

AJAX: UNA TÉCNICA EDUCATIVA DE DESARROLLO WEB

Tomás Clemente Carrilero. Profesor de enseñanza secundaria.

La técnica de desarrollo web Ajax se encuadra en el Decreto 230/2011, de 28/07/2011, por el que se establece el currículo del ciclo formativo de grado superior correspondiente al título de Técnico o Técnica Superior en Desarrollo de Aplicaciones Web en la comunidad Autónoma de Castilla-La Mancha. [2011/11276]

Breve historia

El termino Ajax, fue acuñado por primera vez por Jesse James Garret, fundador de la consultora Adaptive Path, en su artículo "Ajax: A New Approach to Web Applications", en febrero de 2005.

A modo de introducción, podemos destacar de este artículo lo siguiente:

"Los diseñadores de sitios web, no pueden evitar sentir envidia de los programadores de aplicaciones de escritorio. Las aplicaciones de escritorio tienen una apariencia y tiempo de respuesta que parecía fuera del alcance de Internet. La misma simplicidad que ha permitida la rápida proliferación de la Web, también crea una brecha entre las experiencias que podemos ofrecer y las experiencias que los usuarios pueden obtener de las aplicaciones de escritorio"

Introducción

¿Qué es AJAX? AJAX es el acrónimo de Asynchronous Javascript and XML.

Ajax no es una tecnología, no es un nuevo lenguaje de programación ni nada similar, en sí mismo Ajax no es nada. Ajax es realmente la asociación de varias tecnologías que por méritos propios han llegado a donde están.

AJAX es una técnica de desarrollo web que genera aplicaciones web interactivas combinando:

1) XHTML y CSS para la presentación de información.

XHTML es el acrónimo de eXtensible **H**ypertext **M**arkup **L**anguage (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado para sustituir a HTML como estándar para las páginas web. Su objetivo es lograr una web semántica, donde la información y la forma de presentarla estén claramente separadas.

CSS son las siglas de Cascading Style Sheets (hojas de estilo en cascada) CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión enXHTML). La idea que subyace tras estas siglas es separar la estructura de un documento de su presentación.

2) DOM para visualizar dinámicamente e interactuar con la información presentada.

DOM son las siglas de Document Object Model. Es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

3) XML, XSLT y JSON para intercambiar y manipular datos.

XML siglas en inglés de eXtensible Markup Language, no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo...

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

XSLT o Transformaciones XSL presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. XSLT realiza la transformación del documento utilizando una o varias reglas de plantilla. Estas reglas de plantilla unidas al documento fuente a transformar alimentan un procesador de XSLT, el que realiza las transformaciones deseadas poniendo el resultado en un archivo de salida, o, como en el caso de una página web, las hace directamente en un dispositivo de presentación tal como el monitor del usuario.

JSON acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX.

4) El objeto XMLHttpRequest para recuperar datos asincrónicamente.

Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas. La interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

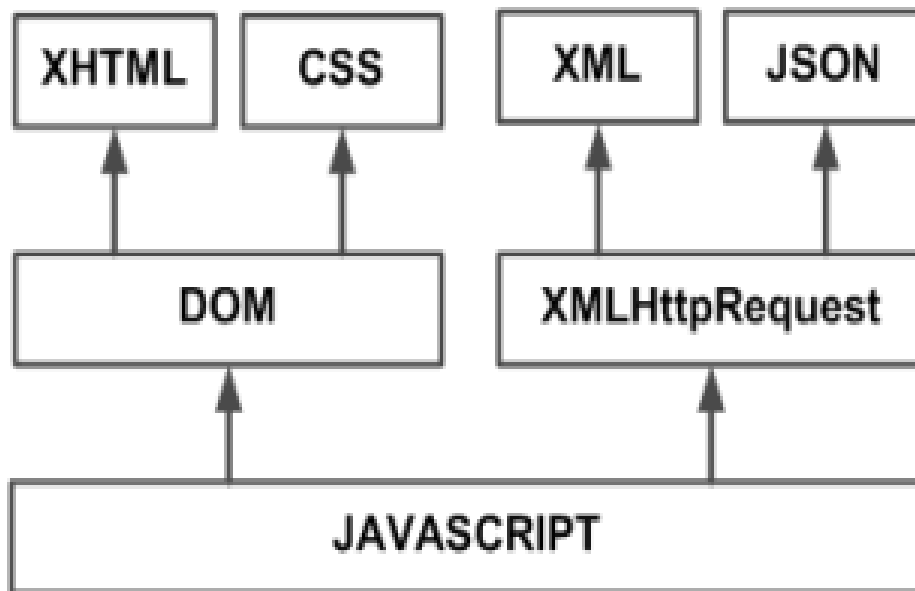
El uso más popular, si bien no el único, de esta interfaz es proporcionar contenido dinámico y actualizaciones asíncronas en páginas WEB mediante tecnologías construidas sobre ella como por ejemplo AJAX.

