

Node js



**Inserta**



Cofinanciado por  
la Unión Europea



MINISTERIO  
DE TRABAJO  
Y ECONOMÍA SOCIAL



Fondos Europeos

**THE BRIDGE**  
DIGITAL TALENT **ACCELERATOR**

# Índice

**Que es Node js**

**Modulos**

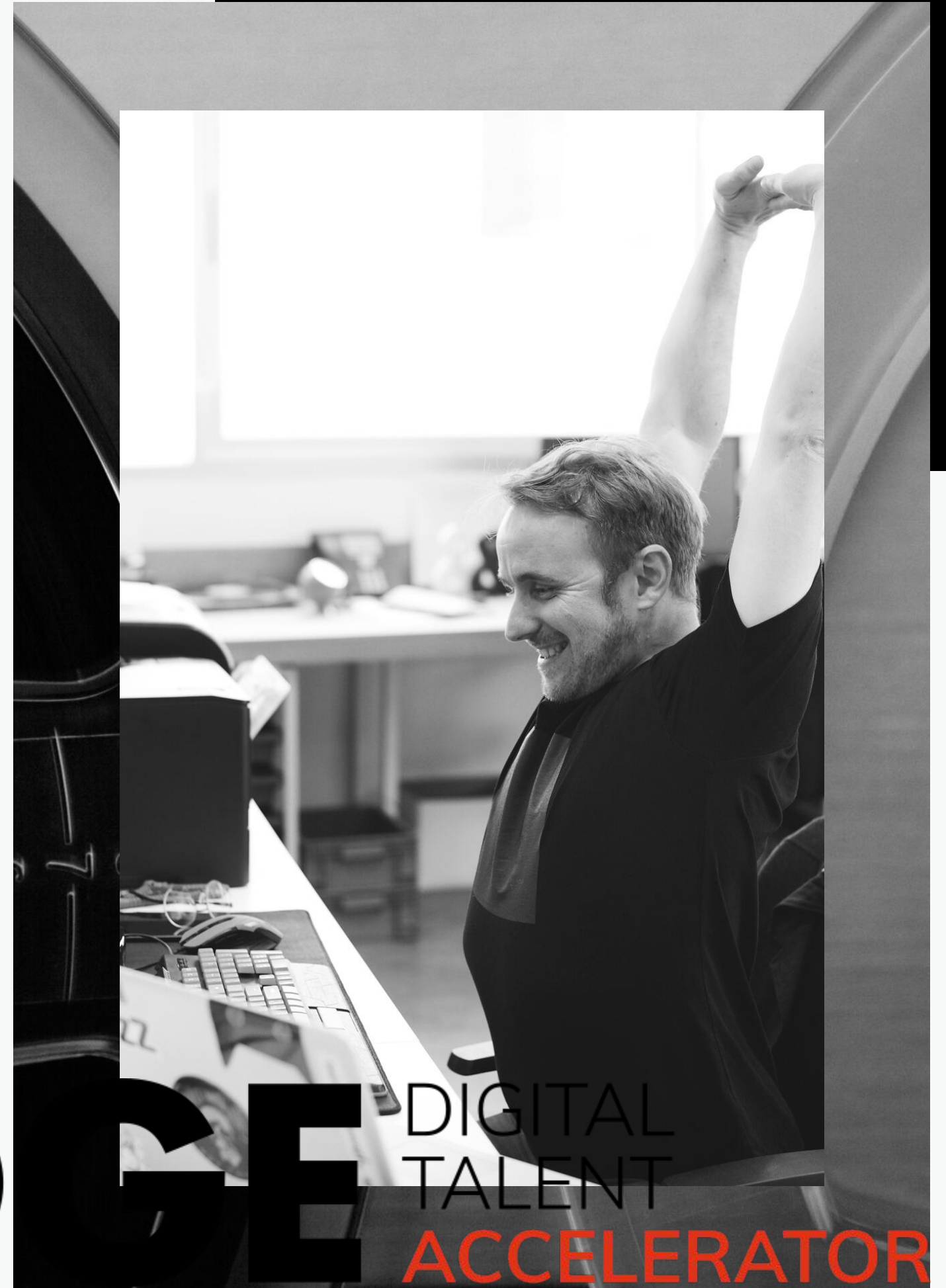
**Modulo HTTP**

**Modulo File System**

**NPM**

**THE  BRIDGE**

**DIGITAL  
TALENT  
ACCELERATOR**



# ¿Qué es Node.js?

- Node.js es un entorno de servidor de código abierto
- Node.js es gratis
- Node.js se ejecuta en varias plataformas (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js usa JavaScript en el servidor



# Instalación

Node js

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

# Empezando

- Descargado e instalado Node.js, intentamos mostrar texto en la consola.
- **Creamos un archivo Node.js llamado "index.js"** y agregamos el siguiente código:

```
const reyesCatolicos = ['Isabel de Castilla',  
'Cristobal Colon'];  
  
for (let i = 0; i < reyesCatolicos.length; i++) {  
  console.log('Rey Católico:', reyesCatolicos[i]);  
}
```

