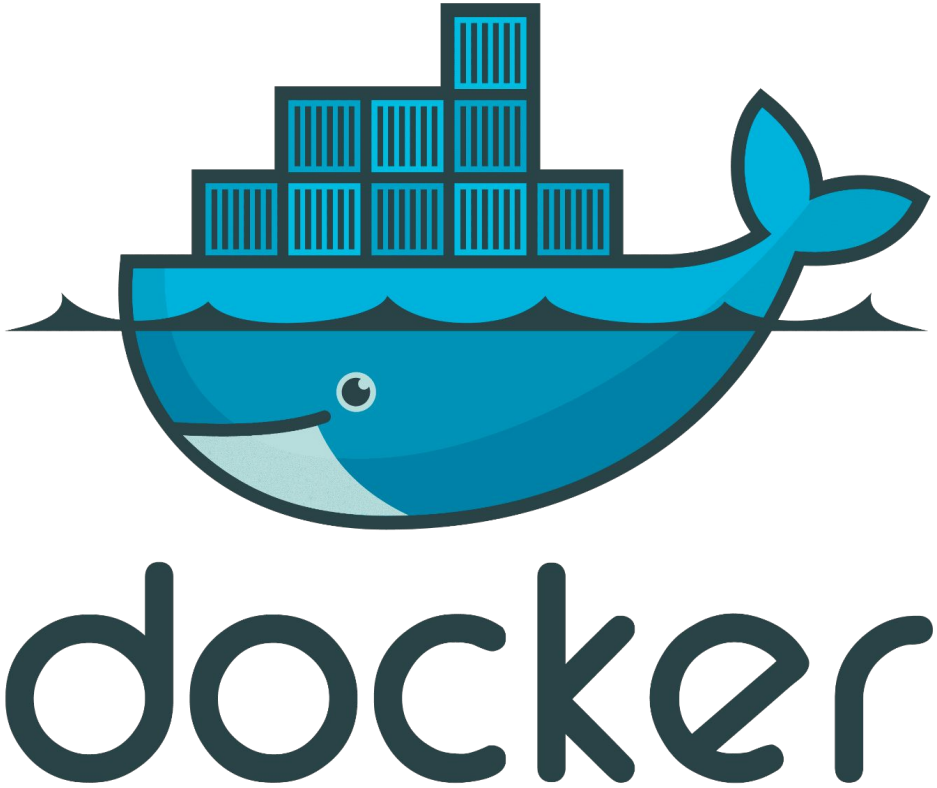
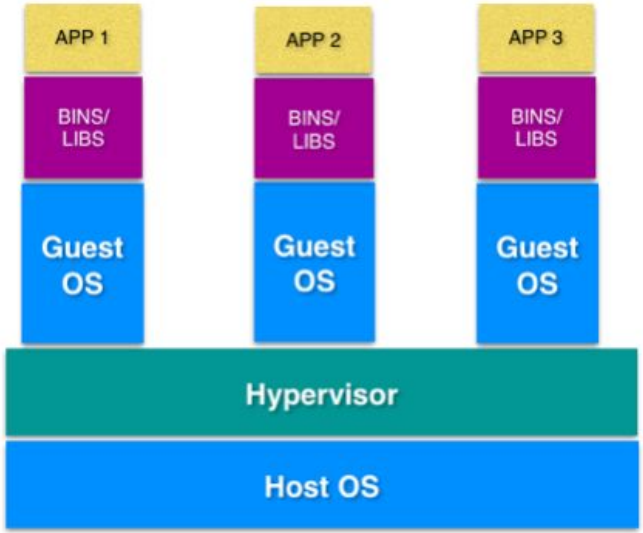
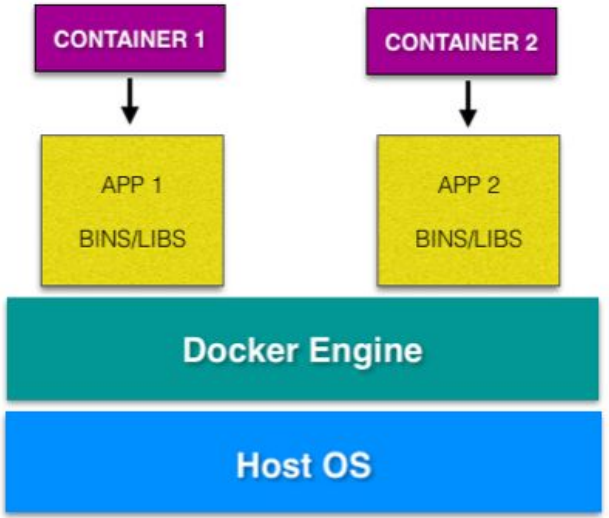

