



resumen

El objetivo del artículo es dar a conocer una visión general sobre la tecnología AJAX. La tendencia en cuanto a aplicaciones Web, se refiere a obtener cada vez una mejora, de tal forma que el usuario final se sienta a gusto con dichas aplicaciones, toda vez que podrá encontrar en ellas interfaces de usuario amenas, menores tiempos de respuesta, tareas simultáneas, funcionalidades similares a las aplicaciones de escritorio, entre otras.

Ajax: Asynchronous JavaScript and XML

Ing. Alejandro León M. • Cristhian Camilo Quintero Y. •
Diana Paola Rojas R.

introducción

Debido al comienzo que tuvo Internet, con sus páginas poco dinámicas y los grandes requerimientos de los clientes, se ha visto la necesidad de optimizar los tiempos de respuesta para alcanzar las exigencias de quienes lo requieren. Para suplir este problema se crearon diferentes soluciones tecnológicas que permitieran a las páginas Web tener un mayor dinamismo; estas soluciones van desde los antiguos scripts CGI pasando por tecnologías tan robustas como ASP. No obstante, empresas de tecnología, instituciones educativas y centros de investigación siguen trabajando en el desarrollo de nuevas tecnologías como Ruby o AJAX, entre otras, que permitan suplir las necesidades del día a día, para mejorar la experiencia del usuario en la Web y el desempeño de las aplicaciones del lado del servidor.

AJAX se ha venido usando ampliamente por varias compañías, pero es poca la documentación y estudios que se encuentran en español, este artículo muestra las ventajas de AJAX en el desarrollo de aplicaciones Web, haciendo un comparativo entre una aplicación con AJAX y esta misma sin esta tecnología.

Como toda tecnología, hay cosas que se sacrifican cuando se trabaja con ella y AJAX no es la excepción, el artículo presenta algunas de las desventajas mas conocidas de dicha tecnología.





cinco
ocho

¿Qué es Ajax?

En 1989 Tim Berners Lee inventó lo que hoy en día se conoce como HTML (Hyper Terminal Markup Language) que permitía separar la implementación, de la información que se le mostraba al usuario, también creó el HTTP (Hyper Text Transfer Protocol) que permite a los servidores comunicarse con los clientes (Navegadores, aplicaciones de escritorio, entre otras).

Cuando aparecieron las páginas Web solamente tenían contenido estático, lo cual para comenzar estaba bien, toda vez que los científicos solo intercambiaban textos de sus investigaciones. Sin embargo, al popularizarse Internet los usuarios comenzaron a demandar un contenido que les permitiera interactuar con los sitios Web. La primera solución que se creó del lado fue la de los scripts CGI (Common Gateway Interface), los cuales pueden ser creados en diferentes lenguajes de programación; estos scripts son rutinas que se ejecutan del lado del servidor cuando el cliente envía una petición. No obstante, los scripts CGI son lentos y poco seguros.

Las empresas de tecnología siguieron trabajando en suplir las necesidades, entre las más destacadas se encuentra Sun Microsystems que creó el lenguaje Java™ e introdujo los Servlets, los cuales se ejecutaban desde el servidor sin necesidad de preocuparse por la versión Java™ instalada en el cliente; a su vez podían ser usados como sustitutos de los scripts CGI.

Los Servlets tienen un completo acceso a la API de Java™ y vienen con una librería para el manejo del protocolo HTTP. Pero, el diseño de interfaces puede ser complejo pues se debe generar el código HTML manualmente.

Fue este problema el que llevó a Sun Microsystems a desarrollar las JSP (Java Server Pages) como contrapropuesta a las ASP (Active Server Pages) de Microsoft las cuales fueron la alternativa creada por esta compañía para la creación de páginas dinámicas, estas páginas del lado del servidor ejecutan sentencias de código para generar documentos en HTML inicialmente y luego en DHTML o XML que pueden ser renderizados del lado del cliente en un explorador Web.

