**PRÁCTICA ENTREGABLE SERVIDOR BÁSICO**

**Resumen:**

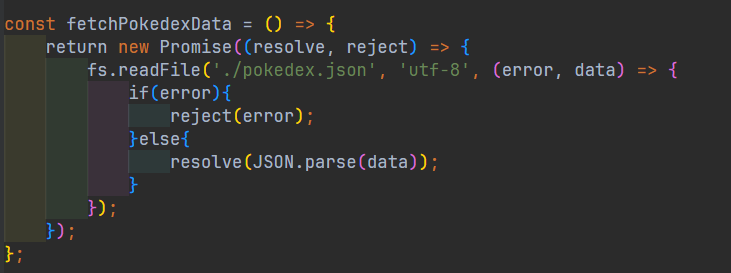
Para poder crear el servidor y leer el json con el que se desarrolla el ejercicio he importado dos módulos de node, el módulo http y el fs.



Luego he creado una función llamada ‘fetchPokedexData’ que mediante una promesa y el módulo ‘fs’ se encarga de leer mi archivo ‘pokedex.json’ gracias a la función .readFile().

Además he especificado el encoding que necesito, en mi caso el utf-8 que es el que me va a permitir leer posteriormente el id o el nombre que le pase en los diferentes idiomas como inglés, japonés, chino y francés.

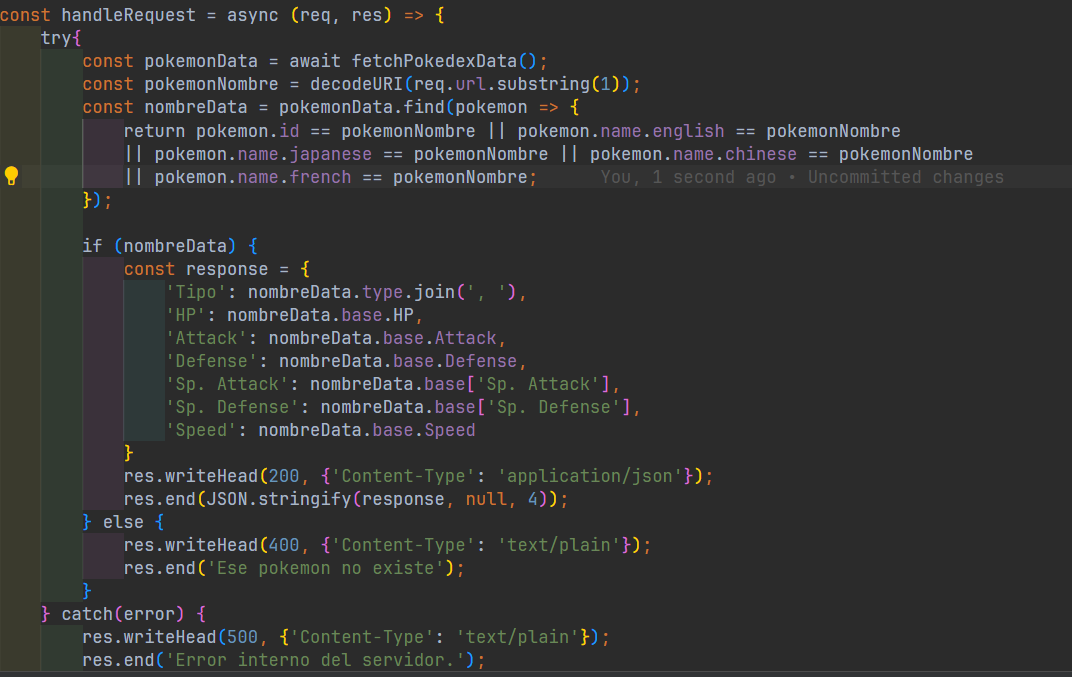
Después he puesto un if para que en caso de error pueda manejarlo, y un else con un resolve en el que se parsean los datos del json si no da ningún error.



Después del método anterior he creado otro método llamado ‘handleRequest’ que va a realizar la llamada al método anterior para traerse los datos del json con el formato que aparece definido en el enunciado. Además he envuelto el código con un try and catch para que en caso de recibir un error pueda manejarlo.

Para ello he definido tres variables, una llamada ‘pokemonData’ que llame al método anterior, otra ‘pokemonNombre’ que va a guardar la última parte de la URL donde se especifica el pokemon del que queremos sacar la información y un ‘nombreData’ que utilizará la variable anterior para hacer un .find y que compruebe si existen datos o no para guardarlos.

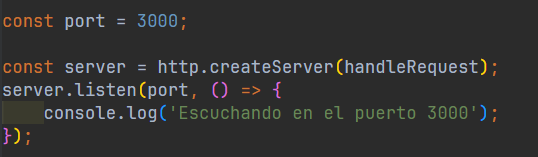
Después en el if, en caso de que sí existan datos sobre ese id o nombre, declaro un response con el formato de salida del enunciado, y lo mando a un JSON. En caso de que no encuentra datos habrá una salida con el mensaje ‘Ese pokemon no existe’.



Por último, he creado una variable definiendo el puerto que voy a utilizar para que escuche el servidor, en mi caso el 3000.

Y he creado un server con el http y la función .createServer() que recibe por parámetro la función de ‘handleRequest’.

Después he hecho un server.listen() con el puerto definido para que el server se ponga a escuchar y nos avise de que está en marcha.



**Conclusiones y observaciones:**

Como conclusión, el ejercicio me ha parecido muy entretenido, ya que, era la primera vez que creaba un server y llamaba a una API con Node.

No le he encontrado mucha dificultad al final, aunque al principio cuando lo leí pensaba que la parte de los idiomas y el que se pudiera llamar mediante el ‘id’ y el ‘nombre’ me iba a costar más.

En cuanto al código que he desarrollado, he seguido la estructura del ejercicio que hicimos en clase y me ha servido de mucha ayuda a la hora de mantener el código limpio y visual.

Aunque pienso que la parte en la que hago el .find para encontrar el pokemon o el id se podría haber hecho de una forma más limpia, en vez de utilizar tantos OR por los diferentes idiomas. El problema es que no se me ocurría, así que lo dejé así porque me funcionaba, también intenté en un principio el hacer una llamada al archivo raw que hay en el github con módulos como ‘request’ pero se me complicó bastante porque nunca lo había utilizado. Y como finalmente no me funcionaba decidí descargarlo directamente y hacerlo como lo habíamos visto :) .

**Enlace al repositorio de Github:** https://github.com/moreno025/NodeJS-UE/tree/main/Practicas\_entregables/Practica\_servidor\_basico