disashop
Blockchain
Visión práctica
RECARGA AQUI
PIDE EN CAJA TUS PINES
disahub



Docker

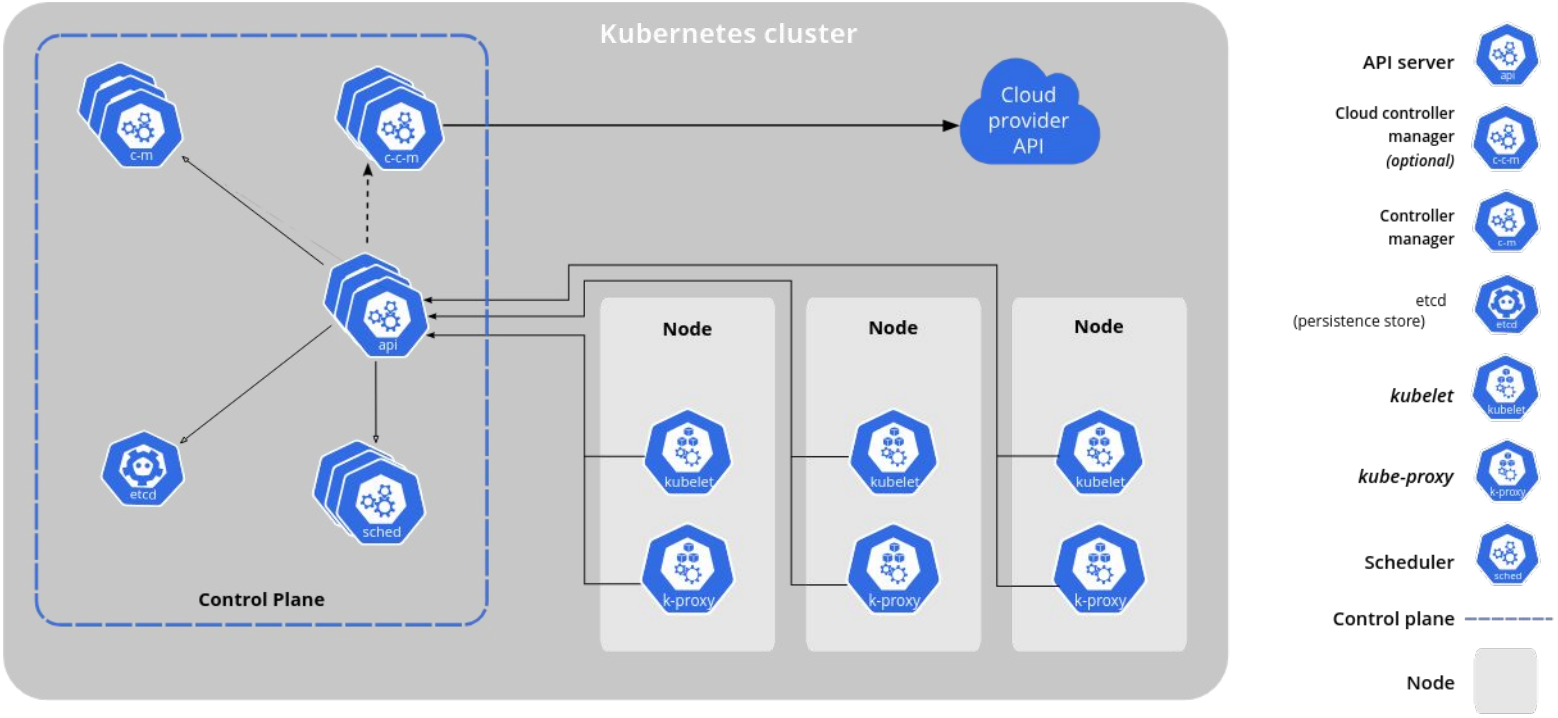


VIRTUAL MACHINE ARCHITECTURE

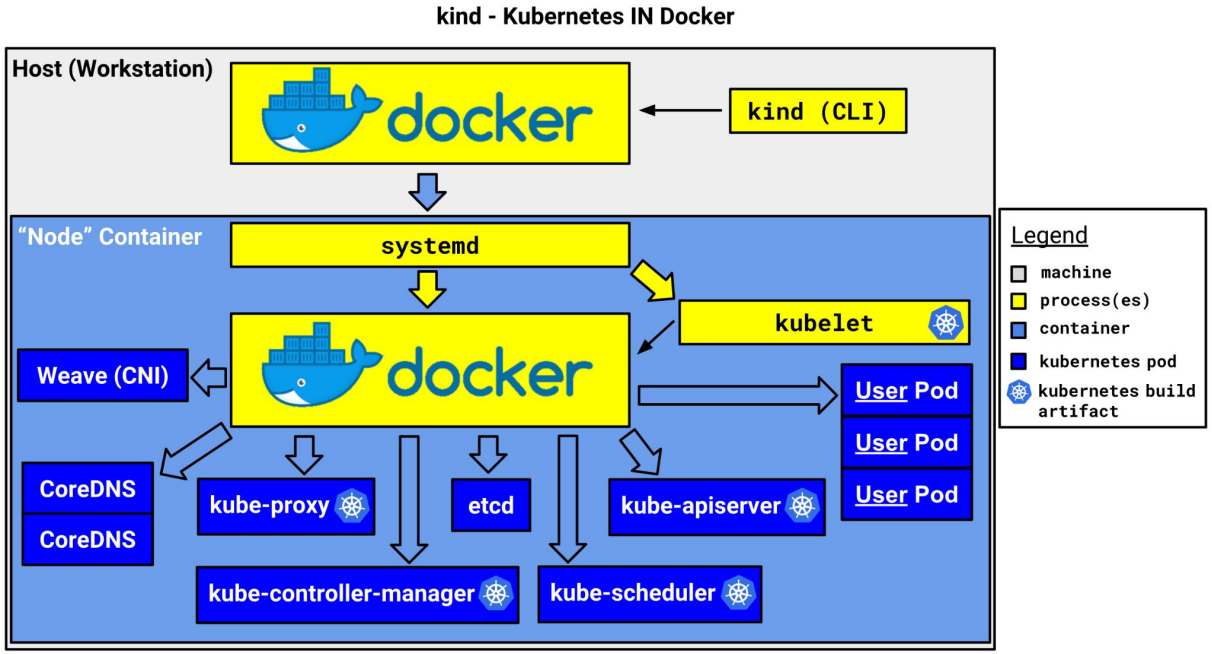


DOCKER ARCHITECTURE

Kubernetes - arquitectura



Kubernetes - kind

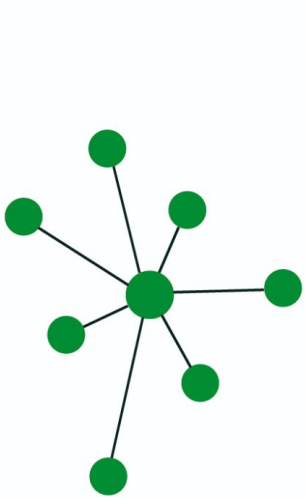


https://kind.sigs.k8s.io/

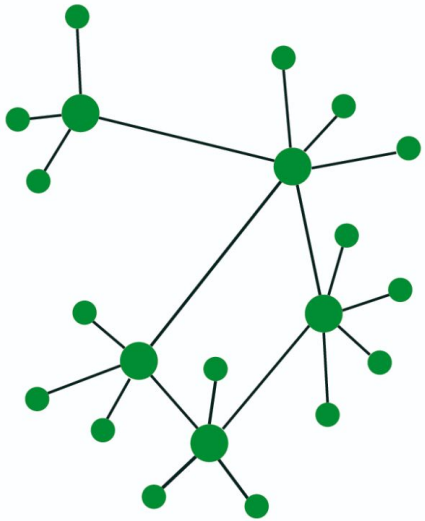
Lanzar cluster

```
ESOLEDSHOP\eduardoalonso@pt-sistemas05u:~/src/golang-blockchain/deployment$ ./cluster.sh
Creating cluster "blockchain-cluster" ...
 ✓ Ensuring node image (kindest/node:v1.27.3) 🖼
 ✓ Preparing nodes 📦 📦 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
 ✓ Joining worker nodes 🚜
Set kubectl context to "kind-blockchain-cluster"
You can now use your cluster with:

kubectl cluster-info --context kind-blockchain-cluster

Thanks for using kind! 😊
[+] Building 0.5s (16/16) FINISHED                                   docker:default
 => [internal] load build definition from Dockerfile                           0.0s
 => => transferring dockerfile: 1.09kB                                         0.0s
 => [internal] load metadata for docker.io/library/bash:5.2.26-alpine3.19      0.0s
 => [internal] load metadata for docker.io/library/golang:1.18.10-alpine3.17   0.4s
