Blockchain

Es una tecnología de registro distribuido que permite el almacenamiento y la transmisión segura de información de manera transparente y sin la necesidad de una autoridad central, en otras palabras, Blockchain es similar a una base de datos distribuida o libro mayor compartido entre los nodos de una red informática .

Consiste en una cadena de bloques, donde cada bloque contiene una lista de transacciones o registros y está enlazado con el bloque anterior mediante un código criptográfico llamado hash.

Se conocen sobre todo por su papel crucial en los sistemas de criptomoneda para mantener un registro seguro y descentralizado de las transacciones, pero no se limitan a los usos de la criptomoneda. Las cadenas de bloques pueden utilizarse para hacer que los datos de cualquier sector sean inmutables, término utilizado para describir la imposibilidad de alterarlos.

¿Qué es la descentralización?

La descentralización se refiere a la transferencia del control y la toma de decisiones de una entidad centralizada (individuo, organización o grupo de ellos) a una red distribuida.

Cada nodo de la red tiene una copia completa de la cadena de bloques y participa en la verificación y validación de las transacciones.

¿Qué aporta?

Mayor seguridad: Al estar distribuida en múltiples nodos, la información es más resistente a ataques y manipulaciones, ya que requeriría controlar la mayoría de los nodos para alterar la cadena de bloques.

Transparencia y confianza: Al ser accesible y verificable por cualquier participante de la red, la blockchain descentralizada ofrece transparencia en las transacciones y genera confianza al eliminar la necesidad de una entidad central que pueda ser corrupta o poco confiable.

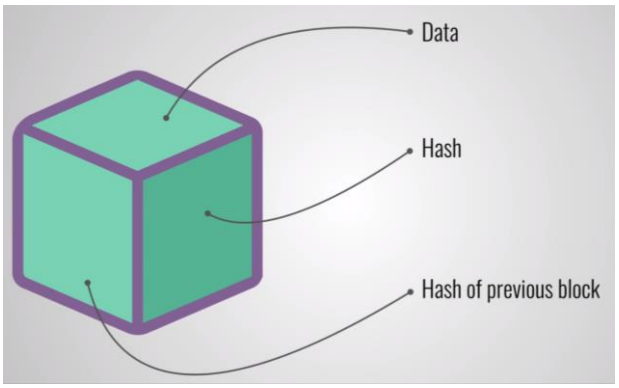
Resistencia a fallos: La descentralización reduce la vulnerabilidad del sistema a fallos técnicos o ataques cibernéticos, ya que si un nodo falla o es atacado, los demás nodos pueden continuar operando y manteniendo la integridad de la red.

Eliminación de intermediarios: Al descentralizar el control, la blockchain puede eliminar o reducir la necesidad de intermediarios o terceros de confianza, lo que puede agilizar y abaratar ciertos procesos.
How does a blockchain work - Simply Explained

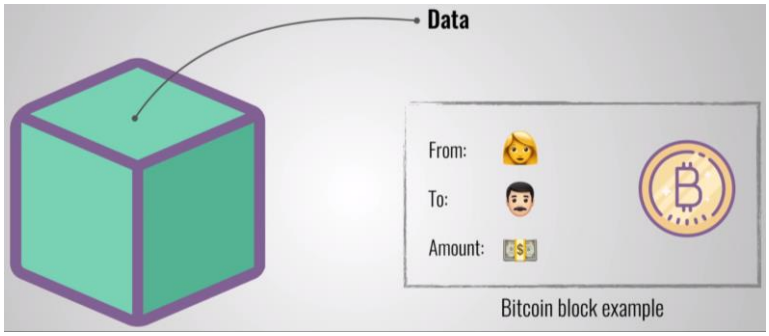
Explicacion del video

¿Cómo funciona eso?

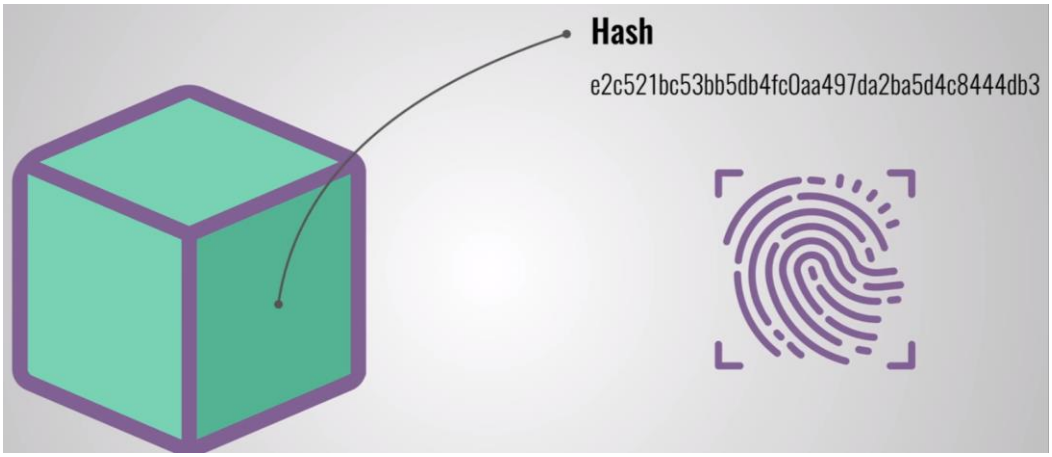
Bueno, demos un vistazo más de cerca a un bloque. Cada bloque contiene algunos datos, el hash del bloque y el hash del bloque anterior. Los datos que se almacenan dentro de un bloque dependen del tipo de blockchain.



