



Rust 2025



clase 9

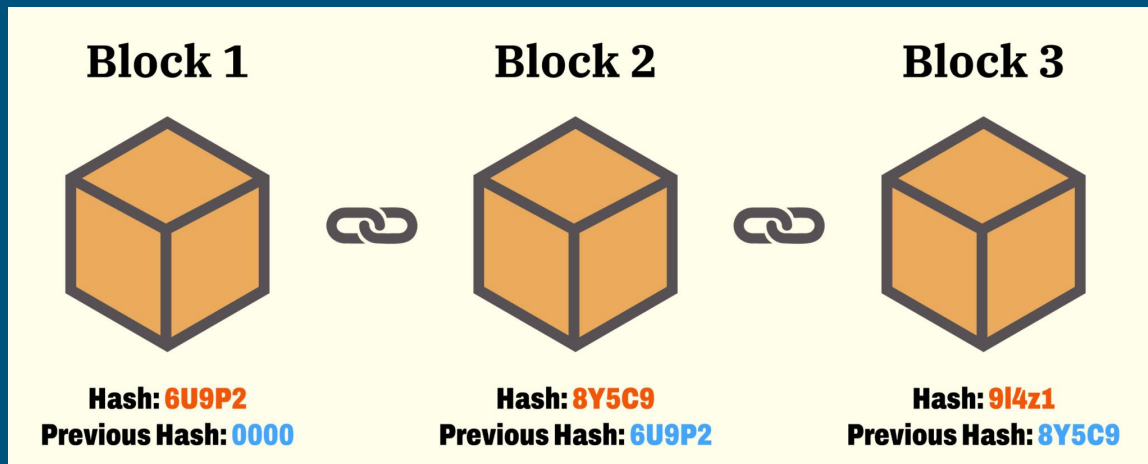


Temario

- Blockchain
- Transacción
- Nodo
- Algoritmo de consenso
- Criptografía

Blockchain

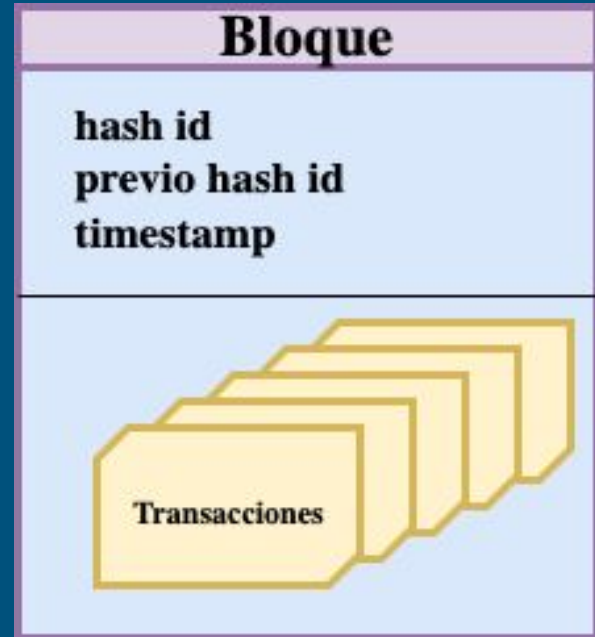
Es una estructura de datos en la que los bloques se enlazan secuencialmente. Cada bloque tiene una referencia al bloque anterior mediante un hash criptográfico. Esto crea una cadena de bloques que proporciona integridad y seguridad a los datos almacenados, ya que cualquier modificación en un bloque afectaría a todos los bloques siguientes.



Blockchain: Bloque

Un bloque contiene la siguiente información:

- ❖ Un hash que lo identifica.
- ❖ Una marca de tiempo.
- ❖ Referencia del bloque anterior.
- ❖ Transacciones del bloque.



Blockchain: Transacción

Una transacción contiene la siguiente información:

- ❖ Un hash que la identifica.
- ❖ El bloque al que pertenece.
- ❖ Un remitente.
- ❖ Un receptor.
- ❖ Valor enviado.
- ❖ Una marca de tiempo.
- ❖ Está firmada criptográficamente.

