esta sección se exhibe la revisión de accesibilidad originada tras las correcciones efectuadas, sobre la página principal del sitio. Se ilustra un resumen en la Tabla 2.

Principios	Problemas	Advertencias	No verificados
	Son necesarias correcciones	Es necesario revisar manualmente	Comprobación completamente manual
Perceptible	0	19	3
Operable	0	16	6
Comprendible	0	2	2
Robusto	1	52	1

Tabla 2. Resumen de la evaluación de principios de AW. Tercer Incremento.

Como se observa en la Tabla 2, el validador no marcó erróneo el Principio Perceptible; sin embargo, detectó problemas en el Principio Robusto; Pauta: Procesamiento; Criterio: 4.1.1. En la Figura 4, proporcionada por la herramienta TAW, se aprecia la ubicación del error en el código fuente vinculado a la llamada a las hojas de estilo.

En relación a esto, algunos navegadores más antiguos pueden no reconocer la instrucción `<style>...</style>` empleada para llamar a las hojas de estilo externas desde un documento HTML. En [22] se recomienda la utilización de la etiqueta `<link>` para realizar esta acción.

También resulta interesante el uso del atributo denominado “media”, que permite especificar el medio concreto al cual está dirigida la hoja de estilo (screen, print, braille, etc.) [22].

A continuación se expone en la Tabla 3, una comparación de los resultados alcanzados en referencia a la AW, en cada uno de los incrementos mencionados que conforman el Mantenimiento Correctivo abordado. Los valores que pueden asumir cada criterio especificado son: se cumplen (SI), no se cumplen (NO), no se aplican (N/A), requieren revisión manual (RRM) y son imposible de realizar comprobación automática (I/R).

Se resume en la Tabla 4, la evaluación detallada en la Tabla 3. La Figura 5, ilustra dicha información de acuerdo al porcentaje correspondiente en cada caso.

```

1. <!DOCTYPE html>
2. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es" dir="ltr">
3. <head>
4. <meta charset="utf-8" />
5. <link rel="shortcut icon" href="http://www.adscripcionesfacena.tk/sites/all/themes/software-responsive-theme/favicon.ico" type="image/vnd.microsoft.icon" />
6. <meta name="viewport" content="width=device-width" />
7. <link rel="alternate" type="application/rss+xml" title="Sistema de Adscripciones FACENA RSS" href="http://www.adscripcionesfacena.tk/rss.xml" />
8. <meta name="Generator" content="Drupal 7 (http://drupal.org)" />
9. ! <title>Sistema de Adscripciones FACENA</title>
10. <style type="text/css" media="all">@import url("http://www.adscripcionesfacena.tk/modules/system/system.base.css?nndcmm");
11. @import url("http://www.adscripcionesfacena.tk/modules/system/system.menus.css?nndcmm");
12. @import url("http://www.adscripcionesfacena.tk/modules/system/system.messages.css?nndcmm");
13. X @import url("http://www.adscripcionesfacena.tk/modules/system/system.theme.css?nndcmm");</style>
14. <style type="text/css" media="all">@import url("http://www.adscripcionesfacena.tk/modules/comment/comment.css?nndcmm");
15. @import url("http://www.adscripcionesfacena.tk/sites/all/modules/date/date_api/date.css?nndcmm");
16. @import url("http://www.adscripcionesfacena.tk/modules/field/theme/field.css?nndcmm");
17. @import url("http://www.adscripcionesfacena.tk/modules/node/node.css?nndcmm");
18. @import url("http://www.adscripcionesfacena.tk/modules/search/search.css?nndcmm");
19. @import url("http://www.adscripcionesfacena.tk/modules/user/user.css?nndcmm");
20. @import url("http://www.adscripcionesfacena.tk/sites/all/modules/views/css/views.css?nndcmm");
21. @import url("http://www.adscripcionesfacena.tk/sites/all/modules/ckeditor/css/ckeditor.css?nndcmm");</style>

```

Figura 4. Detalle del problema detectado en el principio Robusto.