La cadena de bloques de Bitcoin, por ejemplo, almacena los detalles sobre una transacción ahí, tal como emisor, receptor y cantidad de monedas.

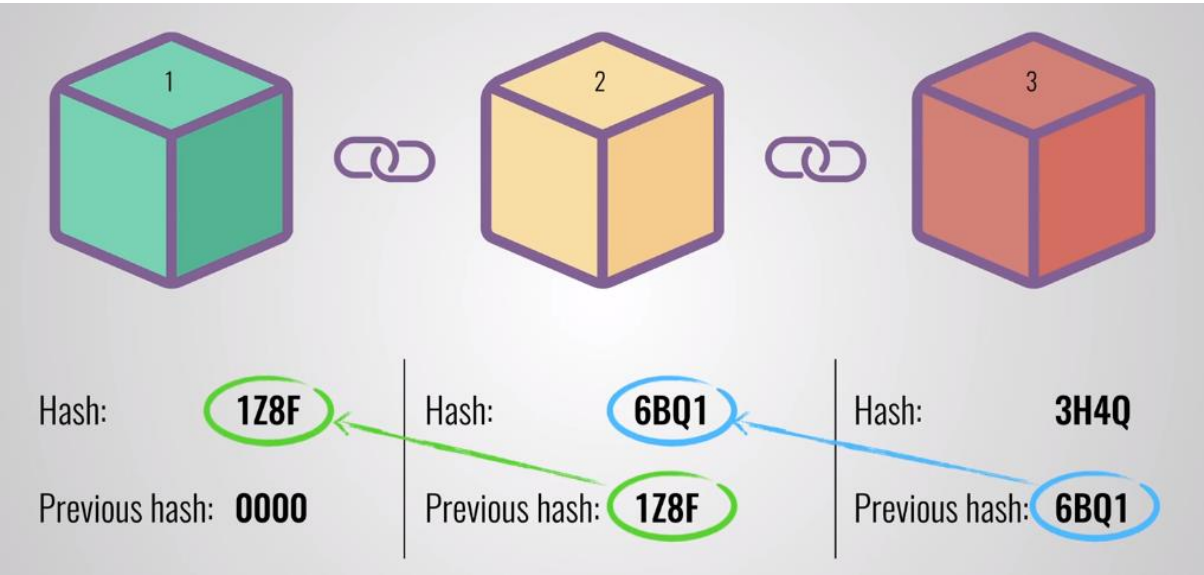


Un bloque también tiene un hash. Puede comparar un hash con una huella dactilar. Identifica un bloque y todos sus contenidos y siempre es único, al igual que una huella dactilar. Una vez que se crea un bloque, su hash es calculado. Cambiar algo dentro del bloque hará que el hash cambie.



En otras palabras: los hashes son muy útiles cuando se desean detectar cambios en los bloques. Si la huella dactilar de un bloque cambia, ya no es el mismo bloque El tercer elemento dentro de cada bloque es el hash del bloque anterior. Esto efectivamente crea una cadena de bloques y es esta técnica la que hace que una cadena de bloques sea tan segura.

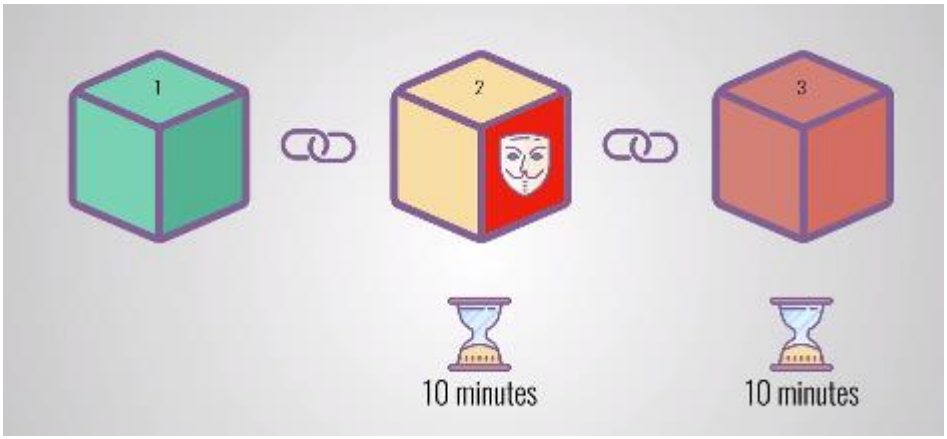
Tomemos un ejemplo. Aquí tenemos una cadena de 3 bloques. Como puedes ver, cada bloque tiene un hash y el hash del bloque anterior. Así que el bloque número 3 apunta al bloque número 2 y el número 2 apunta al número 1. Ahora, el primer bloque es algo especial, no puede señalar bloques anteriores porque es el primero. Llamamos a éste el bloque de génesis.