 => [internal] load .dockerignore                                              0.0s
 => => transferring context: 44B                                               0.0s
 => [builder 1/6] FROM docker.io/library/golang:1.18.10-alpine3.17@sha256:77f25981bd57e60a510165f3be89c901aec90453fd0f1c5a45691f6cb1528807  0.0s
 => [stage-1 1/4] FROM docker.io/library/bash:5.2.26-alpine3.19                0.0s
 => [internal] load build context                                              0.0s
 => => transferring context: 20.64kB                                           0.0s
 => CACHED [builder 2/6] WORKDIR /app                                          0.0s
 => CACHED [builder 3/6] COPY go.mod go.sum ./                                 0.0s
 => CACHED [builder 4/6] RUN go mod graph | awk '{if ($1 !~ "@") print $2}' | xargs go get  0.0s
 => CACHED [builder 5/6] COPY . .                                              0.0s
 => CACHED [builder 6/6] RUN CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -a -installsuffix cgo -o main .  0.0s
 => CACHED [stage-1 2/4] COPY --from=builder /etc/ssl/certs/ca-certificates.crt /etc/ssl/certs/ca-certificates.crt  0.0s
 => CACHED [stage-1 3/4] WORKDIR /app                                          0.0s
 => CACHED [stage-1 4/4] COPY --from=builder /app/main /app                    0.0s
 => exporting to image                                                         0.0s
 => => exporting layers                                                        0.0s
 => => writing image sha256:1679f2f31fad98079b2904597f6a6942f38c7797561990c9191aa3227522b4f6  0.0s
 => => naming to docker.io/library/blockchain-backend:0.1                      0.0s
Image: "blockchain-backend:0.1" with ID "sha256:1679f2f31fad98079b2904597f6a6942f38c7797561990c9191aa3227522b4f6" not yet present on node "blockchain-cluster-worker", loading...
Image: "blockchain-backend:0.1" with ID "sha256:1679f2f31fad98079b2904597f6a6942f38c7797561990c9191aa3227522b4f6" not yet present on node "blockchain-cluster-worker2", loading...
Image: "blockchain-backend:0.1" with ID "sha256:1679f2f31fad98079b2904597f6a6942f38c7797561990c9191aa3227522b4f6" not yet present on node "blockchain-cluster-control-plane", loading...
configmap/start-blockchain-gen created
configmap/start-blockchain-node created
persistentvolume/blockchain-backend created
persistentvolumeclaim/blockchain-backend created
deployment.apps/blockchain-backend created
service/blockchain-backend created
```

