Extendiendo MVC para Diseñar Interfaces de Usuario Accesibles

Brenda V. Bustos Torres
Departamento de Ciencias de la Computación, Universidad Nacional del Comahue,
(8300)Neuquén, Argentina
Bbustos@sumicomp.com

y

Adriana E. Martín, **Alejandra S. Cechich**
Grupo de Investigación en Ingeniería de Software del Comahue (GIISCo)
Departamento de Ciencias de la Computación, Universidad Nacional del Comahue
(8300)Neuquén, Argentina
adrianaelba.martin@gmail.com // acechich@uncoma.edu.ar

Abstract

The lack of Accessibility of users' interfaces of Web sites is a dilemma that affects a large number of persons. Improving accessibility means assuring Internet access "to all" independently of the technology that they use and the different capabilities they possess. Our work proposes the design of accessible user's interfaces by mapping the "Web Content Accessibility Guidelines 1.0" over an architectural framework that associates the strengths of the "Model View Controller" pattern combined with user interface design decisions. Our approach aims to help Web developers capture design decisions when building accessible user interfaces for people with disabilities.

Keywords: Web Accessibility, User Interface Design, MVC.

Resumen

La carencia de Accesibilidad en las interfaces de usuarios de los sitios Web es un dilema que afecta a un gran número de personas. Derribar barreras de Accesibilidad significa asegurar "a todos" el acceso a Internet independientemente de la tecnología que utilicen y de las capacidades diferentes que posean. Nuestro trabajo apuesta al diseño de interfaces de usuario accesibles proponiendo un mapeo de las pautas de la "Web Content Accessibility Guidelines 1.0" sobre un framework arquitectónico que asocia las fortalezas del diseño de interfaces de usuario y del patrón "Model View Controller". Nuestro enfoque fija su objetivo en asistir a los desarrolladores Web en la toma de decisiones de diseño para la construcción de interfaces de usuario más accesibles para personas con capacidades diferentes.

Palabras claves: Accesibilidad Web, Diseño de Interfaz de Usuario, MVC

1 INTRODUCTION

La Web (World Wide Web) se creó como una red universal de conocimiento y ha significado un enorme salto cualitativo y cuantitativo en lo que a adquisición y tratamiento de información se refiere. Así mismo, se ha extendido en diferentes áreas tales como el comercio electrónico "e-commerce", el aprendizaje electrónico "e-learning", el comercio móvil "m-commerce" y los negocios electrónicos "e-bussiness", permitiendo a las organizaciones adaptarse rápidamente a los cambios, automatizar sus procesos de negocio y responder oportuna y adecuadamente a las necesidades de sus clientes y requerimientos del mercado manteniendo la competitividad.

Con el advenimiento de la sociedad de la información, cada vez es más el número de personas que utilizan la Web. Al igual que cualquier otro sistema electrónico de información, una aplicación Web debe presentar la información de tal manera que las personas, independientemente de la tecnología que utilicen (ordenador, PDA, teléfono, etc.) y de las capacidades diferentes que posean (físicas,

psíquicas, sensoriales u otras) estén en igualdad de condiciones en lo que al acceso a dicha información se refiere.
Si tenemos en cuenta que las Tecnologías de la Información y Comunicación (TICs) son determinantes a una sociedad, por ejemplo para el desarrollo de los individuos (oportunidades de formación, acceso a la cultura e información), su falta puede generar desigualdades.

La norma ISO/TC 16027 [2], define Accesibilidad como la facilidad de uso en forma eficiente, eficaz y satisfactoria de un producto, servicio, entorno o instrumento por personas que poseen capacidades diferentes. Por lo tanto, Accesibilidad electrónica hace referencia a que los productos y servicios electrónicos puedan ser utilizados por los usuarios con efectividad, eficiencia y satisfacción en un contexto de uso determinado. Sin embargo, y debido a diferentes motivos, actualmente la mayoría de los sitios Web presentan barreras de Accesibilidad. Por ejemplo, los usuarios discapacitados y las personas de edad avanzada tienen serias dificultades para interactuar con Internet debido a impedimentos físicos y/o cognitivos para manipular los dispositivos y entender los procedimientos y la navegación.

Motivados por esta situación, muchos países ya han legislado para velar por la Accesibilidad de los sitios relacionados con los organismos del estado, es decir aquellos referidos a la administración y provisión de servicios públicos a sus ciudadanos. Así mismo han surgido asociaciones y organizaciones en todo el mundo, que persiguen el objetivo de encaminar la Web a su máximo potencial, lo cual implica una Web para todos: un Acceso Universal. El World Wide Web Consortium (W3C), fue creado en Octubre de 1994 con la visión "Universal Web Access: The Web Anywhere, for Everyone, at Anytime, on Everything". La W3C ha desarrollado guías de Accesibilidad denominadas Web Content Accessibility Guidelines 1.0 (WCAG 1.0) [6] que son consideradas en la Unión Europea como normas de facto, y citadas como referencia obligada en la mayoría de las legislaciones sobre Tecnologías de la Información y Comunicación de todo el mundo. Estas recomendaciones explican cómo hacer Accesible el contenido de la Web a personas con discapacidad y están pensadas tanto para los desarrolladores de contenido (autores de páginas y diseñadores de sitios) como para los desarrolladores de herramientas de autor. Basados en estas recomendaciones un gran número de enfoques han surgido en los últimos años y están disponibles para asistir a los desarrolladores Web en la generación de contenido Accesible. En [4] se presentan y comparan quince de estos enfoques encontrados en la literatura reciente en un framework de evaluación denominado WAAM "Web Accessibility Assessment Model".