Por otro lado, Netscape trabajaba en un lenguaje que permitía a los desarrolladores no estaban familiarizados con Java™ crear aplicaciones dinámicas. Para esto la compañía implementó un lenguaje de encriptación denominado JavaScript, que tenía como objetivo ayudar a los desarrolladores a modificar los atributos de sus páginas de forma dinámica y así proveer al usuario una mejor experiencia. Es de notar que JavaScript depende de la





manera como esté implementado en el navegador del cliente y por ello muchas de sus funcionalidades, dependían del navegador en que se ejecutaran.

En 1997, cuando Netscape y Microsoft lanzaron las versiones 4.0 de sus navegadores, los cuales implementaban la versión 1.1 [1] del protocolo HTTP, los desarrolladores podían crear nuevas aplicaciones basadas en DHTML (Dynamic Hyper Text Language) que es una unión entre HTML, CSS (Cascading Style Sheets) y JavaScript. La combinación de estas tecnologías le permitía a los desarrolladores modificar el contenido y la estructura de las páginas rápidamente. Sin embargo, las implementaciones para cada uno de estos navegadores era diferente lo cual ha sido un problema para DHTML. Además, los costos de desarrollo también fueron altos debido a la gran cantidad de código necesario para crear las aplicaciones.

La W3C (Wide World Web Consortium) creó XML (eXtensible Markup Language) en 1998 como una forma de representar información, también se pretendía estandarizar las páginas Web, debido a que las aplicaciones de XML han llegado mucho más lejos y actualmente es usado como un metalenguaje en diferentes aplicaciones tanto Web, como de escritorio; ya que permite modelar la información por medio de etiquetas personalizables, las cuales deben estructurarse a través de un DTD. Esta flexibilidad para el manejo de la información no era soportada por HTML ya que el conjunto de etiquetas que definía este lenguaje, no permitía contextualizar la información de la página, lo que si se soluciona con XML. Hoy en día, XML funciona tanto del lado del servidor como del cliente e incluso ha sido usado para definir protocolos de comunicación.

Las páginas Web se basaban en el paradigma petición/respuesta donde los usuarios envían solicitudes al servidor normalmente usando los métodos GET o POST y este contesta con una página Web generalmente en lenguaje HTML; el servidor devuelve la página completa y no solamente la sección que se modificó, por lo que se obliga a la aplicación cliente que maneja el usuario a refrescar por completo la página que se está visualizando.

Los primeros intentos para tratar de solucionar este problema se basaron en scripts remotos, los cuales le permiten al desarrollador crear páginas que interactúen con el servidor de forma asíncrona haciendo uso de la versión 1.1 del protocolo HTTP.

Posteriormente, en febrero de 2005, Jesse James Garrett en un debate sobre nuevos acercamientos a aplicaciones Web, utilizó el término AJAX para hacer referencia a la definición de: Asíncrono, XMLHttpRequest, JavaScript, CSS, DOM (Document





Object Model). En dicho debate Garret mostró como esta combinación de tecnologías permitiría crear aplicaciones mucho más ligeras y dinámicas [2].

AJAX, es otra de las soluciones al problema de petición / respuesta; la columna vertebral de esta tecnología es JavaScript, que utiliza el objeto XMLHttpRequest y del estándar DOM nivel 3, establecido por la W3C.

AJAX no requiere de ningún software, ni hardware especial y casi todos los navegadores lo soportan sobre todo en sus últimas versiones; esta tecnología puede usarse con IE 5 en adelante, Mozilla Firefox, Netscape 7.1 y Opera en su versión 8.0 [9].

Otra gran ventaja de esta tecnología es que puede usarse diversos lenguajes de lado del servidor como: .Net, Java™, PHP; entre otros, lo que no obliga al desarrollador a aprender un nuevo lenguaje para usarlo.

Definición

Como se mencionaba, AJAX es un conjunto de varias tecnologías que permiten un mayor rendimiento a las aplicaciones Web.

