
Estado del Arte

AJAX

versión 1.0

Preparado por:

Angelo Ortega Ramírez 2017080055
José Evelio Castro Quesada 2016086968
José Gómez Casasola 2016095929
Alejandro Tapia Barboza 2016167784

18 de Diciembre del 2019

AJAX	3
Historia	3
Tecnologías incluidas	3
Desventajas y Problemas	3
Soporte de AJAX	3
Navegadores que lo permiten	3
Navegadores que no lo permiten	3
Propuesta del Ejemplo	4
Bibliografía	4

AJAX

AJAX significa Asynchronous JavaScript And XML, la cual es una técnica de desarrollo web para la elaboración de aplicaciones interactivas o RIA (Rich Internet Applications). Este tipo de aplicaciones se ejecutan en el cliente mientras mantiene una comunicación asíncrona, en segundo plano, con el servidor. Es una técnica válida para múltiples plataformas, sistemas operativos y navegadores ya que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Con esta técnica es posible realizar cambios sobre las páginas sin necesidad de recargarlas, con lo que se incrementa la velocidad, usabilidad e interactividad en las aplicaciones web. Es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

Las funciones de llamada de AJAX se ejecutan comúnmente mediante JavaScript, mientras que el acceso a los datos mediante XMLHttpRequest.

Historia

AJAX surge de una necesidad de obtener datos de manera dinámica en páginas web. Anteriormente se tenía que recargar la página cada vez que se realizaba una consulta a base de datos, proceso que era sumamente ineficiente dado que incluso los datos estáticos de la página web eran recargados. AJAX se aprovecha de la introducción de tags con capacidades asíncronas de HTML como el iframe o el object.

Tecnologías incluidas

AJAX utiliza la combinación de las siguientes tecnologías:

- HTML y CSS para el diseño que presenta la información (interfaz gráfica)
- Document Object Model, accedido con lenguajes como JavaScript y JScript, con el fin de poder interactuar y mostrar dinámicamente la información presentada.
- El objeto XMLHttpRequest para realizar el intercambio asíncrono de datos con el servidor web.
- XML, es generalmente utilizado para transferir los datos al servidor, aunque también se puede utilizar texto plano, JSON y hasta EBML.

Ventajas

- Mejor experiencia de usuario. Ajax permite que las páginas se modifiquen sin tener que volver a cargarse, dándole al usuario la sensación de que los cambios se producen instantáneamente. Este comportamiento es propio de los programas de escritorio a los que la mayoría de los usuarios están más acostumbrados. La experiencia se vuelve mucho más interactiva.
- Optimización de recursos. Al no re-cargarse la página se reduce el tiempo implicado en cada transacción. También se utiliza menos ancho de banda.

- Alta compatibilidad. Ajax es soportado por casi todas las plataformas Web.

Desventajas

- Problemas de acceso. Normalmente, si un usuario refina una consulta a una base de datos a través de muchos criterios (por ejemplo, categoría, precio, forma de pago, etc.), la página se recargará con una URL que reflejará los parámetros ingresados. El usuario puede guardar esa URL para volver a acceder a los resultados ya filtrados fácilmente. Pero con Ajax la URL no se modifica ante la consulta, por lo que deberemos volver a ingresar cada filtro manualmente cuando queramos recuperar los resultados deseados. Existen métodos para modificar este comportamiento, pero agregan dificultad al desarrollo y peso al sitio.
- Problemas de SEO. Los buscadores tienen dificultades al analizar el código escrito en JavaScript. El hecho de que se no se generen nuevas URL elimina un importante factor de posicionamiento.
- Dificultad. Las aplicaciones con Ajax suelen requerir de un mayor tiempo de desarrollo.

Soporte de AJAX

Navegadores que lo soportan

- Mozilla
- Safari
- Chrome
- Opera (8 en adelante)
- Internet Explorer (5 en adelante)

Navegadores que no lo soportan

- Opera (anteriores a 7)
- Explorer (anteriores a 5)
- Safari (anteriores a 1.2)
- Dillo

Propuesta del Ejemplo

Se hará un ejemplo «*Hello World*» en la que los archivos en cuestión pertenecen obligatoriamente al mismo servidor, es decir, no se podrá hacer uso de información de terceros en nuestro sitio. Además, se podría presentar problemas parciales o totales en navegadores antiguos y navegadores de la familia de Internet Explorer.

