

05/03/2018

Néstor en blockchain

TXT SANTIAGO PALLADINO IMG WALTER MONTES DE OCA

¿Qué onda los bitcoins? ¿Cómo funciona una blockchain?

No hay vuelta que darle: en el 2017 el Bitcoin y las otras cripto se pusieron de moda. Las búsquedas en Google explotaron, y hasta Juan Di Natale le dedicó un programa entero al tema.

Cada vez más gente se sube a este tren con la promesa de plata fácil, sin saber qué están comprando o cómo funciona, y tomando la tecnología que permite que existan las monedas descentralizadas, ***blockchain,*** como sinónimo de *bitcoin*. Pero ojo, ese desconocimiento no es exclusivo del vecino que metió unos pesos en esto un viernes a la noche 'a ver qué pasa', sino que se convirtió en un verdadero fenómeno global. Tal es la influencia de *ponerleblockchainalascosas* que se generó una burbuja especulativa enorme (o no) respecto de todo lo que incluya la palabra.

Sin ir más lejos, hay empresas que están viendo cómo se triplica su valor en bolsa sólo por agregar *blockchain* en su nombre.

Algo importante a destacar en este punto es qué no es este artículo: **no es una guía para inversores**. Especificado el *disclaimer* (y con ahora el 7% del tráfico original en el artículo), podemos empezar.

Función de hash

Para entender cómo funciona una *blockchain* tenemos que introducir antes otro concepto: el de función de hash (que no, nada tiene que ver con la droga). Es una operación que se puede aplicar sobre cualquier dato, y devuelve un código identificador (análogo a un código de barras) que lo identifica (DUH). Algo así como una huella digital, digital. La ventaja es que, en este caso, el tamaño de la huella es siempre el mismo, sin importar el tamaño del dato original. Por ejemplo:

hash de la frase "El gato y la caja 2018"
= +OcHh3Oj/kh1b/9+qZMR8PE4wp/y7wM3p6n4Ok4HYEc=
hash de un capítulo de *Un Libro Sobre Drogas*
= BYE+Hdbx+OMwJt+UjDJaxHIuwigJimF3EtDWkcYsctY=
hash del libro completo *El Señor de los Anillos*
= Xg9w58oVeElknK20ymbAoAupS7cKEte8Intic9+INtE=

Hay muchas funciones de *hash* distintas, y las más divertidas tienen varias propiedades interesantes. Una de ellas es que, dado un hash, es muy difícil (sino imposible) reconstruir el dato original. Tan difícil (sino imposible) como entender cómo es una persona a partir de su huella dactilar.



Acá ‘reconstruyendo’ personas.

Otra propiedad que se desprende de la anterior es que **un cambio mínimo en el dato se traduce en un hash totalmente distinto**. Volviendo a nuestro ejemplo:
hash de “El gato y la caja **2018**”
= +OcHh3Oj/kh1b/9+qZMR8PE4wp/y7wM3p6n4Ok4HYEc=
hash de “El gato y la caja **2017**”
= S+W/QLUexIRBjJ1lCKMux/ftmHbNMbLb1H58rA4tdes=
Esto permite que el *hash* se pueda usar como método para verificar la integridad de un documento o un archivo. Así, si necesito hacerle llegar un archivo a alguien y asegurarme de que le llegue tal cual y sin que nadie lo modifique en el medio, le puedo mandar el archivo por un canal (digamos en un pendrive) y el hash de ese archivo por otro canal (digamos por mail). Con ambas cosas en mano, el receptor puede obtener el hash de la información del pendrive y compararla con el hash que le llegó por mail. Así sabe si le llegó bien y se asegura de que nadie lo modificó en el medio.
Entonces, un hash permite no sólo identificar sino también **verificar la integridad de una información.**

Blockchain 101

Una *blockchain* es una forma de llevar un registro distribuido (es decir, no centralizado) de prácticamente cualquier cosa. Si bien los ejemplos que más resuenan son los de criptomonedas, se podría usar una blockchain para llevar títulos de propiedad sin escribanos de por medio, o guardar el historial de los boletines oficiales.