La Accesibilidad Web se refiere a cómo se debe codificar y presentar la información cuando se diseña un sitio Web para que las personas con capacidades diferentes puedan percibir, entender, navegar e interactuar de forma efectiva con la Web, así como también crear y aportar contenido. Nuestro trabajo se encuadra dentro del área temática Accesibilidad Web. Debido a que es a nivel de interfaz de usuario donde finalmente se manifiestan las barreras de Accesibilidad, proponemos un enfoque para evaluar la Accesibilidad de aplicaciones Web desde la perspectiva de los principios arquitectónicos del diseño de interfaces de usuario. Particularmente nos centramos en presentar un mapeo de las pautas propuestas por la WCAG 1.0 [6] a las clases de decisión de diseño propuestas por Larson en [3] a los efectos de mejorar el desarrollo de aplicaciones Web con interfaces de usuario más Accesibles. De esta manera es posible analizar las ventajas de nuestra propuesta desde uno de los patrones arquitectónicos más ampliamente utilizados en el diseño de interfaces de aplicaciones Web: Model View Controller (MVC) [1].

El resto del trabajo se organiza de la siguiente manera: la Sección2 introduce el contexto de este trabajo ofreciendo para ello una breve presentación a los componentes del mismo: el framework de Larson [3], el patrón MVC y las guías de la WCAG 1.0 [6]. La Sección3 detalla y presenta parte del mapeo propiamente dicho. La Sección4 discute la visualización de las pautas de Accesibilidad de la

WCAG 1.0 [6] en la relación a las clases de decisión de diseño y el patrón arquitectónico MVC para mostrar ventajas de nuestra propuesta. Finalmente la Sección5 presenta las conclusiones y trabajo futuro.

2 INTERFAZ DE USUARIO Y ACCESIBILIDAD

La Accesibilidad aplicada al contenido de Internet se denomina Accesibilidad Web y básicamente se refiere a que las personas con discapacidad puedan acceder y usar la Web. Más específicamente, significa que todas las personas (aun aquellas con limitaciones propias o derivadas del contexto de uso) puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos [5]. Esta definición hace alusión a "las limitaciones propias del individuo" donde las limitaciones no solo son aquellas derivadas de algún tipo de discapacidad, sino también otras tantas como pueden ser el idioma, los conocimientos previos o la experiencia del individuo.
Debido a que la interfaz de usuario es: (i) el principal punto de contacto entre el usuario y la computadora; (ii) la parte de la aplicación Web con la que el usuario interactúa; y (iii) la encargada de informar las posibles acciones a realizar, los cambios producidos y estado actual de la aplicación, es justamente a nivel de interfaz de usuario donde finalmente se manifiestan las barreras de Accesibilidad. Una barrera de Accesibilidad es cualquier condición que dificulte a una persona alcanzar un objetivo cuando está navegando un sitio Web asistido por cierta tecnología. Una de las principales causas que aporta a la presencia de barreras de Accesibilidad, es la manera en que se diseñan las interfaces. Normalmente el diseño se lleva a cabo pensando en que sus usuarios serán personas estándar con todas sus capacidades físicas y cognitivas, lo que deja afuera a las personas con capacidades diferentes o necesidades especiales.

Nuestro trabajo destaca justamente la trascendencia que tiene el diseño de interfaces de usuario accesibles a la universalidad de la Web. Sobre este camino y previo a avanzar sobre nuestro enfoque, a continuación presentamos los fundamentos y principios de diseño que subyacen al mismo.

2.1 Cinco Clases para decidir el Diseño

Desde la perspectiva de los principios arquitectónicos, Larson [3] propone facilitar el complejo proceso de decisión de diseño, dividiendo y agrupando dichas decisiones en diferentes clases. La elección de este framework en particular no es casual y responde a fundamentalmente a dos motivos. El primero es que provee una visión completa de todos los asuntos implicados en el desarrollo de diseños de interfaz de usuario repartiéndolos en diferentes clases lo que le permite a los desarrolladores concentrarse con mayor eficiencia en cada una de estas clases por separado. El segundo es que ofrece un marco de trabajo muy apropiado para tratar dichos asuntos dentro de sus respectivas clases desde la óptica de la Accesibilidad. Así es como un diseño completo surge entonces como la combinación de todas las decisiones de diseño tomadas en cada clase. El framework propuesto por Larson consiste de las siguientes cinco clases de decisión de diseño:

Estructural: En esta clase el analista de la aplicación especifica la estructura del modelo conceptual del usuario final. Esta especificación incluye una descripción de los objetos conceptuales (del dominio de la aplicación) que serán manipulados (consumidos, producidos y/o accedidos) por los usuarios a través de la interfase o funciones de la aplicación. Estas especificaciones también incluyen una descripción de las restricciones y relaciones establecidas entre los objetos conceptuales.

Funcional: En esta clase el analista de la aplicación especifica las funciones (operaciones) que el usuario final puede aplicar a los objetos conceptuales. Se describe en detalle tanto el significado (semántica) de cada función, como las entradas, salidas y posibles errores.