La parte fundamental de AJAX es el objeto XMLHttpRequest, encargado de las peticiones y respuestas que se realizan de forma asincrónica. Por su parte, JavaScript se encarga de procesar la respuesta del lado del cliente con la ayuda de DOM que reconstruye el árbol para modificar la estructura de una parte de la página.

Otras tecnologías que hacen parte de AJAX son HTML como lenguaje de marcado que junto con CSS ayudan a mejorar la apariencia visual para el usuario y XML, encargado de la manipulación de los datos.

AJAX en el mundo

AJAX está siendo usado para crear aplicaciones Web que antes únicamente se veían como aplicaciones de escritorio; a pesar de ser una tecnología relativamente nueva, varias de las más importantes compañías en el mundo están usándola para sus desarrollos, como Google, Yahoo y Amazon, entre otros.

Sin lugar a dudas, Google, ha sido la empresa que más aplicaciones ha realizado con AJAX, incluso antes de que el término fuera definido (definió sus productos como aplicaciones desarrolladas con Javascript antes de que AJAX existiera). Uno de sus primeros productos fue Gmail el cual se lanzó en abril de 2004[3]. Las principales funcionalidades





de AJAX dentro de Gmail se encuentran en su interfaz de usuario, donde es posible realizar acciones como adjuntar archivos mientras se escribe un correo electrónico, listados de contactos y al agregarlos a un grupo, estos se actualizan en forma automática, lo cual no era muy común en la mayoría de proveedores de correo, ni siquiera en clientes de correo como Outlook. A este producto Google ha sumado una considerable cantidad de aplicaciones como Google Maps para realizar búsquedas de las principales ciudades del mundo, mediante mapas en los que se pueden hacer acercamientos para obtener más y mejores detalles, entre otras funciones. Google Spreadsheets para el manejo de hojas de cálculo y Google Docs para la creación de documentos, los cuales son la base del paquete de ofimática que dicha compañía se encuentra desarrollando haciendo uso AJAX.

Por su parte, Amazon, posee una de las tiendas online más famosas y reconocidas del mundo. Sin embargo, debutó con un motor de búsqueda conocido como A9 [4], que a su vez utiliza otros como Google, IMDb [5] y Wikipedia, entre otros; lo que lo hace único y diferente es como funciona su interfaz gráfica. A9 permite seleccionar cuáles motores intervienen en la búsqueda y los resultados se muestran por columnas, además de permitir obtener más información de un elemento de la búsqueda, sin necesidad de recargar toda la página.

En el 2005 Yahoo decidió relanzar su sitio de noticias [6]; los cambios comenzaron por el aspecto gráfico para permitir una mejor interacción a los usuarios.

En este sitio cuando se pasa el mouse sobre el vínculo de una noticia, se despliega un pequeño cuadro que muestra más información y, en algunos casos, una imagen asociada. Cabe destacar que para este caso en particular, el sitio no depende de la funcionalidad de AJAX, es un valor agregado que da una mejor experiencia al usuario.

Otra aplicación muy popular realizada por Yahoo (quizás la más popular de esta compañía) es flickr [7], un sitio que permite subir, organizar y compartir con conocidos, amigos y familiares fotos e imágenes. Uno de los aspectos más importantes es que los usuarios pueden agregar comentarios dentro de sectores de las imágenes.

AJAX vs. aplicaciones convencionales

Dentro de las aplicaciones que comúnmente se han venido trabajando y AJAX se nota una diferencia relacionada con tiempos de respuesta, algunos efectos visuales y tareas simultáneas que hacen de esta tecnología un gran avance para las aplicaciones Web.

Un ejemplo de una aplicación sin AJAX es el proveedor de correo electrónico de la universidad El Bosque, en el cual se debe adjuntar primero el archivo, esperar a que la





operación termine y luego sí ingresar las direcciones de correo destinatarias y el mensaje personal. Con AJAX, por ejemplo en Gmail, se pueden realizar dos tareas simultáneas. Mientras se termina de adjuntar el archivo, el usuario puede ir ingresando las cuentas de correo o el mensaje personal; esto gracias a que las peticiones son dadas únicamente para los datos estrictamente necesarios y al objeto DOM de JavaScript encargado de la estructura de la operación que en ese momento se esté realizando.