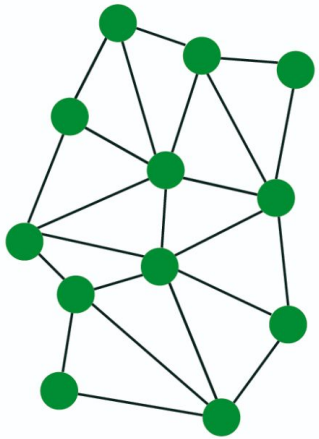
Tipos de redes de datos



red
centralizada



red
descentralizada

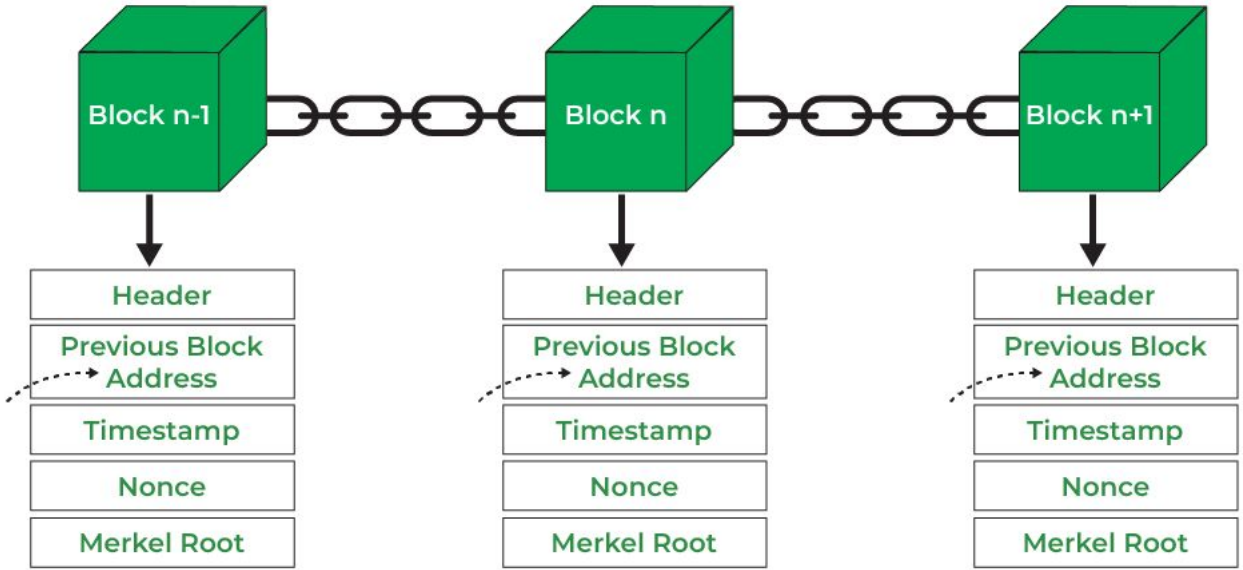


red
distribuida

Tipos de redes de datos

- Las **redes centralizadas** son aquellas que mantienen todos los datos en una única computadora, ubicación y para acceder a la información se debe acceder a la computadora principal del sistema, conocida como “servidor”.
- Las **redes descentralizadas** constituyen una alternativa para gestionar una red a la que necesitan conectarse diferentes equipos y acceder a datos. En lenguaje práctico, una red descentralizada distribuye el procesamiento de la información entre múltiples máquinas, y cada dispositivo actúa como un procesador independiente e interactúa con el resto de la red.
- Una **red distribuida** es una topología de red caracterizada por la ausencia de un centro individual o colectivo. Los nodos se vinculan unos a otros de modo que ninguno de ellos, ni siquiera un grupo estable de ellos, tiene poder de filtro sobre la información que se transmite en la red. Desaparece por tanto la divisoria entre centro y periferia característica de las redes centralizadas y descentralizadas.

Blockchain


Block n-1
Block n
Block n+1
Header
Previous Block Address
Timestamp
Nonce
Merkel Root
Header
Previous Block Address
Timestamp
Nonce
Merkel Root
Header
Previous Block Address
Timestamp
Nonce
Merkel Root

Elementos que forman una transacción blockchain:

- Entradas (inputs). Las entradas son las referencias a una salida de una transacción pasada que no ha sido empleada en ninguna otra transacción. Estas nos permiten confirmar la procedencia de los activos que se utilizarán en una transacción y son las que contienen la dirección donde originalmente se recibieron los bitcoins.
- Salidas (outputs). Estas contienen la dirección a la cual se realiza la transferencia y la cantidad enviada. Además, contienen las **direcciones de cambio o de retorno** donde son enviadas las vueltas de las transacciones (monedas que nos quedan después de realizar un pago), por lo que una transacción puede contener más de una salida.
    a. Para proteger la privacidad del usuario cada vez que se realice una transacción, la diferencia será enviada a una dirección de cambio distinta a la dirección de envío. Y se recomienda no reutilizar las direcciones por medidas de privacidad.

