

RODRIGO MARTÍNEZ DELGADO  
DESARROLLO WEB ENTORNO CLIENTE

# ÍNDICE

20250207 – Instalación Node.js y concepto módulos.....	1
20250210 – Curso Node.js .....	1
20250212 - Git Visual Studio y Ejercicio1 .....	3
20250213 – Ejercicio1 .....	3
20250214 – Ejercicio2 y Inicio React.....	3
20250217 – Ejercicio 2 y inicio JSX.....	4
20250219 – Ejercicio 2 .....	5
20250220 – Ejercicio 3 .....	6
20250222 – Ejercicio 4 y 6 .....	7

## 20250207 – Instalación Node.js y concepto módulos

Instalación del entorno de ejecución Node.js en local y posterior explicación del sistema de módulos tanto con un ejemplo facilitado en la plataforma como la creación de un pequeño ejemplo.

- ejemplo-modulos-20250207
- ejemplo-modulos-avanzado-20250207

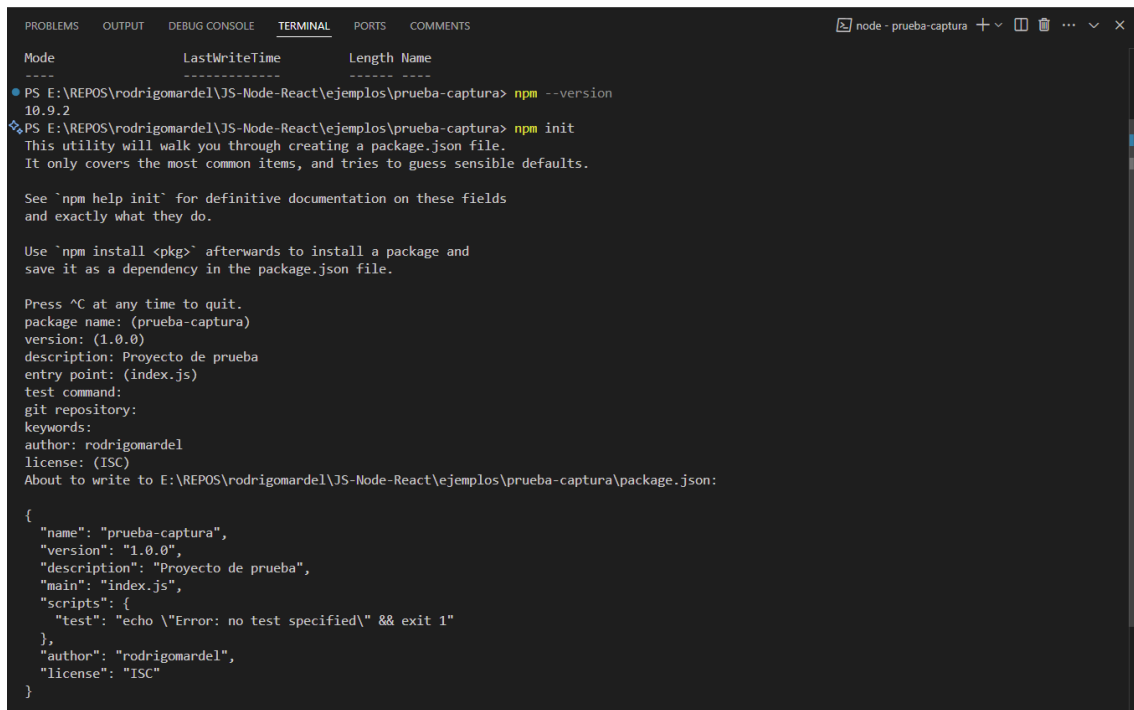
## 20250210 – Curso Node.js

Visualización de parte del curso del divulgador [midudev](#) sobre que es Node.js, Window VS GlobalThis, NPM, Instalar PicoColors.

Posteriormente se ven algunos ejemplos de módulos:

- node:os
- node:path
- node:fs

Creación de nuestro propio proyecto usando el módulo picocolors:



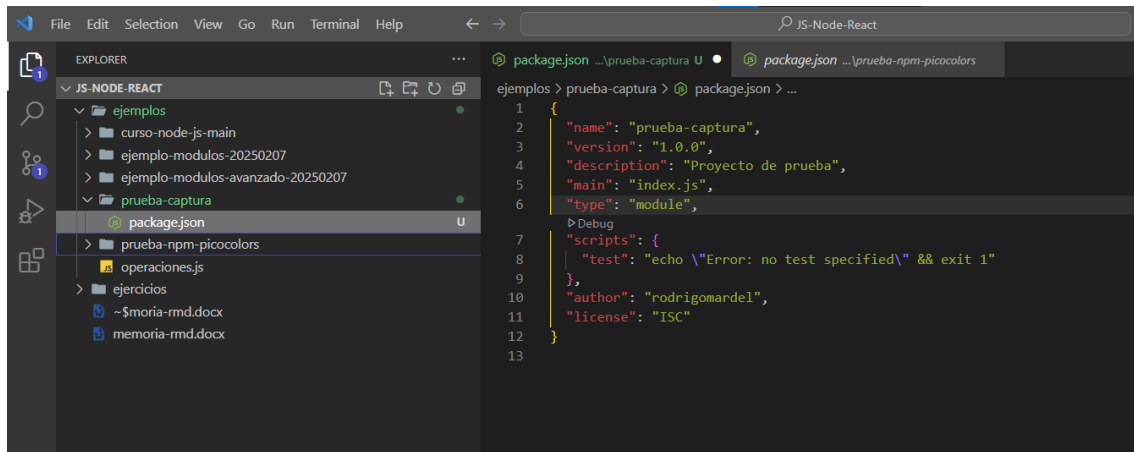
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
node - prueba-captura + - - - - - x

Mode LastWriteTime Length Name
----
PS E:\REPOS\rodrigomardel\JS-Node-React\ejemplos\prueba-captura> npm --version
10.9.2
PS E:\REPOS\rodrigomardel\JS-Node-React\ejemplos\prueba-captura> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

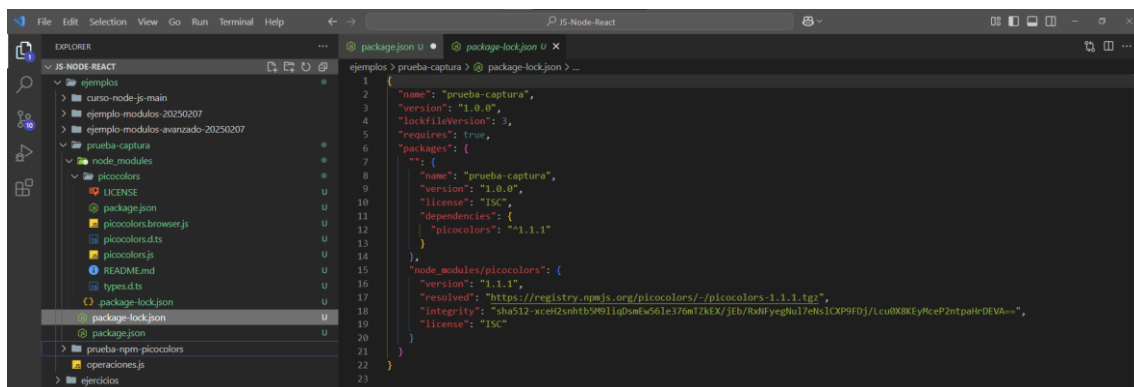
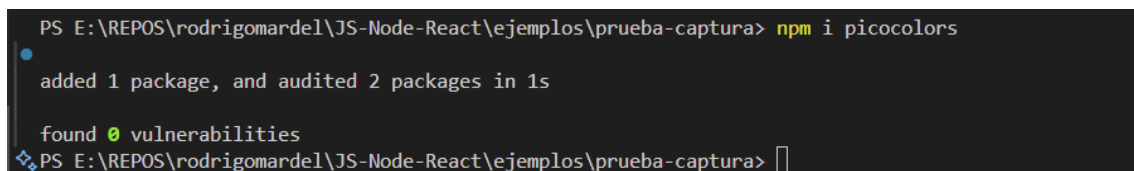
Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (prueba-captura)
version: (1.0.0)
description: Proyecto de prueba
entry point: (index.js)
test command:
git repository:
keywords:
author: rodrigomardel
license: (ISC)
About to write to E:\REPOS\rodrigomardel\JS-Node-React\ejemplos\prueba-captura\package.json:
{
  "name": "prueba-captura",
  "version": "1.0.0",
  "description": "Proyecto de prueba",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "rodrigomardel",
  "license": "ISC"
}
```



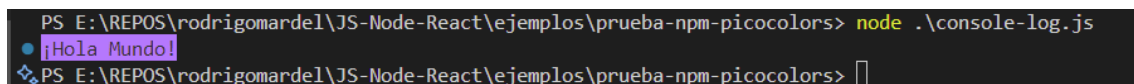
Nos crea un package.json con la propiedades de nuestro proyecto al cual agregamos "type: module" para evitar las extensiones .mjs

Instalamos el módulo [picocolors](#):



Pequeña prueba dando color de fondo a un console.log

- prueba-npm-picocolors-20250210



## 20250212 - Git Visual Studio y Ejercicio1

Creación de repositorio en Github y clonación en local para trabajar desde Visual Studio Code. Se ha realizado sin problema a excepción de que VS no había conectado correctamente el repositorio.

Comandos utilizados:

```
winget install --id Git.Git -e --source winget
```

```
git clone https://github.com/rodrigomardel/JS-Node-React.git
```

```
git remote add pb github.com/rodrigomardel/JS-Node-React.git
```

### [Realización ejercicio1:](#)

A la hora de retornar un valor en una función en el js exportado, no se puede recoger en el js que importo dicha función (revisar fetch y lógica).

## 20250213 – Ejercicio1

Repaso de los conceptos promesas, asincronía y funciones callback.

### [Ejercicio1 promesas:](#)

El problema era el uso del return dentro de la promesa en el método then como retorno de los datos para la función principal.

Se usa como solución la llamada a otra función, dentro del mismo script, para el tratamiento de los datos. De esta forma solo se exporta la función principal.

## 20250214 – Ejercicio2 y Inicio React

Investigación de los módulos necesarios y su funcionalidad para crear una aplicación en Node.js para añadir y listar tareas.

El **módulo fs** de Node.js se utiliza para interactuar con el sistema de archivos, es decir, para leer, escribir, actualizar, eliminar archivos, etc. Su función principal es manipular archivos de forma sincrónica o asincrónica.

Visualización de parte del curso del divulgador [midudev](#) sobre React (Cómo usar React sin dependencias). Explicación del ejemplo (ejemplos-react/react-sin-dependencias-20250214.html) y realización del ejercicio 5.

## 20250217 – Ejercicio 2 y inicio JSX

A la hora de crear la aplicación el planteamiento era erróneo, estaba importando módulos de Node para ser usados en el entorno de ejecución del navegador, esto no es posible. Hay que tener en cuenta que son entornos de ejecución distintos, con diferentes características.

Inicio a JSX con la visualización del curso del divulgador [midudev](#) (JSX desde cero) explicación posterior probando el código en los dos compiladores principales [SWC](#) y [Babel](#)

- ejemplos-react-con-JSX-20250217

Nuevo plantamiento del ejercicio 2. Crear un contador que guarda su valor en un archivo de texto (contador.txt). Puedes incrementar, decrementar o ver el valor actual del contador desde la terminal.

## 20250219 – Ejercicio 2

Finalización del ejercicio 2:

```
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> tree /f
• Listado de rutas de carpetas para el volumen DATOS
El número de serie del volumen es 7C80-2B6A
E:.\
├── contador.js
├── contador.txt
├── package-lock.json
├── package.json
├── node_modules
│   ├── .package-lock.json
│   ├── chalk
│   │   ├── license
│   │   ├── package.json
│   │   └── readme.md
│   └── source
│       ├── index.d.ts
│       ├── index.js
│       └── utilities.js
│   └── vendor
│       ├── ansi-styles
│       │   ├── index.d.ts
│       │   └── index.js
│       └── supports-color
│           ├── browser.d.ts
│           ├── browser.js
│           ├── index.d.ts
│           └── index.js
```

Alerta de type:module

```
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js mostrar
(node:10420) [MODULE_TYPELESS_PACKAGE_JSON] Warning: Module type of file:///E:/REPOS/rodrigomardel/JS-Node-React/ejercicios/ej2_prog-contador/contador.js is not
specified and it doesn't parse as CommonJS.
Reparsing as ES module because module syntax was detected. This incurs a performance overhead.
To eliminate this warning, add "type": "module" to E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador\package.json.
(Use `node --trace-warnings ...` to show where the warning was created)
Valor actual del contador: 2
```

```

ejercicios > ej2_prog-contador > package.json > ...
1  {
2    "type": "module",
3    "dependencies": {
4      "chalk": "^5.4.1"
5    }
6  }
7

```

Funcionamiento:

```

PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js aumentar
▲ Contador aumentado: 1
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js aumentar
▲ Contador aumentado: 2
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js disminuir
▼ Contador disminuido: 1
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js disminuir
? Comandos disponibles: aumentar, disminuir, mostrar
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js disminuir
▼ Contador disminuido: 0
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador> node contador.js mostrar
■ Valor actual del contador: 0
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej2_prog-contador>

```

Lectura de los enunciados de los ejercicios 3 y 4. Primer planteamiento del ejercicio 4 de su posible estructuración.

## 20250220 – Ejercicio 3

Algunos pequeños problemas con la sintaxis y alcance.

