

Node - Dossier

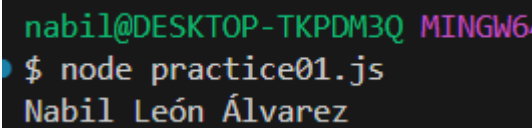
- [Práctica 01](#)
 - [Práctica 02](#)
 - [Práctica 03](#)
 - [Práctica 04](#)
 - [Práctica 05](#)
 - [Práctica 06](#)
 - [Práctica 07](#)
 - [Práctica 08](#)
-

Práctica 01

📁 Crear una aplicación que al ejecutar: `node app.js` nos muestre en pantalla nuestro nombre completo

```
console.log('Nabil León Álvarez');
```

- Captura:



```
nabil@DESKTOP-TKPDM3Q MINGW64  
$ node practice01.js  
Nabil León Álvarez
```

Práctica 02

📁 Hacer uso de nodemon con la app anterior `nodemon app.js` y comprobar que mientras editamos el fichero (agregamos delante de nuestro nombre: "nombre del alumno: ") y guardamos, automáticamente nos muestra la ejecución de los cambios

- Captura:

```
nabil@DESKTOP-TKPDM3Q MINGW64 ~/repositorios-git/pgl-nabil/unit-0
$ nodemon practice01.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node practice01.js`
Nabil León Álvarez
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node practice01.js`
Nombre del alumno: Nabil León Álvarez
[nodemon] clean exit - waiting for changes before restart
```

Práctica 03

📁 Crear la aplicación descrita

```
function getTable(num){
  let limit = 10;
  let result = "";
  for(let i = 1; i <= limit; i++){
    result += `${num} * ${i} = ${num * i}\n`;
  }

  return result;
}

console.log(getTable(4));
```

- Captura:

```
nabil@DESKTOP-TKPDM3Q MINGW64 ~/repositorios-git/pgl-nabil/
$ node practice03.js
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
```

Práctica 04

📁 Crear la aplicación descrita, pero si hay un error no mostrarlo sino mostrar un mensaje que diga que no se pudo grabar

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

function getTable(num){
  let limit = 10;
  let result = "";
  for(let i = 1; i <= limit; i++){
    result += `${num} * ${i} = ${num * i}\n`;
  }

  return result;
}

let num = 4;

fs.writeFile(
  'practice04.txt', getTable(num), (error)=>{
    if(error){
      console.log('No se pudo grabar el archivo');
    } else{
      console.log('Archivo creado con exito');
    }
  }
);
```

- Captura:

```
nabil@DESKTOP-TKPDM3Q MINGW64 ~/
$ node practice04.js
Archivo creado con exito
```

```
practice04.txt X
practices > practice04.txt
1  4 * 1 = 4
2  4 * 2 = 8
3  4 * 3 = 12
4  4 * 4 = 16
5  4 * 5 = 20
6  4 * 6 = 24
7  4 * 7 = 28
8  4 * 8 = 32
9  4 * 9 = 36
10 4 * 10 = 40
11
```

Práctica 05

📁 Crear la aplicación descrita. El fichero: manejarTabla.js debe exportar un método: escribir(nombreDelFichero, textoEscribir) que tiene que devolver una promesa. En el interior del método escribir() se llama a la función: fs.writeFileSync()

```
function getTable(num){
  let limit = 10;
  let result = "";
  for(let i = 1; i <= limit; i++){
    result += `${num} * ${i} = ${num * i}\n`;
  }

  return result;
}

module.exports.createTable = getTable;

const { rejects } = require("assert");
const fs = require(`fs`);

function write(fileName, content) {
  return new Promise((resolve, reject) => {
    try {
      fs.writeFileSync(fileName, content);
      resolve("File saved successfully");
    } catch (error) {
      console.error(error);
    }
  });
}

module.exports = {write};

const {write} = require("../utils/fileManagement");
const {createTable} = require("../model/table");

write("practice05.txt", createTable(7))
  .then(console.log("save ok"))
  .catch(err => console.log("error"));
```

- Captura:

```
nabil@DESKTOP-TKPDM3Q MINGW64
05 (main)
$ node app.js
save ok
```

```
practices > practice05 > practice05.txt
1 7 * 1 = 7
2 7 * 2 = 14
3 7 * 3 = 21
4 7 * 4 = 28
5 7 * 5 = 35
6 7 * 6 = 42
7 7 * 7 = 49
8 7 * 8 = 56
9 7 * 9 = 63
10 7 * 10 = 70
11
```

Práctica 06

📁 Variar la aplicación que tenemos para grabar una tabla en un fichero para que le pasemos como parámetro el número del que queremos que nos haga la tabla

```
function getTable(num){
  let limit = 10;
  let result = "";
  for(let i = 1; i <= limit; i++){
    result += `${num} * ${i} = ${num * i}\n`;
  }

  return result;
}

module.exports.createTable = getTable;

const { rejects } = require("assert");
const fs = require(`fs`);

function write(fileName, content) {
  return new Promise((resolve, reject) => {
    try {
      fs.writeFileSync(fileName, content);
      resolve("File saved successfully");
    } catch (error) {
      reject(error);
    }
  });
}

module.exports = {write};
```

```
const {write} = require("./utils/fileManagement");
const {createTable} = require("./model/table");

function writeTableParam(){

    let num = parseInt(process.argv[2]);

    if(!Number.isInteger(num) || num <= 0){
        console.log("Error: You must use an integer and positive number.");
        process.exit(1);
    }

    write("practice06.txt", createTable(num))
    .then(console.log("save ok"))
    .catch(err => console.log("error"));
}

writeTableParam();
```

- Captura:

```
nabil@DESKTOP-TKPDM3Q MING
06 (main)
$ node app.js 1
save ok
```

```
practices > practice06 > practice06.txt
1 1 * 1 = 1
2 1 * 2 = 2
3 1 * 3 = 3
4 1 * 4 = 4
5 1 * 5 = 5
6 1 * 6 = 6
7 1 * 7 = 7
8 1 * 8 = 8
9 1 * 9 = 9
10 1 * 10 = 10
```

Práctica 07

📁 Variar el ejercicio de la tabla con yargs y nos construya la tabla de un valor (en el ejemplo vemos el 7) mediante: `node app --tabla=7`

```
function getTable(num){
    let limit = 10;
    let result = "";
    for(let i = 1; i <= limit; i++){
        result += `${num} * ${i} = ${num * i}\n`;
    }
}
```

```
    return result;
  }

  module.exports.createTable = getTable;

  const { rejects } = require("assert");
  const fs = require(`fs`);

  function write(fileName, content) {
    return new Promise((resolve, reject) => {
      try {
        fs.writeFileSync(fileName, content);
        resolve("File saved successfully");
      } catch (error) {
        reject(error);
      }
    });
  }

  module.exports = {write};

  const {write} = require("../utils/fileManagement");
  const {createTable} = require("../model/table");

  const yargs = require("yargs/yargs");
  const { hideBin } = require('yargs/helpers')
  const argv = yargs(hideBin(process.argv)).argv

  console.log(argv);

  function writeTable(){
    write("practice07.txt", createTable(argv.table))
      .then(console.log("save ok"))
      .catch(err => console.log("error"));
  }

  writeTable();
```

- Captura:

```
nabil@DESKTOP-TKPDM3Q MINGW64 ~/reposito
07 (main)
$ node app.js --table=21
{ _: [], table: 21, '$0': 'app.js' }
save ok
```

