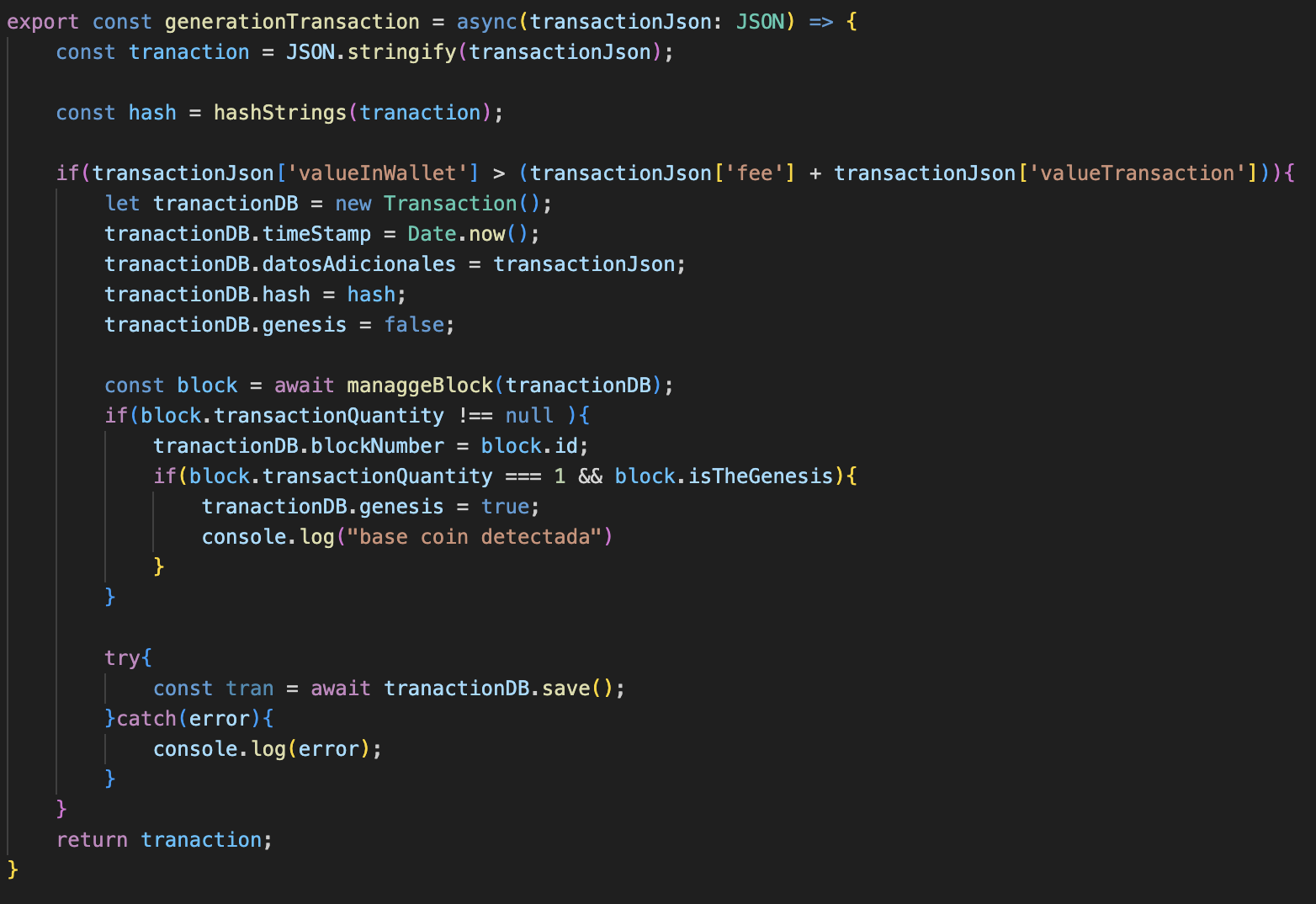
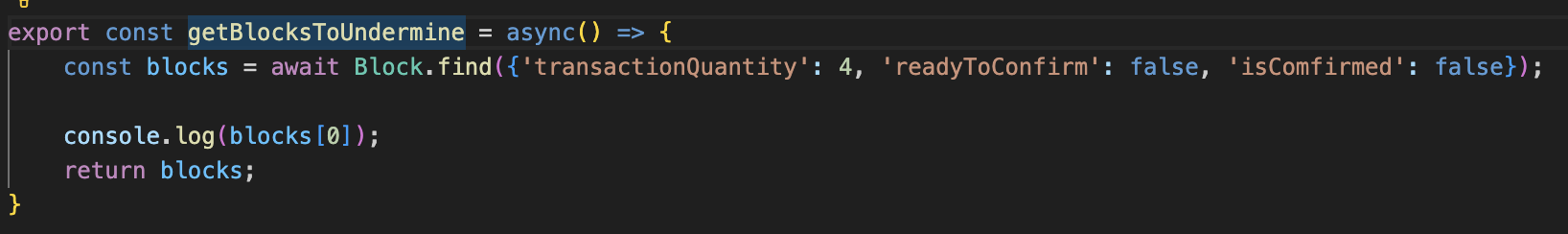
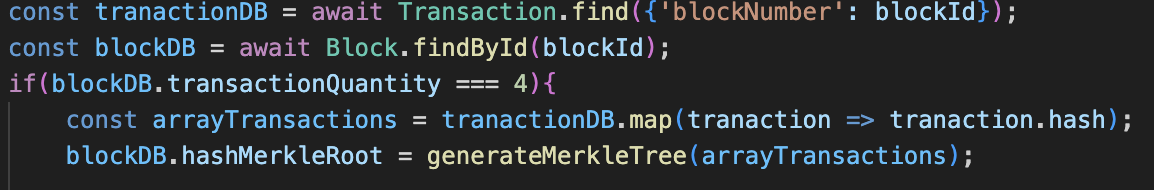