Un mundo sin bancos ni escribanías. Dramatización.

Lo primero que hay que ver es cómo se reparte la información en este *registro distribuido*. Si no hay un agente central que sea dueño de una *blockchain* y la distribuya, ¿cómo se hace para obtenerla y mantenerla funcionando?

En primer lugar, para participar de una blockchain tiene que existir una computadora (que puede ser la de casa, y que va a denominarse *nodo*), que tenga andando un programa encargado de encontrar y conectarse con otros nodos. En su conjunto forman una red descentralizada que está constantemente intercambiando información. En este punto, la *blockchain* recuerda a otros sistemas de intercambio de información descentralizados ya conocidos, como el viejo y querido *BitTorrent*.

Una *blockchain* no es más que una cadena secuencial de **bloques** (*block-chain* es, literalmente, *cadena de bloques* en inglés). A su vez, cada bloque está formado por un conjunto de operaciones que modifican el estado actual de este registro distribuido: puede ser el registro de una nueva propiedad, la publicación de un

nuevo boletín, o una transferencia de *Bitcoins* de una persona a otra. Estas operaciones se llaman **transacciones**.

De esta forma, toda nueva información en una *blockchain* aparece ordenada en un nuevo bloque al final de la cadena, que se retransmite entre todos los nodos de la red.



Todos tuvimos un tío entusiasmadísimo con subirse a la blockchain.

Validando la data

Vamos a suponer como ejemplo que tenemos una *blockchain* que lleva un registro de propiedad automotor, que al parecer no es tan mala idea porque a alguien en Lituania ya se le ocurrió y levantó 15 palos verdes de inversión para hacerlo (¿alguien dijo *ponele-blockchain-a-todo*?).

Dentro de esta *blockchain*, entonces, una transacción puede ser '*Alicia le transfirió su Fiat 600 a Roberto*'. Como veíamos antes, una transacción es algo que modifica el estado del registro distribuido: el Fiat 600 pasa de las manos de Alicia a las de Roberto.

Ahora bien, ¿cómo se sabe que un nodo de la red que está pasando información, no está mintiendo? Si se ve la transacción del Fiat 600 en un bloque de la blockchain, ¿cómo se sabe si es verdad? ¿cómo se sabe si Alicia le dio o no el auto a Roberto?

Lo primero es verificar que la transacción esté firmada por Alicia, que haya sido consentida. No vamos a entrar en firma digital en esta parte (eso está acá), pero

digamos que hay una manera en que se puede verificar *criptográficamente* que Alicia firmó la transacción. El primer requisito para que una transacción sea válida, entonces, es que esté firmada por quien la emite. Incluso podríamos pedir en nuestra blockchain que esté también firmada por Roberto. Porque el consentimiento tiene que ser mutuo.

Lo siguiente, bastante lógico si uno lo piensa, es que Alicia tiene que tener en su poder el auto para poder transferirlo. Si justo a un tal Néstor le falta su Fiat 600 de la puerta de su casa, Alicia no puede transferirlo a Roberto a menos que demuestre que es efectivamente de ella. En términos de blockchain, Alicia tiene que poder probar que había otra transacción que le otorgaba posesión del vehículo antes de poder pasárselo a alguien más.

No sólo eso: Roberto también tiene que asegurarse de que no haya otra transacción en la que Alicia le vende el Fiat a otra persona al mismo tiempo que a él (ataque conocido como *double-spend*). Lo bueno es que todas las transacciones en una *blockchain* están ordenadas según los bloques de la cadena, por eso Roberto puede verificar esto fácilmente.

Supongamos otro escenario más interesante: Roberto sale a dar una vuelta con su flamante Fiat 600, atropella a un transeúnte y, preso del pánico, se da a la fuga. Tenemos ahora un pedido de captura para el dueño del Fiat 600. A Roberto entonces le conviene hacer desaparecer de la *blockchain* la transacción en la que Alicia le vendió el auto, así él no queda pegado. Es decir, Roberto tiene un incentivo para que su nodo deje de enviarle al resto de la red la transacción en cuestión.

