

# APP de Notas

## Objetivo

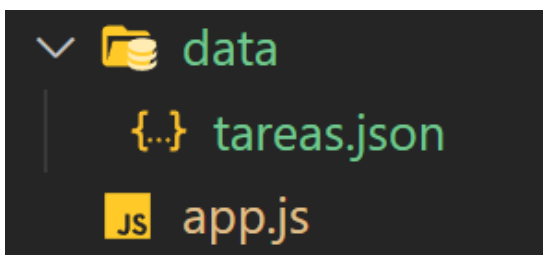
Finalizar la app de tareas utilizando las herramientas que aprendimos en toda la unidad de JS y nodeJS.

## Desafío 1

Crear una carpeta llamada **app-tareas** y dentro de ella crear una carpeta **data** y un archivo **tareas.json** en ella, el cual, debe contener un array de objetos literales.

Cada objeto literal deberá tener las propiedades: **título 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 tengas una idea de lo que te solicitamos te dejamos un ejemplo de carpetas y uno del archivo json.



```
data > {} tareas.json > {} 2 > estado
1  [
2    {
3      "titulo": "Repasar JS",
4      "estado": "terminada"
5    },
6    {
7      "titulo": "Armar aplicación de tareas",
8      "estado": "en progreso"
9    },
10   {
11     "titulo": "Ver inicio del mundial",
12     "estado": "pendiente"
13   }
14 ]
```

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:

## Desafío 2

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**”

Para que logremos lo anterior les dejamos la siguiente pista:

Piensen en el **switch** o **if else**, 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 del código. [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.

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

```
const 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].

*Para este desafío número 2 puedes usar todo lo que fuiste aprendiendo y de la manera que se te ocurra.*

*Te dejamos el resultado final de lo que podemos ver en la terminal.*

Ejemplo de A:

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

Ejemplo de B:

```
$ node app.js

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

Ejemplo de C:

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