



**Curso**

Fundamentos de Programación Web

**Entrega**

Tarea 1

**Estudiante**

Hillary Granados Mora

**Profesor**

Francisco Jose Jimenez Bonilla

**Universidad Cenfotec**

27 Mayo, 2025

II Cuatrimestre, 2025

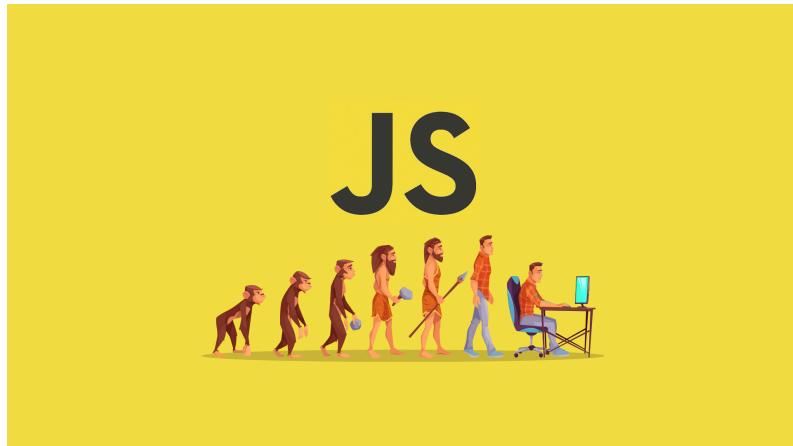
## 1. ¿Escriba la historia de Java Script?

JavaScript fue creado en 1995 por Brendan Eich, un programador de Netscape Communications. La idea surgió cuando Netscape quería un lenguaje de programación ligero que pudiera ejecutarse en los navegadores web para hacer las páginas más interactivas. En solo 10 días, Eich desarrolló un lenguaje llamado inicialmente Mocha, luego renombrado como LiveScript y finalmente llamado JavaScript, en parte por motivos de marketing para aprovechar la popularidad de Java (aunque no están directamente relacionados).

JavaScript se introdujo por primera vez en el navegador Netscape Navigator 2.0 en 1996. Sin embargo, como no existía un estándar oficial, cada navegador implementaba su propia versión, lo que causaba muchos problemas de compatibilidad. Para resolver esto, en 1997, JavaScript fue estandarizado por la organización ECMA International bajo el nombre ECMAScript.

Desde entonces, JavaScript ha evolucionado significativamente. Se han lanzado varias versiones del estándar ECMAScript, como ES5 (2009), ES6 (2015) —una de las más importantes por introducir clases, módulos y muchas mejoras al lenguaje—, y versiones anuales hasta la actualidad.

Hoy en día, JavaScript es uno de los lenguajes más utilizados en el mundo del desarrollo web, tanto en el frontend (navegador) como en el backend (con Node.js).



## 2. ¿Qué es Java Script?

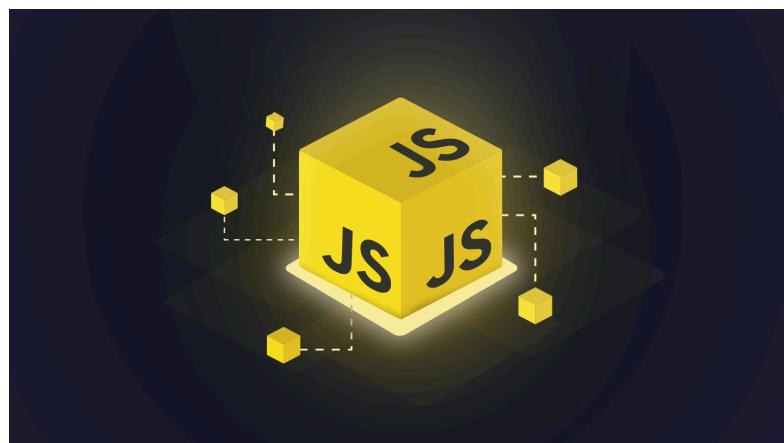
JavaScript es un lenguaje de programación interpretado, de alto nivel, y orientado a objetos que se utiliza principalmente para crear páginas web dinámicas e interactivas. Funciona del lado del cliente (navegador web), aunque también puede ejecutarse del lado del servidor mediante plataformas como Node.js.

