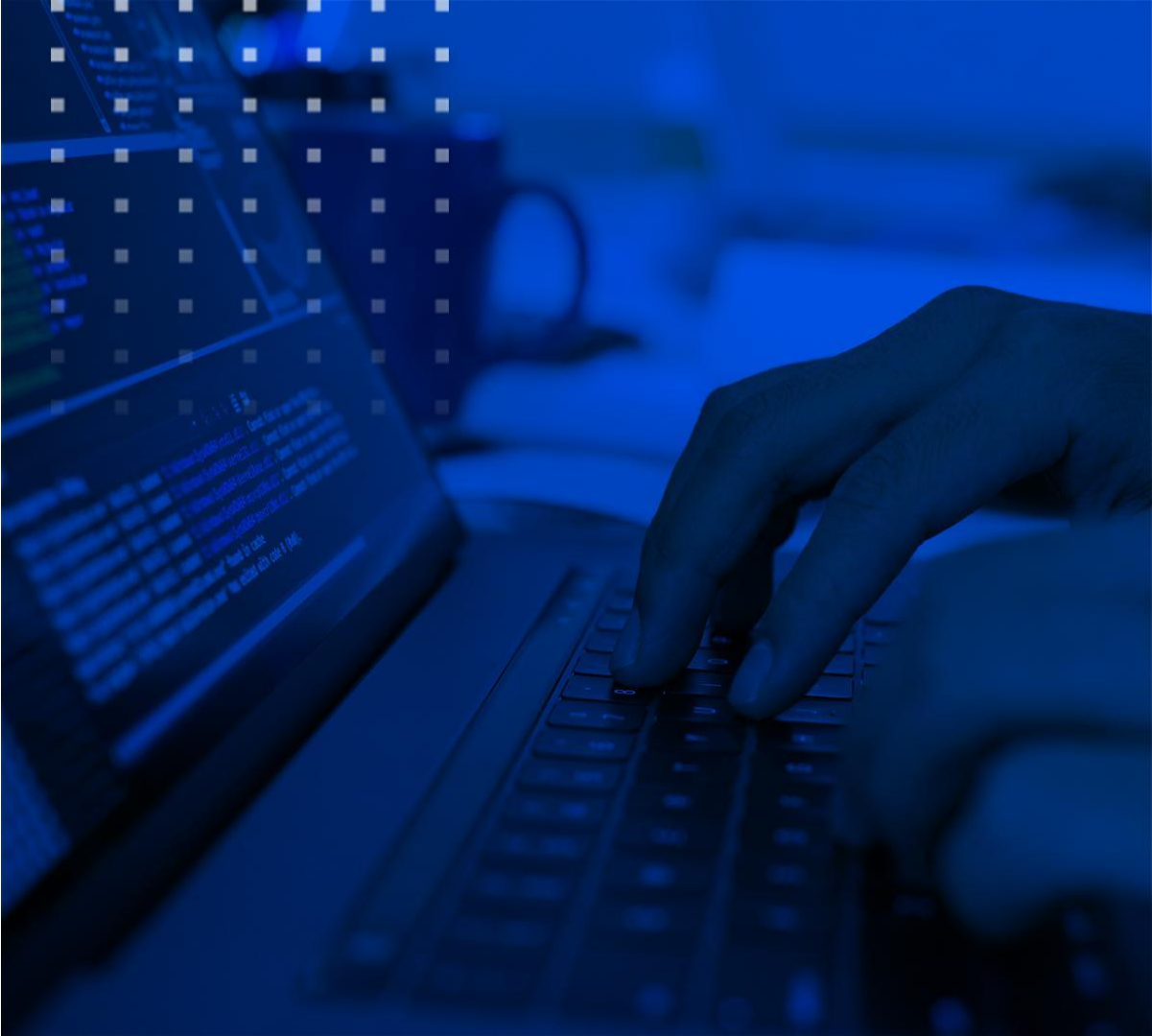


CURSO

FULL STACK DEVELOPER NIVEL INICIAL

UNIDAD 13
Node JS: Introducción a Node



Full Stack Developer Inicial

Node JS: Introducción a Node

Node.js

Node.js es un entorno de ejecución para construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S (Entrada/Salida) sin bloqueo y orientado a eventos, que lo hace liviano y eficiente.

