

# Trabajo sobre Arquitecturas y Tecnologías Cliente

24/10/2022

---

Alfredo Martín Ruiz

Desarrollo Web Entorno Cliente

UT1. JavaScript

I.E.S. Martínez Montañés

## ÍNDICE

1 Modelos de programación en entornos cliente/servidor

2 Mecanismos de ejecución de código en un navegador Web. Capacidades y limitaciones de ejecución. Compatibilidad con navegadores Web.

3 Lenguajes de programación en entorno cliente. Tecnologías asociadas.

4 Características de los lenguajes de script. Ventajas y desventajas.

5 Integración del Javascript con las etiquetas HTML y CSS.

6 Herramientas de programación para Javascript.

## MODELOS DE PROGRAMACIÓN EN ENTORNOS CLIENTE/SERVIDOR

Los modelos arquitectónicos de programas se pueden clasificar en función de en qué lugar se ejecutan, si en el cliente o en el servidor.

- **Capa de Presentación:** lo ve el usuario y suele estar situada en el cliente. Presenta una interfaz gráfica del recurso solicitado y sirve para recoger su interacción. La programación de esta capa se basa en el formateo de la información enviada por el servidor y la captura de las acciones realizadas por el cliente.
- **Capa de Negocio:** puede ser programada tanto en el cliente como en el servidor. Gestiona las funcionalidades que esperamos del sistema o aplicación web.
- **Capa de Datos:** está formada por 1 o más gestores de bases de datos que administran los datos. En esta capa residen los datos y se pueden acceder a ellos desde aquí mismo.

Estos modelos de programación cliente/servidor también se pueden clasificar dependiendo del tamaño de los componentes, por la naturaleza del servicio ofrecido o según el reparto de funciones entre cliente y servidor:

- **Según el Tamaño de los Componentes:** hace referencia a qué elemento de la arquitectura web debe soportar más o menos carga de procesamiento. En la configuración Fat Client, el mayor peso de la aplicación se ejecuta desde el cliente, en cambio el Fat Server es justo lo contrario.
- **Según la Naturaleza del Servicio:** podemos encontrar servidores de ficheros donde el objetivo del cliente es el acceso a datos contenidos en ficheros, servidores de bases de datos que se centran en la provisión y administración de sistemas gestores de bases de datos.

- **Reparto de Funciones entre Cliente y Servidor:** las distintas tecnologías web permiten gestionar las responsabilidades de todas las prestaciones funcionales entre el cliente y el servidor.

## **MECANISMOS DE EJECUCIÓN DE CÓDIGO EN UN NAVEGADOR WEB. CAPACIDADES Y LIMITACIONES DE EJECUCIÓN. COMPATIBILIDAD CON NAVEGADORES WEB**

Cuando nuestro navegador solicita una página a un servidor web, antes se deberá ejecutar algún programa para obtenerla, este código se ejecutará en el servidor web.

Cuando esta llega a nuestro navegador, es posible que incluya algún fragmento de código que se deban ejecutar, que generalmente será de lenguaje JavaScript, que se ejecutará en nuestro navegador.

Estas 2 tecnologías se complementan una con otra, por ejemplo, un programa que se encarga de obtener mensajes de una base de datos se ejecuta en el entorno del servidor, mientras que nuestro navegador ejecuta un código cualquiera, por ejemplo, uno que nos avisa cuando queremos enviar un mensaje y nos hemos olvidado de escribir el asunto.

Una web es compatible con los navegadores cuando se ve igual o de forma similar en todos ellos. No es una tarea fácil, pero conseguir esto en Chrome, Firefox, Opera, Safari y Mozilla es razón para estar satisfecho.

El problema principal es que no todos los navegadores interpretan HTML y CSS de la misma manera. Una buena forma de mejorar la compatibilidad con navegadores sería validar el código a los estándares del W3C. También sería conveniente un código limpio, válido y simple.