- Identificador (TXid). Cada transacción realizada tendrá su propio **hash**. Este hash se genera a partir de las entradas y las salidas. Este valor es el que permite identificar una transacción de forma única e irrepetible dentro de una blockchain.
- Tarifa de comisión (fee). La fee es el pequeño pago que reciben los mineros por procesar una transacción. Así, el minero que genere un nuevo bloque, recibirá una fee por cada transacción procesada dentro de dicho bloque. La comisión no viene de forma explícita en el contenido de una transacción, es decir no se asocia a ninguna salida, ya que no se sabe el minero que recibirá esa fee. Para ello, lo que se hace es dejar una determinada cantidad sin asociar a ninguna salida, y esta será entendida como comisión para los mineros. En nuestro caso se configura una cantidad concreta como fee para los mineros que es 20.

Una transacción blockchain está compuesta de entradas y salidas. El conjunto de entradas y salidas, junto a monedas a enviar y firmas criptográficas, dan como resultado un hash de transacción, llamado HASH ID.

Las entradas son HASH ID de una transacción que recibió el **monedero** y que no han sido usadas previamente, es decir que son UTXO (transacción de salida no gastada), mientras que la salida es la dirección de destino, a la cual se le crearán UTXO que posteriormente podrá usar en una transacción. Una misma dirección puede tener infinitas UTXO. Es por esto que a las UTXO se las define como un conjunto de transacciones.

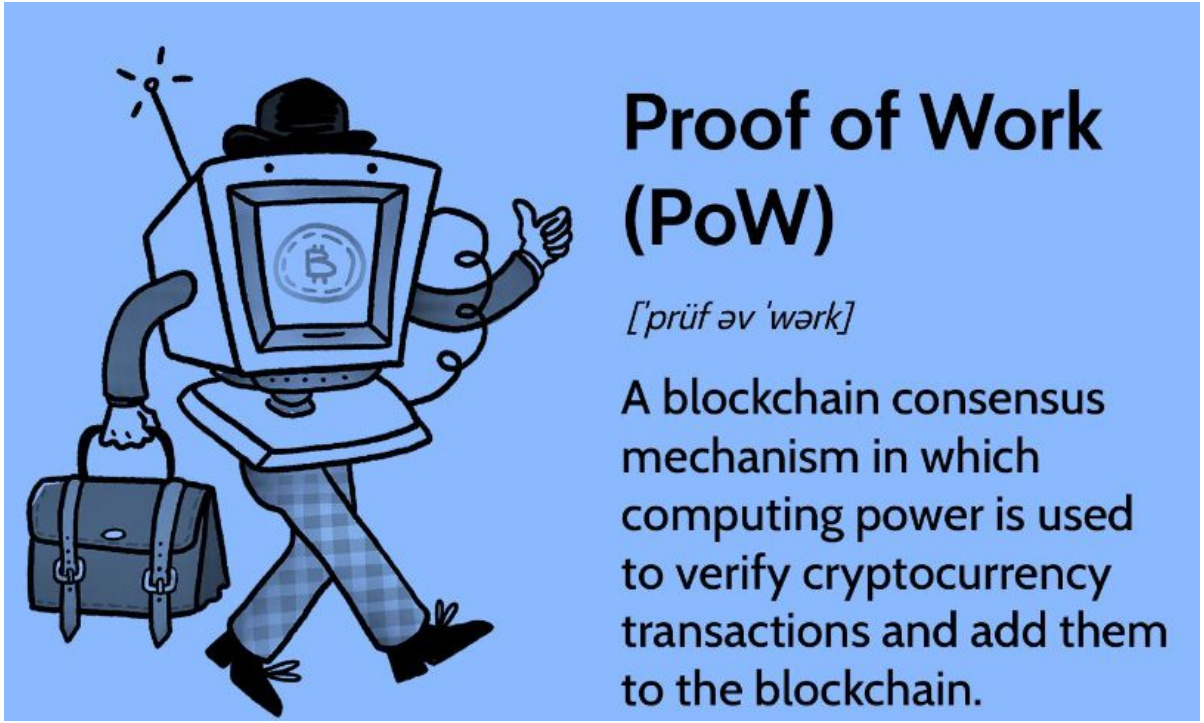
Transacción bitcoin

Monedero (wallet)

Uno de los elementos más importantes en el mundo criptográfico y de la **tecnología blockchain** son las **wallets o monederos**. El uso de estas herramientas es indispensable a la hora de gestionar nuestros activos. Es por ello que elegir una adecuada y que cumpla con nuestras necesidades es tan importante como disponer de una.

