

# AFDA, UNA NUEVA PROPUESTA BASADA EN MÓDULOS REUTILIZABLES JEE PARA EL DESARROLLO DE WEBS ACCESIBLES

Juan Marcos Filgueira Gomis

Universidad Nacional de Educación a Distancia.  
Escuela Técnica Superior de Ingeniería Informática.  
Departamento de Lenguajes y Sistemas Informáticos.  
Madrid, España  
Juan.marcos.filgueira@edu.xunta.es

Covadonga Rodrigo San Juan

Universidad Nacional de Educación a Distancia.  
Escuela Técnica Superior de Ingeniería Informática.  
Departamento de Lenguajes y Sistemas Informáticos.  
Madrid, España  
Covadonga@lsi.uned.es

**Abstract**— Actualmente la mayoría de sitios y aplicaciones web presentan deficiencias en cuanto a los requerimientos de accesibilidad a que obliga la legislación de cada país. El problema de la accesibilidad web no se encuentra plenamente aceptado por la comunidad de desarrolladores de productos software, existiendo una carencia de herramienta de autor orientadas a desarrollo accesible. Este artículo presenta una nueva metodología de desarrollo software centrada en el problema de la accesibilidad que proporciona un marco o base teórica para dar soporte a diferentes herramientas a modo de componentes y entornos de desarrollo JEE. La implementación de estas herramientas es denominada AFDA: Aplicación y Framework para Desarrollo Accesible. AFDA presenta un conjunto de funcionalidades diseñadas para facilitar tareas comunes en el desarrollo de aplicaciones a medida a través de la incorporación de un grupo de módulos reutilizables que pueden integrarse en cualquier componente software que haga uso de esta tecnología.

**Keywords**-component; Accesibilidad web, Metodologías software, Herramientas de desarrollo, Ingeniería web.

## I. INTRODUCCIÓN Y PROBLEMÁTICA

A pesar de que organismos internacionales como el W3C [1] promueven normas y directrices de accesibilidad tanto para la elaboración de contenidos, WCAG [2], como para el desarrollo de herramientas de autor, ATAG [3], la realidad es en cierto modo decepcionante y es habitual encontrar todavía hoy interfaces web no accesibles que impiden o dificultan el acceso a la información y a los servicios web a gran parte de los usuarios de Internet. El problema de la accesibilidad todavía no se encuentra plenamente asumido por los consultores y desarrolladores de software, de forma que todavía existe una carencia de entornos y herramientas de desarrollo que hagan especial hincapié en el cumplimiento de las directrices de accesibilidad. Es en este punto donde el proceso de evaluación de conformidad con las pautas de accesibilidad así como las técnicas y herramientas necesarias para llevarlo a cabo no se encuentran tan bien definidos como en otros contextos.

En la actualidad, la mayoría de metodologías de desarrollo software aplicadas a la evaluación del cumplimiento de las

directrices de accesibilidad se basan en la aplicación de ciclos de validación una vez concluida la fase de desarrollo. Por tanto la validación de la accesibilidad se lleva a cabo al margen del ciclo de vida, o más concretamente, una vez desarrollados los prototipos o componentes finales. Pero incluso en el caso de que existan dentro del proceso de desarrollo tareas específicas de validación y verificación, el enfoque habitual implica "arreglar errores a posteriori". En este planteamiento, el componente software sufre una adaptación una vez que ya ha sido desarrollado, poniéndose en el peor de los casos a la hora de hacer efectivos los recursos y el tiempo de desarrollo.

Este artículo muestra un análisis del proceso de desarrollo de software señalando los puntos de ruptura con métodos anteriores. La metodología propuesta se basa en la implicación de las directrices de accesibilidad dentro del proceso ya desde las primeras fases del ciclo de vida. Este enfoque "a priori" del problema busca la optimización y factorización de elementos clave del componente a desarrollar de manera que aumente la calidad desde los primeros desarrollos y se reduzca el número de iteraciones necesarias a fin de obtener un producto completamente accesible.