Princípio	Pauta	Criterio	Primera medición de AW					Segunda medición de AW					Tercera medición de AW				
			SI	NO	N/A	I/R	RRM	SI	NO	N/A	I/R	RRM	SI	NO	N/A	I/R	RRM
1. Perceivable	1.1 Alternativas textuales	1.1.1 Contenido no textual		x						x							x
	1.2 Medios tempo dependientes	Todos (3 en total)			x						x				x		
	1.3 Adaptable	1.3.1 Información y relaciones					x					x					x
		1.3.2 Secuencia significativa					x					x					x
	1.4 Distinguible	1.4.1 Uso del color				x					x				x		

Princ pio	Pauta	Criterio	Primera medición de AW					Segunda medición de AW					Tercera medición de AW				
			SI	NO	N/ A	I/R	RR M	SI	NO	N/A	I/R	RRM	SI	NO	N/A	I/R	RRM
		1.4.2 Control del audio			X					X					X		
		1.4.3 Contraste				X					X					X	
2. Operable	2.1 Accesible por teclado	2.1.1 Teclado				X					X					X	
		2.1.2 Sin bloqueos de teclado				X					X					X	
	2.2 Tiempo suficiente	2.2.1 Tiempo ajustable				X					X					X	
		2.2.2 Poner en pausa, detener, ocultar				X					X					X	
	2.3 Convulsiones	2.3.1 Umbral de tres destellos o menos				X					X					X	
	2.4 Navegable	2.4.1 Evitar bloques				X					X					X	
		2.4.2 Titulado de páginas					X					X					X
		2.4.3 Orden del foco					X					X					X
		2.4.4 Propósito de los enlaces	X					X					X				

Principio	Pauta	Criterio	Primera medición de AW					Segunda medición de AW					Tercera medición de AW				
			SI	NO	N/A	I/R	RRM	SI	NO	N/A	I/R	RRM	SI	NO	N/A	I/R	RRM
3. Comprensible	3.1 Legible	3.1.1 Idioma de la página	x					x					x				
	3.2 Predecible	3.2.1 Al recibir el foco					x				x					x	
		3.2.2 Al recibir entradas				x					x				x		
	3.3 Entrada de datos asistida	3.3.1 Identificación de errores					x					x					x
		3.3.2 Etiquetas o instrucciones	x					x					x				
4. Robusto	4.1 Compatible	4.1.1 Procesamiento					x		x					x			
		4.1.2 Nombre, función, valor				x					x					x	

Tabla 3. Comparación de las mediciones de AW en los tres incrementos.

Criterios	Primer medición	Segunda medición	Tercer medición
Se cumplen	3	3	3
No se cumplen	1	2	1
No se aplican	4	4	4
Requieren revisión manual	7	5	6
Imposible realizar comprobación automática	11	12	12

Tabla 4. Comparación cuantitativa de criterios de AW en cada medición.