## LENGUAJES DE PROGRAMACIÓN EN ENTORNO CLIENTE. TECNOLOGÍAS ASOCIADAS.

Los lenguajes de lado cliente son aquellos que pueden ser directamente procesados por el navegador sin falta de un tratamiento anterior. Los más conocidos son:

- **HTML:** indica al navegador dónde colocar cada texto, imagen, vídeo... El lenguaje consta de etiquetas con forma **<b>**. Cada etiqueta tiene su correspondiente etiqueta de cierre, menos algunas.
- **JavaScript:** sirve para crear programas que realizan acciones dentro de una página web. Su uso se basa en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.
- **Flash:** es una tecnología para crear efectos especiales en páginas webs. También conseguimos hacer páginas dinámicas del lado del cliente.
- **CSS:** permite dar estilo a una página con código HTML, como incluir márgenes, tipos de letra, colores...

Los **frameworks** de Javascript son una tecnología esencial para el desarrollo web front-end moderno, sirven para crear aplicaciones web interactivas. Cada framework tiene un enfoque diferente para actualizar el DOM y manejar los eventos del navegador. Principales ventajas:

- **Mantenible:** se pueden realizar cambios rápidos sin generar mucho impacto en el resto de la aplicación.
- **Más rápido:** el uso de frameworks hace que todo el desarrollo sea mucho más rápido desde el primer día.

## CARACTERÍSTICAS DE LOS LENGUAJES DE SCRIPT.

Ventajas de los lenguajes de script:

- **Interpretados:** no necesita ser compilado, basta con que sea interpretado por un programa.
- **Declaración de variables:** no necesitan la declaración de variables para asignar su tipo. El tipo de dato de cada variable es deducido por el contexto en el que se utiliza.
- **Utilizan instrucciones:** fundamentales para que el intérprete ejecute las acciones que quiere realizar.
- **Código simple:** son códigos cortos y concisos que realizan una acción concreta, se busca la simplicidad y eficiencia.

Desventajas de los lenguajes de script:

- **Soporte del navegador:** es difícil que todos los navegadores soporten el código de la misma manera, suele haber algunas diferencias mínimas.
- **Seguridad lado del cliente:** cuando se ejecuta el código en el cliente, puede dar lugar a bugs y descuidos que a veces vienen con malos propósitos.

## INTEGRACIÓN DEL JAVASCRIPT CON LAS ETIQUETAS HTML Y CSS.

La combinación entre estos 3 lenguajes es el fundamento de la web como la entendemos hoy día. HTML contiene todo el código principal, CSS se encarga de darle estilo y JavaScript añade dinamismo y distintas funcionalidades a la web.

Hay varias formas de integrar código CSS y JavaScript a una página HTML.

Primeramente se puede hacer de forma externa, con un fichero con extensión .js o .css y lo vinculamos con el .html mediante la etiqueta <link>.

En segundo lugar, se puede hacer implícitamente con una etiqueta <style> (para CSS) o <script> (para JavaScript) dentro de la etiqueta <head> del .html.

