

CURSO PROGRAMACIÓN WEB FULL STACK

NIVEL 2

Módulo 5 - Programación Backend

Trabajo Práctico 8

¿Cómo importar módulos en node?

- ★ Importar módulos del core: `const filesystems = require('fs');`
- ★ Importar módulos de npm: `const express = require('express');`
- ★ Importar un archivo en un proyecto: `const server = require('./archivo.js');`
- ★ Importar un archivo JSON: `const configuracion = require('./config/db.json');`
- ★ Importa un index.js que se encuentra en un directorio sin tener que especificarlo `const rutas = require('./rutas');`

PROGRAMA LECTOR JSON 1

Se lee los archivos JSON con el módulo fs (<https://nodejs.org/api/fs.html>) con las funciones **readFileSync**. La función **readFileSync** lee datos de un archivo de forma **síncrona** Nota: `personas.json` es el archivo generado en el ejercicio 1 del tp7.

```
const fs = require('fs');  
let rawdata = fs.readFileSync('./personas.json');  
let persona = JSON.parse(rawdata);  
console.log(persona);
```

PROGRAMA LECTOR JSON 2

Utilizando **require** es similar a **readFileSync**, también lee datos de un archivo de forma **síncrona** y se lo puede utilizar desde cualquier lugar. Aunque tiene sus limitaciones.

- Require es una función sincrónica y se llama solo una vez, lo que significa que las llamadas reciben un resultado en caché. Si el archivo está actualizado, no puede volver a leerlo con este método.
- Su archivo debe tener la extensión '.json', por lo que no puede ser tan flexible. Sin la extensión adecuada requiere no trata el archivo como un archivo JSON.

```
'use strict';  
let jsonData = require('./personas.json');  
console.log(jsonData);
```

PROGRAMA LECTOR JSON 3

La función **readFile** lee los datos del archivo de forma **asincrónica**. Llama a la función, el proceso de lectura del archivo comienza e inmediatamente el control cambia a la siguiente línea ejecutando las líneas restantes de código. Una vez que se han cargado los datos del archivo, esta función llamará a la función de devolución de llamada que se le proporcionó. De esta manera, no está bloqueando la ejecución del código mientras espera que el sistema operativo se comunique con usted con los datos.

```
'use strict';  
const fs = require('fs');  
fs.readFile('personas.json', (err, data) => {  
  if (err) throw err;  
  console.log(JSON.parse(data));  
});
```

OBTENER NOMBRES

Json se utiliza para intercambiar información. La siguiente función permite extraer un dato de un Objeto JSON y luego lo imprime en consola.

```
const fs = require('fs');  
const rawdata = fs.readFileSync('./personas.json', 'utf8');
```

```
var persona = JSON.parse(rawdata);  
for(var i=0;i<persona['personas'].length;i++){  
console.log(persona['personas'][i]['nombre']);  
}
```

Ejercicio 1

a) Modificar el programa lector 1, reemplazar las 2 últimas líneas por:

```
console.log(JSON.stringify(rawdata));
```

¿Qué sucede cuando utilizamos `JSON.stringify`? Por qué?

¿Qué sucede cuando utilizamos `JSON.parse`? Por qué?

Comentar las respuestas y guardar el programa en `ejercicio1.js`

Ejercicio 2

a) Modificar el programa lector 2, reemplazar la última línea por:

```
console.log(JSON.stringify(jsonData));
```

¿Qué sucede cuando utilizamos `JSON.stringify`? Por qué?

b) Agregar al final:

```
console.log(JSON.parse(jsonData));
```

¿Qué sucede cuando utilizamos `JSON.parse`? Por qué?

Comentar las respuestas y guardar el programa en `ejercicio2.js`

Ejercicio 3

a) Modificar el programa lector 3, reemplazar la anteúltima línea por:

```
console.log(JSON.stringify(data));
```

¿Qué sucede cuando utilizamos `JSON.stringify`?

b) Agregar debajo:

```
console.log(data);
```

¿Que sucede? Por que?

Comentar las respuestas y guardar el programa en ejercicio3.js

Ejercicio 4

Partiendo del programa “Obtener nombres”, crear una función llamada getNombre que reciba el arreglo de las personas y busque todas las personas de la localidad Posadas. Si encuentra debe imprimir el nombre y el apellido de la persona.

Llamar a la función getNombre(persona);

Guardar el programa en ejercicio4.js

Al ejecutar node ejercicio4.js la salida en consola debería ser la siguiente:

```
Maria Nieves  
Ramon Rodriguez
```

Ejercicio 5

Crear un programa (ejercicio5.js) capaz de leer el archivo departamentos.json.

Luego recorrer e imprimir nombre + puesto de los empleados de tecnologia.

Al ejecutar node ejercicio5.js la salida en consola debería ser la siguiente:

```
Pablo Richmon PM  
Marta Fernandez TechLead  
Pedro Mendez FullStack js
```

Los ejercicios subirlos a un repositorio público y compartir la URL en canvas.

Generar un espacio de trabajo con los issues en el repositorio :

<https://github.com/FacundoRauber/fullstack2022/issues/new/choose>

Bibliografía, instaladores, herramientas y mas

https://docs.google.com/presentation/d/1VnpyKTI2zMsabWL_VEm2EWzZ1vvPxedKOOyNcRKfJEw/edit#slide=id.g140ad7a5d79_0_420

<https://www.w3schools.com/nodejs/default.asp>

<https://www.json.org/json-es.html>

<https://nodejs.org/api/fs.html>

<https://nodejs.dev/en/learn/reading-files-with-nodejs/>

<https://www.youtube.com/watch?v=DeUyzJwlu1k>

https://www.w3schools.com/js/js_json_parse.asp

https://www.w3schools.com/js/js_json_stringify.asp