Modelo ontológico de navegación en la red para personas con discapacidad visual

Julián A. Gil⁽¹⁾, Paulo A. Gaona-García^{(1)*}, Carlos E. Montenegro-Marin⁽¹⁾, y Adriana C. Gómez-Acosta⁽²⁾

(1) Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas. Bogotá D.C – Colombia.

(correo-e: jgilsanto@correo.udistrital.edu.co, pagaonag@udistrital.edu.co, cemontenegrom@udistrital.edu.co)

(2) Facultad de Ingeniería, Fundación Universitaria San Mateo, Bogotá D.C – Colombia

(correo-e: agomez@funсанmateo.edu.co)

*Autor a quien debe ser dirigida la correspondencia

Recibido Sep. 23, 2019; Aceptado Nov. 6, 2019; Versión final Dic. 23, 2019, Publicado Abr. 2020

Resumen

Se presenta el diseño de un modelo ontológico para mejorar la accesibilidad a la web de personas no videntes, mediante la implementación de la norma W3C (World Wide Web Consortium) y la Norma Técnica Colombiana NTC 5854. El modelo ontológico se desarrolló utilizando herramientas semánticas web y nuevas tecnologías de diseño web, que están conformadas por cuatro componentes principales: (1) la creación de un esquema de datos representativos, (2) la integración de la herramienta de desarrollo y ontología, (3) la visualización de cada uno de los elementos web etiquetados y la visualización de metadatos semánticos, y (4) validación y proceso de traducción/compilación. Los resultados permitieron identificar aspectos funcionales que facilitaron la generación de un esquema de desarrollo para crear un sitio web accesible para personas no videntes. Se presentan recomendaciones a diseñadores y programadores web para proporcionar estrategias en la creación de productos accesibles para personas no videntes.

Palabras clave: modelo ontológico; accesibilidad; navegación web; discapacidad visual

Ontological model of web navigation for people with visual disabilities

Abstract

This paper presents the design of an ontological model to improve the web accessibility to blind people, through the use of the World Wide Web Consortium standard (W3C) and the Colombian Technical Standard NTC 5854. The ontological model was developed using web semantic tools and new web design technologies, which contains four main components: (1) the creation of representational data schema, (2) the integration of the ontology and the development tool, (3) the visualization of each labelled web elements and visualization of semantic metadata, and (4) validation and translation/compilation process. Preliminary results facilitated the generation of a clear scheme to create a website accessible to blind people. The results allowed identifying functional aspects that facilitated the generation of a development scheme to create a website accessible to blind people. Recommendations are presented to web designers and programmers, to provide strategies for creating products accessible to blind people.

Keywords: ontology model; accessibility; web navigation; visual disability

INTRODUCCIÓN

La World Wide Web ha tenido importantes cambios en los últimos años convirtiéndola en el principal canal de información en el mundo entero, en donde una persona tiene acceso a 4.56 billones de páginas web indexadas en los principales buscadores (World Wide Web Size., 2017), en donde más de 7000 millones de usuarios comparten mensajes personales e imágenes en Facebook, 600 millones comparten mensajes personales y profesionales en Twitter y 9000 millones de operaciones de búsqueda de información son realizadas por Google a diario (Statistic Brain, 2017). Sin duda, la web se ha convertido en el mayor mercado de servicios, en donde las personas pueden satisfacer cualquier tipo necesidad, pasando por aspectos relacionados con adaptación de aplicaciones para la enseñanza, realizar cualquier tipo de compra, transacciones bancarias, clases online, ideas de negocio, entre muchas otras (Acevedo, 2010). Años atrás no se pensaba en esto, de hecho, este tema era considerado un lujo, no obstante, en la actualidad el acceso a la web se está convirtiendo en un requisito para la plena participación en la sociedad (Ashe y Pani, 2013).

El acceso a la información abre grandes oportunidades para potenciar gustos, ideas de negocio y talentos. De esta manera es importante hacer partícipe a las personas discapacitadas que para nuestro caso particular se centra en personas ciegas dentro de cualquier actividad tecnológica de la era digital. Es así, como las personas ciegas podrían beneficiarse más de estas oportunidades para tener acceso a múltiples recursos disponibles que brinda actualmente la web, gracias a estándares y métodos que facilitan la accesibilidad (Shawar, 2015). Colombia de las 48'203.405 personas descritas en la proyección poblacional del 2015 por parte de DANE junto con el Ministerio de Salud, el 2,45% cuentan con alguna discapacidad (Ministerio de Salud y Protección Social, 2015). De este grupo de personas con alguna discapacidad se estima que el 2.77% tienen limitación para ver, de donde este porcentaje solamente el 10 % domina el sistema Braille, y lamentablemente en jóvenes menores de 26 años esta aproximación baja el 5% (DANE, 2006). He aquí un problema de acceso a la información para personas ciegas.

Bajo este panorama, se han establecido diferentes instituciones que regulan y establecen normas para que la industria desarrolladora de software cumpla las pautas correspondientes sobre accesibilidad web para personas con algún tipo de discapacidad. En Colombia el ICONTEC estableció la Norma Técnica Colombiana (NTC) 5854 (Colombia, 2019) que establece los requisitos de accesibilidad aplicables a las páginas web. La intención de esta norma es promover contenido sobre la web accesible a personas con discapacidades visuales (entre otras). De igual manera el Ministerio de las Tecnologías de Información y comunicación (MinTic) en Colombia, estableció una iniciativa llamada Convertic, la cual busca el uso y aprobación de las TIC a través de software que les permita a las personas con discapacidad visual tener acceso a la información (por ejemplo: JAWS, como lector de pantalla y MAGIC, software que amplía el tamaño de las letras). Bajo estos lineamientos, el presente documento presenta un modelo que permita fusionar herramientas de web Semántica, estándares y recomendaciones de institutos como ICONTEC y la W3C mediante la implementación de nuevas tecnologías de desarrollo web. Lo anterior, con el propósito de construir páginas web etiquetadas y con sentido semántico para que puedan ser reproducidas por lectores de pantalla o sintetizadores de voz y mejorar la accesibilidad web sobre personas ciegas.