Diálogo: En esta clase el diseñador especifica el contenido y secuencia del intercambio de información entre el usuario y la aplicación, es decir, especifica el estilo del diálogo. En este punto es necesario distinguir los tres tipos de decisiones más importantes que se deben tomar en esta clase y que responden a las siguientes preguntas: (i) ¿cuáles son las unidades de información que serán intercambiadas entre la aplicación y el usuario?; (ii) ¿cómo serán estructuradas en mensajes de intercambio las unidades de información definidas?; y (iii) ¿cual es la secuencia de intercambio entre el usuario y aplicación más apropiada?

Presentación: En esta clase el diseñador de dialogo selecciona los objetos de interacción que componen la interfase de usuario propiamente dicha, su representación visual y disposición respecto a otros objetos. Informalmente, se denominan objetos de interacción a los objetos visible sobre la pantalla o cualquier otro dispositivo de salida, a través del cual el usuario ingresa u obtiene los "tokens" léxicos (por ejemplo son objetos de interacción los menús, íconos, listas de selección, espacios para ingreso de datos, etc.)

Pragmática: Esta clase abarca todas las decisiones que son hardware dependiente. Es decir todos aquellos aspectos relacionados con la gestión del espacio y dispositivos de hardware que serán usados por el usuario para ingresar la información y por la aplicación para devolver los resultados (por ejemplo la selección de una opción deseada con: un dedo de la mano sobre la pantalla; un puntero o lápiz óptico; un movimiento del cursor utilizando joystick, ratón, "track ball" o "keystroke", etc.)

De esta manera el framework se basa un el principio de separación de clases compuestas por decisiones que satisfacen los criterios de completitud, suficiencia, comprensibilidad, independencia y reusabilidad. A continuación presentamos el patrón MVC y cómo se ligan sus principios arquitectónicos al framework de Larson para asistir en el diseño de interfaces robustas y a las necesidades del usuario Web.

2.2 Una Arquitectura para Diseñar con Clases de Decisión

El patrón de diseño "Model View Controller" (MVC) [1] tiene una amplia utilización como guía en el diseño de arquitecturas de aplicaciones que ofrecen una fuerte interactividad con usuarios. Este patrón proporciona soluciones arquitectónicas al problema de separar los aspectos relacionados al código de la aplicación de los aspectos referidos a la interfaz de usuario, permitiendo a los diseñadores incorporar características deseables en dicha interfaz en una etapa temprana del diseño.

El patrón MVC [1] ha sido portado a una gran cantidad de entornos y frameworks tales como WinForms, ASP.Net, etc. Varias herramientas de programación visual como Visual Basic, Visual Studio.Net, etc., también han aplicado alguna variante del esquema del patrón MVC. Desde la perspectiva de diseño de una aplicación Web, MVC separa: (i) los datos de la aplicación, (ii) la interfaz de usuario, y (iii) la lógica de control, respectivamente en los siguientes tres componentes:

Modelo: representa específicamente el dominio de la información sobre la cual funciona la aplicación, es decir encapsula las funcionalidades y datos. El *Modelo* es independiente de cualquier representación de salida y/o comportamiento de entrada.

Vista: presenta al usuario el *Modelo* en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario en una pantalla de un dispositivo de salida. Pueden existir múltiples vistas del *Modelo* y cada una tener asociado un componente controlador.

Controlador: responde a eventos, generalmente acciones del usuario, e invoca cambios en el *Modelo* y probablemente en la *Vista*.

En la Figura 1, se puede apreciar que la principal característica del patrón MVC [1] es aislar la *Vista* del *Modelo*, separando las responsabilidades. El *Modelo*, representa únicamente el estado y lógica

de la aplicación y no tiene relación en como el estado es presentado al usuario o como la información e instrucciones ingresada por el usuario son recibidas. La *Vista*, en cambio, tiene la responsabilidad únicamente de la creación de la interfase de usuario, en respuesta a las actualizaciones genéricas recibidas desde el *Modelo*. La vista no tiene relación con la lógica de la aplicación o el procesamiento de las entradas ingresadas por el usuario y sólo debe asegurarse que la interfase refleje el estado actual del *Modelo*. Finalmente el componente *Controlador* se ocupa únicamente de trasladar las entradas del usuario (provistas por la *Vista*) en actualizaciones enviadas al *Modelo*. En este esquema las vistas y los controladores conforman la interfaz de usuario. Un mecanismo de propagación de cambios asegura la consistencia entre la interfaz y el *Modelo*. La separación del *Modelo* de los componentes *Vista* y *Controlador* permite tener múltiples representaciones de vistas para un mismo *Modelo*. Si el usuario cambia el *Modelo* a través del controlador de una *Vista*, todas las otras vistas dependientes deben reflejar los cambios. Por lo tanto, el modelo notifica a todas las *Vistas* siempre que sus datos cambien. Las *Vistas* recuperan los nuevos datos del *Modelo* y actualizan la información que muestran al usuario.

