Fundamentos de Blockchain en Algorand

Dra. Rocío A. Aldeco P.

Semestre 2023-1

Algorand Centre of Excellence

3. Sistemas Distribuidos

¿Qué es un sistema distribuido?

Un **Sistema Distribuido** (SD) es una colección de entidades que cooperan para resolver un problema que no puede ser resuelto por una sola entidad.

En computación, es un conjunto de elementos de hardware y software conectados a través de una red de datos, capaz de comunicarse y coordinar sus acciones únicamente mediante el paso de mensajes.

Características

Concurrencia

Dado que la norma dentro de un sistema en red es la ejecución concurrente de programas, es necesario planificar la coordinación de dichos programas al compartir recursos.

Ausencia de un reloj global

Si los programas necesitan cooperar deben coordinarse por el paso de mensajes. Los procesos de alta coordinación dependen de la idea compartida del tiempo en que ocurren los eventos dentro de los programas.

Independencia de fallos

Cada componente del sistema puede fallar, mientras los demás siguen activos.

Características

No comparten memoria

Esta característica es usual en SD, sin embargo, es posible tener memoria común a través de la construcción de “memoria compartida distribuida”.

Separación geográfica

Los elementos que conforman un SD pueden estar conectados a través de una red WAN o LAN, es decir, existe separación física entre los diferentes componentes.

Autonomía y heterogeneidad

Los elementos de un SD están débilmente acoplados (es decir, operan a diferentes velocidades y con diferentes sistemas operativos), y estos normalmente no son un sistema dedicado.

Metas de un Sistema Distribuido

Compartir Recursos

El usuario (humano o programa) puede acceder a los recursos (hardware, software, datos o documentos) remotos de la misma manera que a los recursos locales de manera confiable.

Transparente para el usuario

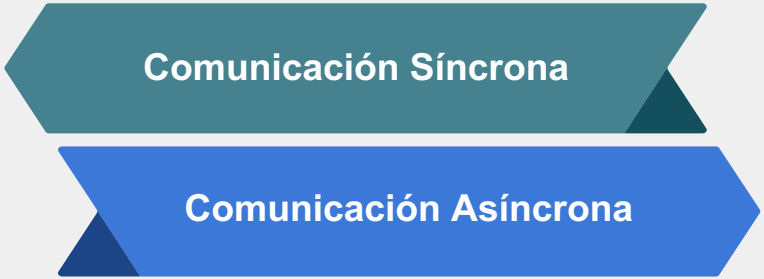
Un SD es capaz de ocultar a los usuarios qué procesos y recursos de los sistemas están distribuidos en múltiples entidades, de tal manera que para el usuario estos parezcan un único sistema de cómputo.

Apertura

Un SD se denomina abierto cuando ofrece la inclusión de nuevos servicios para compartir recursos sin perjudicar ni duplicar a los ya existentes.

Paso de Mensajes

- En un SD, el único medio de comunicación es la red de interconexión, ya que no existe un espacio de memoria compartida para los procesos que se ejecutan en el sistema. Por esta razón la comunicación de procesos se realiza por paso de mensajes.
- Al paso de mensajes se asocia una cola con cada destino de mensaje. Los procesos que envían añaden mensajes a colas remotas y los procesos que reciben remueven mensajes de las colas locales. Así podemos tener dos tipos de comunicación:



Comunicación Síncrona

- Los procesos se ajustan con cada mensaje, las operaciones de envío y recepción son operaciones bloqueantes.
- Cuando se realiza una operación de envío el proceso que envía se bloquea hasta que se emite la recepción correspondiente.
- Si un proceso emite una operación de recepción se bloquea hasta que llega el mensaje.
- Puede suceder también que la operación de recepción sea una no bloqueante (se le permite proceder cuando se ha copiado el mensaje en el buffer local y la transmisión del mensaje procede en paralelo con el envío).