Acá es donde entra la magia de la *blockchain*, gracias a las funciones de *hash*. Una de las propiedades más interesantes que tiene una *blockchain* es la inmutabilidad: **una vez que una transacción fue confirmada no puede ser modificada o removida**.

¿Cómo se logra esto? Resulta que cada bloque en una *blockchain* (variaciones más, variaciones menos) tiene un identificador que es un hash calculado sobre todas sus transacciones, información general como la fecha en que se generó o el hash que identifica al bloque anterior. O sea que el bloque nuevo siempre tiene información

sobre el que está justo antes de él, generando así una cadena secuencial entre los bloques.

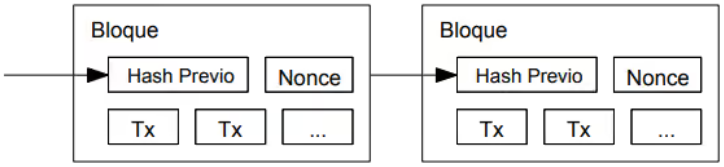
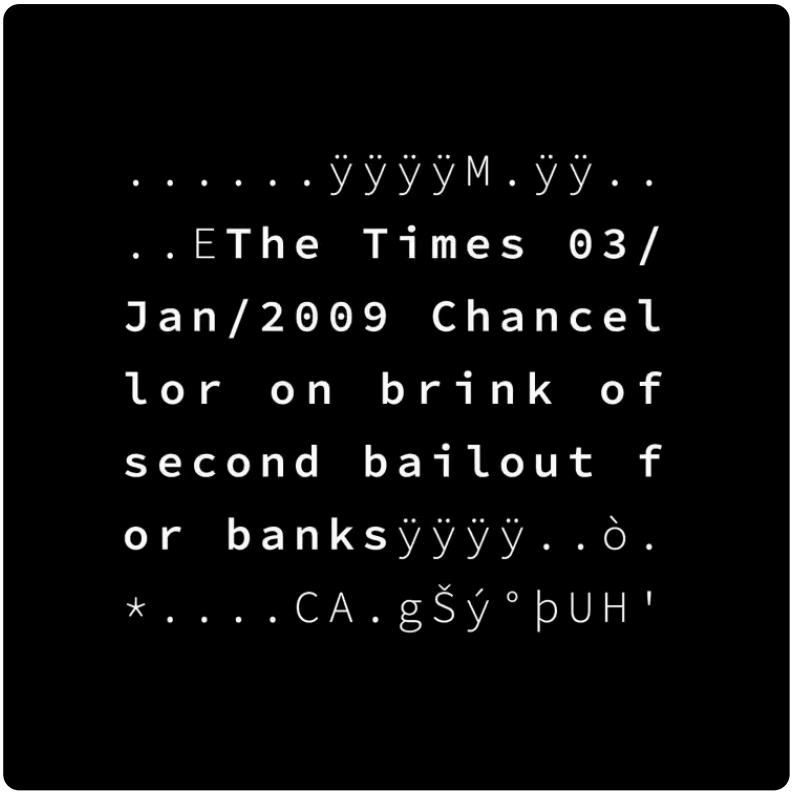


Diagrama sacado derechito del paper original de Bitcoin. Prev Hash es el hash del bloque anterior, Tx es cada transacción, y Nonce es una sorpresa para más adelante. Esta estructura sirve para cualquier blockchain.

Esto da una manera (un *algoritmo*) de validar la integridad de una *blockchain*, es decir, de chequear que no haya sido modificada. Al recibir un bloque nuevo, podemos recalcular el hash de todas sus transacciones (más el hash del bloque anterior y la fecha de creación), y verificar que coincida con su identificador. Y no paramos ahí: repetimos el proceso con el bloque anterior, y el anterior a ese, y así hasta llegar al primer bloque de toda la cadena, también llamado el *genesis block*, que es un bloque conocido por todos.

