

# 1.A. Arquitecturas y lenguajes de programación en clientes web

## 2.- Lenguajes de programación en clientes web.

### 2.1.- Características

Como vimos anteriormente, los lenguajes de programación para clientes web no son un reemplazo de la programación en el lado del servidor. Cualquier web que reaccione dinámicamente a interacciones del usuario o que almacene datos, estará gestionada por lenguajes de script en el lado del servidor, incluso aunque usemos JavaScript en el cliente para mejorar la experiencia de usuario. Las razones son simples:



- **Primero:** JavaScript por sí mismo no puede escribir ficheros en el servidor. Puede ayudar al usuario a elegir opciones o preparar datos para su envío, pero después de eso solamente podrá ceder los datos al lenguaje de servidor encargado de la actualización de datos.
- **Segundo:** no todos los clientes web ejecutan JavaScript. Algunos lectores, dispositivos móviles, buscadores, o navegadores instalados en ciertos contextos están entre aquellos que no pueden realizar llamadas a JavaScript, o que simplemente son incompatibles con el código de JavaScript que reciben. Aunque esto ocurra nuestra página web debería ser completamente funcional con JavaScript desactivado. Utilizaremos JavaScript para conseguir que la experiencia de navegación web sea lo más rápida, moderna o divertida posible, pero no dejaremos que nuestra web deje de funcionar si JavaScript no está funcionando.
- **Tercero:** uno de los caminos que más ha integrado la programación cliente con la programación servidor ha surgido gracias a Ajax. El proceso asíncrono de AJAX se ejecuta en el navegador del cliente y emplea JavaScript. Este proceso se encarga de comunicarse con el servidor antes de enviar la una petición mediante el envío de datos en formato XML, JSON o texto y todo ello de forma transparente en para el usuario. Los datos devueltos por el servidor pueden ser examinados por JavaScript en el lado del cliente, para actualizar secciones o partes de la página web. Es así como funcionan la gran parte de las aplicaciones Web actuales.



JavaScript está orientado a dar soluciones a:

- **Conseguir** que nuestra página web responda o reaccione directamente a la interacción del usuario con elementos de formulario y enlaces hipertexto.
- La **distribución** de pequeños grupos de datos y proporcionar una interfaz amigable para esos datos.
- **Controlar** múltiples ventanas o marcos de navegación, plug-ins, o applets Java basados en las elecciones que ha hecho el usuario en el documento HTML.
- **Pre-procesar** datos en el cliente antes de enviarlos al servidor.
- **Modificar** estilos y contenido en los navegadores de forma dinámica e instantáneamente, en respuesta a interacciones del usuario.
- **Solicitar** ficheros del servidor, y enviar solicitudes de lectura y escritura a los lenguajes de servidor.

El lenguaje JavaScript cumple con el estándar ECMAScript (o ES) que es una especificación de lenguajes de scripting estandarizada por Ecma International en los documentos ECMA-262 e ISO/IEC 16262. La edición actual de este estándar es la 6 también conocida como ES6 o ES2015.

Existen otros lenguajes de scripting que permiten compilar a JavaScript y que aportan una sintaxis más reducida, compacta u orientada a programación funcional que compilan a JavaScript y que pueden ser también utilizados para escribir programas en el lado cliente. Algunos ejemplos de estos lenguajes son TypeScript, CoffeeScript, LiveScript, etc...

Los lenguajes de script como JavaScript no se usan solamente en las páginas web. Los intérpretes de JavaScript están integrados en múltiples aplicaciones de uso cotidiano. Estas aplicaciones proporcionan su propio modelo de acceso y gestión de los módulos que componen la aplicación y para ello comparten el lenguaje JavaScript en cada aplicación. Podríamos citar varios ejemplos como: Google Desktop Gadgets, Adobe Acrobat, Dreamweaver, OpenOffice.org, Google Drive, etc.

El uso de estos lenguajes también está cada vez más extendido en el desarrollo en el servidor. Proyectos como NodeJS permiten la creación de aplicaciones escritas en JavaScript. Al usar el mismo lenguaje en el cliente y en el servidor y usar un formato como JSON basado en JavaScript, la construcción de aplicaciones usando NodeJS y los frameworks que bajo el se desarrollan aumentan notablemente la productividad al trabajar con un solo lenguaje en todas las capas de desarrollo.



Créditos de las imágenes:

JS

Autoría: Chris Williams

Dominio público

Ajax

Autoría: Gummie

Licencia: Creative Commons - ShareAlike

NodeJS

Autoría: NodeJS

Licencia: <https://nodejs.org/static/documents/trademark-policy.pdf>