DigitalHouse > Coding School

Programación Web Full Stack

Práctica Integradora

Objetivo

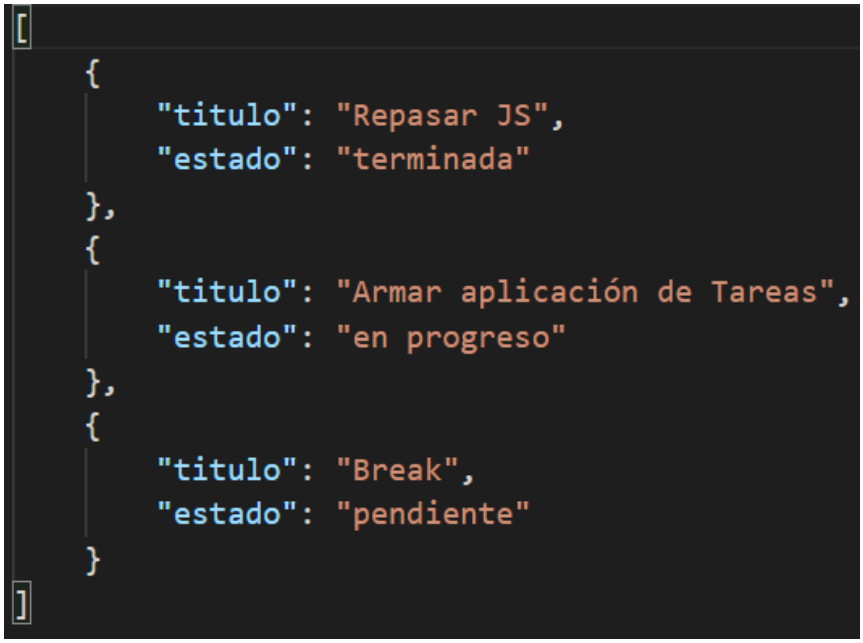
Vamos a armar una app de tareas que nos permita manipular las mismas utilizando las herramientas que hasta el momento hemos aprendido.

Micro desafío 1

El tech leader de nuestro equipo de desarrollo propone programar una app de tareas. Para esto nos encarga:

Crear una carpeta llamada app-tareas y dentro de ella crear un archivo tareas.json, el cual, debe contener un array de objetos literales.Cada objeto literal deberá tener las propiedades: titulo y estado. En donde el título podrá ser cualquier cadena de texto, sin embargo el estado sólo podrá tomar los siguientes valores: “terminada”, “en progreso” o “pendiente”.

Para que tenga una idea de lo que le solicita, aquí le dejamos un ejemplo:



```
[
    {
        "titulo": "Repasar JS",
        "estado": "terminada"
    },
    {
        "titulo": "Armar aplicación de Tareas",
        "estado": "en progreso"
    },
    {
        "titulo": "Break",
        "estado": "pendiente"
    }
]
```

1. Generar un archivo app.js que "consuma" el archivo de tareas.json. Para esto, seguramente nos convenga usar el módulo nativo de NodeJs. File System - FS
2. Mostrar el listado de tareas existente por la terminal. Para esto, seguramente tengamos que guardar el contenido del archivo tareas.json en una variable y convertir la misma a un dato tipo array. ¿Se te ocurre cómo? Aquí te dejamos un enlace donde podrás profundizar sobre el recurso a utilizar:

Micro desafío 2

Ahora se pone algo "picante" el tema. Pues nuestro tech leader nos solicita nuevas funcionalidades, las mismas se detallan a continuación:

1. Permitir que al momento de ejecutar el archivo app.js desde la terminal con Node.js se pueda pasar un argumento después del nombre del archivo de la siguiente manera:
    a. node app.js listar

       Si se escribe la palabra “listar” después del nombre del archivo, se deberán listar todas las tareas existentes en el archivo tareas.json.

    b. node app.js

       Si NO se escribe ninguna palabra después del nombre del archivo, en la terminal deberá aparecer el texto: Atención - Tienes que pasar una acción.

    c. node app.js cualquier texto
       Si se llegase a pasar cualquier otro texto que no sea la palabra listar, en la terminal deberá aparecer el texto: No entiendo qué quieres hacer.

Nuestro tech leader es una buena persona, y para que logremos lo anterior nos dejó la siguiente pista: *piensen en el* ***switch***, adicionalmente, para que puedan tener mucho más clara la idea de cómo poder pasarle algún valor desde la terminal al programa que se está ejecutando desde NodeJs, es fundamental tener en cuenta lo siguiente:

Node Js, nos ofrece un objeto, llamado: **process**, él mismo es un objeto global y puede ser accedido desde cualquier parte. En este enlace puedes ubicar más información: Como puede observar process, posee una cantidad de métodos y propiedades. Uno de ellos es **process.argv**, el cual es un array que contiene los argumentos pasados al programa por la línea de comandos.

El primer elemento será 'node', el segundo elemento será el nombre del archivo JavaScript a ejecutar, Los siguientes elementos serán argumentos adicionales de la línea de comandos. Por lo tanto para lograr efectuar los puntos: a, b y c, debe tener presente el uso de process.argv[ ], para capturar lo que se envía desde la terminal.

Ejemplo: node app.js listar

En nuestro archivo app.js podemos capturar la palabra listar, de la siguiente manera:

```
let accion = process.argv[2];
```

Ya que la palabra node en el array ocupará la posición [0], app.js, ocupará la posición [1] y el argumento listar, ocupa la posición [2].

Y finalmente como nos indico el tech leader al principio con el uso del *switch* podemos evaluar múltiples acciones de acuerdo a lo que recibimos desde la terminal.

```
switch (accion) {
        case 'listar':
                // Aquí debe programar el listado de las tareas
                break;
        default:
                break;
}
```

2. Después de haber logrado lo anterior, nuestro tech leader nos pide modularizar la aplicación, llevando toda la funcionalidad de lectura de tareas a un archivo llamado funcionesDeTareas.js, el cual deberá ser consumido desde el archivo app.js y se espera que todo siga funcionando sin problemas.

No te angusties: Aquí te dejamos una serie de ideas que debes tener presente para lograr este desafío:

- Debes crear el archivo funcionesDeTareas.js
- Ten presente que este es un módulo creado por ti el cual debes luego exportar, para poder ser importando desde el archivo app.js

- Aquí tienes una idea de lo que debe llevar el archivo

```
const fs = require('fs');
let archivoTareas = {
 archivo : 'tareas.json',
  leerArchivo: function(){
     let tareas = fs.readFileSync('tareas.json', 'utf-8');
     return  JSON.parse(tareas);
  }
}
module.exports = archivoTareas;
```

- No se te olvide que debes importar este módulo en tu archivo app.js

```
const archivo = require('./funcionesDeTareas');
```

- Y finalmente cuando requieras usar el archivo de tareas.json, en app.js solo debes hacer:

```
let arrayTareas = archivo.leerArchivo();
```

Al ejecutar la aplicación, desde la terminal de Visual Studio Code, deberías tener el siguiente resultado:

```
$ node app.js listar
Listado de tareas
------------------
1. Repasar JS - terminada
2. Armar aplicación de Tareas - en progreso
3. Break - pendiente
```

```
$ node app.js

Atención - Tienes que pasarme una acción
Las acciones disponibles son: listar
---------------------------------------
```

```
$ node app.js crear
----------------------------------
No entiendo qué quieres hacer
Las acciones disponibles son: listar
----------------------------------
```

Si llegamos hasta aquí, el tech leader del equipo debe estar extremadamente alegre con nuestro trabajo y desempeño. Buen trabajo, Felicitaciones !

Si nos trabamos en alguna parte, a no preocuparnos, es totalmente natural. Son muchos conceptos nuevos y necesitan de un tiempo prudente para asentarse. No olvidemos compartir las partes que no salieron con nuestros profesores y compañeros para despejar dudas.

¡Hasta la próxima!