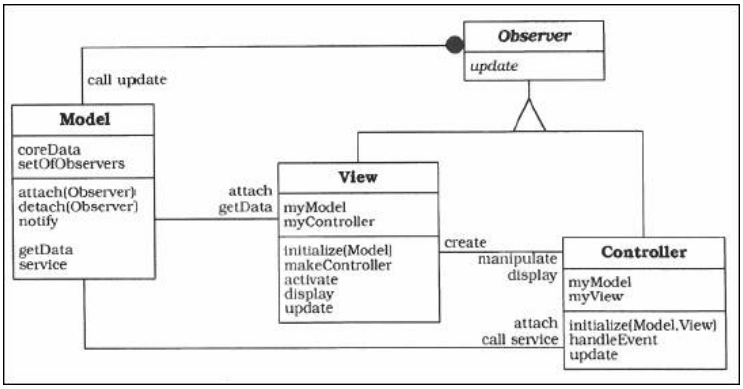

Figura 1: Diagrama de Clases del MVC [1]

Ahora bien, es importante en este punto describir al patrón MVC [1] desde una visión arquitectónica y en relación al framework de clases de decisiones de diseño de interfaz propuesto por Larson. La razón en la que se basa esta asociación es lograr un framework arquitectónico sólido donde discutir eficientemente el diseño de interfaces de usuarios Accesibles. Ya señalamos en la Sección 2.1 cuales son las ventajas que aporta el enfoque de Larson a este propósito. MVC por su parte provee un paradigma arquitectónico, cuya filosofía conceptual es totalmente aplicable a las plataformas de desarrollo y entornos inter-organizacionales actuales. Así, el objetivo buscado en esta asociación entre MVC y las clases de decisión de diseño de Larson es especificar: (i) la estructura fundamental de la interfaz de usuario de una aplicación Web; (ii) el conjunto de componentes predefinidos; y (iii) las responsabilidades, reglas y guías para organizar las relaciones entre estos componentes. Entonces, revisando la arquitectura de alto nivel del MVC podemos asociar las responsabilidades de sus tres componentes principales a las clases del framework de Larson de la siguiente manera:

- *Modelo:* contiene los algoritmos computacionales necesarios para dar soporte a las clases de decisión de diseño *Estructural* y *Funcional*.
- *Vista:* contiene los algoritmos y heurísticas relacionadas a los aspectos de visualización de información de la clase de decisión de diseño *Presentación* (“Outputs”)
- *Controlador:* contiene los algoritmos y heurísticas relacionadas al ingreso de información por parte del usuario, en relación a las decisiones correspondientes a la clase de *Presentación* y todos los aspectos vinculados a la clase de decisión de diseño *Diálogo* (“Inputs”)

Como se puede observar, la clase de decisión de diseño *Presentación* se divide en dos subclases: *vista de la Presentación* que abarca los aspectos relacionados a la visualización de la información y *control de la Presentación* que abarca los aspectos del ingreso de la información. Esta última subclase se combina a la clase de decisión de diseño *Diálogo* para formar el componente *Controlador*.

Para completar la descripción de los fundamentos y principios de diseño que subyacen a nuestro enfoque, a continuación ofrecemos un breve resumen de las guías de la WCAG 1.0 desarrolladas por la W3C, organismo reconocido mundialmente como referente de la Accesibilidad Web.

2.3 Accesibilidad Web definida por la W3C

Dentro de los tipos de guías redactados por la W3C se encuentran las "Web Content Accessibility Guidelines 1.0" (WCAG 1.0) [6], que tratan de un conjunto de recomendaciones para que las páginas Web sean accesibles para todos con la ayuda de la tecnología existente. Estas pautas no descartan los contenidos multimediales, solo apuntan a que la información sea accesible a todos los usuarios Web que así lo requieran. La WCAG 1.0 establece diferentes puntos de verificación a satisfacer con distinto grado de importancia (o nivel de prioridad) que se fija de acuerdo al impacto (barrera de Accesibilidad) al que enfrenta a los posibles usuarios Web. Cada punto de verificación tiene asignado uno de los siguientes tres niveles de prioridad:

- Prioridad 1: puntos de verificación que el desarrollador *tiene* que satisfacer porque sino algunos grupos de personas serán incapaces de acceder a la información de un sitio (16 puntos de verificación tienen Prioridad 1)
- Prioridad 2: puntos de verificación que el desarrollador *debe* satisfacer porque sino alguien encontrará muchas dificultades para acceder a la información (30 puntos de verificación tienen Prioridad 2)
- Prioridad 3: puntos de verificación que el desarrollador *puede* satisfacer porque de lo contrario algunas personas hallarán dificultades en su acceso (19 puntos de verificación tienen Prioridad 3)
- Directamente relacionado con los niveles de conformidad o adecuación con los puntos de verificación, la W3C estableció los logotipos de conformidad con las directrices de Accesibilidad para el contenido Web.

3 EVALUACIÓN DE ACCESIBILIDAD BASADA EN LA ARQUITECTURA

Nuestro trabajo propone un enfoque para evaluar la Accesibilidad de aplicaciones Web desde la perspectiva de los principios arquitectónicos para el diseño de interfaces de usuario. Básicamente la actividad concentra sus esfuerzos en mapear cada una de las pautas correspondientes a la WCAG 1.0 a las clases de decisión de diseño propuestas por el framework de Larson, con el objeto de contribuir desde las primeras etapas del ciclo de vida al desarrollo de aplicaciones Web con interfaces de usuario más Accesibles.

Para desarrollar este mapeo, consideramos las 14 pautas de las WCAG 1.0 [6] y las clases de decisión de diseño *Diálogo*, *Presentación* y *Pragmática* de Larson. Cabe señalar que sólo se consideran para el mapeo estas tres clases del framework, debido a que se relacionan directamente con la interacción usuario-sistema donde se manifiestan las barreras de Accesibilidad. Las clases de decisión de diseño *Estructural* y *Funcional* se corresponden con las decisiones de diseño asociadas a la representación específica del dominio de la información sobre la cual funciona la aplicación Web, es decir encapsulan su lógica (funcionalidades y datos).