```
practices > practice07 > practice07.txt
```

```
1 21 * 1 = 21
2 21 * 2 = 42
3 21 * 3 = 63
4 21 * 4 = 84
5 21 * 5 = 105
6 21 * 6 = 126
7 21 * 7 = 147
8 21 * 8 = 168
9 21 * 9 = 189
10 21 * 10 = 210
```

Práctica 08

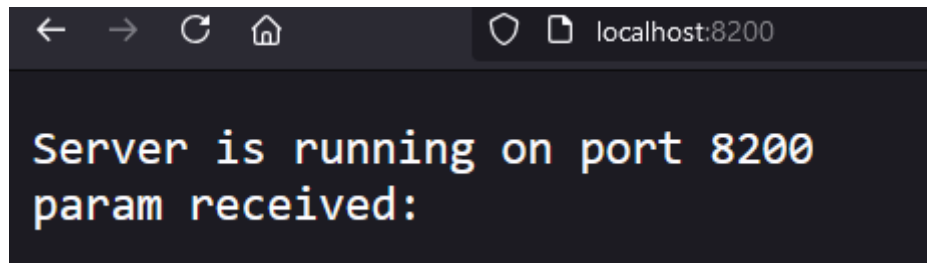
📁 Crear el ejercicio descrito. Modificar el puerto al 8000 comprobar que arranca correctamente y pasarle diferentes parámetros tomar captura de pantalla de lo obtenido

```
const http = require('http');
const url = require('url');

const port=8200;

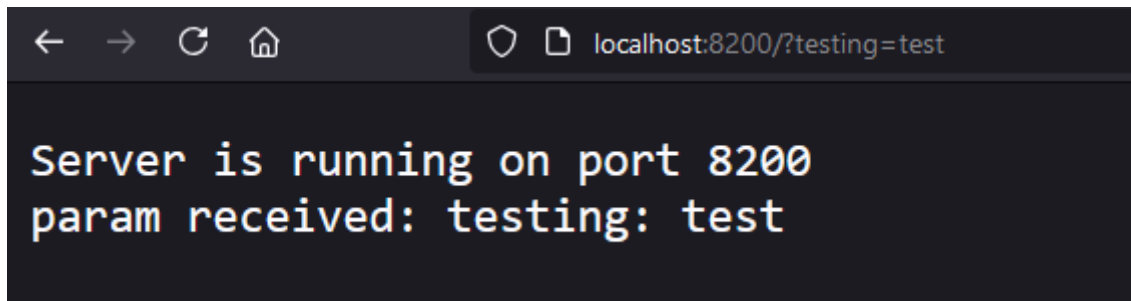
const server = http.createServer((req, res) => {
  res.write(`Server is running on port ${port}\n`);
  const queryObject = url.parse(req.url, true).query;
  console.log(queryObject);
  let text="";
  Object.entries(queryObject).forEach((par) => text +=
    (par[0]+": "+par[1]+"\n"));
  res.write(`param received: ${text}`);
  res.end();
}).listen(port);
```

- Captura:



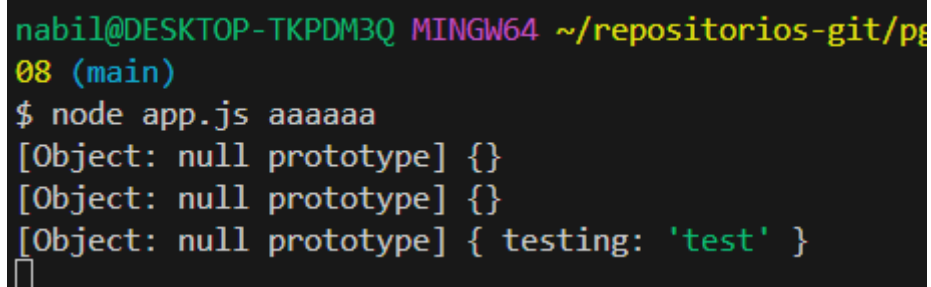
← → ↻ 🏠 localhost:8200

Server is running on port 8200
param received:



← → ↻ 🏠 localhost:8200/?testing=test

Server is running on port 8200
param received: testing: test



```
nabil@DESKTOP-TKPDM3Q MINGW64 ~/repositorios-git/pg
08 (main)
$ node app.js aaaaaa
[Object: null prototype] {}
[Object: null prototype] {}
[Object: null prototype] { testing: 'test' }
```
