

Dossier

NODE

CEDRIC CHRISTOPH

19/01/22

ÍNDICE

PÁGINAS

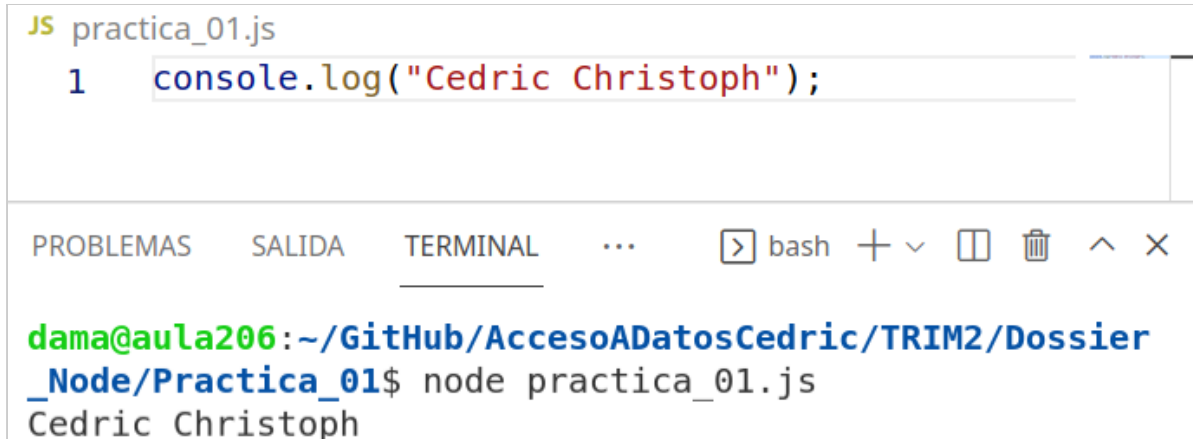
Práctica 01	3
Práctica 02	3
Práctica 03	4
Practica 04	5
Practica 05	6
Práctica 06	7
Práctica 07	8
Práctica 08	9

Práctica 01

Crear una aplicación que al ejecutar:

```
node app.js
```

nos muestre en pantalla nuestro nombre completo.



The screenshot shows a code editor with a file named `practica_01.js` containing the following JavaScript code:

```
1 console.log("Cedric Christoph");
```

Below the code editor is a terminal window. The terminal tabs are labeled `PROBLEMAS`, `SALIDA`, and `TERMINAL`. The terminal shows the command `node practica_01.js` being executed, resulting in the output `Cedric Christoph`.

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.

```
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node practica_02.js`
Cedric Christoph
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node practica_02.js`
nombre del alumno: Cedric Christoph
[nodemon] clean exit - waiting for changes before restart
█
```

Práctica 03

Crear la aplicación descrita

```
JS practica_03.js > getTabla
1
2 // Cedric Christoph
3 console.log(getTabla(5));
4
5 function getTabla(tabla) {
6     let limite = 10;
7     let respuesta = "";
8     for (let i = 1; i <= limite; i++) {
9         respuesta += `${tabla} * ${i} = ${tabla * i} \n`;
10    }
11    return respuesta;
12 }
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN bash + - [] [X] ^ X

```
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_03$ node practica_03.js
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Práctica 04

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

```
JS practica_04.js > fs.writeFile('tabla.txt') callback
1  // Cedric Christoph
2  const fs = require('fs');
3
4  function getTabla(tabla) {
5      let limite = 10;
6      let respuesta = "";
7      for (let i = 1; i <= limite; i++) {
8          respuesta += `${tabla} * ${i} = ${tabla * i} \n`;
9      }
10     return respuesta;
11 }
12
13 fs.writeFile(
14     'tabla.txt',
15     getTabla(4),
16     (err) => {
17         if (err)
18             console.log("No se pudo guardar");
19         else
20             console.log("Se ha grabado");
21     }
22 );
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN bash + v [] [X] ^

dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_04\$ node practica_04.js
Se ha grabado

tabla.txt output:

```
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 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()

JS app.js > ...

```
1 // APP.js
2 // Cedric Christoph
3 const {escribir} = require("../utils/fileManager");
4 const {crearTabla} = require("../modelo/tabla");
5
6 escribir("tabla.txt", crearTabla(7))
7   .then(console.log("ok grabado"))
8   .catch(err => console.log(err));
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN

```
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_05$ node app.js
ok grabado
```

utils > JS fileManager.js > ...

```
1 // fileManager.js
2 // Cedric Christoph
3 const fs = require('fs');
4
5 async function escribir(fichero, contenido) {
6   fs.writeFileSync(fichero, contenido)
7 }
8 exports.escribir = escribir;
```

modelo > JS tabla.js > ...

```
1 // tabla.js
2 // Cedric Christoph
3 function getTabla(tabla) {
4   let limite = 10;
5   let respuesta = "";
6   for (let i = 1; i <= limite; i++) {
7     respuesta += `${tabla} * ${i} = ${tabla * i} \n`;
8   }
9   return respuesta;
10 }
11 exports.crearTabla = getTabla;
```

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.

```
JS app.js > ...
1  // APP.js
2  // Cedric Christoph
3  const {escribir} = require("../utils/fileManager");
4  const {crearTabla} = require("../modelo/tabla");
5
6  let respuesta = process.argv[2];
7
8  if (respuesta !== null) {
9      escribir("tabla.txt", crearTabla(Number(respuesta)))
10         .then(console.log("ok grabado"))
11         .catch(err => console.log(err));
12 } else {
13     console.log("No se han establecidos parámetros de entrada")
14 }
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN

```
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_06$ node app.js
No se han establecidos parámetros de entrada
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_06$ node app.js 5
ok grabado
```

tabla.txt output

```
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Práctica 07

Variar el ejercicio de la tabla con yargs y nos construya la tabla de un valor (en el ejemplo vemos el 8) mediante:

```
node app --tabla=8
```

JS app.js > ...

```
1  // APP.js
2  // Cedric Christoph
3  const {escribir} = require("../utils/fileManager");
4  const {crearTabla} = require("../modelo/tabla");
5
6  const yargs = require('yargs/yargs')
7  const { hideBin } = require('yargs/helpers')
8  const argv = yargs(hideBin(process.argv)).argv
9
10 let respuesta = argv.tabla;
11
12 if (respuesta !== null) {
13     escribir("tabla.txt", crearTabla(Number(respuesta)))
14     .then(console.log("ok grabado"))
15     .catch(err => console.log(err));
16 } else {
17     console.log("No se ha establecido el parámetro de " +
18     "entrada\n --tabla");
19 }
```

PROBLEMAS SALIDA TERMINAL

✓ **TERMINAL**

```
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_07$ node app.js
No se ha establecido el parámetro de entrada
--tabla
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_07$ node app.js --tabla=8
ok grabado
```


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.

```
JS app.js > http.createServer() callback
1  const http = require('http');
2  const url = require('url');
3
4  const port = 9000;
5  http.createServer( (req, res) => {
6      res.write(`Server running on port ${port}`);
7      const queryObject = url.parse(req.url, true).query;
8      console.log(queryObject);
9      let text = "";
10     Object.entries(queryObject)
11     .forEach( (par) => text += (par[0] + ": " + par[1]));
12     res.write(` Received parameter ${text}`);
13     res.end();
14 }).listen(port);
```

PROBLEMAS SALIDA TERMINAL node + -

```
dama@aula206:~/GitHub/AccesoADatosCedric/TRIM2/Dossier_Node/Practica_08$ node app.js
[Object: null prototype] { parametro1: 'hola_que_tal' }
```