En el trabajo titulado “Marco Ontológico Dinámico Semántico” (Rodríguez y Aguilar, 2017) se planteó un modelo llamado MODS el cual permite formalizar e interpretar las consultas realizada por los usuarios en lenguaje natural. Para ello, MODS transforma la consulta en un formato de Representación del Significado de la Consulta (RSC), el cual está basado en un lenguaje ontológico (OWL) para poder ser usado por la web Semántica, utilizando los diferentes componentes que lo integran. La consulta para el MODS, más que una petición de información, es un elemento cargado de información útil para formarse una idea del tipo de usuario y aproximarse, de manera sucesiva, a una respuesta que cada vez más satisfaga las necesidades del mismo.

Por su parte en el trabajo “Modelo de Interfaz Adaptativa basada en Perfiles de Usuario y Ontologías para Recomendación de Objetos de Aprendizaje” (Quiroz, Salazar y Ovalle, 2018) los autores plantearon un modelo de interfaz gráfica que se adapta a las preferencias cognitivas de los usuarios y que es capaz de recomendar objetos de aprendizaje (OAs). La propuesta hace uso de la metodología Methontology (Fernandez-López et al, 1997) (Romano y McDonal, 2011) la cual permite representar los conocimientos del dominio a partir de ontologías, y proporciona las guías para el desarrollo e implementación de la ontología a través de las siguientes etapas: (1) fase de especificación, (2) fase de conceptualización, (3) fase de formalización, (4) fase de implementación y (5) fase de mantenimiento. Las ontologías proporcionan una gama de herramientas que permiten representar el conocimiento (Ovalle et al., 2014; Durán et al., 2016) y especificar un marco de trabajo conceptual (Rodríguez, 2017), en este caso es utilizada para poder inferir cual es la distribución propicia de paneles en la GUI para un usuario específico.

Finalmente en propuesta titulada “Modelo de Recomendación Personalizada en Cursos Virtuales basado en Computación Ubicua y Agentes Inteligentes”, realizado por (Ovalle et al., 2014) el objetivo de este artículo era

proponer un modelo de recomendación personalizada de recursos educativos para cursos virtuales adaptativos que incorpora las bondades de la computación ubicua y de los agentes inteligentes. Por su parte, los Cursos Virtuales Adaptativos (CVAs) son herramientas computacionales capaces de guiar al alumno a lo largo de un dominio particular del conocimiento. Cabe señalar que los contenidos educativos más utilizados en la actualidad en el marco de los CVAs son los Objetos de Aprendizaje (OAs), que se definen como entidades auto descritas mediante metadatos bien definidos cuyo principal objetivo es permitir su almacenaje y recuperación por tantos usuarios como sea posible, con el fin de realizar sistemas de recomendación de OAs se utiliza el concepto de ontología la cual trabaja sobre el dominio específico de conocimientos del CVA y aplica un método para obtener una representación formal de los conceptos del dominio y las relaciones que existen entre estos.

MODELO ONTOLÓGICO

El objetivo de este apartado es proveer un modelo ontológico basado en especificaciones (Fernández-López, 1997; Guzmán, 2012) con el propósito de diseñar y construir una página web la cual contendrá etiquetas especiales que darán valor semántico a cada uno de los elementos de la misma. Adicional a eso, el modelo propuesto está diseñado con el fin de facilitar el proceso de incluir metadatos para que los lectores de pantalla puedan reconocerlas y traducirlas a una persona ciega, proceso que normalmente no suele hacerse dentro del desarrollo de una aplicación web. Además de eso, para los programadores, la accesibilidad suele ser un proceso demorado, costoso en tiempo y difícil de aplicar.

Para este propósito, en el primer componente del modelo se desarrolla una ontología que define los conceptos semánticos y relaciones que describen los diferentes elementos de una página web. El segundo componente facilita la integración de la ontología creada con la herramienta de desarrollo web. Para este paso hay que identificar la transcripción de la ontología a un formato que sea entendible y manipulable dentro de la herramienta. El tercer componente transforma el código generado previamente en un documento HTML que se visualiza en el navegador, a su vez, este documento debe contener las etiquetas y metadatos semánticos necesarios para que su posterior lectura. El cuarto componente es el que está encargado de la validación de los metadatos y etiquetas embebidas, y debe validar que correspondan a los estándares de la WCGA. Finalmente, el quinto componente describe la integración de un software externo que interprete y traduzca la página web al usuario ciego.

Arquitectura del modelo

La arquitectura del modelo es presentada en la Fig. 1, y consta de cuatro componentes principales: Modelo ontológico, Módulo de integración y creación de componentes web, Renderizado de los componentes HTML, y Validación e Interpretación.

Fig. 1: Arquitectura del modelo planteado.

Modelo ontológico

Este primer componente del modelo propuesto describe la creación de una Ontología que contiene la información de los elementos de la página web y las relaciones entre ellos. Esta información se usa para adaptar la página web, enriqueciendo semánticamente cada elemento que la compone, con la información almacenada y registrada en la ontología. Para ello es usada la metodología *Ontology Development 101*.

Módulo de integración y creación de componentes web

El módulo de integración es el encargado de relacionar el modelo ontológico con la herramienta de desarrollo web (React.js). Es decir, el modelo ontológico es exportado en un formato estándar (formato XML y convertido a json), para que React.js la implemente, haciendo referencia a cada uno de los conceptos y relaciones que la componen. El formato JSON es un formato de texto ligero e ideal para el intercambio de datos, por lo que su uso es esencial en la descripción del modelo ontológico dentro de la librería de desarrollo. En este paso, la página web se modulariza en componentes, y cada componente es un elemento HTML, por ejemplo: Contenedores, Barra de Navegación, Botones, Tablas, Campos de Texto, etc. El objetivo es agregar anotaciones semánticas a cada uno de los componentes. A su vez, éstas etiquetas tienen relación con los conceptos de la ontología, por ejemplo: Si el desarrollador quiere agregar un botón (button) en la interfaz, dentro de la ontología deber a estar un concepto con el mismo nombre "button" y este concepto deberá tener atributos, como por ejemplo id, class, type y un text.

Renderizado de los componentes HTML

React.js, junto con otras herramientas (que ayudan a su compilación y ejecución), crean un documento HTML que se visualiza en el navegador y dentro de él, una estructura ordenada jerárquicamente en forma de árbol, llamado DOM. En este punto cada elemento HTML (componente web en React.js) debe tener las etiquetas o metadatos específicos, que están descritos en la Ontología y que fueron relacionados y cargados en el módulo anterior: Módulo de Integración y Creación de Componentes web.