```

const os = require("node:os");
          ^
ReferenceError: require is not defined in ES module scope, you can use import instead

E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej3_modulos-os&fs\systemInfo.js:23
writeFile(filePath, infoText, (err) => {
^
ReferenceError: writeFile is not defined

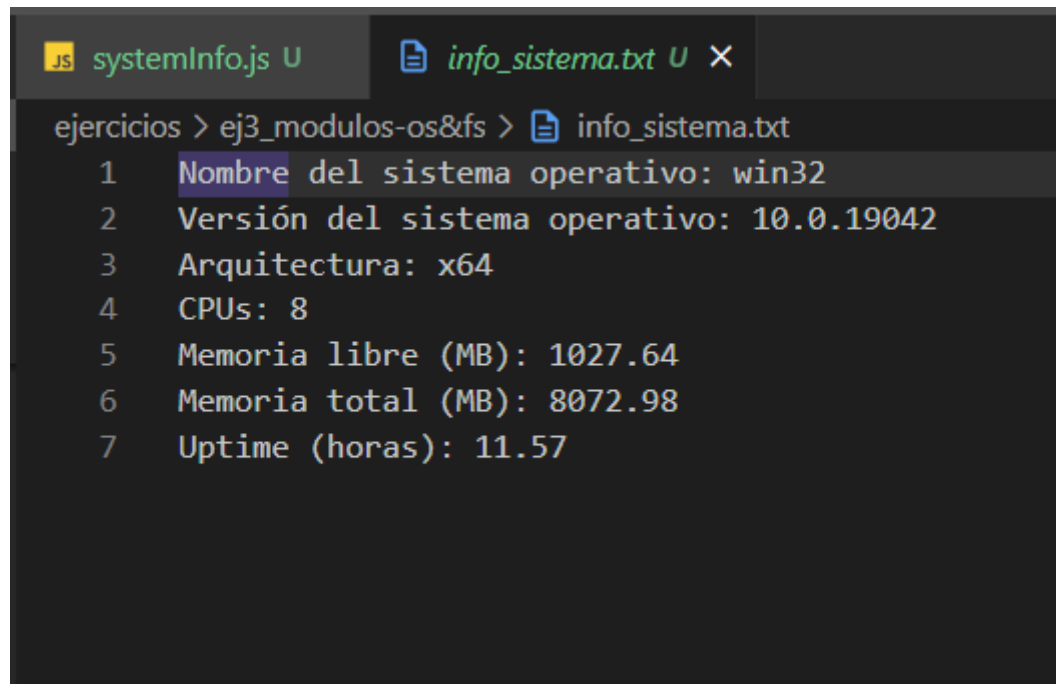
```



## Funcionamiento:

```
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej3_modulos-os&fs> node .\systemInfo.js
Archivo creado con éxito: info_sistema.txt

Contenido del archivo:
Nombre del sistema operativo: win32
Versión del sistema operativo: 10.0.19042
Arquitectura: x64
CPUs: 8
Memoria libre (MB): 1027.64
Memoria total (MB): 8072.98
Uptime (horas): 11.57
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej3_modulos-os&fs> 
```



The screenshot shows a code editor with two tabs: 'systemInfo.js' and 'info\_sistema.txt'. The 'info\_sistema.txt' tab is active, displaying the following content:

```
ejercicios > ej3_modulos-os&fs > info_sistema.txt
1  Nombre del sistema operativo: win32
2  Versión del sistema operativo: 10.0.19042
3  Arquitectura: x64
4  CPUs: 8
5  Memoria libre (MB): 1027.64
6  Memoria total (MB): 8072.98
7  Uptime (horas): 11.57
```

## 20250222 – Ejercicio 4 y 6

Investigación y uso de la librería readline para interactuar con el usuario.

```
PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej4_librerias> node usuario.js
● Ingresa tu nombre: Rodrigo
  Ingresa tu edad: 37
  ¡Hola, Rodrigo! Tienes 37 años.
● PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej4_librerias> node usuario.js
  Ingresa tu nombre: Jaime
  Ingresa tu edad: 15
  ¡Hola, Jaime! Tienes 15 años.
● PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej4_librerias> node usuario.js
  Ingresa tu nombre: Miguel
  Ingresa tu edad: 72
  ¡Hola, Miguel! Tienes 72 años.
❖ PS E:\REPOS\rodrigomardel\JS-Node-React\ejercicios\ej4_librerias> 
```