El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo.

Node.js

- Fue concebido como un entorno de ejecución de JavaScript orientado a eventos **asíncronos** y está diseñado para construir aplicaciones en red escalables.
- Node permite además la creación de sitios y aplicaciones web enteramente con JavaScript, así como también la creación y utilización de un sin fin de herramientas.

Instalación

- Sitio oficial

[https://nodejs.org/es/.](https://nodejs.org/es/)

- Una vez terminada la instalación podemos verificar que todo haya salido correctamente abriendo la consola (CMD) y escribiendo **node -v**.



Uso básico de Node

En su forma más simple, podemos abrir la consola y escribir solamente **node**. Esto abrirá una instancia que nos permitirá ir ingresando expresiones de Javascript que serán evaluadas e impresas en el momento.

Uso básico de Node

Algunos ejemplos de cosas que podemos probar:

- Operaciones matemáticas
- Asignaciones de variables
- Definición de funciones
- Y cualquier otro código Javascript válido

Para salir de la instancia de node escribimos **.exit** (con punto antes de la e) o presionamos la combinación de teclas CTRL+C.

También podemos ejecutar scripts directamente con Node.

Creamos un archivo nuevo llamado index.js y escribimos el siguiente código:

```
let hoy = new Date()

console.log('Hoy es ' + hoy)

for(i = 0; i < 10; i++) {
  console.log(i)
}
```

Lo guardamos y lo ejecutamos escribiendo **node app.js**.

El resultado debería ser la fecha y hora actual y un conteo de 0 a 9 como respuesta en la consola.

Inicialización de un proyecto con Node.js

- Para comenzar un proyecto con Node.js desde cero es necesario correr el script de inicialización **npm init**. Este comando debe ser corrido desde la consola, en la carpeta donde deseemos inicializar el proyecto.

Inicialización de un proyecto con Node.js

- El script nos hará una serie de preguntas básicas sobre nuestro proyecto. Podemos contestar todas apretando la tecla ENTER o llenar los datos que creamos necesarios.
- Si no deseamos completar los datos en ese momento, podemos utilizar `npm init -y`

package.json

Al finalizar, encontraremos un archivo llamado **package.json**, donde se guardó toda la información que acabamos de cargar. Así mismo, en ese archivo, es donde el manejador de paquetes de Node, npm, va a ir guardando las dependencias o librerías de nuestro proyecto, así como también, varias de las herramientas que usaremos para desarrollo.

Instalación y uso de librerías con npm

La instalación de librerías se hace mediante el comando **npm** install. A este debemos indicarle qué librería deseamos que baje e instale y este lo hará de forma automática. Las librerías las podemos buscar en el sitio <https://www.npmjs.com/>. que es el repositorio central de npm.

The logo for npm (Node Package Manager) is displayed in a bold, black, lowercase font.

Ejemplo

A modo de ejemplo, vamos a instalar una librería llamada moment (<http://momentjs.com/>). Su página de npm es <https://www.npmjs.com/package/moment>, donde encontraremos información de la misma.

Ejecutamos el siguiente comando **npm install moment**.

Una vez terminada la instalación podemos abrir el archivo package.json para verificarlo.

package.json npm install.

En el apartado dependencias del archivo package.json se lista las librerías que requiere nuestro proyecto para funcionar. Las librerías que hayamos descargado de esta forma se graban automáticamente en la carpeta node_modules.

package.json npm install.

Por convención, si estamos usando GIT se evita versionar esta carpeta agregándola al archivo .gitignore.

En caso de que trabajemos en un proyecto que incluya un archivo package.json podemos descargar todas sus dependencias utilizando el comando **npm install**.

package-lock.json

En la versión 5, npm introdujo el package-lock.json.

El objetivo del package-lock.json es realizar un seguimiento de la versión exacta de cada paquete que se instala para que **un producto sea 100% reproducible de la misma manera.**

1. Instalar node js.
2. Ejecutar código js con node.

1. Crea un proyecto node con **npm init**.
2. Instalá una dependencia npm **npm install**.
3. Verifica tus archivos **package.json** y **package-lock.json**.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES