

Esto importa dos decoradores de NestJS que se utilizarán para definir controladores y rutas HTTP.

Mapeamos la ruta en el controlador

```
src > TS app.controller.ts > AppController > getHello
  Click here to ask Blackbox to help you code faster
1  import { Controller, Get } from '@nestjs/common';
2  import { AppService } from './app.service';
3
4  @Controller('api/v1/bm')
5  export class AppController {
6    constructor(private readonly appService: AppService) {}
7
8    @Get()
9    getHello(): string {
10     return this.appService.getHello();
11   }
12 }
13
```

Se define un método llamado gethello que devuelve un tipo string, retorna lo que contenga appservice en getHello.

Importa el módulo fs de Node.js, que proporciona funciones para trabajar con el sistema de archivos del sistema operativo

Importa el módulo path de Node.js, que proporciona funciones para trabajar con rutas de archivos y directorios.

```
import { Injectable } from '@nestjs/common';
import * as fs from 'fs';
import * as path from 'path';

@Injectable()
export class AppService {
  getHello(): string {
    // Obtiene la ruta del directorio del proyecto
    const projectDir = path.resolve(__dirname, '..');

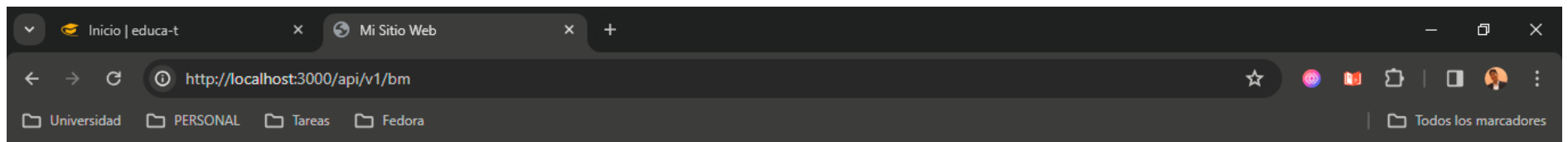
    // Lee el archivo index.html y devuelve su contenido
    const content = fs.readFileSync(path.join(projectDir, 'src/index.html'), 'utf8');
    return content;
  }
}
```

Obtiene la ruta del directorio del proyecto utilizando el método resolve() del módulo path. \_\_dirname es una constante global de Node.js que representa el directorio en el que se encuentra el archivo actual. Se usa .. para navegar al directorio padre.

Lee el contenido del archivo index.html ubicado en el directorio src del proyecto. Utiliza el método readFileSync() del módulo fs para leer el archivo de forma síncrona y path.join() para construir la ruta absoluta al archivo.

Retorna lo que contenga contenido y en ese tiene el html.

Salida



Hola Mundo