Con JavaScript se pueden realizar tareas como:

- Validar formularios en una página web sin necesidad de recargarla.

- Crear animaciones o efectos visuales.
- Controlar eventos del usuario (clics, movimientos del ratón, teclado, etc.).
- Manipular el contenido del sitio web (DOM).
- Realizar solicitudes a servidores web sin recargar la página (AJAX).

JavaScript es uno de los tres pilares del desarrollo web, junto con HTML (estructura) y CSS (diseño). Además, es compatible con todos los navegadores modernos y es esencial para la mayoría de las aplicaciones web actuales.



### **3. ¿Cuál es la relación entre HTML y Java Script?**

HTML (HyperText Markup Language) es el lenguaje que se utiliza para estructurar el contenido de una página web, como textos, imágenes, formularios, botones, etc, mientras que JavaScript es el lenguaje que le da interactividad y dinamismo a esa estructura creada con HTML.

La relación entre ambos es muy estrecha: JavaScript necesita del HTML para actuar, y HTML se enriquece gracias a JavaScript. Por ejemplo, HTML puede definir un botón, pero JavaScript es el que le da vida al hacer que, al hacer clic, ocurra algo (como mostrar un mensaje o cambiar el color de fondo).

JavaScript puede leer, modificar o eliminar elementos HTML mediante el uso del DOM (Document Object Model), una estructura que representa la página web como un árbol de elementos.



#### **4. ¿En qué beneficia usar Bootstrap para sitios y aplicaciones web en JS?**

Bootstrap es un framework de código abierto que facilita el desarrollo de sitios y aplicaciones web modernas. Su principal beneficio es que permite crear diseños rápidos, consistentes y adaptables (responsive) sin tener que escribir mucho código CSS desde cero.

Cuando se usa junto con JavaScript, Bootstrap ofrece componentes interactivos listos para usar, como modales, menús desplegables, carruseles, alertas, pestañas, entre otros. Estos componentes ya vienen programados con JavaScript y se activan fácilmente mediante clases HTML, lo que ahorra tiempo de desarrollo y evita errores comunes.

*Beneficios principales:*

- Diseño responsive: Las páginas se ven bien en computadoras, tabletas y celulares.
- Ahorro de tiempo: Tiene estilos y funciones ya predefinidos.
- Componentes interactivos: Facilita el uso de JavaScript sin escribirlo desde cero.
- Compatibilidad: Funciona bien con todos los navegadores modernos.
- Comunidad activa: Hay muchos recursos, plantillas y documentación disponibles.



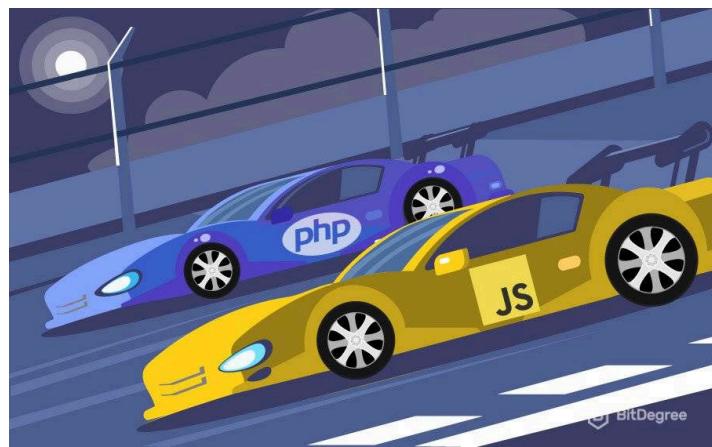
#### **5. ¿Qué semejanza y diferencia tienen los lenguajes web PHP y Java Script?**

PHP y JavaScript son dos lenguajes de programación muy utilizados en el desarrollo web. Ambos son lenguajes de scripting, lo que significa que no necesitan ser compilados antes de ejecutarse. Además, los dos pueden integrarse fácilmente con HTML y permiten crear páginas dinámicas, también comparten ciertas similitudes en su sintaxis, como el uso de estructuras de control (if, while, for, etc.) y el uso de punto y coma para terminar instrucciones. Gracias a su amplia comunidad, existe una gran cantidad de recursos, librerías y soporte para ambos lenguajes.

Sin embargo, también presentan diferencias importantes, la principal es dónde se ejecutan: PHP se ejecuta en el servidor, mientras que JavaScript se ejecuta en el navegador del cliente. Esto significa que PHP es ideal para tareas como conectarse a bases de datos, generar páginas dinámicas en el servidor y gestionar formularios antes de enviarlos al cliente. Por otro lado, JavaScript permite interacción en tiempo real con el usuario sin recargar la página, como mostrar mensajes emergentes, validar formularios en el momento o actualizar el contenido visual dinámicamente.

Además, JavaScript tiene acceso completo al DOM (Document Object Model), lo que le permite modificar elementos de la página web mientras el usuario la está viendo, mientras que PHP no puede hacer eso, ya que su ejecución ocurre antes de que la página llegue al navegador. También se diferencia en su sensibilidad a las mayúsculas y minúsculas: en PHP las variables distinguen entre mayúsculas y minúsculas, pero las funciones no siempre lo hacen; en cambio, en JavaScript todo es sensible a mayúsculas.

En resumen, PHP es ideal para tareas del lado del servidor, como el procesamiento de datos y la conexión a bases de datos, mientras que JavaScript es esencial para crear una experiencia interactiva para el usuario desde el navegador.



## 6. ¿Cite 3 formas en que se puede agregar código JS en una página web?

Hay tres formas principales de agregar código JavaScript a una página web, y cada una se usa dependiendo del tipo de proyecto y de cómo se desea organizar el código.

La primera manera consiste en escribir el código JavaScript directamente dentro del archivo HTML, usando la etiqueta <script>. Este código se suele colocar dentro del cuerpo (<body>) del documento, y se ejecuta a medida que el navegador va leyendo el contenido de la página. Este método es útil para scripts simples y directos, como mostrar un mensaje emergente cuando se carga la página.

La segunda forma es insertar el código JavaScript dentro de la etiqueta <head> del documento HTML, esta técnica permite que el código se cargue antes de que el cuerpo de la página esté disponible. Sin embargo, si no se usa correctamente, puede hacer que la carga del contenido visual se retrase, ya que el navegador intenta ejecutar el script antes de mostrar la página. Para evitar eso, se puede agregar el atributo defer para que el código se ejecute solo después de que todo el HTML haya sido procesado.

La tercera forma, y una de las más recomendadas en proyectos más grandes o profesionales, es usar un archivo JavaScript externo. Esto significa guardar el código en un archivo aparte con extensión .js y luego enlazarlo al documento HTML usando la etiqueta <script> con el atributo src. Esta forma permite tener el código más organizado, facilita el

mantenimiento y mejora la velocidad de carga porque el archivo puede ser reutilizado y cargado de forma más eficiente por el navegador.



## 7. ¿Cuál es la función principal de la consola en JS?

La consola en JavaScript es una herramienta fundamental que permite a los desarrolladores ver mensajes, errores y resultados directamente desde el navegador mientras desarrollan una página web. Su función principal es servir como un medio para depurar y probar el código, es decir, identificar errores y verificar que el comportamiento del programa sea el esperado.

Una de las funciones más comunes es `console.log()`, que permite imprimir información como variables, textos o resultados de cálculos directamente en la consola del navegador. Esto es muy útil para seguir el flujo del programa y entender qué está ocurriendo en cada parte del código sin tener que mostrarlo al usuario en la pantalla.

Además de `console.log()`, la consola ofrece otras herramientas como `console.error()` para mostrar errores, `console.warn()` para advertencias y `console.table()` para ver datos en forma de tabla. Estas funciones ayudan a organizar y visualizar la información de forma clara durante el desarrollo.

En resumen, la consola es una herramienta de diagnóstico y prueba, que permite escribir, observar y corregir el comportamiento del código JavaScript sin afectar la vista final del usuario en la página web.

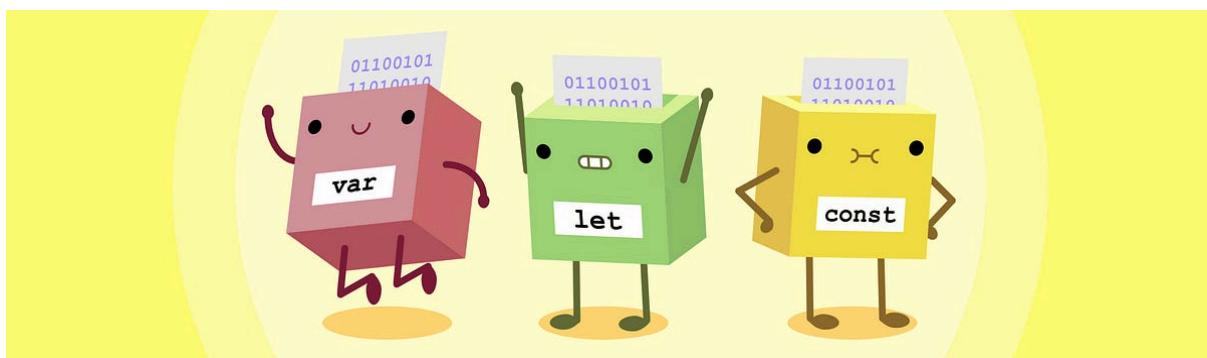


**8. ¿Cuál es la diferencia que existe en las declaraciones var, let y const en JS?**  
¿Brinde un ejemplo de cada una?

En JavaScript, las palabras clave var, let y const se utilizan para declarar variables, pero cada una se comporta de manera diferente. La palabra var fue la forma original de declarar variables y tiene un alcance de función, lo que significa que solo está disponible dentro de la función donde fue declarada. Por ejemplo, si se escribe var nombre = "Carlos"; se crea una variable que luego puede cambiar de valor (nombre = "Luis";) o incluso ser redeclarada con var nombre = "Ana"; sin generar error.

Por otro lado, let fue introducido en versiones más modernas de JavaScript y tiene alcance de bloque, es decir, solo existe dentro del bloque (como un if, for o una función) donde fue declarada. Si se escribe let edad = 25; se puede cambiar su valor con edad = 30;, pero no se puede volver a declarar esa misma variable dentro del mismo bloque, ya que eso generaría un error.

Finalmente, const también tiene alcance de bloque, pero se usa para declarar constantes, es decir, valores que no deben cambiar. Por ejemplo, si se escribe const pais = "Costa Rica"; esa variable no puede cambiar su valor después, y cualquier intento de hacerlo causará un error, const se usa con objetos o arreglos, su contenido interno sí puede modificarse, aunque la referencia principal no.

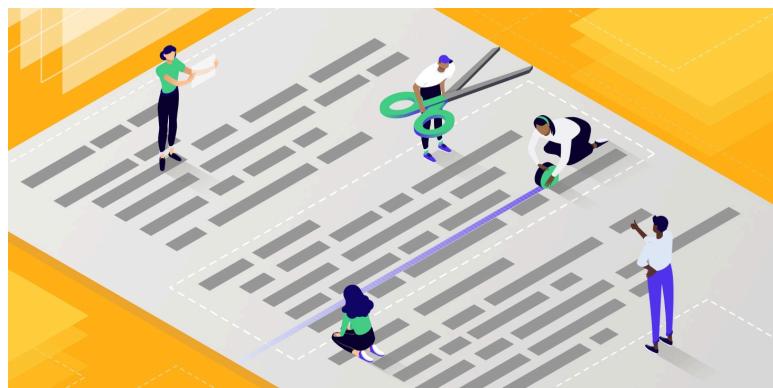


**9. ¿Cuál es la función de minificar archivos Java Script?**

La función principal de minificar archivos JavaScript es optimizar el código para ahorrar espacio, reducir los tiempos de carga de la página web y disminuir el uso del ancho de banda. La minificación consiste en eliminar todos los elementos que no son necesarios para que el código funcione correctamente, como los espacios en blanco, saltos de línea y comentarios, sin alterar la funcionalidad del programa.

Este proceso hace que el archivo JavaScript sea mucho más compacto y rápido de descargar, lo que mejora la velocidad de carga y la experiencia del usuario. Sin embargo, al eliminar estos elementos, el código minificado se vuelve casi ilegible para los desarrolladores. Por eso, generalmente se mantiene una versión original para trabajar y solo se minifica la versión final que se publica.

Para minificar un archivo JavaScript, es necesario analizar y comprimir el código cuidadosamente. A veces se hacen ajustes adicionales, como reescribir variables locales o simplificar estructuras, para que el código sea más eficiente y funcione igual pero con menor tamaño.



**10. ¿Qué es ECMAScript6? Explique claramente.**

ECMAScript 6, también conocido como ES6 o ECMAScript 2015, es una versión moderna y significativa del estándar que define cómo debe funcionar el lenguaje JavaScript. Este estándar fue publicado en 2015 y representa una gran actualización con muchas nuevas características que facilitan la escritura de código más claro, eficiente y estructurado.

Entre las novedades más importantes de ECMAScript 6 se encuentran nuevas formas para declarar variables (`let` y `const`), la introducción de funciones flecha, clases, módulos, promesas, y varias mejoras en la manipulación de objetos y arreglos. Estas características permiten a los desarrolladores crear aplicaciones más potentes y mantenibles con menos código.

ES6 marcó un cambio importante en el desarrollo con JavaScript, convirtiéndose en la base para la mayoría del desarrollo moderno. Desde su lanzamiento, los navegadores han ido implementando estas funcionalidades para que los programadores puedan aprovecharlas en sus proyectos.



## **Conclusión**

JavaScript ha transformado profundamente la forma en que interactuamos con las páginas web, permitiendo que dejen de ser estáticas para volverse dinámicas y participativas. Al conocer su historia, evolución y aplicaciones, se comprende que no es solo un lenguaje más, sino una herramienta clave en la creación de experiencias digitales modernas. Reflexionar sobre su importancia nos invita a valorar el papel que juega en casi todo lo que vemos y usamos en internet. Aprender JavaScript no es solo adquirir una habilidad técnica, sino también abrir la puerta a infinitas posibilidades de innovación y creatividad.

## Referencias Web

Mozilla Contributors. (s.f.). *Guía de JavaScript – Introducción*. MDN Web Docs. Recuperado el 26 de mayo de 2025, de  
<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introduction>

Wikipedia contributors. (s.f.). *JavaScript*. Wikipedia. Recuperado el 26 de mayo de 2025, de  
<https://es.wikipedia.org/wiki/JavaScript>

Mozilla Contributors. (s.f.). *JavaScript*. MDN Web Docs. Recuperado el 26 de mayo de 2025, de <https://developer.mozilla.org/es/docs/Web/JavaScript>

Mozilla Contributors. (s.f.). *Agregar interactividad a tu primera página web*. MDN Web Docs. Recuperado el 26 de mayo de 2025, de  
[https://developer.mozilla.org/es/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Adding\\_interactivity](https://developer.mozilla.org/es/docs/Learn_web_development/Getting_started/Your_first_website/Adding_interactivity)

IONOS. (s.f.). *PHP vs JavaScript: ¿Cuál es la diferencia?* IONOS Digital Guide. Recuperado el 26 de mayo de 2025, de  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/php-vs-javascript/>

Bootstrap Contributors. (s.f.). *Introducción a Bootstrap*. Bootstrap. Recuperado el 26 de mayo de 2025, de <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Mozilla Contributors. (s.f.). *API de la consola*. MDN Web Docs. Recuperado el 26 de mayo de 2025, de <https://developer.mozilla.org/es/docs/Web/API/console>

Mozilla Contributors. (s.f.). *Declaraciones de variables (var, let y const)*. MDN Web Docs. Recuperado el 26 de mayo de 2025, de  
[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar\\_and\\_types#declaraciones-de-variables](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar_and_types#declaraciones-de-variables)

Kinsta. (s.f.). *¿Qué es minificar JavaScript?*. Kinsta Blog. Recuperado el 26 de mayo de 2025, de <https://kinsta.com/es/blog/minificar-javascript/>

JSCode. (s.f.). *¿Qué es ECMAScript 6 y qué novedades tiene?* Recuperado el 26 de mayo de 2025, de [https://jscode.es/que\\_es\\_ecmascript\\_6](https://jscode.es/que_es_ecmascript_6)