5) JavaScript como nexo de unión de todas estas tecnologías.

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Gráficamente, este podría ser el esquema de Ajax:



Comunicación síncrona frente a comunicación asíncrona.

Para entender correctamente en qué consiste Ajax, juega un papel muy importante el tipo de comunicación que usamos: síncrona o asíncrona.

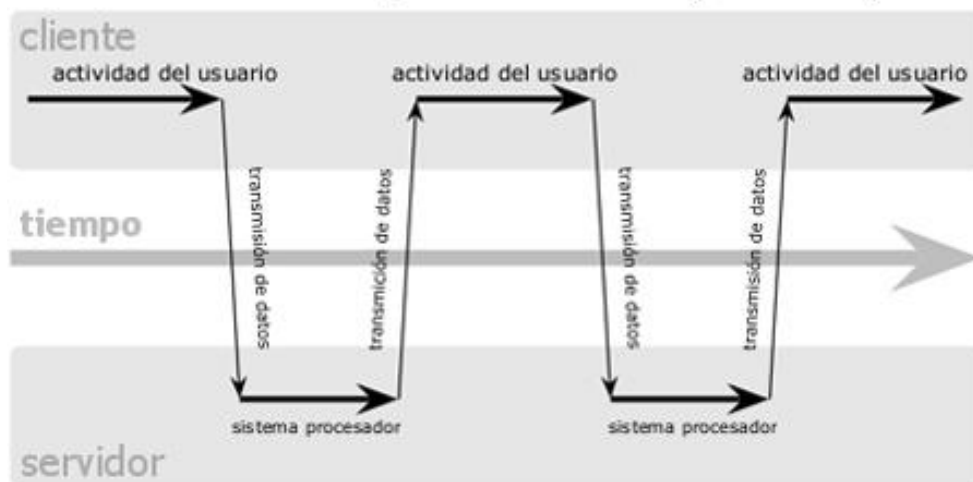
En una comunicación síncrona:

1. El usuario realiza una petición al servidor.
2. El servidor envía la página solicitada.

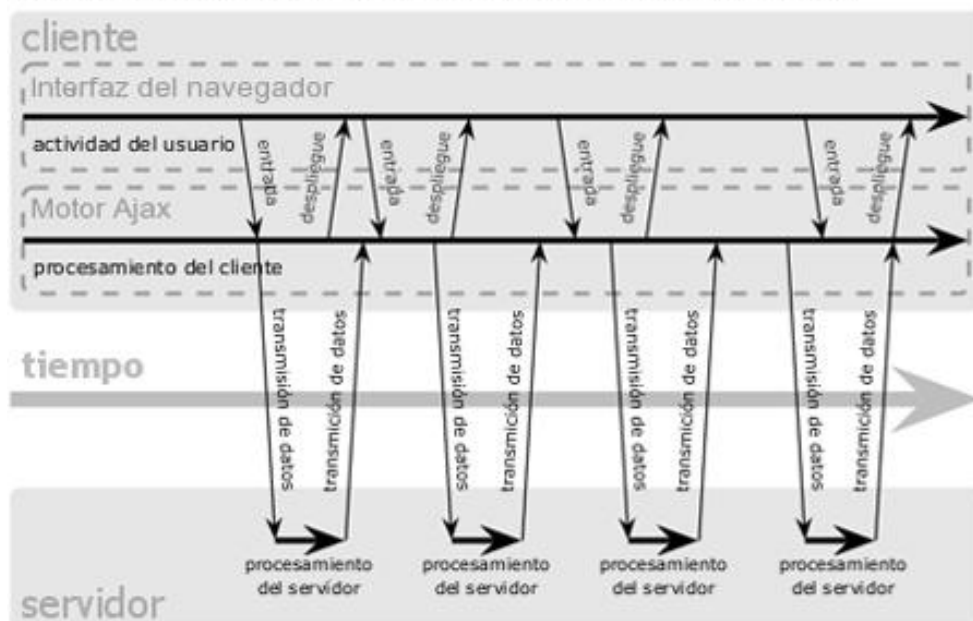
3. El usuario comienza a "leer" la página.
 4. En un momento dado, el usuario desea cambiar de página y mediante formularios o links realiza una petición al servidor volviendo al paso número 1.
 En cambio, la comunicación asíncrona no implica sincronismo ante los eventos del usuario, es decir:

1. El usuario realiza una petición al servidor.
2. El servidor envía la página solicitada.
3. El usuario comienza a "leer" la página.
4. En un momento dado, el usuario quiere alterar la información que contiene la página. En ese momento, la aplicación teniendo definidos los eventos posibles actúa en consecuencia a la interacción que haya podido suceder. De esta manera podremos modificar la página para satisfacer al usuario sin necesidad de cambiar de página.

modelo clásico de aplicaciones web (síncrono)



modelo Ajax de aplicaciones web (asíncrono)

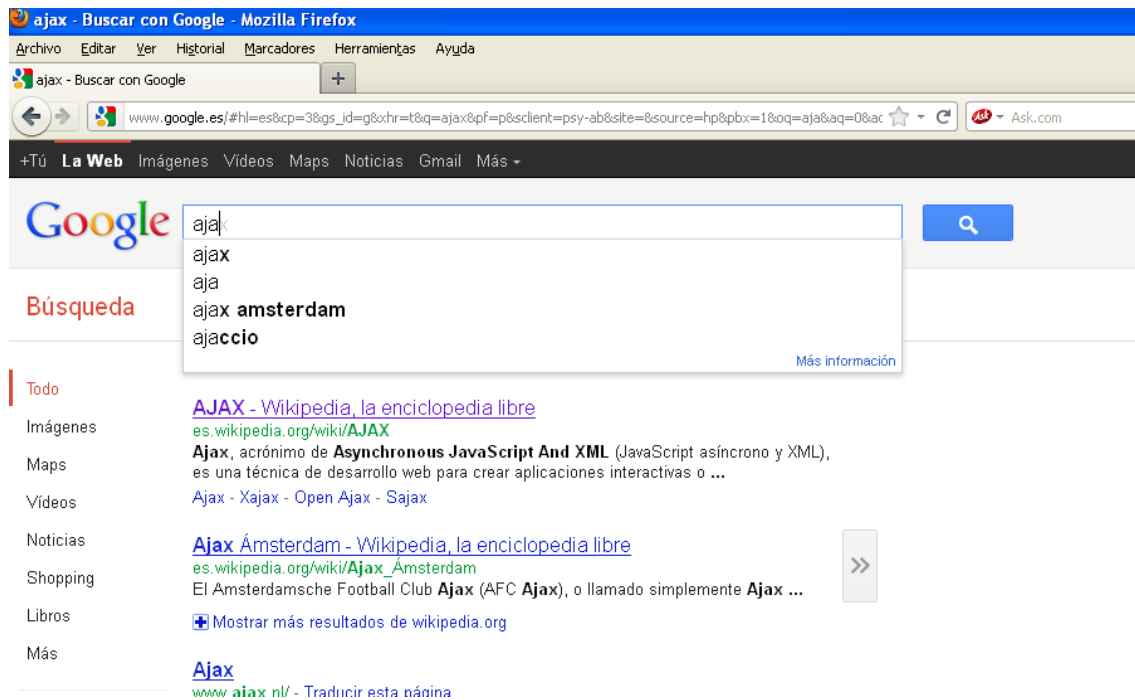


Es por esto que una aplicación ajax termina con la idea "arrancar-frenar-arrancar frenar". Cada interacción de un usuario generaría un requerimiento HTTP para po-

der mostrar la nueva información de la página, pero mediante ajax las peticiones al servidor pueden dejarse al "motor ajax" siendo este el que procesaría las peticiones contra el servidor, permitiendo o no al usuario interactuar con la página mientras el servidor está procesando la información, evitando de este modo los tiempo de carga de composición de la nueva página mientras recibe la información del servidor. Las peticiones HTTP al servidor se sustituyen por peticiones Javascript. Por tanto, la magia de ajax consiste en la posibilidad de poder, ante cualquier evento, ejecutar código JavaScript pudiendo alterar la estructura (DOM) de la página.

