

LENGUAJES INTERPRETADOS EN EL CLIENTE



INTRODUCCIÓN A AJAX

- AJAX (Asynchronous JavaScript And XML), es una técnica que hace referencia a un conjunto de tecnologías de uso común en la Web que, combinadas adecuadamente, permiten mejorar la experiencia del usuario con las aplicaciones de Internet. No es un lenguaje de programación y tampoco es una tecnología en sí mismo. Necesitamos un servidor al que enviarle peticiones y obtener información.
- Con AJAX se consigue una navegación ágil, rápida y dinámica.
- AJAX permite comunicarse con sistemas remotos y/o refrescar partes de una página si necesidad de recargar la página.

USOS

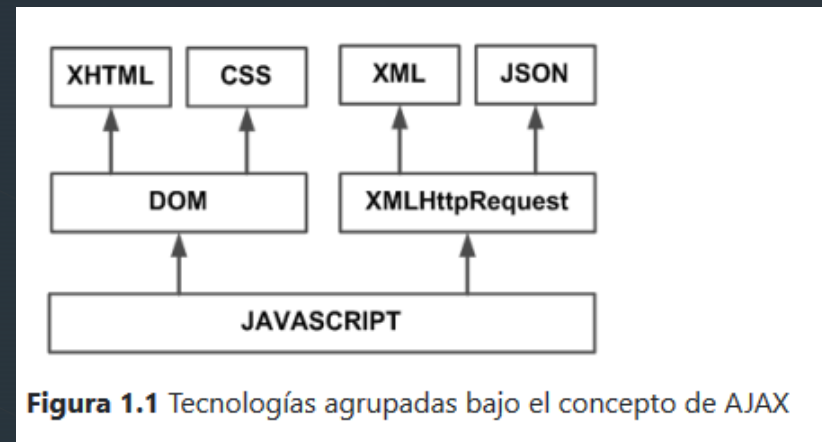
- Comunicación entre usuarios
- Diseño de formularios
- Filtrado y manipulación de datos o resultados de búsqueda
- Autocompletado (Local y remoto)
- Lista desplegable
- Menús dinámicos
- Personalización de página
- Etc.

Dónde no utilizar AJAX

- Envíos a través de formularios simples
- Búsquedas Navegación básica
- Reemplazar grandes cantidades de texto
- Manipulación de la interfaz
- Validación de usuarios

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.




¿Qué puede hacer un desarrollador con AJAX?

- Actualizar una página web sin recargar la página
- Solicitar datos de un servidor, después de que se haya cargado la página
- Recibir datos de un servidor, después de que la página se haya cargado
- Enviar datos a un servidor, en segundo plano

Conceptos Clave

GET vs. POST

- Los dos métodos HTTP más comunes para enviar una petición a un servidor son GET y POST.
- El método GET debe ser utilizado para obtener datos del servidor, pero no modificando. Por otro lado, las solicitudes GET pueden ser almacenadas en la cache del navegador, pudiendo conducir a un comportamiento impredecible si no se lo espera. Generalmente, la información enviada al servidor, es enviada en una cadena de datos (en inglés *query string*).

- 
- El método POST debe ser utilizado para operaciones en donde se está incorporando información al servidor. Por otro lado, este tipo de método no se guarda en la cache del navegador. Además, una cadena de datos puede ser parte de la URL, pero la información tiende a ser enviada de forma separada

Tipos de datos

text

Para el transporte de cadenas de caracteres simples.

html

Para el transporte de bloques de código HTML que serán ubicados en la página.

script

Para añadir un nuevo *script* con código JavaScript a la página.

json

Para transportar información en formato JSON, el cual puede incluir cadenas de caracteres, vectores y objetos.

Es recomendable utilizar los mecanismos que posea el lenguaje del lado de servidor para la generación de información en formato JSON.

jsonp

Para transportar información JSON de un dominio a otro.

xml

Para transportar información en formato XML.

A pesar de los diferentes tipos de datos de que se puede utilizar, es recomendable utilizar el formato JSON, ya que es muy flexible, permitiendo por ejemplo, enviar al mismo tiempo información plana y HTML.

VENTAJAS DE AJAX


Sus ventajas son:

- Mejora la experiencia del usuario
- Puede ser utilizada en cualquier plataforma y navegador
- Menor transferencia de datos cliente/servidor
- Optimización de recursos (tiempo de operaciones)
- Portabilidad y usabilidad (permite realizar una petición de datos al servidor y recibirla sin necesidad de cargar la página entera)

DESVENTAJA DE AJAX

Sus desventajas son:

- JavaScript debe estar activado en el navegador web para funcionar
- Utilizar AJAX preferentemente en formularios de contacto, validación de correo electrónico y contraseñas para no afectar el posicionamiento web (SEO)
- Tiempos de respuesta del servidor en momentos determinados
- Tiempo de desarrollo (es necesario tener conocimiento medio/alto de las tecnologías que hacen parte de AJAX)
- Algunas funciones a las que estamos acostumbrados en la navegación web pueden no funcionar como esperamos. Por ejemplo, el botón de atrás, guardar marcador o actualizar la página web en cualquier momento.

- 
- El punto débil de AJAX es que hay que invertir tiempo para escribir código para cada uno de los navegadores que utilizaremos, pero con jQuery se nos permite utilizar AJAX con la siguiente línea de código:
 - **`$.ajax({name:value, name:value, ... })`**

Navegadores que no permiten AJAX

- Opera 7.0 y anteriores, versión actual 65 (Enero de 2020).
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores.
- Anteriores a Safari 1.2.
- Dillo.
- Navegadores basados en texto como Lynx (navegador)|Lynx y Links.
- Navegadores para personas con capacidades especiales visuales (Braille (lectura)|Braille).
- Algunos navegadores de teléfonos móviles.
- Navegador de la PlayStation Portable|PSP.


