

Trabajo Practico N° 8

```
// Ejercicio 1
// a)Modificar el programa lector 1, reemplazar las 2 últimas líneas por:
// Comentar las respuestas y guardar el programa en ejercicio1.js
// console.log(JSON.stringify(rawdata));
// ¿Qué sucede cuando utilizamos JSON.stringify? Por que?
const fs = require('fs');
let rawdata = fs.readFileSync('./personas.json');
let persona = (JSON.stringify(rawdata));
console.log(persona);
// Stringify convierte un objeto javascript en una cadena de texto con
// formato json. y se usa para enviar datos a un servidor web en el caso del
// ejercicio al pasar a su vez por la función readFileSync convierte el
// archivo personas.json en una serie de números que representan a cada
// carácter escrito. Sigue manteniendo el orden del objeto que se pretende
// convertir a cadena de texto, sin embargo, no es entendible para el
// usuario, solamente para el servidor.
// ¿Qué sucede cuando utilizamos JSON.parse? Por que?
const fs = require('fs');
let rawdata = fs.readFileSync('./personas.json');
let persona = JSON.parse(rawdata);
console.log(persona);
// Parse convierte una cadena de texto con formato json en un objeto
// javascript. y se usa generalmente cuando se recibe datos de un servidor
// web, ya que al recibir datos, estos siempre son una cadena de texto en el
// caso del archivo persona.json, como estamos recibiendo texto con formato
// json, lo transforma en un objeto javascript manteniendo el formato legible
// para el usuario ya que al recibir datos, estos siempre son una cadena de
// texto.
```

```
// Ejercicio 2
// Comentar las respuestas y guardar el programa en ejercicio2.js
// a)Modificar el programa lector 2, reemplazar la última línea por:
// console.log(JSON.stringify(jsonData));
// ¿Qué sucede cuando utilizamos JSON.stringify? Por que?
'use strict';
let jsonData = require('./personas.json');
console.log(JSON.stringify(jsonData));
// En este caso, al usar stringify trae los datos del objeto json y lo
// convierte en una cadena de texto en una sola línea.

// b)Agregar al final:
// console.log(JSON.parse(jsonData));
// ¿Qué sucede cuando utilizamos JSON.parse? Por que?
'use strict';
```

```
let jsonData = require('./personas.json');
console.log(JSON.parse(jsonData));
// Al tratar de parsear tira un error undefined, con error de sintaxis.
```

```
// Ejercicio 3
// Comentar las respuestas y guardar el programa en ejercicio3.js
// a) Modificar el programa lector 3, reemplazar la anteúltima línea por:
// console.log(JSON.parse(jsonData));
// ¿Qué sucede cuando utilizamos JSON.stringify?
'use strict';
const fs = require('fs');
fs.readFile('personas.json', (err, data) => {
  if (err) throw err;
  console.log(JSON.stringify(data));
});

// Vuelve a tirar la serie de números representando cada uno de los
// caracteres utilizados en el archivo personas.json.

// b) Agregar debajo:
// ¿Qué sucede? Por qué?
'use strict';
const fs = require('fs');
fs.readFile('personas.json', (err, data) => {
  if (err) throw err;
  console.log(JSON.parse(data));
});
console.log(data);

// Convierte cada dato del objeto en su valor hexadecimal.
```

```
// Ejercicio 4
// OBTENER NOMBRES

// 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 Rodrigue

const fs = require("fs");
const rawdata = fs.readFileSync("./personas.json", "utf8");
var persona = JSON.parse(rawdata);

function getNombre(buscar) {
  for (var i = 0; i < buscar["personas"].length; i++) {
    if (buscar["personas"][i]["Localidad"] === "Posadas") {
      console.log(buscar["personas"][i]["Nombre"] + " " +
buscar["personas"][i]["Apellido"]);
    }
  }
}

getNombre(persona);
```

```
// Ejercicio 5
// Crear un programa (ejercicio5.js) capaz de leer el archivo
departamentos.json.
// Luego recorrer e imprimir nombre + puesto de los empleados de
tecnología.
// Al ejecutar node ejercicio5.js la salida en consola debería ser la
siguiente:
// Pablo Richmon PM
// Marta Fernandez TechLead
// Pedro Mendez FullStack js

const fs = require("fs");
const rawdata = fs.readFileSync("./departamentos.json", "utf8");
var departamentos = JSON.parse(rawdata);
//console.log(departamentos);

function empleados(buscar) {
  for (var i = 0; i < buscar["departamentos"]["Tecnología"].length;
i++) {
    console.log(buscar["departamentos"]["Tecnología"][i]["nombre"] +
" " + buscar["departamentos"]["Tecnología"][i]["puesto"]);
  }
}
```

```
empleados(departamentos);
```