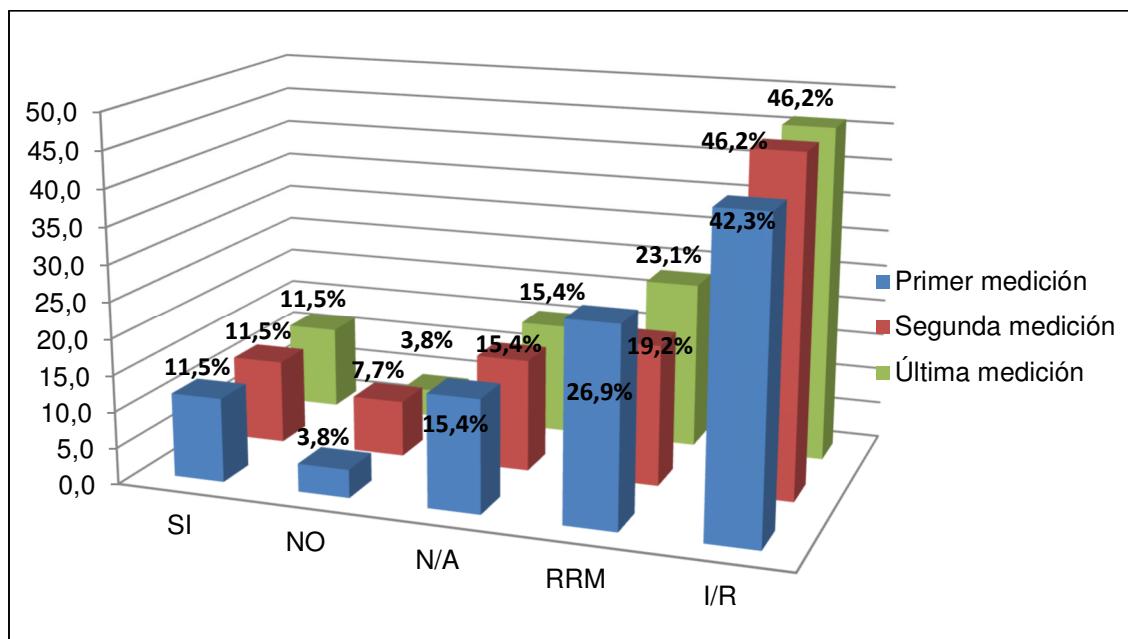


Figura 5. Comparación porcentual de criterios de AW en cada medición.

IV. CONCLUSIONES

Al concluir el estudio realizado con respecto al mantenimiento de software, enfatizando en el mantenimiento Correctivo y su desarrollo experimental, se puede ratificar la importancia de su abordaje en el proceso de desarrollo de un producto software. Concretamente, la indagación ha permitido:

- sugerir un modelo de proceso de software con énfasis en el mantenimiento ágil, diseñando y desarrollando versiones frecuentes hasta lograr la aceptación del mismo;
- determinar el cumplimiento de las pautas de accesibilidad en una plantilla que responde a lineamientos de diseño web adaptativo en el CMS elegido;
- mejorar las funcionalidades proporcionadas por un sistema gestor de contenidos;
- disminuir los costos vinculados con el Mantenimiento Correctivo;
- realizar correcciones en lo pertinente al Mantenimiento Correctivo.

Dado que el desarrollo se basó en el Mantenimiento Correctivo aplicado a Drupal, un CMS de libre distribución, se consideran los hallazgos un aporte a la comunidad de desarrolladores.

A modo de futuros trabajos, se continuará indagando en métodos y herramientas para efectivizar el MS en sistemas gestores de contenidos. Además, se completará el Mantenimiento Correctivo aplicando mejoras de código según los detalles determinados bajo la denominación “requiere revisión manual”.

REFERENCIAS

- [1] I. Sommerville, “Ingeniería del Software”, 7ma edición: Pearson, 2005.
- [2] IEEE, "IEEE Standard Glossary of Software Engineering Terminology". IEEE Std 610.121990, 1990.
- [3] R. Pressman, “Ingeniería del Software. Un enfoque práctico”. 7ma edición: McGraw-Hill/Interamericana Editores, 2010.
- [4] R. Pressman, “Ingeniería del Software. Un enfoque práctico”. 5ta edición: McGraw Hill/Interamericana Editores, 2002.
- [5] T. Pigoski, SWEBOK Knowledge Area Description for Software Evolution and Maintenance (version 0.5). [En línea]. [http://www.elen.ktu.lt/~rsei/Swebok/KA_Description_Software_Evolution_Maintenance\(Version_0_5\).PDF](http://www.elen.ktu.lt/~rsei/Swebok/KA_Description_Software_Evolution_Maintenance(Version_0_5).PDF)
- [6] R. Pressman, “Ingeniería del Software. Un enfoque práctico”. 5ta. edición: Mc Graw Hill/Interamericana Editores, 2002.
- [7] G. A. Toledo, "Accesibilidad digital para usuarios con limitaciones visuales", Universidad Nacional de la Plata, Facultad de Informática, La Plata, 2012. [En línea]. <http://hdl.handle.net/10915/24871>.
- [8] A. Fernández Vázquez, J. J. Acevedo, S. I. Mariño, M. V. Godoy, and P. Alfonzo, “Medición de la accesibilidad en dos sitios web municipales de las provincias de Corrientes y

- Chaco, Argentina". *Revista Telematique*, enero-junio 2013. ISSN: 1856-4194.
- [9] J. J. Acevedo, L. Gómez Solis, S. I. Mariño, and M. V. Godoy, "A guidelines for evaluating web accessibility, Level A". *Journal of Computer Science & Technology*. JCS&T, 13(2): 76-83, 2013.
- [10] S. I. Mariño, R. Alderete, S. Ferrari Alve, C. R. Primorac, and M. V. Godoy, "Evaluación de accesibilidad en sitios Web educativos basados en CMS". *Revista Digital Sociedad de la Información*, no. 39, 2013.
- [11] P. Alfonzo, S. Mariño, J. I. Cavalieri, and A. Gómez Codutti, "Accesibilidad Web: su abordaje en Congresos Argentinos de Informática en el período 2012-2013". *Novática - Revista de la Asociación de Técnicos de Informática*, no. 229, p. 102, 2014.
- [12] D. Casaro, P. Alfonzo, S. Mariño, M. Godoy, "Mantenimiento Correctivo Aplicado a un Sitio Basado en Joomla. Una Propuesta Centrada en la Accesibilidad". *Revista Latinoamericana de Ingeniería de Software*, 3(2): 101-107, ISSN 2314-2642
- [13] W3C. Web Accessibility Initiative. [En línea]. <http://w3c.es/Traducciones/es/WAI/intro/accessibility>
- [14] M. Pérez-Montoro Gutiérrez, "Sistemas de gestión de contenidos en la gestión del conocimiento". *Bid Textos universitaris de biblioteconomia i documentació*, no. 14, 2005.
- [15] J. Rodríguez i Gairín, "PHP-Nuke: software de código abierto en la gestión de contenidos Web". *El Profesional de la Informacion*, vol. 13, no. 6, 2004.
- [16] J. Betetta, M. Castro Díaz, C. Flores, and R. Palavecino, "Evaluación de las características y comparación de los Sistemas de Gestión de Contenidos", *CACIC 2010 - XVI Congreso Argentino de Ciencias de la Computación*, 2010.
- [17] Drupal Hispano. [En línea]. <http://drupal.org.es/caracteristicas>.
- [18] Joomla. Joomla Homepage. [En línea]. <http://www.joomla.org/>
- [19] Wordpress. Wordpress Homepage. [Online] <https://wordpress.org/>
- [20] S. Mariño, R. Alderete, S. Ferrari Alve, C. Primorac, M. V. Godoy, "Evaluación de accesibilidad en sitios web educativos basados en CMS". *Sociedad de la Información*, no. 39, 2013.
- [21] S. Mariño, P. Alfonzo, A. Gómez Codutti, and M. V. Godoy, "Automatic Evaluation of WCAG 2.0 Guidelines in a Drupal-based Platform", *International Journal of Information Science and Intelligent System*, vol. 4, no. 1, pp. 35-42, 2015.
- [22] W3C. WCAG. Web Content Accessibility Guidelines (WCAG) 2.0. [En línea]. <http://www.w3.org/TR/WCAG20/>
- [23] W3C. Consorcio World Wide Web. [En línea]. <http://www.w3c.es/>
- [24] TAW. Test de Accesibilidad Web "TAW". [En línea]. <http://www.tawdis.net/>
- [25] W3C. W3C Markup Validation Service. [En línea]. <http://validator.w3.org/>