Los pasos previos antes de codificar el ejemplo son:

- 1) Descargar e instalar Node.js para poder crear un servidor local (Puede ser otro entorno)
- 2) Instalar el paquete http-server —se recomienda de manera global— con el comando: `npm install -g http-server`.

- 3) Crear un archivo con extensión .txt (funcional para este ejemplo) y escribir en el: "Hola mundo".
- 4) Crear un archivo con extensión .html.
- 5) Contener ambos archivos en la misma carpeta (ver imagen 1).

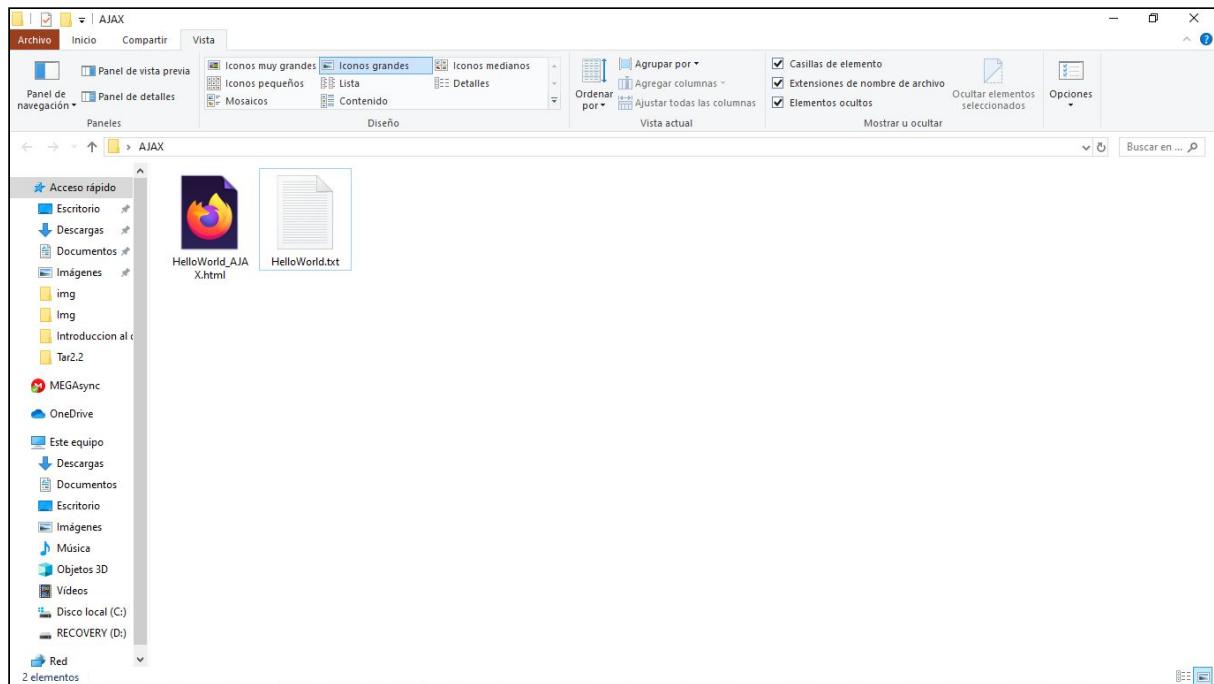


Imagen 1

Indicaciones para el .html:

- 1) Indicar los tag: `<html></html>`
- 2) Dentro de los tag `<html></html>`, dar formato al título agregando:
`<html><head><title>Ejemplo-AJAX</head>`
- 3) Dentro de los tag `<html></html>`, agregar: `<body><h2>Ejemplo con AJAX</h2><script></script>` (Ver imagen 2)
- 4) Dentro de los tag `<script></script>` (práctica no recomendada dentro de un archivo html) se realizan las siguientes acciones:
 - a) Instanciar un objeto XMLHttpRequest:
`connection = new XMLHttpRequest();`
 - b) Indicar un manejador de eventos para la información entrante:
`connection.onreadystatechange = function() {
 if (this.readyState == 4) { //El 4 indica operación finalizada.
 alert(this.responseText);
 }
 };`
 - c) Hacer la llamada:
`connection.open("GET", "HelloWorld.txt");
 connection.send(null);`
 (Ver imagen 3)
- 5) En la carpeta donde se contienen los archivos se abre una terminal y se corre un servidor con el comando: `http-server`.

- 6) En el navegador de preferencia se busca: *http://localhost:8080/*, y se abrirá la página tal como se muestra en la imagen 4, concluyendo el ejemplo.