Validación e Interpretación

Este punto dentro del modelo propuesto es muy importante ya que en este paso se evidencia si el enmarcado y uso de metadatos fue implementado acorde con las pautas internacionales de accesibilidad de contenido en la web del World Wide Web Consortium (W3C). Para ello se usan herramientas y extensiones online, éstas, validan automáticamente la accesibilidad de la página, señalando, errores, advertencias y puntos para revisar manualmente. Estas herramientas están destinadas tanto a usuarios sin experiencia como para profesionales de campo como webmasters, desarrolladores y diseñadores de páginas web.

Desarrollo del modelo ontológico

El desarrollo del modelo ontológico se basa principalmente en la creación de la ontología que contiene el modelo conceptual de los elementos HTML que componen una página web. Se orienta a representar la información y relaciones necesarias para hacerla accesible. La referencia que se hace a la ontología es para añadir etiquetas y metadatos semánticos a la página web. Esta referencia es creada manualmente con ayuda de la herramienta Protégé y se actualizará dependiendo de los criterios del programador (Stanford University, 2016). La ontología es escrita usando el lenguaje OWL e incluye la descripción de todos los elementos de la página web, como (botones, campos de entrada de texto, imágenes, enlaces, etc), a su vez algunas propiedades de tipo son incluidas para proveer una mayor descripción a cada elemento, como (title, id, class y propiedades ARIA y Dublin Core). Como lo exponen (Noy y McGuinness, 2001; Katifori et al., 2007) no hay una vía correcta para desarrollar una ontología. Es así como a continuación se describe en detalle cada una de las fases y la creación de la ontología en el modelo propuesto.

Determinación del dominio y alcance de la ontología

Teniendo en cuenta que las páginas web son representadas por medio del lenguaje de marcado HTML, cuya sintaxis y estructura es transformada por el DOM (Document Object Model) en un árbol de nodos el cual el navegador web puede comprender y mostrar en pantalla, se distingue como dominio de la ontología el Modelo Objeto del Documento - DOM. El alcance de la ontología está definido por los elementos principales que pueden conformar una página web y que están dentro del árbol de nodos. De esta forma se establece el límite de profundización que contendrá la jerarquía de la ontología y a la vez el alcance de la misma.

Definición de la ontología

A la hora de reconsiderar estándares de tipo ontológico en el ámbito web, se siguió una parte de la estructura de la ontología @dapt++ que proponen (Sorrentino et al., 2017) de la cual se toma como referencia la estructura que hace referencia a la clase de la ontología, que describe a los elementos web. Esta ontología fue creada para identificar y priorizar los elementos de una página web, para luego tratarlos y darles significado semántico incluyendo metadatos. A su vez (Sosa, Sánchez, Macías, & Lozano-Tello, 2003) describen la ontología Ontoblind que como la anterior, también contiene el modelo conceptual de los elementos que componen las páginas web que es usada por otras dos aplicaciones dentro de su arquitectura llamada KAIKO

para anotar páginas web. De esta manera el modelo ontológico propuesto no reusa una ontología existente y previamente desarrollada, pero si toma como base la estructura e implementación de las ontologías previamente descritas.

Enumeración de Términos Importantes de la Ontología

Es importante hacer una lista de los términos sobre los cuales se van a hacer declaraciones de conceptos, relaciones, propiedades a usar, clases y otras más para que futuros lectores e investigadores reconozcan fácilmente los términos usados. Entre los más importante encontramos Tabla 1.

Tabla 1: Listado de términos a tener en cuenta para la creación de la ontología.

<i>Elementos HTML</i>	<i>Términos Ontológicos</i>	<i>Propiedades</i>	<i>Relaciones</i>
Botones Formularios Listas Encabezados Enlaces Imagen Pie de pagina Sección, Artículo Entrada de texto Texto, DOM	Clases Subclases Propiedades de datos Propiedades de objeto Instancias Pre jos OWL RDF	Dublín Core Propiedades WAI-AIRA HTML Propiedades generales	Compuesto por Contiene Es parte de Subclase de.

Después de identificar los elementos básicos HTML y de conocer el ambiente de trabajo de la herramienta a usar (Protégé), el primer paso es agregar las clases dentro de nuestro modelo Ontológico en orden descendente, tales como: el header, la barra de navegación nav, y el elemento principal que agrupa el contenido, llamado main. Dentro del elemento main, se puede encontrar nuevos elementos incorporados (a su vez agregando nuevos niveles dentro de la jerarquía) tales como: article, párrafos p, formularios form, encabezados hn (h1 ... h6) y el elemento aside. Esto ayuda a entender cuáles son los elementos HTML o componentes web a usar, siendo cada uno de ellos un nuevo concepto dentro del modelo ontológico. La Fig. 2 muestra un desglose de los diferentes niveles de la jerarquía dentro del modelo ontológico.

Fig. 2: Jerarquía de clases

Visualización del modelo ontológico

A través del proceso realizado en las fases descritas en la metodología Ontology Development 101, y la construcción de la Ontología a través de Protégé se obtiene una estructura de conocimiento organizada que envuelve todos los términos, propiedades y relaciones correspondientes a los elementos HTML que será usada por la herramienta de diseño web. En la Fig. 3 se muestra una representación gráfica de un fragmento del modelo Ontológico creado. Los cuadros de color verde representan los atributos o “Data Properties” de un elemento HTML en color amarillo aparece el rango o el tipo de dato de cada uno. Por otro lado, los cuadros azules representan las relaciones existentes entre clases o elementos, como lo es el caso de “contain”, “composed by”, “is part of”, entre otros.

Fig. 3: Fragmento del modelo ontológico.

Módulo de integración y componentes web

La comunicación se hace prácticamente como una arquitectura tipo Cliente-Servidor, donde el servidor es la aplicación hecha en Node.js con ayuda del framework Express.js que expone la Ontología en formato XML como un archivo estático, que es expuesto localmente en una ruta o URL específica <http://localhost:3000/ontology>, mientras que el cliente es la aplicación web hecha con React.js y que al ser cargada en el navegador, realiza la petición HTTP automáticamente al servidor web y obtiene el recurso pedido. La Fig. 4 muestra de manera detallada la comunicación entre los dos servidores.