Comunicación Asíncrona

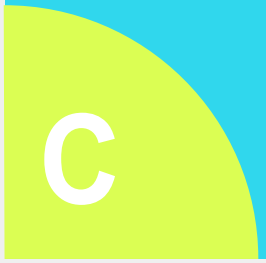
- En este caso, cuando el emisor llega a la instrucción en la que se produce el envío, no se bloquea a la espera de que el programa destinatario llegue a la instrucción en que lo recibe, sino que sigue ejecutándose con normalidad.
- Por otra parte, el programa receptor recibirá el mensaje en cualquier momento posterior al envío sin frenar con esto al emisor hasta que la recepción se produzca.
- Pero si el receptor ha llegado a una orden de recepción de mensaje pero el emisor aún no ha enviado nada, el receptor sí se bloqueará a la espera de que el emisor realice el envío.

Comunicación cliente-servidor

Es un modelo de comunicación en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Servidores

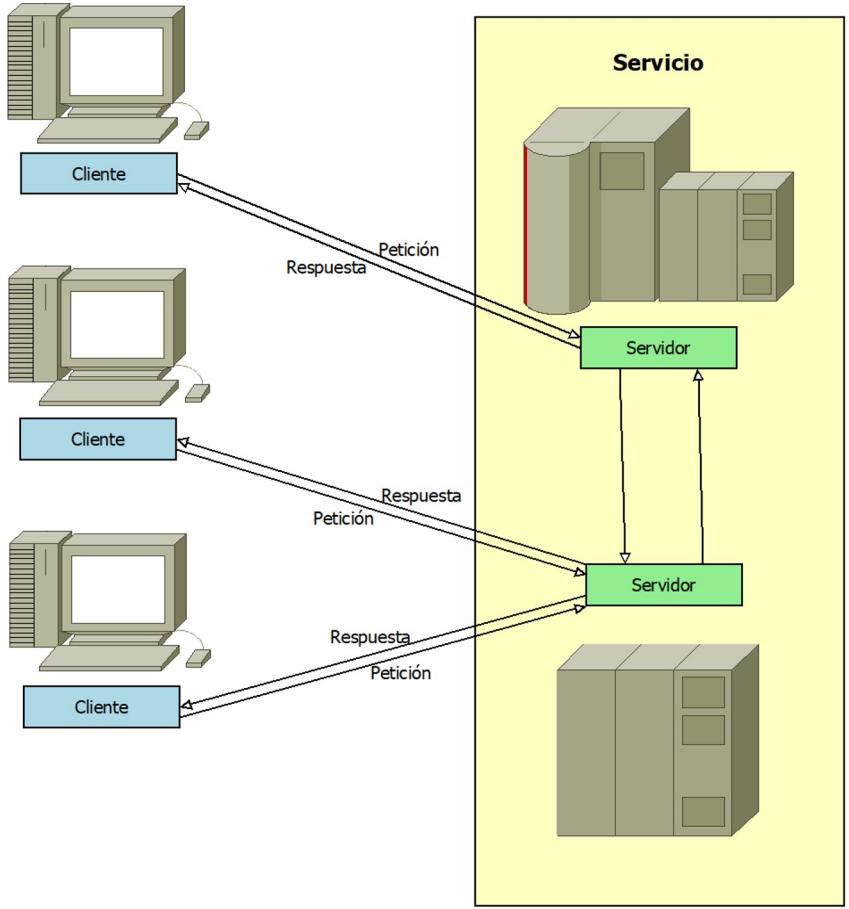
Son procesos que implementan un servicio y siempre están a la escucha de peticiones provenientes de una IP y puerto específico.