Dato de color: en el caso de *Bitcoin*, el *genesis block* tiene los primeros 50 bitcoins (hoy cerca de medio millón de dólares), junto con una frase sacada del diario del día como yapa que es un guiño a la motivación detrás de las criptomonedas:



"Diario The Times, 03/Enero/2009, Canciller cerca del segundo rescate financiero a los bancos". Siendo que Bitcoin fue creado como un mecanismo que haría innecesarias las instituciones financieras, el titular es bastante idóneo. (imagen vía @blancoagostina)

Volvamos al ejemplo del Fiat 600. Si Roberto decide retransmitir el bloque que tenía su transacción pero sin incluirla, entonces el hash de ese bloque cambia, lo que causa que **también cambien los hashes de todos los bloques subsiguientes** en la cadena, que hacen referencia a ese bloque.[°B] Como el hash de cada bloque depende del anterior, un cambio (por mínimo que sea) causa una reacción en cadena que altera todos los identificadores que siguen.

En otras palabras, si Roberto quiere 'reescribir la historia', tiene que ocuparse de cambiar no sólo el bloque que le interesa, sino todos los que siguen en la cadena. **Si cada bloque fuera un diario y cada transacción un artículo, la *blockchain* fuerza a que, para cambiar una palabra en un artículo del diario de la semana pasada, sea necesario también reimprimir todos los diarios desde ese día hasta hoy.** Digamos que si hubiera existido esta tecnología en 1984 (la novela, no el año) el trabajo del protagonista en el Ministerio de la Verdad hubiese sido mucho más complicado.

Todo re lindo hasta acá, pero ¿cómo hacemos que Roberto efectivamente no vaya a reescribir todos los diarios hasta hoy? Al fin y al cabo, sabemos que calcular un hash es fácil, y nada le impediría recalcular los hash de todos los bloques desde el que modificó hasta el más reciente, y empezar a distribuirlos por la red.
Eso depende directamente de cómo se agregan nuevos bloques en una cadena. Acá empieza la segunda parte de la historia: los mineros y la *proof of work* (prueba de trabajo).



A la izquierda, dos mineros. A la derecha, **Phineas Gage**. (imagen de Wikipedia Commons)

Minado de bloques

Resulta que para que un nuevo bloque sea considerado *válido* por toda la red, además de que todas sus transacciones sean válidas, y el bloque parta de un bloque también válido, tiene que atravesar una demostración de que, quien agregó ese bloque a la cadena, transpiró haciéndolo. En términos computacionales: tuvo que **invertir una cantidad significativa de poder de cómputo** para generar ese bloque. A eso lo llamamos tener una prueba de trabajo (*proof of work*).
Si pensamos que modificar un bloque implica modificar también todos los subsiguientes, vemos que modificar la cadena no es algo trivial, sino algo que requiere un esfuerzo más que considerable para recalcular la prueba de trabajo de todos los bloques afectados.
¿Cómo se construye dicha prueba? El mecanismo es súper ingenioso, y recurre una vez más a las funciones de hash y a la idea de que es imposible predecir la pinta que va a tener un hash viendo solamente el dato. La prueba de trabajo se construye

estableciendo que **un bloque es válido sólo si su hash tiene determinada pinta que nos interesa**.

Volviendo al ejemplo del hash de El Gato y La Caja de antes. Sólo nos van a interesar los hash de la frase "El Gato y la caja AÑO" (para distintos AÑOs) en los que el hash de la frase empiece con un cero. Por ejemplo, mirando abajo donde están los hash, el hash de "El gato y la caja 2018" empieza con un +, y el de "El gato y la caja 2019" empieza con una I.

Si queremos saber cuál es el próximo AÑO en el que vamos a tener un hash que nos resulte interesante (es decir, empiece con un cero), no nos queda otra que ir probando con 2018, 2019, y así sucesivamente. Resulta que, por cómo funciona el cálculo del hash, esto recién se cumple en el 2105 (y si queremos que empiece con un doble cero, recién en 3896).

hash de "El gato y la caja 2018"
= +OcHh3Oj/kh1b/9+qZMR8PE4wp/y7wM3p6n4Ok4HYEc=
hash de "El gato y la caja 2019"
= IFVvTrl6aesGPyrfssT8LzcMgIHTj8NBbz+S3bF7h6c=
hash de "El gato y la caja 2020"
= +CPJZXJaM3c4Fd4A0ljYIfaavPzjivkknZkoYf6vz9o=
...
hash de "El gato y la caja 2105"
= 0X/ZCJ2Ei1VNLegxL4tA98yFmNqLHDC3kXaLF3WtKyg=
...
hash de "El gato y la caja 3896"
= 00OtCB2q+Gp7lsnuCu86x1N6OHtxKAgtzLTobaRx7ng=

¿Cómo se relaciona esto con la prueba de trabajo? Pues bien, en el hash de un bloque había un término misterioso llamado *nonce*. Resulta que no es más que un número arbitrario que se puede cambiar y permite generar distintos hashes hasta obtener uno con la pinta deseada. Es decir, cumple la misma función que el AÑO en el ejemplo que veíamos.

Quien quiera generar un bloque nuevo, entonces, tiene que primero recolectar las transacciones a incluir en el bloque, y luego **calcular el hash del bloque con**

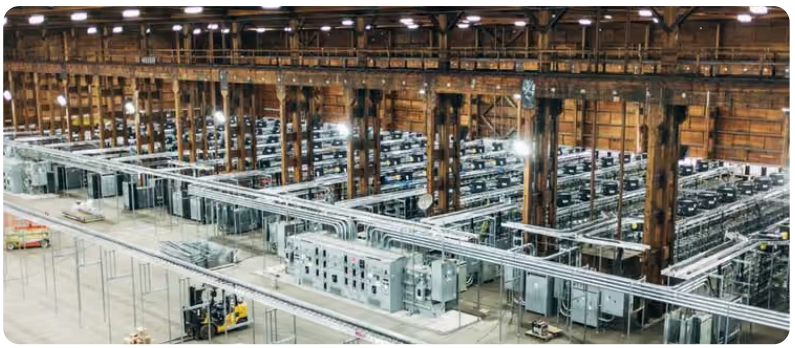
distintos *nonces* hasta dar con uno que empiece con cierta cantidad de ceros o con el criterio que esté buscando. La cantidad de ceros es llamada la 'dificultad' de encontrar un bloque, y es ajustada dinámicamente por la red de nodos para intentar mantener un intervalo de tiempo constante entre cada bloque. Así, la red asegura una generación a un ritmo constante de bloques.

Ahora bien, para que haya un incentivo para efectivamente ponerse a hacer tanta cuenta, la red 'premia' a quien logra agregar un bloque nuevo en la cadena. En el caso de *Bitcoin*, se otorga una recompensa (nada despreciable) en BTC, además de una comisión por cada transacción incluida en ese bloque. Esto hace que se genere una competencia para agregar un nuevo bloque a la cadena, a fines de obtener esa recompensa, entre todos los mineros de la red. Así, **los mineros no son más que nodos que participan en esta carrera para intentar agregar nuevos bloques.**

Siguiendo un poco más con *Bitcoin*, vale decir en este punto que las recompensas por bloque minado son la única manera en la que se emiten nuevos *Bitcoins*, y el protocolo está diseñado para que esa recompensa vaya decreciendo en el tiempo, hasta llegar a un tope preestablecido de 21 millones de BTC, a partir del cual el único incentivo de un minero estará dado por las comisiones asociadas a procesar cada transacción (pero con el sistema ya instalado y andando). Este tope permite también asegurar que no haya inflación por emisión continua, algo que es frecuente en la moneda de cualquier país.

Un dato colorido es que ya 3 de cada 4 bitcoins existentes están en circulación y que, por diseño, es progresivamente más y más difícil generar bitcoins nuevos. Por eso ser minero en 2012 era más fácil y redituable que serlo ahora, dato importante solamente para quienes pueden viajar al pasado.

Si bien en un principio era posible para cualquiera minar con una computadora hogareña, hoy la dificultad y la competencia en las *blockchains* más populares es tal que se necesita hardware especializado y muchísimas máquinas dedicadas, dando lugar a enormes granjas de minado, formadas por millones de máquinas intentando obtener el próximo bloque, consumiendo en total tanta energía como una nación entera.



Granja de bitcoins. Donde se minan bitcoins. Porque un minero trabaja en una granja, obvio.

Con esto en vista, y volviendo al ejemplo, si Roberto quiere reescribir parte de la historia, realmente no le será tarea fácil, considerando todo el de poder de cómputo que necesitaría para poder recalcular todos los bloques desde el modificado hasta el actual.

Sin embargo, queda aún un cabo suelto: Roberto podría simplemente reescribir el bloque que contiene la transacción que quiere eliminar, y anunciar a la red que la cadena de bloques termina ahí. Es decir, en lugar de reescribir toda los bloques hasta el actual, simplemente los descarta. Así, un nodo cualquiera que se conecte al nodo de Roberto para obtener una copia de la blockchain, estaría recibiendo la versión adulterada con menos bloques. Más aún, si ese nodo se conecta a otro, y ve una versión distinta de la *blockchain* con más bloques, ¿cómo sabe en cuál confiar?

En otras palabras: dadas dos versiones distintas de una misma cadena, ¿cómo es posible acordar cual es la verdadera sin tener un organismo autoritativo 'central' que lo determine?

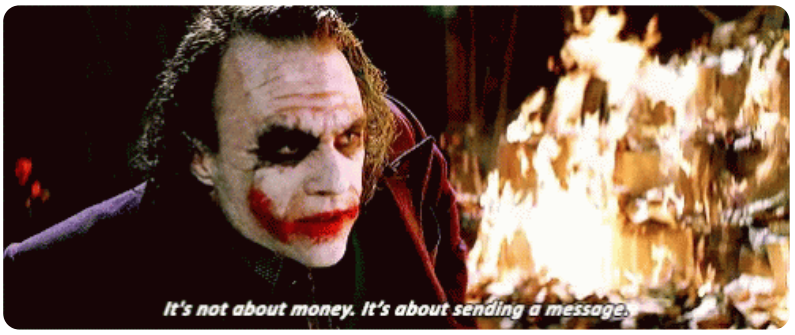
Y acá llega la última parte de la historia: el consenso de Nakamoto, llamado así en honor al creador (o creadora, o creadores, o creadoras) del *Bitcoin*. Las reglas del juego establecen que, dadas dos cadenas válidas, la red acuerda que la verdadera es aquella con más bloques o, más precisamente, **con más trabajo de cómputo acumulado**. Para decirlo mal, pronto y con dejos de colegio técnico estereotípico, 'gana la más larga'.

Esta regla obliga a Roberto a no sólo tener que reescribir su bloque, sino que también tiene que generar tantos bloques después como la cadena más larga que haya. Es decir, tiene que **competir contra todo el resto de la red**, para poder producir más rápido más bloques de los que haya al momento. Esto hace que una

transacción en el bloque más reciente pueda llegar a perderse, pero una vez que transcurrió una cierta cantidad de bloques (también llamado *confirmaciones* de la transacción), eso ya sea prácticamente imposible.

Pero esto no es sobre *Bitcoin* sino sobre *blockchain*, y no todas las *blockchains* funcionan por *prueba de trabajo*. Considerando la problemática ambiental causada por la cantidad de energía consumida en resolver cuentas que no aportan ningún beneficio más que el de validar un bloque, hay mucha investigación en usar otros mecanismos.

Un ejemplo es ***proof of stake*** (no confundir stake con steak o skate), que consiste en ir eligiendo un nodo distinto cada vez que va a ser el encargado de armar el siguiente bloque. El nodo tiene que 'bancar' (usaria 'dar fe', pero es idioma escribano) que el bloque que construyó sea correcto con una cierta cantidad de plata, y si la red descubre que hubo alguna trampa involucrada, esa plata se destruye (o *quema*) a modo de castigo.



Quemar una moneda digital no queda tan canchero como quemar la real, pero es más fácil.

Este mecanismo, que está en desarollo para ser implementado en la red *Ethereum* en una futura versión, no sólo reduce el consumo de energía, sino también la centralización de la red en unos pocos 'carteles' de mineros que pueden afrontar el gasto de mantener el hardware necesario para los cálculos que requiere la prueba de trabajo.

Blockchains on the wild

Pero *Bitcoin* no está sola, ***Ethereum*** es otro ejemplo conocido de *blockchain*. Esta red permite no sólo transaccionar una moneda (el ETH) sino también escribir

programas (marketineramente llamados *contratos inteligentes*) que son ejecutados de manera descentralizada por la propia *blockchain*. Esto permite implementar todo tipo de aplicaciones descentralizadas, que pueden ir desde una moneda inventada hasta gatitos digitales, sin necesidad de construir desde cero toda la infraestructura requerida para sostener una *blockchain*, sino utilizando *Ethereum* como plataforma de desarrollo. De hecho, muchos de los proyectos que *le-ponen-blockchain-a-todo* están en realidad construidos sobre esta red.

Otro ejemplo es ***Filecoin***, una red (aún en desarrollo) cuyo objetivo es proveer almacenamiento descentralizado de datos. Así, en lugar de guardar fotos en Google, podés pagarle a alguien (¡en *filecoins*!) para que los guarde por vos. Y al revés también: si tenés un disco que no usás, podés 'alquilarlo' a cambio de *filecoins*. Esta red no funciona con *proof of work* ni *proof of stake*, sino con *proof of storage*: a través de pruebas del guardado de los archivos.

Pero esto no es todo amigos, incluso hay empresas como IBM o Microsoft que proveen servicios de ***blockchains* privadas**, es decir, que sólo pueden participar usuarios con permisos para hacerlo. Algo menos extremo es una *blockchain* donde cualquiera puede participar, pero quienes construyen los bloques (el equivalente a los mineros en *Bitcoin* y otras *proof-of-work*) son un grupo cerrado. Un ejemplo de esto último es *Ripple*, un proyecto de blockchain para pagos globales entre entidades bancarias, en el que hay un conjunto de validadores designados por Ripple que se encargan de mantener el registro distribuido de la moneda.

En definitiva, hoy en día hay cientos de proyectos trabajando alrededor de *blockchain*, sea a modo de monedas digitales o para otras aplicaciones, que puedan beneficiarse de tener un seguimiento público y distribuido de cualquier cosa. Y así como hay proyectos que son puro humo, que se aprovechan de la burbuja para llevarse una parte de la torta, también hay muchos desarrollos muy interesantes buscando construir nuevas aplicaciones sobre esta tecnología, de la que todavía queda mucho por ver.

BONUS TRACK

Algunos proyectos copados sobre blockchain elegidos con un criterio absolutamente subjetivo.

- Redes de crédito, que facilitan préstamos entre pares sin bancos como intermediarios (ej WeTrust, Ripio Credit Network).
- Micropagos directos a los generadores del contenido que consumís online, eliminando los modelos centralizados de ads actuales (ej Basic Attention Token del navegador Brave, PROPS tokens de YouNow).
- Donaciones directas a proyectos sociales, con transparencia sobre el uso de los fondos (ej Giveth, Alice).
- Experimentos de *ingreso básico universal* via criptomonedas, otorgándole a cualquier persona una renta mínima regular (ej Circles, Grantcoin).
- Prototipos de monitoreo y seguimiento de productos desde su producción hasta su recepción por el cliente final, posiblemente en el otro lado del mundo (ej Shipchain).
- Pilotos para llevar un registro de la propiedad sobre *blockchain* en distintos países del mundo (ej Rusia, Bermuda, India, Suecia).
- Verificación de las ediciones del Boletín Oficial de la Nación, para evitar que puedan ser modificados a posteriori (fuente).
- Y un último, Bananacoin, incluido en esta lista solamente para terminar en un gran PONELE BLOCKHAIN A TODO.



Boom, la realidad supera a la ficción.

Referencias

Satoshi Nakamoto, **Bitcoin white paper**

elgatoylacaja.com/nestor-en-blockchain

Sumate en
eglc.ar/bancar