Fig. 4: Arquitectura integración entre React.is y la Ontología HTML.

La anterior figura define los dos servidores creados, el primero (Webpack) que corre la aplicación hecha en React.js y el segundo (servidor web) que consulta la Ontología y la sirve como un archivo estático XML. Todo lo anterior mediante una petición HTTP hecha por el primer servidor web (Aplicación React.js). La Fig. 5, muestra de una manera detallada como está conformado el servidor que corre la aplicación web, es decir el cliente (llamase cliente como un dispositivo dentro de la arquitectura cliente-servidor que consume un recurso). En donde se evidencia que al igual que el servidor HTTP, es ejecutado una aplicación web hecha en Node.js, en donde se sirven dos archivos, el primero, el archivo *index.html* que contiene la estructura básica de la página con metadatos semánticos y que va a hacer referencia al archivo JavaScript que contiene toda la lógica hecha por React.js y es compactada en un solo archivo llamado *bundle.js*.

Fig. 5: Arquitectura integración entre React.js y la Ontología HTML.

Análisis de la ontología en XML y conversión a JSON

XML durante un buen tiempo fue la única opción para compartir datos libremente, por lo cual sigue manteniendo su popularidad, al tener un lenguaje de marcado similar al de HTML. Pero en términos de simplicidad, apertura e interoperabilidad, JSON y XML tienen las mismas ventajas. XML como lenguaje de marcado y JSON formato basado en texto para representar datos estructurados son usados para el intercambio de información entre servicios web y APIs REST. Pero cuando se habla de sencillez, fácil proceso de datos e interoperabilidad, se encuentran algunas diferencias y el formato JSON puede ser más amigable. A continuación, se mencionan algunos puntos claves a analizar. Al tener en cuenta la complejidad de consulta y acceso a la información de la Ontología, se decide hacer la conversión de formato XML a formato JSON, ya que por medio de JavaScript el acceso a los datos a un objeto JSON mediante uso pares clave-valor es mucho más rápido a comparación del método iterativo que se tiene que hacer con un archivo XML (Halpin, 2006). Por lo cual permite obtener con mayor rapidez los atributos y propiedades de las clases de la Ontología que son necesarios como metadatos para marcar los componentes web creados desde el lado de la aplicación cliente, es decir de la aplicación hecha con React.js.

La página web generada por el modelo, presenta una estructura accesible compuesta de acuerdo a recomendaciones realizadas por (Ziener, 2001) por los principales componentes web como: el encabezado principal header, la barra de navegación nav, un formulario form, un campo de texto de entrada input, un botón button, enlaces a, artículos articles, secciones de información sections, un apartado lateral adicional aside y un pie de página footer, cada uno asociado con los atributos de su clase correspondiente y descritos en el modelo ontológico, Figura 6. Por ejemplo, el componente web button, está asociado con la clase input del objeto individual de la ontología y toma de ella los atributos aria-describedby para describir por medio de texto su funcionalidad, el id y el nombre de la clase class. Adicional el componente web está enmarcado dentro de la etiqueta `<input />`, dándole significado semántico para que cualquier dispositivo tercero u ordenador pueda identificar que se trata de un control de formulario. Lo mismo sucede con el componente web de información complementaria representado con la etiqueta HTML aside, que toma de los atributos necesarios como id, class y un role WAI-ARIA (Carreras Montot, 2015) de tipo complementary del objeto individual de la Ontología llamado por el mismo nombre aside. Fig. 6.

Fig. 6: Aplicación web con metadatos y tributos semánticos.

De la misma forma los demás componentes web contienen los metadatos accesibles y semánticos que fueron descritos en la creación de los diferentes objetos individuales dentro de la fase de creación de instancias del modelo ontológico. De esta manera se puede evidenciar una relación directa entre la ontología y cada uno de los elementos web. Prácticamente se puede evidenciar que el renderizado de la página web es el reflejo del esquema representacional hecho en la ontología. De esta manera puede construirse una representación conceptual y gráfica por cada sitio web que un diseñador o programador desee crear. Esto ha permitido llevar a un modelo conceptual como base para entablar relaciones directas entre ontologías que permitan describir las páginas web.

Fig. 7: Ontología en formato json dentro del estado de la aplicación.

La Fig.7 muestra como la ontología es cargada en el estado principal de la aplicación después de haber sido consultada vía HTTP al servidor que la expone. Con esto se logra obtener la ontología almacenada permanentemente dentro de la aplicación, lo cual facilita la consulta a los diferentes objetos individuales que contienen los metadatos específicos para que los componentes web sean cargados semánticamente.

Adicional a eso, también pueden ser consultadas las clases y relaciones de la ontología, al ser ahora un objeto tipo json, por medio de una búsqueda clave-valor (entiendase como la estructura base que conforma un objeto JSON, un identificador “clave” que apunta o se relaciona a un “valor” facilitando la organización de información) se puede acceder a cualquier parámetro que se necesite.

RESULTADOS Y DISCUSIÓN

Los resultados que se obtuvieron se presentan a continuación en las siguientes tres secciones: prueba con WAVE; prueba de usuario, y test de navegación

Prueba con WAVE

Para poder validar la accesibilidad web del prototipo generado usando WAVE, únicamente es necesario instalar Wave en el navegador Google Chrome. Luego de cargar la página web con metadatos semánticos, se habilita la extensión dando doble click sobre su icono. Después de ello WAVE arroja un reporte mostrando diferentes puntos a tener en cuenta, dentro de ellos están: errores, alertas, características accesibles insertadas, elementos HTML accesibles usados, atributos WAI-ARIA (Carreras Montot, 2015) encontrados y errores de contraste en los colores usados. Fig. 8.

Fig. 8: Validación con WAVE Evaluation Tool.

En la Fig. 8 se observa los resultados del test usando Wave, en donde se encuentran: un error de referencia de un atributo WAI-ARIA a un elemento de un formulario tipo `<input/>`; cuatro alertas que describen, dos de los cuales anuncian que los textos no deben estar justificados ya que se puede tener problemas con el lector de pantalla, una descripción de la imagen que está bastante larga y una etiqueta de uno de los elementos del formulario que no está asociada correctamente. Las alertas son parámetros que se pueden tener en cuenta para establecer buenas prácticas como: imagen enlazada con atributo “alt” para describir la imagen y 25 atributos WAI-ARIA (Carreras Montot, 2015) encontrados. Posteriormente, los errores se corrigieron manualmente y se tuvieron en cuenta para los ajustes. Para este caso, la herramienta describió elementos de accesibilidad web implementados tales como: etiquetas semánticas, elementos de estructura semántica, elementos aria, encabezados entre otros.

