**Clase Blockchain**

Constructor

* **constructor()**: Este método inicializa una nueva instancia de la cadena de bloques. Crea un array vacío llamado **chain** para almacenar los bloques y llama al método **addBlock** para agregar el bloque génesis a la cadena.

Método para Crear el Bloque Génesis

* **createGenesisBlock()**: Este método devuelve un nuevo bloque que representa el bloque génesis de la cadena. El bloque génesis tiene un índice de 0, un hash anterior de "0" y transacciones iniciales predefinidas.

Método para Agregar un Nuevo Bloque

* **addBlock(newBlock)**: Este método agrega un nuevo bloque a la cadena de bloques. Primero, establece el hash anterior del nuevo bloque al hash del último bloque en la cadena (o "0" si la cadena está vacía). Luego, realiza la minería del nuevo bloque llamando al método **mineBlock** y finalmente agrega el nuevo bloque al array **chain**.

Método para Obtener el Último Bloque

* **getLatestBlock()**: Este método devuelve el último bloque de la cadena de bloques. Verifica si la cadena de bloques tiene al menos un bloque y devuelve el último bloque en el array **chain**.

Método para Verificar la Validez de la Cadena de Bloques

* **isValid()**: Este método verifica la validez de la cadena de bloques iterando sobre todos los bloques en la cadena. Para cada bloque, verifica si el hash calculado del bloque coincide con el hash almacenado, si el hash anterior del bloque coincide con el hash del bloque anterior, y si la raíz de Merkle del bloque coincide con la raíz calculada. Si alguna de estas verificaciones falla, la cadena de bloques se considera inválida.

**Clase Block**

Constructor

* **constructor(index, previousHash, transactions)**: Este método inicializa una nueva instancia de un bloque. Toma como argumentos el índice del bloque en la cadena, el hash del bloque anterior, y un array de transacciones. También inicializa la marca de tiempo, el nonce y calcula la raíz de Merkle y el hash del bloque.

Método para Calcular el Hash del Bloque

* **calculateHash()**: Este método calcula y devuelve el hash del bloque concatenando y hasheando el índice, el hash anterior, la marca de tiempo, el nonce y la raíz de Merkle del bloque (para esto se usa el SHA256 como método de codificación).

Método para Minar el Bloque

* **mineBlock()**: Este método realiza la minería del bloque al encontrar un hash válido que cumpla con ciertos criterios (en este caso, un hash que comience con un número específico de ceros, 4 para ser precisos). Incrementa el nonce y recalcula el hash repetidamente hasta encontrar un hash válido.

Método para Calcular la Raíz de Merkle

* **calculateMerkleRoot()**: Este método calcula y devuelve la raíz de Merkle de las transacciones en el bloque. Utiliza un algoritmo iterativo para construir el árbol de Merkle y calcular la raíz.

**Clase Transaction**

Constructor

* **constructor(sender, recipient, amount)**: Este método inicializa una nueva instancia de una transacción con un remitente, un destinatario y una cantidad.

Método para Calcular el Hash de la Transacción

* **calculateHash()**: Este método calcula y devuelve el hash de la transacción concatenando y hasheando el remitente, el destinatario y la cantidad (para esto se usa el SHA256 como método de codificación).