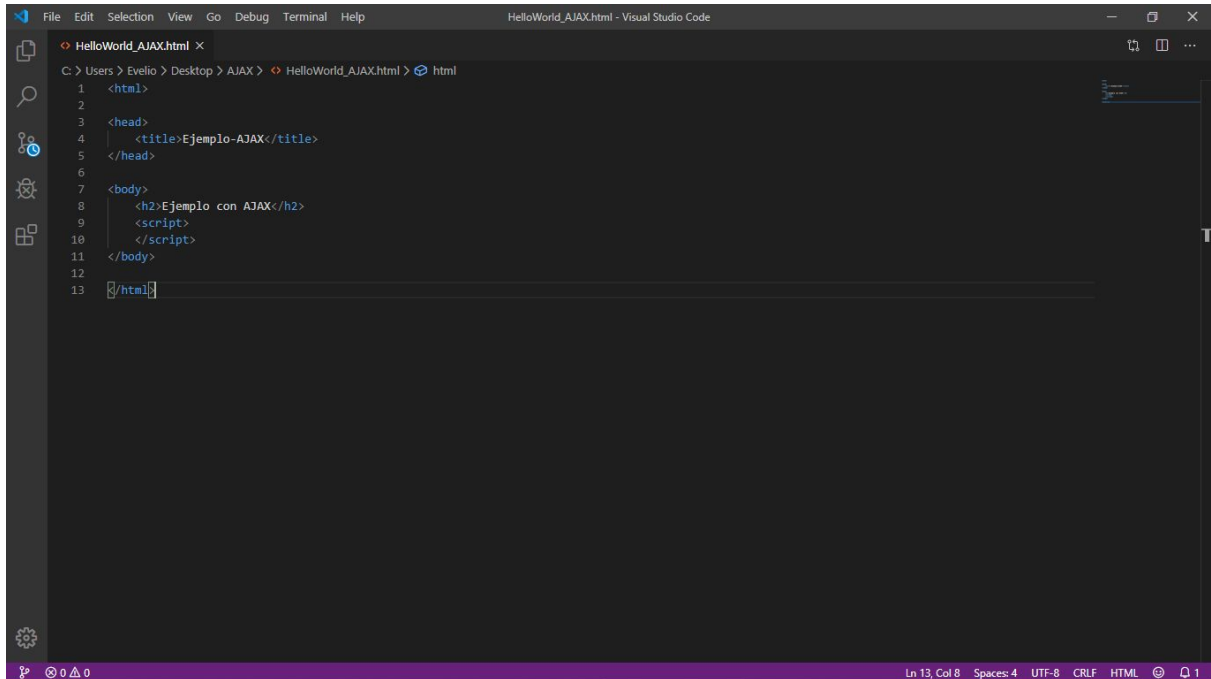


Imagen 2

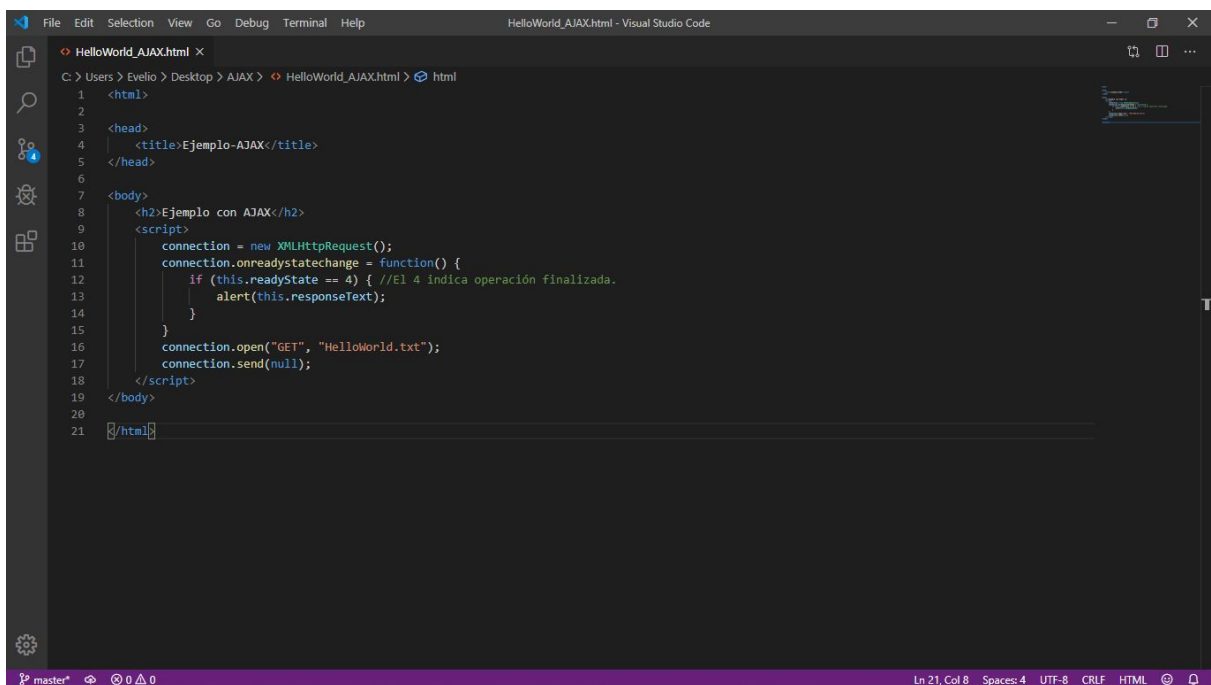


Imagen 3

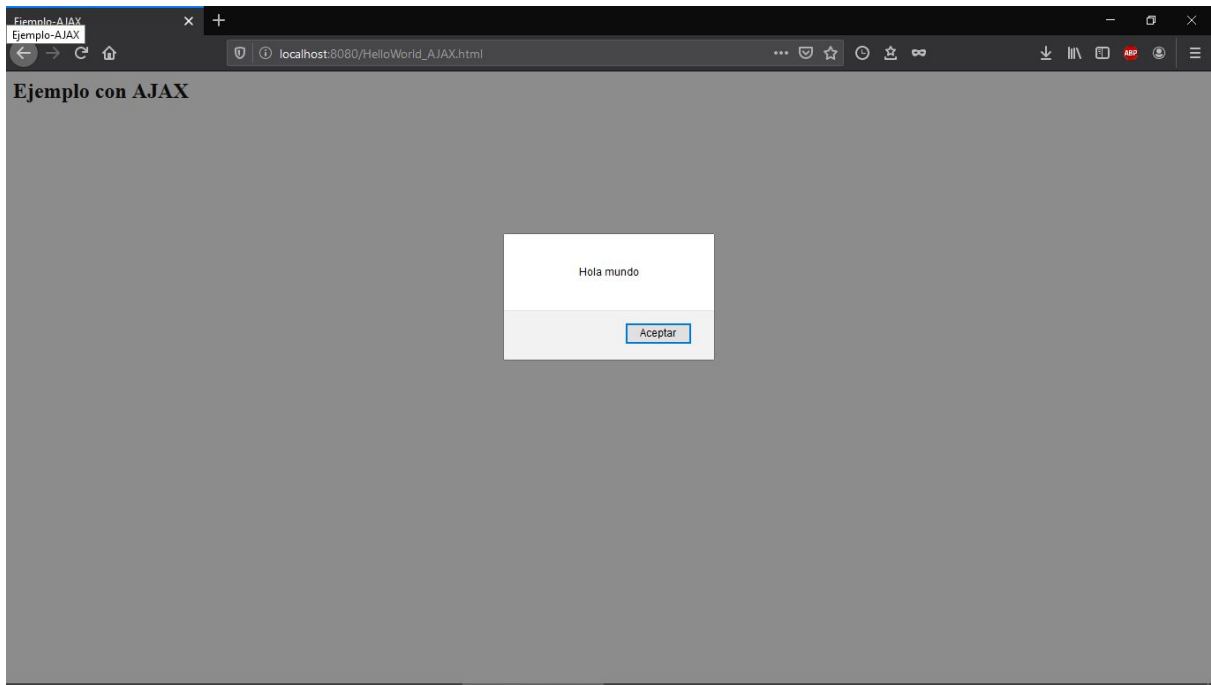


Imagen 4

Bibliografía

- [1] Anyelguti, A. (2018, 13 julio). Qué es Ajax. Recuperado 18 diciembre, 2019, de https://aprende-web.net/progra/ajax/ajax_1.php
- [2] Digital Learning. (2017, 14 abril). ¿Qué es AJAX? Recuperado 18 diciembre, 2019, de <https://www.digitallearning.es/blog/que-es-ajax/>
- [3] Universidad de Alicante. (s.f.). AJAX. Servicio de Informática Mootools. Recuperado 18 diciembre, 2019, de <https://si.ua.es/es/documentacion/mootools/ajax.html>