



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



Programación web

Investigación AJAX

Nombre del alumno: Luis Carlos Anaya Avalos

Matrícula: 1630227

Nombre del maestro: Luis Roberto Flores



04/06/2020

¿Qué es AJAX y cómo funciona?

AJAX significa JavaScript asíncrono y XML (Asynchronous JavaScript and XML). Es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano. Espera, ¿qué es AJAX de nuevo? Vamos a revisar cada término por separado.

JavaScript es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como **HTML**, como lo sugiere su nombre: eXtensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos.

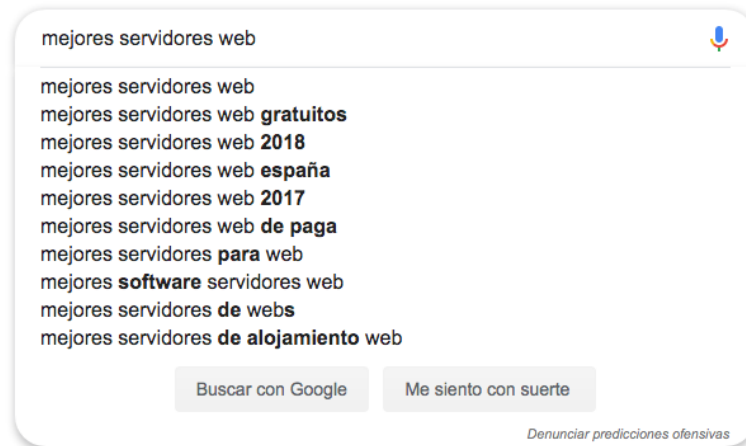
Tanto JavaScript como XML funcionan de forma asíncrona en AJAX. Como resultado, cualquier aplicación web que use AJAX puede enviar y recuperar datos del servidor sin la necesidad de volver a cargar toda la página.

Contenido

Ejemplos prácticos de AJAX

Tomemos como ejemplo la función de autocompletado de Google; esta te ayuda a completar tus palabras clave mientras las escribes. Las palabras clave cambian en tiempo real, sin embargo, la página como tal no cambia. A principios de los años 90, cuando la Internet no era tan avanzada, la misma función requeriría que Google volviera a cargar la página cada vez que apareciera una nueva recomendación en tu pantalla. AJAX permite que el intercambio de datos y la capa de presentación funcionen simultáneamente sin que interfieran la una con la otra.





Autocompletado de Google

El concepto de AJAX ha existido desde mediados de los años 90. Sin embargo, obtuvo un reconocimiento más amplio cuando Google comenzó a incorporar el concepto en Google Mail y Google Maps en el 2004. Hoy en día, se usa ampliamente en varias aplicaciones web para agilizar el proceso de comunicación del servidor.

Aquí hay más ejemplos útiles de AJAX en nuestra vida cotidiana.

- **Sistemas de votación y calificación**

¿Alguna vez has calificado un producto que compraste online? ¿Alguna vez has llenado un formulario de votación en línea? De cualquier manera, ambas operaciones usan AJAX. Una vez que haces clic en el botón de calificación o de votación, el sitio web actualizará el cálculo pero la página completa permanecerá sin cambios.

- **Salas de chat**

Algunos sitios web tienen un chat incorporado en su página principal, mediante el cual puedes hablar con un agente de atención al cliente. No tienes que preocuparte si quieres explorar la página al mismo tiempo. AJAX no volverá a cargar la página cada vez que envíes y recibas un mensaje nuevo.



- **Notificación de tendencias de Twitter**

Twitter ha incorporado AJAX recientemente para sus actualizaciones. Cada vez que se crean nuevos tweets sobre ciertos temas de tendencia, Twitter actualizará las nuevas cifras sin afectar la página principal.

En pocas palabras, AJAX hace que las funciones multitarea sean fáciles. Si alguna vez observas una situación similar en la que dos operaciones funcionan simultáneamente, con una en ejecución y la otra inactiva, es posible que sea AJAX en acción.

Tecnologías incluidas en Ajax

Ajax es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios. PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico también utilizado en el método Ajax.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Antecedentes de Ajax

A pesar de que el término «Ajax» fue creado en 2005 por Jesse James Garrett,¹ la historia de las tecnologías que permiten Ajax se remonta a una década antes con la iniciativa de Microsoft en el desarrollo de Scripting Remoto. Sin embargo, las técnicas para la carga asíncrona de contenidos en una página existente sin requerir recarga completa remontan al tiempo del elemento iframe (introducido en Internet Explorer 3 en 1996) y el tipo de elemento layer (introducido en Netscape 4 en 1997, abandonado durante las primeras etapas de desarrollo de Mozilla).



Ambos tipos de elemento tenían el atributo src que podía tomar cualquier dirección URL externa, y cargando una página que contenga JavaScript que manipule la página paterna, pueden lograrse efectos parecidos al Ajax.

El Microsoft's Remote Scripting (o MSRS, introducido en 1998) resultó un sustituto más elegante para estas técnicas, con envío de datos a través de un applet Java el cual se puede comunicar con el cliente usando JavaScript. Esta técnica funcionó en ambos navegadores, Internet Explorer versión 4 y Netscape Navigator versión 4. Microsoft la utilizó en el Outlook Web Access provisto con la versión 2000 de Microsoft Exchange Server.

La comunidad de desarrolladores web, primero colaborando por medio del grupo de noticias [microsoft.public.scripting.remote](#) y después usando blogs, desarrollaron una gama de técnicas de scripting remoto para conseguir los mismos resultados en diferentes navegadores. Los primeros ejemplos incluyen la biblioteca JSRS en el año 2000, la introducción a la técnica imagen/cookie² en el mismo año y la técnica JavaScript bajo demanda (JavaScript on Demand)³ en 2002. En ese año, se realizó una modificación por parte de la comunidad de usuarios⁴ al *Microsoft's Remote Scripting* para reemplazar el applet Java por XMLHttpRequest.

Frameworks de Scripting Remoto como el ARSCIF⁵ aparecieron en 2003 poco antes de que Microsoft introdujera Callbacks en ASP.NET.⁶

Desde que XMLHttpRequest está implementado en la mayoría de los navegadores, raramente se usan técnicas alternativas. Sin embargo, todavía se utilizan donde se requiere una mayor compatibilidad, una reducida implementación, o acceso cruzado entre sitios web. Una alternativa, el Terminal SVG⁷ (basado en SVG), emplea una conexión persistente para el intercambio continuo entre el navegador y el servidor.

Problemas e inconvenientes

- Las páginas creadas dinámicamente mediante peticiones sucesivas AJAX, no son registradas de forma automática en el historial del navegador, así que haciendo clic en el botón de "volver" del navegador, el usuario no será devuelto a un estado anterior de la página, en cambio puede volver a la última página que visitó. Soluciones incluyen el uso de IFrames invisible para desencadenar cambios en el historial del navegador y el cambio de la porción de anclaje de la dirección (después de un #).
- Los motores de búsqueda no analizan JavaScript. La información en la página dinámica no se almacena en los registros del buscador.



Exceptuando Google, que desde 2011 sí indexa contenido Ajax y JavaScript. Matt Cutts (director del departamento contra el spam en web de Google) lo confirmó en Twitter: "Googlebot keeps getting smarter. Now has the ability to execute AJAX/JS to index some dynamic comments."

- Hay problemas usando Ajax entre nombres de dominios, a esto se le conoce como *Same Origin Policy* o *Política del Mismo Origen*, lo cual es una medida de seguridad que puede ser solucionada con *Cross-Origin Resource Sharing* (CORS).
- Dependiendo de como se desarrolle el sitio web, puedes mejorar o empeorar la carga en el servidor. Ajax puede ayudar al servidor a evitar la fase de renderización de HTML, dejándole ese trabajo al cliente, pero también puede sobrecargar al servidor si se hace varias llamadas a Ajax.
- Es posible que páginas con Ajax no puedan funcionar en teléfonos móviles, PDA u otros aparatos. Ajax no es compatible con todos los software para invidentes u otras discapacidades.

¿Como funciona AJAX?

Ten en cuenta que AJAX no es una sola tecnología, ni es un lenguaje de programación. Como se dijo antes, AJAX es un conjunto de técnicas de desarrollo web. El sistema generalmente comprende:

- **HTML/XHTML** para el lenguaje principal y **CSS** para la presentación.
- **El Modelo de objetos del documento (DOM)** para datos de visualización dinámicos y su interacción.
- **XML** para el intercambio de datos y **XSLT** para su manipulación. Muchos desarrolladores han comenzado a reemplazarlo por **JSON** porque es más similar a JavaScript en su forma.
- El objeto **XMLHttpRequest** para la comunicación asíncrona.
- Finalmente, el lenguaje de programación **JavaScript** para unir todas estas tecnologías.

Es necesario algún conocimiento técnico para entenderlo completamente. Sin embargo, el procedimiento general de cómo funciona AJAX es bastante simple. Dale un vistazo al diagrama y la tabla siguientes para mayor explicación.



