

Comparativa de navegadores y motores de navegación

Esther Ruiz Rodríguez

2º WEB



Índice

Práctica 1	3
Chrome	4
Firefox	5
Safari	6
Edge	8
Opera	9



Realiza una investigación sobre los principales navegadores web (Chrome, Firefox, Safari, Edge y Opera)

Para cada uno:

1. Identifica el motor de ejecución que utiliza.
2. Describe brevemente sus características principales.
3. Comenta si existen diferencias en la interpretación de HTML, CSS o JavaScript y qué problemas puede ocasionar esto en el desarrollo web.
4. Incluye un ejemplo real de una web que presente diferencias visuales o de funcionamiento entre navegadores.

Producto final: un informe en PDF con una tabla comparativa y una breve reflexión personal sobre cuál consideran el navegador más relevante a la hora de desarrollar.



CHROME

1. El motor de ejecución de Chrome se llama V8.
2. V8 traduce el código JavaScript directamente a código máquina para que los ordenadores puedan entenderlo, y luego ejecuta el código traducido, o compilado.

Se trata de un motor de código abierto para JavaScript y Web Assembly, creado por Google, siendo su programador jefe Lars Bak.

Este motor no solo ha sido usado del lado del cliente, sino también del lado del servidor Couchbase, MongoDB y [Node.js](#)

Está escrito en C++ y es usado en Chromium, Microsoft Edge desde 2020, Brave y Opera. También está integrado en el navegador de internet del sistema operativo Android 2.2 "Froyo". Implementa ECMAScript como especifica ECMA-262 5ª edición y corre en Windows XP, Vista, Mac OS X 10.5 (Leopard) y Linux en procesadores IA-32 y ARM.

Chrome V8 puede funcionar de manera individual (Standalone) o incorporada a cualquier aplicación C++.

Destaca por velocidad y uso en múltiples entornos (navegadores y servidores como Node.js).

3. Aunque los navegadores intentan seguir los mismos estándares (W3C, ECMA), cada motor los implementa de forma diferente. En el caso de Chrome, usa:

Blink (motor de renderización) -> interpreta HTML y CSS

V8 -> interpreta y ejecuta JavaScript

Esto puede provocar que algunas etiquetas HTML5 o propiedades CSS no se muestren igual en todos los navegadores (por ejemplo, sombras, flexbox...etc).



El motor V8 de Chrome optimiza el JavaScript de forma distinta que otros motores, lo que a veces da diferencias en el rendimiento o compatibilidad.

4. La página de Whatsapp Web en su momento no funcionaba en Internet Explorer porque dependía de APIs modernas de JavaScript (Promises, WebRTC) que V8 en Chrome ya soportaba, pero IE no.

FIREFOX

1. Mozilla Firefox utiliza un motor de ejecución llamado SpiderMonkey.
2. El motor SpiderMonkey (Firefox) ejecuta JavaScript y WebAssembly. Fue el primer motor JavaScript creado, por Netscape en 1996.

Usa varios compiladores JIT (Baseline, IonMonkey) para mejorar el rendimiento.

Suele implementar rápidamente nuevas características de ECMAScript para cumplir con los estándares.

Tiene buenas herramientas para depuración en Firefox Developer Edition.

Destaca por ser el más estándar y pionero, con foco en compatibilidad y depuración.

3. Firefox usa:

Gecko -> motor de renderizado para HTML y CSS

SpiderMonkey -> motor de ejecución de JavaScript

Interpretación de Firefox:



Algunas propiedades CSS nuevas o experimentales pueden necesitar prefijos específicos para funcionar correctamente.

Diferencias en la ejecución de JavaScript moderno (ES6+) respecto a otros navegadores si se usan APIs no implementadas todavía.

Problemas de compatibilidad con sitios desarrollados pensando solo en Chrome/V8, provocando que ciertos scripts o animaciones no funcionen igual.

Los sitios pueden verse desalineados, con elementos fuera de lugar o funcionar mal si no se prueban en Firefox.

Los desarrolladores deben usar polyfills, prefijos CSS o librerías compatibles para garantizar que el sitio funcione en todos los navegadores principales.

4. Ejemplo Youtube.

Youtube utiliza JavaScript moderno y CSS avanzado.

Funciona correctamente en Firefox gracias a SpiderMonkey y Gecko, pero en navegadores antiguos o desactualizados (por ejemplo, Internet Explorer 11) algunas funciones no funcionan.

SAFARI

1. El motor de ejecución de Safari para JavaScript se llama JavaScriptCore, también conocido como Nitro.
2. Este motor ejecuta JavaScript y WebAssembly dentro de Safari.

Utiliza compilación JIT (Just In Time) para convertir JavaScript en código máquina y mejorar el rendimiento.



Optimiza funciones que se usan con frecuencia para acelerar la ejecución.

Implementa las últimas características del estándar ECMAScript.

Forma parte del motor de renderizado WebKit, que interpreta HTML y CSS.

Su enfoque está en rendimiento y eficiencia energética, especialmente en dispositivos Apple.

