

# NODE JS



## Instalación y uso

Juan Mateo Fischer, Gonzalo Larrosa, Gabriel Martinez

# ¿Qué es NodeJS y un poco de historia?

---

- Es un entorno de ejecución para JavaScript
- Multiplataforma, de código abierto, asíncrono y orientado a eventos
- Creado en el año 2009 por Ryan Dahl para cambiar el concepto de Javascript
- Los programas escritos en JS son compilados, optimizados e interpretados por la máquina virtual V8 de chrome



# ¿Para qué sirve?

---

- Ejecutar JavaScript desde nuestro ordenador sin ningun browser
- Ideal para construir aplicaciones altamente escalables y de uso intensivo de operaciones de entrada y salida
- Crear sitios web dinámicos muy eficientes escritos en JavaScript
- Agregar, borrar y modificar datos en la base de datos
- Crear, abrir, leer, escribir, eliminar y cerrar archivos en el servidor
- Y más...

# ¿Qué ventajas tiene frente a lo tradicional?

---

- Lenguaje sencillo basado en JavaScript, así que puede ser manejado fácilmente por programadores de Java.
- La compilación de Node.js se realiza en tiempo de ejecución, Just In Time (JIT), esto trae consigo una mayor optimización a las funciones que más veces sean llamadas.
- Facilita la creación de aplicaciones de internet de gran competencia, ya que administrador - usuario incorporan estrategias de codificación similares.
- Podemos expandir nuestro código añadiendo módulos de forma fácil gracias al Node Package Manager (NPM).
- En startups o equipos pequeños podremos realizar front-end, back-end y hasta una aplicación móvil con un mismo lenguaje.
- Cuenta con una gran comunidad virtual, que ha crecido de manera exponencial en los últimos años.

# Tipos de módulos y algunos ejemplos

---

Tres tipo de módulos:

- **Core:** Módulos principales incluyen funcionalidades mínimas de Node.js
- **Local:** Módulos locales son módulos creados localmente en la aplicación Node.js.
- **Terceros:** Son módulos locales empaquetados y distribuidos a través de NPM, para que la comunidad de pueda usarlo

Módulos de Core:

- **http, url y querystring:** incluyen clases, métodos y eventos para crear el servidor http
- **path:** incluye métodos para manejar rutas de archivos
- **fs:** incluye clases, métodos y eventos para trabajar con archivos de E/S
- **util:** incluye funciones útiles para los programadores

# ¿Cómo se manejan los módulos?

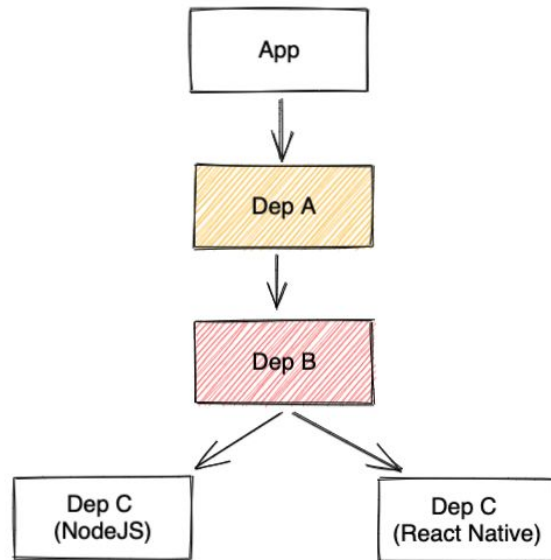
## NPM:

- Node Package Manager (NPM) es una herramienta de línea de comandos que instala, actualiza o desinstala paquetes de Node.js.
- También es un repositorio en línea para paquetes Node.js de código abierto.
- La comunidad de nodos de todo el mundo crea módulos útiles y los publica como paquetes en este repositorio.

## Paquete:

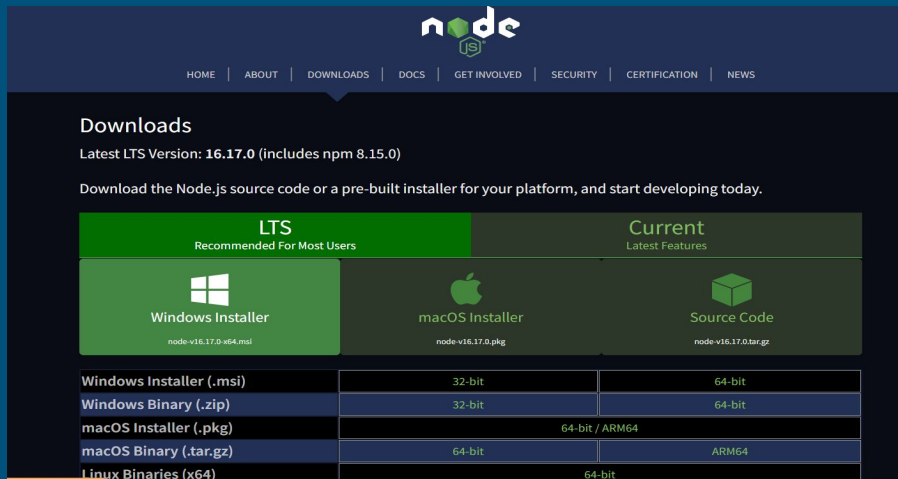
- Un paquete como vimos anteriormente es un módulo local empaquetado
- **Instalar:** `npm install <package name>`
- **Guardar:** `--save` para guardarlas en el `package.json`

Nos permite versionar las dependencias y manejarlas adecuadamente, esto guarda relación con lo visto en el curso sobre SCM.



# Paso 1: Instalación y ejecución básica

<https://nodejs.org/en/download/>



The screenshot shows the Node.js Downloads page. At the top is the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. The main heading is "Downloads", followed by "Latest LTS Version: 16.17.0 (includes npm 8.15.0)". Below this is a description: "Download the Node.js source code or a pre-built installer for your platform, and start developing today." The page is divided into two main sections: "LTS Recommended For Most Users" and "Current Latest Features". Under "LTS", there are three options: "Windows Installer" (node-v16.17.0-x64.msi), "macOS Installer" (node-v16.17.0.pkg), and "Source Code" (node-v16.17.0.tar.gz). Under "Current", there are three options: "Windows Installer (.msi)", "Windows Binary (.zip)", "macOS Installer (.pkg)", "macOS Binary (.tar.gz)", and "Linux Binaries (x64)". A table at the bottom lists the available binaries for each platform and architecture.

	32-bit	64-bit
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	

## Paso 2: Ejecución de un servidor HTTP

---

Cuando visualiza una página web en su navegador, está realizando una solicitud a otro equipo en Internet, que a continuación proporciona la página web como respuesta. Esa computadora con la que está hablando a través de Internet es un servidor web. Un servidor web recibe solicitudes HTTP de un cliente, como su navegador, y proporciona una respuesta HTTP, como una página HTML o JSON desde una API.

- Veremos un ejemplo de un Servidor HTTP local.



# Paso 3: Ejecución de un servidor EXPRESS

---

¿Por qué? ¿Y para qué?

- Express.js es un marco de aplicación web para Node.js. Proporciona varias funciones que hacen que el desarrollo de aplicaciones web sea rápido y fácil, lo que de otro modo llevaría más tiempo usando solo Node.js
- Express.js se basa en el módulo de middleware de Node.js denominado **connect**, que a su vez utiliza el módulo **http**

Veremos un ejemplo de hola mundo con un Post.



FIN