El término **wallet** hace referencia a una **cartera, billetera o monedero virtual** en el que podemos gestionar nuestros activos criptográficos. Es un **software o hardware diseñado exclusivamente para almacenar y gestionar las claves públicas y** claves privadas **de nuestras criptomonedas.**

Proof of Work

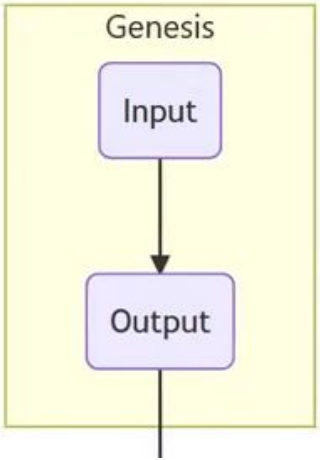


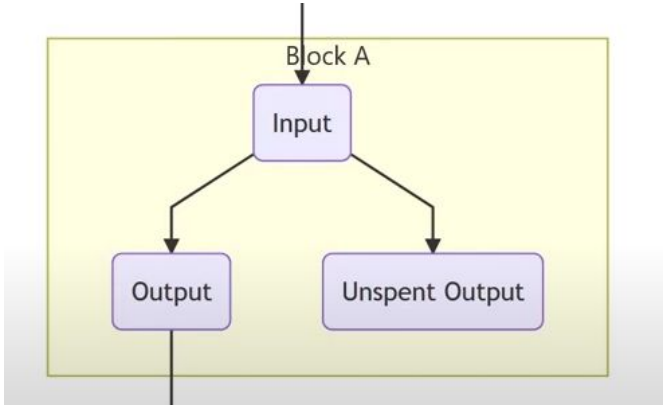
UTXO (Transacción de salida no gastada)

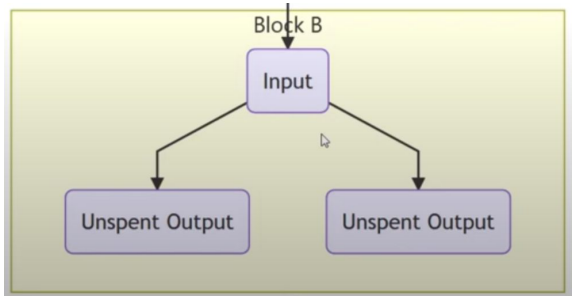
Cuando una persona necesita enviar una transacción, esta ha de nutrirse de UTXO. Es decir, de transacciones que ha recibido y que no han sido gastadas. Esto significa que una persona puede usar para una misma transacción una o más UTXO. De hecho, esos UTXO pueden formar parte de una o más direcciones de tu monedero. Y vamos más allá, incluso una transacción podría ser creada con UTXO de direcciones de diferentes monederos, siempre que se firme cada una con su correspondiente clave privada claro.

Todo esto lleva a un lugar: **una UTXO solo puede ser usada una vez**. Y esto es fundamental dentro del funcionamiento de la tecnología blockchain, pues es parte del conjunto de herramientas que garantiza que unas monedas no sean usadas más de una vez (el famoso doble gasto).

UTXO (Transacción de salida no gastada)