¿Qué es la librería jQuery?

- **jQuery es una librería de código abierto de JavaScript.** Que simplifica la tarea de programar en JavaScript y **permite agregar interactividad a un sitio web** sin tener conocimientos del lenguaje.

MÉTODOS jQuery para realizar peticiones AJAX al servidor

Method	Description
<u>\$.ajax()</u>	Performs an async AJAX request
<u>\$.ajaxPrefilter()</u>	Handle custom Ajax options or modify existing options before each request is sent and before they are processed by \$.ajax()
<u>\$.ajaxSetup()</u>	Sets the default values for future AJAX requests
<u>\$.ajaxTransport()</u>	Creates an object that handles the actual transmission of Ajax data
<u>\$.get()</u>	Loads data from a server using an AJAX HTTP GET request
<u>\$.getJSON()</u>	Loads JSON-encoded data from a server using a HTTP GET request
<u>\$.parseJSON()</u>	Deprecated in version 3.0, use <u>JSON.parse()</u> instead. Takes a well-formed JSON string and returns the resulting JavaScript value
<u>\$.getScript()</u>	Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request
<u>\$.param()</u>	Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests)
<u>\$.post()</u>	Loads data from a server using an AJAX HTTP POST request
<u>ajaxComplete()</u>	Specifies a function to run when the AJAX request completes
<u>ajaxError()</u>	Specifies a function to run when the AJAX request completes with an error
<u>ajaxSend()</u>	Specifies a function to run before the AJAX request is sent
<u>ajaxStart()</u>	Specifies a function to run when the first AJAX request begins
<u>ajaxStop()</u>	Specifies a function to run when all AJAX requests have completed
<u>ajaxSuccess()</u>	Specifies a function to run when an AJAX request completes successfully
<u>load()</u>	Loads data from a server and puts the returned data into the selected element
<u>serialize()</u>	Encodes a set of form elements as a string for submission
<u>serializeArray()</u>	Encodes a set of form elements as an array of names and values

Ver ejemplos en: https://www.w3schools.com/jquery/jquery_ref_ajax.asp

- 
- Existen muchos plugins gratis que resuelven situaciones concretas dentro del maquetado de un sitio web, en la web hay cientos de blogs que recopilan y analizan los plugins según sus funcionalidades, reuniendo en un sólo post los links a varios plugins de función similar, lo que facilita mucho la búsqueda.

- Ver los pasos para instalar un plugin en el siguiente enlace:

<https://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>

- Todos los plugins de JQuery necesitan la librería de JQuery para funcionar. La librería es un archivo .js que se puede descargar desde el sitio oficial:

<https://jquery.com/> colocar en una carpeta js y luego vincular con una etiqueta `<script>`

- EJEMPLO:

Colocamos dentro del body:

```
<script src="js/jquery-3.5.1.min.js"></script>
```

Si descargamos el archivo y lo vinculamos de esa manera, debemos subir el archivo a nuestro hosting.

Otra forma de vincular la librería es utilizando el servidor de Google. De esta manera no necesitamos descargarla ni subirla a nuestro servidor. En ese caso, el código es el mismo pero con ruta absoluta:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jq  
uery.min.js"></script>
```

SINTAXIS DEL MÉTODO \$.ajax()

```
$.ajax(url, { objeto-configurable } ); // o $.ajax( { objeto-configurable } );
```

El objeto configurable contendrá uno o varios de los parámetros siguientes:

- type : tipo de la petición, GET o POST (GET por defecto).
- url : dirección a la que se envía la petición.
- data : datos a enviar al servidor.
- dataType : tipo de datos que se espera obtener del servidor (si no se especifica, jQuery intenta averiguar de qué tipo se trata).
- success : función que se ejecuta cuando se obtiene una respuesta con éxito.
- error : función que se llama si la petición no tiene éxito.

EJEMPLO CON EL MÉTODO \$.ajax()

- La función es un bloque de código JavaScript, en esta se colocan instrucciones para el plugin.
- Se escribe dentro de las etiquetas **<script></script>**. Puede ir debajo de la estructura HTML del plugin, dentro del body o al final del documento.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
      $("#div1").html(result);
    }});
  });
});
</script>
</head>
<body>

<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

<button>Get External Content</button>

</body>
</html>
```


EJEMPLO CON EL MÉTODO \$.get()

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.get("demo_test.asp", function(data, status){
            alert("Data: " + data + "\nStatus: " + status);
        });
    });
});
</script>
</head>
<body>

<button>Send an HTTP GET request to a page and get the result back</button>

</body>
</html>
```


BIBLIOGRAFÍA

- AJAX - Web 2.0 para profesionales. Por FIRTMAN, Maximiliano, 2008, Alfaomega.
- AJAX EN J2EE. 2ª Edición. Por MARTIN SIERRA, Antonio J, 2011, Ra-Ma.
- w3Schools. **jQuery AJAX Methods. Recuperado de**
https://www.w3schools.com/jquery/jquery_ref_ajax.asp
- Chuburu Laura.(2020). **Qué es JQuery y cómo implementarlo. Recuperado de**
<https://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>
- **Arkaitz Garro. Capítulo 7. AJAX. Recuperado de**
<https://www.arkaitzgarro.com/jquery/capitulo-7.html>
- **S.N.(2017) Curso de Ajax con Java – Manual del alumno. Recuperado de**
<https://www.elsaber21.com/curso-de-ajax-con-java-manual-del-alumno>