Prueba de usuario

En esta sección, se presentan los resultados obtenidos de la evaluación del prototipo generado por el modelo propuesto mediante la realización de un test de navegación web con participantes no videntes. De acuerdo con (Nielsen, 2012), las pruebas de usabilidad permiten comprobar la manera en que se relaciona el usuario con discapacidad visual con la interfaz generada por el modelo desarrollado en este trabajo. Las pruebas de usabilidad han permitido mejorar aspectos de navegación a usuarios con el propósito de facilitar acceso a recursos digitales (Gaona-García et al 2016; 2018). Teniendo en cuenta este tipo de pruebas, es necesario

contar con criterios que permitan adaptar la prueba de usabilidad para este tipo de población discapacitada. En ese orden de ideas se tuvo en cuenta aspectos evaluados por (Ribera, Térmens, y García-Martín, 2008) relacionados con: i) familiarización con las herramientas, ii) screening, iii) prueba piloto, iv) espacio de los usuarios y herramientas, v) planificación de la duración, y vi) elección estándar accesibilidad.

i) Familiarización con las herramientas. Antes de empezar a trabajar con los participantes es necesario que el equipo de investigación se habitúe con la tecnología de asistencia correspondiente para entender mejor las acciones de estos y los posibles problemas que se puedan encontrar.

ii) Screening. Es una técnica que consiste en que el equipo de investigación simula la realización de un test usando las tecnologías de asistencia. La recomendación es usar esta técnica con un dominio mínimo de los comandos básicos necesarios para la tarea y probarlos realizando el test en las mismas condiciones que los usuarios (por ejemplo, con la pantalla apagada y sin ratón).

iii) Prueba piloto. Como en la mayoría de los test de usabilidad, idealmente se debe realizar con unos pocos usuarios para detectar situaciones no previstas (Nielsen 2012), necesidades no atendidas o para comprender mejor el funcionamiento de la interacción. Para este caso se hizo la prueba piloto con un participante con discapacidad visual, con el propósito de realizar ajustes. Como resultado de este proceso se logró identificar que no necesariamente la implementación de gran cantidad de metadatos semánticos hacen la aplicación web legible para un usuario con discapacidad, tal vez se pueda caer en el error de que entre más metadatos y atributos sean inyectados, la implementación va a ser accesible. Simplemente se deben identificar el propósito del sitio web e identificar los principales metadatos e implementar. De igual manera la estructura del sitio web es esencial, ya que el lector de pantalla se basará en esta, y si no se usa una jerarquía semántica con las etiquetas HTML5 correspondientes a cada componente o sección de la misma, el usuario puede perderse fácilmente.

iv) Espacio de los usuarios y herramientas. En cualquier caso, para garantizar el éxito de la prueba se debe tener en cuenta aspectos como 1) saber que dispositivos se usan y conocerlos. Como lo menciona (Arnautovic, 2017), no solo porque los usuarios tienen discapacidad visual significa que son expertos manejando lectores de pantalla. 2) garantizar que dispongan de una pantalla o monitor y 3) asegurar que tengan acceso a internet si la prueba es web, si no, identificar los programas necesarios para la prueba.

v) Planificación de la duración. Por último, se debe tener en cuenta que una persona con discapacidad visual tarda entre dos y tres veces más que un usuario sin discapacidad para completar la misma tarea, y los requerimientos cognitivos de mantener la atención concentrada por el oído son superiores a los de la vista.

vi) Elección estándar accesibilidad. Adicionalmente (Hawkes Lewis, 2008) describe la importancia de elegir un estándar de accesibilidad durante la fase de pruebas en un equipo de investigación, para posteriormente analizar y contrastar los resultados obtenidos con los principios y pautas del estándar seleccionado. Para este caso particular, se utilizó las sugerencias definidas por WCAG 2.0 (Roger, 2017; Sidar, 2009) de acuerdo a los siguientes motivos: están diseñadas a partir de las necesidades humanas básicas aplicables a tecnologías diferentes de HTML y CSS; documentan de manera muy esmerada las razones de cada uno de los criterios de conformidad; sugieren técnicas prácticas para satisfacer los criterios de conformidad utilizando tecnologías actuales; garantizan que todas las disposiciones se puedan someter a pruebas, y son un estándar internacional.

Test de navegación

La tabla 2 presenta a los participantes. Son presentados por su género, edad, nivel de experiencia usando lectores de pantalla y actividad u ocupación. Dada la naturaleza del estudio no es necesario que los participantes tuviesen conocimiento sobre ontologías, herramientas de web semántica ni herramientas modernas de desarrollo web. En su lugar se tomó en cuenta el conocimiento que pudiesen tener en el manejo de computadores y en el uso de lectores de pantalla. Teniendo en cuenta recomendaciones realizadas por (Nielsen, 2012; Bevan et al., 2003) donde indican que cualquier prueba de usabilidad puede contar entre 3 y 5 participantes, y dada la naturaleza de este tipo de pruebas, se ha contado con la participación de seis (6) personas para llevar a cabo las pruebas del modelo ontológico planteado. Así mismo se contempló el hecho de que los participantes proceden de una misma condición de discapacidad (población), y son sometidos a las mismas tareas (Sauro, 2010). A los participantes se les provee dos páginas web (mediante un navegador), la primera, visualmente es la misma página web prototipo, pero sin metadatos semánticos y con una estructura simple usando etiquetas HTML genéricas, lo cual hace que sea más compleja de navegar o de reconocer “accesiblemente”, Fig. 9. Mientras que la segunda, es el prototipo generado por el modelo propuesto en el proyecto de investigación la cual incluye metadatos semánticos accesibles, Fig. 10. Esto es con el fin de que

el participante con discapacidad visual pueda reconocer fácilmente y contrastar cuál de las dos contiene metadatos semánticos y cual carece de una estructura accesible.

Tabla 2: Participantes- Test navegación y de accesibilidad web.. Frecuencia es Frecuencia de uso por semana.

