



**Certified Tech
Developer**

The Ultimate Degree



Programación Imperativa

Ejercicios Lectura y Escritura de Archivos

Objetivo

Vamos a armar una **app de clases de la materia** que nos permita conocer el estado de la misma utilizando las herramientas que hasta el momento hemos aprendido.

Micro desafíos:

1. Crear un archivo **clases.json** que contenga un array de objetos literales con formato JSON. Cada objeto literal deberá tener las propiedades: **título y estado**. En donde el título podrá ser cualquier cadena de texto por ejemplo *"Introduccion a la Programacion"* y el estado podrá ser: *"terminada"*, *"repasar"*, *"pendiente"*.

XtraPlus1: agregar otra propiedad **descripción**.

SOLUCIÓN :

<https://drive.google.com/file/d/1rinXVpDE4sPOdAkcQlyTq7l3Jd5sCEQE/view?usp=sharing>

2. Generar un archivo **app.js** que "consume" el archivo de **clases.json**. Mostrar el listado de clases existentes por consola. Para esto, ¿necesitamos usar el módulo nativo "fs"? ¿o con sólo `require("archivo.json")` ya obtenemos el array de objetos?.

XtraPlus2: Manejar el error si no encuentra el archivo. (try catch)

SOLUCIÓN:

```
console.log("***** MD1 punto 2) *****");
console.log(require('./clases.json'));
```

SOLUCIÓN XTRAS:

```
console.log("***** MD1 punto 2) XtraPlus2 *****");

const leerArchivoManejandoError = (rutaAlArchivo) => {
  let jsonData = ""
  try {
    jsonData = require(rutaAlArchivo)
  } catch (error) {
    console.log("ups! qué pena contigo... " + error);
  } finally {
    return jsonData
  }
}

console.log(leerArchivoManejandoError("./clases.json")); //lista contenido
console.log(leerArchivoManejandoError("./unArchivo.json")); //log del error
```

Sin try/catch

```
const fs = require('fs')
let clasesJSON = fs.readFileSync('./clases.json','utf-8')
let clases = JSON.parse(clasesJSON)
console.log(clases)
```

3. Crear una función para filtrar las clases por su estado, asignándolas a diferentes arrays de objetos literales y guardar todo esto ordenado por estado, en un archivo.json.

XtraPlus3: Mediante una arrow func guardar cada array en un archivo diferente según corresponda el estado. Por ejemplo: "repasar.json", "terminadas.json"

SOLUCIÓN:

```
console.log("***** MD1 punto 3) *****")
const fs = require('fs')

const divideBy = (estado) => {
  let parsedJson = leerArchivoManejandoError('./clases.json')
  let arrayFiltrado = parsedJson.filter(clase => clase.estado === estado)
  return arrayFiltrado
}

let terminadas = divideBy("terminada")
let repasar = divideBy("repasar")
let pendiente = divideBy("pendiente")
console.log(terminadas);
console.log(repasar);
console.log(pendiente);

data = [terminadas, repasar, pendiente]
fs.writeFileSync("./clasesOrdenadas.json", JSON.stringify(data))
```

SOLUCIÓN XTRAS:

```
console.log("***** MD1 punto 3) XtraPlus3 *****")
/*
let data = JSON.stringify(terminadas)
let ruta = "./terminadas.txt"
fs.writeFileSync(ruta, data)

data = JSON.stringify(repasar)
ruta = "./repasar.txt"
fs.writeFileSync(ruta, data)

data = JSON.stringify(pendiente)
ruta = "./pendiente.txt"
fs.writeFileSync(ruta, data) */

//con arrow func
const guardar = (ruta, datos) => {
  let data = JSON.stringify(datos)
  fs.writeFileSync(ruta, data)
}

guardar("./terminadas.txt", terminadas)
guardar("./repasar.txt", repasar)
guardar("./pendiente.txt", pendiente)
```

```
/*
console.log("***** MD1 punto 3) XtraPlus3++ *****")
const guardar = (datos) => {
  let name = Object.keys(datos)[0]
  let data = JSON.stringify(datos[name])
  name = "/" + name + ".txt"
  fs.writeFileSync(name, data)
}
guardar({ terminadas })
guardar({ repasar })
guardar({ pendiente }) */
```