Ahora digamos que alteras el segundo bloque. Esto hace que el hash del bloque cambie también. A su vez eso hará inválido al bloque 3 y a todos los siguientes bloques porque ya no almacenan un hash válido del bloque anterior. Así que cambiar un solo bloque hará inválidos todos los siguientes bloques. Pero usar hashes no es suficiente para prevenir la alteración. Las computadoras en estos días son muy rápidas y pueden calcular cientos de miles de hashes por segundo.

Podrías alterar efectivamente un bloque y recalcular todos los hashes de los otros bloques para hacer tu cadena de bloques válida de nuevo.

Proof of work (prueba de trabajo), Para mitigar esto, las cadenas de bloques tienen algo llamado prueba de trabajo. Es un mecanismo que ralentiza la creación de nuevos bloques. En el caso de Bitcoins: se tarda unos 10 minutos para calcular la prueba de trabajo requerida y agregar un nuevo bloque a la cadena. Este mecanismo hace que sea muy difícil alterar los bloques, porque si alteras 1 bloque, tendrás que volver a calcular la prueba de trabajo para todos los siguientes bloques.



Entonces, la seguridad de una cadena de bloques proviene del uso creativo del hashing y del mecanismo de la prueba de trabajo. Pero hay una forma más de que las cadenas de bloques se aseguren a sí mismas y eso es por ser distribuidas. En lugar de usar una entidad central para administrar la cadena, los blockchains usan una red peer-to-peer y cualquiera puede unirse.

Cuando alguien se une a esta red, obtiene una copia completa de la cadena de bloques. El nodo puede usar esto para verificar que todo está en orden todavía. Ahora veamos qué sucede cuando alguien crea un nuevo bloque. Ese nuevo bloque se envía a todos en la red. Luego cada nodo verifica el bloque para asegurarse de que no ha sido alterado. Si todo sale bien, cada nodo agrega este bloque a su propia cadena de bloques. Todos los nodos en esta red crean consenso. Acuerdan qué bloques son válidos y cuáles no lo son. Los bloques alterados serán rechazados por otros nodos en la red. Así que para alterar con éxito una cadena de bloques tendrás que manipular todos los bloques en la cadena, rehacer la prueba de trabajo para cada bloque.

Smarts contracts:

La tecnología de blockchain también está en constante evolución. Uno de los desarrollos más recientes es la creación de **contratos inteligentes**. Estos contratos son programas simples que son almacenados en el blockchain y pueden ser utilizados para intercambiar automáticamente monedas basados en ciertas condiciones Más sobre contratos inteligentes en un video posterior. La creación de la tecnología blockchain ha maximizado el interés de muchas personas Pronto, otros se dieron cuenta de que esta tecnología puede ser utilizada para otras cosas como el almacenamiento de registros médicos, crear un notario digital e incluso recolectar impuestos.

Un contrato inteligente es un acuerdo o contrato autoejecutable que se codifica como un algoritmo y se almacena en una cadena de bloques (blockchain). Es un programa que ejecuta automáticamente los términos del acuerdo una vez que se cumplen ciertas condiciones predefinidas. Los contratos inteligentes eliminan la necesidad de intermediarios o terceros en las transacciones y permiten interacciones confiables, seguras y transparentes entre las partes.

La ejecución de un contrato inteligente se basa en las entradas que recibe y las condiciones predefinidas codificadas en él. Estas condiciones pueden basarse en varios factores, como el paso del tiempo, la ocurrencia de eventos específicos o la verificación de ciertos datos. Cuando se cumplen las condiciones, el contrato inteligente activa automáticamente las acciones especificadas o transfiere activos de acuerdo con los términos acordados.

Ejemplo

Home | Calculadora.sol | Holamundo.sol

```
1  pragma solidity ^0.8.3;
2
3  contract Holamundo {
4
5      string private _mensaje;
6
7      constructor(string memory mensaje) {
8          _mensaje = mensaje;
9      }
10
11     function getmensaje() public view returns (string memory) {
12         return _mensaje;
13     }
14
15     function setmensaje(string memory newmensaje) public {
16         _mensaje = newmensaje;
17     }
18 }
19
```


Home
Calculadora.sol
// SPDX-License-Identifier: MIT
// Importar la biblioteca Solidity
pragma solidity ^0.8.0;
// Definir el contrato
contract Calculadora {
// Definir una función para sumar dos números enteros
function sumar(uint a, uint b) public pure returns (uint) {
return a + b;
}
// Definir una función para restar dos números enteros
function restar(uint a, uint b) public pure returns (uint) {
return a - b;
}
// Definir una función para multiplicar dos números enteros
function multiplicar(uint a, uint b) public pure returns (uint) {
return a * b;
}
// Definir una función para dividir dos números enteros
function dividir(uint a, uint b) public pure returns (uint) {
require(b != 0, "No se puede dividir por cero");
return a / b;
}
}