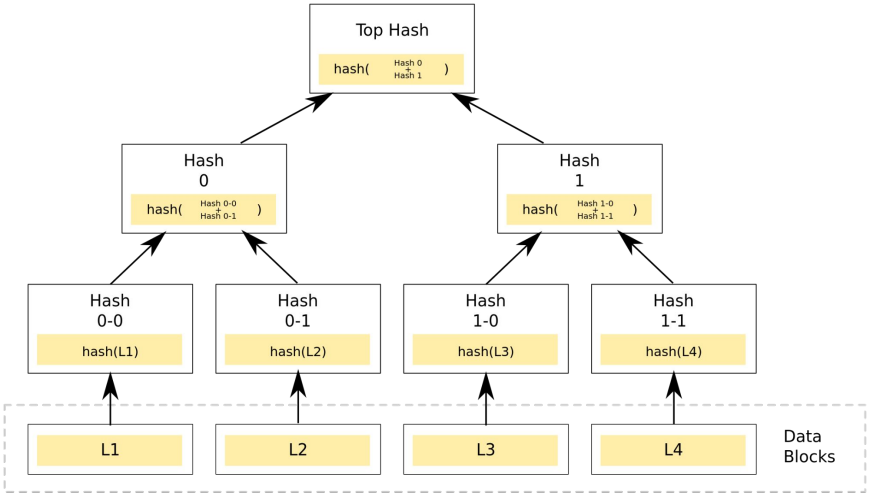




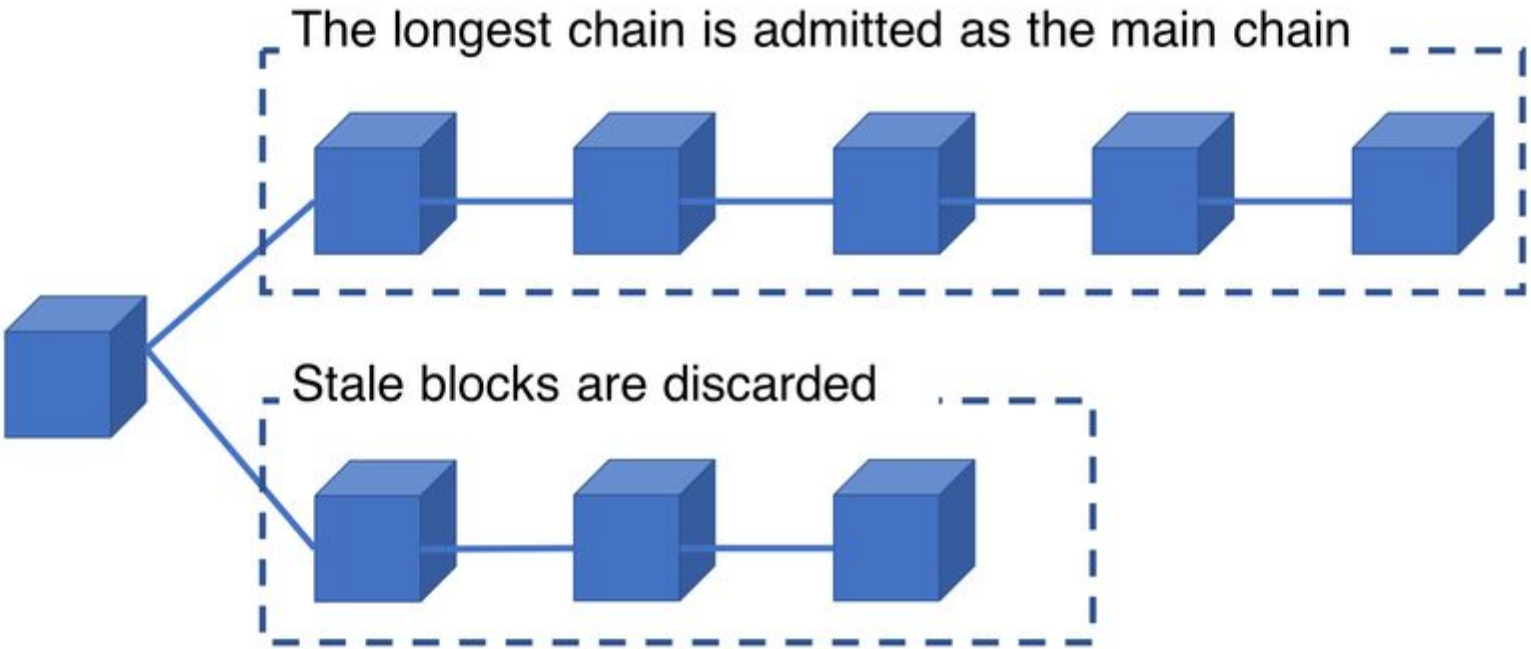


Árbol de Merkle

Un árbol Merkle, es una estructura de datos divida en varias capas que tiene como finalidad relacionar cada nodo con una raíz única asociada a los mismos. Para lograr esto, cada nodo debe estar identificado con un identificador único (**hash**). Estos nodos iniciales, llamados nodos hijos (hojas), se asocian luego con un nodo superior llamado nodo padre (rama). El nodo padre, tendrá un identificador único resultado del hash de sus nodos hijos.



Cadena más larga



Microfrontends

disashop

Quiz

https://quizizz.com/


disashop
Blockchain
RECARGA AQUI
PIDE EN CAJA TUS PINES
disahub
Disashop
@disashop_es
/DisashopSL
www.disashop.com/es