Clientes

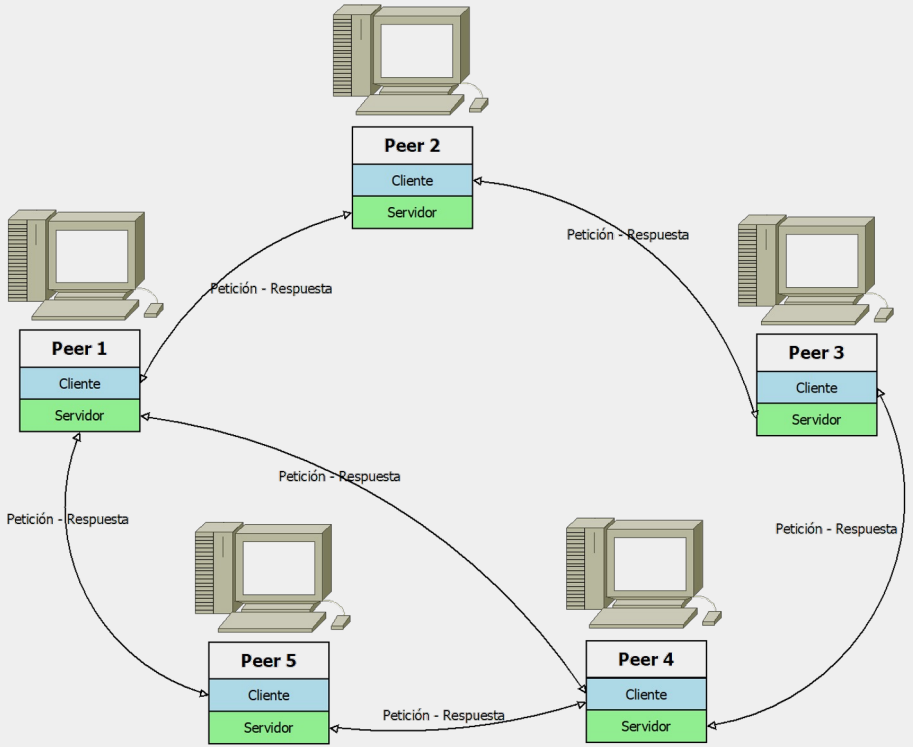
Son procesos que solicitan un servicio a los servidores a través de una petición y esperan por una respuesta del servidor

Modelo Cliente - Servidor



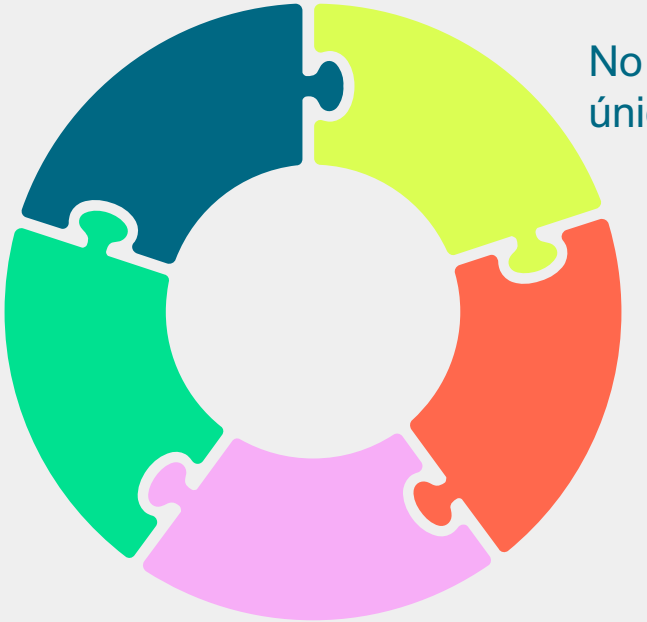
Comunicación punto a punto

En los sistemas punto a punto (peer to peer, en inglés o P2P) los servidores dedicados y clientes no existen, en contraste con la arquitectura cliente-servidor. Aquí las entidades conectadas al sistema se denominan nodos (o peers, en inglés) y estos pueden tomar el papel de cliente y servidor a la vez.



Beneficios de los sistemas P2P:

Todos los nodos comparten recursos



No existen puntos únicos de falla

Facilidad de escalamiento

Puede desplegar cualquier tipo de algoritmo distribuido

Mayor robustez del sistema

Diferencias

Sistema descentralizado

- No existe un único nodo central, sino que tiene un centro colectivo de diversos puertos de conexión.
- Todos los nodos del sistema se encuentran conectados entre sí, sin que tengan que pasar por algún punto central.
- Se rige por el principio de adhesión o participación.
- No existe un nodo único que tome decisiones, cada nodo toma su propia. decisión

Sistema distribuido

- Posee un centro individual y colectivo ausente.
- Ningún nodo posee el poder de filtrar la información que se distribuye en la red.
- Si se cayera algún nodo no desconectará a ningún otro.
- La red se rige por el principio de la interacción, y cada nodo es independiente.
- Todos los equipos están conectados entre sí mediante un protocolo de comunicaciones estándar y trabajan como una única súper computadora.