A continuación explicamos algunos conceptos que fundamentan el trabajo realizado sobre cada una de las 14 pautas de la WCAG 1.0 [6] durante el proceso de mapeo.

3.1 Arquitectura con Clases de Decisión Accesibles

Cada una de las pautas de la WCAG 1.0 [6] enfoca en un objetivo de Accesibilidad y se compone de un conjunto de puntos de verificación. Cada punto de verificación tiene a su vez su propio objetivo de Accesibilidad destinado a consolidar el objetivo de Accesibilidad de la pauta a la que pertenece. Dada una página Web, asegurar la conformidad para un determinado punto de verificación significa evitar la presencia de barreras de Accesibilidad para determinados usuarios Web. El grado del aporte a la Accesibilidad de la página Web que proporciona la conformidad de un determinado punto de verificación, se refleja en el nivel de prioridad que tiene asignado. Estos conceptos que fundamentan la filosofía de las guías de la WCAG 1.0, se utilizaron para trabajar individualmente sobre cada una da las pautas a través de sus respectivos puntos de verificación.

Tabla 1: Entradas definidas para preservar la filosofía de la WCAG 1.0

Nivel de Prioridad:		
Pauta	**Punto de Verificación**	**Clase de Decisión de Diseño Componente MVC**
Ejemplo		

Ahora bien, preservar intacta esta filosofía de la WCAG 1.0 [6] es un requerimiento que establecimos como prioritario al mapeo sobre las clases de decisión de diseño de Larson. Es fundamental a nuestro enfoque que un punto de verificación mapeado a su clase de decisión siga reflejando su objetivo de Accesibilidad y nivel de prioridad. La preservación de esta información inherente a cada punto de verificación permitirá a los diseñadores conocer la relevancia y determinar las consecuencias (desde el punto de vista de la Accesibilidad) de tomar un conjunto de decisiones de diseño al desarrollo de una interfaz de usuario con determinado nivel de conformidad. La Tabla 1 muestra las entradas definidas para guiar el proceso de mapeo a los efectos de preservar en forma clara los contenidos inherentes a la filosofía de WCAG 1.0, esenciales a nuestro enfoque. Tal como podemos observar, la entrada denominada *Nivel de Prioridad* permite agrupar el análisis de los puntos de verificación que comparten el mismo nivel de prioridad. La segunda entrada provee tres columnas. Una entrada en la segunda columna denominada *Punto de Verificación* identifica cada punto de verificación por su número y objetivo de Accesibilidad. Una entrada en la primera columna denominada *Pauta* identifica cada pauta por su número y objetivo de Accesibilidad y la asocia a sus respectivos puntos de verificación del nivel de prioridad que esta bajo análisis. Una entrada en la tercera columna denominada *Clase de Decisión de Diseño-Componente MVC* indica la clase y a través de ella el componente a la que se corresponde al punto de verificación situado en la segunda columna concretando el mapeo. Finalmente, la entrada denominada *Ejemplo* provee tópicos de cómo implementar el punto de verificación utilizando ejemplos bien formados en HTML. Estas entradas propuestas en la Tabla 1 permiten a los diseñadores tener presentes de manera sintética y clara cuáles son los principios que subyacen al diseño de interfaces de usuario accesibles y cómo se vinculan estos con las clases de decisiones de diseño y el componente MVC [1] respectivo para sustentar dicha Accesibilidad.

Remitiéndonos específicamente al proceso de mapeo, para determinar la existencia de una correspondencia entre un punto de verificación de la WCAG 1.0 [6] y una clase de decisión de diseño de Larson ligada a un componente MVC [1], fijamos y detallamos los siguientes criterios:

- El conjunto de decisiones que se toman para implementar cada punto de verificación es lo suficientemente significativo para producir un efecto sobre el diseño de la interfaz de usuario y justificar su correspondencia con alguna clase de decisión diseño ligada a un componente MVC.

- Todos los puntos de verificación deberían tener una correlación con alguna clase de decisión de diseño y a través de ésta con un componente MVC.
- Para establece una correspondencia de un punto de verificación con la clase de decisión de diseño *Diálogo* y a través de ésta con el componente MVC *Controlador*, la implementación de dicho punto de verificación determina: (i) las unidades semánticas de información que serán intercambiadas en la interacción usuario-aplicación; (ii) la estructura de estas unidades en mensajes para su intercambio; y (iii) la secuencia y el tiempo en que el usuario es forzado a intercambiar estas unidades con la aplicación.
- Para establecer una correspondencia de un punto de verificación con la clase de decisión de diseño *Presentación* y a través de sus subclases *vista de Presentación* y *control de Presentación* con los componentes MVC *Vista y Controlador* respectivamente, la implementación de dicho punto de verificación determina aspectos puramente gráficos o de cosmética de apariencia "look-and-feel" de la interfaz de usuario. Estas decisiones están relacionadas directamente con la representación visual de los objetos de presentación y su disposición respecto de otros objetos sobre el mismo dispositivo.
- Para establecer una correspondencia de un punto de verificación con la clase de decisión de diseño *Pragmática*, la implementación de dicho punto de verificación implica decisiones hardware dependiente, es decir aspectos relacionados a la gestión de espacio y dispositivos de hardware que serán usados por el usuario para ingresar la información y por la aplicación para devolver los resultados.