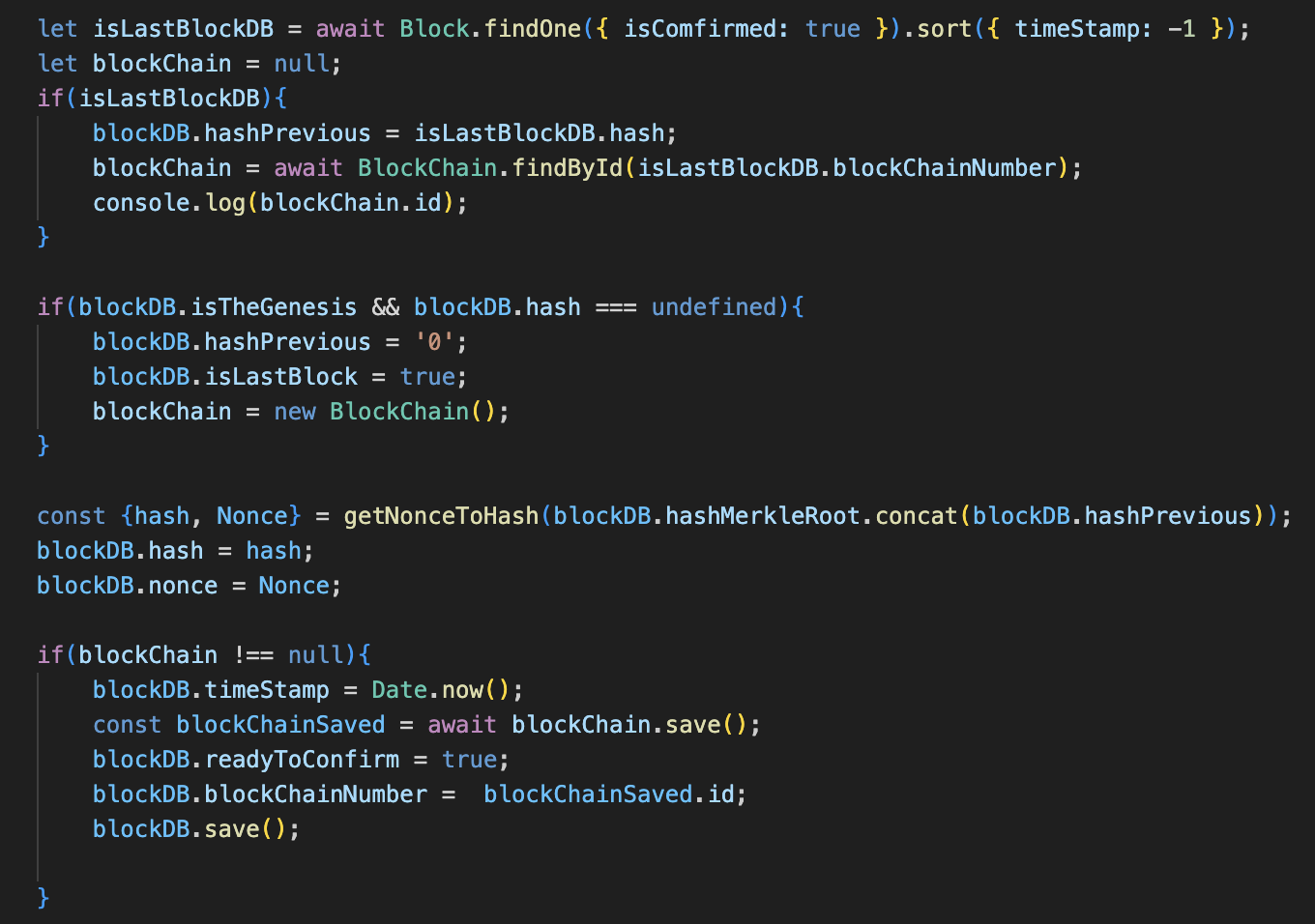
BLOCKCHAIN  
Para este desarrollo implementé 5 endpoints. Aquí explicaré que hice en cada uno:

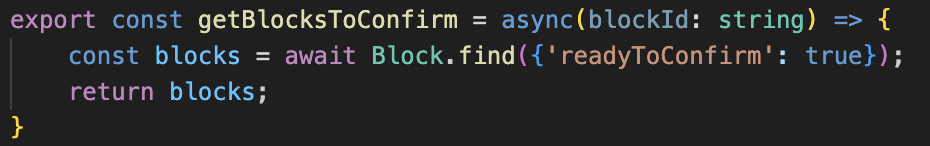
1. SendTransaction: En este endpoint, recibo la orden enviada desde Postman, hallo el hash y asocio a un bloque. En este desarrollo el número máximo de transacciones por bloque es de 4. Inicialmente valido que la transacción tenga valores coherentes:  
   

A la hora de crear el bloque se valida si el primer bloque y que tenga las transacciones necesarias.

1. GetBlocks: En este endpoint se traen todos los bloques que pueden ser minados, para esto se tiene en cuenta tres atributos: Que esté listo para minar, que no esté confirmado y que tenga asignado 4 transacciones:  
   
2. UndermineBlocks: En este endpoint comienzo obteniendo las transacciones y sus respectivos hashes para encontrar el “root hash”. Para hallar este uso una función recursiva, la cual en cada iteración va a recibir cada vez menos valores en el array y retorna el hash más arriba, el cual es el “root hash”:  
   

Ahora, valida si es un bloque Genesis o si solo es uno más que puede pertenecer a la cadena. Inicialmente se crea la cadena si es el Genesis y se le asocia al bloque el id de la cadena, pero a la cadena aún no porque no ha sido confirmado por otro nodo.  


1. GetReadyToConfirmBlocka: Este endpoint va a devolver los bloques que ya fueron minados, para que otro nodo pueda confirmarlo o no.



1. ConfirmBlock: En este endpoint se repiten ciertas validaciones, como el que la transacciones tenga valores validos. Se halla el root hash, el hash previo, el hash del bloque que se está validando y el Nonce, para al final comparar si el hash y el Nonce del bloque a confirmar y el hash y Nonce recientemente dado, si coinciden, si es así, se confirma bloque y se agrega a la cadena, esto se hace actualizando los valor lastHash y previousHash del la blockchain en la base de datos:

