
Lenguajes de programación



Ricardo Sánchez

Índex

1. Programas informáticos.....	3
2. Lenguajes de Programación.....	5
Tipos de lenguajes de programación.....	5
3. JavaScript.....	7
Características.....	7

1. Programas informáticos

Un **programa informático** es un conjunto de instrucciones que una vez ejecutadas realizarán una o varias tareas en una computadora. Sin programas, estas máquinas no pueden funcionar. Al conjunto general de programas, se le denomina **software**, que más genéricamente se refiere al equipamiento lógico o soporte lógico de una computadora digital.

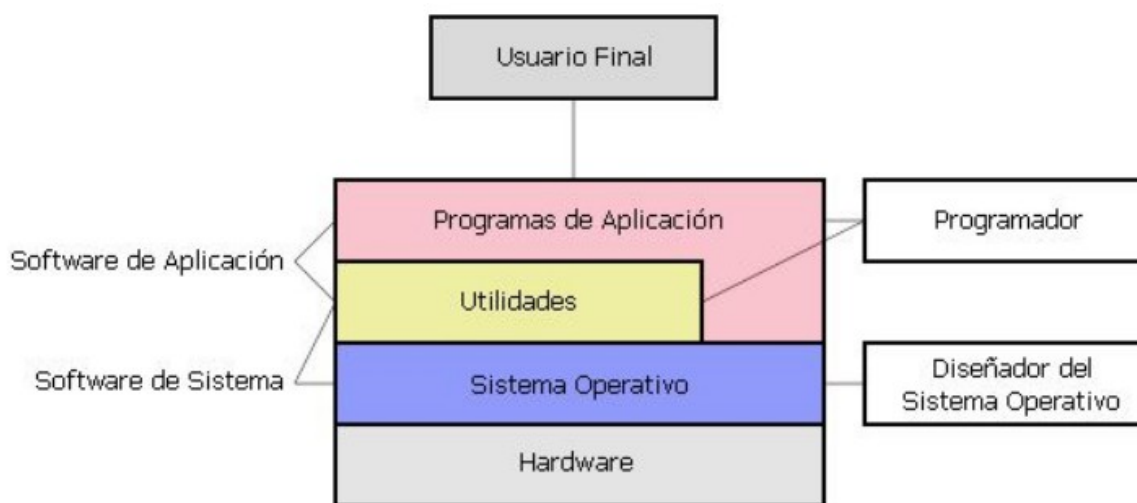
La función del software en una computadora es similar a la del pensamiento en un ser humano, por eso al software se lo define también como la parte intangible de la informática.

El software es el nexo entre el **hardware** (computadora) y el **hombre** (usuario). La computadora, por sí sola, no puede comunicarse con el usuario y viceversa, ya que los separa la barrera del lenguaje. El software trata de eliminar esa barrera, estableciendo procedimientos de comunicación entre el hombre y la máquina; es decir, el software obra como un intermediario entre el hardware y el hombre.

Existen diferentes tipos de software: software de sistema y software de aplicación:

Una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Esto lo diferencia principalmente de otros tipos de programas (sistema) que realizan tareas más avanzadas y no pertinentes al usuario común, como los sistemas operativos (que hacen funcionar al ordenador, como Windows, Mac o Linux), las utilidades (que realizan tareas de mantenimiento o de uso general), y los lenguajes de programación (con el cual se crean los programas informáticos).

Las aplicaciones suelen diseñarse para la automatización de ciertas tareas complicadas o tediosas como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén. Algunos ejemplos de programas de aplicación son los procesadores de textos, hojas de cálculo, y base de datos.



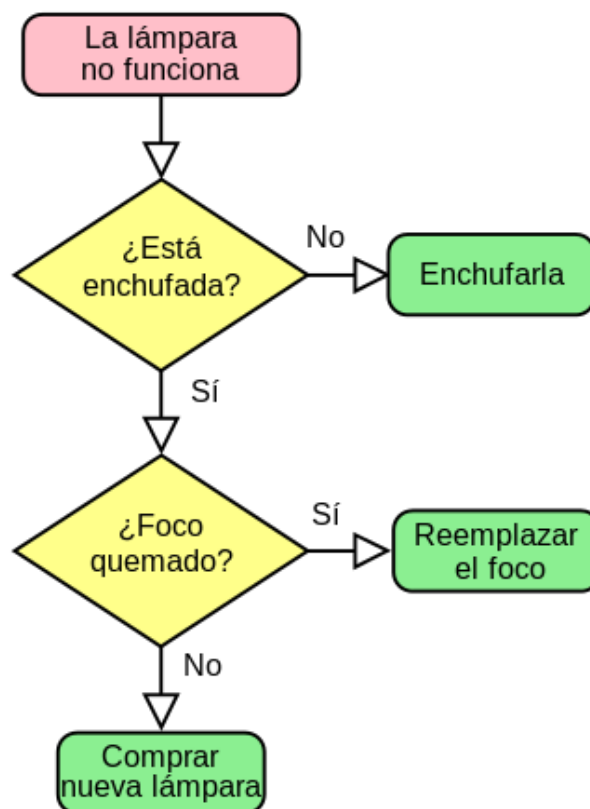
Un **algoritmo** es la forma de resolver un problema, los pasos a seguir para resolver un problema estándar:

Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas.

El algoritmo de “ir a la compra” sería:

- 1) Hago la lista de la compra
- 2) Voy a comprar
- 3) Guardo las cosas.

Otro ejemplo de un algoritmo expresado con un diagrama de flujo:



En programación, los algoritmos se definen antes de pasar a su codificación en un lenguaje de programación determinado. Un mismo algoritmo puede ser codificado de diferentes formas según el lenguaje de programación utilizado.



2. Lenguajes de Programación

Un “**lenguaje de programación**” es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. Hay muchísimos, de toda clase de tipos y características, inventados para facilitar el abordaje de distintos problemas, el mantenimiento del software, su reutilización, mejorar la productividad, etc.

Tipos de lenguajes de programación

Los lenguajes de programación se pueden clasificar según varios criterios. Hay que tener en cuenta también, que en la práctica, la mayoría de lenguajes no pueden ser puramente clasificados en una categoría, pues surgen incorporando ideas de otros lenguajes y de otras filosofías de programación, pero no importa al establecer las clasificaciones, pues el auténtico objetivo de las mismas es mostrar los rangos, las posibilidades y tipos de lenguajes que hay.

1. Nivel de abstracción.

Según el nivel de abstracción, es decir, según el grado de cercanía a la máquina:

- **Lenguajes de bajo nivel:** La programación se realiza teniendo muy en cuenta las características del procesador. Ejemplo: Lenguajes ensamblador.
- **Lenguajes de alto nivel:** Más parecidos al lenguaje humano. Manejan conceptos, tipos de datos, etc., de una manera cercana al pensamiento humano ignorando (abstrayéndose) del funcionamiento de la máquina. Ejemplos: Java, Ruby, Javascript.

2. Propósito.

Según el propósito, es decir, el tipo de problemas a tratar con ellos:

- **Lenguajes de propósito general:** Aptos para todo tipo de tareas: Ejemplo: C, Java.
- **Lenguajes de propósito específico:** Hechos para un objetivo muy concreto. Ejemplo: Javascript y PHP para aplicaciones web.
- **Lenguajes de programación de sistemas:** Diseñados para realizar sistemas operativos o drivers. Ejemplo: C, C++.

3. Manera de ejecutarse.

Según la manera de ejecutarse:

Una vez escrito el código (**código fuente**), éste se debe traducir a lenguaje binario (**código objeto**) que entienda la máquina. Según la forma en que se realice este proceso, los lenguajes pueden ser:

- **Lenguajes compilados:** La traducción se realiza una vez finalizado el programa y produce un código binario ejecutable. Suelen ser más rápidos en ejecución. Ejemplos C, C++.

- **Lenguajes interpretados:** También denominados lenguajes de script. El código se guarda en un script (texto plano) que es compilado cada vez que se ejecuta. Eso hace la velocidad de ejecución sea mucho más lenta que en los lenguajes compilados, pero es adecuado para pequeños programas y aplicaciones. También se pueden ejecutar en múltiples plataformas. Ejemplos: Javascript, Python.
- **Mixtos:** pensados para ser ejecutados en múltiples plataformas. El código fuente se compila en un código intermedio capaz de ser ejecutado por una máquina virtual en diferentes plataformas hardware y sistemas operativos. Ejemplos: Java y plataforma .NET.

4. Orientación a objetos:

La **programación orientada a objetos (POO, u OOP** según sus siglas en inglés) es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

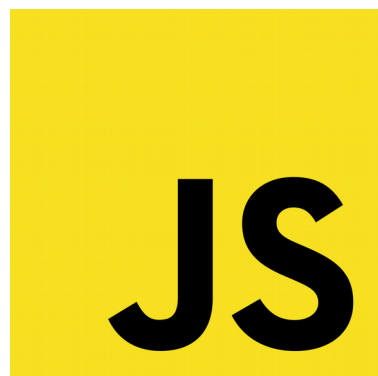
3. JavaScript

JavaScript (abreviado comúnmente **JS**) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como **orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico**.