A continuación se presenta la arquitectura convencional y la usada con AJAX para poder tener un mayor entendimiento sobre el funcionamiento de ésta tecnología.

El funcionamiento de las aplicaciones convencionales consiste en:

- Definir una página para cada evento.
- Cada acción debe retornar la página completa al navegador.
- La página es mostrada al usuario [8].

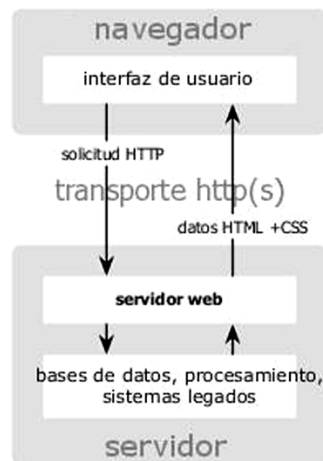


Fig. 1. Modelo aplicación convencional [8]

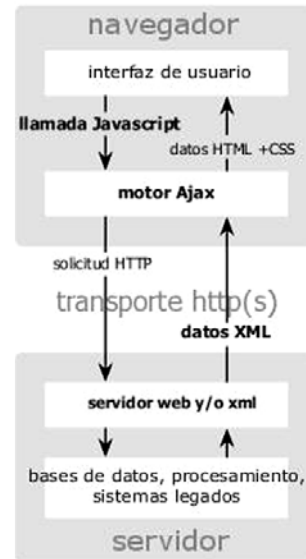
El funcionamiento de las aplicaciones con AJAX consisten en:

- El usuario hace una petición mediante una URL.
- El navegador retorna la página original.
- El navegador crea un árbol DOM.
- Algunas acciones ocasionan peticiones asincrónicas a unas URL sin generar el árbol DOM.
- El navegador retorna la información mediante la función existente dentro de la página. [8].





Fig. 2. Modelo aplicación con AJAX ^[8]



Desventajas

Así como AJAX ofrece varias ventajas, de igual forma registra desventajas que no se pueden dejar de lado. Una de ellas está relacionada con la funcionalidad que presenta el botón atrás del navegador, pues con el uso de AJAX esta funcionalidad queda completamente deshabilitada al igual que la recarga de la página, la cual puede tener resultados inesperados.

Otro inconveniente que se presenta está relacionado con las peticiones realizadas desde el cliente, toda vez que estas corren en forma oculta; el usuario no puede ver el estado o progreso de la acción requerida al igual que los favoritos. Como hay tantas peticiones al servidor no se sabe en qué punto se agregan datos y la información puede volverse errónea.

También es posible tener inconsistencias en la manera como se despliega la información al usuario; muchas veces títulos, símbolos y texto son usados para resaltar un dato o para atraer la atención del usuario. Cuando AJAX permite modificar la información puntual dentro del árbol del documento (DOM) es posible que se modifique solo una parte sin tener en cuenta las demás ramas relacionadas con la información que se está modificando.

Desde el punto de vista de diseño, pueden generarse problemas al tratar de implementar AJAX en una aplicación cuya arquitectura no se asemeje a la arquitectura Call and Return, típica en la programación Orientada a Eventos. Por ello, en el ejemplo esbozado





en este artículo fue necesario cambiar algunas de las clases en la implementación hecha con AJAX.

Cabe aclarar que AJAX es una tecnología relativamente nueva y aún se encuentra en una etapa de consolidación, lo cual significa que a futuro muchas de las desventajas antes mencionadas serán remediadas.