Blockchain: Transacción


Transaction Details < >

SALES! Get 15% off (one-time) for any new API Pro subscription. Code:ESFP15Q223

Overview



State

Comments

Transaction Hash:	0xb56e9c2949a80c866dd3bead6ad78c5a2b84f46fe1dfcfe9858ac23647daf9d6 
Status:	Success
Block:	17417514 1 Block Confirmation
Timestamp:	13 secs ago (Jun-05-2023 11:18:59 PM +UTC)

Sponsored:

Se quitó el anuncio. [Detalles](#)

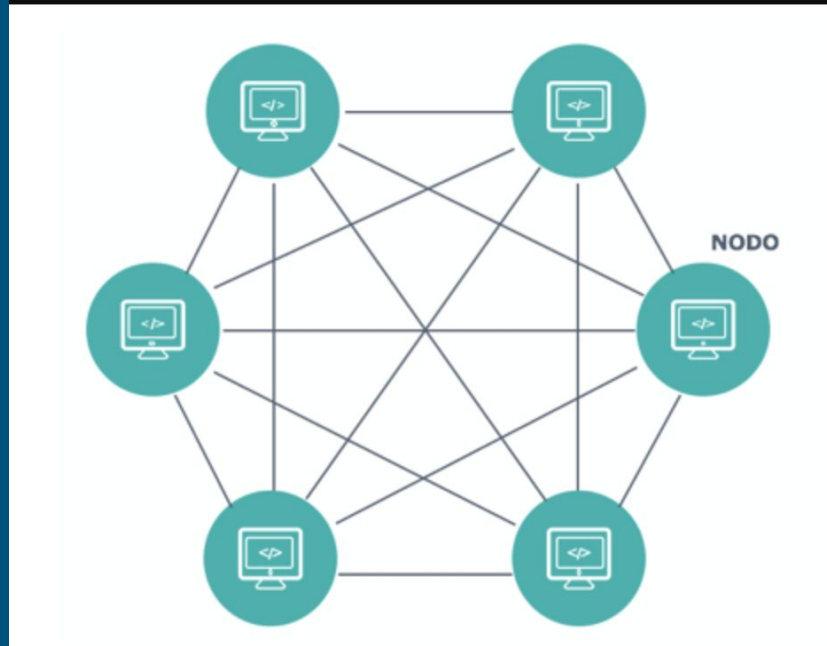
From:	rsync-builder.eth (rsync-builder) 
To:	0xcba0074a77A3aD623A80492Bb1D8d932C62a8bab 
Value:	0.076820177147786288 ETH (\$139.18)
Transaction Fee:	0.000416763474603 ETH (\$0.76)
Gas Price:	19.845879743 Gwei (0.000000019845879743 ETH)

Blockchain: Nodo

Un nodo se refiere a un dispositivo o computadora que participa en la red de blockchain. Cada nodo tiene una copia de la cadena de bloques completa o una parte de ella, dependiendo de la arquitectura de la red. Los nodos son responsables de mantener la integridad de la red y de validar las transacciones y los bloques.

Los nodos se comunican entre sí para compartir información y mantener la consistencia de la red. Su participación en la red permite la descentralización y la confianza en la seguridad y la validez de las transacciones en blockchain.

Blockchain: Nodo



Blockchain: Validación de una txn

La validación de una transacción en blockchain es el proceso mediante el cual se verifica y confirma la autenticidad y la integridad de la transacción antes de ser registrada en la cadena de bloques. Los nodos de la red de blockchain, utilizando reglas y protocolos predefinidos, revisan la validez de la transacción, asegurándose de que cumpla con los requisitos y las reglas establecidas. Esto puede implicar verificar la firma digital, comprobar el saldo suficiente del remitente y aplicar las reglas de consenso para evitar transacciones fraudulentas o inválidas. Una vez validada, la transacción se agrega a un bloque y se propaga a través de la red para su posterior confirmación y consenso por parte de los nodos.

Blockchain: Algoritmos de consenso

Un algoritmo de consenso es un conjunto de reglas y mecanismos que permite a los participantes de la red llegar a un acuerdo sobre la validez y el orden de las transacciones. Estos algoritmos garantizan la seguridad, la integridad y la descentralización de la red blockchain, y pueden variar en términos de requerimientos computacionales, participación de los nodos y distribución de poder.

Blockchain: Algoritmos de consenso

Existen varios algoritmos de consenso según el tipo de blockchain, los más conocidos son:

- PoW : Proof of work (Prueba de trabajo).
- PoS: Proof of stake (Prueba de participación).

Blockchain: Algoritmos de consenso

PoW

En este algoritmo, los nodos compiten para resolver un desafío criptográfico complejo, lo que requiere una gran cantidad de poder computacional. El primer nodo en encontrar la solución correcta tiene derecho a agregar un nuevo bloque a la cadena y recibir una recompensa, como criptomonedas. Este proceso se conoce como "minería" y ayuda a garantizar la seguridad y la integridad de la red blockchain.

Blockchain: Algoritmos de consenso

PoS

En este algoritmo los participantes de la red bloquean una cantidad de sus activos criptográficos como "apuesta" o "participación" en la red. La probabilidad de ser seleccionado para validar un bloque y recibir una recompensa se basa en la cantidad de activos que se haya apostado.

Blockchain: Criptografía

La criptografía es un campo de estudio que se ocupa de asegurar la comunicación y el almacenamiento de información mediante técnicas de codificación y decodificación.

Técnicamente, la criptografía se basa en algoritmos matemáticos que transforman la información original (texto plano o datos sin cifrar) en una forma ilegible llamada texto cifrado o criptograma.

Blockchain: Criptografía

La criptografía también se utiliza para la generación de firmas digitales, que garantizan la autenticidad e integridad de los datos. Una firma digital se genera a partir de la combinación de un mensaje y una clave privada, y puede ser verificada utilizando la clave pública correspondiente.

En blockchain es esencial para asegurar la confidencialidad, la integridad, la autenticidad y la no repudiación de los datos almacenados en la cadena de bloques. A través de algoritmos criptográficos, como el cifrado, los hash y las firmas digitales, se protege la información y se garantiza la seguridad en la red blockchain.

Blockchain: Criptografía

Frase semilla (Seed Phrase): Una frase semilla, también conocida como semilla mnemotécnica, es una secuencia de palabras que se utiliza como punto de partida para generar una serie de claves privadas en una cartera de criptomonedas. La frase semilla generalmente consta de 12, 18 o 24 palabras en un orden específico y se genera mediante un algoritmo criptográfico determinista.

Blockchain: Criptografía

Clave privada (Private Key): Una clave privada es una cadena de caracteres generada aleatoriamente que se utiliza en criptografía asimétrica. En el contexto de las criptomonedas una clave privada es un número secreto que permite al titular acceder y controlar los activos digitales asociados con una dirección de criptomoneda. La clave privada se deriva de la frase semilla utilizando una función matemática conocida como algoritmo de derivación determinista.

Blockchain: Criptografía

Clave pública (Public Key): Una clave pública es la contraparte de una clave privada en criptografía asimétrica. Se deriva de la clave privada utilizando una función matemática conocida como algoritmo de derivación de clave pública. La clave pública se utiliza para recibir transacciones y validar firmas digitales. En el contexto de las criptomonedas, una clave pública se deriva de la clave privada y se utiliza para generar una dirección de criptomoneda. La dirección de criptomoneda es la que se comparte públicamente y se utiliza para recibir fondos.

Blockchain: Criptografía

Resumiendo: la frase semilla se utiliza para generar la clave privada, y la clave privada se utiliza para derivar la clave pública. La clave privada permite el control y la firma de transacciones, mientras que la clave pública se utiliza para recibir fondos y verificar firmas. La relación entre la frase semilla, la clave privada y la clave pública son fundamentales para la seguridad y realizar una transacción.

Blockchain: Criptografía

Resumiendo: la frase semilla se utiliza para generar la clave privada, y la clave privada se utiliza para derivar la clave pública. La clave privada permite el control y la firma de transacciones, mientras que la clave pública se utiliza para recibir fondos y verificar firmas. La relación entre la frase semilla, la clave privada y la clave pública son fundamentales para la seguridad y realizar una transacción.

Blockchain: Ciclo de vida de una txn

1. Creación y firma
2. Propagación a la red
3. Espera en el mempool
4. Un validador o minero la selecciona
5. Se agrega a un bloque y es validado.
6. Confirmación de la transacción

Blockchain: Bifurcación (fork)

En el caso de PoW: Ocurre cuando se producen dos bloques válidos al mismo tiempo, pero solo uno de ellos puede ser agregado a la cadena principal. Esto puede deberse a la propagación lenta de la información en la red o a la competencia entre los mineros para resolver el siguiente bloque, o por algún acto malicioso. En este caso, se forma una bifurcación temporal donde las dos ramas compiten por convertirse en la cadena principal. La resolución de cuál de las ramas es válida se resuelve por la que tenga mayor cantidad de bloques.

video recomendado de lo visto hasta ahora: <https://youtu.be/V9Kr2SuiqHw>

Temario

- Tipos de Blockchain
- Smart Contract
- Substrate & !Ink

Tipos de blockchain

- 1era Gen
- 2da Gen
- 3era Gen

Tipos de blockchain: 1era Gen

Bitcoin(2009)

- 1 bloque c/10 min
- 6 confirmaciones
- cada bloque peso max 2MB (2500 txns aprox.)
- transfers

whitepaper: <https://bitcoin.org/bitcoin.pdf>

Tipos de blockchain: 2da Gen

Ethereum(2015)