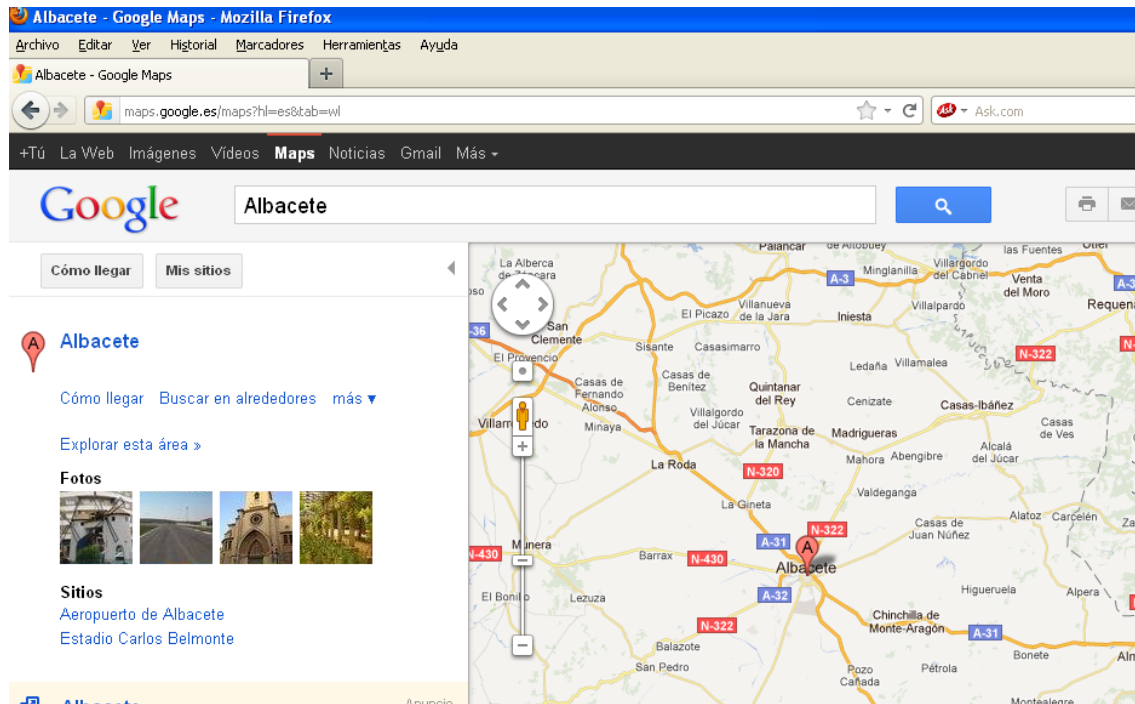
Ejemplos prácticos

1) Google suggest.



Google Suggest permite comprobar instantáneamente el número de resultados que una búsqueda va a tener mientras se está escribiendo la misma. Este ejemplo demuestra la utilidad de ajax para realizar búsquedas rápidas (ej: Datos personales, números de teléfono) o solucionar problemas a la hora de auto completar campos específicos dentro de formularios.

2) Google Maps.



Google Maps es una herramienta mediante la cual podemos consultar mapas y fotos por satélite de gran resolución de cualquier parte del mundo a una velocidad con la que muchas aplicaciones de escritorio no pueden competir. Ajax es empleado en google maps para poder recargar pequeños cuadros de imagen que forman el plano visible.

Ventajas / Desventajas

Como principales ventajas de Ajax podríamos citar las siguientes:

- Ajax es independiente del tipo de tecnología que el servidor utilice.
- El tiempo de espera cuando un usuario hace una petición al servidor se reduce, ya que el servidor no envía toda la página.
- El sitio con Ajax usa más recursos en el servidor.
- No requiere plugins o capacidades específicas en los navegadores.
- Se reduce mucho el tamaño de la información intercambiada.

Desventajas:

- "Se necesita un navegador que soporte javascript". Hoy por hoy, la práctica totalidad de navegadores soportan Ajax
- Un uso indebido de Ajax puede sobrecargar al servidor de peticiones.
- Para desarrollar algo en Ajax, se necesitan programadores con conocimientos de todas las tecnologías involucradas en Ajax.

Conclusión

Visto lo visto, cabe la pregunta ¿hasta qué punto merece la pena Ajax? Para casos puntuales y específicos sí, para todas las web no. No debemos olvidar que Ajax es una opción eficiente y rápida de navegar por la web, pero no es la única forma de navegar.

Ajax debería utilizarse en aquellas web, que por su idiosincrasia quieran emular una aplicación, un programa.

SOCIEDAD DE LA INFORMACION

www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe

Jefe de publicaciones: Antero Soria Luján

D.L.: AB 293-2001

ISSN: 1578-326x