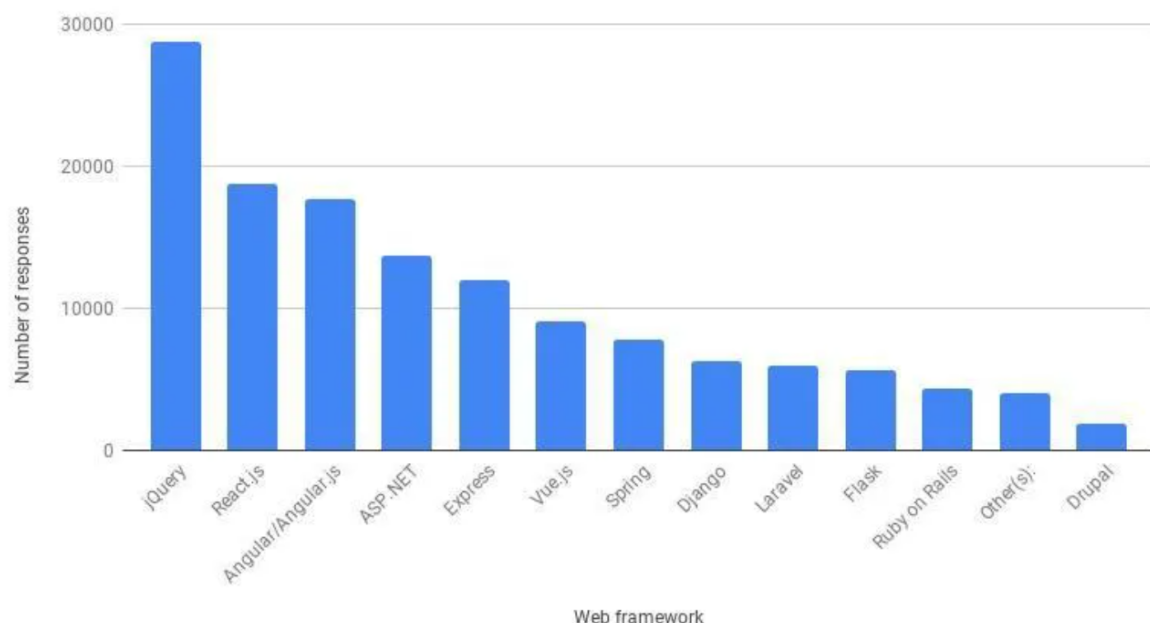
Por último, dentro de los elementos html podemos aplicar estilos con el atributo style.

## HERRAMIENTAS DE PROGRAMACIÓN PARA JAVASCRIPT.

Hay varias herramientas útiles para desarrolladores de código JavaScript:

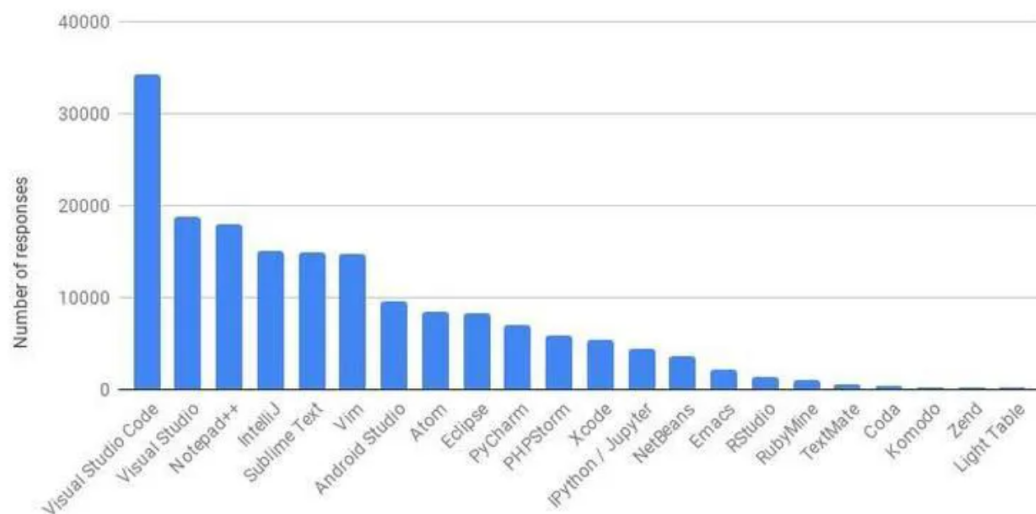
- **Frameworks:** lo hemos mencionado antes, aporta un código mantenible y que se puede desarrollar en menos tiempo. Angular o Ember son unos de los más usados.

JavaScript developers most popular web framework



- **Librerías JS:** la más utilizada es JQuery, permite interactuar con el HTML, el DOM, hacer todo tipo de animaciones... tiene soporte para CSS3, permite manipular totalmente las hojas CSS y soporta extensiones.
- **IDE y editores de código:** los más utilizados por los desarrolladores de JavaScript son Visual Studio Code y Notepad++, son los más sencillos y útiles a la hora de programar.

JavaScript developers favorite IDEs and code editors



- **Bases de datos:** la más usada es MySQL, destaca por su velocidad y su facilidad.

JavaScript developers most popular databases