- 1 bloque c/15 segs aprox
- 64 confirmaciones
- cada bloque 200 txns aprox
- transfers y ejecución de código(smart contracts)

whitepaper: <https://ethereum.org/en/whitepaper/>

Tipos de blockchain: 3ra Gen

Polkadot(2020)

- Multichain (Relay Chain)
- Parachain
- Bridges

whitepaper: <https://polkadot.network/whitepaper/>

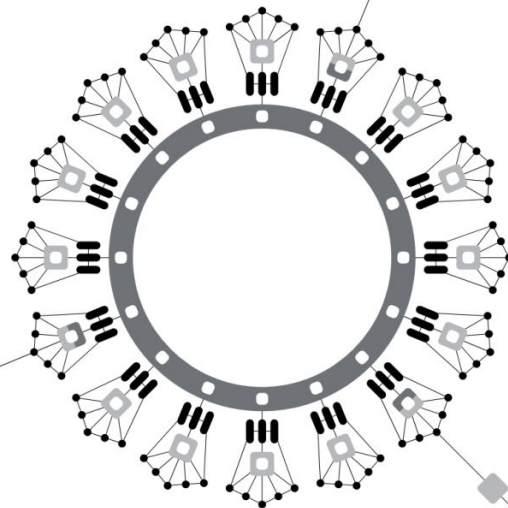
Polkadot: arquitectura

Arquitectura de Polkadot

9 •

Conectando los puntos

Polkadot une una red de fragmentos heterogéneos de cadenas de bloques llamados parachains. Estas cadenas se conectan y son aseguradas por la Relay chain de Polkadot. También pueden conectarse con redes externas a través de puentes.



Polkadot: arquitectura



Relay Chain

El corazón de Polkadot, responsable de la seguridad de la red, el consenso y la interoperabilidad de la cadena cruzada.



Parachains

Cadenas de bloques soberanos que pueden tener sus propios tokens y optimizar su funcionalidad para casos de uso específicos. Para conectarse a la Relay Chain, las Parachains pueden pagar a medida que avanzan o alquilan un espacio para una conectividad permanente.



Bridges

Cadenas de bloques especiales que permiten a los fragmentos de Polkadot conectarse y comunicarse con redes externas como Ethereum y Bitcoin.

Smart contract

Es código que se ejecuta de manera automática y autónoma en una blockchain.

Está escrito en un lenguaje de programación específico y se almacena en la blockchain como parte de un contrato digital. Una vez desplegado en la blockchain, el smart contract se ejecuta automáticamente cuando se cumplen ciertas condiciones predefinidas.

El propósito es automatizar y asegurar la ejecución de acuerdos sin necesidad de confiar en terceros. Al estar basados en tecnología blockchain, los smart contracts son transparentes, inmutables y verificables por todos los participantes de la red. Esto brinda confianza y reduce la necesidad de intermediarios, lo que puede agilizar y simplificar procesos comerciales y legales.

Smart contract

Lenguajes de programación:

- Solidity - Ethereum
- Vyper - Ethereum
- Rust - Polkadot, Solana, Cosmos
- Plutus - Cardano

Polkadot - Substrate & !Ink

Substrate es un framework para poder crear una parachain sobre polkadot :

<https://substrate.io/vision/substrate-and-polkadot/>

!Ink es un sdk que nos permite desarrollar smart contracts sobre blockchain construidas con substrate:

<https://use.ink/es/>

!Ink: tools local

1- install: <https://crates.io/crates/cargo-contract>

2- en vez de rust-analyzer se puede usar <https://crates.io/crates/ink-analyzer-ir>

3- comandos:

- a- cargo contract new nombre_del_contrato

- b- cargo contract build

- c- cargo test

!Ink: testnet

1- wallets: <https://wiki.polkadot.network/docs/wallets>

-firefox: <https://addons.mozilla.org/es/firefox/addon/polkadot-js-extension/>

-otros(chrome): <https://polkadot.js.org/extension/>

-firefox: <https://addons.mozilla.org/en-US/firefox/addon/polkgate/>

-otros(chrome, etc):<https://polkgate.xyz/>

2- faucet a zero: <https://faucet.test.azero.dev/>

3- explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

4- cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

5- ink! : <https://use.ink/es/>

extra: llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>

!Ink

faucet a zero: <https://faucet.test.azero.dev/>

explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

cargo contract new nombre_del_contrato

cargo contract build

llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>

!Ink

faucet a zero: <https://faucet.test.azero.dev/>

explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

cargo contract new nombre_del_contrato

cargo contract build

llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>