Se utiliza principalmente en su forma del **lado del cliente** (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del **Document Object Model (DOM)**.



Características

Orientado a objetos

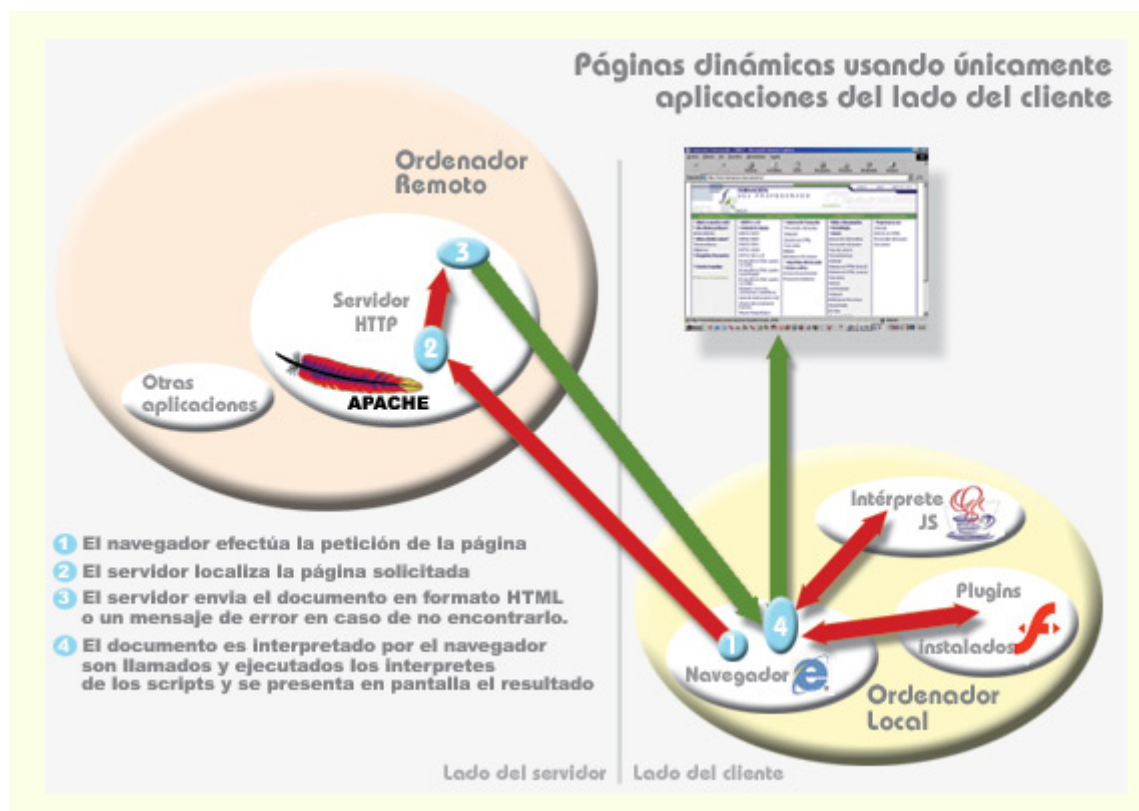
Aunque sólo permite la definición de clases a partir del estándar ECMA 2015, Javascript tiene soporte de objetos nativo desde el principio, pudiendo manejar el árbol DOM de las páginas web.