Bases de Datos Distribuidas Punto a Punto (BDD)

Una BDD es una colección de múltiples bases de datos interconectadas que se extienden físicamente en varias ubicaciones y se comunican a través de una red informática.

Bases de Datos Distribuidas Punto a Punto (BDD)

- Una base de datos punto a punto utiliza una red de recursos de varios individuos de manera colectiva para difundir datos e información entre ellos.
- Los participantes comparten los recursos (capacidad de procesamiento, el ancho de banda, espacio de almacenamiento) para aumentar la capacidad de la red colectiva.
- La potencia de cálculo se reparte entre una variedad de recursos. Al descentralizar la capacidad, se pueden añadir más clientes al sistema de lo que sería posible de otro modo. La transferencia de datos no se ralentiza con un mayor volumen de usuarios como ocurriría con una red centralizada.
- Ejemplos: Napster, FastTrack, Gnutella, BitTorrent and LimeWire.

Ventajas de las BDD

- Fiabilidad
- Seguridad
- Acceso Local
- Escalabilidad
- Desempeño
- Velocidad y eficiencia de los recursos
- Distribución del trabajo
- Reducción en el sobreflujo de mensajes dentro de la red

Descentralizadas vs Centralizadas

Algorand
Centre of Excellence

Protocolos de consenso

Algorand
Centre of Excellence

Problema del consenso

- En el problema del consenso se requiere poner de acuerdo a múltiples procesos cuando estos sólo pueden comunicarse con mensajes.
- Su objetivo es lograr la confiabilidad del sistema aún y cuando existan procesos con fallos o intentos de conspiración.
- Es un problema fundamentar el cómputo distribuido y sistemas multiagentes.
- Se asume que los mensajes se envían por un canal seguro.

Protocolos de consenso

Todos los participantes deciden y acuerdan colectivamente lo que es mejor para la red.

No sólo están de acuerdo con la mayoría, sino también con una opción que les beneficia a todos.

Funcionan en ambientes donde no hay confianza.

Son capaces de trabajar correctamente bajo fallas de algunos participantes.

Evitan la colusión de los participantes.

Propiedades

Garantizar finalización

Busqueda de acuerdos

Colaborativo

Igualitario

Inclusivo

Participativo

Garantizar integridad

Tipos de Protocolos de Consenso

Consenso basado en voto

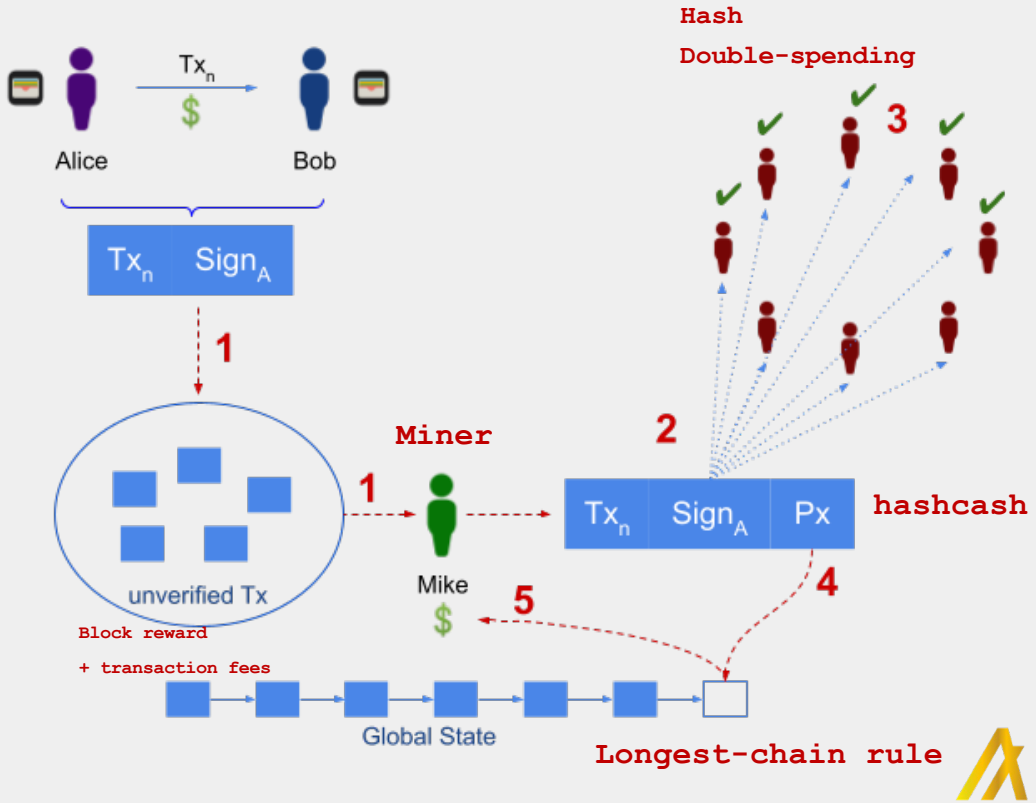
- BFT
  - PBFT
  - Corda
  - Oroha
  - RPCA
  - Stellar
- CFT
  - RAFT

Consenso basado en prueba

- PoW
- PoS
- DPoS
- PoET
- PoA
- PoSp
- PoH

Proof of Work (PoW)

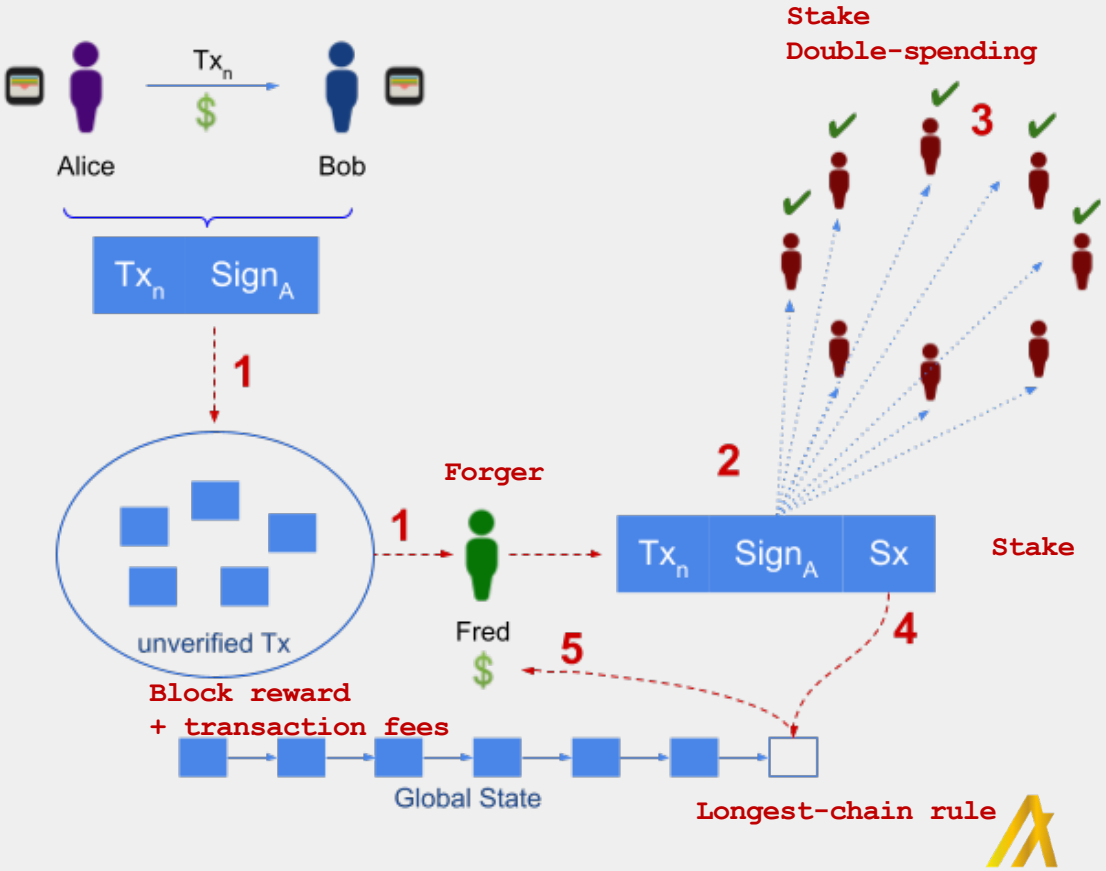
- Los nodos ganadores deben demostrar que el **trabajo realizado** y presentado reúne las condiciones para recibir el derecho a añadir nuevas transacciones a la cadena de bloques.
- Los nodos **compiten** utilizando su capacidad de procesamiento.
- Bitcoin, Litecoin, Ethereum, Zcash and bitcoin SV.



Dwork, Cynthia; Naor, Moni (1993). "Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology". CRYPTO'92: Lecture Notes in Computer Science No. 740. Springer: 139–147.

Proof of Stake (PoS)

- **Apuesta**: monedas que posee un participante y que puede invertir.
- Un **forjador** gana si apuesta más que sus competidores. La tasa de transacción es la recompensa.
- Proceso de selección pseudoaleatorio que mezcla varios factores: Selección aleatoria de bloques y selección de la antigüedad de las monedas.
- Peercoin, Cardano, Ethereum 2.0, Nxt, ShadowCash, Qora, BlackCoin.



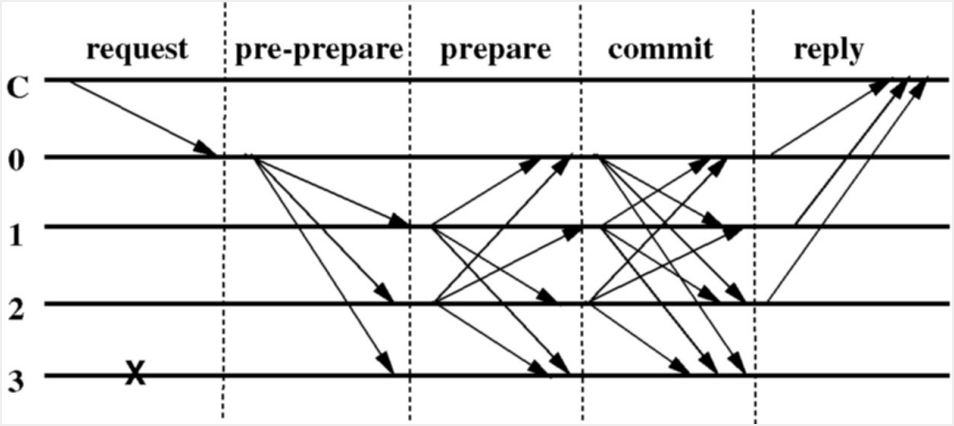
King, Sunny and Scott Nadal. "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake." (2012).

PBFT

- *Practical Byzantine Fault Tolerance*, funciona de la siguiente manera:

1. Un cliente envía una solicitud para invocar un de servicio al servidor primario.
2. El servidor primario difunde la solicitud a las réplicas.
3. Las réplicas ejecutan la solicitud y envían una respuesta al cliente.
4. El cliente espera *f+1** respuestas de diferentes réplicas con el mismo resultado; éste es el resultado de la operación.

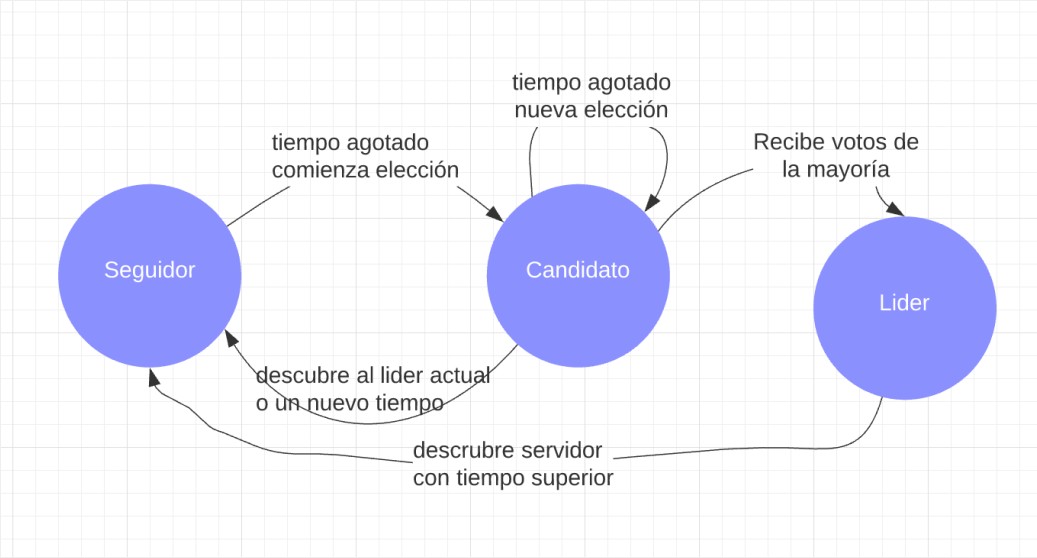
* *f* es el número máximo de réplicas que podrían fallar.



https://www.usenix.org/legacy/events/osdi99/full_papers/castro/castro_html/castro.html

RAFT

- Raft funciona eligiendo un que se encarga de aceptar las peticiones de los clientes y de gestionar la replicación del registro.
- Raft descompone el consenso en tres subproblemas:
- **Elección del líder**: Es necesario elegir un nuevo líder en caso de que falle el existente.
- **Replicación de registros**: El líder necesita mantener los registros de todos los servidores sincronizados con los suyos mediante la replicación.
- **Seguridad**: Si uno de los servidores ha enviado un registro en un índice particular, ningún otro servidor puede aplicar una entrada de registro diferente para ese índice.



https://raft.github.io/

References

- A Brief History of Blockchain. (n.d.). Retrieved November 30, 2020, from https://hbr.org/2017/02/a-brief-history-of-blockchain
- Nakamoto, S. (n.d.). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from www.bitcoin.org
- Sherman, A. T., Javani, F., Zhang, H., & Golaszewski, E. (2019). On the Origins and Variations of Blockchain Technologies. *IEEE Security and Privacy*, *17*(1), 72–77. https://doi.org/10.1109/MSEC.2019.2893730
- Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2019). A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Communications Surveys and Tutorials*, *22*(2), 1432–1465. https://doi.org/10.1109/COMST.2020.2969706
- Kaur, S., Chaturvedi, S., Sharma, A., & Kar, J. (2021). A Research Survey on Applications of Consensus Protocols in Blockchain. *Security and Communication Networks*, Vol. 2021. https://doi.org/10.1155/2021/6693731
- Pahlajani, S., Kshirsagar, A., & Pachghare, V. (2019). Survey on Private Blockchain Consensus Algorithms. *Proceedings of 1st International Conference on Innovations in Information and Communication Technology, ICIICT 2019*. https://doi.org/10.1109/ICIICT1.2019.8741353
- Bonifati, A., Bonifati, A., C, V. P. B., Chrysanthis, P. K., Sattler, K., Ilmenau, T., & Ouksel, A. M. (2008). *Distributed Databases and Peer-to-Peer Databases: Past and present*. 5--11. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.305.1464
- Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *2017 4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017*. https://doi.org/10.1109/ICACCS.2017.8014672
- Maull, R., Godsiff, P., Mulligan, C., Brown, A., & Kewell, B. (2017). Distributed ledger technology: Applications and implications. *Strategic Change*, *26*(5), 481–489. https://doi.org/10.1002/jsc.2148
- Lin, I. C., & Liao, T. C. (2017). A survey of blockchain security issues and challenges. *International Journal of Network Security*, *19*(5), 653–659. https://doi.org/10.6633/IJNS.201709.19(5).01