En el marco de nuestro enfoque, el grado de Accesibilidad alcanzado por una interfaz de usuario estará determinado por el nivel de conformidad que reflejen las decisiones de diseño tomadas dentro del framework clases de Larson-componentes MVC. A continuación, en la Sección 3.2 damos paso al mapeo propiamente dicho presentando parte del mismo.

3.2 Correspondencias Accesibilidad-Decisiones de Diseño

A continuación presentamos el resultado del proceso de mapeo de las pautas de WCAG 1.0 [6]; por razones de espacio adjuntamos solo el mapeo de seis puntos de verificación de Prioridad 1 correspondientes a las pautas 1, 4, 5, 6 y 12; dos puntos de verificación de Prioridad 2 correspondiente a las pautas 13 y 3; y un punto de verificación de Prioridad 3 correspondiente a la pauta 5.

Nivel de Prioridad: 1		
Pauta	**Punto de Verificación**	**Clase de Decisión de Diseño Componente MVC**
Pauta 1: *Proporcione alternativas equivalentes para el contenido visual y auditivo.*	**1.1** Proporcione un texto equivalente para todo elemento no textual (p.e. a través de "alt", "longdesc" o en el contenido del elemento). Esto incluye: imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (p.e. GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del video y videos.	*Presentación* *Vista*
	1.4 Para toda presentación multimedia tiempo dependiente (p.e. una película o animación) sincronice alternativas equivalentes (p.e. subtítulos o descripciones de la banda de visual) con la presentación.	*Presentación*, *Diálogo* y *Pragmática* *Vista* y *Control*
Ejemplo 1.1 En HTML una forma de proporcionar texto equivalente, es con el uso del atributo "alt" para los elementos IMG, INPUT y APPLET, "content" para los elementos OBJECT y APPLET y si se requiere contenido complejo "longdesc" para los elementos IMG o FRAME. *<IMG src="lupa.jpg" width="50" heigth="100" alt="Botón para acceder a la búsqueda">*		

<table>
<tr><td colspan="3"><img src="navbarra.jpg" width="500" heigth="80" alt="Barra de Navegación" longdesc="navbarra_desc.html">

Ejemplo 1.4 Algunos formatos de medios (p.e. QuickTime 3.0 y SMIL) permiten añadir archivos sonoros y visuales separados para combinarlos con los archivos de texto a través del archivo de sincronización para crear audio y películas subtitulados. Pueden proporcionarse equivalentes para los sonidos en forma de una frase de texto en la página, que vincula a una trascripción del texto o una descripción del archivo de sonido.
<!-- AUDIO DESCRIPTIONS -->
<audio src="cardesc.rm" systemAudioDesc="on"
systemOverdubOrSubtitle="overdub"/>
<!-- TEXT DESCRIPTIONS -->
<textstream src="cardescap.rt" systemAudioDesc="on"
systemCaptions="on"/></td></tr>
<tr><td>Pauta 9: Diseñe para la independencia del dispositivo</td><td>9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.</td><td>Vista y Control.
Presentación, Diálogo y Pragmática</td></tr>
<tr><td colspan="3">Ejemplo 9.1 Las zonas de la imagen que pueden ser definidas en forma geométrica serán proporcionadas por un mapa de imagen controlado por el cliente. En HTML indicando cada área y tipo:
shape="RECT" . Crea un área rectangular.
<area shape="RECT" coords="X1,Y1,X2,Y2" href="#">
shape="CIRCLE". Crea un área circular.
<area shape="CIRCLE" coords="X1,Y1,R" href="#">
shape="POLY" . Crea área poligonal.
<area shape="POLY" coords=" X1,Y1, X2,Y2, X3,Y3, X4,Y4" href="#">
Las zonas de la imagen que no puedan ser definidas en forma geométrica serán procesadas a través de un script en el servidor.

<img src="mapa1.gif" ismap border="0">

En el archivo mapa1.map están definidas las coordenadas de los enlaces de la imagen sensitiva mapa1.gif. Al hacer un click sobre la imagen, se realiza sobre coordenadas determinadas de la misma, que son enviadas al servidor y en función de estas se consulta el archivo mapa1.map y se halla la dirección final del enlace.</td></tr>
<tr><td>Pauta 6.
Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correcta y apropiadamente</td><td>6.3 Asegúrese de que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, “applets” u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.</td><td>Vista y Control
Presentación y Diálogo</td></tr>
<tr><td colspan="3">Ejemplo 6.3 El elemento <NOSCRIPT> de HTML permite a los diseñadores proporcionar contenido alternativo cuando un script no es ejecutado.
<SCRIPT type="text/tcl">
...un script Tcl para mostrar un resumen de búsquedas realizadas realizada...
</SCRIPT>
<NOSCRIPT>
<P>Históricos de búsquedas realizadas por mes:</P>
<DL>
<DT>ENE 91, FEB 80.
<DD>
<A href="estadistica.html">Estadística de búsquedas por páginas </A>
...mas resultados...
</DL>
</NOSCRIPT>
En el elemento <APPLET>, el atributo <alt> permite incoporar un equivalente textual.
<APPLET code="Press.class" width="500" height="500"
alt="Applet Java: estadística de búsquedas por criterios">
La cantidad de búsquedas realizadas en los últimos meses …
</APPLET></td></tr>
<tr><td>Pauta 12.
Proporcione información de contexto y</td><td>12.1 Titule cada marco para facilitar su identificación y navegación de los mismos.</td><td>Vista
Presentación</td></tr>
</table>