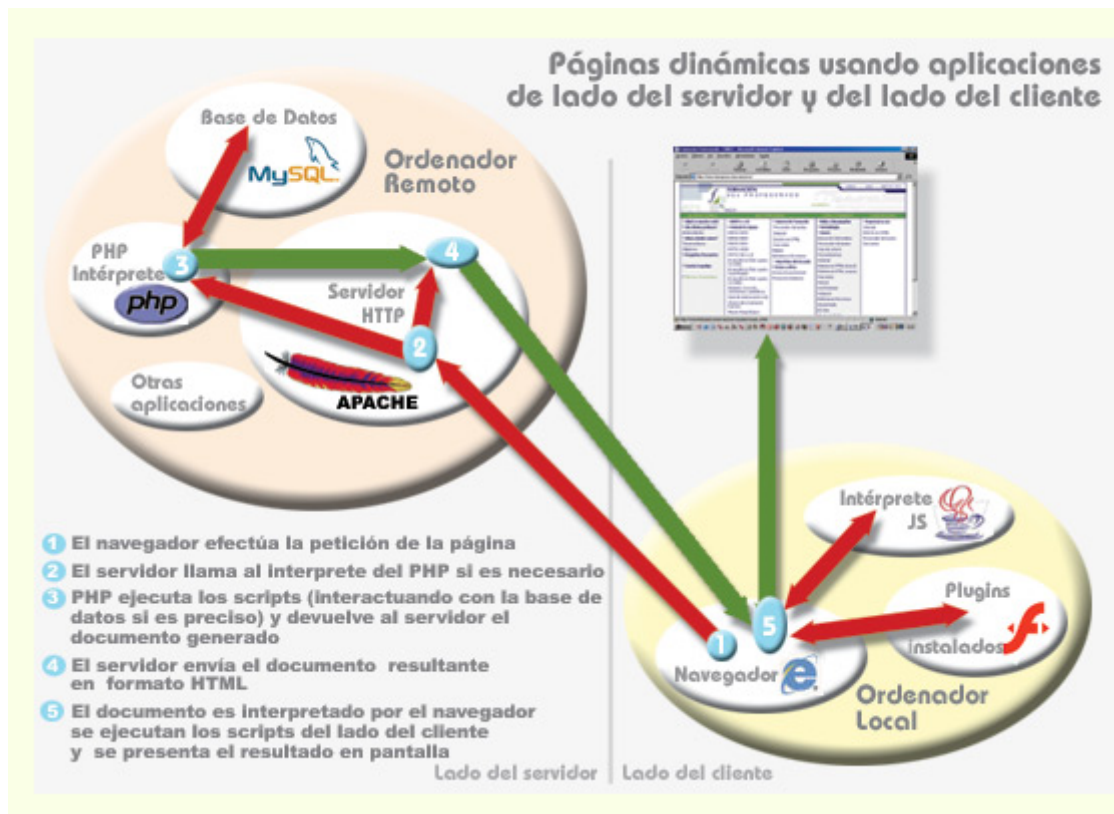
En realidad, JavaScript es un lenguaje **basado en prototipos**, ya que debe simular muchas de las características de orientación a objetos que se dan en la mayoría de los lenguajes. A cambio proporciona diversos paradigmas de programación (funcional, imperativo, dinámico y orientado a objetos) lo que lo convierte en un lenguaje tremendamente versátil.

Lenguaje del lado del cliente

Javascript es un lenguaje utilizado en tecnologías web que se ejecuta en el cliente web (navegadores) en el sistema del usuario. Normalmente se usa para tratar datos de formularios antes de ser enviados al servidor, modificar las páginas web, etc. La parte del procesamiento de la información en el servidor se realiza por otro lenguaje, normalmente **PHP**, **ASP**, etc.

Pero, también hay implementaciones de Javascript en el lado del servidor. A partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor. **Node.js** es uno de los notables ejemplos de JavaScript en el lado del servidor, siendo usado en proyectos importantes.





Débilmente tipado

En JavaScript una misma variable puede contener primero un texto, luego un número, luego una fecha o un objeto, etc... Esto es un arma de doble filo pues permite una gran flexibilidad a cambio de muchos posibles errores si no tenemos cuidado.

Lenguaje interpretado

los programas JavaScript son archivos de texto que pueden leer tanto los ordenadores como las personas. Estos textos, también llamados **scripts**, se compilan cada vez que se ejecutan por el intérprete, que es normalmente el **navegador web**.

No es lo mismo que Java

Dado que comparten una palabra al principio de su nombre, muchas personas piensan que Java y JavaScript están relacionados, o que JavaScript es un subconjunto de Java y cosas parecidas... Nada más lejos de la realidad:

Java nació en el año 1991 en la empresa **Sun Microsystems**, de la mano de **James Gosling**. Su objetivo inicial era crear aplicaciones para receptores de televisión y dispositivos embebidos, aunque al final haya terminado funcionando en casi todas partes.

JavaScript es un lenguaje de programación posterior, creado a mediados de los años '90 por **Brendan Eich**, de la empresa **Netscape**. Su objetivo era **dotar de interactividad a las páginas web** visualizadas con su navegador de Internet, **Netscape Navigator**, el líder del mercado por aquella época. Su nombre en código durante el desarrollo del lenguaje era *Mocha*, y pensaban lanzarlo finalmente como *LiveScript*, aunque al final se acabó denominando JavaScript por la gran influencia en su sintaxis por parte del lenguaje Java, aunque las similitudes son únicamente estéticas.

Referencias

- <https://es.wikipedia.org/wiki/Algoritmo>
- <https://es.wikipedia.org/wiki/JavaScript>
- <https://www.campusmvp.es/recursos/post/Java-y-JavaScript-son-lo-mismo.aspx>
- <https://www.javascript.com/>