3. Safari utiliza:

WebKit -> motor de renderizado para HTML y CSS

JavaScriptCore (Nitro) -> motor de ejecución de JavaScript

Algunas propiedades CSS avanzadas o experimentales pueden necesitar prefijos.

JavaScript moderno puede ejecutarse con ligeras diferencias en comportamiento y optimización respecto a otros motores (V8 en Chrome o SpiderMonkey en Firefox).

Algunas APIs de navegador pueden tener soporte parcial o diferencias en la implementación.

Sitios web pueden verse o comportarse distinto en Safari si no se adaptan correctamente. Es necesario probar y usar prefijos CSS, polyfills y librerías compatibles para asegurar funcionalidad y estilos consistentes.

4. Ejemplo de sitio web <http://www.figma.com>

Figma es una aplicación web que depende de JavaScript avanzada y CSS moderno.

Funciona correctamente en Safari gracias a JavaScriptCore y WebKit, pero en navegadores antiguos o sin soporte completo algunas funciones pueden no actualizarse correctamente.

Animaciones o transiciones CSS pueden no mostrarse.

Comparado con Firefox o Chrome, las diferencias suelen ser de rendimiento o fluidez, no de funcionalidad total, gracias a JavaScriptCore.



EDGE

1. El motor de ejecución de Edge depende de la versión del navegador.

Edge (Chromium, versión moderna), utiliza el motor de ejecución V8 (el mismo que Chrome y Opera). Su compatibilidad es total con JavaScript moderno y WebAssembly.

Edge (versión Legacy, antes de 2020), utiliza el motor de ejecución Chakra (propio de Microsoft). Su motor de renderizado es EdgeHTML. Y su compatibilidad es limitada con algunas APIs modernas y características ES6+.

2. Microsoft Edge (chromium):

Su motor de renderizado es Blink, que interpreta HTML y CSS.

Su motor de ejecución es V8, ejecuta JavaScript y WebAssembly.

Su rendimiento es muy rápido gracias a la compilación JIT y optimizaciones dinámicas de V8.

Su compatibilidad soporta todas las características modernas de HTML5, CSS3 y ECMAScript.

Su integración es basada en Chromium, por lo que comparte compatibilidad con extensiones de Chrome.

3. En Edge (Chromium):

HTML y CSS puedes renderizar con pequeñas diferencias (márgenes, fuentes, animaciones).

JavaScript puede ejecutarse con ligeras diferencias según el motor (V8 en Edge/chrome, SpiderMonkey en Firefox, JavaScriptCore en Safari).



Algunos sitios pueden verse desalineados o romperse en navegadores distintos.

Requiere pruebas cruzadas, polyfills, prefijos CSS y librerías compatibles.

4. Ejemplo de sitio web sería <http://www.trello.com>.

Trello es una aplicación web que usa JavaScript moderno (ES6+), WebSockets y CSS avanzado (Flexbox, Grid, animaciones).

Funciona correctamente en Edge (V8+ Blink) y Chrome, pero en navegadores antiguos o menos compatibles (como Internet Explorer 11) algunas funciones fallan:

- Tableros no se actualizan en tiempo real.
- Drag y drop de tarjetas puede no funcionar.
- Animaciones de transición entre listas pueden no mostrarse.

Edge ofrece rendimiento fluido y total compatibilidad.

OPERA

1. Opera utiliza V8 como motor de ejecución de JavaScript.

Su motor de renderizado es Blink, igual que Chrome.

2. Ejecuta JavaScript y WebAssembly de forma muy rápida.

Utiliza compilación JIT (Just In Time) y optimización dinámica de funciones.

Es compatible con las últimas características de ECMAScript.

Su motor de renderizado es Blink. Interpreta HTML y CSS siguiendo los estándares web modernos.

Permite renderizado rápido y soporte para animaciones y efectos avanzados.



Su compatibilidad en general funciona casi idéntica a Chrome, permitiendo usar extensiones de Chrome y sitios web modernos sin problema.

3. Aunque Opera comparte base tecnológica con Chrome, pueden existir diferencias menores como:

Pequeñas variaciones en renderizado de CSS (márgenes, fuentes, animaciones).

Diferencias en el comportamiento de APIs experimentales de JavaScript si se usan funciones muy nuevas.

En la práctica, casi todos los sitios modernos funcionan igual que en Chrome, pero es recomendable probar en Opera por si existen pequeños desajustes visuales o de rendimiento.

El uso de prefijos CSS o polyfills puede ser necesario en casos muy específicos.

4. Un ejemplo de sitio web sería <http://spotify.com>

Spotify Web Player utiliza JavaScript avanzado, WebAssembly y CSS moderno.

Funciona correctamente en Opera gracias a V8 y Blink, pero en navegadores antiguos o menos compatibles algunas funciones fallan:

- Reproducción de audio en tiempo real puede no funcionar.
- Interfaz interactiva (menús, listas, animaciones) puede verse desalineada o sin respuesta.

En Opera moderno, la web funciona igual que en Chrome, con fluidez y todas las funcionalidades.

Las diferencias se pueden notar solo en navegadores que no usan V8/Blink, como Safari o Internet Explorer, donde algunos scripts o animaciones no funcionan.