# Iniciando Node

- Para iniciar la interfaz , abrimos la consola y escribimos `node index.js` y presionamos enter.
- De esta forma tu ordenador ejecutara el codigo de Javascript directamente.

```
node index.js
```

# Módulos

**Node js**

[https://www.w3schools.com/nodejs/nodejs\\_modules.asp](https://www.w3schools.com/nodejs/nodejs_modules.asp)

# ¿Qué es un Módulo en Node.js?

- Los módulos son lo mismo que las bibliotecas de JavaScript.
- Un conjunto de funciones que puedes incluir en tu aplicación.
- **Node.js** tiene un módulo integrado llamado **HTTP**, que permite a Node.js transferir datos a través del Protocolo de transferencia de hipertexto (HTTP).
- Para incluir un módulo, se utiliza la función **require()** con el nombre del módulo:

```
let http = require("http");
```



# Crea tus propios módulos

- Puedes crear tus propios módulos e incluirlos fácilmente en tu código.
- En el siguiente ejemplo creamos un módulo que devuelve una función suma.
- Utilizamos la palabra clave **exports** para que las propiedades y los métodos estén disponibles fuera del archivo del módulo.

```
exports.sum = (a, b) => {  
  return a + b;  
};
```



## Importando nuestro módulo

- Podemos incluir y usar el módulo en cualquiera de nuestros archivos Node.js.
- Para probarlo escribimos el siguiente código en nuestro index.js

```
const math = require('./math');
```

```
console.log(math.sum(1,2));
```

# Creando un servidor con node

- Con el módulo HTTP podemos crear un servidor http que escucha en el puerto asignado y envía respuesta al cliente.
- Utilizamos **createServer()** para crear el servidor HTTP:

```
const http = require("http");

http.createServer((req, res)=>{
  res.writeHead(200, { "Content-Type": "text/html" });
  res.end("Aprendiendo en The Bridge!");
}).listen(8080);
```

# Modulo File System

- Este módulo nos permite trabajar con los archivos del sistema de nuestro ordenador.
- Para incluirlo simplemente usamos el método require():

```
const fs = require('fs');
```

## Leyendo archivos

- El método **fs.readFile()** se usa para leer archivos.
- Supongamos que tenemos el siguiente archivo HTML (ubicado en la misma carpeta que index.js):

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>
    <h1>Test 1</h1>
</body>
</html>
```

## Leyendo archivos

- Implementamos el siguiente código en nuestro archivo Node.js que lea el archivo HTML y devuelva el contenido:

```
const http = require("http");
const fs = require('fs');

http.createServer((req, res) => {
  fs.readFile('test.html', (err, data) => {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```



# NPM

NPM es un administrador de paquetes para paquetes de Node.js.

[www.npmjs.com](http://www.npmjs.com) alberga miles de paquetes gratuitos para descargar y usar.

El programa NPM se instala en su computadora cuando instala Node.js



# package.json

La manera de trabajar es utilizar el package.json que contiene toda la información de nuestro proyecto

Además es donde se registran las dependencias que vamos instalando.

Para crearlo ejecutamos el comando:

```
$ npm init
```



# Instalación de paquetes local

Un paquete instalado con el siguiente comando se descarga y se guarda en la carpeta **node\_modules/package-name**:

**\$ npm install package-name**

Además de registra en el package.json de dependencies.

Para instalar un paquete sin registrarlo, usa la marca **--no-save**.

# Otras instrucciones de NPM

Para desinstalar paquetes instalados usamos:

**\$ npm uninstall nombre del paquete**

Para actualizar:

// Para actualizar todos los paquetes instalados.

**\$ npm update**

// Para actualizar un paquete determinado.

**\$ npm update nombre del paquete**

## Scripts

- Otras de las posibilidades que nos brinda NPM es la definición de scripts en el package.json que podemos lanzar desde la consola.

```
"scripts": {  
  "start": "node index.js"  
},
```

## Descargando un paquete

- Abrimos la consola y con NPM descargamos el paquete que queremos. En este caso un paquete llamado "upper-case":

```
$npm install upper-case
```

## Usando el paquete

- Incluimos el paquete "upper-case" de la misma manera que incluimos cualquier otro módulo.
- Creamos un archivo Node.js que convertirá la salida "¡Hola The Bridge!" en mayúsculas:

```
const http = require('http');
const uc = require('upper-case');

http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(uc.upperCase("¡Hola The Bridge!"));
  res.end();
}).listen(8080);
```

## Instalando Nodemon

- nodemon es una herramienta que ayuda a desarrollar aplicaciones basadas en node.js al reiniciar automáticamente la aplicación cuando se detectan cambios en los archivos del directorio.

Definimos que es una dependencia de desarrollo

```
$npm install -D nodemon
```

# Script para Nodemon

- Ahora en nuestro package.json creamos un script para ejecutar nodemon.
- Para ejecutarlo, abrimos la consola y ejecutamos : **\$ npm run dev**

```
"scripts": {  
  "start": "node index.js",  
  "dev": "nodemon index.js"  
},
```