

Explicación del código. Tarea: Crear API con método Post.

Explicación:

En el frontend, se lanza un contenedor llamado "express" utilizando una "intención" y un "índice persistente" que permite que los recursos de la aplicación manejen solicitudes HTTP. En la segunda línea, el esquema fs se transfiere desde el sistema de archivos y se almacena en la constante fs, que puede leer y escribir archivos. En la tercera línea, se crea una instancia llamando a `express()` y se compila en un programa normal para configurar las rutas y los medios. En la cuarta línea, el puerto implícito se establece en 3000, lo que indica el puerto en el que escuchar las conexiones entrantes.

En la sexta línea, se agrega un método global a `app.use(express.json())` que convierte automáticamente el cuerpo de la solicitud JSON en objetos JavaScript, exponiéndolos como `req.body`. En la octava línea, la ruta POST para la URL `/estudiante` se declara usando `app.post`, una función de controlador que acepta los parámetros `req` y `res`. La novena línea contiene los campos "informe", "nombre", "GPA", "clase" y "categoría" de `req.body` usando "eliminar" para manipularlos directamente.

La línea once comienza con la validación: verifica que todos los espacios estén presentes y que el GPA no sea anormal. Si hay alguno, se presenta la aplicación y se envía una respuesta HTTP 400 al cliente con un objeto JSON que le informa que todos los espacios son obligatorios. En la línea dieciséis, se crea un nuevo objeto estudiante a partir de los datos.

Los estudiantes se presentan por primera vez en la línea 18. En la línea diecinueve, `fs.existsSync` comprueba si existe el archivo `student.json`. Si es así, el contenido se leerá como texto utilizando. En la línea veintiuno, JSON se convierte en un objeto JavaScript utilizando `JSON.parse`, y los resultados se pasan a los estudiantes para preservar las expresiones anteriores.

En la línea veinticuatro, se agrega `nuevoEstudiante` al final de la secuencia mediante `push`. En la línea veintisiete, el cliente recibe una respuesta `res.status(201)` que indica una creación exitosa y se envía un objeto JSON para confirmar el guardado y los nuevos datos del estudiante. En la trigésima línea, se llama a `app.listen` y se llama al puerto con una devolución de llamada; Esto inicia el esclavo y lo configura para escuchar solicitudes. En la línea treinta y uno, se utiliza la llamada `console.log`, imprimiendo un mensaje a la consola, mostrando la URL del servidor, sin embargo, al utilizar comillas simples en lugar de comillas, el argumento ``${port}`` no se imprime correctamente, pero la intención es mostrarlo. Distribución.

