

Taller módulos Built-in Node

Haciendo uso de la librería de Express y de los módulos *built-in* de *Node*, construir el siguiente proyecto:

Crear un proyecto que contenga un archivo en formato JSON con un arreglo de elementos (Cada estudiante debe tener un recurso diferente que lo debe reportar en el siguiente Excel: <https://docs.google.com/spreadsheets/d/1qs079z6XFPx0DO3879RCxHYD8Si3m5cpbyQyM7dJF2E/edit#gid=0>), a continuación se muestra el ejemplo de los *endpoints* y el contenido usando la información de estudiantes:

```
{
  "id": 4,
  "code": "87653",
  "name": "Pepito",
  "lastname": "Perez",
  "age": 26,
  "current_university_degree_program": "Ingeniería Informática"
}
```

Esta aplicación debe tener como mínimo cinco rutas que se listan a continuación (Este es el ejemplo con estudiantes, cada uno debe hacerlo con la temática elegida):

- ***/students (GET)***: Ruta para entregar todo el listado de estudiantes almacenados en el archivo JSON.
- ***/students/:studentId (GET)***: Ruta para entregar la información del estudiante con el ID recibido por parámetro.
- ***/students (POST)***: Ruta para agregar un nuevo estudiante al archivo JSON.
- ***/students/:studentId (PUT/PATCH)***: Ruta para actualizar la información de un estudiante (El que tenga el ID de la ruta).
- ***/students/:studentId (DELETE)***: Ruta para eliminar un estudiante del archivo JSON.

Condiciones generales del proyecto

- Crear un nuevo proyecto desde cero y a través del comando *npm init* crear el archivo *package.json* con información válida.
- Instalar las librerías requeridas (Como mínimo *express*)
- Se deben crear las cinco rutas especificadas anteriormente con el tipo de petición HTTP correcto.
- El recurso elegido (En mi caso estudiantes) debe tener al menos ocho claves con al menos 4 tipos de datos diferentes.

- Se debe usar el módulo *built-in* de *FileSystem* para leer/escribir en el archivo que almacenará los datos.
- Se debe crear un módulo aparte que realice las operaciones de lectura y escritura del archivo, este debe ser requerido en el módulo principal que contiene el servidor.
- Se debe validar que la información que se recibe sea válida y se encuentre en el formato correcto, recomendando la lectura de mi [post en médium](#) y la implementación de la librería *Joi*. Se deben tener al menos dos validaciones por cada uno de los ocho campos.
- Adjuntar la colección de *Postman* para probar el funcionamiento de la aplicación.
- Se debe subir el código realizado a la plataforma *Github* y compartir el enlace del repositorio.