## II. OBJETIVOS

Es posible integrar requisitos de accesibilidad en todas las fases del ciclo de vida, y además existen ejemplos de aplicación de herramientas que realizan funciones tales como validación de normas de accesibilidad o publicación de contenido web basadas en la estándar universal de lenguaje de marcado multipropósito XML.

Añadiendo a estas capacidades las características dinámicas de los lenguajes de servidor, que permiten la generación de código adaptado a múltiples parámetros, estas soluciones pueden ser utilizadas para desarrollar soluciones comunes a problemas de accesibilidad que hagan uso de componentes modulares y posibiliten su reutilización.

Si a su vez estos conjuntos de soluciones se integran en entornos de desarrollo lo suficientemente abiertos y adaptables como para poder compartirse entre componentes software de

nueva creación además de en futuros desarrollos, estaremos ante una solución lo bastante próxima a la problemática situación actual como para hacerla viable, puesto que no supondría dejar atrás los componentes software ya existentes en el proceso.

Por otro lado, resulta fundamental recordar que los distintos módulos y elementos deben ser totalmente independientes entre sí para alcanzar una solución realmente abierta, además de ser capaz de adaptarse a cualquier problema o entorno seleccionando los módulos más adecuados para cada situación.

Por último, centrándose en el ámbito de la accesibilidad y atendiendo a las principales recomendaciones y regulaciones a nivel mundial al respecto [4] [5], el objetivo principal sería conseguir el nivel doble A como requisito mínimo para cada componente generado, lo que aseguraría cumplir con las diferentes normativas. Por otra parte, sería recomendable incluir además el mayor número posible de elementos triple A de manera que se mantengan unos niveles de usabilidad lo suficientemente altos, buscando siempre el equilibrio entre ambos aspectos garantizando la mayor facilidad de uso para los usuarios en los componentes software desarrollados.

### III. METODOLOGÍA

A continuación se presenta una nueva metodología de desarrollo accesible. Esta metodología proporciona el marco teórico necesario para dar soporte al desarrollo de AFDA (Aplicación y Framework para Desarrollo Accesible) desde su versión inicial hasta sus evoluciones futuras. Es por tanto que la solución que se propone en este artículo es una solución conjunta, formada por: metodología, marco de arquitectura, aplicación y entorno de desarrollo. Por su parte, aplicación y *framework* se aglutinan en AFDA como una solución basada en *frameworks* del lado del servidor haciendo uso de tecnologías JEE que toma parte en todas las etapas del ciclo de vida de desarrollo.

Dentro del planteamiento de esta metodología resulta fundamental integrar requerimientos de accesibilidad desde las primeras fases del proceso, empezando por la fase de análisis y empleándolos para, en un primer momento, implementar objetos con la suficiente calidad como para que resulten reutilizables y aplicables al mayor número posible de componentes software. Podemos diferenciar dos principales categorías de metodologías sobre accesibilidad, en primer lugar, las que se basan en la fase de evaluación de componentes, como la propuesta por el proyecto europeo UWEM [6]. Este tipo de soluciones buscan mejorar la calidad de las evaluaciones así como ampliar o concretar el ámbito de las WCAG a dominios particulares. En segundo lugar, se encuentran aquellas metodologías, propuestas o trabajos que se centran en el área del diseño centrado en el usuario, ya sea orientadas hacia una usabilidad universal [7], o más específicas para el problema de la accesibilidad con una implicación directa de las directrices correspondientes [8]. Cabe destacar también que en algunas de estas propuestas se alcanza incluso el ámbito de las aplicaciones RIA [9]. En general, podemos extraer que estos trabajos proporcionan pautas o recomendaciones de diseño generales atendiendo a necesidades específicas de los usuarios agrupándolos en diferentes perfiles.

Dentro del proceso de ciclo de vida existe un vacío entre estos dos ámbitos de actuación. El objetivo de la metodología aquí presentada es establecer un marco para todo el proceso de ciclo de vida, permitiendo siempre su integración con cualquier otra de estas iniciativas, de manera que todas tengan cabida en una solución conjunta. La idea es ampliar el modelo actual para abarcar la totalidad de fases implicadas en un desarrollo para optimizar así el uso de recursos.

#### A. Fase 1: Desarrollo de soluciones base.

##### 1) Análisis y Diseño.

El planteamiento general de la metodología implica, para cada fase del proceso incluir requerimientos específicos de accesibilidad sin olvidar las correspondientes necesidades de usabilidad. Empezando por las fases de análisis y diseño se deben incluir requisitos específicos de usuario, directrices de accesibilidad (WCAG, XAG [10], ARIA [11], ...), adecuación a estándares (HTML, CSS, XML, ...), requisitos de tecnologías asistivas y requisitos semánticos. Cabe destacar que en este planteamiento tendría cabida cualquier otra metodología de diseño centrado en el usuario que se requiriera emplear.

##### 2) Implementación.

A continuación, a partir del diseño específico hacia la accesibilidad del punto anterior, se definen una serie de módulos de desarrollo que hacen uso de conjuntos de elementos reutilizables para la creación de componentes software.

##### 3) Revisión y depuración.

Cada elemento dentro de un módulo se depura, valida y mejora hasta que presenta la suficiente calidad como para asegurar el desarrollo de nuevos elementos completamente accesibles desde el primer momento en que son creados. Los errores o posibles mejoras detectadas retroalimentan la fase implementación o, en caso de considerarlo necesario, las de diseño e incluso análisis, para a continuación volver a implementarlo y depurarlo en un desarrollo iterativo.

Para un ámbito de actuación tan amplio como el que nos ocupa, cabe recordar que se está considerando establecer una solución para el mayor número de aplicaciones web posible, el coste de desarrollo inicial de añadir todos estos nuevos requerimientos no cabe duda de que será elevado. La idea es que este esfuerzo se vea recompensado con cada implantación individual de la solución obtenida resultando de esta forma viable.

#### B. Fase 2: Integración y nuevos desarrollos.

A partir de la solución estable obtenida en la primera fase pasamos a analizar si se adapta a nuestro problema concreto. En caso de no ser así podría volverse a aplicar la primera fase, atendiendo a las condiciones específicas del problema o, en caso de ser válida la solución ya disponible, hacer uso de ésta directamente. Esto puede llevarse a cabo o bien integrando el *framework* en componentes ya existentes o, en un segundo modo de aplicación, a partir del mismo y el componente base para desarrollar aplicaciones completamente nuevas.

### 1) Nuevos desarrollos.

A partir de la solución base seleccionada comienza a desarrollarse una nueva aplicación haciendo uso de las funcionalidades del componente base y/o el entorno de desarrollo. A continuación tiene lugar una revisión de los componentes software desarrollados a través de la validación de accesibilidad y estándares siguiendo el modelo tradicional de ciclos de validación.

### 2) Integración en componentes software ya existentes.

En soluciones ya existentes y definidas, puede integrarse el *framework* en el componente software para ir añadiendo o modificando elementos de manera parcial hasta adaptar la totalidad del componente. Los elementos a incorporar a partir del entorno de desarrollo podrán configurarse según cada problema o situación. Del mismo modo que en el apartado anterior debe tener lugar una revisión o validación del componente según se vayan incorporando elementos a fin de evaluar la calidad del mismo.

## IV. MARCO DE ARQUITECTURA

Como extensión de la metodología anteriormente descrita se define el siguiente marco de arquitectura, presentado a grandes rasgos enumerando sus principales características y componentes:

Como principal característica y piedra angular de la solución real, estaría el módulo de desarrollo implementado a través de *frameworks* de lado del servidor que factoricen funcionalidades comunes a las aplicaciones a medida del dominio.

Un segundo aspecto sería la inclusión de un módulo de validación, basado inicialmente en el modelo de los ciclos de validación y ampliado para cubrir además las nuevas necesidades actuales, aquellas que no se incluyen en las pautas WCAG 1.0, como validaciones para interpretación de metainformación por parte de tecnologías asistivas. Es para este punto donde trabajos como [12] o [13] mostrarían el camino a seguir. Además de estas, deberían considerarse también un nuevo conjunto de validaciones basadas en reglas adaptadas para aplicaciones RIA.

