



Universidad

Universidad Autónoma de Sinaloa

Carrera

Lic. en Informática

Materia

Desarrollo web del lado del servidor

Actividad

Crear API con método POST

Grupo

2-3

Fecha

08/06/2025 Culiacán, Sinaloa

Maestro

José Manuel Cazarez Alderete

Alumna

Núñez Sarabia Jessica Anahí



ÍNDICE

03

JUSTIFICACIÓN

05

EVIDENCIA

JUSTIFICACIÓN

La API tiene un endpoint `/alumno` que recibe un objeto con los datos del alumno, verifica que se hayan enviado correctamente, los guarda en un archivo JSON y responde al cliente con un mensaje de éxito o error

Se importan los módulos necesarios:

- `express`: Para manejar solicitudes HTTP.
- `fs`: Para manipular archivos.

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

Se define el puerto en el que se ejecutará la API y se inicializa Express.

```
const app = express();  
const PORT = 3000;
```

Este middleware permite que el servidor interprete solicitudes con datos en formato JSON.

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

Se crea el método POST, que recibe los datos del alumno y verifica que estén completos.

```
app.post('/alumno', (req, res) => {  
  const { cuenta, nombre, promedio, grado, grupo } = req.body;
```

Si falta algún campo, se devuelve un error 400 Bad Request.

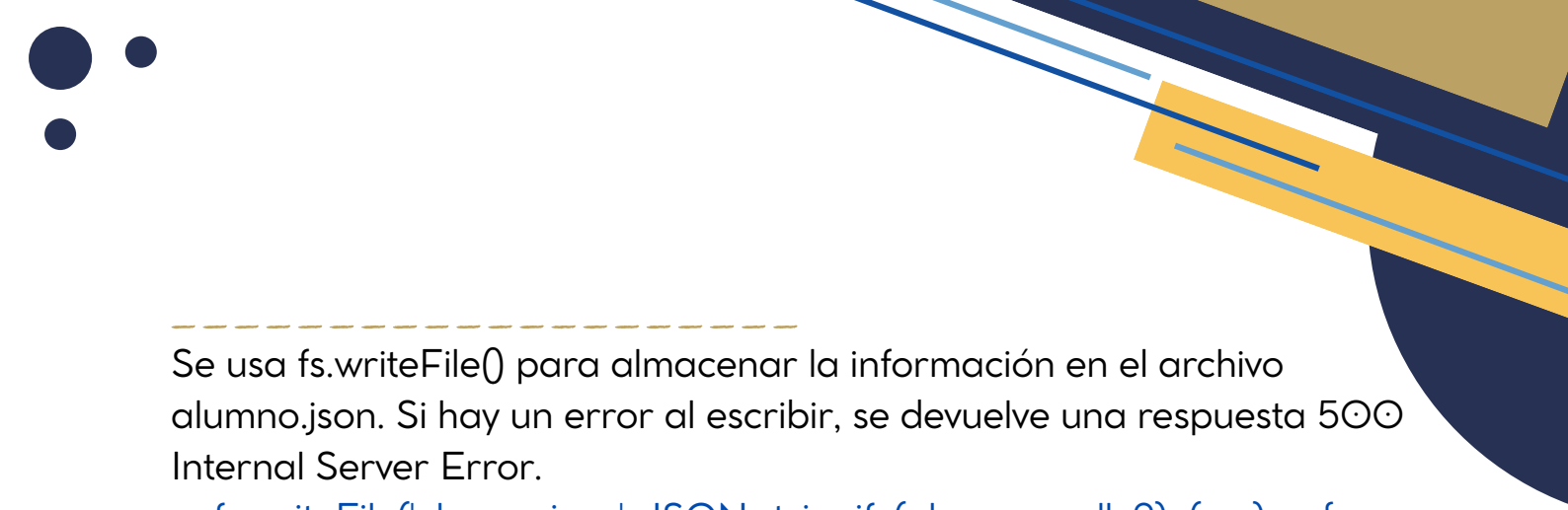
```
  if (!cuenta || !nombre || !promedio || !grado || !grupo) {  
    return res.status(400).json({ error: 'Todos los campos son obligatorios' });  
  }
```

Luego, se crea un objeto alumno con los datos recibidos.

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

Se usa `fs.writeFile()` para almacenar la información en el archivo `alumno.json`. Si hay un error al escribir, se devuelve una respuesta 500 Internal Server Error.

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



Se usa `fs.writeFile()` para almacenar la información en el archivo `alumno.json`. Si hay un error al escribir, se devuelve una respuesta 500 Internal Server Error.

```
fs.writeFile('alumno.json', JSON.stringify(alumno, null, 2), (err) => {  
  if (err) {  
    console.error('Error al guardar el archivo:', err);  
    return res.status(500).json({ error: 'No se pudo guardar la  
información' });  
  }  
  res.status(200).json({ message: 'Información guardada  
correctamente', alumno });  
});
```

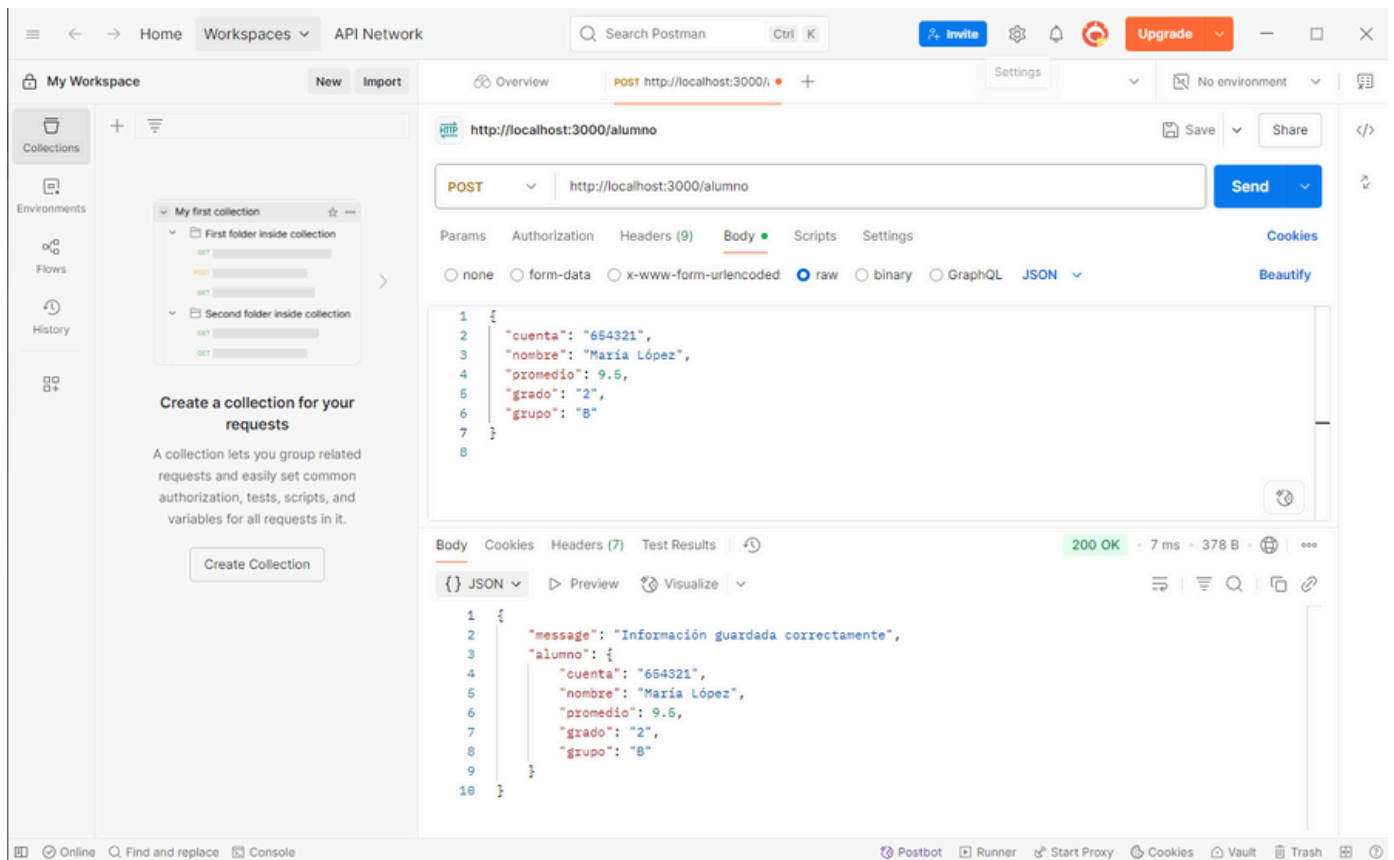
El servidor comienza a escuchar en el puerto 3000 y muestra un mensaje de confirmación.

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



EVIDENCIA

A continuación, se presenta la evidencia de la correcta ejecución de la API mediante una solicitud POST en Postman, donde los datos del alumno fueron procesados y almacenados exitosamente en el archivo JSON.



Esta imagen demuestra que la API responde adecuadamente, validando los datos y enviando una confirmación tras guardarlos en el sistema.