orientación		

Ejemplo 12.1 En HTML los elementos FRAME, IFRAME y FRAMESET disponen de atributos, tales como <title> y <longdesc>, para proveer el nombre y relación entre frames:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<HTML>
<HEAD>
        <TITLE>Las noticias de hoy</TITLE>
</HEAD>
<FRAMESET cols="10%,*,10%">
<FRAMESET rows="20%,*">
    <FRAME src="promo.html" name="promo" title="promociones">
    <FRAME src="barranavega.html" name="barranavega"
    title="Barra de navegación global del sitio"
    longdesc="frameset-desc.html#barranavega">
</FRAMESET>
    <FRAME src="historia.html" name="noticia"
    title="Noticia seleccionada - contenido principal"
        longdesc="frameset-desc.html#noticia">
… continuación de html…
```

El contenido en la página frameset-desc.html podría ser el siguiente:

#barranavega - este marco contiene vínculos a las secciones más importantes de este sitio: Noticias del mundo, Noticias nacionales, Noticias locales, Noticias tecnología, y Noticias del ocio.
#noticia - este marco contiene la historia actualmente seleccionada.

Pauta 5: *En las tablas de datos, identifique los encabezamientos de fila y columna.*	**5.1** En las tablas de datos, identifique los encabezamientos de fila y columna.	*Vista* *Presentación*

Ejemplo 5.1 En HTML utilice las etiquetas <TABLE> para crear la tabla, <TR> para crear cada fila y <TD> para crear cada columna, y el atributo "headers" para especificar una lista de celdas de encabezamiento (etiquetas de fila y columna) asociadas con la celda de datos actual.

```
<TABLE border="1"
summary="Esta tabla esquematiza el número de tazas de café consumidas por cada senador, el tipo de café (descafeinado o normal)y si se ha tomado con azúcar.">
<CAPTION>Tazas de café consumidas por cada senador</CAPTION>
<TR>
    <TH id="header1">Nombre</TH>
    <TH id="header2">Tazas</TH>
    <TH id="header3" abbr="Tipo">Tipo de café</TH>
    <TH id="header4">¿Azúcar?</TH>
<TR>
    <TD headers="header1">T. Sexton</TD>
    <TD headers="header2">10</TD>
    <TD headers="header3">Expreso</TD>
    <TD headers="header4">No</TD> …
```

Nivel de Prioridad: 2		
Pauta 13: *Proporcione mecanismos claros de navegación.*	**13.1** Identifique claramente el objetivo de cada vínculo.	*Vista* *Presentación*

Ejemplo 13.1 En HTML, tanto la etiqueta A como LINK disponen del atributo <title> que permite añadir información sobre la naturaleza de un vínculo.

```
<A href="pagina1.html" title="Ir a la sección 1: Cómo ingresar" >Sección 1</A>
```

Pauta 3: *Utilice marcadores y hojas de estilo y hágalo apropiadamente*	**3.5** Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.	*Vista* *Presentación*

Ejemplo 3.5 En HTML las etiquetas que definen los títulos de sección son <H1>, <H2>, <H3>, <H4>, <H5> y <H6>. La etiqueta <H1> es la de mayor importancia y por tanto se utiliza para definir los titulares de la página. La importancia del resto de etiquetas es descendiente, de forma que la etiqueta <H6> es la que se utiliza para delimitar las secciones menos importantes de la

página:

```
<HEAD>
    <TITLE>Buscador WEB</TITLE>
    <STYLE type="text/css">
    /* Sangra el encabezamiento y el contenido
    siguiente */
                DIV.section2 { margin-left: 5% }
    </STYLE>
</HEAD>
<BODY>
    <H1>Página principal buscador Web</H1>
    ... algún texto aquí ...
    <DIV class="sección2">
                <H2>Búsqueda simple</H2>
                ... texto para esta sección ...
    </DIV>
    <DIV class="sección2">
          <H2>Búsqueda avanzada</H2>
          ... texto para esta sección ...
    </DIV>
```

Nivel de Prioridad: 3		
Pauta 5: *En las tablas de datos, identifique los encabezamientos de fila y columna.*	**5.5.** Proporcione resúmenes de las tablas	*Vista Presentación*

Ejemplo 5.5 En el lenguaje HTML puede realizarse a través del atributo "summary".