Una tercera característica se basaría en la integración de un módulo de sistema de gestión de contenidos (CMS) [14] que cubriría el ámbito de aplicación de estas herramientas, siendo capaz de integrarse con los componentes software generados.

Un cuarto elemento surge como extensión derivada de las necesidades de usabilidad. Los requerimientos actuales de las aplicaciones web hacen necesario contemplar la inclusión de tecnologías RIA, lo que se haría a través de un módulo de desarrollo específico y teniendo siempre en cuenta que la inclusión de estos elementos debe hacerse a modo de extensión o complemento de la salida HTML principal.

Por último, derivado del uso de tecnologías Web 2.0 [15], se incorporaría un módulo semántico. Tendría como objetivo extender la funcionalidad de los objetos a través de metadatos y facilitar así su interpretación por parte de las API de accesibilidad y tecnologías asistivas.

## V. ESTADO ACTUAL

A partir del modelo conceptual, el siguiente paso fue comenzar a construir una solución inicial con una orientación práctica adaptada a las condiciones reales, aunque limitada por la viabilidad del proyecto. La solución se encuentra formada por un entorno de desarrollo que se complementa con una aplicación base, elementos que se encuentran a día de hoy ya implementados incorporando el número suficiente de módulos como para permitir el desarrollo de aplicaciones completas. Sus características principales se describen a continuación a modo de resumen ya que por los límites de extensión del artículo no es posible un análisis en detalle a nivel de código o diagramas de clases.

### A. Framework

El componente principal de esta solución es el módulo de desarrollo. Este módulo se encuentra formado por un conjunto de componentes que funcionan a modo de *framework* para desarrollo accesible, que incluyen además consideraciones de usabilidad. La idea tras este entorno de desarrollo es permitir la integración de una amplia variedad de componentes que permitan ir añadiendo características de una forma lo más independiente posible, de manera que puedan ser reutilizados en otras situaciones, tanto como solución conjunta como de forma separada. La solución que se encuentra actualmente implementada ya incorpora los suficientes elementos accesibles como para desarrollar aplicaciones a medida completas, como por ejemplo: una estructura de aplicación basada en plantillas, estilos, menús, gestión de mensajes y ayudas integrada, así como etiquetas, modelos de formularios de entrada de datos y pantallas de presentación de información.