Género	Experiencia	Lector	Frecuencia	Edad	Ocupación
Masculino	Intermedio	NVDA	2-3	31	Contador/Deportista
Masculino	Experto	JAWS	Diario	36	Instructor/Lectores de Pantalla
Masculino	Experto	JAWS	Diario	54	Profesor IT
Masculino	Intermedio	JAWS	Diario	29	Actor/Locutor
Masculino	Novato	NVDA	2-3	46	Pensionado/Discapacidad
Masculino	Intermedio	NVDA	Diario	51	Profesor/Antropólogo/Psicólogo

Fig. 9: Página web con estructura semántica y metadatos accesibles.

Fig. 10: Momento de interacción de usuarios no videntes en prueba de navegación

Para analizar la eficacia en la navegación y rendimiento del prototipo, al participante con discapacidad visual (Fig. 10) se le plantean una serie de preguntas las cuales están relacionadas con identificar algunos de los principales elementos HTML, tales como: encabezados, formularios, lista de ítems, barra de navegación, gráficos y párrafos. Estas preguntas estaban divididas en dos secciones: (i) preguntas relacionadas con navegación web y lector de pantalla; y (ii) preguntas generales de accesibilidad web, con el propósito de identificar puntos subjetivos de la interacción que ha tenido el participante con discapacidad visual con sitios web, y que puedan servir para trabajos futuros. Usando el lector de pantalla de su preferencia (JAWS o NVDA) el usuario podía navegar y obtener información relevante como insumo para dar respuesta a las preguntas de la prueba.

Los resultados del estudio de accesibilidad y navegación web aplicado al prototipo generado por el modelo planteado, permitieron evidenciar aspectos relacionados con: i) el uso excesivo de atributos y metadatos semánticos dentro de los elementos HTML podría haber generado confusión en términos de navegación al usuario invidente. Esto posiblemente por la asociación de más atributos y metadatos semánticos para la descripción de un recurso web. En el análisis de resultados se encontró que el uso de atributos WAI-ARIA para describir un elemento HTML que ya usa una etiqueta semántica, hace que las herramientas de pruebas automatizadas arrojen error de redundancia en sus informes de validación, y de igual forma, que el lector de pantalla duplique la lectura para el usuario invidente. ii) no necesariamente al realizar el test de accesibilidad y navegación a una persona invidente significa que entiende y tiene experiencia en el manejo de herramientas de asistencia (lectores de pantalla), por lo que en el plan de pruebas de usuario se debe contemplar el tiempo que sea necesario para hacer una introducción del uso de las herramientas de asistencia, pruebas piloto y screening (Ribera et al., 2008) que faciliten el reconocimiento de situaciones no previstas y necesidades no atendidas o para comprender mejor el funcionamiento de la interacción; y finalmente iii) la falta de estudio sobre la accesibilidad y descripción elementos multimedia como videos publicitarios y de métodos de verificación humano-ordenador tipo Captcha's, que se manifestaron en los resultados obtenidos dentro de las preguntas de tipo general, y que son clave a futuros trabajos de investigación.

A partir de los resultados obtenidos, relacionados con las preguntas de navegación y accesibilidad web se logra identificar que el prototipo generado por el modelo ontológico presenta una relación positiva entre los principios y pautas descritos por la WCAG 2.0 en (Roger, 2017; Sidar, 2009). Se destaca el cumplimiento de parámetros analizados como perceptibilidad, comprensibilidad, operabilidad y robustez, adicional a eso la efectiva evaluación de estructura accesible y el reconocimiento de elementos HTML con atributos semánticos mediante el uso de herramientas automatizadas. Estos resultados guardan relación con los que sostiene (Sorrentino et al., 2017) en su modelo de mejora de accesibilidad web para personas con dificultades visuales, que señala como a partir de un modelo ontológico se puede abstraer los diferentes elementos de una página web, "realizando un modelo conceptual y representacional de la misma" para posteriormente identificar sus principales atributos y metadatos e incorporarlos en el diseño y fase de construcción, dando al usuario invidente una mejor descripción de cada uno de los componentes de la página web.

También Sorrentino destaca la validación de los principios de la WCAG 2.0 definidos en (Roger, 2017; Sidar, 2009), mejorando la proporcionalidad de la cantidad de elementos HTML anotados semánticamente y que están referenciados en la ontología, mejorando la relación entre los parámetros evaluados navegación, estructura semántica y reconocimiento de elementos principales (encabezados, listas, enlaces y controles de formulario). Adicional a ello, propuestas como la de (Sosa et al., 2003) para el uso de ontologías para páginas web permite mejorar la accesibilidad a invidentes, resalta la asociación de la ontología directamente con el contenido HTML haciendo referencia a cada uno de sus instancias individuales para obtener y añadir los metadatos semánticos necesarios. No obstante dentro de la propuesta planteada, faltan todavía elementos que permitan satisfacer la demanda de una población cada vez más absorbida para la cantidad de información que vive la sociedad actualmente, lo cual representa un reto para llegar a este tipo de población y la diversidad de medios y formatos que actualmente disponemos en Internet.

Por otro lado, se puede mencionar que al analizar los resultados de navegación y usabilidad por los usuarios invidentes, se identificó que el uso no medido de los atributos semánticos, puede generar confusión en la interpretación del contenido de la página web, es decir, el sobre exceso de los atributos hace que: 1) las herramientas de pruebas automatizadas identifican redundancia de implementación en un elemento HTML, entre su etiqueta semántica por defecto y la adición de un atributo WAI-ARIA (Carreras Montot, 2015) que describa el objetivo de ese elemento, por ejemplo la etiqueta nav y el uso de del atributo role con valor navigation, debido a que representan lo mismo. 2) la descripción de elementos HTML con labels (otros elementos HTML), dado que, si no se implementa de forma correcta, puede que el lector de pantalla los repita al usuario invidente, haciendo que crea que se trata de dos elementos contiguos, pero tratándose de solo uno.

