Ethereum

Una nueva visión de la cadena de bloques

Ana Peña
Pedro Bonilla

27 de noviembre de 2017

Universidad de Granada

Contenidos.

1. Introducción.
2. Ethereum.
3. Aplicaciones.
4. Particularidades.
5. Conclusiones

Introducción.

Objetivo.

Tras el boom de las cadena de bloques, las criptodivisas simples quedaban cortas y, con el objetivo de dar más funcionalidad a esta tecnología, nació ethereum.

Historia.

- 2014 • WhitePaper de Vitalik Buterin.
- 2014 • Crowdfounding.
- 2015 • Lanzamiento de la plataforma.
- 2016 • Desastre de la DAO.
- 2016 • Nace Ethereum classic.
- 2016 • Dos hard-fork más.
- 2017 • Refuerzo de la seguridad.
- 2017 • Prohibición de las ICO en China.
- 2017 • Exposición de Fundamentos de Redes

Ethereum.

Una vez vemos el objetivo que tiene este sistema, vamos a profundizar en detalles de su funcionamiento,

Bitcoin como sistema de transición de estados.

Podemos ver el la cadena como un sistema de transición de estados. Esto Seá importante para ver las posibilidades de esta entidad.

- Los estados serían el conjunto de la información escrita en cada bloque.
- Tendríamos una función de transicion:
  $APPLY(S, TX) \rightarrow S'$ o $ERROR$

Bitcoin como sistema de transición de estados.

Ejemplifiquemos el uso de este sistema.

- $APPLY(\{Ana : 50, Pedro : 50\}, \text{"enviar 20 de Ana a Pedro"})$
  Ana: 30 , Pedro: 70
- $APPLY(Ana : 50, Pedro : 50, \text{"enviar 70 de Ana a Pedro"})$
  ERROR

Cadena de bloques.

Una cadena de bloques es una base de datos distribuida, formada por cadenas de bloques diseñadas para evitar su modificación una vez que un dato ha sido publicado usando un sistemabasado en el tiempo y el consenso de los usuarios de esta.

Cuentas.

Las cuentas son las entidades base de ethereum, y contiene 4 campos:

- Nonce.
- Valor.
- Código de Contrato.
- Almacenamiento.

Aplicaciones.

Sistemas de Token.

Algorithm 1 Contrato de Tokens.

```
1: procedure TOKEN(msg, contract)        ▷ msg contiene la información de la comunicación
2:                                       ▷ contract contiene información del contrato
3:     from = msg.sender
4:     to = msg.data[0]
5:     value = msg.data[1]
6:     if contract.storage[from] >= value then
7:         contract.storage[from] = contract.storage[from]  value
8:         contract.storage[to] = contract.storage[to] + value
9:     end if
10: end procedure
```

Sistema de identidad.

Algorithm 1 Sistemas de identidad.

```
1: procedure IDENTIDAD(msg, contract)                ▷ msg contiene la información
2:     if contract.storage[tx.data[0]] ≠ 0 then
3:         contract.storage[tx.data[0]] = tx.data[1]
4:     end if
5: end procedure
```

Más aplicaciones.

¡En el github `https://github.com/pedrobn23/Ether` tenemos más, mirenlo para profundizar vuestro saber!

Particularidades.

Conclusiones

¡Muchas gracias!