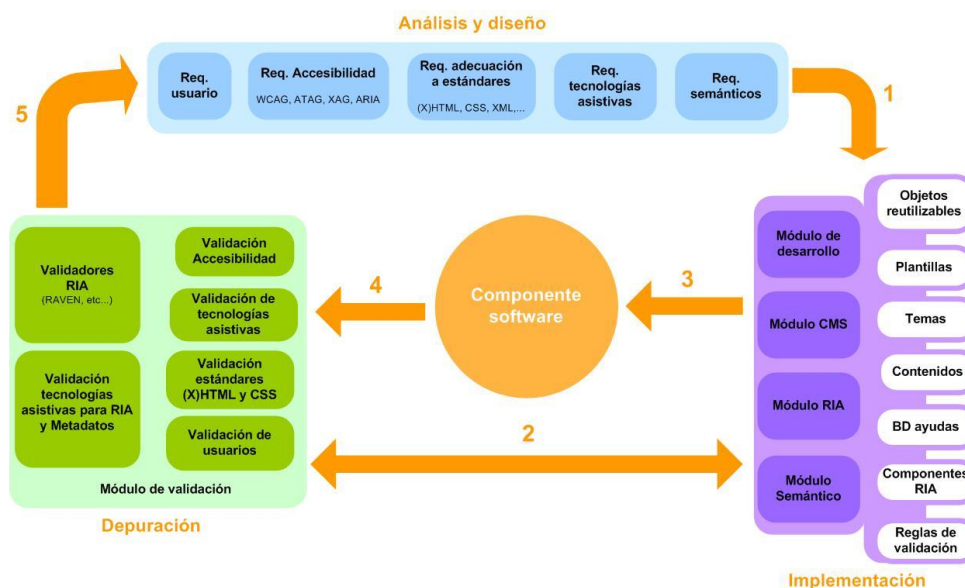


Figure 1. Marco de arquitectura

## B. Aplicación

En segundo lugar, la solución aquí presentada incorpora además una aplicación o componente software base a modo de estándar de la solución donde se incluyen características comunes, además de código para el desarrollo de nuevos componentes. El objetivo de esta aplicación base es doble, además de proporcionar un ejemplo completo de funcionamiento representa un punto de partida para el desarrollo de nuevos componentes software a partir de su modificación con una serie de tareas automatizadas.

En esta solución inicial, se busca establecer un *framework* básico que funcione como núcleo del sistema conteniendo el conjunto principal de módulos reutilizables, aunque además permanece abierto a la incorporación progresiva de nuevos módulos. Otro factor a considerar es la automatización de cualquier tipo de proceso o funcionalidad, de modo que la generación de nuevos elementos usables y accesibles sea más una cuestión de configuración o programación que un trabajo manual para el desarrollador.

En este momento, AFDA a partir de la aplicación base y el *framework* implementados ya es capaz de generar todas las partes necesarias de una aplicación hecha a medida, siendo todas ellas accesibles desde el primer prototipo generado.

## VI. RESULTADOS

En los primeros experimentos, el objetivo fue la generación de aplicaciones destinadas al ámbito de la e-Administración europea. Se han desarrollado diversas aplicaciones de entrada y presentación de información atendiendo a casos reales basados en la experiencia de los desarrolladores.

La validación de normas y directrices de accesibilidad inmediatamente posterior a la generación devuelve una serie de resultados de estas nuevas aplicaciones que coinciden con los de la aplicación base AFDA. Estas nuevas aplicaciones cumplen con los estándares doble-A, incluyendo algunos elementos de Triple-A, para herramientas como TAW [16] o HERA [17]. Por otra parte presentan un código HTML y CSS que es 100% válido.

En términos de coste de desarrollo, AFDA proporciona dos mejoras básicas. En primer lugar, reducir los costes de desarrollo accesible respecto a soluciones basadas en ciclos de validación, debido al hecho de que las aplicaciones son accesibles ya desde sus primeros prototipos. En segundo lugar, AFDA proporciona integración con otros *frameworks* JEE, como por ejemplo Struts [18], además de un sistema de menús y una modelo de estructura predeterminada. Todas estas características se implementan empleando un simple paso de configuración.

Un nivel por debajo se encontraría cuantificar la optimización en el proceso de desarrollo de otras páginas internas, como formularios. En este sentido, hemos realizado mediciones empleando un modelo ampliado de WebMo [19] con un grupo de desarrolladores senior con nociones básicas de accesibilidad. Los resultados permiten una estimación de una reducción de los costes de desarrollo en torno al 60-65% para cada formulario accesible de entrada de datos.

## VII. CONCLUSIONES

Como conclusión, decir que este sistema representa, desde nuestro punto de vista, un nuevo enfoque en el ámbito del desarrollo web accesible, al incluir aspectos de accesibilidad desde las primeras etapas del desarrollo software, en la búsqueda de mejorar las técnicas y procedimientos utilizados en el desarrollo tradicional. La principal contribución es establecer el modelo AFDA como una nueva metodología de desarrollo, donde las aplicaciones web sean accesibles desde su prototipo inicial. Esta orientación evita caer en los costosos ciclos sucesivos de validación que se requieren para reparar y corregir errores individualmente.

Paralelamente a este planteamiento, la solución hace uso de *framework* del lado del servidor, proporcionando una salida única válida para cualquier cliente, por tratarse de código estándar, ya accesible. Este enfoque se fundamenta en el hecho de un diseño universal, único y para todos, independientemente del cliente, garantizando el funcionamiento mínimo del sistema en todo caso.

Además, esta metodología permite un sistema compuesto por módulos independientes optimizados para el desarrollo accesible que pueden crecer y mejorar a lo largo del tiempo, definiendo un ciclo de retroalimentación entre los distintos proyectos permitiendo sustentar una labor de aprendizaje que no tendría cabida en métodos tradicionales.

Desde una perspectiva profesional, AFDA proporciona una solución práctica orientada al mundo real, para el que fue desarrollada. Desde esta, su primera versión asegura el desarrollo de aplicaciones a medida accesibles desde el primer momento en que son creadas. En este sentido, su diseño modular permitirá incluir nuevas funcionalidades que la dotarán en el futuro de una mayor potencia de desarrollo. La solución inicial aquí presentada busca proporcionar un punto de partida que de manera progresiva vaya creciendo con nuevas funcionalidades y herramientas hasta llegar la solución completa en que se basa la metodología como objetivo final.

## VIII. REFERENCIAS

- [1] World Wide Web Consortium, <http://www.w3.org>
- [2] W3C (1999), Web Content Accessibility Guidelines 1.0, <http://www.w3.org/tr/WCAG10>
- [3] W3C (2000), Authoring Tool Accessibility Guidelines 1.0, <http://www.w3.org/tr/wai-autotools>
- [4] Europa (2002). eEurope 2005: Una sociedad de la información para todos. Comisión al Consejo al Parlamento Europeo, al Comité Económico y Social Europeo y al Comité de las Regiones. Disponible en: <http://ec.europa.eu>.
- [5] Estados Unidos (2001), Estándares de la sección 508. Disponible en: <http://www.section508.gov>.
- [6] Web Accessibility Benchmarking Cluster (2007), UWEM, Unified Web Evaluation Methodology version 1.2. [http://www.wabcluster.org/uwem1\\_2/](http://www.wabcluster.org/uwem1_2/)
- [7] Newell, A.F., Gregor, P. (2000), User Sensitive Inclusive Design: in search of a new paradigm. En CUU 2000 First ACM Conference on Universal Usability. p.39-44.
- [8] Stephanidis, C. y otros, (1998), Universal Accessibility in HCI: Process oriented design guidelines and tool requirements. En Proceedings of the 4th ERCIM Workshop on User Interfaces for All. Disponible en: <http://ui4all.ics.forth.gr>

- [9] Velasco C.A. y otros (2008), A web compliance engineering framework to support the development of accessible rich internet applications. En ACM International Conference Proceeding Series; Vol. 317.
- [10] W3C (2002), XML Accessibility Guidelines (XAG). Disponible en: <http://www.w3.org/TR/xag>.
- [11] W3C (2007), Roadmap for Accessible Rich Internet Applications (WAI-ARIA Roadmap). Disponible en: <http://www.w3.org/TR/aria-roadmap/>
- [12] Fairweather P. y otros (2007), From Assistive Technology to a Web Accessibility Service. En Proceedings of the fifth international ACM conference on Assistive technologies.
- [13] Tsonos D. y otros (2007), Auditory Accessibility of Metada in Books: A Design for All Approach. En Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction (Part III).
- [14] Rainville-Pitt S., D'Amour J-M. (2007), Using a CMS to create fully accessible websites. En ACM International Conference Proceeding Series.
- [15] Cooper, M. (2007), Accessibility of Emerging Rich Web Technologies: Web 2.0 and the Semantic Web. En ACM International Conference Proceeding Series.
- [16] Web Accessibility Test, <http://www.tawdis.net/taw3/cms/en>
- [17] Fundación Sidar, HERA <http://www.sidar.org/hera/>
- [18] Apache Software Foundation, (2007), Documentación oficial sobre Struts. Disponible en: <http://struts.apache.org>.
- [19] Reifer, D. (2002), Estimating Web Development Costs: There Are Differences. CrossTalk, en The Journal of Defense Software Engineering Vol 15, N° 6, p. 13-17.