DISCUSIÓN FINAL

De acuerdo con los resultados obtenidos se considera que trabajar con un modelo ontológico que represente la estructura de una página web dándole significado semántico, es una propuesta innovadora, ya que va de la mano con el propósito de la web semántica con la idea de poder representar contenido web con sentido, y con el objetivo de mejorar internet ampliando la interoperabilidad entre sistemas informáticos usando “agentes inteligentes” (programas en las computadoras que buscan información sin operadores humanos). Para ello hay que resaltar que se debe enfocar esfuerzos en i) en análisis del dominio de la ontología, ya que puede ser abstracto y estrechamente relacionado con los elementos y estructura del cual esté compuesta la página web, ii) la definición de los atributos de clase, como los object y data properties. debe ser relevante y concreta, para concuerde con la información básica que necesita ser incluida en un elemento HTML y iii) la integración visual del modelo ontológico, es significativa para el desarrollador, ya que en ella se puede identificar la el tamaño y rango de cada uno de los elementos y componentes web por la cual está compuesta la página web.

Por otro lado, el diseño del modelo ontológico facilitó el almacenamiento y estructuración flexible de la información relacionada con una página web, permitiendo futuras modificaciones en el modelo ontológico por parte de los involucrados (desarrolladores y diseñadores de páginas web). Cabe resaltar que el componente de integración tiene gran importancia dentro de la arquitectura del modelo planteado, ya que es allí en donde se lleva la representación del modelo ontológico creado a una simple estructura de datos que es fácil de implementar sobre una moderna herramienta de desarrollo web.

Por otro lado, la implementación de una herramienta de desarrollo web como React.js facilita el uso de atributos accesibles y semánticos dentro de elementos HTML y permite al diseñador y desarrollador web: i) manejo de estructuras de representación de conocimiento dentro de recursos web, ii) integración con esquemas de representación de conocimiento, iii) rápida composición y renderizado de componentes web, iv) fácil asignación de atributos y metadatos semánticos y v) vinculación con otras tecnologías web que hacen el desarrollo contenga un alto performance e interacción con el desarrollador, como por ejemplo el uso de JSX que facilita la lectura de código, siendo similar a la usada por etiquetas HTML, el uso de componentes web con estado (en donde se pueden almacenar la información del modelo ontológico haciendo únicamente una petición a la ontología), el ciclo de vida de los componentes que facilita el llamado y borrado de los atributos y metadatos a la ontología y finalmente el rendimiento gracias al DOM virtual. Finalmente, que pueda ser integrada por uno de los entornos JavaScript más usados del momento como lo es Node.js, generando un ambiente de trabajo completo entorno al lenguaje JavaScript.

CONCLUSIONES

De acuerdo al trabajo presentado y a los resultados obtenidos, se pueden plantear las siguientes conclusiones principales:

- 1.- El modelo ontológico propuesto posibilitó la representación de un sitio web bajo una taxonomía. Esto facilitó obtener abstracciones de contenido web jerarquizados y organizados que fácilmente pueden ser entendidos por ordenadores, mejorando su navegación y la relación entre sus componentes.
- 2.- La fase de diseño del modelo ontológico, facilitó el almacenamiento y estructuración flexible de la información relacionada con una página web, permitiendo futuras modificaciones en el modelo ontológico por parte de las personas involucradas en este proceso entre desarrolladores y diseñadores de páginas web.
- 3.- El componente de integración del modelo ontológico propuesto facilitó la publicación y servicio de la ontología como un archivo estático que puede ser consultado por aplicaciones terceras. Lo anterior mediante independiencia entre la construcción del modelo ontológico y la integración con la aplicación web generada por React.js. Además se consolidó una arquitectura cliente servidor, que descentralizó la conexión entre ambas partes y permitió la escalabilidad por parte del servidor ontológico para la realización de futuros end-point's o creación de rutas para consultas específicas a la ontología por medio de peticiones HTTP.
- 4.- El uso de una herramienta de moderna de desarrollo web dentro del modelo ontológico planteado en el trabajo de investigación, fomentó la inclusión de principios y pautas de accesibilidad para personas invidentes dentro del proceso de desarrollo de software y desarrollo de páginas web. Mediante la inclusión de React.js como herramienta para aplicar atributos y metadatos semánticos accesibles en elementos HTML, se creó un eje transversal que podría impactar a programadores y desarrolladores web que buscan cada día implementar nuevas tecnologías y tendencias del mercado, cambiando la filosofía tediosa y costosa que se tiene acerca de aplicar estándares de accesibilidad.

5.- Las pruebas de usuario demostraron que los archivos multimedia (tales como imágenes, audio y video) no son tomados en cuenta correctamente y la falta descripción con etiquetas y metadatos semánticos los hace inaccesibles. Normalmente los usuarios pueden encontrar como banner's publicitarios, muchas veces interrumpiendo su flujo, lectura, navegación o perder el atributo *autofoco* dentro de un formulario web.