Gestión y control de actas con AJAX

La forma en la cual se comprobó la utilidad de AJAX fue mediante dos aplicaciones que gestionan actas para el manejo y control de los proyectos de grado de la Facultad de Ingeniería de Sistemas de la Universidad El Bosque. Dicha aplicación permite crear, modificar, eliminar, consultar y firmar actas, modificar compromisos, retirar un participante ya asignado a un acta. Cabe aclarar que el acta no debe estar firmada para la eliminación y modificación de la misma, el retiro del participante y la modificación de los compromisos.

Para la creación de un acta se debe ingresar la información requerida en especial el nombre, luego ingresar los temas, asignar los usuarios, los compromisos, asignar responsables a los mismos y finalmente guardar el acta.

Si por el contrario se desea consultar, eliminar o firmar un acta se debe seleccionar una de la lista de las ya creadas y continuar con la operación.

La modificación del acta y del compromiso son opciones similares en las que se debe seleccionar el acta creada con anterioridad y luego ingresar la información solicitada (nombre, temas, fechas, descripción entre otros) que después será enviada para culminar el proceso.

Finalmente, para retirar un usuario es necesario seleccionar el acta de la cual se va a retirar al usuario; luego seleccionar al usuario de una lista que aparece y enviar la solicitud.

La primera aplicación fue realizada sin AJAX (ninguna de las tecnologías que lo componen); el desarrollo fue realizado con JSP's y Servlets; el manejo de las validaciones de campos nulos, datos y fechas inconsistentes estuvieron del lado del servidor; algunos de los problemas presentados se dieron en el transporte de los datos y la aplicación se volvía lenta; otro inconveniente presentado fue la integridad de los datos que se presentó para alguna información requerida.





Los inconvenientes que obtuvo el usuario final estuvieron relacionados con la pérdida de datos; demasiados pasos para la creación de un acta y con la interfaz la cual no era amena, carecía de efectos visuales.

A continuación se presentan los pasos a seguir para la creación de un acta en la aplicación sin AJAX.

Por favor ingrese la siguiente información para la creación del acta:

Nombre:

Fecha: / /

Descripción:

No. Temas:

Fig. 3. Crear acta paso 1 ^[10]

Por favor ingrese la información correspondiente para la creación de los temas:

Nombre Tema 1:

Nombre Tema 2:

Fig. 4. Crear tema ^[10]

Por favor ingrese la siguiente información para la asignación de usuarios al acta:

Nombre Participante	Asignar al Acta
CAROLINA RAMIREZ	<input type="checkbox"/>
SANDRA ZARATE	<input type="checkbox"/>
DIANA ROJAS	<input type="checkbox"/>
JUANIX RUBIO	<input type="checkbox"/>
CAMILO QUINTERO	<input type="checkbox"/>

No. Compromisos:

Fig. 5. Asignar participante ^[10]





Nombre:

Fecha de Entrega: 14 / 5 / 2007

Descripcion:

Responsables:

CAROLINA RAMIREZ

SANDRA ZARATE

DIANA ROJAS

JUANIX RUBIO

CAMILO QUINTERO

COMPROMISO 2

Nombre:

Fecha de Entrega: 14 / 5 / 2007

Descripcion:

Responsables:

CAROLINA RAMIREZ

SANDRA ZARATE

DIANA ROJAS

JUANIX RUBIO

CAMILO QUINTERO

Fig. 6. Crear compromisos ^[10]

Por el contrario, la segunda aplicación realizada con AJAX y las tecnologías que lo componen fue mucho más eficiente; con una interfaz de usuario más rica y no presentaba inconvenientes en la inconsistencia de los datos. Para este caso las validaciones fueron hechas del lado del cliente con la ayuda de JavaScript y el objeto DOM, evitando así idas y uso de los recursos en el servidor; esto permitió eliminar algunas de las clases que componían la primera aplicación. Además de la implementación básica con AJAX se usaron diferentes frameworks que permitieron disminuir en forma significativa el tiempo de desarrollo de la aplicación. No obstante, estos tiempos se utilizaron en el conocimiento y manejo de los mismos, pero a posteriori dicho lapso podrá ser significativo, en la medida en que para ese entonces se tendrá un conocimiento previo del tema.

A continuación se presentara una imagen con los pasos a seguir para la creación de un acta en la aplicación con AJAX (ver Fig. 7 página 52).

Conclusiones

AJAX como conjunto de tecnologías ofrece muchas ventajas relacionadas con la facilidad para el manejo de las aplicaciones y su presentación; la transmisión de datos que se disminuye hasta en un 90%; el número de variables de sesión, por ejemplo para la creación de un acta de dos temas y un compromiso es igual a diez, una por cada campo que pueda





presentar errores -el usuario y el acta que se van modificando en el proceso de la creación- mientras que usando AJAX solo se requiere la variable de usuario.

CREACIÓN ACTA

Por favor ingrese la siguiente información para la creación del acta:

Nombre:

Fecha:

Descripción:

[AGREGAR TEMA](#) [RETIRAR TEMA](#)

Tema 1:

Seleccionar Participantes

CAROLINA RAMIREZ

SANDRA ZARATE

DIANA ROJAS

JUANIX RUBIO

CAMILO QUINTERO

[AGREGAR COMPROMISO](#) [RETIRAR COMPROMISO](#)

COMPROMISO 1

Nombre:

Fecha:

Descripción:

Responsables

DIANA ROJAS

CAMILO QUINTERO

Fig. 7. Crear acta ^[10]

La aplicación con AJAX además de ser mejor en rendimiento, también permite que el desarrollador tenga mayor granularidad de la misma, aislando diferentes componentes en capas más especializadas.

Es decir, para la aplicación realizada sin AJAX, se encontraba una arquitectura de tres niveles:

- Presentación: donde se encontraban las JSP's.
- Lógica: donde se encuentra toda la lógica del negocio.
- Persistencia: donde se encuentra la comunicación con medio persistente (Base de Datos).

Para la aplicación realizada con AJAX esta estructura de tres niveles persiste; sin embargo, la capa de presentación puede ser dividida en varias subcapas.





- Componentes Gráficos: el aspecto de la aplicación ya no es manejado dentro de las JSP's como atributos de las etiquetas; se hace mediante Hojas de Estilo CSS, permitiendo un mayor control en el aspecto grafico global de la aplicación.
- Lógica del lado del Cliente: este aspecto es manejado con Javascript; se busca tener una consistencia en la información que se verifica antes de ser enviada.

Referencias

- [1] INTERNET SOCIETY. Hypertext Transfer Protocol RFC 2068. Jun. 1999 (Consultado 21 Sep. 2006). <http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html#sec1>
- [2] Para más información consultar el artículo completo en: <http://adaptivepath.com/publications/essays/archives/000385.php>
- [3] <http://softwarelibre.uca.es/node/87?PHPSESSID=e7059890f8aac3252953970768fefdc2>
- [4] <http://www.a9.com>
- [5] Internet Movie Data Base
- [6] <http://news.yahoo.com>
- [7] <http://www.flickr.com/>
- [8] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
- [9] Para un listado de otros navegadores de escritorio y móviles que soportan Ajax ver: <http://olgacarreras.blogspot.com/2007/02/ajax-accesible.html>
- [10] QUINTERO, Camilo y ROJAS, Diana. "Tutorial para el desarrollo de una aplicación Web en AJAX para la creación y control de actas". Universidad El Bosque, Bogotá.2007.

Alejandro León M. Profesor asociado, Facultad de Ingeniería de Sistemas, Universidad El Bosque. Especialización en Construcción de Software, Universidad de los Andes, en curso. Ingeniero de Sistemas, Universidad El Bosque

Crishtian Camilo Quintero Y. Estudiante de Décimo Semestre de Ingeniería de Sistemas, Universidad El Bosque.

Diana Paola Rojas R. Estudiante de Décimo Semestre de Ingeniería de Sistemas, Universidad El Bosque.

