Computación Distribuida

UNAM, Facultad De Ciencias.

Profesor. Fernando Michel Tavera
Ayudante. Mauricio Riva Palacio Orozco
Ayudante. Yael Antonio Calzada Martín

Tarea 4.

Alumnos:
Arrieta Mancera Luis Sebastian (318174116)
Góngora Ramírez Dania Paula (318128274)

---

4 de mayo de 2023
Ciudad de México.

1. Ejercicios

1. **EJERCICIO 1.** El siguiente algoritmo describe un algoritmo para el acuerdo asíncrono con f fallas de colisión *(crash failures)* en una red de paso de mensajes totalmente conectada. La idea es obtener valores de $n - f$ procesos en cada una de las m rondas, y luego decidir sobre el valor más pequeño obtenido. El valor de m es un parámetro del algoritmo y puede depender de n y f. Como es habitual, cuando se esperan mensajes de la ronda i, los mensajes entregados con otros números de ronda se almacenarán internamente y se procesarán cuando el algoritmo esté preparado para ellos.

   OBSERVACIÓN. Cuando un proceso envía un mensaje a todos los procesos, se incluye a sí mismo.

   Demuestra que, para cualquier n y $0 < f < n/2$, existe un valor de m tal que el algoritmo satisface el acuerdo, la terminación y la validez; o muestre cómo construir una ejecución para cualquier n, $0 < f < n/2$ y m que haga que el algoritmo falle al menos uno de estos requisitos.

   Algorithm 1 Algoritmo candidato para el acuerdo asíncrono

   ```
   1: preference = input
   2: for i = 1 to m do
   3:     send (i, preference) to all processes
   4:     wait to receive (i, v) from n − f processes
   5:     for each (i, v) received do
   6:         preference = min(preference, v)
   7: decide preference
   ```

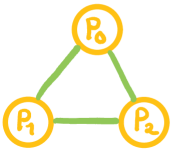
Opción 1

P.D. Existe un valor m para cualquier n, $0 < f < n/2$ t. el acuerdo satisface

- El acuerdo
- La terminación
- La validez

Demostración:

Hagamos una ejecución sencilla con la siguiente gráfica:

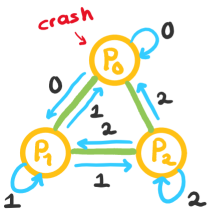


Esta gráfica cumple con ser completa.

n: 3 nodos
f: 1 falla (digamos P_0 falla)
m: se construirá a partir del desarrollo de la ejecución

$0 < f < n/2$ se cumple ✓

RONDA 1 ($i = 1 \leq m$)



P_0
preference = 0, received = [_, _, _]
P_1
preference = 1, received = [0, 1, 2]
P_2
preference = 2, received = [_, 1, 2]

Notemos que en esta ronda el nodo P_0 manda su entrada a sí mismo y a P_1 pero falla antes de poder mandar su entrada a P_2.
Al ejecutar min(received) los valores son actualizados...

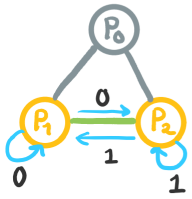
P_1
preference = min(0, 1, 2) = 0
P_2
preference = min(1, 2) = 1

} No satisface consenso

Veamos que pasa si realizamos una segunda ronda

Veamos que pasa si realizamos una segunda ronda

RONDA 2 ($i = 2 \leq m$)



P_1
preference = 0, received = [_, 0, 1]
P_2
preference = 1, received = [_, 0, 1]

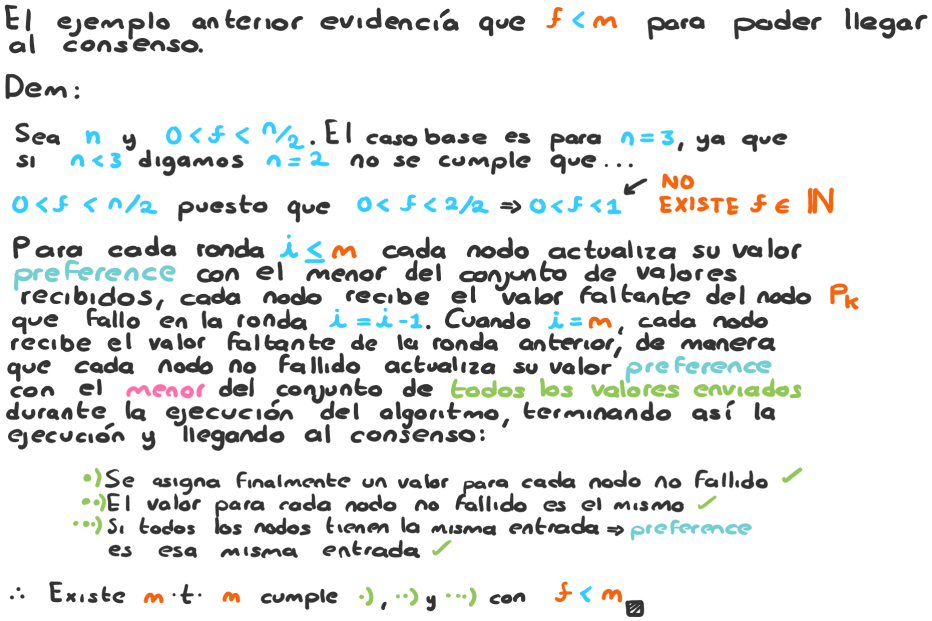
Actualizamos los valores

$preference_{P_1} = \min(0, 1) = 0$
$preference_{P_2} = \min(0, 1) = 0$

} Se llega al consenso

- Terminación ✓
- Acuerdo ✓
- Validez ✓

Con $m = 2$
$f < m$



2. ¿Cuáles son las diferencias entre un sistema centralizado, un sistema descentralizado y un sistema distribuido?

   En un sistema de procesamiento de transacciones completamente centralizado, todos los mensajes de entrada de transacciones se envían al sistema central, donde la transacción se procesa y los mensajes de salida se envían de vuelta. Por lo tanto, el sistema centralizado tiene esta sobrecarga y demora de comunicaciones independientemente de la localidad geográfica. [32]

   Por otro lado, un sistema informático descentralizado es una colección de computadoras autónomas que se comunican entre sí para realizar un servicio común, no dependientes de un nodo de procesamiento central dentro de la red. Sin embargo, los sistemas descentralizados siempre requerirán un diseño, una planificación y una gestión más cuidadosos que sus homólogos centralizados. [33]

   En cambio, un sistema de computación distribuido es un sistema con múltiples componentes ubicados en diferentes máquinas que comunican y coordinan acciones para aparecer como un único sistema coherente para el usuario final. Debido a tecnologías modernas, se ha vuelto posible conectar varios dispositivos informáticos diferentes en un sistema. Todos los dispositivos informáticos se pueden conectar a través de un cable de red o de forma inalámbrica. Las computadoras también se pueden vincular en una o varias áreas geográficas con la capacidad de agregar y quitar dispositivos del sistema. [34]

3. Explica en que consiste el teorema CAP y ¿Qué modelos de redes fueron utilizados para demostrar la conjetura de CAP hecha por Eric Brewer?

   El teorema CAP es un teorema en la teoría de sistemas distribuidos que **ilustra una compensación general en la computación distribuida**: la imposibilidad de garantizar tanto la **seguridad** como la **vitalidad** en un sistema distribuido poco confiable. **Eric**

Brewer introdujo la idea de que existe una compensación fundamental entre **consistencia**, **disponibilidad** y **tolerancia** a la partición de una red. Debido a esta compensación, es necesario sacrificar una de estas propiedades. En consecuencia, algunos sistemas garantizan una fuerte consistencia y brindan disponibilidad con mayor esfuerzo mientras que otros garantizan la disponibilidad y brindan la consistencia con mayor esfuerzo.

Para probar el teorema, fue necesario entender cuidadosamente cada uno los tres términos, para cada término se basaron en un modelo de computo distribuido, **sistemas síncronos** para la consistencia, **sistemas asíncronos** para la disponibilidad y **sistemas propensos/tolerantes a particiones** para la tolerancia. [25]

4. ¿Cuál es la definición de Layman respecto a la 'blockchain'?

   En un término sencillo, blockchain se define como una cadena de bloques digitales conectados y asociados entre sí como un libro compartido e inmutable. Inicialmente, se usaba para almacenar solo transacciones de monedas digitales, pero luego comenzó a usarse en otras aplicaciones más allá de las monedas y los pagos. [26]

5. Explica cuales son los tipos de 'blockchain'.

   La 'Blockchain' se está utilizando para realizar y transferir las transacciones o para el intercambio de información a través de una red segura.

   Básicamente, hay dos tipos de Blockchain, privado y público, se recalca que cada tipo de Blockchain consiste básicamente en un grupo de nodos, y esto funciona en el sistema de red peer-to-peer (P2P) [1]

   a) Blockchain publica:
   Es abierta y descentralizada, en este tipo las redes informáticas son accesibles para cualquier persona interesada en las transacciones, aqui se utilizan dos tipos de modelos: de Proof-of-work y Proof of-stake, también otorga autorización con respecto a la verificación de registros actuales y pasados. Además, se está utilizando para extraer e intercambiar criptomoneda.
   [1]
   Algunos ejemplos de este tipo son: bitcoin, y litecoin.

   b) Blockchain privada:
   Es restringida y no abierta, otorgan privacidad total, alta eficiencia, rapidez en transacciones, mayor escabilidad, mayor velocida. Con frecuencia son utilizadas para n sistemas y redes cerrados y estos suelen ser útiles en organizaciones, en las que cuales sólo se pueden unir miembros seleccionados. Además posee la seguridad, las autorizaciones, los permisos y la accesibilidad adecuados [1]

6. ¿Cuál fue la primera 'blockchain'?
   La blockchain se combinó con varias otras tecnologías y conceptos informáticos para la creación de las criptomonedas actuales, es decir, el efectivo electrónico protegido a través de mecanismos criptográficos en lugar de un repositorio o de una autoridad central, la primera criptomoneda basada en blockchain de este tipo fue Bitcoin en 2008. Con la bitcoin se pueden firmar digitalmente y transferir derechos sobre esa información a otro usuario y la cadena de bloques de Bitcoin registra esta transferencia públicamente, lo que permite que todos los de la red verifiquen de forma independiente la validez de estas transacciones.[2]

7. ¿Qué es el bitcoin y cuál es su relación con la 'blockchain' y los sistemas distribuidos?

   Bitcoin es un protocolo de comunicación en línea que facilita el uso de una moneda virtual, incluyendo pagos electrónicos, este se basa en un registro de transacciones que se distribuye a través de una red de computadoras. [3]

Su relación con los sistemas distribuidos es muy importante, ya que blockchain es un sistema distribuido que combina almacenamiento distribuido, P2P y algoritmo de consenso, además es un algoritmo de cifrado, el cual se caracteriza por la distribución, seguridad y confiabilidad.

También Los bloques de datos son generados por el consenso de todos los nodos distribuidos, solo que en este sistema todos los nodos del sistema distribuido son iguales, sin ningún nodo distinguido. [4]

8. ¿Cuál es la relación entre el teorema CAP y la 'blockchain'?
El teorema CAP establece que en un sistema distribuido no se puede garantizar tanto la consistencia como la disponibilidad simultaneamente, como respondimos en la pregunta anterior, la blockchain es un sistema distribuido, por lo que se ve afectado por este teorema.[5]

Conforme al teorema CAP

Consistencia:
Sabemos que esta compuesta por una gran cantidad de nodos lo cuales contienen exactamente la misma información, esto quiere decir que no hay puntos de falla en el sistem, por lo que, esta propiedad esta totalmente garantizada.
Disponibilidad:
Podemos ver que los datos pueder tardar un tiempo en propagarse a los demás, esto significa que los datos almacenados en un nodo no son los mismos que los almacenados en otro nodo en tiempo real, por lo que esta propiedad no esta garantizada.

9.

EJERCICIO 9. Explica cada una de la siguientes propiedades de la 'blockchain':

- Coherencia/Consistencia
- Tolerante a fallas
- Finalidad
- Inmutabilidad
- Solo añade (Only append)
- A prueba/resistente de manipulaciones (Tamper Resistant/Proof)
- Validez
- Garantía de terminación de las operaciones de la 'blockchain': get(), append(), verify()
- Orden
- Verificable

- Coherencia/Consistencia.
  Característica la garantiza de que todas las partes honestas generan la misma secuencia de bloques durante la ejecución del protocolo. [14]
- Tolerante a fallas.

Varias plataformas alternativas de blockchain propuestas en los últimos años intentan evitar estas limitaciones mediante el empleo de protocolos de consenso Bizantinos tolerantes a fallas (BFT) más tradicionales. [18]

- Finalidad.

- Inmutabilidad.

  Es una propiedad emergente de un DLS y no una propiedad intrínseca de la blockchain. Es bien sabido dentro de la comunidad de ingeniería de software que verificar las propiedades del sistema emergente en sistemas complejos es un problema extremadamente difícil. [15]

- Solo añade.

  Lo cual significa que los datos solo se pueden agregar a la blockchain en orden secuencial ordenado por tiempo, esto implica que una vez que se agregan datos a la blockchain, es casi imposible cambiar esos datos. [19]

- A prueba/resistente de manipulaciones.

  La blockchain utiliza una función hash y un mecanismo de cifrado asimétrico criptografía para garantizar que no se altere la información de la blockchain. [20]

- Validez.

  En la validación, cada par valida los cambios de estado de las transacciones aprobadas con respecto a la política de aprobación y la serialización. [22]

- Garantía de terminación de las operaciones.

  Dado que cada nodo en realidad se ejecuta de manera relativamente independiente, puede haber estados de inconsistentes en diferentes nodos durante la ejecución. Sin embargo, dado que cada tarea requiere que el usuario subsiguiente verifique esta restricción, después de completar la concurrencia, el nodo completará automáticamente la clasificación de las tareas. [24]

- Orden.

  En el ordenamiento, se utiliza un protocolo de consenso para producir una secuencia totalmente ordenada de transacciones son respaldadas y agrupadas en bloques de forma que este orden se transmite a todos los pares. [22]

- Verificable.

  En esta propiedad cualquier proceso debería poder verificar que los mensajes enviados se hayan contado correctamente. [23]

10. Explica como funciona la 'blockchain'

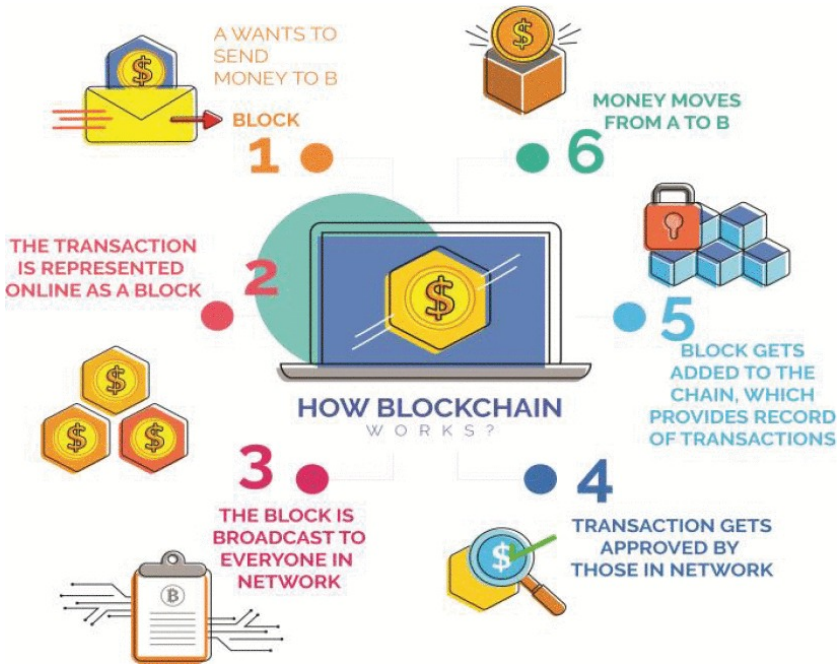
Blockchain es utilizada principalmente para crear la criptomoneda, Bitcoin, a través del mantenimiento de registros distribuidos inmutables en muchos de nodos.

Para conocer su funcionamientos supongamos que hay dos usuarios, el usuario A hace una transacción al usuario B, como blockchain es una red P2P, para seguridad esta red utiliza una prueba de identidad criptográfica para identificar al usuario A y al usuario B de manera única, tras esto la transacción se transmitirá al grupo de memoria de la red donde esperara de la verificación y validación de la transacción, tras obtener cierto número de nodos con la aprobación (es decir un consenso), despues de esto, habra un nuevo "bloqueçon la blockchain formada completamente, en la que cada nodo se actualizará con su respectiva copia, este nuevo bloque contiene todas las transacciones que ocurrieron durante este tiempo a este proceso se le denomina minería, ya que en la red P2P llega a un consenso sobre el estado actual de forma que cada nodo puede votar a través de su potencia de CPU para aceptar bloques válidos tomando extensiones o rechazar bloques no válidos negando expansiones. [6]

11. ¿Qué es un bloque?¿En qué consiste? y ¿Cuál es su función en la 'blockchain'?

    La innovación de la blockchain tiene relevancia para cualquier intercambio de recursos. Sin embargo, un cierto nivel de engaño es inevitable en los intercambios, por ello se ocupa el trabajo de terceros para proteger, validar y autorizar intercambios.

    El sistema Bitcoin ordena las transacciones/intercambios colocándolos en eventos sociales llamados **bloques**. Cada bloque después de cierto tiempo lo conecta a la cadena de bloques (blockchain), para poder ser conectado es necesario ser verificado para que el bloque encaje en la cadena de bloques. Cada bloque es distribuido en la red, estos bloques se comparan con bloques relacionados entre sí que contienen el hash del bloque anterior. Si el bloque es aprobado, entonces se conecta a la cadena de bloques generando así un historial con un nuevo registro de transacciones en el libro que todos los usuarios de la red comparten. [27]



12. Explica en que consiste la minería de bitcoins

    En la pregunta anterior se menciono que para verificar cada transacción es necesario el trabajo de terceros para proteger, validar y autorizar los bloques de código que se generan a la cadena. Quién lleva a cabo este trabajo son los famosos mineros.

    La base de Bitcoin es la minería llevado a cabo por los mineros, que consiste en un proceso intensivo de computación que se utiliza para verificar las transacciones de Bitcoin con fines de lucro. Para mantener la validez de las transacciones red de Bitcoin, debe haber un incentivo para contribuir a verificar las transacciones dentro de la cadena de bloques. Bitcoin proporciona este incentivo al recompensar a los mineros que contribuyen con nuevos bitcoins por cada bloque creado. Sin mineros, no se pueden agregar nuevas transacciones al libro público, y Bitcoin no funciona.

    La minería consiste en buscar un valor criptográfico dentro de un bloque tal que el hash del bloque cae dentro de un cierto rango. La red escala el rango para mantener una tasa promedio de un bloque nuevo cada diez minutos. Como resultado, los mineros compiten naturalmente entre sí para obtener una fracción más alta de la tasa de hash de la red para maximizar la recompensa.

Algorithm 1 Mining Process

```
nonce ← 0
while nonce < 2^32 do
    threshold ← ((2^16 − 1) ≪ 208)/D(t)
    digest ← SHA-256(SHA-256(header))
    if digest < threshold then
        return nonce
    else
        nonce ← nonce + 1
    end if
end while
```

    En resumen, la minería es una búsqueda del valor *nonce* que da como resultado un doble valor de resumen de hash SHA-256 menor que un valor dado límite. El *nonce* es un campo de 32 bits dentro de un bloque de 1024 bits. Con el fin de verificar las transacciones a un ritmo constante, este umbral varía con el tiempo en función de la dificultad $D(t)$. La red ajusta la dificultad regularmente, se espera encontrar una solución aproximadamente cada 10 minutos independientemente de la tasa de hash colectiva de la red. [28]

13. ¿Qué es ethereum?

    La tecnología blockchain es un enfoque relativamente nuevo en el campo de las tecnologías de la información. Como una de sus primeros implementaciones, bitcoin como criptomoneda ha ganado mucho atención. Junto con Ethereum, implementación de blockchain con un enfoque en los contratos inteligentes, representan el núcleo mismo del desarrollo de criptomonedas modernas. [29]

14. Realiza una comparativa entre la arquitectura de los nodos de bitcoin y los nodos de ethereum

Satoshi Nakamoto propuso un sistema distribuido con servidor de marcas de tiempo P2P que sirve como generador de la prueba computacional de las órdenes cronológicas de transacciones. Una moneda electrónica de bitcoin es definida como una cadena de firmas digitales. Cada transacción es definida como un conjunto de hash firmado digitalmente por la anterior transacción y la llave pública del siguiente propietario. La llave privada se utiliza para firmar la transacción, y la llave pública se utiliza para la verificación de la transacción. La llave pública se guarda en la billetera, la cual puede implementarse en software, hardware u online. Cada nodo en esta red P2P mantiene una copia del libro de contabilidad hardware o en línea.

Si un usuario quiere enviar cierta cantidad de monedas a otro, puede hacerlo públicamente anunciando esta transacción y depende de la red verificar su corrección. La red Bitcoin comienza con nuevas transacciones siendo transmitidas a todos los nodos. Cada nodo reúne transacciones en un bloque y trabaja en encontrar la prueba de trabajo (`proof of work`), después de lo cual transmite su bloque a la red. Los nodos en la red. aceptar el bloque como válido solo si todas las transacciones dentro de él son correctas. Si el bloque es aceptado por la red, se continúa creando el siguiente bloque y añadiéndole el hash del bloque añadido previamente.

En cambio, la blockchain de Ethereum consiste en contratos inteligentes que residen en la capa lógica de datos de la cadena de bloques. Los nodos pueden leer otros contratos inteligentes y ejecutarlos.

Todos los nodos en la cadena de bloques de Ethereum operar en tiempo real. Esto asegura que todos y cada uno la transacción que sucede es verificada por todos los nodos o ninguno. Si hay una discrepancia en un nodo en el red, todos los nodos circundantes abandonan el contrato y crear una bifurcación en la red. Esto crea una divergencia en la cadena de bloques que se descarta. La blockchain mantiene un libro de contabilidad que está completamente en línea, por lo tanto, no puede ser manipulado. Si alguien lo cambiara, el nodo sería rechazado. Los contratos inteligentes están escritos en solidity, un lenguaje de programación que se ejecuta directamente en la blockchain de Ethereum.

Los contratos son ejecutados por todos los nodos y la información actualizada se comparte entre otros nodos después de un intervalo regular. Estos contratos deben ser validados por al menos 2 nodos para activarse. A pesar de que Ethereum blockchain es de uso gratuito, cuesta *ether* para nodos que ejecutan un contrato inteligente. Este costo se refiere a *gas*. En Ethereum el gas varía dependiendo del contrato inteligente y sus funcionalidades. Además los contratos inteligentes no requieren prueba de trabajo ya que cada nodo realizara optimizaciones a las transacciones con el contrato. [30] [31]

15. Además de la arquitectura de nodos en que se diferencian bitcoin y ethereum?

Además de la arquitectura, tienen diferentes propósitos y aplicaciones, mientras bitcoin desempeña el papel de un activo de reserva digital, ethereum actúa como una blockchain de bloques para un amplio ecosistema de casos comerciales, también podemos ver a bitcoin como un almacén de valor digital global mientras que a ethereum como un motor para aplicaciones descentralizadas.

Bitcoin es como una cadena de bloques pública la cual no tiene, líderes con una amplia población de usuarios y un gobierno descentralizado, aunque su distribución de grados se aproxima a una función de ley de potencia, aun con esto, su estructura de control es muy descentralizada.

Mientras que la distribución de grados de ethereum se asemeja a una función de ley de potencia, su estructura de liderazgo es más precisa.[10]

Por lo que una diferencia importante es que el grado de descentralización en bitcoin es mayor, por otro lado el grado de descentralización en ethereum es más estable.[11]

16. Explica el algoritmo de consenso PBFT (Practical Byzantine Fault Tolerance).

El algoritmo PBFT es un algoritmo de consenso para resolver problemas bizantinos, este fue propuesto por primera vez por Miguel Castro y Barbara Liskov en 1999, es uno de los algoritmos más populares para blockchain.

El algoritmo de consenso PBFT se compone principalmente de un protocolo de consenso, un protocolo de reemplazo de vista y un protocolo de punto de control. El protocolo de coherencia se utiliza para garantizar la coherencia de los datos guardados por todos los nodos en la red, lo que se realiza a través de la comunicación mutua entre los nodos durante tres etapas; el protocolo de reemplazo de vista se utiliza para reemplazar el nodo defectuoso.

El algoritmo de consenso PBFT se divide principalmente en los siguientes tres procesos:

a) Preparación previa: después de que el nodo maestro recibe el mensaje para la solicitud de servicio y este verifica que es correcto, se genera un mensaje de preparación previa de acuerdo y este es transmitido a los nodos.

b) Preparar: después de recibir el mensaje de preparación previa del nodo maestro, el nodo esclavo verifica si el contenido del mensaje ha sido alterado, después de que se verifico el mensaje, el nodo esclavo generará un mensaje de preparación de acuerdo con el mensaje de preparación previa y lo transmitirá a todos los demás nodos.

c) Confirmación: se transmiten mensajes de confirmación a otros nodos tras recibir mensajes de confirmación de $2f + 1$ nodos, se ha llegado a un consenso, por lo que los nodos ejecutarán la solicitud y escribirán datos.

Toda esta información se saco de esta fuente [12]

17. Explica el algoritmo de consenso de Paxos descubierto por Lamport.

Ya que uno de los integrantes de este equipo expuso sobre este tema, reciclaremos su información.

Consenso de Paxos: Explicación El algoritmo de consenso de Paxos fue desarrollado por Leslie Lamport, lo publico en 1998 en el paper llamado The Part-Time Parliament.

El algoritmo de consenso de Paxos como tal, es una familia de algoritmos, un protocolo a seguir para el consenso.

Sabemos que el consenso es un problema en el cual requerimos que una red de procesos lleguen a un valor acordado. En redes asincrona, se asume que los mensajes pueden retrasar arbitrariamente.

Lamport hace una analogía con un parlamento en el que solo los miembros que asisten a la sesión tienen derecho a votar. Lamport utiliza esta analogía para explicar cómo se puede lograr el consenso utilizando el algoritmo Paxos. El parlamento se llama "Part-Time Parliament"porque solo algunos miembros del grupo están presentes en cada sesión. [7]

En paxos, en la descripción simple de Lamport se basa en tres secciones: [8]

- Proponentes: Los encargados de proponer valores para el consenso. Tienen un id cons dos carecteristicas: unico y siempre es mayor a cualquier id anterior

- Aceptadores: Se encargan de aceptar (o ignorar) los valores propuestos. Basicamente es utilizarse como una memoria en la cual almacena los valores mandado por los proponentes

- Aprendices: Aprenden el valor acordado.

Hay 2 fases para llegar al consenso [9]:

a) Fase de preparación:
   - Preparar: un proponente elige la ronda en la cual empezar, digamos ronda i, comienza la ronda enviando un mensaje de $< Preparar >$ a todos los aceptantes.
   - Promesa: Cuando un aceptador recibe el mensaje ¡Preparar¿en la ronda i, envía un mensaje $< Promise >$ de vuelta al proponente (a menos que haya prometido no hacerlo). De esta forma, el aceptador promete que no participará en ninguna ronda menor que i y se mantendrá esta promesa. Junto con la promesa, el aceptante envía el último valor que ha votado y la ronda asociada a este valor.

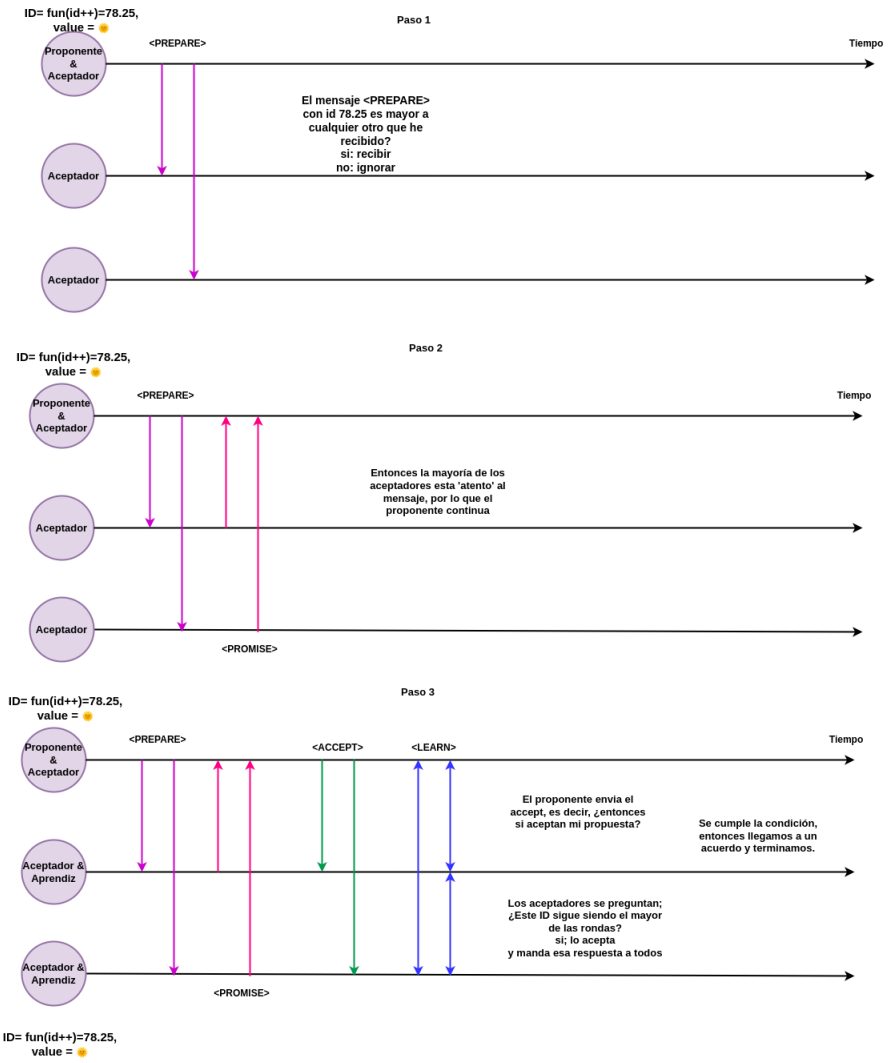
b) Fase de aceptación:
   - Aceptar: Después de reunir un quórum de $n - f$ promesas, (la mayoria de promesas, es decir si eran 3 aceptadores, necesitamos 2 promesa) para la ronda i de los aceptantes, el proponente envía un mensaje de $< Aceptar >$ a todos los aceptantes en la que el proponente pide votar por un valor seleccionado de la siguiente manera:

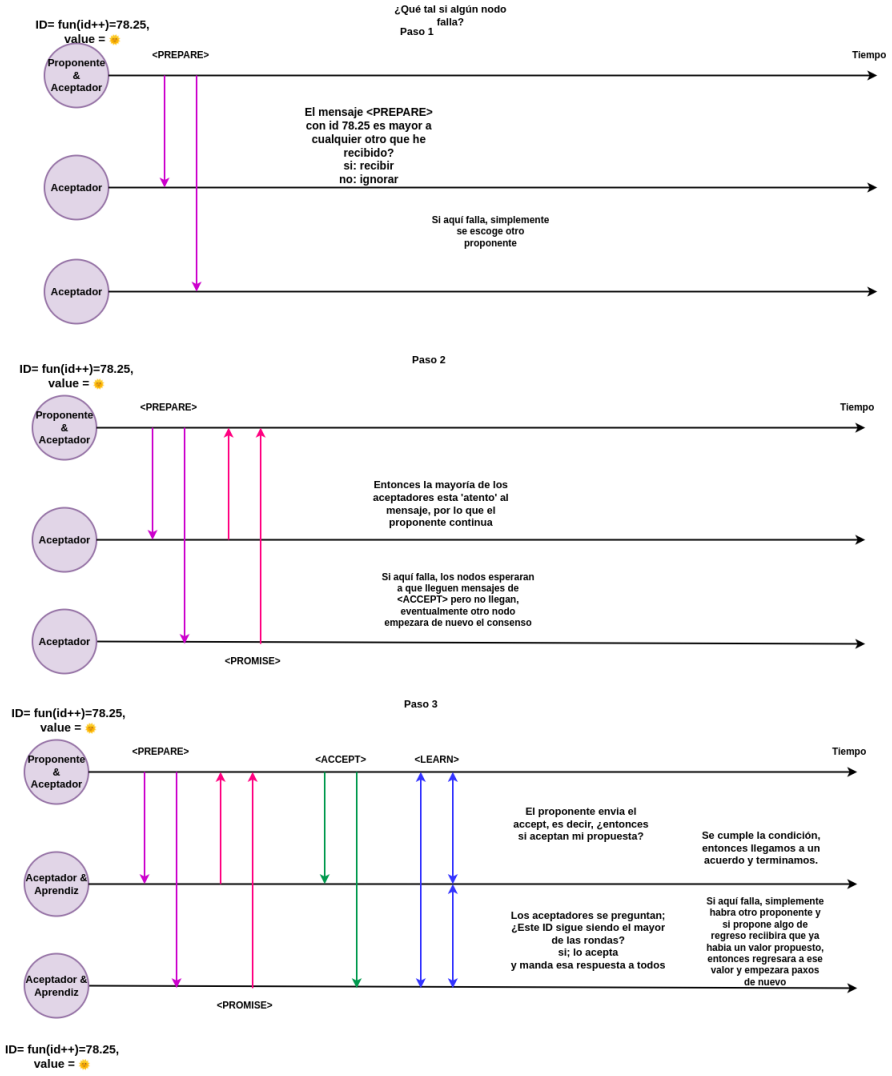
     Sea un valor v propuesto por el proponente, si no hay aceptador en el quórum (la mayoría), ha votado al menos alguna vez:

El valor *vval* en las promesas que está asociado con la ronda más alta será el valor aceptado.

- Aprender: Si un aceptador recibe un mensaje $< Aceptar >$, y este aun no ha prometido otra cosa, vota por el valor en el mensaje y envía un mensaje $< Learn >$ a todos los aprendices para informarles sobre la votación. Los aceptantes votan solo una vez en cada ronda.
- Se elige el valor: Si un alumno recibe $n - f < Learn >$ mensajes para la misma ronda y con el mismo valor de un quórum de $n - f$ aceptantes, entonces se elige este valor. Es decir; si se recibe la mayoria de mensajes, con el mismo id y valor aceptados, se escoge ese valor como el acordado.
  En este algoritmo basicamente se busca que lo que la mayoria acepte se el consenso final.

En este algoritmo basicamente se busca que lo que la mayoria acepte se el consenso final.
Consenso de Paxos: Ejecución




¿Qué tal si algún nodo falla?
Paso 1
ID= fun(id++)=78.25, value =
Proponente & Aceptador
<PREPARE>
Tiempo
El mensaje <PREPARE> con id 78.25 es mayor a cualquier otro que he recibido?
si: recibir
no: ignorar
Aceptador
Si aquí falla, simplemente se escoge otro proponente
Aceptador
Paso 2
ID= fun(id++)=78.25, value =
Proponente & Aceptador
<PREPARE>
Tiempo
Entonces la mayoría de los aceptadores esta 'atento' al mensaje, por lo que el proponente continua
Aceptador
Si aquí falla, los nodos esperaran a que lleguen mensajes de <ACCEPT> pero no llegan, eventualmente otro nodo empezara de nuevo el consenso
Aceptador
<PROMISE>
Paso 3
ID= fun(id++)=78.25, value =
Proponente & Aceptador
<PREPARE>
<ACCEPT>
<LEARN>
Tiempo
El proponente envia el accept, es decir, ¿entonces si aceptan mi propuesta?
Se cumple la condición, entonces llegamos a un acuerdo y terminamos.
Aceptador & Aprendiz
Los aceptadores se preguntan; ¿Este ID sigue siendo el mayor de las rondas?
si; lo acepta
y manda esa respuesta a todos
Si aquí falla, simplemente habra otro proponente y si propone algo de regreso recibira que ya habia un valor propuesto, entonces regresara a ese valor y empezara paxos de nuevo
Aceptador & Aprendiz
<PROMISE>
ID= fun(id++)=78.25, value =

18. Explica en que consiste el consenso de nakamoto (Nakamoto Consensus).

    El consenso de Nakamoto es el protocolo de consenso de Bitcoin. En este protocolo, las transacciones se empaquetan en una lista ordenada de bloques, con forma de árbol de Merkle, un bloque es válido si su hash criptográfico calculado tiene n ceros iniciales, donde n es el parámetro de dificultad y se ajusta constantemente, un nuevo bloque generado que haya sido validado se transmitirá al resto de nodos. Un nodo agregará el nuevo bloque recibido en la cadena de bloques local almacenado, si verifica la corrección del bloque, la seguridad del protocolo está garantizada esto gracias a que la mayoría del poder del hash en todo el sistema apuntará a extender la cadena legítima más rápido que cualquiera que intente reescribir la historia o gastar doblemente la moneda. [21]

19. ¿Qué es 'Polkadot'? ¿En qué consiste? y ¿Cómo funciona?

    Polkadot es un prótocolo de blockchain múltiple heterogeneo, tiene como objetivo reunir el poder de seguridad de todos estas cadenas juntas en un sistema de seguridad compartido, fue presentado por primera vez en 2016 por Gavin Wood. Polkadot utiliza una cadena central llamada cadena de retransmisión que se comunica con múltiples cadenas fragmentadas heterogéneas e independientes llamadas paracadenas.

    Las paracadenas son clientes de la cadena de retransmisión, que proporciona un servicio de seguridad a estos clientes, incluida una comunicación segura. Ese es el único propósito de la cadena de relevos; las paracadenas son las entidades que proporcionan funcionalidad a nivel de aplicación. Las paracadenas solo necesitan adherirse a una interfaz especificada. Algunas de estas expectativas son componentes naturales de las blockchains. En terminos generales Polkadot son blockchains multtiples de forma heterogénea escalable.

    La red de la cadena de retransmisión de Polkadot consta de nodos y roles, los nodos son las entidades a nivel de red que ejecutan físicamente a Polkadot y los roles son entidades a nivel de protocolo que realizando un propósito particular.

    [13]

20. ¿Qué es un NFT? y ¿Cuál es su relación con la 'blockchain'?

    Kevin McCoy creó el primer NFT en 2014 y lo apodó Quantum en la blockchain de Namecoin. Un token no fungible (non-fungible token) es un activo criptográfico en una blockchain que contiene información de identificación única y códigos que los separan unos de otros, la razón porla cual se popularizo es porque los inversores acudieron en masa a numerosos mercados de criptomonedas, incluido NFT, como resultado de la fuerte caída de las tasas de interés del mercado mundial esto a causa del COVID-19. Los NFT se enfocan en evitar la falsificación, ya que cada token lleva la firma digital del propietario y, por lo tanto, es único.[16]

    Entonces podemos ver que un NFT es una unidad de datos almacenada en una blockchain que certifica que un activo digital es único, no intercambiable, de forma que ofrece un

certificado de propiedad digital único para el NFT, en otras palabras, una NFT permite obtener de donde procede el objeto digital, ofreciendo datos de quién lo posee, quien poseyó anteriormente y quién creó la NFT, así como cuál de las muchas copias es la original.

Como dato adicional; los NFT formaban parte de la blockchain de Ethereum, pero cada vez más cadenas de bloques han implementado sus propias versiones de NFT4. [17]

Referencias

[1] *Aithal, P and Saavedra, P and Aithal, Sreeramana and Ghoash, Surajit (2021, december). Blockchain Technology and its Types-A Short Review. pags:189-200 DOI:10.30954/2322-0465.2.2021.7.*

[2] *Yaga, D. , Mell, P. , Roby, N. and Scarfone, K. (2018), Blockchain Technology Overview pags:4-6.* https://doi.org/10.6028/NIST.IR.8202

[3] *Böhme, Rainer, Nicolas Christin, Benjamin Edelman, and Tyler Moore. 2015. "Bitcoin: Economics, Technology, and Governance."Journal of Economic Perspectives, 29 (2): 213-38. DOI: 10.1257/jep.29.2.213.*

[4] *Sun, E., Meng, K., Yang, R., Zhang, Y. Li, M. (2021). Research on Distributed Data Sharing System based on Internet of Things and Blockchain. Journal of Systems Science and Information, 9(3), 239-254.* https://doi.org/10.21078/JSSI-2021-239-16

[5] *Sankagiri, S., Wang, X., Kannan, S., Viswanath, P. (2021). Blockchain CAP Theorem Allows User-Dependent Adaptivity and Finality. In N. Borisov, C. Diaz (Eds.), Financial Cryptography and Data Security - 25th International Conference, FC 2021, Revised Selected Papers (pp. 84-103).* https://doi.org/10.1007/978-3-662-64331-0_5

[6] *Chen, G., Xu, B., Lu, M. et al. Exploring blockchain technology and its potential applications for education. Smart Learn. Environ. 5, 1 (2018)* https://doi.org/10.1186/s40561-017-0050-x

[7] *Lamport, Leslie (1998). FThe Part-Time Parliament DOI:10.1145/279227.279229.* https://doi.org/10.1145/279227.279229

[8] Delzanno, G., Tatarek, M., Traverso, R. (2014, Agosto 26). Model Checking Paxos in Spin. DOI: 10.4204/EPTCS.161.13

[9] *Zhao, Wenbing (Enero 2015). Fast Paxos Made Easy: Theory and Implementation. DOI:10.4018/ijdst.2015010102.* https://doi.org/10.4018/ijdst.2015010102

[10] *Partida, A., Gerassis, S., Criado, R., Romance, M., Giráldez, E., Taboada, J. (2022). Modeling Bitcoin plus Ethereum as an Open System of Systems of Public Blockchains to Improve Their Resilience against Intentional Risk. Electronics* http://dx.doi.org/10.3390/electronics11020241

[11] *Lin, Qinwei Li, Chao Zhao, Xifeng Chen, Xianhai. (2021). Measuring Decentralization in Bitcoin and Ethereum using Multiple Metrics and Granularities.*

https://doi.org/10.48550/arXiv.2101.10699

[12] *Xiandong Zheng, Wenlong Feng. (2021). Research on Practical Byzantine Fault Tolerant Consensus Algorithm Based on Blockchain. Journal of Physics: Conference Serie DOI: 10.1088/1742-6596/1802/3/032022*

[13] *Burdges, Jeff, Cevallos, Alfonso, Czaban, Peter, Habermeier, Rob, Hosseini, Syed, Lama, Fabio, Alper, Handan, Luo, Ximin, Shirazi, Fatemeh Stewart, Alistair Wood, Gavin. (2020). Overview of Polkadot and its Design Considerations.* https://doi.org/10.48550/arXiv.2005.13456

[14] *Lucianna Kiffer and Rajmohan Rajaraman and abhi shelat, (2022), A Better Method to Analyze Blockchain Consistency, DOI:10.1145/3243734.3243814*

[15] *Conte de Leon, Daniel Stalick, Antonius Jillepalli, Ananth Haney, Michael Sheldon, F.T.. (2017). Blockchain: properties and misconceptions. Asia Pacific Journal of Innovation and Entrepreneurship. 11. 286-300. DOI:10.1108/APJIE-12-2017-034.*

[16] *Taherdoost, H. (2022). Non-Fungible Tokens (NFT): A Systematic Review. Information, 14(1), 26. MDPI AG* http://dx.doi.org/10.3390/info14010026

[17] *Nadini, Matthieu Alessandretti, Laura Di Giacinto, Flavio Martino, Mauro Luca, Maria Baronchelli, Andrea. (2021). Mapping the NFT revolution: market trends, trade networks and visual features.* https://doi.org/10.48550/arXiv.2106.00647

[18] *J. Sousa, A. Bessani and M. Vukolic, .A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform,"2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg, Luxembourg, 2018, pp. 51-58, doi: 10.1109/DSN.2018.00018.*

[19] *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition: Bashir, Imran, ISBN: 9781788839044*

[20] *Takaaki Yanagihara, Akihiro Fujihara, Cross-Referencing Method for Scalable Public Blockchain, Internet of Things,Volume 15,2021,* https://doi.org/10.1016/j.iot.2021.100419.

[21] *Jiao, Zhenzhen Tian, Rui Shang, Dezhong Ding, Hui. (2018). Bicomp: A Bilayer Scalable Nakamoto Consensus Protocol.* https://doi.org/10.48550/arXiv.1809.01593

[22] *Ruan, Pingcheng and Loghin, Dumitrel and Ta, Quang-Trung and Zhang, Meihui and Chen, Gang and Ooi, Beng Chin, A Transactional Perspective on Execute-Order-Validate Blockchains, DOI:10.1145/3318464.3389693*

[23] *Tassos Dimitriou, Efficient, Coercion-free and Universally Verifiable Blockchain-based Voting, Computer Networks,Volume 174,2020* https://doi.org/10.1016/j.comnet.2020.107234.

[24] *Y. Wang, Z. Tu, Y. Bai, H. Yuan, X. Xu and Z. Wang, .A Blockchain-based Infrastructure for Distributed Internet of Services,"2021 IEEE World Congress on Services (SERVICES), Chicago, IL, USA, 2021, pp. 108-114, doi: 10.1109/SERVICES51467.2021.00045.*

[25] *Gilbert, S. C., Lynch, N. R. (2012). Perspectives on the CAP Theorem. IEEE Computer, 45(2), 30-36. DOI:* https://doi.org/10.1109/mc.2011.389

[26] *Bhutta, M. A. R., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. S., Hanif, M., Song, H., Alshamari, M. A., Cao, Y. (2021). A Survey on Blockchain Technology: Evolution, Architecture and Security. IEEE Access, 9, 61048-61073. DOI:* https://doi.org/10.1109/access.2021.3072849

[27] *S. Rajput, A. Singh, S. Khurana, T. Bansal and S. Shreshtha, "Blockchain Technology and Cryptocurrenices,"2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 909-912, DOI:* 10.1109/AICAI.2019.8701371.

[28] *M. Vilim, H. Duwe and R. Kumar, .Approximate bitcoin mining,"2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 2016, pp. 1-6, DOI:* 10.1145/2897937.2897988

[29] *D. Vujičić, D. Jagodić and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview,"2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2018, pp. 1-6, DOI:* 10.1109/INFOTEH.2018.8345547

[30] *D. Vujičić, D. Jagodić and S. Ranđić, "Blockchain technology, bitcoin, and Ethereum: A brief overview,"2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2018, pp. 1-6, DOI:* 10.1109/INFOTEH.2018.8345547

[31] *R. A. Canessane, N. Srinivasan, A. Beuria, A. Singh and B. M. Kumar, "Decentralised Applications Using Ethereum Blockchain,"2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 75-79, DOI:* 10.1109/iconstem.2019.8918887

[32] *B. Ciciani, D. M. Dias, B. R. Iyer and P. S. Yu, .A hybrid distributed centralized system structure for transaction processing,ïn IEEE Transactions on Software Engineering, vol. 16, no. 8, pp. 791-806, Aug. 1990, DOI:* 10.1109/32.57619

[33] *J. N. Gray, .An approach to decentralized computer systems,ïn IEEE Transactions on Software Engineering, vol. SE-12, no. 6, pp. 684-692, June 1986, DOI:* 10.1109/TSE.1986.6312966

[34] *A. Alalawi and A. Al-Omary, .A Survey On Cloud-Based Distributed Computing System Frameworks,"2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-6, DOI:* 10.1109/ICDABI51230.2020.9325662

2. Herramientas adicionales

- **Obtener el pdf de un artículo usando su DOI:** https://sci-hub.se