```
<TABLE border="1"
summary="Esta tabla esquematiza el número de búsquedas realizadas por cada usuario, la fecha de la búsqueda y si accedióo nó a la página.">
```

4 DISCUSIÓN

A los efectos de poder llevar adelante una discusión sobre las ventajas de nuestro enfoque, en primer lugar debemos recordar por qué la atención de nuestra propuesta se centra en el diseño de interfaces de usuario. Dada una aplicación Web, es a nivel de interfaz de usuario donde definitivamente se manifiestan las barreras de Accesibilidad. Por lo tanto el desarrollador de aplicaciones Web debe diseñar interfaces de usuario que aseguren el acceso a todas las personas independientemente de sus capacidades físicas y cognitivas y del contexto particular de uso. Dicho esto, queda claro que la interfaz de usuario es una pieza fundamental a la Accesibilidad Web. Nuestro enfoque apuesta fuertemente a la etapa de desarrollo de las interfaces de usuario proveyendo un instrumento que atiende simultáneamente a los aspectos esenciales de diseño y de Accesibilidad. La fundamentación de nuestro trabajo está basada en un mapeo de las pautas de la WCAG 1.0 [6] a un framework arquitectónico, que permite a los desarrolladores construir interfaces visualizando el efecto que tendrán sus decisiones de diseño en el producto final.

Ahora bien, para no quedarnos sólo en la predica de las ventajas de nuestro enfoque, aportemos un ejemplo habitual al mundo de los negocios. Con el fenómeno de la globalización la mayoría de las organizaciones en la actualidad llevan a cabo sus negocios dentro de un esquema que normalmente se denomina inter-organizacional donde organizaciones e individuos interactúan en la búsqueda y/o provisión de productos y servicios. En este contexto, aplicamos nuestro trabajo a una empresa del medio, a la que por confidencialidad llamaremos "Petróleo SA". En esta organización, gran parte de las aplicaciones son Web, sobre un estándar corporativo .NET y guías de desarrollo que atienden los conceptos básicos relacionados tanto con la plataforma como con las herramientas de desarrollo. La metodología de trabajo sobre estas herramientas promueve la filosofía de desarrollo en capas: de datos, de lógica de negocio y de *Presentación*, y de buenas prácticas de usabilidad en lo que a la capa de *Presentación* respecta. Dadas las características de la empresa, pero fundamentalmente la

de los usuarios de sus aplicaciones: (i) edades, características físicas, cognitivas y de habla diferentes, (ii) gran distribución geográfica (iii) condiciones de conectividad y de dispositivos diferentes (notebooks, PDAs, celulares etc.), la consideramos un escenario ideal donde nuestro enfoque podía contribuir de manera diferenciada. Fundamentalmente la aplicación administra la información asociada a emisiones e incidentes medio ambientales ocurridos en instalaciones de la compañía, las medidas tomadas y acciones llevadas a cabo, como así también sus respectivas remediaciones. La disponibilidad de esta información tiene entre otros objetos informar a las diferentes autoridades (nacionales, provinciales y municipales), en determinados plazos de tiempo, cumpliendo con las legislaciones establecidas vigentes. Básicamente nuestro enfoque fue puesto en práctica sobre un módulo de la aplicación, para asistir al desarrollo de su interfaz como una *Vista* accesible. Así, los desarrolladores pudieron descubrir el conjunto de especificaciones de Accesibilidad que se requerían anexar a las ya existentes a partir de visualizar los efectos de sus decisiones de diseño en la capa de *Presentación* de dicha *Vista.* Para citar un ejemplo, el enfoque reveló que ninguna de las imágenes dispuestas en la interfaz de usuario proveía texto alternativo (Pauta 1 - Punto de Verificación 1.1) y los efectos que esta carencia creaba en el uso de dicho módulo. Esto dio origen a la especificación: "colocar en cada elemento IMG textos descriptivos en sus atributos <alt> y <longdesc> según corresponda …" y al detalle de cada imagen y texto asociado.

5 CONCLUSIONES Y TRABAJO FUTURO

El desarrollo de aplicaciones Web accesibles es un factor fundamental a la concreción de la universalidad de la Web. La interfaz de usuario es el componente donde se manifiestan los problemas derivados de la ausencia de Accesibilidad. Motivados por esta realidad, desarrollamos nuestro enfoque, que asiste al diseño de interfaces de usuarios con un grado de Accesibilidad que depende directamente del nivel de conformidad que reflejen las decisiones de diseño tomadas dentro del framework clases de Larson-componentes MVC por el desarrollador.

La intencionalidad es que aplicaciones Web con necesidades de dominios específicas basadas en el framework MVC, puedan reusar todo o parte del sistema, aumentando la facilidad de mantenimiento, integración, extensión y modularidad del mismo y disminuyendo los defectos, tiempos y costos incurridos en el proceso de desarrollo dentro del marco que brindan los principios el Accesibilidad. Nuestro trabajo futuro se orienta a medir y evaluar estas cualidades empíricamente.

REFERENCIAS

[1] Buschman, F. Meunier, R. Rohnert, H. Sommerlad, P. and Stal, M. Pattern Oriented Software Architecture: A System of Patterns. John Wiley and Sons, ISBN: 978-0-471-95869-7, 1996.

[2] ISO/TS 16071 International Organization for Standardization/Technical Specification. Ergonomics of Human-System Interaction - Guidance on Accessibility for Human-Computer Interfaces, 2002.

[3] Larson, J. Interactive Software: Tools for Building Interactive User Interfaces. Yourdon Press Computing Series, Prentice Hall, ISBN: 0-13-924044-6, 1992.

[4] Martín, A. Cechich, A and Rossi, G. Comparing Approaches to Web Accessibility Assessment. *Handbook of Research on Web Information Systems Quality*, Idea Group Inc., C. Calero, Mª Á. Moraga, M. Piattini (Eds.), ISSN: 0950-5849, 2008.

[5] W3C Web Accessibility Initiative (WAI). Introduction to Web Accessibility. Available at http://www.w3.org/WAI/intro/accessibility.php.

[6] WCAG 1.0 Web Content Accessibility Guidelines 1.0. World Wide Web Consortium (W3C) Recommendations, from http://www.w3.org/TR/WAI-WEBCONTENT/, 1999.