



RODRIGO MARTÍNEZ DELGADO
DESARROLLO WEB ENTORNO CLIENTE

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

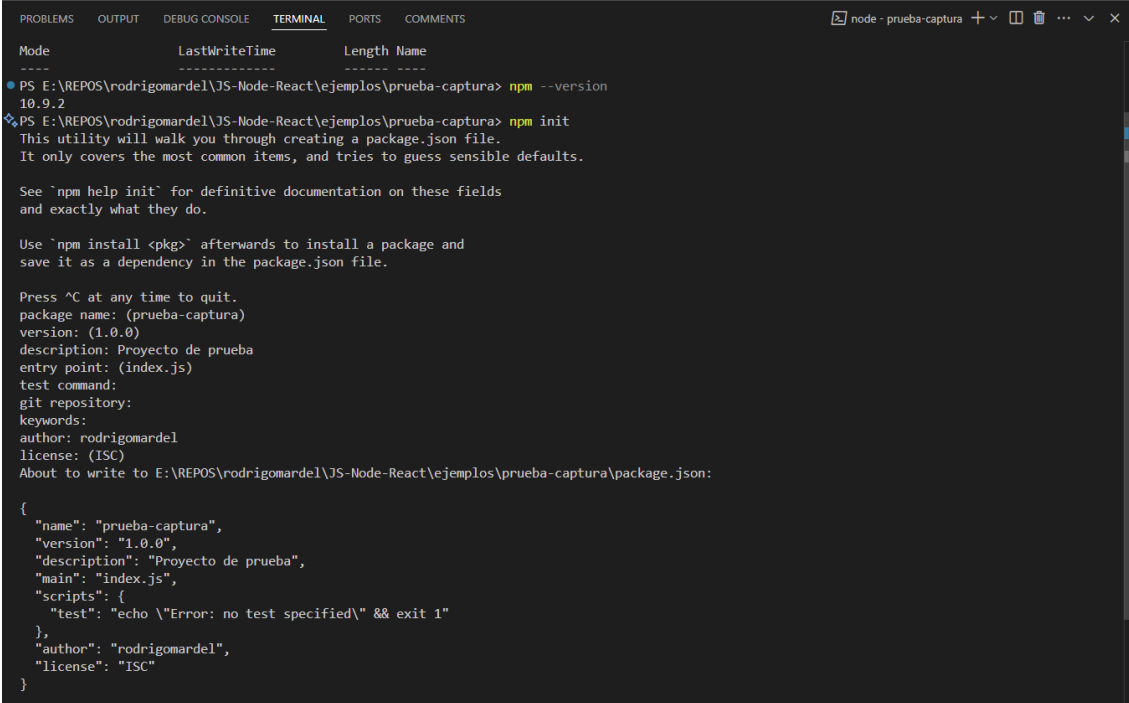
20250210 – Curso Node.js

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:



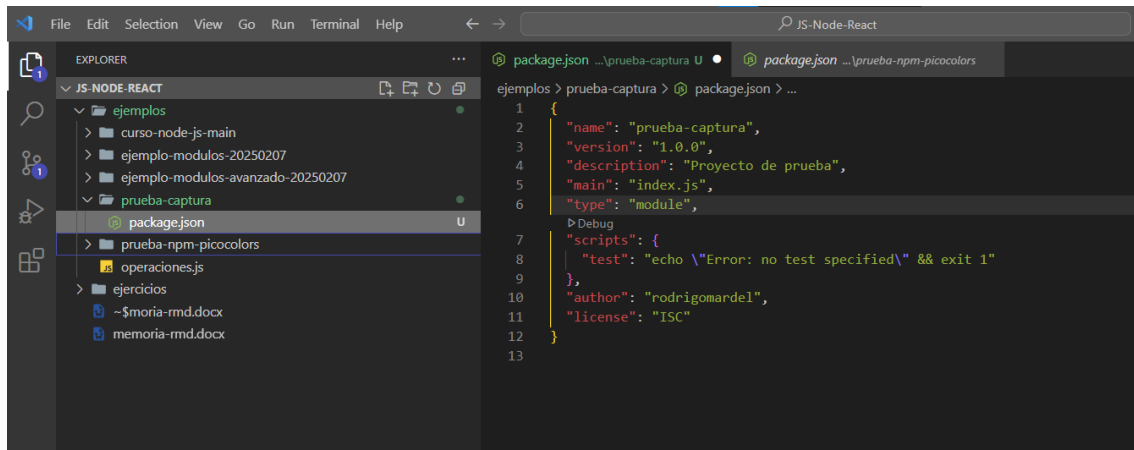
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
node - prueba-captura + - - - - - x

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.

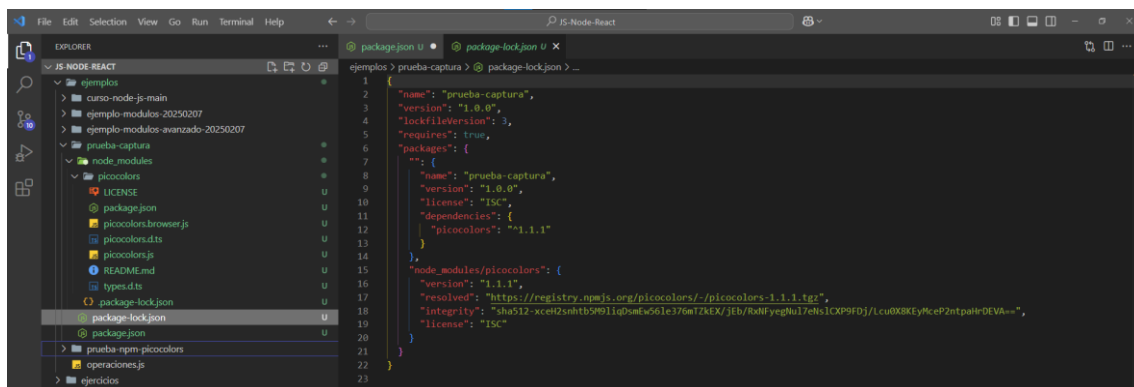
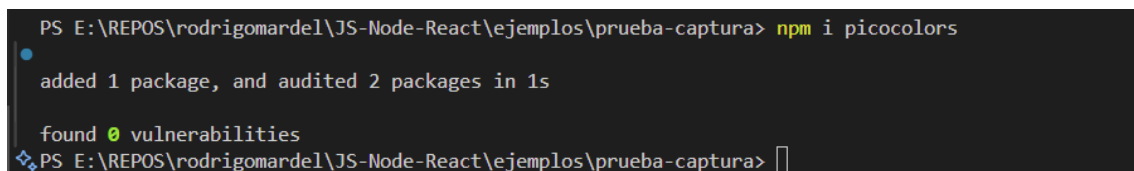
Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

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"
}
```



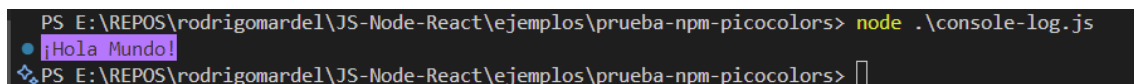
Nos crea un package.json con la propiedades de nuestro proyecto al cual agregamos "type: module" para evitar las extensiones .mjs

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 – 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) e inicio del ejercicio 5.