Diagrama:

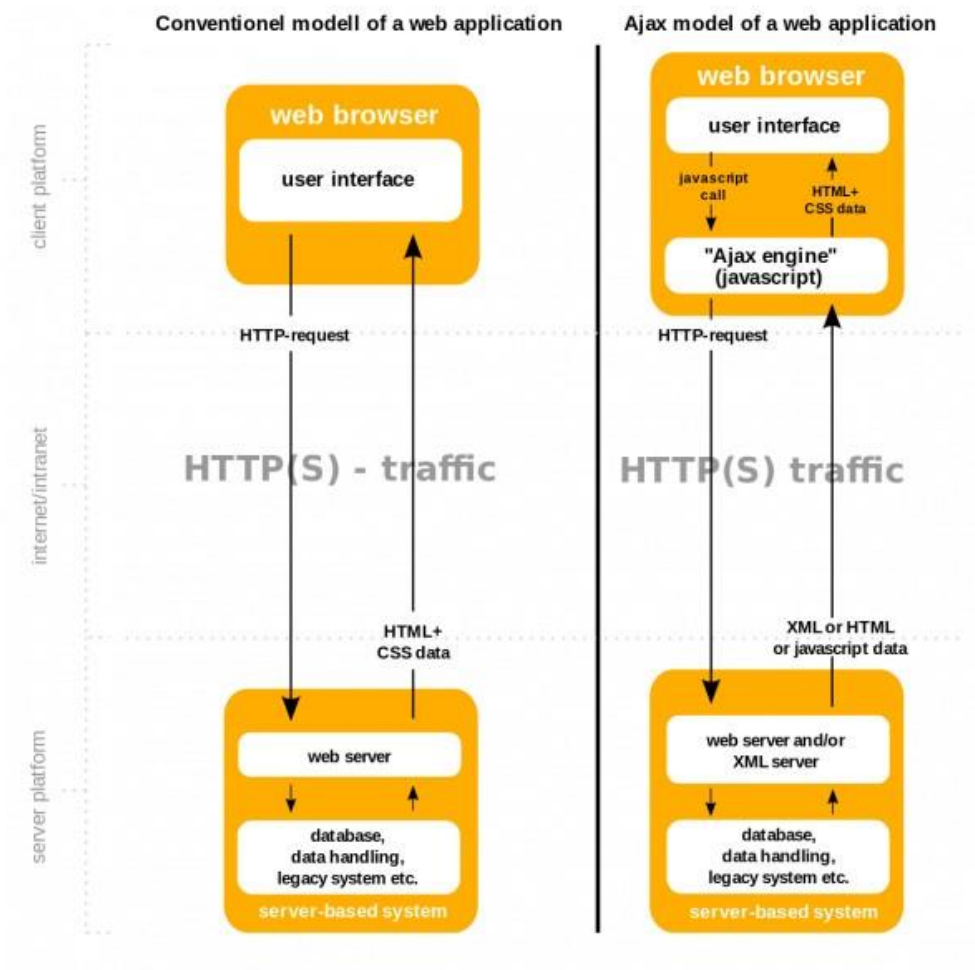


Tabla comparativa:

Modelo convencional

1. Se envía una solicitud HTTP desde el navegador web al servidor.
2. El servidor recibe y, posteriormente, recupera los datos.
3. El servidor envía los datos solicitados al navegador web.
4. El navegador web recibe los datos y vuelve a cargar la página para que aparezcan los datos.

Durante este proceso, los usuarios no tienen más remedio que esperar hasta que se complete todo el proceso. No solo consume mucho tiempo, sino que también supone una carga innecesaria en el servidor.

Modelo AJAX

1. El navegador crea una llamada de JavaScript que luego activará XMLHttpRequest.
2. En segundo plano, el navegador web crea una solicitud HTTP al servidor.
3. El servidor recibe, recupera y envía los datos al navegador web.
4. El navegador web recibe los datos solicitados que aparecerán directamente en la página. No se necesita recargar.

En resumen

Dejando de lado la definición, la mayor ventaja de usar AJAX es que optimiza la experiencia del usuario. Tus visitantes no tienen que esperar mucho tiempo para acceder a tu contenido. Sin embargo, también depende de lo que necesites. Google, por ejemplo, le permite a los usuarios elegir entre AJAX y una versión convencional al usar Google Mail. Lo mejor es poner las necesidades de los usuarios como prioridad en tu lista y usar AJAX de acuerdo a eso.