REFERENCIAS

- Acevedo, C. P., Arciniegas, J. L., García, X., y Perrinet, J., *Proceso de Adaptación de una Aplicación de e-aprendizaje a t-aprendizaje*, <http://dx.doi.org/10.4067/S0718-07642010000600005>, Información tecnológica, 21(6), 27-36. (2010).
- Arnautovic, V., *Four things we learnt from facilitating usability testing sessions with blind users*. Obtenido de Medium corporation: <https://medium.com/seek-blog/four-things-we-learned-from-facilitating-usability-testing-sessions-with-blind-users-2298dac58ae2>. [Última consulta: Junio 30 2019] (2017).
- Ashe, S. S., y Pani, S., *A Review of Trends in Research on Web Accessibility*, International Journal of Engineering Research and Development 8(1), 34–37 (2013).
- Bevan, N., Barnum, C., Cockton, G., Nielsen, J., y otros., *The magic number 5: is it enough for web testing?*, In CHI'03 extended abstracts on Human factors in computing systems, 698-699, ACM (2003).
- Carreras M.O., *Usable Accesible*. Obtenido de WAI-ARIA. Introducción, referencias, ejemplos, herramientas: <https://olgacarreras.blogspot.com/2007/09/wai-aria-introduccion-referencias.html> [Última consulta: Junio 30 2019] (2015).
- Colombia NTC., *Introducción a la norma ntc 5854*. [Última consulta: Junio 30 2019] <https://ntc5854.accesibilidadweb.co/>, (2019).
- DANE. *Discapacidad personas con limitaciones permanentes*, Bogotá D.C. <https://www.dane.gov.co/index.php/estadisticas-por-tema/demografia-y-poblacion/discapacidad>, [Última consulta: Junio 30 2019] (2006).
- Durán, E. B., Álvarez, M. M., y Únzaga, S. I., *Ontological model for the personalization of u-learning applications*. In 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), 1-5. IEEE (2016).
- Fernández-López, M., Gómez-Pérez, A., y Juristo, N., *Methontology: From Ontological Art Towards Ontological Engineering*, <https://doi.org/10.1109/AXMEDIS.2007.19>, AAAI-97 Spring Symposium Series, SS-97-06, 33–40 (1997).
- Gaona-García, P. A., Stoitsis, G., Sánchez-Alonso, S., y Biniari, K., *An exploratory study of user perception in visual search interfaces based on SKOS*, <https://doi.org/10.5771/0943-7444-2016-4-217>, knowledge Organization, 43(4), 217-238. (2016).
- Gaona-García, P.A., Montenegro-Marin C.E., Martín-Moncunill, D. *Entornos de búsquedas navegacionales a partir de esquemas de representación de conocimiento*. Editorial UD, ISBN: 9789587870046 (2018).
- Guzmán, J. A., López, M., y Durley, I. *Metodologías y métodos para la construcción de ontologías*. Scientia et technica, 2 (50), 133–140 (2012).
- Halpin, H., y Thompson, H., *One document to bind them: Combining xml, web services, and the semantic web*, <https://doi.org/10.1145/1135777.1135877>, International Conference on World Wide Web, 679–686 (2006).
- Hawkes Lewis, B., *Pruebas de accesibilidad*. Obtenido de Universitat Oberta de Catalunya - mosaic: <https://mosaic.uoc.edu/ac/le/es/m5/ud2/index.html#understandingrequirements> [Última consulta: Junio 30 2019], (2008).
- Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., y Giannopoulou, E., *Ontology visualization methods—a survey*, <https://doi.org/10.1145/1287620.1287621>, ACM Computing Surveys (CSUR), 39(4), 10 (2007).
- Ministerio de Salud y Protección Social. Sala situacional de Personas con Discapacidad (Nacional), 56 p. Retrieved from <https://discapacidadcolombia.com/phocadownloadpap/ESTADISTICAS/Sala-situacional-discapacidad-Nacional-agosto-2015.pdf> [Última consulta: Junio 30 2019], (2015).
- Nielsen, J., *Usability 101: Introduction to Usability*. Obtenido de Nielsen Norman Group: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> [Última consulta: Junio 30 2019] (2012).
- Noy, N. F., y McGuinness, D. L., *Ontology Development 101: A Guide to Creating Your First Ontology*. <https://doi.org/10.1016/j.artmed.2004.01.014>, Stanford Knowledge Systems Laboratory, 25 (2001).
- Ovalle, D.A., Salazar, O. M., Duque, N. D. *Modelo de Recomendación Personalizada en Cursos Virtuales basado en Computación Ubicua y Agentes Inteligentes*, <http://dx.doi.org/10.4067/S0718-07642014000600016>, Inf. tecnol., La Serena, v. 25, n. 6, p. 131-142 (2014).
- Quiroz, T., Salazar, O. M., Ovalle, D. A., *Modelo de Interfaz Adaptativa basada en Perfiles de Usuario y Ontologías para Recomendación de Objetos de Aprendizaje*, <http://dx.doi.org/10.4067/S0718-07642018000600295>, Inf. tecnol., La Serena, v. 29, n. 6, p. 295-306 (2018).
- Rodríguez, T., Aguilar, J. *Implementación del Marco Ontológico Dinámico Semántico*, <http://dx.doi.org/10.4067/S0718-33052017000300430>, Ingeniare. Rev. chil. ing., Arica, v. 25, n. 3, p. 430-448 (2017).
- Rogers, M. Power M., *Government accessibility standards and WCAG 2*. Obtenido de: <https://www.powermapper.com/blog/government-accessibility-standards/> [Última consulta: Junio 30 2019] (2017).

- Romano, R., y McDonald, C., *Assessing the quality of ontology*. <https://doi.org/10.1016/j.datak.2004.11.010>, Proceedings of MCIS, 2011, 1-11 (2011).
- Ribera, P. M., Térmens, M., y García-Martín, M. Cómo realizar tests de usabilidad con personas ciegas. <https://doi.org/10.3145/epi.2008.ene.12>, El Profesional de La Informacion, 17(1), 99–105 (2008).
- Sauro, J., *A Brief History of the Magic Number 5 in Usability Testing*. Obtenido de *Measuring Usability*. <http://www.measuringusability.com/blog/five-history.php> / [Última consulta: Noviembre 15 2019] (2010).
- Shawar, B. A., *Evaluating {Web} {Accessibility} of {Educational} {Websites}*, <http://dx.doi.org/10.3991/ijet.v10i4.4518>, International Journal of Emerging Technologies in Learning (IJET), 10(4), 4–10. Retrieved from <http://online-journals.org/index.php/i-jet/article/view/4518>, (2015).
- Sidar, F., *W3C Recommendation*. Obtenido de Web Content Accessibility Guidelines (WCAG) 2.0: <http://www.sidar.org/traducciones/wcag20/es/> [Última consulta: Junio 30 2019] (2009).
- Sorrentino, T., Santos, A., y otros 6 autores. *On the Move to Meaningful Internet Systems: OTM 2016 Workshops*, <https://doi.org/10.1007/978-3-319-55961-2>, On the Move to Meaningful Internet Systems: OTM 2016 Workshops, 10034, 682–689 (2017).
- Sosa, E., Sánchez, F., Macías, M., y Lozano-Tello, A., *Uso de Ontologías en Páginas Web para Mejorar su Accesibilidad a Invidentes*. VIII Jornadas de Ingeniería Del Software y Bases de Datos : Actas : (JISBD'03), 625–634 (2003).
- Standford University. Protégé. Obtenido de A free, open-source ontology editor and framework for building intelligent systems: <https://protege.stanford.edu/>, [Última consulta: Junio 30 2019] (2016).
- Statistic Brain. Static Brain. Obtenido de Google Annual Search Statistics: <http://www.statisticbrain.com/google-searches/>, [Última consulta: Junio 30 2019], (2017).
- World Wide Web Size. Obtenido de Daily estimated size of the world wide web: <http://www.worldwidewebsite.com/> [Última consulta: Junio 30 2019], (2017).
- Ziener, C., *Web accessibility for people with disabilities*. Library Journal, 126(1), 144, (2001).

Página en blanco