



Escuela:

Facultad de Informática

Carrera:

Licenciatura en informática

Hecho por:

Gastelum Landeros Antonio

Grupo: 2-1

Código Explicado:

Importación de módulos y configuración inicial

```
const express = require('express');
```

```
const fs = require('fs');
```

```
const app = express();
```

```
const PORT = 3000;
```

¿Qué hace?

- `express`: importa el framework Express, que simplifica la creación de servidores web.
- `fs`: permite trabajar con archivos (leer, escribir, etc.).
- `app`: instancia de Express.
- `PORT`: el número del puerto donde se ejecutará la API (<http://localhost:3000>).

Middleware para leer JSON

```
app.use(express.json());
```

¿Qué hace?

Permite que el servidor entienda y procese el cuerpo de las peticiones POST en formato JSON. Sin esto, no podrías leer `req.body`.

Ruta POST /alumno

```
app.post('/alumno', (req, res) => {
```

```
  const { cuenta, nombre, promedio, grado, grupo } = req.body;
```

```
  if (!cuenta || !nombre || promedio == null || !grado || !grupo) {
```

```
    return res.status(400).json({ error: 'Faltan datos del alumno.' });
```

```
  }
```

¿Qué hace?

- Define la ruta `/alumno`, que solo acepta peticiones POST.

- Lee los campos del cuerpo JSON enviado por el cliente.
- Si falta alguno de los campos, responde con un código 400 (Bad Request) y un mensaje de error.

Preparar el nuevo alumno

```
const nuevoAlumno = { cuenta, nombre, promedio, grado, grupo };
```

¿Qué hace?

Crea un objeto alumno con los datos recibidos para guardarlo en el archivo.

Leer el archivo existente o empezar uno nuevo

```
let alumnos = [];  
  
const archivo = 'alumnos.json';  
  
if (fs.existsSync(archivo)) {  
  
  const contenido = fs.readFileSync(archivo, 'utf-8');  
  
  try {  
  
    alumnos = JSON.parse(contenido);  
  
  } catch (e) {  
  
    alumnos = [];  
  
  }  
  
}
```

¿Qué hace?

- Si ya existe el archivo alumnos.json, lo lee y convierte en un array de objetos con JSON.parse().
- Si el archivo no existe o tiene un formato incorrecto, se crea un arreglo vacío.

Agregar el nuevo alumno y guardar

```
alumnos.push(nuevoAlumno);
```

```
fs.writeFileSync(archivo, JSON.stringify(alumnos, null, 2), 'utf-8');
```

¿Qué hace?

- Agrega el nuevo alumno al arreglo.
- Guarda el arreglo actualizado en el archivo alumnos.json en formato legible (con sangría de 2 espacios).

Enviar respuesta al cliente

```
res.status(201).json({ mensaje: 'Alumno guardado exitosamente.', alumno: nuevoAlumno });  
});
```

¿Qué hace?

- Responde al cliente con un código 201 (Created) y un mensaje confirmando que el alumno fue guardado.

Iniciar el servidor

```
app.listen(PORT, () => {  
  
  console.log(`Servidor escuchando en http://localhost:${PORT}`);  
  
});
```

¿Qué hace?

- Inicia el servidor Express en el puerto 3000.
- Muestra un mensaje en la consola para confirmar que está funcionando.