Node.js

1. **Descripción general**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.4​ Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.5​6​

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y Java EE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS.7​ Node.js incluye un entorno REPL para depuración interactiva.

1. **Características**
   1. **Concurrencia**

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto.8​ Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como cluster9​ para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

* 1. **V8**

V8 es el entorno de ejecución para JavaScript creado para Google Chrome. Es software libre desde 2008, está escrito en C++ y compila el código fuente JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

Node.js contiene libuv para manejar eventos asíncronos. Libuv es una capa de abstracción de funcionalidades de redes y sistemas de archivo en sistemas Windows y sistemas basados en POSIX como Linux, Mac OS X y Unix.

El cuerpo de operaciones de base de Node.js está escrito en JavaScript con métodos de soporte escritos en C++.

* 1. **Módulos**

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y otros módulos fundamentales, como por ejemplo Path, FileSystem, Buffer, Timers y el de propósito más general Stream. Es posible utilizar módulos desarrollados por terceros, ya sea como archivos ".node" precompilados, o como archivos en javascript plano. Los módulos Javascript se implementan siguiendo la especificación CommonJS para módulos,10​ utilizando una variable de exportación para dar a estos scripts acceso a funciones y variables implementadas por los módulos.

Los módulos de terceros pueden extender node.js o añadir un nivel de abstracción, implementando varias utilidades middleware para utilizar en aplicaciones web, como por ejemplo los frameworks connect y express. Pese a que los módulos pueden instalarse como archivos simples, normalmente se instalan utilizando el Node Package Manager (npm) que nos facilitará la compilación, instalación y actualización de módulos así como la gestión de las dependencias. Además, los módulos que no se instalen en el directorio por defecto de módulos de Node necesitarán la utilización de una ruta relativa para poder encontrarlos. El wiki Node.js proporciona una lista de varios de los módulos de terceros disponibles.

* 1. **Desarrollo homogéneo entre cliente y servidor**

Node.js puede ser combinado con una base de datos documental (por ejemplo, MongoDB o CouchDB) y bases de datos relacionales como MySQL, PostgreSQL, entre otros, lo que permite desarrollar en un entorno de desarrollo JavaScript unificado. Con la adaptación de los patrones para desarrollo del lado del servidor tales como MVC y sus variantes MVP, MVVM, etc. Node.js facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor.

* 1. **Bucle de eventos**

Node.js se registra con el sistema operativo y cada vez que un cliente establece una conexión se ejecuta un callback. Dentro del entorno de ejecución de Node.js, cada conexión recibe una pequeña asignación de espacio de memoria dinámico, sin tener que crear un hilo de ejecución. A diferencia de otros servidores dirigidos por eventos, el bucle de gestión de eventos de Node.js no es llamado explícitamente sino que se activa al final de cada ejecución de una función callback. El bucle de gestión de eventos se termina cuando ya no quedan eventos por atender.

1. **Versiones**

Las nuevas versiones principales de Node.js se sacan de la rama máster de GitHub cada seis meses. Las versiones pares se sacan en abril, y las impares en octubre. Cuando se libera una versión impar, la versión par anterior pasa a soporte a largo plazo (Long Term Support, LTS), que da a la versión un soporte activo de 18 meses desde la fecha de inicio de la LTS. Después de estos 18 meses, la versión recibe otros 12 meses de soporte de mantenimiento. Una versión activa recibirá los cambios compatibles unas pocas semanas después de que aterricen en la versión estable actual. Una versión de mantenimiendo recibirá solamente actualizaciones críticas y de documentación.

La estrategia y política de las versiones LTS se gestiona por el LTS Working Group en colaboración con el Technical Steering Committee de Node.js Foundation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Versión** | **Nombre** | **Fecha de liberación** | **Estado LTS** | **Inicio LTS** | **Inicio de mantenimiento** | **Fin de mantenimiento** |
| v0.10.x |  | 2013-03-11 | Sin soporte | - | 2015-10-01 | 2016-10-31 |
| v0.12.x |  | 2015-02-06 | Sin soporte | - | 2016-04-01 | 2016-12-31 |
| 4.x | Argon[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 2015-09-08 | Sin soporte | 2015-10-01 | 2017-04-01 | 2018-04-30 |
| 5.x |  | 2015-10-29 | No **No LTS** | N/D | | 2016-06-30 |
| 6.x | Boron[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 2016-04-26 | Sin soporte | 2016-10-18 | 2018-04-30 | Abril de 2019 |
| 7.x |  | 2016-10-25 | No **No LTS** | N/D | | 2017-06-30 |
| 8.x | Carbon[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 2017-05-30 | Sin soporte | 2017-10-31 | 2019-01-01[14](https://es.wikipedia.org/wiki/Node.js#cite_note-14)​ | Diciembre de 2019 |
| 9.x |  | 2017-10-01 | No **No LTS** | N/D | | 2018-06-30 |
| 10.x | Dubnium[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 2018-04-24 | Mantenimiento | 30 de octubre de 2018 | 19 de mayo de 2019 | 30 de abril de 2021 |
| 11.x |  | 2018-10-23 | No **No LTS** | N/D | | 1 de junio de 2019 |
| 12.x | Erbium[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 2019-04-23 | Mantenimiento | 21 de octubre de 2019 | 20 de octubre de 2020 | 30 de abril de 2022 |
| 13.x | - | 22 de octubre de 2019 | No **No LTS** | - | 1 de abril de 2020 | 1 de junio de 2020 |
| **14.x** | Fermium[13](https://es.wikipedia.org/wiki/Node.js#cite_note-release-13)​ | 21 de abril de 2020 | **Activo** | 20 de octubre de 2020 | 19 de octubre de 2021 | 30 de abril de 2023 |
| **15.x** | - | 22 de octubre de 2020 | No **No LTS** | - | 1 de abril de 2021 | 1 de junio de 2021 |
| 16.x | - | 21 de abril de 2021 | Prevista | 26 de octubre de 2021 | 18 de octubre de 2022 | 30 de abril de 2024 |